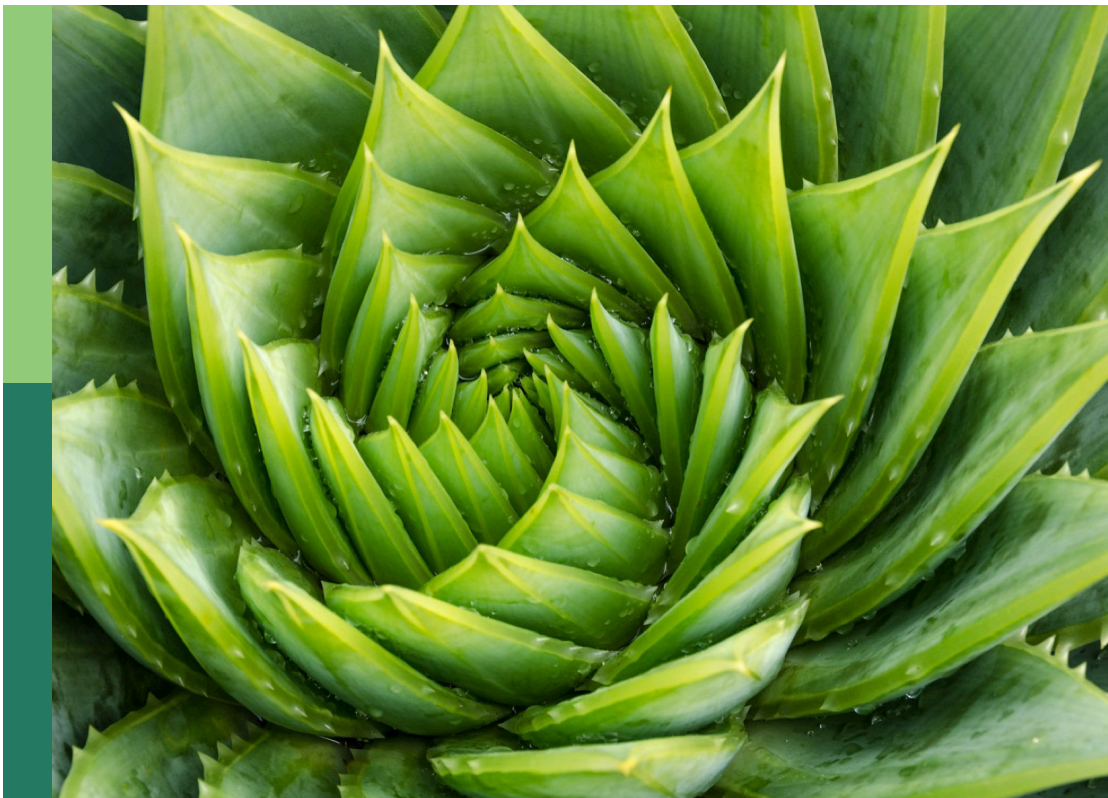
frontiers | Research Topics

# Advanced AI methods for plant disease and pest recognition

**Edited by**
Jucheng Yang, Yalin Wu, Alvaro Fuentes, Sook Yoon and Tonghai Liu

**Published in**
Frontiers in Plant Science



FRONTIERS EBOOK COPYRIGHT STATEMENT

The copyright in the text of individual articles in this ebook is the property of their respective authors or their respective institutions or funders. The copyright in graphics and images within each article may be subject to copyright of other parties. In both cases this is subject to a license granted to Frontiers.

The compilation of articles constituting this ebook is the property of Frontiers.

Each article within this ebook, and the ebook itself, are published under the most recent version of the Creative Commons CC-BY licence. The version current at the date of publication of this ebook is CC-BY 4.0. If the CC-BY licence is updated, the licence granted by Frontiers is automatically updated to the new version.

When exercising any right under the CC-BY licence, Frontiers must be attributed as the original publisher of the article or ebook, as applicable.

Authors have the responsibility of ensuring that any graphics or other materials which are the property of others may be included in the CC-BY licence, but this should be checked before relying on the CC-BY licence to reproduce those materials. Any copyright notices relating to those materials must be complied with.

Copyright and source acknowledgement notices may not be removed and must be displayed in any copy, derivative work or partial copy which includes the elements in question.

All copyright, and all rights therein, are protected by national and international copyright laws. The above represents a summary only. For further information please read Frontiers' Conditions for Website Use and Copyright Statement, and the applicable CC-BY licence.

ISSN 1664-8714
ISBN 978-2-8325-5009-0
DOI 10.3389/978-2-8325-5009-0

## About Frontiers

Frontiers is more than just an open access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

## Frontiers journal series

The Frontiers journal series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the *Frontiers journal series* operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

## Dedication to quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews. Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

## What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the *Frontiers journals series*: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area.

Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers editorial office: frontiersin.org/about/contact

# Advanced AI methods for plant disease and pest recognition

**Topic editors**

Jucheng Yang — Tianjin University of Science and Technology, China
Yalin Wu — Lushan Botanical Garden (CAS), China
Alvaro Fuentes — Jeonbuk National University, Republic of Korea
Sook Yoon — Mokpo National University, Republic of Korea
Tonghai Liu — Tianjin Agricultural University, China

**Citation**

Yang, J., Wu, Y., Fuentes, A., Yoon, S., Liu, T., eds. (2024). *Advanced AI methods for plant disease and pest recognition*. Lausanne: Frontiers Media SA.
doi: 10.3389/978-2-8325-5009-0

# Table of contents


05 Editorial: Advanced AI methods for plant disease and pest recognition
Jucheng Yang, Tonghai Liu, Sook Yoon, Alvaro Fuentes and Yalin Wu

08 Enhancing host-pathogen phenotyping dynamics: early detection of tomato bacterial diseases using hyperspectral point measurement and predictive modeling
Mafalda Reis Pereira, Filipe Neves dos Santos, Fernando Tavares and Mário Cunha

26 Local refinement mechanism for improved plant leaf segmentation in cluttered backgrounds
Ruihan Ma, Alvaro Fuentes, Sook Yoon, Woon Yong Lee, Sang Cheol Kim, Hyongsuk Kim and Dong Sun Park

43 Construction and verification of machine vision algorithm model based on apple leaf disease images
Gao Ang, Ren Han, Song Yuepeng, Ren Longlong, Zhang Yue and Han Xiang

56 MixSeg: a lightweight and accurate mix structure network for semantic segmentation of apple leaf disease in complex environments
Bibo Lu, Jiangwen Lu, Xinchao Xu and Yuxin Jin

72 Embracing limited and imperfect training datasets: opportunities and challenges in plant disease recognition using deep learning
Mingle Xu, Hyongsuk Kim, Jucheng Yang, Alvaro Fuentes, Yao Meng, Sook Yoon, Taehyun Kim and Dong Sun Park

86 A tea bud segmentation, detection and picking point localization based on the MDY7-3PTB model
Fenyun Zhang, Hongwei Sun, Shuang Xie, Chunwang Dong, You Li, Yiting Xu, Zhengwei Zhang and Fengnong Chen

104 A New Deep Learning-based Dynamic Paradigm Towards Open-World Plant Disease Detection
Jiuqing Dong, Alvaro Fuentes, Sook Yoon, Hyongsuk Kim, Yongchae Jeong and Dong Sun Park

122 An iterative noisy annotation correction model for robust plant disease detection
Jiuqing Dong, Alvaro Fuentes, Sook Yoon, Hyongsuk Kim and Dong Sun Park

136 A ResNet50-DPA model for tomato leaf disease identification
Jin Liang and Wenping Jiang

152 Xoo-YOLO: a detection method for wild rice bacterial blight in the field from the perspective of unmanned aerial vehicles
Pan Pan, Wenlong Guo, Xiaoming Zheng, Lin Hu, Guomin Zhou and Jianhua Zhang

166 Convolutional neural network in rice disease recognition: accuracy, speed and lightweight
Hongwei Ning, Sheng Liu, Qifei Zhu and Teng Zhou

182 Zero-exemplar deep continual learning for crop disease recognition: a study of total variation attention regularization in vision transformers
Boyuan Wang

198 A novel ensemble learning method for crop leaf disease recognition
Yun He, Guangchuan Zhang and Quan Gao

214 A series of methods incorporating deep learning and computer vision techniques in the study of fruit fly (Diptera: Tephritidae) regurgitation
Tongzhou Zhou, Wei Zhan and Mengyuan Xiong

225 YOLOV5-CBAM-C3TR: an optimized model based on transformer module and attention mechanism for apple leaf disease detection
Meng Lv and Wen-Hao Su

238 DIC-Transformer: interpretation of plant disease classification results using image caption generation technology
Qingtian Zeng, Jian Sun and Shansong Wang

259 Real-time and lightweight detection of grape diseases based on Fusion Transformer YOLO
Yifan Liu, Qiudong Yu and Shuze Geng

274 RAAWC-UNet: an apple leaf and disease segmentation method based on residual attention and atrous spatial pyramid pooling improved UNet with weight compression loss
Jianlong Wang, Junhao Jia, Yake Zhang, Haotian Wang and Shisong Zhu

293 A deep semantic network-based image segmentation of soybean rust pathogens
Yalin Wu, Zhuobin Xi, Fen Liu, Weiming Hu, Hongjuan Feng and Qinjian Zhang

306 A deep multi-task learning approach to identifying mummy berry infection sites, the disease stage, and severity
Hongchun Qu, Chaofang Zheng, Hao Ji, Rui Huang, Dianwen Wei, Seanna Annis and Francis Drummond

327 A few-shot learning method for tobacco abnormality identification
Hong Lin, Zhenping Qiang, Rita Tse, Su-Kit Tang and Giovanni Pau

frontiers | Frontiers in Plant Science

TYPE Editorial
PUBLISHED 30 May 2024
DOI 10.3389/fpls.2024.1434320

Check for updates

OPEN ACCESS

EDITED AND REVIEWED BY
Lei Shu,
Nanjing Agricultural University, China

*CORRESPONDENCE
Jucheng Yang
jcyang@tust.edu.cn

RECEIVED 17 May 2024
ACCEPTED 22 May 2024
PUBLISHED 30 May 2024

CITATION
Yang J, Liu T, Yoon S, Fuentes A and Wu Y (2024) Editorial: Advanced AI methods for plant disease and pest recognition.
*Front. Plant Sci.* 15:1434320.
doi: 10.3389/fpls.2024.1434320

COPYRIGHT
© 2024 Yang, Liu, Yoon, Fuentes and Wu. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# Editorial: Advanced AI methods for plant disease and pest recognition

Jucheng Yang[1]*, Tonghai Liu[2], Sook Yoon[3], Alvaro Fuentes[4,5] and Yalin Wu[6]

[1]College of Artificial Intelligence, Tianjin University of Science and Technology, Tianjin, China, [2]College of Computer and Information Engineering, Tianjin Agricultural University, Tianjin, China, [3]Department of Computer Engineering, Mokpo National University, Mokpo, Republic of Korea, [4]Department of Electronics Engineering, Jeonbuk National University, Jeonju, Republic of Korea, [5]Core Research Institute of Intelligent Robots, Jeonbuk National University, Jeonju, Republic of Korea, [6]Lushan Botanical Garden, Chinese Academy of Sciences, Jiujiang, China

KEYWORDS

**plant disease, pest recognition, artificial intelligence, deep learning, disease identification, disease detection, disease segmentation**

Editorial on the Research Topic
Advanced AI methods for plant disease and pest recognition

Plant diseases and pests cause significant losses to farmers and threaten food security worldwide. Monitoring the growing conditions of crops and detecting plant diseases is critical for sustainable agriculture. Traditionally, crop inspection has been carried out by people with expert knowledge in the field. However, regarding any activity carried out by humans, this activity is prone to errors, leading to possible incorrect decisions. Innovation is, therefore, an essential fact of modern agriculture. In this context, deep learning has played a key role in solving complicated applications with increasing accuracy over time, and recent interest in this type of technology has prompted its potential application to address complex problems in agriculture, such as plant disease and pest recognition. Although substantial progress has been made in the area, several challenges remain, especially those that limit systems to operate in real-world scenarios. This Research Topic aims to explore recent advanced AI methods for plant disease and pest recognition for real-world applications.

In this Research Topic, a total of 21 papers have been published, encompassing the contributions of 84 distinct authors. The content primarily delves into the realms of disease detection and identification pertaining to crops such as tomatoes, peppers, rice, corn, soybeans, alongside fruits including apples, grapes, and blueberries. Furthermore, the scope encompasses the automation of harvesting processes and anomaly detection within economically significant crops like tea and tobacco. Notably, there are also inquiries into the behavioral patterns of Diptera Tephritidae.

The focus of these studies centers on the identification, detection, and segmentation of plant diseases, as well as aspects related to harvesting, pests, and growth monitoring. Among the research papers, 18 are dedicated to the study of plant diseases, while the remaining cover various other topics. The primary subjects of investigation include leaves, fruits, and flowers of plants, with additional examinations into pests and fungi. Specifically,

there are 17 papers primarily focused on leaf analysis, 3 on fruit analysis, and 1 each on pests, flowers, and fungi. From this, it is evident that the prevailing research methodologies predominantly focus on the recognition of plant diseases through leaf analysis. From a task-oriented perspective, the requirements for identification tend to be idealized, often limited to utilizing images containing a single leaf. However, as research and applications progress, achieving segmentation, detection, and optimized deployment in real-world scenarios becomes increasingly crucial. Consequently, there is a burgeoning interest in exploring the vast research potential surrounding fruits, flowers, pests, and fungi.

In identification tasks, the primary challenge lies in the feature extraction capacity of models, mainly due to the visual resemblance of different diseases across various crops. This challenge is exacerbated by the incompleteness of datasets and resource constraints during deployment. The ResNet50-DPA model (Liang and Jiang) addresses this challenge by enhancing the network structure with attention mechanisms, thereby augmenting the backbone network's capability in extracting image features. On the other hand, attention is directed towards improving the network's classification head in the AD approach. He et al. achieved this by training multiple models tailored to different disease recognition tasks and refining the classification results through the optimization of ensemble weights. Plant disease datasets often suffer from sample scarcity, domain shifts, and the presence of unknown classes. Lin et al. addressed the issues of sample scarcity and domain shifts through few-shot learning, while Wang tackled the problem of unknown classes using zero-shot learning. Zeng et al., on the other hand, proposed a multimodal approach. They leveraged image features to generate textual descriptions of diseases, thereby facilitating model training. This not only enhanced the quality of extracted features but also alleviated the scarcity of samples in the dataset. In-depth studies, such as Qu et al., have delved into establishing severity grading and stage progression of diseases. This approach provides valuable references for the practical deployment of fine models in specialized fields. Ning et al. reviewed the performance of lightweight models in rice disease recognition. Furthermore, Ang et al. employed structural reparameterization techniques to reduce model parameter size, facilitating practical deployment.

In segmentation tasks, the primary challenges stem from the difficulty in distinguishing between background and foreground and the presence of small yet densely packed objects. Most of these studies rely on meticulously collected and annotated datasets, which are essential prerequisites for addressing the challenges. However, various methods have been proposed to tackle these challenges. To segment tea leaf buds, Zhang et al. enhances model attention and further computes picking points for the buds during post-processing. Additionally, Ma et al. employs a series of preprocessing techniques such as Gaussian filtering to refine the image background, thereby reducing the noise that the model needs to handle. Leveraging a blend of CNN and Transformer models, Lu et al. achieved fine-grained semantic segmentation by harnessing the strengths of each approach. Building upon the UNet model, Wang et al. utilized multi-scale features to achieve granular segmentation.

In detection tasks, the primary challenges continue to arise from dataset limitations. This is because the task closely simulates real farm environments, leading to issues such as unknown classes, annotation errors, and labeling inaccuracies within the dataset. Additionally, the challenge of detecting small yet densely packed objects persist. Dong et al. proposed two approaches to address these challenges. Firstly, they introduce a method that utilizes teacher-student networks for self-supervised repair of imprecise and incomplete annotations. Secondly, they present an Open-World detection method (Dong, et al.) that combines incremental learning with open-set detection. These approaches address the major drawbacks of imperfect datasets and provide valuable insights for deployments closer to real-world applications. Similarly to segmentation tasks, methods involving attention mechanisms (Liu et al.; Lv and Su) are employed to ensure the model's capability in detecting small and densely packed targets. Moreover, perspective correction of detection models (Pan et al.) based on drone-captured imagery offers a distinct research perspective for practical deployment.

Some innovative research endeavors also demonstrate immense potential. Reis Pereira et al., for instance, cultivated diseased plants in a controlled laboratory environment and utilizes hyperspectral imaging to predict asymptomatic diseases in advance based on disease stage. Furthermore, Wu et al. conducted semantic segmentation of fungal pathogens causing diseases. While the aforementioned detections cannot be accomplished using conventional cameras, their profound exploration of diseases provides crucial reference points for subsequent research. Insect pest detection represents another vital research direction. Zhou et al. shifted its focus towards insects, analyzing pest generation through detection, segmentation, and tracking of insect behavior.

In conclusion, regardless of identification, segmentation, or detection tasks, dataset limitations remain the primary bottleneck in the development of advanced AI methods for plant disease and pest recognition. As articulated in the perspective article (Xu et al.), imperfect datasets always entail additional risks and challenges. However, effectively leveraging these datasets can still reduce costs and enhance efficiency. For Research Topics in plant disease and pest recognition that rely heavily on domain expertise, further refining the objectives of recognition and expanding the applicability of models to minimize data dependency in real-world environments are two key objectives for future development.

## Author contributions

JY: Writing – review & editing, Writing – original draft, Supervision, Conceptualization. TL: Writing – original draft, Formal Analysis, Data curation. SY: Writing – original draft, Validation, Investigation, Conceptualization. AF: Writing – original draft, Methodology, Investigation. YW: Writing – original draft, Validation, Data curation.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The author(s) declared that they were an editorial board member of Frontiers, at the time of submission. This had no impact on the peer review process and the final decision.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 16 August 2023
DOI 10.3389/fpls.2023.1242201

Check for updates

OPEN ACCESS

EDITED BY
Tonghai Liu,
Tianjin Agricultural University, China

REVIEWED BY
Arpit Jain,
Consultant, Bengaluru, India
Bin Liu,
Northwest A&F University, China

*CORRESPONDENCE
Mário Cunha
mccunha@fc.up.pt

RECEIVED 18 June 2023
ACCEPTED 27 July 2023
PUBLISHED 16 August 2023

CITATION
Reis Pereira M, Santos FNd, Tavares F and Cunha M (2023) Enhancing host-pathogen phenotyping dynamics: early detection of tomato bacterial diseases using hyperspectral point measurement and predictive modeling.
*Front. Plant Sci.* 14:1242201.
doi: 10.3389/fpls.2023.1242201

COPYRIGHT
© 2023 Reis Pereira, Santos, Tavares and Cunha. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# Enhancing host-pathogen phenotyping dynamics: early detection of tomato bacterial diseases using hyperspectral point measurement and predictive modeling

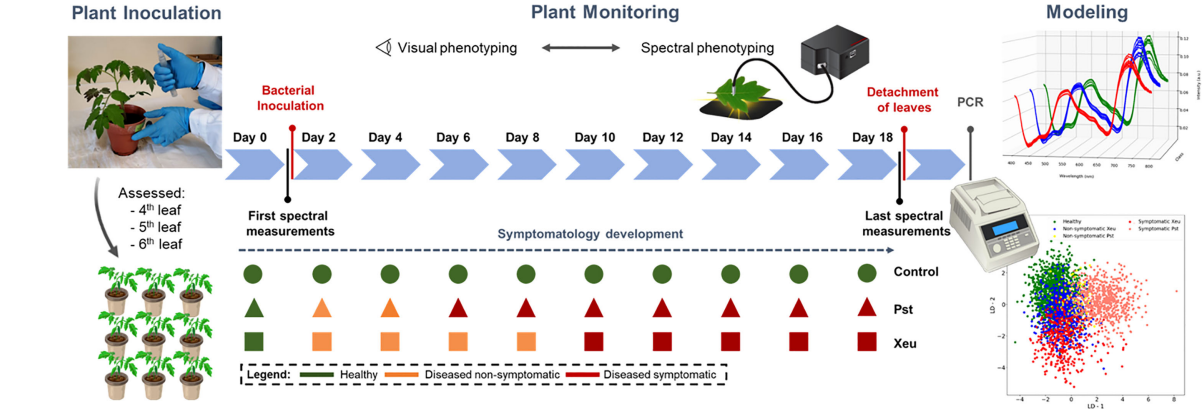
Mafalda Reis Pereira[1,2], Filipe Neves dos Santos[2], Fernando Tavares[1,3,4] and Mário Cunha[1,2]*

[1]Faculdade de Ciências da Universidade do Porto (FCUP), Rua do Campo Alegre, Porto, Portugal, [2]Institute for Systems and Computer Engineering, Technology and Science (INESC TEC), Campus da Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, Porto, Portugal, [3]CIBIO, Centro de Investigação em Biodiversidade e Recursos Genéticos, InBIO Laboratório Associado, Campus de Vairão, Universidade do Porto, Vairão, Portugal, [4]BIOPOLIS Program in Genomics, Biodiversity and Land Planning, CIBIO, Campus de Vairão, Vairão, Portugal

Early diagnosis of plant diseases is needed to promote sustainable plant protection strategies. Applied predictive modeling over hyperspectral spectroscopy (HS) data can be an effective, fast, cost-effective approach for improving plant disease diagnosis. This study aimed to investigate the potential of HS point-of-measurement (POM) data for in-situ, non-destructive diagnosis of tomato bacterial speck caused by *Pseudomonas syringae* pv. *tomato* (Pst), and bacterial spot, caused by *Xanthomonas euvesicatoria* (Xeu), on leaves (cv. cherry). Bacterial artificial infection was performed on tomato plants at the same phenological stage. A sensing system composed by a hyperspectral spectrometer, a transmission optical fiber bundle with a slitted probe and a white light source were used for spectral data acquisition, allowing the assessment of 3478 spectral points. An applied predictive classification model was developed, consisting of a normalizing pre-processing strategy allied with a Linear Discriminant Analysis (LDA) for reducing data dimensionality and a supervised machine learning algorithm (Support Vector Machine – SVM) for the classification task. The predicted model achieved classification accuracies of 100% and 74% for Pst and Xeu test set assessments, respectively, before symptom appearance. Model predictions were coherent with host-pathogen interactions mentioned in the literature (e.g., changes in photosynthetic pigment levels, production of bacterial-specific molecules, and activation of plants' defense mechanisms). Furthermore, these results were coherent with visual phenotyping inspection and PCR results. The reported outcomes support the application of spectral point measurements acquired *in-vivo* for plant disease diagnosis, aiming for more precise and eco-friendly phytosanitary approaches.

KEYWORDS

**plant disease diagnosis, early diagnosis, proximal sensing, hyperspectral spectroscopy, point of measurement, applied aredictive modeling, linear discriminant analysis, machine learning**




GRAPHICAL ABSTRACT

## 1 Introduction

The tomato (*Solanum lycopersicum* L.) crop holds great importance worldwide due to its significant impact on agriculture, the economy, and human nutrition. This globally cultivated vegetable crop is very sensitive to diseases leading to dramatic yield and economic losses (Blancard, 2012). Bacterial diseases of tomato plants caused by the Gram-negative bacteria *Pseudomonas syringae* pv. *tomato* (Pst, bacterial speck) and *Xanthomonas euvesicatoria* (Xeu) formerly known as *Xanthomonas campestris* pv. *vesicatoria*, bacterial spot, are two important etiological agents responsible for several plant outbreaks and considerable losses in tomato production worldwide. These two diseases are responsible for severe alterations in the host physiology, biochemistry, and structural composition, causing plant phenotype modifications (e.g., reduction of the photosynthetic capacity of diseased foliage, defoliation, flower abortion, and fruit lesions, among others). Ultimately, they result in yield reductions due to the damage caused to plants and fruits, which makes them unsuitable for the fresh market or processing. Control measures for these two crop diseases may be ineffective, especially when the bacteria are well-established in a production site (medium to late stage of the disease infection process. Phytosanitary products, such as copper and antibiotics (Alves et al., 2023), can be applied to mitigate the negative effects of the disease. Nevertheless, this approach can lead to bacteria tolerance to phytosanitary compounds (Blancard, 2012), and conduct to considerable damage to the environment and food security due to non-targeted applications of these products (Zhang et al., 2020).

Nowadays, bacterial diseases are diagnosed essentially through scouting and 'wet lab' -based approaches. The first requires a careful and detailed inspection of crop fields (usually visual) by specialized trained observers. They must detect and identify diseased plants based on modifications to the characteristic phenotype of the crop, and the presence of disease symptoms (Parker et al., 1995). Thus, it is subjective, error-prone (as symptoms alone are not entirely disease-specific, and can be promoted by other biotic and abiotic stresses), labor-intensive, time-consuming, and expensive (Mahlein, 2016). In turn, laboratory-based techniques consist of serological and molecular assays, frequently applied due to their sensitivity, accuracy, and effectiveness. The most widespread lab methods include Enzyme-Linked Immunosorbent Assay (ELISA) and Polymerase Chain Reaction (PCR) methods. They involve comprehensive sampling procedures, which require several hours to be completed, and destructive sample preparation, precluding the accompaniment of disease development nor its field mapping to support precision agriculture systems (e.g. Site-Specific Management) (Fang and Ramasamy, 2015; Martinelli et al., 2015). Nevertheless, laboratory-based approaches lack appropriate high throughput and speed for supporting real-time agronomic precision decisions in-field since they were developed to verify the presence of pathogens. They also still have some diagnostic constraints, mostly in the non-symptomatic and early disease infection stages, related to the irregular spread of bacteria inside plants (Fang and Ramasamy, 2015; Martinelli et al., 2015).

Hyperspectral spectroscopy (HS) is one innovative approach that has been studied and successfully applied to assess different plant(host)-pathogen interactions in a fast, sensitive, standardized cost-effective, high-throughput, and non-invasive way (Golhani et al., 2018). Through spectral measurements in the visible (VIS, 400-700 nm) and infrared (IR, 800-2500 nm) regions, HS showed the capability of effectively assessing a wide variety of plant structural, chemical, biophysical, and metabolic traits in living tissues (Thenkabail et al., 2000; Delalieux et al., 2007). Changes in the typical spectral phenotype of a crop may indicate deviations in its health status, leading to an indirect method of diagnosing diseases. Plant-pathogen interactions shift plant metabolism and tissue composition, resulting in detectable variations in the plant's optical behavior. In brief, these dynamics typically promote modifications in the VIS spectra of plants, due to changes in pigments' concentration and physiological processes. Furthermore, variations in the IR region may also occur and are essentially linked to leaf water levels, chemical compounds (namely lignin's and proteins content), structural elements, and internal scattering processes (Thenkabail et al., 2014; Tosin et al., 2022).

Different types of pathogens, such as pests (Herrmann et al., 2017; Zhang et al., 2017), fungi (Yu et al., 2018; Skoneczny et al., 2020),

bacteria (Bagheri et al., 2018), and viruses (Morellos et al., 2020) affecting different crops have already been detected using the HS technique, mostly in symptomatic stages. Thus, this spectral phenotyping technique constitutes an interesting diagnosis method, allowing the distinction between the spectral signature of healthy and disease tissues, as well as between the spectral signature of diseased tissues infected with different pathogens.HS holds great potential for early disease diagnosis, i.e., when plants are diseased but still don't manifest any visual symptoms of the infection (Gold et al., 2020a; Reis-Pereira et al., 2022). However, the use of this approach for non-symptomatic plant disease diagnosis remains largely unexplored. Understanding host-pathogen specific interactions and overcoming technical challenges related to the biophysical status of infected plants, organ of the plant assessed, sensing technology, data processing, and modeling approaches is essential for the effective application of HS *in vivo* crop disease diagnosis (Mahlein et al., 2018). Addressing these challenges is crucial for real-time monitoring of disease progression.

The most used sensing devices for plant disease detection are non-imaging (e.g., point-of-measurement, POM) and imaging sensors. In POM sensing, light usually enters the leaf, and undergoes internal reflections conditioned by tissue structures and composition status. Thus, this technique can indirectly infer certain internal tissue characteristics affected by the host-pathogen interaction. POM sensors are typically designed to measure specific parameters without being significantly affected by factors like lighting conditions or surface textures. This reduces the potential for external interference and ensures more accurate and consistent measurements. This allied with their higher spectral resolution, cost-effectiveness, compactness, and reduced data processing requirements, makes them an attractive option for plant studies (Martins et al., 2022).

Spectral information provided by HS is extremely valuable, nonetheless, in biological tissues, it is super-imposed in the recorded spectra at different scales of interference (Barroso et al., 2022; Tosin et al., 2022). Moreover, HS data can present substantial amounts of redundant information in contiguous wavelengths, and just some specific spectral features might be relevant to predict and classify diseased tissues (Caicedo et al., 2014; Rivera et al., 2014). Applied predictive classification modeling strategies can be developed to study spectral data and extract useful information. Diverse approaches of data correction and pre-processing (e.g., data scaling and normalization) can be computed to reduce undesired spectral effects, such as 'noise' and scattering effects. Additionally, modeling strategies, as well as feature selection (FS), feature extraction and dimensionality reduction techniques (DR), can be useful for determining the wavelength features which have more influence in disease discrimination (Mahlein et al., 2010; Ahmadi et al., 2017). In plant disease research, different predictive approaches using HS data have been explored to classify tissues affected by biotic stress, considering all the spectral features or only specific variables, designated by FS or DR techniques (Gold et al., 2020b; Meng et al., 2020). Nevertheless, there is a lack of standardized protocols for acquiring hyperspectral data from tomato leaves. Different studies employ various acquisition setups, lighting conditions, and preprocessing techniques, making comparing and integrating findings challenging.

This work addresses the main technological challenges for efficiently applying hyperspectral technologies in phenotyping to diagnose plant diseases. Conducting analysis for healthy and bacterial inoculated plants over time, this study aims i) to compare visual phenotyping against spectral phenotyping based on the hyperspectral point-of-measurement (HS-POM) for healthy and diseased tomato leaflets, ii) to evaluate the HS-POM ability to accurately classify samples at various stages of disease development, including those without any visible symptoms and iii) distinguish the etiological agents of distinct tomato bacterial diseases. The specific goals include developing an applied predictive modeling strategy (combining data pre-processing, dimensionality reduction, and a supervised machine learning algorithm) for tomato bacterial disease classification and establishing causal relationships between plant health status, specific spectra characteristics, and the physiological changes that occur during infection dynamics to advance theoretical knowledge and provide a foundation for further research.

# 2 Materials and methods

## 2.1 Bacterial inoculation and plant growth

### 2.1.1 Inoculation on tomato leaflets

Tomato (*Solanum lycopersicum* L.) plants of the cultivar Cherry were grown in 200 mL pots containing a commercial potting substrate, in a walk-in plant growth chamber under controlled conditions (25-27 °C, humidity of approximately 60%, photoperiod of 12/12 h and light intensity 30W). Plants were divided into three groups of three plants each (nine plants in total), being a) one group of plants inoculated with *Pseudomonas syringae* pv. *tomato* DC 3000 (Pst) bacteria, b) a second group of plants inoculated with *Xanthomonas euvesicatoria* LMG 905 (Xeu) bacteria, and c) a third group of plants was treated with sterile distilled water only (Control group) (Figure 1). Plants were physically separated to avoid cross-contamination.

Plants were inoculated in the laboratory, at the growth stage of 5-6 fully expanded leaves, by spraying until they became fully wet, and run-off occurred. The bacterial suspensions used for these inoculation assays consisted of 1 x $10^8$ cells/mL. They were prepared from 48-h-old cultures of Pst grown in KB medium (peptone, 20.0g; $K_2HPO^4$, 1.5g; $MgSO^4$, 1.5g; glycerol, 10 mL; agar, 15g; distilled water up to 1.0 liter), and of Xeu cultures grown in YDC medium (yeast extract, 10.0g; dextrose, 20.0g; $CaCO_3$, 20.0g; agar, 15.0g; distilled water up to 1.0 liter). The inoculated plants were then covered with transparent polythene bags for 48 h to increase the relative humidity that fosters bacterial entry into plant tissues through natural openings such as stomata (Lamichhane, 2015). Plants were monitored daily for symptom development for 18 days (Figure 1).

During the inoculation period, to verify if the bacteria cultures used in these inoculation tests were viable, 20 µL of Pst solution and 20 µL of Xeu solution were cultured in Petri dishes containing KB and YDC media, respectively. After 48 h was possible to observe the bacteria growth in both nutrient media, proving that bacteria were viable at the moment of inoculation.

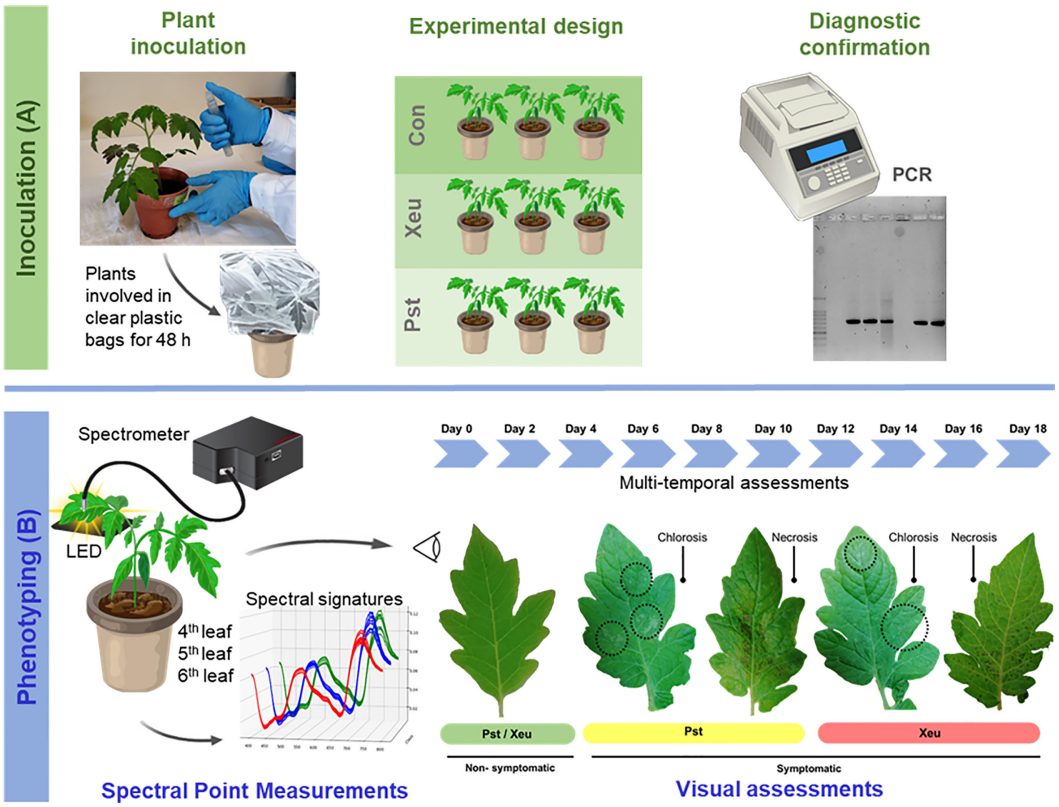



FIGURE 1
Experimental setup of the bacterial inoculation assay performed on tomato leaves **(A)**, and visual and spectral assessments (of the 4th, 5th, and 6th leaves) made in a dark room **(B)**. Spectral measurements were performed on the adaxial side of leaflets, using a spectrometer combined with an optical fiber bundle with a reflection probe. A white LED was placed beneath each leaflet. Both visual and spectral assessments were made 18 Days After Inoculation (DAI), collecting leaflets' spectral signatures and registering modifications in their phenotype (e.g., the appearance of the first symptoms, both chlorosis and necrosis).

### 2.1.2 Bacterial isolation from diseased leaflets

After the last spectral measurement, sample preparation for bacterial isolation was performed for all the leaflets. Leaflets were excised from plants using a sterile scalpel (Fernandes et al., 2017). Bacterial isolation was carried out as defined by Fernandes et al. (2017; 2021). Briefly, each sample of excised leaflet tissue was disinfected by immersion in 70% ethanol followed by washing with sterile distilled water (SDW), and then macerated with SDW in extraction bags. The suspensions obtained, and corresponding dilutions, were streaked on KB (samples inoculated with Pst bacteria), and on YDC medium (samples infected with Xeu pathogen). Characteristic colonies from these two bacteria species (milky white colonies in the case of Pst, and mucoid yellow colonies in the case of Xeu) were selected for growth on fresh nutrient agar medium to ensure purity.

Pst characteristic symptoms resemble small greasy dark stains (circular or slightly angular), that become brown to black, and appear randomly on the leaflets (often on the youngest or the ones located at the edge of the canopy plant). These lesions may typically show a yellow halo of various sizes. They are about 2–3 mm and can develop and coalesce (especially in the presence of moisture), affecting large areas of the leaf, that may later become necrotic and desiccate (Blancard, 2012). In turn, Xeu characteristic symptoms comprise small, circle, or slightly angular, translucent, and water-soaked lesions, which turn brown with time. They appear randomly in leaflets, and eventually become necrotic spots, with light gray centers and dark margins, which also can become surrounded by a yellow hallow with time. Smaller lesions can coalesce into each other forming larger injuries, whose diameter can range from 2 to 3 mm. In severe cases, tissues in the center of a lesion become dry and fall out, leading to "shot-hole" symptoms (Ritchie, 2000; Blancard, 2012).

### 2.1.3 Colony PCR protocol

A colony PCR was performed to validate the presence of both bacteria species on tomato leaflets isolates. PST2 (Vieira et al., 2007) and XV14 (Albuquerque et al., 2012) were the chosen markers, for Pst and Xeu, respectively, with amplicon lengths of 200, and 713 bp, correspondingly. A 20 µL PCR reaction mix consisted of 1 × DreamTaq Buffer (ThermoFisher Scientific, Waltham, MA, USA), 0.2 mM of each deoxynucleotide triphosphate (dNTP) (Grisp, Porto, Portugal), 0.2 mM of each forward and reverse primers, 1 U of DreamTaq DNA Polymerase (ThermoFisher Scientific, Waltham, MA, USA) and 10 µL of DNA isolate solution. Sterile distilled water was used as the negative control. PCR cycling parameters were defined as stated by Vieira et al. (2007) for Pst, and Albuquerque et al. (2012) for Xeu. PCR products were then separated by electrophoresis on a 0.8% agarose gel (1 × TAE buffer)

and visualized using Xpert Green DNA stain (Grisp, Porto, Portugal) with a Molecular Imager Gel Doc XR+ System (Bio-Rad, Hercules, CA, USA).

## 2.2 Spectral measurements *in vivo* tissue

### 2.2.1 Experimental setup for plant spectral acquisition

Figure 1 presents the main procedures for spectral measurements in the experimental setup. Hyperspectral point-of-measurements (HS-POM) were collected *in vivo* from the adaxial side of healthy and diseased leaflets of the nine tomato plants in the study, in a dark room. For each plant, spectral assessments were performed randomly on nine points of different leaflets, belonging to the 4$^{th}$, 5$^{th}$, and 6$^{th}$ expanded leaflets.

Hyperspectral data were acquired using a Hamamatsu Photonics K.K. TM Series C11697MB spectrometer, which covers a wavelength range of 200-1100 nm with a spectral resolution of 0.6 nm. A transmission optical fiber bundle (FCR-7UVIR200-2-45-BX, Avantes, Eerbeek, The Netherlands) with a range of 200-2500 nm was used along with a stainless-steel slitted reflection probe that was positioned 0.5 cm above the sample surface to capture the leaflet's spectral signal and direct it to the spectrometer's entrance lens. A white LED light was placed underneath the leaflet to provide uniform illumination to its entire abaxial surface. The spectral range of the LED emits light from 390 to 800 nm. Therefore, the LED spectra were used as a reference to the spectral range measured by the spectrophotometer and to check measurement and light emission stability (Figure 1B). The hyperspectral data were collected using specialized evaluation software (SpecEvaluationUSB2.exe, Hamamatsu Photonics K.K., Japan).

### 2.2.2 Preprocessing hyperspectral data

The performance of the modeling approach in detecting bacterial diseases in tomato leaflets was assessed using only the spectral region of 400 to 800 nm, approximately. This decision was based on the spectral wavelength range of the light LED source used (where possible useful information could be retrieved) and due to the observation of spectral noise near the limits of the equipment's spectral range, which could negatively affect the performance of the classification process. Therefore, a total of 944 features (wavelength) were used in the development of the prediction modeling (Figure 2).

Preprocessing data was performed following spectra normalization (Figures 2, 3). This approach aimed to standardize the data to a common scale, enabling meaningful comparison and analysis across different scenes or datasets. Additionally, it aims to decrease spectral signal oscillations, related to measurement equipment specifications (including devices' internal noise), variations in data assessment conditions (comprising differences in global spectral trend, total energy, high-frequency noise, and/or local background) (Randolph, 2006), associated to changes in environmental conditions or induced by the operator in the moment of assessment (e.g. variations in sample-sensor distance, uneven illumination conditions, choice of leaflets sample point location, appropriate scan parameters, spectral calibration, among others). This results in model abilities improvement by aiding in class separation (Randolph, 2006; Guezenoc et al., 2019). Furthermore, this process enables the elimination of the spectral response of both the sensor and light source, making possible the transfer of the acquired classifier to a different sensing device. Spectral data retrieved from measurements in tomato leaflets $S(\lambda_n)_m^{raw}$ were normalized through their division by the white LED source spectral signature $S(\lambda_n)^{reference}$ (considering the time of exposure of the spectral measurements), through the computation of the following forming (Equation 1):

$$S(\lambda_n)_m^{normalized} = S(\lambda_n)_m^{raw} / S(\lambda_n)^{reference} \quad (1)$$

## 2.3 Modeling leaflets symptomatology over time

### 2.3.1 Data set structure

Seeking bacterial tomato disease classification, spectral signatures from leaflets were then grouped to perform an applied predictive modeling approach related to the plants' experimental condition. Leaflets were classified according to the plant treatment group and their health status, including the classes: i) healthy, including all the measurements which were performed before bacteria inoculation, and the remaining assessments that were made in non-inoculated plants considered as control plants; ii) non-symptomatic Pst; iii) non-symptomatic Xeu; iv) symptomatic Pst; and v) symptomatic Xeu. All the spectral data collected from tomato leaflets on different dates were unified in a single classification model (Figures 1, 2).

Data classification was, thus, performed seeking the unraveling of spectral phenotyping differences between i) healthy and non-symptomatic diseased tissues (early diagnosis), ii) healthy and diseased tissues (showing visual modifications due to changes in chemical and structural composition), ii) healthy and diseased tissues affected by different bacterial etiological agents (which present distinct host-pathogen specific interactions), iii) and diseased tissues infected by different bacteria species (responsible for causing similar visual symptoms but showing different pathogenic dynamics).

### 2.3.2 Dimensionality reduction of spectral data

Multi-scale interference in plants' tissue promotes superimposition on hyperspectral data, resulting in autocorrelations in their spectral signal at several scales (Martins et al., 2022). To mitigate the effects of high dimensional, redundant information, several methodologies have been cited in the state-of-the-art, including dimensionality reduction (DR) approaches (Lapajne et al., 2022; Reis-Pereira et al., 2022). DR techniques are a class of predictor transformations. They can reduce data by creating a minor set of predictors that aim to retain most of the information contained in the original variables. Usually, these approaches generate new predictors which are functions of the original ones (signal extraction or feature extraction techniques) (Kuhn and Johnson, 2013).

This study examined a DR approach called Linear Discriminant Analysis (LDA), generally computed as a pre-processing. It is a supervised learning algorithm used for classification tasks. LDA is usually applied as a feature extraction technique, performed to reduce the dimensionality of the data while maximizing the class separability. It projects the high-dimensional data onto a lower-dimensional space while preserving the discriminative information between classes. In brief, data is projected onto a linear subspace that maximizes the ratio of between-class variance to within-class variance. Thus, the projected data points are as far apart as possible in the new space, while the points belonging to the same class are as close as possible. Therefore, LDA contributes to reducing the problem's computational complexity and avoiding overfitting. It can also be useful for visualizing the data in a lower-dimensional space, helping interpret patterns in data (Tharwat et al., 2017). Furthermore, this technique was applied since our dataset is not linearly separable, and LDA can organize it in another space with the maximum possible linear separability (Sachin, 2015).

LDA feature space loadings (also called coefficients or weights) were additionally used to infer the most relevant wavelength variables, through the computation of the interquartile range (IQR) for the weights. A threshold at 1.5 times the IQR beyond the upper quartile was established. This process aimed to increase sensitivity to the weight distribution, enabling the capture of outliers and extreme values. An applied predictive classification model was later computed to help deal with the non-linearity of the data.

### 2.3.3 Machine learning classification model

A Support Vector Machines (SVMs) algorithm was chosen to integrate this modeling strategy. This supervised machine learning algorithm performs classification based on the concept of optimal separating hyperplane (Vapnik, 1999; Mosavi et al., 2021). SVMs are nonlinear approaches that discover the most extensive margin between two classes in feature space. These approaches aim to decrease the error test and model complexity (Ballabio and Sterlacchini, 2012).

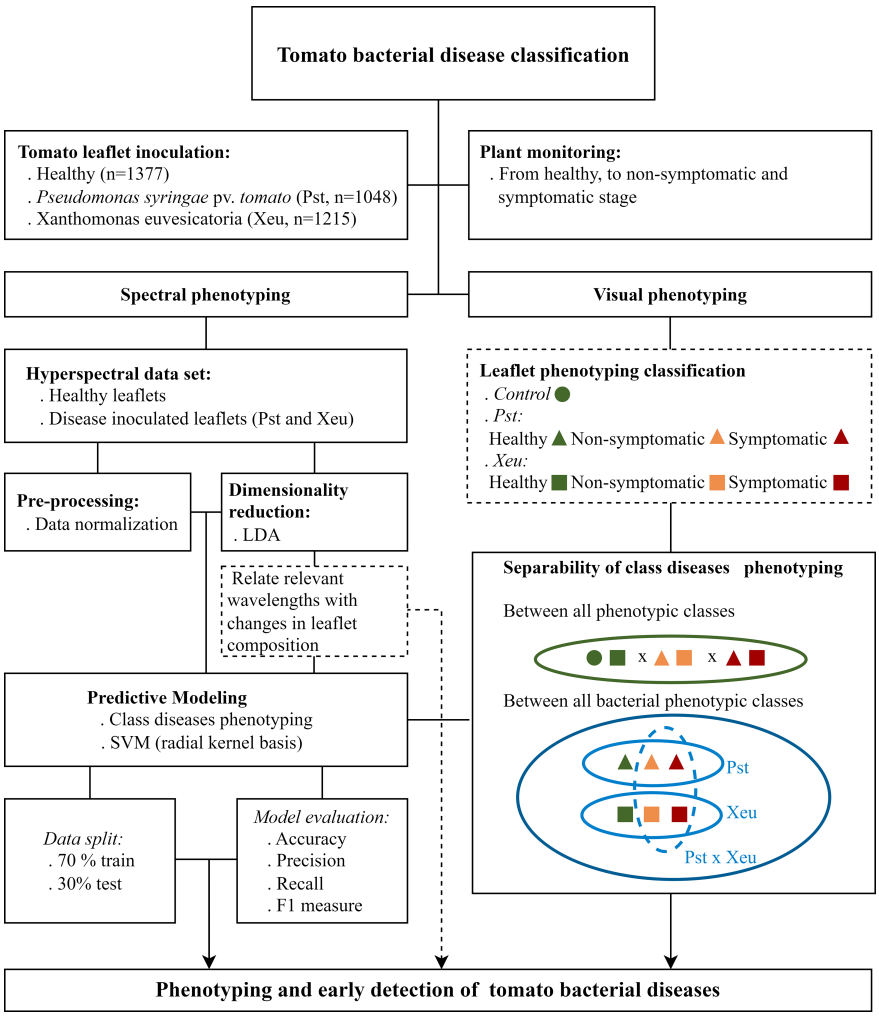



FIGURE 2
Conceptual diagram for the applied predictive modeling approaches of bacterial tomato leaflet disease.

SVMs can present distinct hyperparameters and kernel forms, which convert raw data inputs from the original user space into kernel space through a user-defined feature map (Patle and Chouhan, 2013; Ding et al., 2021). This study used a radial basis function (RBF) kernel was used since it allows SVMs to capture non-linear relationships between input features and target variables. It may also allocate distinct weights to different points since they learn the decision surface according to the relative importance of the data points in the training set (being well-suited for handling outliers and noisy data) (Xulei et al., 2005). Moredetailed information about the SVM algorithm, including relevant principles and calculation formulas, can be found in Ballabio and Sterlacchini (2012) and in Chang and Lin (2011). The parameters of the SVM applied corresponded to the default values of the algorithm implemented in the 'Scikit-learn' machine learning library (Pedregosa et al., 2011), which also can be found in Table 1.

0The datasets were divided into training data (70% of random observations) and validation data (30% of the remaining observations) (Kuhn and Johnson, 2013), following a holdout method (Lantz, 2019). The training and validation sets combined the pairs of concurrent measurements of the group and health status and the corresponding values of the predicting variables. A resampling strategy was performed as stated in Reis-Pereira et al. (2022) to reduce the possibility of overfitting (Berrar, 2019; Valier, 2020).

### 2.3.4 Model performance evaluation

Different metrics were additionally retrieved to investigate model performance, namely the Confusion Matrix (CM), accuracy score (Equation 2), and F1-Score (Equation 3) whose description is detailed in Reis-Pereira et al. (2022). Furthermore, precision (the fraction of correct positive predictions out of all positive predictions, Equation 4) and recall (fraction of correct positive predictions out of all observed positive samples, Equation 5) were also computed using the following formula, where true positive, false positive, false negative, and true negative values are denoted by TP, FP, FN, and TN, respectively:

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_N + F_P} \quad (2)$$

$$F1\ \ score = \frac{2 * T_P}{2 * T_P + F_P + F_N} \quad (3)$$

$$Precision = \frac{T_P}{T_P + F_P} \quad (4)$$

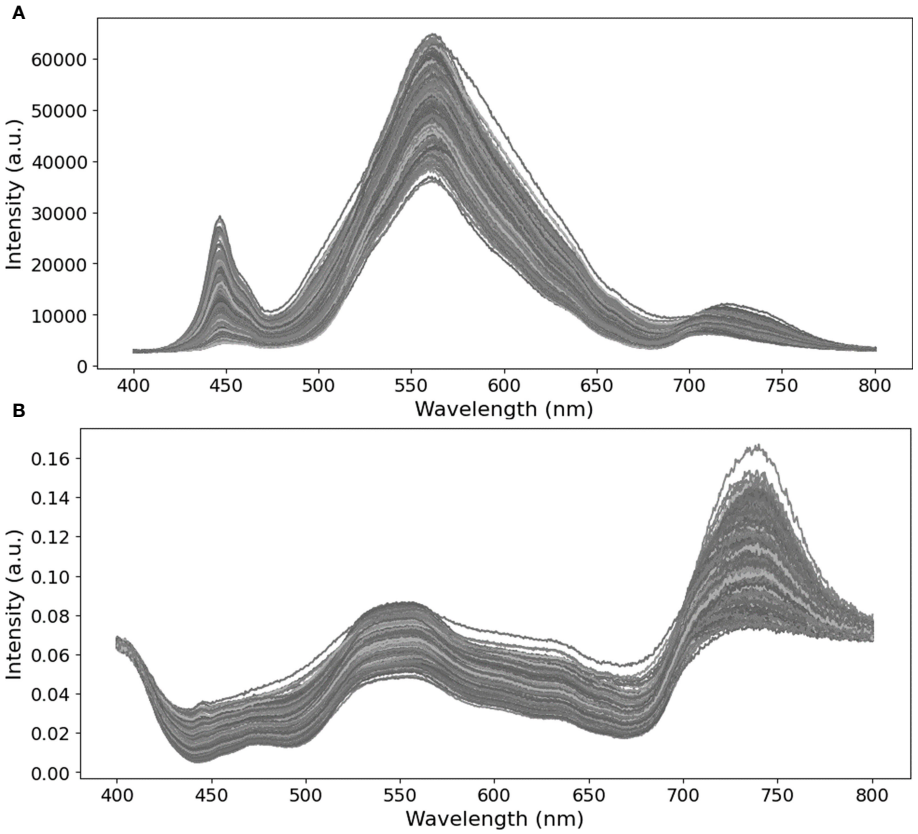



FIGURE 3
Original (raw, **A**) and normalized **(B)** hyperspectral signatures assessed in tomato leaflets during the experimental assay.

TABLE 1 Default parameters of the SVM algorithm of 'Scikit-learn' library used in this study.

| Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value |
|---|---|---|---|---|---|---|---|
| *C* | 1.0 | *Probability* | False | *Verbose* | False | *Break ties* | False |
| *Kernel* | rbf | *Shrinking* | True | *Cache size* | 200 | *Tolerance* | 1e-3 |
| *Gamma* | 'scale' 1/(n_features *X.var()) | *Class weight* | None | *Decision function shape* | One-vs-rest (ovr) | *Random state* | None |
| | | *Max iteration* | -1 | | | | |

$$Recall = \frac{T_P}{T_P + F_N} \quad (5)$$

All the computational analyses were performed in the Jupyter Notebook software using the libraries 'Matplotlib' (Ari and Ustazhanov, 2014), 'numpy', 'pandas' (Betancourt et al., 2019), and 'Scikit-learn' (Pedregosa et al., 2011).

# 3 Results

## 3.1 Observational-based phenotyping of leaflets symptomatology over time

### 3.1.1 PCR validation

Tomato plants were inoculated with Pst and Xeu bacteria, respectively. After spectral analysis, leaf samples from each treatment were tested for the presence of these bacteria. Proper controls from samples known to be positive and negative for Pst and Xeu bacteria were included to confirm the assay results. After the colony PCR reaction, the amplified products were separated by agarose gel electrophoresis and visualized under UV light. The PCR results showed bacteria-specific bands for each bacteria species, namely a 200-base pair (bp) fragment of Pst, and a 713 bp fragment for Xeu, indicating that Pst and Xeu bacteria were present in each inoculation treatment group. No PCR amplification was observed from samples collected from healthy leaves.

### 3.1.2 Visual and hyperspectral phenotyping

Tomato plants infected with Pst bacteria showed the first visual typical chlorotic symptoms mostly between four and five days after infection (DAI). These spots evolved into necrotic lesions at six to seven DAI. In turn, chlorotic lesions in samples inoculated with Xeu mainly developed among twelve to fifteen DAI, only evolving to the necrotic stage at seventeen to eighteen DAI. Pst-infected plants died 12 DAI (Figure 4).

Table 2 presents the dataset structure used, composed of 3478 spectral point measurements, from which 1377 (39.6%) observations correspond to the healthy class. Of these, 1215 (34.9%) assessments belonged to Control leaflets, 81 to measurements performed on Pst leaflets before bacteria inoculation, and 81 to captures made on Xeu leaflets also before bacterial infection. Spectral records performed before symptom appearance reached the value of 844 (24.3%), where 101 (2.9%) measurements belonged to non-symptomatic leaflets inoculated with Pst, and 743 (21.4%) to leaflets inoculated with Xeu bacteria. Lastly, after symptom development, 1257 (36.1%) spectra were captured (866 – 24.90% – from symptomatic Pst leaflets, and 391 – 11.24% – from Xeu symptomatic tissue). Class imbalance is observed due to the disease infection process's dynamic, resulting in symptoms appearing throughout the measurements dates at different rates (Table 1). Spectral assessments were performed during 18 DAI for Control and Xeu leaflets. For Pst, the process was only made until 15 DAI because the plants presented high-

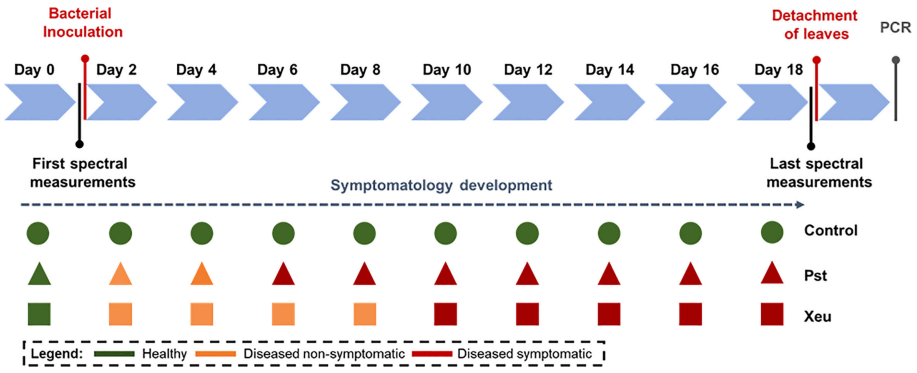



FIGURE 4
Observational-based phenotyping of leaflet symptomatology over time. Spectral measurements were performed before bacteria inoculation (Day 0), until day 15 (*Pseudomonas syringae* pv. *tomato* diseased leaflets), and 18 days after infection (Control and *Xanthomonas euvesicatoria* diseased leaflets). In the last measurement date, tomato leaflets were detached from each diseased plant and isolated in different bags for later performing the bacteria isolation assay.

10.3389/fpls.2023.1242201

TABLE 2 Spectral data characterization of the measurements randomly performed on tomato leaflets (healthy, diseased with *Pseudomonas syringae* pv. *tomato* – Pst –, and diseased with *Xanthomonas euvesicatoria* – Xeu), showing the number of assessments made by class and date.

| Days after Infection (DAI) | Non-inoculated | Inoculated classes | | | |
|---|---|---|---|---|---|
| | | Non-symptomatic | | Symptomatic | |
| | | Xeu | Pst | Xeu | Pst |
| 0 | 243* | 0 | 0 | 0 | 0 |
| 3 | 81 | 81 | 81 | 0 | 0 |
| 4 | 81 | 81 | 17 | 0 | 64 |
| 5 | 81 | 81 | 3 | 0 | 78 |
| 6 | 81 | 81 | 0 | 0 | 81 |
| 7 | 81 | 81 | 0 | 0 | 81 |
| 8 | 81 | 81 | 0 | 0 | 81 |
| 10 | 81 | 71 | 0 | 10 | 80 |
| 11 | 81 | 63 | 0 | 18 | 80 |
| 12 | 81 | 33 | 0 | 48 | 80 |
| 13 | 81 | 28 | 0 | 53 | 81 |
| 14 | 81 | 28 | 0 | 53 | 79 |
| 15 | 81 | 34 | 0 | 47 | 81 |
| 17 | 81 | 0 | – | 81 | – |
| 18 | 81 | 0 | – | 81 | – |
| **Total (n=3478)** | **1377** | **743** | **101** | **391** | **866** |
| | 39.6% | 21.4% | 2.9% | 11.2% | 24.9% |

* Including all plants. After day 0, only Control plants belong to this class.
Bold values correspond to the total number of assessments.

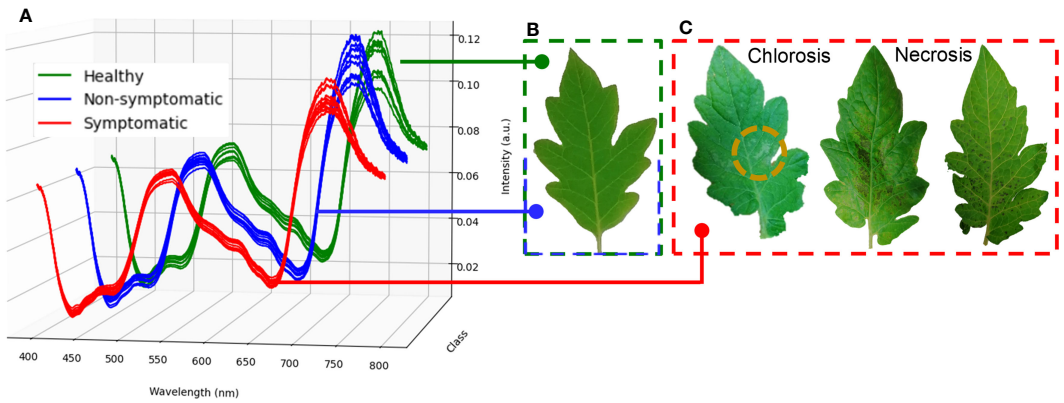



FIGURE 5
Mean normalized spectra of healthy, non-symptomatic, and symptomatic leaflet measurements for the first ten measurements performed (12 DAI, **A**). Healthy and non-symptomatic infected leaflets presented equal visual phenotype **(B)**. With infection evolution over time, chlorotic symptoms started to appear and later turned into necrotic lesions **(C)**.

stress levels, and leaf dehydration after this date, interfering with the spectral signal recording (Figure 1; Table 1).

Hyperspectral signatures captured in healthy leaflets showed the typical spectral behavior of healthy green tissues. On the other hand, spectral assessments belonging to disease leaflets (both with Pst and Xeu bacteria) presented deviations in their format (Figure 5). Thus, a more detailed analysis was performed for these measurements to evaluate the spectral modifications caused by the different bacteria, resulting in a higher number of classes in the study. Spectra signatures belonging to Pst inoculated samples had a more distinct spectral curve (for both, non-symptomatic and symptomatic stages) compared to the healthy measurements, showing higher intensity on the wavelength ranges of approximately 430 to 520 nm, and 560 to 680 nm. Nevertheless, the lower intensity was captured from 710 to 800 nm (Figures 6A, B). Xeu-inoculated tissues also displayed modification in their spectral signature in these regions. The intensity measured in the first two spectral intervals was marginally higher than the one captured on healthy leaflets. However, a more evident variance was observed in the 710 to 780 nm range (Figures 6A, C). When measurements belonging to disease samples were compared, the data showed differences between the samples infected with the different etiological agents. Pst measurements (for both non- and symptomatic stages) demonstrated greater intensity in the ranges of 430 to 520 nm, and 560 to 680 nm, but lower intensity in the 710 to 800 nm interval (Figures 6A, D).

## 3.2 Hyperspectral sensing-based phenotyping of leaflets symptomatology over time

### 3.2.1 Reducing the spectral dataset dimensionality

A Linear Discriminant Analysis (LDA) was performed to reduce the dimensionality of the normalized dataset, organizing the spectral observations in a new space as the maximum linear separability possible. LDA results were plotted and showed spectral separability between the different classes studied (Figure 7A). It was possible to see an evolution pattern through LD 1 for spectral data belonging to healthy, and Pst diseased leaflets regardless of whether they exhibit symptoms or not (Figures 7A, B). In turn, healthy and Xeu-diseased leaflets (including, non- and symptomatic data) presented a spectral separation gradient through LD2

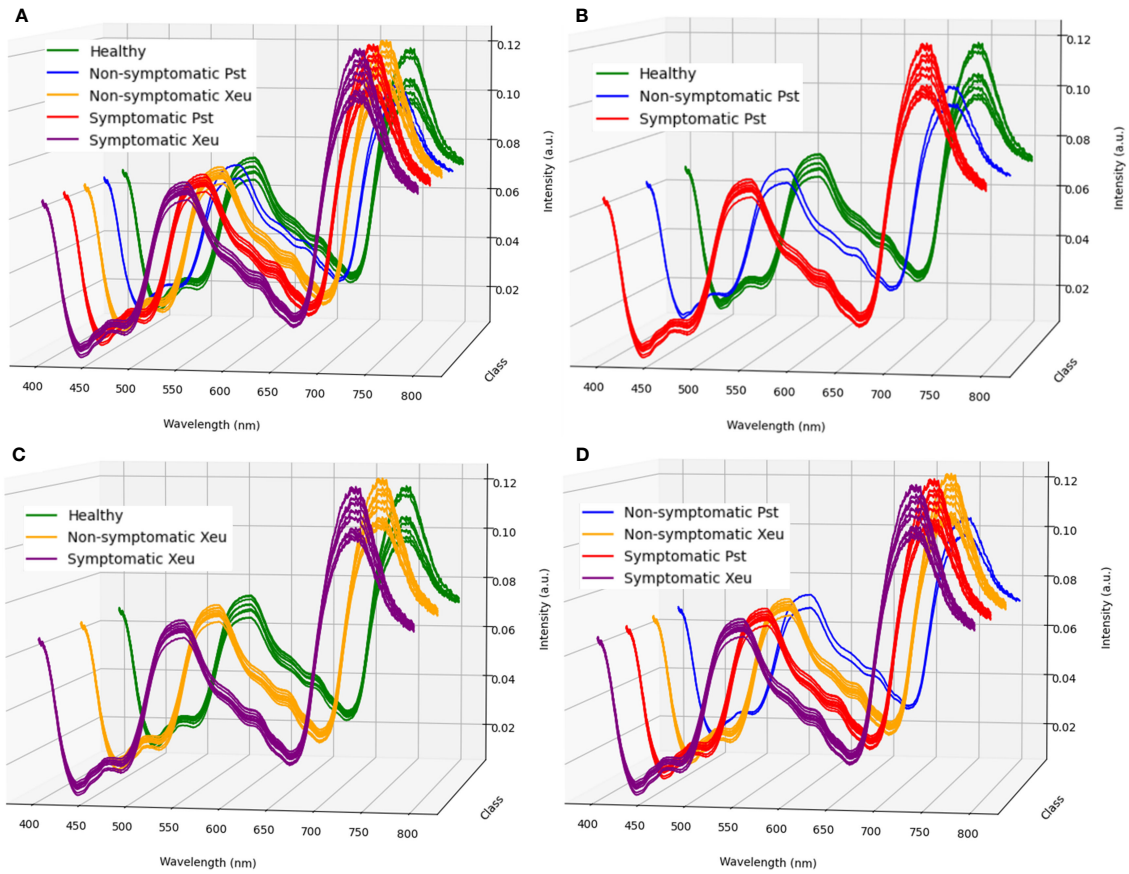



FIGURE 6
Mean normalized spectra for all classes in study (i.e., healthy, non-symptomatic, and symptomatic *Pseudomonas syringae* pv. tomato – Pst – leaflet measurements, and non-symptomatic *Xanthomonas euvesicatoria* – Xeu – assessments) for the first ten measurements performed (12 DAI, **A**). Different behaviours of healthy samples compared to Pst **(B)**, and Xeu **(C)** diseased leaflets are shown, as well as, between non-symptomatic and symptomatic diseased leaflets **(D)**.

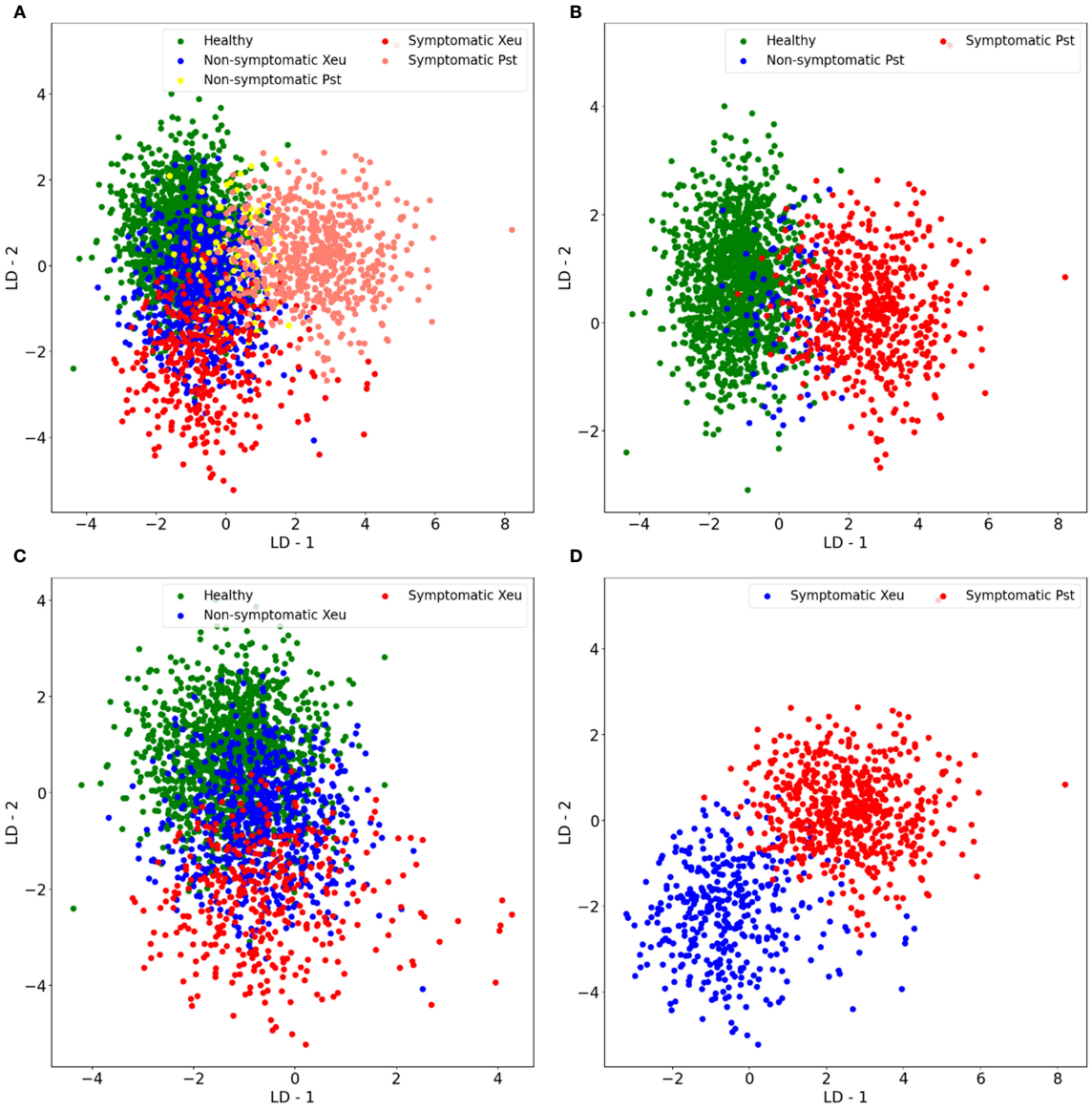



FIGURE 7
Scatter plots of the outcomes of the application of Linear Discriminant Analysis on the normalized data, for Linear Discriminant 1 (LD1) and Linear Discriminant 2 (LD2), when were used all the classes in study **(A)**, only healthy and Pseudomonas syringae pv. tomato infected samples **(B)**, just healthy and *Xanthomonas euvesicatoria* diseased leaflets **(C)**, and only diseased symptomatic samples **(D)**.

(Figures 7A, C). When data of diseased leaflets infected with distinct bacteria were compared, it was possible to observe a divergence gradient between the LD1 and LD2, especially at the symptomatic stage (Figures 7A, D). Since data presented a non-linear characteristic, overlapping between classes was observed. Thus, these findings demonstrated the efficacy of the LDA technique for reducing the dataset dimensionality to the most important features. LDA's DR results were, then, applied in the following steps of the modeling process helping in the classification task and avoiding overfitting.

The most relevant wavelength variables for LD1 were assessed based on their coefficients, equaling 44 features. These variables were mostly located in the blue-green and red VIS regions of the electromagnetic spectrum (blue - 434.9, 435.72, 438.17, 438.58, 440.21, 441.44, 442.67, 443.08, 445.53, 445.94, 448.4, 448.81, 494.6 nm; green - 503.74, 508.74, 527.53 nm; red - 556.09, 562.0, 562.84, 590.37, 607.82, 609.1, 611.24, 618.5, 643.36, 650.24, 673.97, 680.02 nm), coinciding with the wavelength absorption range of chlorophylls (430 to 480 nm, and 640 to 700 nm), and carotenoids pigments, namely β-carotenes (whose primary and secondary absorption peaks are respectively located at 450 to 480 nm, and 600 to 650), and xanthophylls (520 to 580 nm) (Figure 8). This coincides with the action of Pst and Xeu bacteria on tomato leaves' levels of photosynthetic pigments during the infection process.

Other plants whose metabolites are affected by these two bacteria also have their absorption spectrum coinciding with the selected wavelengths of LD1, namely some phenolic compounds

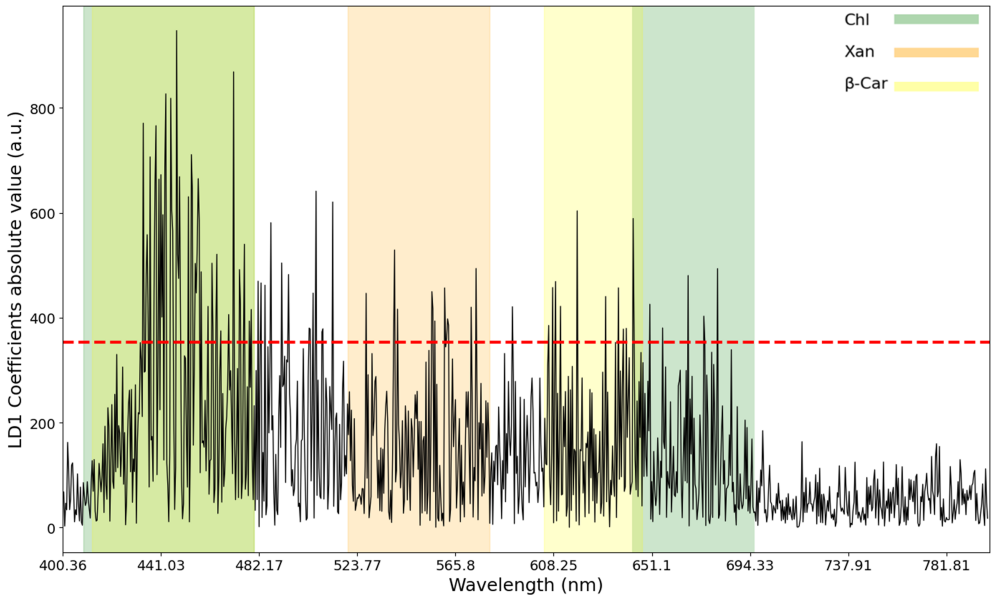



FIGURE 8
Absolute values of the coefficients results of Linear Discriminant Analysis for Linear Discriminant 1. Forty-four spectral wavelengths were identified as relevant when variable weights were computed. These variables coincided with the absorption spectra of different photosynthetic pigments, namely chlorophylls (Chl, highlighted in green for chlorophyll), and carotenoids (β-carotenes, β-car, highlighted in yellow; and xanthophyll's, Xan, highlighted in orange).

(e.g., flavonoids, 400 to 500 nm), and composts derived from chlorophylls decomposition, namely pheophytins (400 to 500 nm, and 600 to 700 nm) (Figure 8).

Applied predictive classification modeling was, then, performed using the LDA-reduced normalized data (including all classes: i) healthy; ii) non-symptomatic diseased Pst leaflets; iii) non-symptomatic Xeu samples; iv) symptomatic inoculated Pst tissues; v) symptomatic Xeu observations) and an SVM algorithm with a Radial Basis Function (RBF) kernel. The model was trained using 70% (2434) of the spectral observations (randomly selected), and then, it was validated using the remaining 30% (1044) of the observations (test set), and the complete dataset. The test set comprised 413 healthy samples, 30 non-symptomatic Pst disease leaflets, 223 non-symptomatic Xeu, 260 symptomatic Pst observations, and 118 symptomatic Xeu.

The developed model performed well for both the test set and the complete dataset. The model achieved an accuracy of 0.85 for the test set and 0.86 for the complete dataset, indicating that it can correctly classify most of the measurements (Table 3; Figure 9). Furthermore, it demonstrated high metric values (precision, recall, and F1-score) for all the classes, indicating that it can identify both healthy and infected measurements. In detail, higher precision, recall, and F1-score values were found for the healthy and non-symptomatic Pst leaflets measurements (Table 3). This shows that the model more easily predicted spectral assessments belonging to these classes. Nevertheless, it showed more difficulties in classifying measurements of Xeu inoculated leaflets, especially those captured before symptom appearance (indicated by lower metric scores). It is important to note that the model's performance was consistent across both the test set and the complete dataset, indicating that the

TABLE 3 Performance metrics for the classification SVMs-based model using all the data (train and test set – All), only the train set (Trn) and only the test set (Test).

| Class of leaflets status | Precision | | | Recall | | | F1-score | | | Accuracy | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *Trn* | *All* | *Test* | *Trn* | *All* | *Test* | *Trn* | *All* | *Test* | *Trn* | *All* | *Test* |
| *Healthy* | 0.86 | 0.85 | 0.84 | 0.89 | 0.89 | 0.88 | 0.88 | 0.87 | 0.86 | 0.89 | 0.85 | 0.88 |
| *N Sym. Pst* | 0.97 | 0.96 | 0.94 | 0.86 | 0.90 | 1.00 | 0.91 | 0.93 | 0.97 | 0.86 | 0.90 | 1.00 |
| *N Sym. Xeu* | 0.78 | 0.78 | 0.77 | 0.74 | 0.74 | 0.74 | 0.76 | 0.76 | 0.75 | 0.74 | 0.74 | 0.74 |
| *Sym. Pst* | 0.94 | 0.94 | 0.94 | 0.95 | 0.94 | 0.93 | 0.95 | 0.94 | 0.93 | 0.95 | 0.94 | 0.93 |
| *Sym. Xeu* | 0.83 | 0.83 | 0.83 | 0.80 | 0.79 | 0.77 | 0.81 | 0.81 | 0.80 | 0.79 | 0.79 | 0.77 |
| Weighted Avg ± s.d | 0.86± 0.07 | 0.86 ± 0.07 | 0.85± 0.07 | 0.86± 0.07 | 0.86 ± 0.08 | 0.85± 0.10 | 0.86± 0.07 | 0.86 ± 0.07 | 0.85 ± 0.08 | 0.86 ± 0.07 | 0.86 ± 0.08 | 0.85 ± 0.10 |

SVMs, Support Vector Machines; Trn, Train; N Symp., Non- symptomatic; Sym., Symptomatic; Avg., Average; s.d., standard deviation.

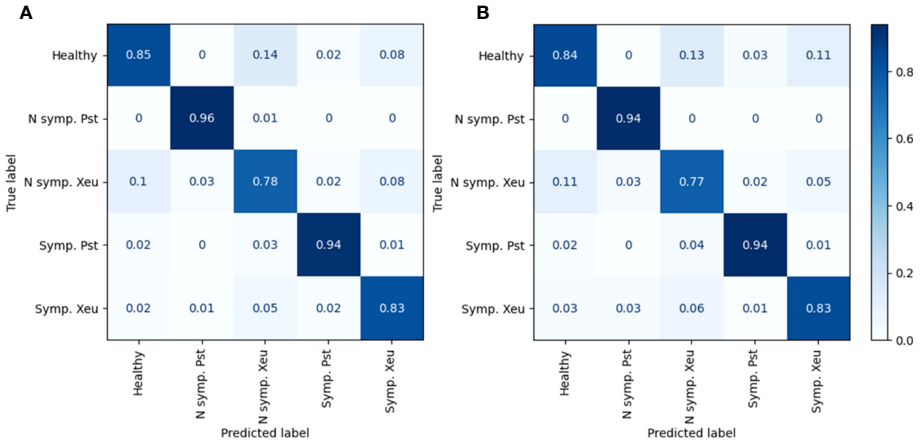



FIGURE 9
Confusion Matrix of the percentage of predicted samples for each class (column) that were correctly classified for each true class (row), for the complete **(A)** and test **(B)** sets. (N Symp., Non-symptomatic; Sym., Symptomatic).

model is robust and can be used to classify new spectral samples accurately.

Model predictions for the non-symptomatic Pst class did not present any misclassification in the test set. In the complete dataset, the model accurately predicted 96% of the spectral measurements but missed 1% of the predictions, which it classified as assessments made on non-symptomatic leaflets infected by Xeu (Figure 9). Symptomatic spectral captures performed in Pst diseased leaflets were correctly categorized in 94% of the cases for both the test and complete sets. Nevertheless, the model mistakenly classified these assessments as non-symptomatic Xeu observations in 4% and 3% of the cases, and as healthy samples in 2% when the test set and complete dataset were used, respectively. Predictions of Xeu spectral assessments were more challenging to the model, presenting a higher number of wrong classifications in the non-symptomatic class than in the remaining classes studied. In fact, the model successfully classified 77% of the measurements of this class in the test set, and 78% when all data was used. However, it attributed 11% and 10% of the measurements as healthy, 5% and 8% as symptomatic diseased Xeu leaflets assessments, 3% as non-symptomatic inoculated Pst observations, and 2% as symptomatic Pst captures, when the test set and complete dataset were used, respectively. The model showed more efficacy in identifying symptomatic Xeu leaflets measurements, predicting 83% of these samples in the test and complete datasets. In terms of missed classifications, it predicted 6% and 5% of the assessments as non-symptomatic, 3% and 2% as healthy, 3% and 1% as non-symptomatic spectral captures of Pst infected leaflets, and 1% and 2% as symptomatic Pst, in the test set and complete dataset, respectively (Figures 9A, B).

For the complete dataset prediction, we investigated the number of misclassifications per class and date (Figure 10). As expected, the observed tendency for healthy spectral assessments showed a regular number of observations per date (81). Nonetheless, the developed model categorized more samples than the true value per date, except for 7, 13, 17, and 18 DAI. On the other hand, the spectral model consistently underfit the infected Xeu leaflets, regardless of whether they exhibit symptoms or not (Figure 10A).

In plants inoculated with Xeu, discrepancies between observed and predicted classes are more evident in the non-symptomatic Xeu class in the observations recorded up to 10 days after infection. During this period, which included seven measurement dates of the non-symptomatic Xeu class, 53 observations were recorded below the predicted value of the developed model. In contrast, the healthy class accumulated 47 observations above the predicted value during the same period. Furthermore, according to the confusion matrix results (All data), 10% (148) of the non-symptomatic Xeu observations were misclassified as healthy. Considering the period up to 10 days after infection (data not shown), out of the 150 observations wrongly classified as healthy, 100 were from the non-symptomatic Xeu class. These results indicate that in the early stages of Xeu-induced disease infection, the symptoms developed in the plant leaflets are not strong enough for the developed model to distinguish them from healthy observations efficiently. Therefore, the non-symptomatic Xeu class, compared to other tested classes, exhibits the lowest model performance indicators (all data: accuracy 0.74, precision 0.78, recall 0.74, and F1-score 0.76). For the non-symptomatic stage, the actual observations presented a stable pattern until 8 DAI, and after a sharp drop was observed until 13 DAI, where the rate of infected leaflets increased up to 65%. A stable number of observations was maintained until 15 DAI, after which a period of exponential increase in observed symptomatic spectral measurements was registered. After this day, all leaflets were symptomatic. The model was rigorous in discriminating non-symptomatic Xeu leaflet measurements after 9/10 DAI, presenting a percentage of error inferior to 10% (correctly classifying 64 of the 71 observations) when about 90% of the sampled leaflets (71 of the initial 81 assessments) still didn't show any typical symptoms of the disease (Figure 10B).

For the prediction of the Pst disease samples, the non-symptomatic phase was very similar for both observed and predicted. Nevertheless, the prediction of the symptomatic phase showed irregularities between the five and seven days (corresponding to the dates were necrosis appeared). Is possible to observe that most of the Pst inoculated leaflets (79%) started to

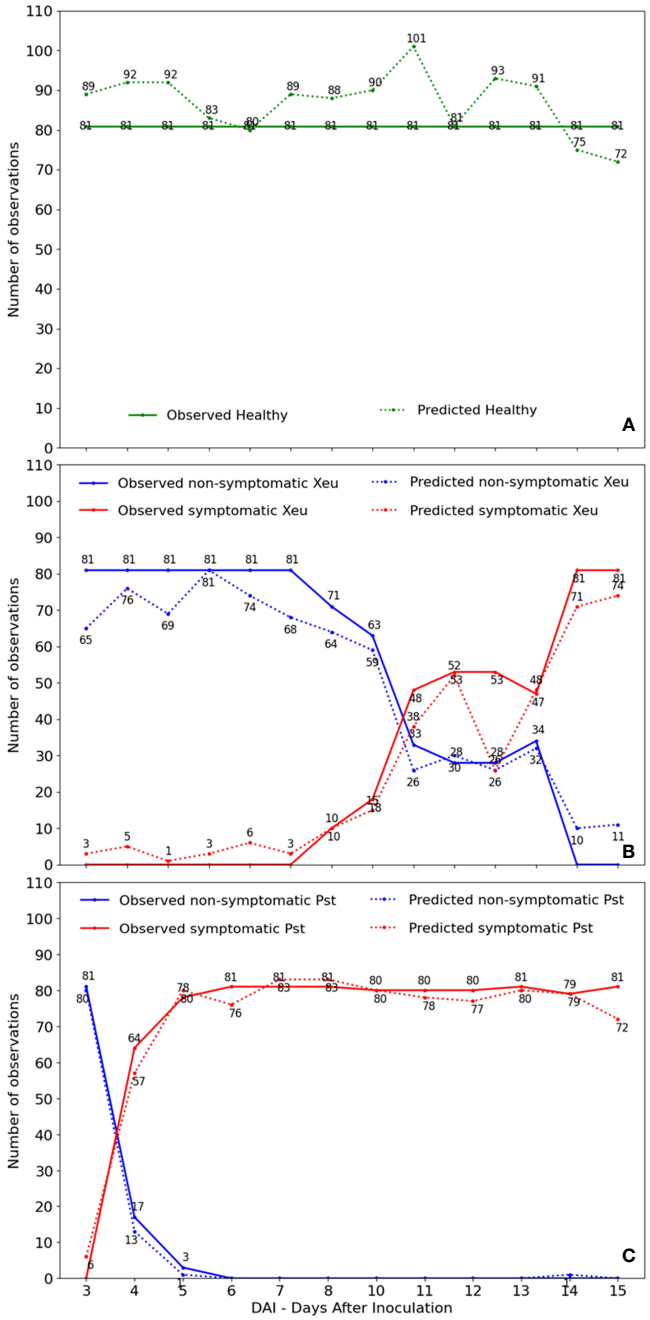



FIGURE 10
Number of observed and predicted samples by date of measurement for healthy **(A)**, *Xanthomonas euvesicatoria* diseased **(B)**, and *Pseudomonas syringae* pv. *tomato* diseased **(C)** leaflets' assessments.

show the first symptoms of the disease 4 DAI. The number of symptomatic sampled leaflets increased until 6 DAI, where all the leaflets assessed were symptomatic (Figure 10C).

## 4 Discussion

Plant infectious diseases are critical in agriculture and food security, impacting crop yields and quality. Understanding and effectively managing them is crucial for more sustainable agriculture, based on more preventive measures and early diagnosis.

The suitability of spectral phenotyping based on hyperspectral spectroscopy point-of-measurement (HS-POM) for diagnosing bacterial infectious diseases in tomato plants, namely bacterial speck and spot, was evaluated. In this approach, light penetrates the leaflet tissue and undergoes internal reflections, before ultimately being redirected to the spectrometer *via* a central fiber optics pinhole. This method ensures that all light reaching the

sensor interacts with the leaf tissues, thereby maximizing the spectral information from all internal tissues, including any changes caused by the interaction between the host and bacteria.

An applied predictive model integrating an SVM algorithm showed the capacity to accurately classify healthy and diseased tomato leaflets at various stages of disease development (specifically healthy, non-symptomatic diseased Pst, non-symptomatic disease Xeu, symptomatic Pst, and symptomatic Xeu). Even before symptom appearance, it showed a classification accuracy of 74% for Xeu and 100% for Pst diseased leaflets measurements, and a weighted average accuracy, precision, recall, and F1-score of 85%.This model was, thus, capable of categorizing healthy, disease (both non-symptomatic and symptomatic), and disease leaflet tissues infected with distinct bacteria species (both before and after symptom appearance), being coherent with visual phenotyping and PCR results. These outcomes, thus, demonstrate the suitability of this technique for performing an early disease assessment and class distinction (according to the phytosanitary health status, and type of pathogen responsible for the infection). This is extremely valuable since crops in the field are generally exposed to variable environmental and phytosanitary conditions and vulnerable to different types of abiotic and biotic stresses (which may cause similar visual lesions, difficult to distinguish by the naked eye). Also, bacterial spot and speck of tomato can develop in 6 to 14 days, depending on several factors (e.g., environmental conditions, pathogen strain, infection severity, inoculum concentration, and the susceptibility of the plants' variety) (Horst, 2013; Borkar and Yumlembam, 2016), and their spread among several plants in a production field is not immediate and may take time to occur. Thus, early diagnosis is crucial to prevent disease spread, promote preventive treatments, and lead to environmentally friendly practices, promoting precision agriculture principles.

LDA computation revealed spectral divergence between the different classes in study through LD1 and LD2 and uncovered relevant wavelengths for diagnosing the diseases caused by *Pseudomonas syringae* pv. *tomato* (Pst), and *Xanthomonas euvesicatoria* (Xeu). These were mostly located in the blue-green and red visible regions of the electromagnetic spectrum, corresponding to chlorophyll (mainly: 430 to 480 nm, and 640 to 700 nm) and carotenoid pigments' absorption spectra (i.e., 450 to 480 nm, 520 to 580 nm, and 600 to 650 nm), indicating modifications in the photosynthetic pigment's levels throughout the infection process. These findings are aligned with the impact of both bacteria species on host leaves' pigments values during infection, which start prior to symptoms appearance and became more pronounced with the formation of chlorotic and necrotic lesions. In this medium/late stages of infection, the breakdown of chlorophyll, in particular, can result in a subsequent accumulation of pheophytins (brown pigments, whose maximum absorption peak is located at 660-670 nm, and secondary peak around 430-450 nm), which also affect plant spectral behavior (Bhandari et al., 2015). Also, spectral divergences in the 700 to 800 nm range may indicate that structural components of leaves are affected during the infection process, resulting in the degradation of leaf structures along disease development. Spectral divergence between diseased leaves infected by different bacteria may be related to the production of specific molecules by each pathogen, which may affect the host spectral signature. As an example, Pst produces a phytotoxin called coronatine which alters chlorophyll fluorescence (by modifying the photosystem II – PSII) and can affect the absorption and scattering of light by plant tissues, leading to modifications in the spectra (Zhang et al., 2021). In turn, the host plant can activate different defense responses when in contact with distinct pathogens, triggering a series of biochemical and molecular responses, which also promote spectral modifications in the visible wavelength ranges. An example are phytoalexins (e.g., flavonoids), whose production was hypothesized to be related to increased plants' spectral reflectance in the VIS range (Leucker et al., 2016).

Hence, the present research findings demonstrate that HS-POM holds promise as an effective, fast, and cost-effective overtime method for early diagnosis of two bacterial infections caused by distinct pathogen species *in vivo* tomato plants, and for unraveling specific host-pathogen spectral dynamics. In the future, it is advisable to conduct further analysis, entailing the expansion of the dataset under study, test various values for SVM algorithm parameters, and enhance the modeling algorithms, among other potential approaches. This study corroborates previous research performed by our team using HS-POM for the early detection of bacterial tomato spot caused by Xeu bacteria. The spectral response properties of disease tomato leaves presented a divergent behavior when compared to healthy tissues, even before symptom appearance. This tendency was more evident in the absorption regions of photosynthetic pigments (namely, chlorophyll). A Principal Component Analysis (PCA) allowed the identification of relevant discriminative wavelengths at approximately 454-654 nm (Reis-Pereira et al., 2021), coinciding with the wavelengths identified by the LDA approach.

Other studies also demonstrated the potential of hyperspectral data and SVM-based classification modeling for disease diagnosis, presenting similar model evaluation metrics. As an example, Cen et al. (2022) studied the possibility of early detection of bacterial wilt in tomato by applying a portable hyperspectral spectrometer. Their model combined Genetic Algorithms and SVM and achieved overall accuracies (OA) of 90.7% in the distinction of healthy and symptomatic tissues. Tomaszewski et al. (2023) also demonstrated the suitability of hyperspectral measurements and machine learning for the early detection of anthracnose, bacterial speck, early blight, late blight, and septoria leaf, using a temporally-aggregated approach. When all the data were analyzed, the researchers found that the best-quality classification approach (integrating a Ridge classifier) presented an F1 score ranging from 0.71 to 0.95 (0.84 average) for the period of the first two weeks from inoculation. Despite being possible to find research diagnosing different types of biotic stress agents in the same assay, they are usually more related to fungi identification. Scarce results can be retrieved for studies comparing the assessment of diseases caused by different types of bacterial species.

Besides tomato crop studies, hyperspectral measurements were also valuable to achieve disease diagnosis in several plant species with agronomic interest. For instance, Rumpf et al. (2010) studied the suitability of hyperspectral reflectance, SVM, and vegetation indexes (VIs) for detect and classify diseases on sugar beet leaves (namely, Cercospora leaf spot, leaf rust, and powdery mildew).

Early differentiation between healthy and inoculated plants, as well as among specific diseases was achieved using SVM, registering accuracy values ranging from 65 to 90%. When data belonging to healthy and diseased leaves (including all the samples affected by the three different pathogens) was used, the classification model achieved an accuracy higher than 86%. Furthermore, Tian et al. (2021) also proved the efficacy of spectroscopy and machine learning techniques for rice leaf blast infection from non-symptomatic to mild stages. An approach integrating an SVM algorithm achieved maximum classification accuracies of over 80% and 83% for the early infection stage of the 2018 and 2019 experiments.

The desirable possibility of applying hyperspectral data for in-field detection and classification of diseases was also proved. Deng et al. (2019) also demonstrated the possibility of applying hyperspectral reflectance in-field detection and classification of citrus Huanglongbing disease. They developed an SVM learner which achieved 90.8% accuracy in healthy, asymptomatic, and symptomatic discrimination. Our team, likewise demonstrated the capability of using HS to diagnose *in situ* bacterial canker disease, caused by another *Pseudomonas* pathovar, specifically *Pseudomonas syringae* pv. *actinidiae* (also known as Psa). Asymptomatic and symptomatic leaves were successfully discriminated through the computation of several modeling approaches involving different feature selection techniques, as well as multivariate analysis methods and machine learning algorithms. The best predictive classification model for discriminating the bacterial kiwi canker disease showed an overall accuracy of 0.85, with an F1-score (Reis-Pereira et al., 2022). These findings suggest that hyperspectral data can be successfully used to predict plant diseases both indoor and infield conditions, caused by different etiological agents (e.g., fungi, bacteria, and virus), in both herbaceous and woody crops. Despite these encouraging findings, it is important to highlight that comparison between different research can be challenging due to the pathogens in study (e.g., generally disease detection using HS is mostly performed for fungal infections), host-pathogen specific interactions, number of samples used, number of classes analyzed, moment of disease assessment (before or after symptoms appearance, in a specific date or overtime), environmental and experimental conditions on the moment of data acquisition, among others. Thus, future studies using tomato plants should be performed to evaluate the efficacy of this approach.

In summary, point-of-measurement Hyperspectral Spectroscopy devices combined with applied predictive models seem to be suitable for spectral phenotyping of bacterial-infected tomato leaflets. Nevertheless, HS-POM approaches as plant disease diagnostic methods are still in a very initial phase of development, and their Technology Readiness Levels (TRLs) must be improved. Standardized protocols for hyperspectral data acquisition should be developed aiming to uniformize the diagnosis processes and reduce noise and undesired spectral interferences. Also, more research on different host-pathogen interactions must be performed. Classification models developed under controlled conditions can be highly effective and constitute an important step for improving and maturing the diagnosis process. In fact, these models usually can detect symptoms earlier than in field assays (since optimal conditions for bacteria development, dissemination, and infection can be recreated), making the process faster and specific to the host-pathogen in study. Hence, the more challenging in-field application of HS-POM for disease diagnosis, posing additional complexities due to sensing system configurations (e.g., light source, probe position, among others), can be established and improved.

Future studies must be conducted to complement these gaps and validate the application of this technique as a suitable tool for accurately predicting different host-pathogen interactions and their impact on the crops' spectral signature. Further methodological developments are necessary to address these challenges and enhance the suitability of HS-POM for real-time disease monitoring and precision agriculture systems. Moreover, the implementation of feature selection techniques and dimensionality reduction approaches can help identify relevant wavelengths for distinguishing crop diseases, making possible the development and production of more cost-effective multiband sensors. These devices can be integrated into different platforms, enabling spectral data acquisition at different levels, such as leaf, single-plant, and canopy scales.

## 5 Conclusion

The present research explored the application of *in-vivo* POM hyperspectral spectroscopy combined with applied predictive modeling to classify bacterial leaf diseases in tomato crop, caused by *Pseudomonas syringea* pv. *tomato* and *Xanthomonas euvesicatoria*. Healthy leaves showed a characteristic spectral signature of green and photosynthetically active vegetation, while symptomatic leaves presented differences in their spectral signature in the VIS region. Spectral differentiation between healthy and diseased leaves was observed, even in the early stages of the infection process, when diseased samples didn't present any visual symptom (asymptomatic stage). Furthermore, plants inoculated with Pst bacteria also revealed a divergent spectral behavior from the ones infected with Xeu, indicating that this approach may be suitable for differentiating the etiological agents. Colony PCR also validated the effectiveness of the infection process for each sample group. The developed model revealed a classification accuracy for the test set of 100% for Pst disease leaflets without any visual symptom, and of 74% for Xeu disease leaflets also in a non-symptomatic stage of infection. The developed model achieved a weighted average accuracy, precision, recall, and F1-score of 85% for the test set. These findings strength the applicability of applied predictive classification modeling using HS-POM to early detect bacterial crop diseases. Nevertheless, complementary, and additional studies are recommended to unravel the host-pathogen interactions and their impact on the crop spectral signature. More economic, multiband devices can be developed hereafter considering the features selected for crop disease discrimination. Thus, different agronomic tasks (including mapping, monitoring, scouting, and treatment of plant diseases) can be performed more accurately with this methodology, fulfilling the precision agriculture concept. Spectroscopy sensors can also be mounted on diverse platforms, creating different functioning measurement systems,

which can assess spectral data on distinct levels (namely, leaf, single-plant, and canopy scale).

## Data availability statement

The original contributions presented in the study are included in the article/supplementary files, further inquiries can be directed to the corresponding author/s.

## Author contributions

MRP, FS, FT, and MC contributed to the conception and design of the study. MRP organized the database. MRP and FS performed the data analysis. MRP and FS performed the applied predictive modeling. MRP wrote the first draft of the manuscript. MRP, FS, FT, and MC wrote sections of the manuscript. MRP and MC developed the figures to be integrated into the manuscript. All authors contributed to the article and approved the submitted version.

## Funding

This work is partially financed by National Funds through the FCT—Fundação para a Ciência e a Tecnologia, I.P. (Portuguese Foundation for Science and Technology) within the project OmicBots, with reference PTDC/ASP-HOR/1338/2021.

## Acknowledgments

Mafalda Reis-Pereira was supported by the Fundação para a Ciência e Tecnologia (FCT) fellowship with the reference SFRH/BD/146564/2019.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

Ahmadi, P., Muharam, F. M., Ahmad, K., Mansor, S., and Abu Seman, I. (2017). Early detection of ganoderma basal stem rot of oil palms using artificial neural network spectral analysis. *Plant Dis.* 101 (6), 1009–1016. doi: 10.1094/PDIS-12-16-1699-RE

Albuquerque, P., Caridade, C. M. R., Rodrigues, A. S., Marcal, A. R. S., Cruz, J., Cruz, L., et al. (2012). Evolutionary and experimental assessment of novel markers for detection of Xanthomonas euvesicatoria in plant samples. *PloS One* 7 (5), e37836. doi: 10.1371/journal.pone.0037836

Alves, A., Ribeiro, R., Azenha, M., Cunha, M., and Teixeira, J. (2023). Effects of exogenously applied copper in tomato plants' Oxidative and nitrogen metabolisms under organic farming conditions. *Horticulturae* 9 (3), 323. doi: 10.3390/horticulturae9030323

Ari, N., and Ustazhanov, M. (2014). "Matplotlib in python," in *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*, Abuja, Nigeria, 2014, pp. 1-6. doi: 10.1109/ICECCO.2014.6997585

Bagheri, N., Mohamadi-Monavar, H., Azizi, A., and Ghasemi, A. (2018). Detection of Fire Blight disease in pear trees by hyperspectral data. *Eur. J. Remote Sens.* 51 (1), 1–10. doi: 10.1080/22797254.2017.1391054

Ballabio, C., and Sterlacchini, S. (2012). Support vector machines for landslide susceptibility mapping: the Staffora River Basin case study, Italy. *Math. Geosci.* 44 (1), 47–70. doi: 10.1007/s11004-011-9379-9

Barroso, T. G., Ribeiro, L., Gregório, H., Monteiro-Silva, F., Neves dos Santos, F., and Martins, R. C. (2022). Point-of-care using Vis-NIR spectroscopy for white blood cell count analysis. *Chemosensors* 10 (11), 460. doi: 10.3390/chemosensors10110460

Berrar, D. (2019). Cross-validation. Editor(s): S. Ranganathan, M. Gribskov, K. Nakai and C. Schönbach, *Encyclopedia of Bioinformatics and Computational Biology* (Academic Press) 2019, 542–545. doi: 10.1016/B978-0-12-809633-8.20349-X

Betancourt, R., Chen, S., Betancourt, R., and Chen, S. (2019). *pandas Library. Python for SAS Users: A SAS-Oriented Introduction to Python*. (CA: Apress Berkeley), 65–109.

Bhandari, D. R., Wang, Q., Friedt, W., Spengler, B., Gottwald, S., and Römpp, A. (2015). High resolution mass spectrometry imaging of plant tissues: towards a plant metabolite atlas. *Analyst* 140 (22), 7696–7709. doi: 10.1039/C5AN01065A

Blancard, D. (2012). "3 - principal characteristics of pathogenic agents and methods of control," in *Tomato Diseases (Second Edition)*. Ed. D. Blancard (London: Academic Press), 413–650.

Borkar, S. G., and Yumlembam, R. A. (2016). *Bacterial diseases of crop plants* (Boca Raton: CRC Press).

Caicedo, J. P. R., Verrelst, J., Muñoz-Marí, J., Moreno, J., and Camps-Valls, G. (2014). Toward a semiautomatic machine learning retrieval of biophysical parameters. *IEEE J. Selected Topics Appl. Earth Observ. Remote Sens.* 7 (4), 1249–1259. doi: 10.1109/JSTARS.2014.2298752

Cen, Y., Huang, Y., Hu, S., Zhang, L., and Zhang, J. (2022). Early detection of bacterial wilt in tomato with portable hyperspectral spectrometer. *Remote Sens.* 14 (12), 2882. doi: 10.3390/rs14122882

Chang, C.-C., and Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM Trans. Intelligent Syst. Technol. (TIST)* 2 (3), 1–27. doi: 10.1145/1961189.1961199

Delalieux, S., van Aardt, J., Keulemans, W., Schrevens, E., and Coppin, P. (2007). Detection of biotic stress (Venturia inaequalis) in apple trees using hyperspectral data: Non-parametric statistical approaches and physiological implications. *Eur. J. Agron.* 27 (1), 130–143. doi: 10.1016/j.eja.2007.02.005

Deng, X., Huang, Z., Zheng, Z., Lan, Y., and Dai, F. (2019). Field detection and classification of citrus Huanglongbing based on hyperspectral reflectance. *Comput. Electron. Agric.* 167, 105006. doi: 10.1016/j.compag.2019.105006

Ding, X., Liu, J., Yang, F., and Cao, J. (2021). Random radial basis function kernel-based support vector machine. *J. Franklin Institute* 358 (18), 10121–10140. doi: 10.1016/j.jfranklin.2021.10.005

Fang, Y., and Ramasamy, R. P. (2015). Current and prospective methods for plant disease detection. *Biosensors-Basel* 5 (3), 537–561. doi: 10.3390/bios5030537

Fernandes, C., Albuquerque, P., Mariz-Ponte, N., Cruz, L., and Tavares, F. (2021). Comprehensive diversity assessment of walnut-associated xanthomonads reveal the occurrence of distinct Xanthomonas arboricola lineages and of a new species (Xanthomonas euroxanthea) within the same tree. *Plant Pathol.* 70 (4), 943–958. doi: 10.1111/ppa.13355

Fernandes, C., Albuquerque, P., Sousa, R., Cruz, L., and Tavares, F. (2017). Multiple DNA Markers for Identification of Xanthomonas arboricola pv. juglandis isolates and its direct detection in plant samples. *Plant Dis.* 101 (6), 858–865. doi: 10.1111/ppa.13355

Gold, K. M., Townsend, P. A., Chlus, A., Herrmann, I., Couture, J. J., Larson, E. R., et al. (2020a). Hyperspectral measurements enable pre-symptomatic detection and

differentiation of contrasting physiological effects of late blight and early blight in potato. *Remote Sens.* 12 (2), 286. doi: 10.3390/rs12020286

Gold, K. M., Townsend, P. A., Herrmann, I., and Gevens, A. J. (2020b). Investigating potato late blight physiological differences across potato cultivars with spectroscopy and machine learning. *Plant Sci.* 295, 110316. doi: 10.1016/j.plantsci.2019.110316

Golhani, K., Balasundram, S. K., Vadamalai, G., and Pradhan, B. (2018). A review of neural networks in plant disease detection using hyperspectral data. *Inf. Process. Agric.* 5 (3), 354–371. doi: 10.1016/j.inpa.2018.05.002

Guezenoc, J., Gallet-Budynek, A., and Bousquet, B. (2019). Critical review and advices on spectral-based norMalization methods for LIBS quantitative analysis. *Spectrochimica Acta Part B: Atomic Spectrosc.* 160, 105688. doi: 10.1016/j.sab.2019.105688

Herrmann, I., Berenstein, M., Paz-Kagan, T., Sade, A., and Karnieli, A. (2017). Spectral assessment of two-spotted spider mite damage levels in the leaves of greenhouse-grown pepper and bean. *Biosyst. Eng.* 157, 72–85. doi: 10.1016/j.biosystemseng.2017.02.008

Horst, R. K. (2013). *Westcott's plant disease handbook* (Dordrecht, Berlin, Heidelberg, New York: Springer Science & Business Media).

Kuhn, M., and Johnson, K. (2013). "Data pre-processing," in *Applied Predictive Modeling* (New York, NY: Springer New York), 27–59.

Lamichhane, J. R. (2015). Bacterial diseases of crops: elucidation of the factors that lead to differences between field and experimental infections. *Adv. Agronomy Vol* 134 134, 227–246. doi: 10.1016/bs.agron.2015.06.006

Lantz, B. (2019). *Machine learning with R: expert techniques for predictive modeling* (Birmingham B3 2PB, UK: Packt publishing ltd).

Lapajne, J., Knapič, M., and Žibrat, U. (2022). Comparison of selected dimensionality reduction methods for detection of root-knot nematode infestations in potato tubers using hyperspectral imaging. *Sens. (Basel)* 22 (1), 367. doi: 10.3390/s22010367

Leucker, M., Wahabzada, M., Kersting, K., Peter, M., Beyer, W., Steiner, U., et al. (2016). Hyperspectral imaging reveals the effect of sugar beet quantitative trait loci on Cercospora leaf spot resistance. *Funct. Plant Biol.* 44 (1), 1–9. doi: 10.1071/FP16121

Mahlein, A.-K. (2016). Plant disease detection by imaging sensors – parallels and specific demands for precision agriculture and plant phenotyping. *Plant Dis.* 100 (2), 241–251. doi: 10.1094/PDIS-03-15-0340-FE

Mahlein, A.-K., Kuska, M. T., Behmann, J., Polder, G., and Walter, A. (2018). Hyperspectral sensors and imaging technologies in phytopathology: state of the art. *Annu. Rev. Phytopathol.* 56 (1), 535–558. doi: 10.1146/annurev-phyto-080417-050100

Mahlein, A. K., Steiner, U., Dehne, H. W., and Oerke, E. C. (2010). Spectral signatures of sugar beet leaves for the detection and differentiation of diseases. *Precis. Agric.* 11 (4), 413–431. doi: 10.1007/s11119-010-9180-7

Martinelli, F., Scalenghe, R., Davino, S., Panno, S., Scuderi, G., Ruisi, P., et al. (2015). Advanced methods of plant disease detection. A review. *Agron. Sustain. Dev.* 35 (1), 1–25. doi: 10.1007/s13593-014-0246-1

Martins, R. C., Barroso, T. G., Jorge, P., Cunha, M., and Santos, F. (2022). Unscrambling spectral interference and matrix effects in Vitis vinifera Vis-NIR spectroscopy: Towards analytical grade '*in vivo*' sugars and acids quantification. *Comput. Electron. Agric.* 194, 106710. doi: 10.1016/j.compag.2022.106710

Meng, R., Lv, Z., Yan, J., Chen, G., Zhao, F., Zeng, L., et al. (2020). Development of spectral disease indices for southern corn rust detection and severity classification. *Remote Sens.* 12 (19), 3233. doi: 10.3390/rs12193233

Morellos, A., Tziotzios, G., Orfanidou, C., Pantazi, X. E., Sarantaris, C., Maliogka, V., et al. (2020). Non-destructive early detection and quantitative severity stage classification of tomato chlorosis virus (ToCV) infection in young tomato plants using Vis–NIR spectroscopy. *Remote Sens.* 12 (12), 1920. doi: 10.3390/rs12121920

Mosavi, A., Sajedi Hosseini, F., Choubin, B., Taromideh, F., Ghodsi, M., Nazari, B., et al. (2021). Susceptibility mapping of groundwater salinity using machine learning models. *Environ. Sci. pollut. Res.* 28 (9), 10804–10817. doi: 10.1007/s11356-020-11319-5

Parker, S. R., Shaw, M. W., and Royle, D. J. (1995). The reliability of visual estimates of disease severity on cereal leaves. *Plant Pathol.* 44 (5), 856–864. doi: 10.1111/j.1365-3059.1995.tb02745.x

Patle, A., and Chouhan, D. S. (2013). "SVM kernel functions for classification," in *2013 International Conference on Advances in Technology and Engineering (ICATE)*, Mumbai, India, 2013, pp. 1–9. doi: 10.1109/ICAdTE.2013.6524743

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* 12, 2825–2830. doi: 10.5555/1953048.2078195

Randolph, T. W. (2006). Scale-based norMalization of spectral data. *Cancer Biomarkers* 2 (3-4), 135–144. doi: 10.3233/CBM-2006-23-405

Reis-Pereira, M., Martins, R. C., Silva, A. F., Tavares, F., Santos, F., and Cunha, M. (2021). Unravelling plant-pathogen interactions: proximal optical sensing as an effective tool for early detect plant diseases. *Chem. Proc.* 5 (1), 18. doi: 10.3390/CSAC2021-10560

Reis-Pereira, M., Tosin, R., Martins, R., Neves dos Santos, F., Tavares, F., and Cunha, M. (2022). Kiwi Plant Canker Diagnosis Using Hyperspectral Signal Processing and Machine Learning: Detecting Symptoms Caused by Pseudomonas syringae pv. actinidiae. *Plants* 11 (16), 2154. doi: 10.3390/plants11162154

Ritchie, D. (2000). Bacterial spot of pepper and tomato. *Plant Health Instructor*. doi: 10.1094/PHI-I-2000-1027-01

Rivera, J. P., Verrelst, J., Delegido, J., Veroustraete, F., and Moreno, J. (2014). On the semi-automatic retrieval of biophysical parameters based on spectral index optimization. *Remote Sens.* 6 (6), 4927–4951. doi: 10.3390/rs6064927

Rumpf, T., Mahlein, A. K., Steiner, U., Oerke, E. C., Dehne, H. W., and Plümer, L. (2010). Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance. *Comput. Electron. Agric.* 74 (1), 91–99. doi: 10.1016/j.compag.2010.06.009

Sachin, D. (2015). Dimensionality reduction and classification through PCA and LDA. *Int. J. Comput. Appl.* 122 (17), 4–8. doi: 10.5120/21790-5104

Skoneczny, H., Kubiak, K., Spiralski, M., and Kotlarz, J. (2020). Fire blight disease detection for apple trees: hyperspectral analysis of healthy, infected and dry leaves. *Remote Sens.* 12 (13), 2101. doi: 10.3390/rs12132101

Tharwat, A., Gaber, T., Ibrahim, A., and Hassanien, A. E. (2017). Linear discriminant analysis: A detailed tutorial. *AI Commun.* 30, 169–190. doi: 10.3233/AIC-170729

Thenkabail, P. S., Gumma, M. K., Teluguntla, P., and Mohammed, I. A. (2014). Hyperspectral remote sensing of vegetation and agricultural crops. *Photogrammetric Eng. Remote Sens.* 80 (8), 697–709. doi: 20.500.11766/5374

Thenkabail, P. S., Smith, R. B., and De Pauw, E. (2000). Hyperspectral vegetation indices and their relationships with agricultural crop characteristics. *Remote Sens. Environ.* 71 (2), 158–182. doi: 10.1016/S0034-4257(99)00067-X

Tian, L., Xue, B., Wang, Z., Li, D., Yao, X., Cao, Q., et al. (2021). Spectroscopic detection of rice leaf blast infection from asymptomatic to mild stages with integrated machine learning and feature selection. *Remote Sens. Environ.* 257, 112350. doi: 10.1016/j.rse.2021.112350

Tomaszewski, M., Nalepa, J., Moliszewska, E., Ruszczak, B., and Smykała, K. (2023). Early detection of Solanum lycopersicum diseases from temporally-aggregated hyperspectral measurements using machine learning. *Sci. Rep.* 13 (1), 7671. doi: 10.1038/s41598-023-34079-x

Tosin, R., Martins, R., Pocas, I., and Cunha, M. (2022). Canopy VIS-NIR spectroscopy and self-learning artificial intelligence for a generalised model of predawn leaf water potential in Vitis vinifera. *Biosyst. Eng.* 219, 235–258. doi: 10.1016/j.biosystemseng.2022.05.007

Valier, A. (2020). *The Cross Validation in Automated Valuation Models: A Proposal for Use* (Cagliari, Italy: Computational Science and Its Applications – ICCSA 2020, Cham, Springer International Publishing).

Vapnik, V. (1999). *The nature of statistical learning theory* (New York, NY: Springer science & business media).

Vieira, J., Mendes, M., Albuquerque, P., Moradas-Ferreira, P., and Tavares, F.J.L.i. (2007). A novel approach for the identification of bacterial taxa-specific molecular markers. *Lett. Appl. Microbiol.* 44, 5, 506–512. doi: 10.1111/j.1472-765X.2007.02109.x

Xulei, Y., Qing, S., and Cao, A. (2005). "Weighted support vector machine for data classification," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005, Montreal, QC, Canada*. 2, 859–864. doi: 10.1109/IJCNN.2005.1555965

Yu, K., Anderegg, J., Mikaberidze, A., Karisto, P., Mascher, F., McDonald, B. A., et al. (2018). Hyperspectral canopy sensing of wheat Septoria tritici blotch disease. *Front. Plant Sci.* 9 (1195). doi: 10.3389/fpls.2018.01195

Zhang, Z., He, B., Sun, S., Zhang, X., Li, T., Wang, H., et al. (2021). The phytotoxin COR induces transcriptional reprogramming of photosynthetic, hormonal and defence networks in tomato. *Plant Biol.* 23, 69–79. doi: 10.1111/plb.13239

Zhang, J. C., Wang, N., Yuan, L., Chen, F. N., and Wu, K. H. (2017). Discrimination of winter wheat disease and insect stresses using continuous wavelet features extracted from foliar spectral measurements. *Biosyst. Eng.* 162, 20–29. doi: 10.1016/j.biosystemseng.2017.07.003

Zhang, N., Yang, G. J., Pan, Y. C., Yang, X. D., Chen, L. P., and Zhao, C. J. (2020). A review of advanced technologies and development for hyperspectral-based plant disease detection in the past three decades. *Remote Sens.* 12 (19), 3188. doi: 10.3390/rs12193188

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 30 August 2023
DOI 10.3389/fpls.2023.1211075

Check for updates

OPEN ACCESS

EDITED BY
Mario Cunha,
University of Porto, Portugal

REVIEWED BY
Ricardo Santos,
Federal University of Mato Grosso do Sul, Brazil
Parvathaneni Naga Srinivasu,
Prasad V. Potluri Siddhartha Institute of Technology, India

*CORRESPONDENCE
Sang Cheol Kim
sckim7777@jbnu.ac.kr
Hyongsuk Kim
hskim@jbnu.ac.kr

†These authors have contributed equally to this work

RECEIVED 22 May 2023
ACCEPTED 08 August 2023
PUBLISHED 30 August 2023

CITATION
Ma R, Fuentes A, Yoon S, Lee WY, Kim SC, Kim H and Park DS (2023) Local refinement mechanism for improved plant leaf segmentation in cluttered backgrounds. *Front. Plant Sci.* 14:1211075. doi: 10.3389/fpls.2023.1211075

COPYRIGHT
© 2023 Ma, Fuentes, Yoon, Lee, Kim, Kim and Park. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# Local refinement mechanism for improved plant leaf segmentation in cluttered backgrounds

Ruihan Ma[1,2†], Alvaro Fuentes[1,2†], Sook Yoon[3], Woon Yong Lee[4], Sang Cheol Kim[2*], Hyongsuk Kim[1,2*] and Dong Sun Park[1,2]

[1]Department of Electronics Engineering, Jeonbuk National University, Jeonbuk, Republic of Korea, [2]Core Research Institute for Intelligent Robots, Jeonbuk National University, Jeonbuk, Republic of Korea, [3]Department of Computer Engineering, Mokpo National University, Jeonnam, Republic of Korea, [4]Department of Food Engineering Research, Intelligent Robot Studio Co. Ltd., Gyeonggi-do, Republic of Korea

Plant phenotyping is a critical field in agriculture, aiming to understand crop growth under specific conditions. Recent research uses images to describe plant characteristics by detecting visual information within organs such as leaves, flowers, stems, and fruits. However, processing data in real field conditions, with challenges such as image blurring and occlusion, requires improvement. This paper proposes a deep learning-based approach for leaf instance segmentation with a local refinement mechanism to enhance performance in cluttered backgrounds. The refinement mechanism employs Gaussian low-pass and High-boost filters to enhance target instances and can be applied to the training or testing dataset. An instance segmentation architecture generates segmented masks and detected areas, facilitating the derivation of phenotypic information, such as leaf count and size. Experimental results on a tomato leaf dataset demonstrate the system's accuracy in segmenting target leaves despite complex backgrounds. The investigation of the refinement mechanism with different kernel sizes reveals that larger kernel sizes benefit the system's ability to generate more leaf instances when using a High-boost filter, while prediction performance decays with larger Gaussian low-pass filter kernel sizes. This research addresses challenges in real greenhouse scenarios and enables automatic recognition of phenotypic data for smart agriculture. The proposed approach has the potential to enhance agricultural practices, ultimately leading to improved crop yields and productivity.

KEYWORDS

**deep learning, leaf instance segmentation, cluttered background, filtering, plant phenotyping**

# 1 Introduction

Understanding the growth processes of plants is essential for optimizing crop cultivation conditions (Hilty et al., 2021). The interpretation of crop responses is often tied to environmental and nutritional factors, and visual observations of plant development play a significant role in this understanding (Heuvelink, 2005). These visual cues offer tangible evidence of a plant's well-being and the effects of different conditions on its growth. However, comprehending the intricate processes involved in plant growth and development is not a trivial task. It demands a high level of expertise and intuition, acquired through experience and dedicated study. Researchers, agronomists, and farmers continually strive to deepen their knowledge of plant growth processes and develop innovative approaches to harness this understanding for sustainable and efficient agricultural practices (Costa et al., 2019).

Plant development processes, including stems, leaves, flowers, and fruit ripening, directly impact plant yield, quality, and quantity of final products. Phenotyping becomes indispensable in identifying these changes and understanding plant responses (Pieruschka and Schurr, 2019). For example, in tomato plants, critical phenotyping variables such as leaf color, shape, size, and stem diameter offer insights into the plant's health, stress levels, and the potential presence of diseases or pests (Geelen et al., 2018).

Recent advances in computer vision and deep learning have prompted significant interest in plant-related research (Costa et al., 2019). Previous studies have successfully employed techniques (Liu and Wang, 2021) such as image classification, object detection, and instance segmentation for tasks such as detecting diseases and pests (Mohanty et al., 2016; Fuentes et al., 2017; Fuentes et al., 2018; Jiang et al., 2020; Fuentes et al., 2021; Dong et al., 2022), counting leaves (Farjon et al., 2021), and detecting fruits (Afonso et al., 2020). In relation to our research, Das et al. (2023) proposed an ensemble segmentation model with UNet as the base encoder–decoder for detecting coleoptile emergence time, showcasing its potential for phenotyping applications. Similarly, Yang et al. (2020) utilized the Mask Region-based Convolutional Neural Network (Mask R-CNN) architecture for leaf segmentation. The researchers conducted thorough investigations to identify optimal hyperparameters for both segmentation and classification techniques. Despite these significant achievements, the challenge of deploying systems in real-world scenarios with diverse variables and cluttered backgrounds persists (Barbedo, 2018).

In real-world scenarios, plant leaves often overlap or get occluded by other elements, making it challenging for segmentation models to accurately distinguish individual instances (Zhang and Zhang, 2023). Additionally, variations in lighting, shadows, and image quality, with issues like blurred leaves and noise in the images can impact the model's ability to extract meaningful features for accurate segmentation (Rzanny et al., 2017). Moreover, the limited availability of annotated training data for specific plant species (Xu et al., 2023) and growth stages poses a significant challenge in achieving robust and generalized segmentation models (Okyere et al., 2023). Furthermore, existing methods may struggle with instances of varying sizes and shapes, leading to incomplete or inaccurate segmentation results (Yang et al., 2020). Addressing these problems is critical to advancing the field of plant leaf instance segmentation and enabling applications in precision agriculture and automated plant phenotyping.

To address these technical gaps, this paper proposes a systematic deep learning-based approach for leaf instance segmentation in cluttered backgrounds. The study investigates the application of a filter-based instance refinement mechanism to enhance leaf instance segmentation, exploring its application on both training and testing data. Figure 1 showcases the segmentation process of plant leaves within a cluttered greenhouse background. The proposed approach employs a refinement mechanism based that operates locally on target areas, leading to enhanced recognition of individual leaf instances. This refinement step is crucial for overcoming challenges related to occlusion, blurriness, and focus commonly encountered in real-world data collection scenarios. The output of the segmentation process provides segmented masks and bounding box information for each detected leaf instance. Leveraging these results, further processing is conducted to derive essential phenotypic characteristics, including the accurate counting of leaves and the determination of their respective areas. This comprehensive approach not only successfully identifies and segments plant leaves amidst cluttered backgrounds but also enables the extraction of critical phenotypic information that offers valuable insights into the plant's health, growth, and overall performance. The results obtained from this figure demonstrate the effectiveness and potential of the proposed method for advancing plant phenotyping in greenhouse environments, contributing to the optimization of agricultural practices and crop management.

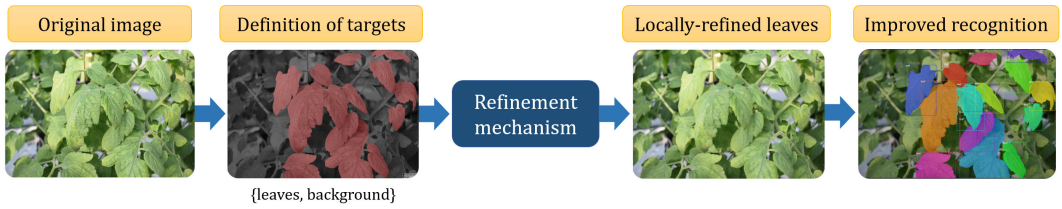



FIGURE 1
Overview of the proposed framework for instance segmentation of plant leaves in cluttered greenhouse backgrounds. It incorporates a refinement mechanism that operates locally on target areas, leading to enhanced recognition of individual leaf instances. The output results from this process allow us to derive essential phenotypic characteristics, including the accurate counting of leaves and the determination of their respective areas.

The contributions of this work are summarized as follows:

- A deep learning-based method for segmenting plant leaf instances, with instance segmentation and mask detection, is proposed and thoroughly validated on experiments conducted on our tomato plant dataset.
- The introduction of a simple yet effective local refinement mechanism based on filtering techniques applied locally to the leaf instances significantly improves the robustness of data used for training and testing, overcoming challenges related to data collection such as occlusion, blurriness, and focus.
- Our study offers a practical method for plant phenotyping using RGB images from real greenhouse environments, providing insights into data utilization for this application.

The rest of the paper is organized as follows: Related works on leaf instance segmentation and plant phenotyping techniques are reviewed in Section 2. The proposed method and strategy are introduced in Section 3. Experimental results, both qualitative and quantitative, are presented in Section 4. Finally, Section 5 concludes the research and outlines potential directions for future work.

# 2 Related works

This section presents an overview of the techniques used for leaf segmentation and plant phenotyping, including both traditional approaches and deep learning-based studies.

## 2.1 Traditional techniques for plant phenotyping

Plant phenotyping is a critical field in agriculture, providing valuable insights into crop growth and characteristics (Walter et al., 2015). Traditional methods have been utilized in this domain, including manual measurements of plant organ features and machine vision techniques for data collection (Kolhar and Jagtap, 2021). For instance, Praveen Kumar and Domnic (2019) employed statistical-based image enhancement, graph-based leaf region extraction, and circular Hough Transform for leaf counting. Zhang et al., 2018 explored plant segmentation using contour techniques and hand-crafted features, while Tian et al., 2019 used an adaptive K-means algorithm for tomato leaf image segmentation. Although these methods can be effective in controlled scenarios, their performance might be limited when applied in real-world situations with diverse variations and challenges.

As agriculture often involves cluttered backgrounds, occlusions, varying lighting conditions, and other complexities, these traditional approaches may struggle to handle the level of intricacy present in real-life environments. Consequently, the adoption of learnable approaches, such as deep learning, becomes more appropriate for tackling these challenging conditions (Xiong et al., 2021).

## 2.2 Leaf instance segmentation in cluttered backgrounds

In recent years, there has been a growing demand for systematic plant phenotyping, leading to increased interest in utilizing deep learning and computer vision-based techniques for image-based plant analysis (Costa et al., 2019; Fuentes et al., 2019). The main objective is to extract meaningful features from specific plant organs, such as leaves, flowers, stems, and fruits, to effectively characterize and evaluate their condition (Singh et al., 2018). Detection or segmentation architectures are commonly employed to provide detailed information at the instance level, such as bounding boxes (Dong et al., 2022) or masks (Xu et al., 2022), which prove valuable for applications such as plant disease and pest detection, as well as leaf, flower, or fruit counting.

The Leaf Segmentation Challenge (LSC) (Scharr et al., 2015) and the Workshop on Computer Vision Problems of Plant Phenotyping (CVPP) (Scharr et al., 2017) have significantly advanced plant phenotyping research. These initiatives aimed to develop state-of-the-art techniques for automatically obtaining phenotyping characteristics, with a particular focus on counting the number of leaves. As part of these efforts, they introduced new datasets with annotation labels for leaves and plants, inspiring various studies to address the challenge. For example, some researchers proposed methods for leaf segmentation using information like leaf borders, color, and texture features (Pape and Klukas, 2015), while others introduced neural network architectures for leaf counting (Aich and Stavness, 2017). Despite having limited training data, these approaches achieved satisfactory results. To tackle the issue of limited data availability, Kuznichov et al. (2019) explored data augmentation techniques to create synthetic samples based on existing data.

In the realm of plant segmentation with complex backgrounds, significant contributions have been made in recent years. For instance, Yang et al. (2020) employed Mask R-CNN with a VGG-16 feature extractor for leaf segmentation in complicated backgrounds, achieving a performance of 91.5%. The dataset used in their study consisted of images with clear leaf information, making leaves easily distinguishable from the background. Similarly, Br et al. (2021) proposed a segmentation method based on leaf images was proposed to identify the attributes of plant diseases. The researchers used a comprehensive dataset of various plant leaf images and developed a two-stream deep learning framework that accurately segments plants and counts leaves of different sizes and shapes. In Fan et al. (2022), the researchers introduced an auxiliary binary mask from the segmentation stream to enhance counting performance, reducing the impact of complex backgrounds. More recently, Lin et al. (2023) proposed a self-supervised semantic segmentation model that groups semantically similar pixels based on self-contained information, enabling a color-based leaf segmentation algorithm to identify leaf regions jointly. Furthermore, they introduced a self-supervised color correction model for images captured under complex illumination conditions.

While substantial progress has been made in plant leaf segmentation, most of the work has focused on outdoor environments, primarily due to the availability of datasets. In contrast, our research focuses on complex real-world greenhouse environments of tomato plants, where challenges such as leaf occlusions and varying scales are prevalent. To address these issues, we introduced a refinement mechanism based on filtering techniques, aiming to enhance the robustness of leaf instance segmentation and overcome the problem of image blurring. Our approach contributes to the advancement of plant phenotyping in challenging greenhouse settings and holds potential implications for agricultural practices and automation.

# 3 Proposed method

This section provides a detailed explanation of the proposed approach and the techniques utilized for segmenting leaf instances in cluttered backgrounds. The primary architecture takes an input image and generates output results in the form of leaf instance masks. A pivotal aspect of our method is the data refinement mechanism, which enhances the robustness of the images used for both training and testing. This is achieved by locally applying filtering techniques to each target leaf instance. The implementation involves two distinct stages: one for training data and another for test data. An overview of the implementation process is illustrated in Figure 2.

## 3.1 Dataset description

In this study, we created a dataset specifically designed for the segmentation of leaf instances and the analysis of cluttered backgrounds. The dataset comprises 372 images of tomato plants, captured using multiple camera devices in various greenhouse environments. The images were taken under changing lighting conditions and feature diverse backgrounds. Each photo was captured parallel to the plants, encompassing surrounding areas as depicted in Figure 3A. The dataset exhibits complexities such as (1) variations in target leaf sizes and appearances, (2) different levels of leaf occlusion, and (3) blurred regions caused by camera movement and focus.

For generating ground-truth data, leaf regions were meticulously annotated using masks, regardless of their visual appearance, encompassing both well-defined and blurred samples. The annotations were performed manually utilizing an available toolbox for mask segmentation, as shown in Figure 3B. Overall, the annotations encompass 3,636 instances, with 2,045 instances allocated to the training set, 641 to the validation set, and 950 to the test set.

## 3.2 Instance segmentation architecture

Leaf instance segmentation has been implemented using Mask R-CNN (He et al., 2017) as the core architecture. Mask R-CNN is a two-stage framework designed for both instance segmentation and object detection tasks. It leverages a Feature Pyramid Network (FPN) as its backbone to extract essential features from input images. In the first stage, a Region Proposal Network (RPN) generates Region of Interest (RoI) proposals, while in the second stage, Mask R-CNN predicts bounding boxes, class labels, and masks for each RoI. The overall architecture for leaf instance segmentation is illustrated in Figure 4.

During training, the end-to-end instance segmentation model aims to minimize the multi-task loss for each sampled RoI, which is composed of three components: classification loss $\boldsymbol{L_{cls}}(\boldsymbol{p_i}, \boldsymbol{p_i^*})$, bounding box regression $\boldsymbol{L_{bbox}}(\boldsymbol{t}, \boldsymbol{t_i^*})$, and mask loss $\boldsymbol{L_{mask}}$ as shown in Equation (1).

$$\boldsymbol{L} = \boldsymbol{L_{cls}} + \boldsymbol{L_{bbo}} + \boldsymbol{L_{mask}} \tag{1}$$

The classification loss is a logarithmic loss over two classes (object or not object) and is computed based on the output score $\boldsymbol{p_i}$ of the classification branch for each anchor $i$ and its corresponding ground-truth label $\boldsymbol{p_i^*}$.

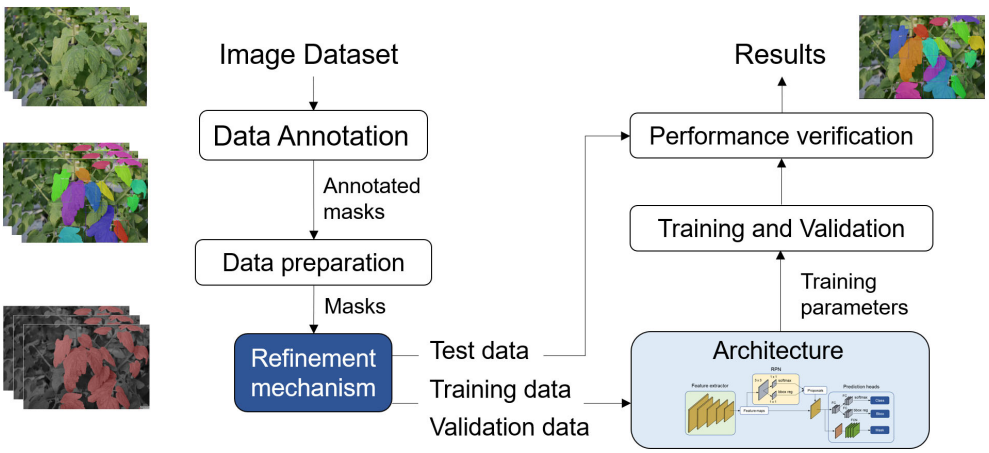



FIGURE 2
Overview of the proposed approach for plant leaf instance segmentation in cluttered backgrounds. The model encompasses two key elements: a refining mechanism directly applied to the data used for training or testing, and an instance segmentation architecture responsible for generating accurate leaf instances in the images.




FIGURE 3
Examples of the tomato plants dataset, showcasing the images of the plants **(A)** alongside their corresponding mask annotations **(B)**. The mask annotations were applied to the foreground leaves, encompassing both clear and blurred samples, to provide comprehensive ground-truth data for the segmentation task.

The regression loss is activated only when the anchor contains an object. It computes the difference between the predicted bounding box parameters $t_i$ and the ground-truth parameters $t_i^*$, which include four variables $[t_x,t_y,t_w,t_h]$, where $(x,y)$ are the coordinates of the bounding box center, and its width and height $(w,h)$.

The mask loss is an average binary cross-entropy loss applied to the dedicated mask branch. As an instance segmentation approach, $L_{mask}$ utilizes the classification branch to allow the network to generate masks for each class separately, avoiding confusion among different categories.

## 3.3 Proposed local refinement mechanism

During data collection for our application, camera focus and blur were the most common image quality issues. These issues had a significant impact, particularly when dealing with cluttered background conditions and defining target areas accurately. Our research aims to address this challenge by introducing a "local refinement mechanism," a simple yet effective technique that enhances the robustness of training and test data. The goal is to enable the system to accurately segment leaves regardless of background information.

After obtaining the annotated dataset, we applied the local refinement mechanism to the instances in both the training and test data. The main methods involved using Gaussian low-pass filtering and High-boost filtering, either independently or in combination, to improve the system's recognition capabilities.

### 3.3.1 Gaussian low-pass filter

GLPF allows transmitting signals with lower frequency, thereby helping to reduce noise and blurring regions in the image (Gonzalez and Woods, 2018). It smooths the image by averaging nearby pixels within a local region, reducing the disparity between pixel values. The effect of image blurring results is larger, as the smoothing mask also becomes larger. The GLPF generates blurring instance regions to assess the model's ability to segment leaves under these

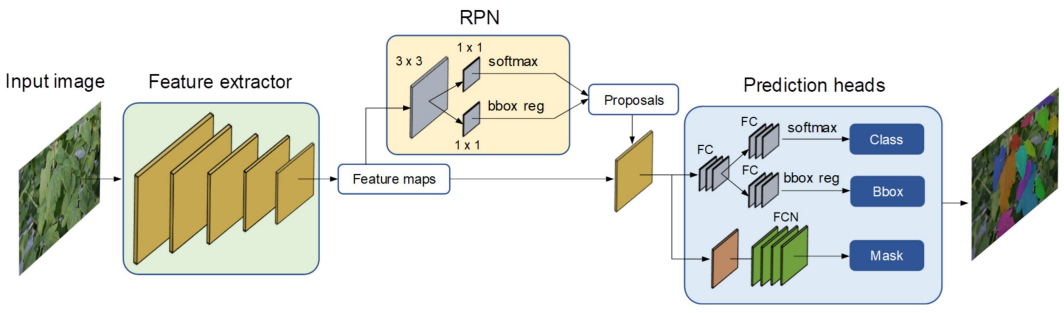



FIGURE 4
Instance segmentation architecture based on Mask R-CNN.

conditions. Equation (2) specifies a GLPF:

$$f_G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

where $x$ is the distance from the center on the horizontal axis, $y$ is the distance from the center to the vertical axis, and σ is the standard deviation of the Gaussian distribution.

### 3.3.2 High-boost filter

HBF emphasizes high-frequency image details without eliminating low-frequency components. It sharpens the image and enhances edges (Gonzalez and Woods, 2018). Multiplying the original image by an amplification factor $A$ yields the definition of an HBF. The value of $A$ determines the nature of the HBF, where higher values lead to brighter backgrounds, resulting in noise enhancement and image sharpening. Equation (3) defines the HBF:

$$f_{HB}(x,y) = (A-1)f(x,y) + f_{hp}(x,y) \quad (3)$$

where $A$ represents the amplification factor, and $f_{hp}$ is a high-pass filter. We applied the HBF locally to leaf instances to improve their regions' sharpness, facilitating leaf boundary detection, especially in cases with occlusion. We experimented with different kernel sizes to find the optimal value for our approach.

We devised two scenarios for applying the refinement mechanism:

- Scenario 1: We aimed to determine whether applying the refinement mechanism enhances the robustness of features in the training dataset, as shown in Figure 5B.
- Scenario 2: We applied the refinement mechanism to the test data to assess whether the features from the training dataset effectively handle changes in the test data, as shown in Figure 5B.

We evaluated the system's response to these changes by applying the local refinement filter with different kernel sizes. Figures 5C, D illustrate example images after applying the GLPF and HBF, respectively. In Section 4, we present the qualitative and quantitative results of our approach. Additional specific illustrations of the applied local refinement mechanism can be found in Figures A1 and A2 of the Appendix. These figures showcase how the mechanism is implemented on both the training and test datasets.

To avoid overfitting, data augmentation techniques were employed to increase the number of images in the training dataset on the two aforementioned cases. From this point onwards, we will use the abbreviation (ATD) to refer to the augmented training dataset. We used both online and offline data augmentation, including intensity and geometric transformations. Specifically, online data augmentation was executed during training, applying operations such as horizontal flip, Gaussian blur, brightness and contrast enhancement, and pixel loss. Offline data augmentation, performed as a separate process to the entire dataset before training, generated more images using techniques such as brightness and contrast enhancement, pixel dropout, horizontal flipping, rotation, and random combinations of all of them.

## 3.4 Evaluation metrics

We evaluated the performance of the proposed model using the Intersection Over Union (IoU) thresholding operation and the mean Average Precision (mAP) metric (Everingham et al., 2009). The standard MS COCO metrics were used for instance segmentation and bounding box detection. The mAP is calculated by computing the AP for each class and then averaging across all classes, taking into account the trade-off between precision and recall, and considering false positives (FPs) and false negatives (FNs). Equation (4) presents the formula for the mAP calculation.

$$\boldsymbol{mAP} = \frac{1}{\boldsymbol{N}} \sum_{\boldsymbol{i}=1}^{\mathrm{N}} \boldsymbol{AP_i} \quad (4)$$

Our primary focus in this evaluation was on the system's ability to accurately identify leaf instances and potentially predict more leaf samples than those available in the training dataset. We present the results of our experiments in the following section to support our claims.

# 4 Experimental results

In this section, we provide the implementation details and present both quantitative and qualitative experimental results on the tomato plants dataset. These evaluations demonstrate the performance of our applied strategy in real-field scenarios.

## 4.1 Implementation details

For our implementation, we fine-tuned the model end to end using a pre-trained model on the MS-COCO dataset. To train the network, we utilized Stochastic Gradient Descent (SGD) along with the Adam optimizer, setting the learning rate to 0.000125, momentum to 0.9, and weight decay to 1e-4. After training the model for 50 epochs, we obtained the final instance segmentation weights. The training process was conducted on a computer equipped with 4 GPUs Titan RTX.

The original images had a size of (4,032 and 3,024), and we resized the input images to (1,333 and 1,000). For implementation, we used the PyTorch framework, where the input tensor size was (6, 3, 1,333, and 1,000), which corresponds to the batch size, number of channels, width, and height, respectively. The first layer of the network used a 7 × 7 kernel size with a stride of 2. In the following convolutional layers, the kernel size was predominantly 3 × 3, and the stride was either 1 or 2, depending on the layer. In the Feature Pyramid Newtok (FPN), 1 × 1 and 3 × 3 convolutional layers were used. ReLU was applied after each convolutional layer to introduce non-linearity into the model. In the final stage of Mask R-CNN, a

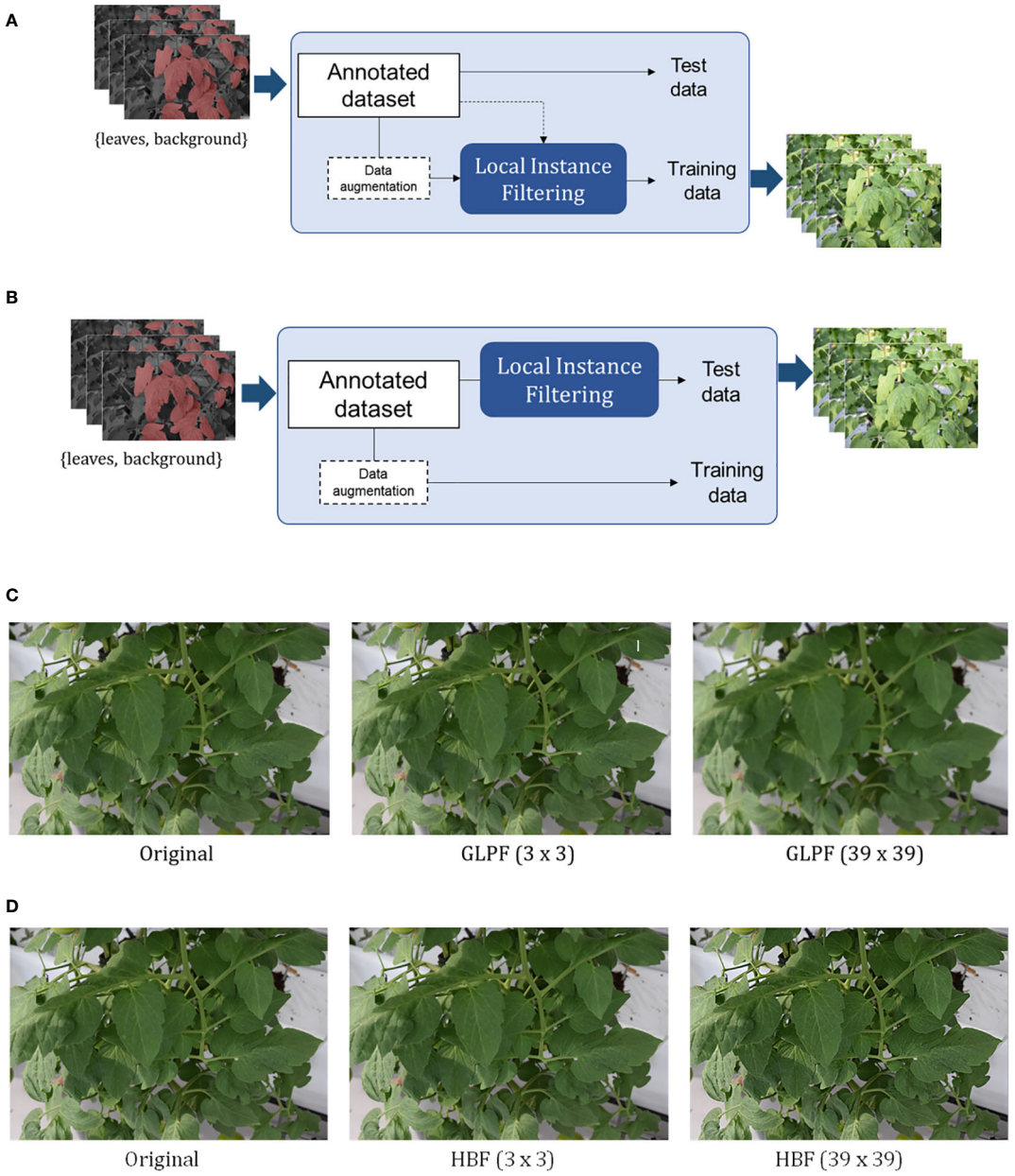



FIGURE 5
Application of the local refinement mechanism either on the training dataset **(A)** or the test dataset **(B)**. The impact of the filters with different kernel sizes on the images is demonstrated in the examples presented in **(C, D)** for the Gaussian low-pass and High-boost filters, respectively. [See Figures A1 and A2 in the Appendix for more detailed representations of the schemes in **(A, B)**].

sigmoid activation function was used in the mask branch. The training curves of the model are presented in Figure A3 in the Appendix.

## 4.2 Backbone feature extractor

We initiated our experiments by comparing the performance of different backbone architectures, namely, ResNet-18, ResNet-34, ResNet-50, and ResNet-101, to determine the most suitable one for our specific application. For this comparison, we directly trained the model using the original images without applying the local refinement mechanism on the leaf instances. The results of this evaluation are presented in Table 1. Among the tested networks, ResNet-50 demonstrated the highest performance in segmenting instance leaves, achieving an IoU > 0.5 of 91.6%. Our findings indicated that Mask R-CNN benefited significantly from deeper networks, particularly ResNet-50. As a result, we selected this architecture as the baseline backbone to conduct further experiments.

TABLE 1 Backbone architecture.

| Model | Segmentation | | | Bounding box detection | | |
|---|---|---|---|---|---|---|
| | $AP_{50}$ | $AP_{75}$ | $AP_{50-95}$ | $AP_{50}$ | $AP_{75}$ | $AP_{50-95}$ |
| ResNet-18 | 73.0 | 37.2 | 38.6 | 72.6 | 29.9 | 35.6 |
| ResNet-34 | 82.1 | 51.6 | 48.1 | 81.4 | 48.8 | 46.7 |
| ResNet-50 | 91.6 | 83.4 | 74.5 | 91.3 | 81.4 | 71.6 |
| ResNet-101 | 90.2 | 75.6 | 67.2 | 89.0 | 69.2 | 60.9 |

## 4.3 Refinement mechanism applied to the training dataset

In this experiment, we focused on evaluating the first scenario presented in Section 3.3 and illustrated in Figure 5A. The goal was to assess the impact of the refinement mechanism when applied to the local leaf instances of the training dataset, with the intention of emulating the presence of blurry leaves in the data. By introducing blurriness, we aimed to generate the necessary features that would allow the model to perform well on the original test dataset, which contains instances of leaves with clearer visual appearance.

To achieve this, we utilized a GLPF in two different configurations:

### 4.3.1 Refinement mechanism applied to the augmented training dataset

In this configuration, the model was trained on the augmented training dataset, which included instances of leaves with varying levels of blurriness introduced through the GLPF. The objective here was to assess the model's ability to generalize effectively on the test data, which comprises images of original uncorrupted leaves. Figure A1 A in the Appendix illustrates the implemented strategy for this scenario.

### 4.3.2 Refinement mechanism applied to the augmented training dataset and combined with the original samples

In this case, we combined the blurred dataset with the original augmented dataset. The purpose was to provide the model with more detailed features of the target areas, and the refinement mechanism acted as a type of data augmentation technique. However, for our specific task, we aimed to examine its impact as part of a partially corrupted dataset. Figure A1 B in the Appendix shows the strategy implemented for this configuration.

To comprehensively evaluate the model's performance under different settings, we conducted a thorough analysis involving the number of predicted masks corresponding to leaves and the AP on the test dataset. This evaluation was carried out by applying various kernel sizes for the GLPF, which introduced multiple levels of blurring in the training data. To ensure the reliability of our findings, we conducted three rounds of model training and calculated the standard deviation.

The results presented in Table 2 unveiled two prominent trends: In the first scenario, where the refinement mechanism was applied solely to the ATD, we observed a slight reduction in AP. However, an interesting phenomenon occurred; the model seemed to learn to associate the noise generated by applying the GLPF. Consequently, while the AP decreased slightly, the number of detected masks increased. This intriguing observation suggests that the model acquired enhanced capabilities to handle such blurred data during training, thereby becoming more robust against such changes.

In contrast, the second scenario, where original data were combined with the ATD, revealed a different outcome. Here, the performance of the model decreased, accompanied by a decline in the number of predicted masks. This decline can be attributed to the model's primary focus on recognizing clear data. Consequently, when confronted with blurred data, the model became frequently

TABLE 2 Results of the refinement mechanism applied to the training dataset.

| Model | Kernel size | Predicted masks | Segmentation | | | Bounding box detection | | |
|---|---|---|---|---|---|---|---|---|
| | | | $AP_{50}$ | $AP_{75}$ | $AP_{50-95}$ | $AP_{50}$ | $AP_{75}$ | $AP_{50-95}$ |
| Baseline | – | 889 ± 11 | 91.5 ± 0.4 | 83.5 ± 0.5 | 74.5 ± 0.1 | 91.2 ± 0.3 | 80.9 ± 1.4 | 71.4 ± 0.7 |
| GLPF on applied to the ATD | 5 | 897 ± 3 | 91.9 ± 0.1 | 83.3 ± 1.8 | 75.0 ± 0.3 | 91.3 ± 0.8 | 80.4 ± 1.1 | 71.4 ± 0.3 |
| | 7 | 909 ± 4 | 91.3 ± 0.2 | 83.3 ± 1.0 | 74.6 ± 0.3 | 91.2 ± 0.2 | 80.5 ± 0.3 | 71.1 ± 0.4 |
| | 9 | 915 ± 4 | 91.5 ± 0.1 | 83.3 ± 0.5 | 74.6 ± 0.2 | 91.1 ± 0.2 | 80.3 ± 1.2 | 71.1 ± 0.1 |
| GLPF applied to the ATD + OI | 5 | 863 ± 4 | 89.7 ± 0.4 | 82.2 ± 0.8 | 74.4 ± 0.1 | 88.9 ± 0.6 | 79.3 ± 1.0 | 0.9 ± 0.2 |
| | 7 | 865 ± 7 | 89.7 ± 0.3 | 82.2 ± 1.1 | 74.2 ± 0.2 | 89.4 ± 0.2 | 79.2 ± 0.6 | 70.8 ± 0.1 |
| | 9 | 870 ± 2 | 90.1 ± 0.8 | 82.2 ± 1.0 | 74.3 ± 0.6 | 89.2 ± 0.3 | 79.6 ± 0.2 | 71.2 ± 0.9 |

ATD, Augmented training dataset.
OI, Original images.

confused, leading to a drop in performance. As the kernel size for the GLPF increased and blurring became more severe, this confusion further exacerbated the model's inability to accurately segment leaves.

These findings strongly indicate that the blurring data introduced by the GLPF, when applied to the training dataset, significantly contributed to making the model robust against blurring effects in the data. Consequently, this adaptation played a vital role in improving the model's ability to accurately segment leaves. Figure 6 provided some qualitative examples of the model's performance, further highlighting the challenges and limitations posed by introducing blurriness in the training dataset.

This study showcased the significance of the refinement mechanism, particularly when applied to the ATD, in enhancing the model's robustness against blurriness in the data, leading to improved leaf instance segmentation performance. However, caution is required when combining original and blurred data during training, as it may adversely affect the model's ability to handle blurriness. These insights have practical implications for real-world applications.

## 4.4 Refinement mechanism applied to the test data

In the previous experiment, we applied the refinement mechanism to the training data, which resulted in a decline in performance. To address this challenge, we conducted two additional experiments, focusing on the test dataset to explore alternative solutions. These experiments correspond to the second scenario outlined in Section 3.3 and Figure 5B, and their outcomes are summarized in Table 3.

In this experiment, we employed the refinement mechanism in two different configurations:

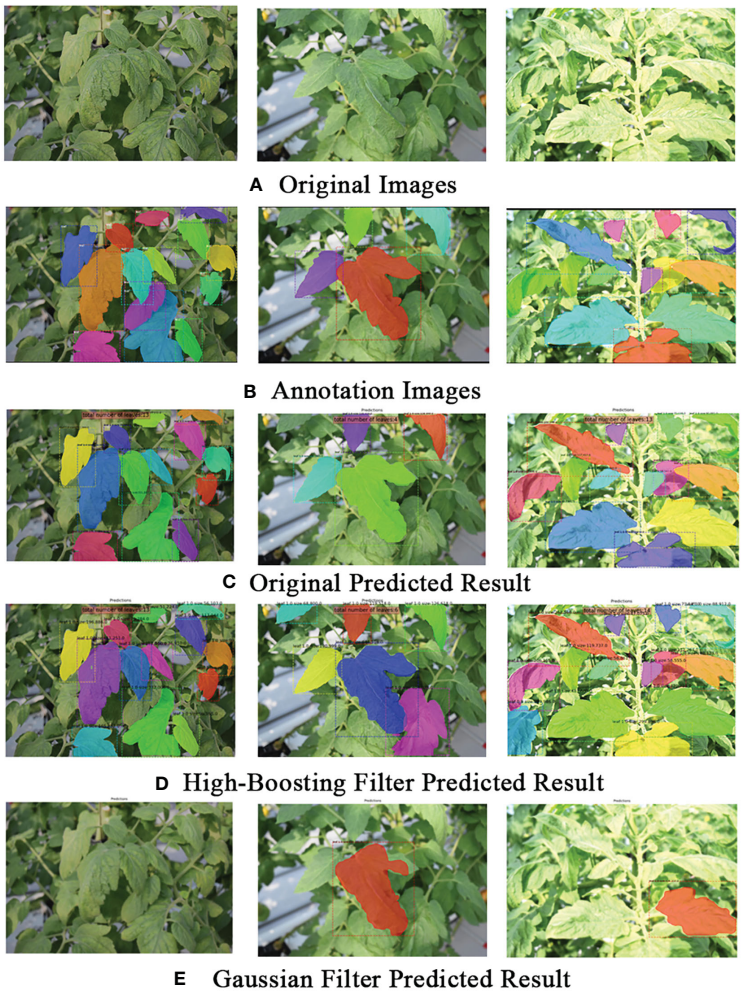



FIGURE 6
Example qualitative results on the tomato plant dataset. **(A)** Original images. **(B)** Ground truth (actual annotations). **(C)** Predicted results on the original images. **(D)** Predicted results using Gaussian low-pass filter on the training dataset. **(E)** Predicted results using the High-boost filter on the test dataset. The visual comparison highlights how different approaches, such as applying filters to the training or test datasets, influence the model's predictions.

TABLE 3 Results of the refinement mechanism applied as postprocessing.

| Method | Kernel size | Predicted masks | Segmentation | | | Bounding box detection | | |
|---|---|---|---|---|---|---|---|---|
| | | | $AP_{50}$ | $AP_{75}$ | $AP_{50-95}$ | $AP_{50}$ | $AP_{75}$ | $AP_{50-95}$ |
| Baseline | - | 889 ± 11 | 91.5 ± 0.4 | 83.5 ± 0.5 | 74.5 ± 0.1 | 91.2 ± 0.3 | 80.9 ± 1.4 | 71.4 ± 0.7 |
| GLPF applied to the test dataset | 5 | 869 ± 15 | 91.2 ± 1.1 | 82.6 ± 1.3 | 73.9 ± 0.5 | 90.4 ± 0.5 | 79.5 ± 0.4 | 70.9 ± 0.4 |
| | 7 | 840 ± 38 | 90.4 ± 1.1 | 81.6 ± 0.8 | 72.7 ± 0.4 | 89.7 ± 1.2 | 75.8 ± 1.5 | 72.8 ± 0.4 |
| | 9 | 811 ± 48 | 89.2 ± 1.0 | 81.1 ± 1.6 | 71.4 ± 0.5 | 88.7 ± 2.0 | 77.8 ± 1.0 | 68.5 ± 0.9 |
| HBF applied to the test dataset | 5 | 901 ± 4 | 92.0 ± 0.8 | 83.9 ± 0.7 | 75.1 ± 0.3 | 91.8 ± 0.5 | 81.6 ± 0.7 | 72.0 ± 0.3 |
| | 7 | 909 ± 13 | 92.3 ± 0.8 | 83.8±1.2 | 75.3±0.6 | 91.9±1.2 | 82.1±2.6 | 72.9±2.5 |
| | 9 | 918 ± 2 | 92.7 ± 0.9 | 84.5±1.3 | 75.9±1.1 | 92.6±1.5 | 82.8±3.3 | 73.5±3.8 |

#### 4.4.1 Refinement mechanism with GLPF applied to the test dataset

The objective of this experiment was to assess how the presence of instance blurriness in the test data influences the model's predictions. As revealed by the results in Table 3, increasing the kernel size of the GLPF had an adverse effect on both AP and the number of predicted masks. Larger kernel sizes caused the RoIs to become more blurred, resulting in a challenging situation for the model to accurately detect the presence of leaves. The leaves tended to merge with the background, leading to a reduction in overall performance.

#### 4.4.2 Refinement mechanism with HBF applied to the test dataset

In this case, we sought to determine whether applying HBF to the test data, utilizing the refined instances, could enhance the prediction of leaf samples (see Figure A2 A in the Appendix for the implemented strategy). As indicated in Table 3, by locally applying HBF, the system predicted more leaves, a favorable outcome for downstream processing to obtain phenotypic data. Notably, the AP also improved for both segmentation and bounding box detection, signifying an overall enhancement in performance compared with the baseline.

The results of these experiments demonstrate the advantageous impact of the refinement mechanism, particularly when using HBF. The HBF approach enabled the model to capture more intricate information, resulting in an increased number of correctly predicted leaf instances. While the application of GLPF had a detrimental impact due to increased blurriness, the usage of HBF significantly improved the prediction of leaf instances, contributing to a more effective and precise segmentation.

Figure 6E provides an example of a qualitative result, showcasing the visual impact of the strategy on the model's predictions. This illustration further supports the effectiveness of using the refinement mechanism with HBF in improving leaf instance segmentation in the tomato plant dataset.

### 4.5 Effects of the implemented strategies

#### 4.5.1 Effect of the refined data by HBF

To gain further insights into the contribution and impact of the refinement mechanism, we conducted an in-depth analysis using both GLPF and HBF on the test dataset. First, we applied a GLPF to the test dataset, generating fuzzy instances, and then consecutively applied an HBF to the same areas. For this analysis, we utilized the weights of the model trained with the original augmented images to make predictions on the test data. (See Figure A2 B in the Appendix for the implemented strategy).

Figure 7A illustrates the changes in the predicted leaf instances based on the size of the HBF core, taking into account the accepted level of blur given by the GLPF. It becomes evident that the model started to benefit from an HBF kernel size greater than $7 \times 7$ while being constrained by a GLPF kernel size of $3 \times 3$ or $5 \times 5$. Furthermore, a trade-off between blurriness and refinement was observed. Larger HBF kernel sizes, such as $15 \times 15$, exhibited better performance, generating more accurately segmented leaves than those present in the original test data. Additionally, we computed the average change rate (ave) for the GLPF kernel sizes, and it became apparent that the model was generally influenced by more significant levels of blurriness provided by the GLPF.

#### 4.5.2 Effect of the blurred data by GLPF

The effect of the blurred data by the GLPF is depicted in Figure 7B, showing the corresponding impact of applying GLPF on the instances of the test data. We used the results obtained with different kernel sizes to measure the changes in predicted leaf instances. Consistent with the findings in Section 4.4, it was observed that the level of blur introduced by the GLPF, based on its kernel size, negatively affected the number of predicted masks. As a result, larger values of kernel size led to a reduction in the presence of predicted leaves.

#### 4.5.3 Effect of the refinement mechanism on the prediction of ground-truth labels

Figure 7C complements the aforementioned analysis by showing the performance gain of the predicted instances compared with the ground truth of the test data. The application of HBF substantially improved the predictions regardless of the presence of blur samples. The performance enhancement was found to be dependent on the size of the kernel. Specifically, a $15 \times 15$ kernel size positively influenced the final results, effectively overcoming the issues caused by GLPF blurring effects.

To visually illustrate the effects of the refinement mechanism on the test data with GLPF and HBF, we present qualitative examples




FIGURE 7
Effects of the implemented refinement strategy on the predicted leaf instances. **(A)** Effect of HBF and GLPF kernel sizes: A kernel size of 15 × 15 positively influenced the model's performance, resulting in more segmented regions compared with the original test dataset. The "ave" value represents the average change rate across all kernel sizes. **(B)** Effect of GLPF kernel sizes: The level of blurriness had a negative impact on the number of predicted samples. Larger kernel sizes resulted in reduced presence of predicted leaves. **(C)** Improved segmentation of leaves through HBF on GLPF-filtered instances: HBF significantly enhanced the segmentation of leaves, based on the ground-truth labels in the test data, even when blurriness was present in the GLPF-filtered samples.

in Figures 8 and 9. The figures showcase two cases: one with multiple leaves (Figure 8) and the other with few leaves (Figure 9). Notably, the use of GLPF and HBF resulted in contrasting performance. While larger kernel sizes of the GLPF negatively impacted the prediction of the ground truth, the larger kernel sizes of the HBF proved beneficial by increasing the number of correctly predicted samples without compromising performance. The HBF effectively enhanced the clarity of RoIs and counteracted the blurring effects of GLPF. Consequently, the model segmented more leaves when the HBF was applied. However, it is important to note that this outcome was highly dependent on the size of the kernel used by the HBF filter.

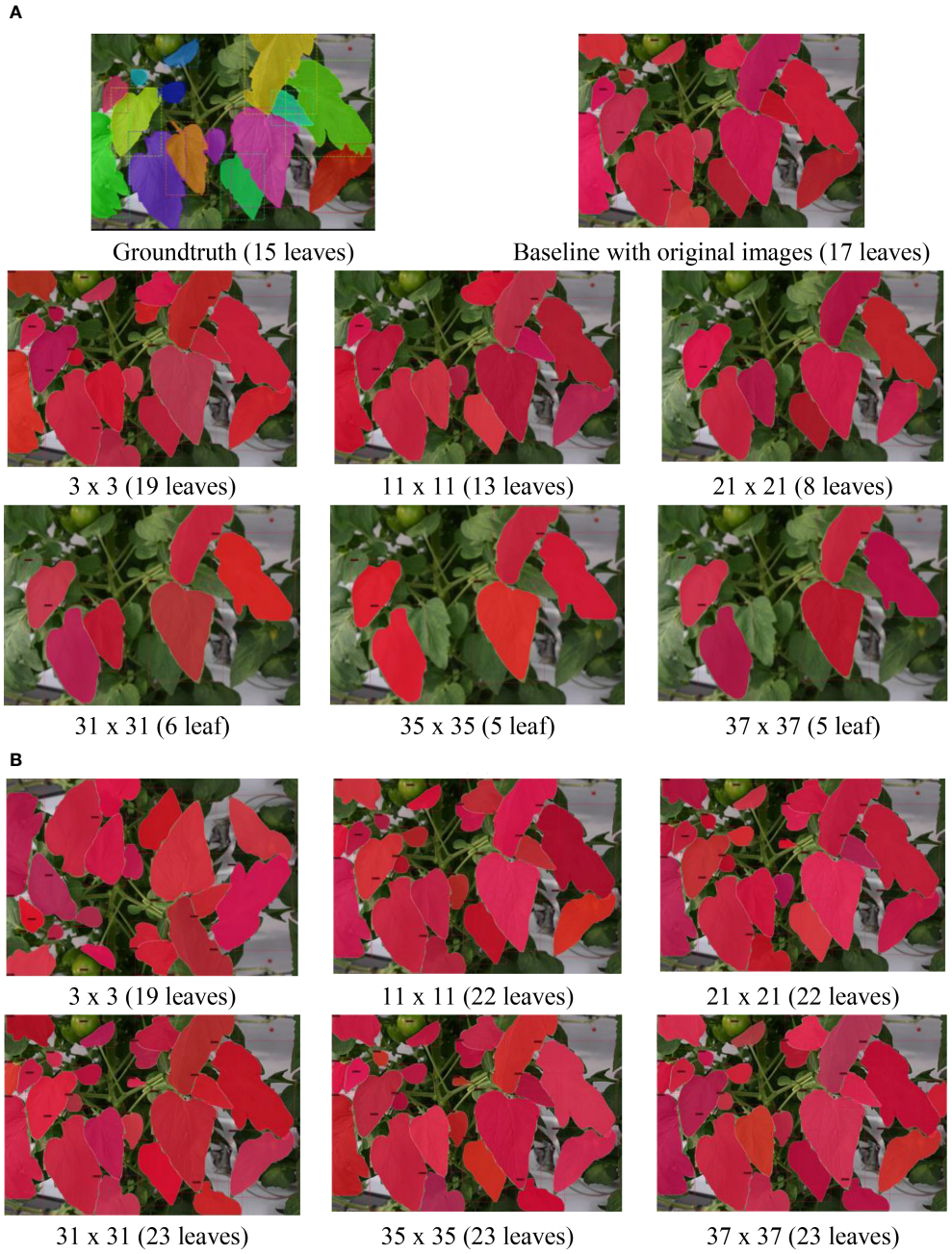



FIGURE 8
Example results of applying **(A)** GLPF and **(B)** HBF on the test data using an image with multiple leaves. As the GLPF kernel size increased, the prediction performance declined. However, with HBF, the system benefited from larger kernel sizes, resulting in the generation of more accurately segmented leaf instances.

## 4.6 Comparison with other state-of-the-art architectures

In order to thoroughly assess the effectiveness of our refinement mechanism, we conducted comparative experiments using the HBF on the test data alongside other state-of-the-art methods such as PointRend (Kirillov et al., 2020), Mask Scoring R-CNN (Huang et al., 2019), CARAFE (Wang et al., 2020), Hybrid Task Cascade (HTC) (Wang et al., 2020), Cascade R-CNN (Cai and Vasconcelos, 2018), and Mask R-CNN (He et al., 2017). To ensure fair comparisons, all models were based on the Albumentation transformations method, with (w) and without (w/o) the inclusion of our refinement strategy (Buslaev et al., 2020).

The experimental results, presented in Table 4, clearly demonstrate that the proposed refinement strategy significantly improved the performance of all implemented models. Regarding segmentation metrics, Mask R-CNN with the refinement strategy achieved the highest performance with an AP of 92.7% when IoU > 0.5. The HTC model also exhibited comparable capabilities with an $AP_{50}$ score of 92.1% when using our strategy. Notably, the Cascade




FIGURE 9
Example results of applying **(A)** GLPF and **(B)** HBF on the test data using an image with few leaves. As the GLPF kernel size increased, the prediction performance declined. However, with HBF, the system benefited from larger kernel sizes, resulting in the generation of more accurately segmented leaf instances.

R-CNN model exhibited the highest improvement of 3.2% after incorporating our refinement mechanism.

In terms of bounding box detection, our improved Mask R-CNN achieved the top score with an $AP_{50}$ of 92.6%. Among the models, Mask Scoring R-CNN displayed the most substantial improvement in performance, with an $AP_{50}$ score of 87.7%, representing an increase of approximately 18.7%. Overall, all models experienced performance gains through the application of our refinement strategy, demonstrating its effectiveness in enhancing leaf instance segmentation in cluttered background conditions.

## 5 Conclusion

This paper introduced an approach for leaf instance segmentation based on deep learning, specifically this research

TABLE 4 Comparison with other state-of-the-art methods.

| Model | Refinement | | Segmentation | | | Bounding box detection | | |
|---|---|---|---|---|---|---|---|---|
| | w/o | w | $AP_{50}$ | $AP_{75}$ | $AP_{50-95}$ | $AP_{50}$ | $AP_{75}$ | $AP_{50-95}$ |
| Point Rend | ✓ | | 86.9 | 82.0 | 76.4 | 85.8 | 80.6 | 74.1 |
| | | ✓ | 88.8 | 85.0 | 78.4 | 88.6 | 82.7 | 76.1 |
| Mask scoring R-CNN | ✓ | | 86.3 | 79.4 | 72.4 | 69.0 | 86.5 | 76.9 |
| | | ✓ | 87.4 | 80.5 | 73.6 | 87.7 | 79.7 | 70.8 |
| CARAFE | ✓ | | 89.2 | 84.4 | 76.8 | 88.1 | 82.3 | 74.9 |
| | | ✓ | 91.2 | 85.7 | 78.2 | 90.0 | 84.7 | 76.2 |
| Cascade R-CNN | ✓ | | 86.6 | 82.9 | 75.9 | 85.5 | 80.0 | 75.1 |
| | | ✓ | 89.8 | 86.1 | 78.7 | 89.3 | 83.0 | 78.0 |
| Hybrid task cascade | ✓ | | 91.8 | 85.5 | 77.9 | 91.4 | 81.9 | 75.0 |
| | | ✓ | 92.1 | 86.0 | 78.4 | 91.7 | 83.3 | 75.7 |
| Mask R-CNN | ✓ | | 91.6 | 83.4 | 74.5 | 91.3 | 81.4 | 71.6 |
| | | ✓ | 92.7 | 84.5 | 75.9 | 92.6 | 82.8 | 73.5 |

represents a significant step forward in the domain of leaf instance segmentation, offering an innovative and effective approach to tackle the challenges associated with cluttered backgrounds and varying image quality. Through the integration of a local refinement mechanism, we have demonstrated improvements in the accuracy and robustness of leaf instance segmentation. Our proposed refinement mechanism, incorporating Gaussian low-pass and HBF, serves as a key driver behind the effectiveness of our approach. The ability to apply this mechanism either during training or on the test dataset highlights its versatility and adaptability to different scenarios. The refined feature representations within leaf instances enabled the model to better distinguish target leaves, even in the presence of blurriness and cluttered backgrounds. Our qualitative and quantitative experimental results performed on our tomato leaf dataset reinforced the reliability and accuracy of our system in data from real-world greenhouse scenarios. The ability to accurately segment target leaves despite challenging conditions, such as occlusion and overlapping, highlights the potential applications of our approach in plant phenotyping.

## Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation. For details contact at afuentes@jbnu.ac.kr.

## Author contributions

RM performed the experiments. AF collaborated on the framework design, and data acquisition, and wrote the manuscript. DP and SY advised on the system's design, analyzed the strategies, and supervised its development. SK, HK, and WL collaborated on the project and its implementation. All authors contributed to the article and approved the submitted version.

## Funding

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2019R1A6A1A09031717); by Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry (IPET) and Korea Smart Farm R&D Foundation (KosFarm) through Smart Farm Innovation Technology Development Program, funded by Ministry of Agriculture, Food and Rural Affairs (MAFRA) and Ministry of Science and ICT(MSIT), Rural Development Administration (RDA)(1545027569); and in part by the Agricultural Science and Technology Development Cooperation Research Program (RS-2021-RD009890).

## Conflict of interest

Author WL is employed by Intelligent Robot Studio Co. Ltd.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

# References

Afonso, M., Fonteijn, H., Fiorentin, F. S., Lensink, D., Mooij, M., Faber, N., et al. (2020). Tomato fruit detection and counting in greenhouses using deep learning. *Front. Plant Sci.* 11, 1759. doi: 10.3389/FPLS.2020.571299/BIBTEX

Aich, S., and Stavness, I. (2017). Leaf counting with deep convolutional and deconvolutional networks *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*. 2080–2089. doi: 10.1109/ICCVW.2017.244

Barbedo, J. G. A. (2018). Factors influencing the use of deep learning for plant disease recognition. *Biosyst. Eng.* 172, 84–91. doi: 10.1016/J.BIOSYSTEMSENG.2018.05.013

Br, P., Av, S. H., and Ashok, A. (2021). "Diseased leaf segmentation from complex background using indices based histogram," in *Proceedings of the 6th International Conference on Communication and Electronics Systems, ICCES 2021*. 1502–1507. doi: 10.1109/ICCES51350.2021.9489112

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., and Kalinin, A. A. (2020). Albumentations: fast and flexible image augmentations. *Inf.* 11 (2), 125. doi: 10.3390/INFO11020125

Cai, Z., and Vasconcelos, N. (2018). "Cascade R-CNN: delving into high quality object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6154–6162. doi: 10.1109/CVPR.2018.00644

Costa, C., Schurr, U., Loreto, F., Menesatti, P., and Carpentier, S. (2019). Plant phenotyping research trends, a science mapping approach. *Front. Plant Sci.* 9, 426195. doi: 10.3389/FPLS.2018.01933/BIBTEX

Das, A., Das Choudhury, S., Das, A. K., Samal, A., and Awada, T. (2023). EmergeNet: A novel deep-learning based ensemble segmentation model for emergence timing detection of coleoptile. *Front. Plant Sci.* 14, 1084778. doi: 10.3389/FPLS.2023.1084778/BIBTEX

Dong, J., Lee, J., Fuentes, A., Xu, M., Yoon, S., Lee, M. H., et al. (2022). Data-centric annotation analysis for plant disease detection: Strategy, consistency, and performance. *Front. Plant Sci.* 13, 1037655. doi: 10.3389/FPLS.2022.1037655/BIBTEX

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., Zisserman, A., et al. (2009). The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vision* 88 (2), 303–338. doi: 10.1007/S11263-009-0275-4

Fan, X., Zhou, R., Tjahjadi, T., Das Choudhury, S., and Ye, Q. (2022). A segmentation-guided deep learning framework for leaf counting. *Front. Plant Sci.* 13, 844522. doi: 10.3389/FPLS.2022.844522/BIBTEX

Farjon, G., Itzhaky, Y., Khoroshevsky, F., and Bar-Hillel, A. (2021). Leaf counting: fusing network components for improved accuracy. *Front. Plant Sci.* 12, 1063. doi: 10.3389/FPLS.2021.575751/BIBTEX

Fuentes, A., Yoon, S., Kim, S., and Park, D. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* 17 (9), 2022. doi: 10.3390/s17092022

Fuentes, A. F., Yoon, S., Lee, J., and Park, D. S. (2018). High-performance deep neural network-based tomato plant diseases and pests diagnosis system with refinement filter bank. *Front. Plant Sci.* 9, 1162. doi: 10.3389/FPLS.2018.01162/BIBTEX

Fuentes, A., Yoon, S., Lee, M. H., and Park, D. S. (2021). Improving accuracy of tomato plant disease diagnosis based on deep learning with explicit control of hidden classes. *Front. Plant Sci.* 12, 2938. doi: 10.3389/FPLS.2021.682230/BIBTEX

Fuentes, A., Yoon, S., and Park, D. S. (2019). Deep learning-based phenotyping system with glocal description of plant AnoMalies and symptoms. *Front. Plant Sci.* 10, 460700. doi: 10.3389/FPLS.2019.01321/BIBTEX

Geelen, P. A. M., Voogt, J. O., and van Weel, P. A. (2018). *Plant Empowerment: The Basic Principles: how an Integrated Approach Based on Physics and Plant Physiology Leads to a Balanced Growing Method for Protected Crops Resulting in Healthy Resilient Plants, High Yield and Quality, Low Energy Costs and Economic* (The Netherlands: LetsGrow.com).

Gonzalez, R., and Woods, R. (2018). *Digital Image Processing. 4th edn* (New York: Pearson).

He, K., Gkioxari, G., Dollar, P., and Girshick, R. (2017). "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 2980–2988. doi: 10.1109/ICCV.2017.322

Heuvelink, E. (2005). *Tomatoes. 13th edn* (CABI: CABI Publishing). doi: 10.1079/9780851993966.0000

Hilty, J., Muller, B., Pantin, F., and Leuzinger, S. (2021). Plant growth: the what, the how, and the why. *New Phytol.* 232 (1), 25–41. doi: 10.1111/NPH.17610

Huang, Z., Huang, L., Gong, Y., Huang, C., and Wang, X. (2019). "Mask scoring R-CNN," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June. 6402–6411. doi: 10.1109/CVPR.2019.00657

Jiang, D., Li, F., Yang, Y., and Yu, S. (2020). "A tomato leaf diseases classification method based on deep learning," in *Proceedings of the 32nd Chinese Control and Decision Conference, CCDC 2020*. 1446–1450. doi: 10.1109/CCDC49329.2020.9164457

Kirillov, A., Wu, Y., He, K., and Girshick, R. (2020). "PointRend: image segmentation as rendering," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9796–9805. doi: 10.1109/CVPR42600.2020.00982

Kolhar, S., and Jagtap, J. (2021). Plant trait estimation and classification studies in plant phenotyping using machine vision – A review. *Inf. Process. Agric.* 10, 114–135. doi: 10.1016/J.INPA.2021.02.006

Kuznichov, D., Zvirin, A., Honen, Y., and Kimmel, R. (2019). "Data augmentation for leaf segmentation and counting tasks in rosette plants," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2019-June. 2580–2589. doi: 10.1109/CVPRW.2019.00314

Lin, X., Li, C. T., Adams, S., Kouzani, A. Z., Jiang, R., He, L., et al. (2023). Self-supervised leaf segmentation under complex lighting conditions. *Pattern Recognit.* 135, 109021. doi: 10.1016/J.PATCOG.2022.109021

Liu, J., and Wang, X. (2021). Plant diseases and pests detection based on deep learning: a review. *Plant Methods* 17 (1), 1–18. doi: 10.1186/S13007-021-00722-9/TABLES/4

Mohanty, S. P., Hughes, D. P., and Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Front. Plant Sci.* 7, 1–10. doi: 10.3389/fpls.2016.01419

Okyere, F. G., Cudjoe, D., Sadeghi-Tehran, P., Virlet, N., Riche, A. B., Castle, M., et al. (2023). Machine learning methods for automatic segmentation of images of field- and glasshouse-based plants for high-throughput phenotyping. *Plants* 12 (10), 2035. doi: 10.3390/PLANTS12102035/S1

Pape, J.-M., and Klukas, C. (2015). Utilizing machine learning approaches to improve the prediction of leaf counts and individual leaf segmentation of rosette plant images. in *Proceedings of the Computer Vision Problems in Plant Phenotyping (CVPPP)* 3, 1–12. doi: 10.5244/C.29.CVPPP.3

Pieruschka, R., and Schurr, U. (2019). Plant phenotyping: Past, present, and future. *Plant Phenomics* 2019, 7507131. doi: 10.34133/2019/7507131

Praveen Kumar, J., and Domnic, S. (2019). Image based leaf segmentation and counting in rosette plants. *Inf. Process. Agric.* 6 (2), 233–246. doi: 10.1016/J.INPA.2018.09.005

Rzanny, M., Seeland, M., Wäldchen, J., and Mäder, P. (2017). Acquiring and preprocessing leaf images for automated plant identification: Understanding the tradeoff between effort and information gain. *Plant Methods* 13 (1), 1–11. doi: 10.1186/S13007-017-0245-8/FIGURES/8

Scharr, H., Minervini, M., French, A. P., Klukas, C., Kramer, D. M., Liu, X., et al. (2015). Leaf segmentation in plant phenotyping: a collation study. *Mach. Vision Appl.* 27 (4), 585–606. doi: 10.1007/S00138-015-0737-3

Scharr, H., Pridmore, T., and Tsaftaris, S. A. (2017). "Computer vision problems in plant phenotyping, CVPPP 2017: introduction to the CVPPP 2017 workshop papers," in *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, 2018-January. 2020–2021. doi: 10.1109/ICCVW.2017.236

Singh, A. K., Ganapathysubramanian, B., Sarkar, S., and Singh, A. (2018). Deep learning for plant stress phenotyping: trends and future perspectives. *Trends Plant Sci.* 23 (10), 883–898. doi: 10.1016/J.TPLANTS.2018.07.004

Tian, K., Li, J., Zeng, J., Evans, A., and Zhang, L. (2019). Segmentation of tomato leaf images based on adaptive clustering number of K-means algorithm. *Comput. Electron. Agric.* 165, 104962. doi: 10.1016/J.COMPAG.2019.104962

Walter, A., Liebisch, F., and Hund, A. (2015). Plant phenotyping: From bean weighing to image analysis. *Plant Methods* 11 (1), 1–11. doi: 10.1186/S13007-015-0056-8/FIGURES/3

Wang, J., Chen, K., Xu, R., Liu, Z., Loy, C. C., and Lin, D. (2019). "CARAFE: content-aware ReAssembly of FEatures'. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3007–3016).

Xiong, J., Yu, D., Liu, S., Shu, L., Wang, X., and Liu, Z. (2021). A review of plant phenotypic image recognition technology based on deep learning. *Electronics* 10 (1), 81. doi: 10.3390/ELECTRONICS10010081

Xu, M., Yoon, S., Fuentes, A., Yang, J., and Park, D. S. (2022). Style-consistent image translation: a novel data augmentation paradigm to improve plant disease recognition. *Front. Plant Sci.* 12, 773142. doi: 10.3389/FPLS.2021.773142/BIBTEX

Xu, M., Kim, H., Yang, J., Fuentes, A., Meng, Y., Yoon, S., et al. (2023) *Embrace Limited and Imperfect Training Datasets: Opportunities and Challenges in Plant Disease Recognition Using Deep Learning*. Available at: https://arxiv.org/abs/2305.11533v2 (Accessed 31 July 2023).

Yang, K., Zhong, W., and Li, F. (2020). Leaf segmentation and classification with a complicated background using deep learning. *Agronomy* 10 (11), 1721. doi: 10.3390/AGRONOMY10111721

Zhang, J., Kong, F, Wu, J., Han, S., and and Zhai, Z. (2018). Automatic image segmentation method for cotton leaves with disease under natural environment. *J. Integr. Agric.* 17 (8), 1800–1814. doi: 10.1016/S2095-3119(18)61915-X

Zhang, S., and Zhang, C. (2023). Modified U-Net for plant diseased leaf image segmentation. *Comput. Electron. Agric.* 204, 107511. doi: 10.1016/J.COMPAG.2022.107511

# Appendix

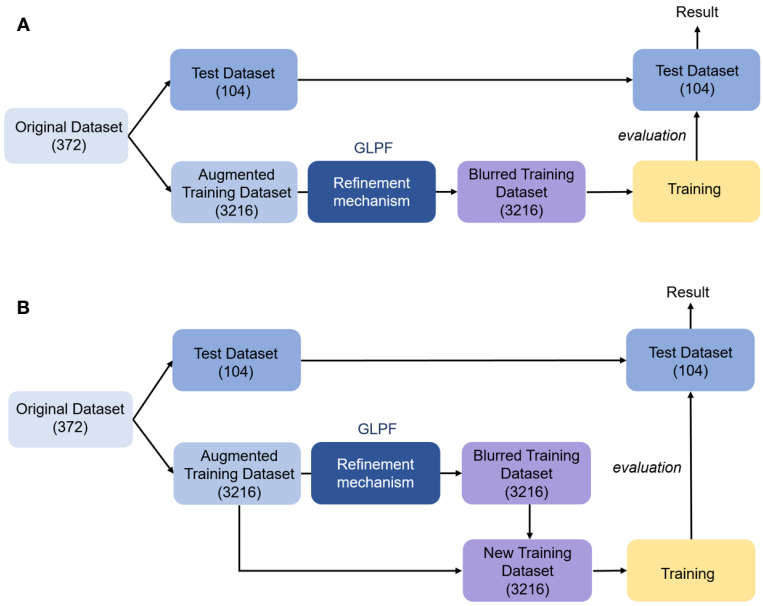



FIGURE A1
Local refinement mechanism (GLPF) applied to the augmented training dataset. **(A)** Applying the mechanism to the original images. **(B)** Combining the blurred dataset with the original images. The number inside the parenthesis shows the number of images.

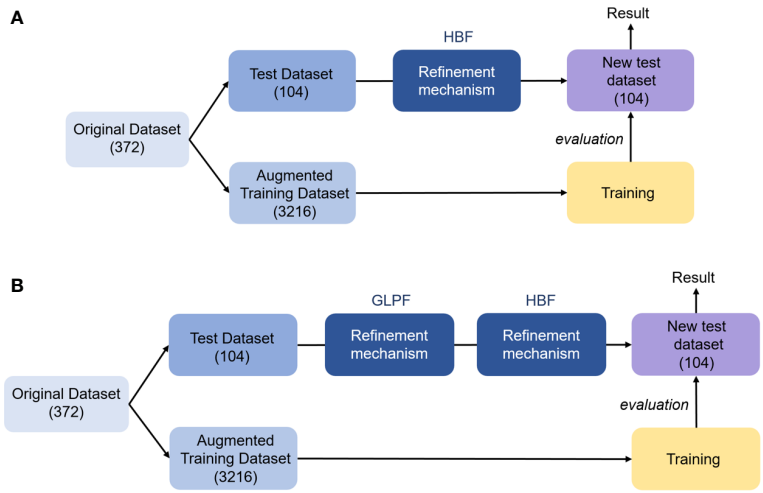



FIGURE A2
Local refinement mechanism applied to the test dataset. **(A)** HBF. **(B)** GLPF followed by HBF. The number inside the parenthesis shows the number of images.

10.3389/fpls.2023.1211075

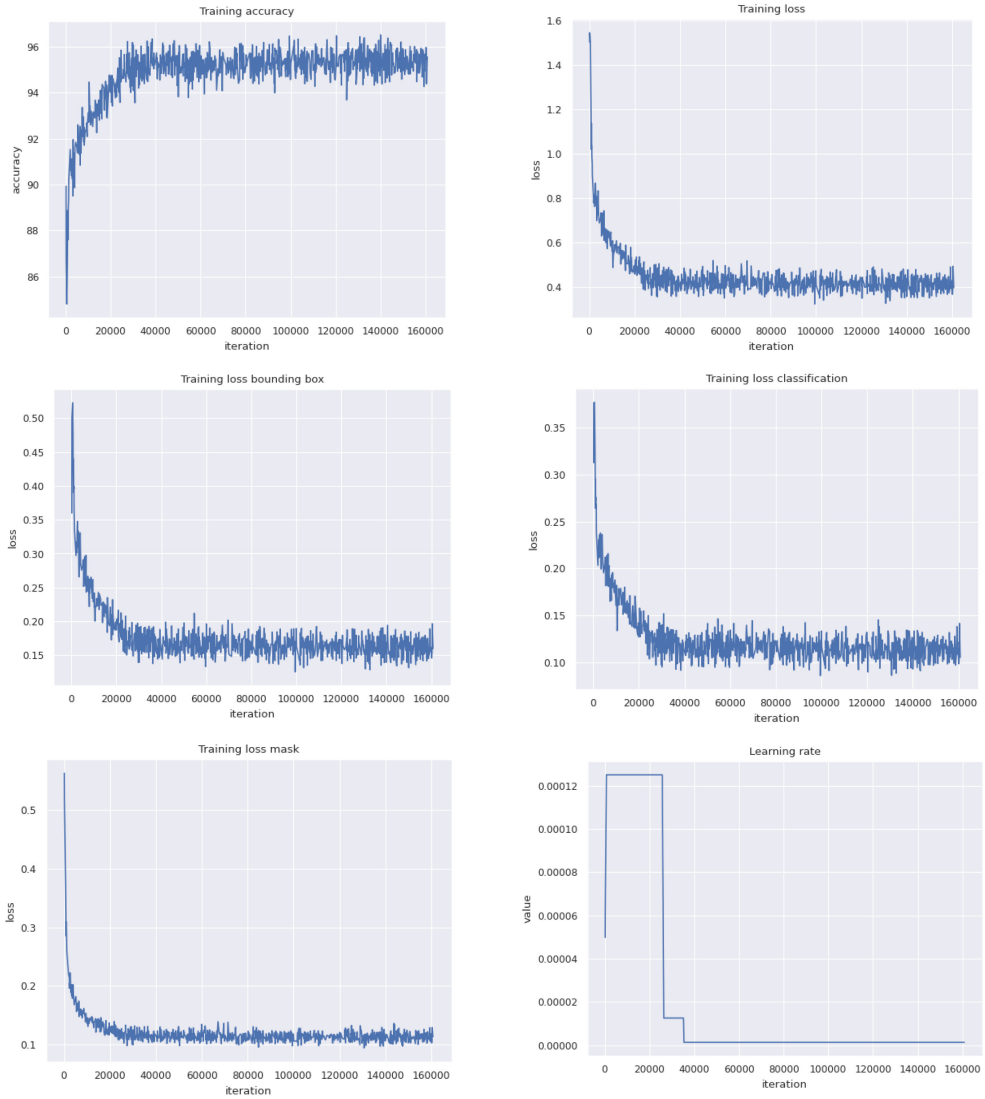



FIGURE A3
Training curves of the model.

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 13 September 2023
DOI 10.3389/fpls.2023.1246065

Check for updates

OPEN ACCESS

EDITED BY
Tonghai Liu,
Tianjin Agricultural University, China

REVIEWED BY
Aibin Chen,
Central South University Forestry and Technology, China
Jun Liu,
Weifang University of Science and Technology, China

*CORRESPONDENCE
Song Yuepeng
uptonsong@sdau.edu.cn
Ren Longlong
renlonglong@sdau.edu.cn

RECEIVED 24 June 2023
ACCEPTED 14 August 2023
PUBLISHED 13 September 2023

CITATION
Ang G, Han R, Yuepeng S, Longlong R, Yue Z and Xiang H (2023) Construction and verification of machine vision algorithm model based on apple leaf disease images. *Front. Plant Sci.* 14:1246065. doi: 10.3389/fpls.2023.1246065

COPYRIGHT
© 2023 Ang, Han, Yuepeng, Longlong, Yue and Xiang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# Construction and verification of machine vision algorithm model based on apple leaf disease images

Gao Ang[1], Ren Han[1], Song Yuepeng[1,2,3]*, Ren Longlong[1,2,3]*, Zhang Yue[1] and Han Xiang[1]

[1]College of Mechanical and Electronic Engineering, Shandong Agricultural University, Tai'an, Shandong, China, [2]Key Laboratory of Horticultural Machinery and Equipment of Shandong Province, Shandong Agricultural University, Tai'an, Shandong, China, [3]Intelligent Engineering Laboratory of Agricultural Equipment of Shandong Province, Shandong Agricultural University, Tai'an, Shandong, China

Apple leaf diseases without timely control will affect fruit quality and yield, intelligent detection of apple leaf diseases was especially important. So this paper mainly focuses on apple leaf disease detection problem, proposes a machine vision algorithm model for fast apple leaf disease detection called LALNet (High-speed apple leaf network). First, an efficient sacked module for apple leaf detection, known as EALD (efficient apple leaf detection stacking module), was designed by utilizing the multi-branch structure and depth-separable modules. In the backbone network of LALNet, (High-speed apple leaf network) four layers of EALD modules were superimposed and an SE (Squeeze-and-Excitation) module was added in the last layer of the model to improve the attention of the model to important features. A structural reparameterization technique was used to combine the outputs of two layers of deeply separable convolutions in branch during the inference phase to improve the model's operational speed. The results show that in the test set, the detection accuracy of the model was 96.07%. The total precision was 95.79%, the total recall was 96.05%, the total F1 was 96.06%, the model size was 6.61 MB, and the detection speed of a single image was 6.68 ms. Therefore, the model ensures both high detection accuracy and fast execution speed, making it suitable for deployment on embedded devices. It supports precision spraying for the prevention and control of apple leaf disease.

KEYWORDS

**apple leaf disease, deep learning, deep separable convolution, re-parameterization, leaf detection network**

# 1 Introduction

There are approximately more than 80 countries worldwide engaged in large-scale apple production, and as the area under apple production continues to expand (Zhang, 2021), the incidence of pests and diseases affecting apples has become increasingly severe. Apple leaf diseases, if left untreated, would pose a serious threat to the growth, development and quality of apples. Currently, traditional methods of diagnosing apple leaf diseases rely heavily on human judgment, requiring experienced and highly skilled field workers. Errors in worker judgment can lead to delayed prevention or excessive control measures, both of which can be detrimental. Therefore, efficient and rapid assessment of apple leaf diseases plays a critical role in improving apple quality and increasing grower profitability.

With the development of computer vision and artificial intelligence, deep learning has received increasing attention in the field of image processing (Shun et al., 2019; Zhang and Lu, 2021), while deep learning techniques have a wide range of ap plications in agriculture (Kamilaris and Prenafeta-Boldú, 2018; Zheng et al., 2019; Sharma et al., 2020; Arumugam et al., 2022). In the research of plant leaf disease classification, Aditya Karleka et al. designed a deep learning convolutional neural network Soybean leaf diseases classification (SoyNet) by increasing the diversity of pooling operations, adding Relu functions and dropout operations rationally for identifying and classifying soybean plant The proposed model achieved 98.14% recognition accuracy with good precision, recall and F1 score (Guo et al., 2022). Paul Shekonya Kanda et al. proposed an intelligent method based on deep learning to identify nine common tomato diseases. The method employed a residual neural network algorithm to identify tomato diseases and used five network depths to measure the accuracy of the network. According to the experimental result, this method obtained the highest F1 score of 99.5%, outperforming most previous competing methods in tomato leaf disease identification (Zhang et al., 2021). Laixiang Xu et al. proposed a new deep learning model for peanut leaf disease recognition. This proposed model was a combination of an improved X-ception, a partially activated feature fusion module and two attention enhancement branches. The model obtained 99.69% accuracy in the test set, which is 9.67% - 23.34% higher than Inception-V4, ResNet 34 and MobileNet-V3, demonstrating the feasibility of the model (Gill and Khehra, 2022). It shows that by designing specific network parameter settings in convolutional neural networks for plant disease classification, adding residual structure, adding attention mechanism, and other operations were capable of achieving higher accuracy.

In the study of apple leaf disease classification convolutional neural network model numerous scholars have done a lot of researches on improving the accuracy of apple tree leaf disease classification recognition, reducing the parameters and training time of specific recognition networks. For example, Yong et al. proposed a DenseNet-121 deep convolutional network based on three methods of regression, multi-label classification and focal loss function to identify apple leaf diseases. The proposed method achieved 93.51%, 93.31%, and 93.71% accuracy on the test set, respectively, outperforming the traditional cross-entropy loss function-based multi-classification method with 92.29% accuracy (Zhong and Zhao, 2020). Lili et al. proposed a convolutional neural network based on the AlexNet model for the classification of five diseases of apple tree leaves, which uses dilated convolution to extract coarse-grained features of diseases in the model, which helps to reduce the number of parameters while maintaining a large field of perception, and adds parallel convolutional modules to extract leaf disease features at multiple scales. Subsequently, a series of $3 \times 3$ convolutional shortcut connections allowed the model to handle additional nonlinearities. The final recognition accuracy of the model was 97.36% and the model size was 5.87 MB (Li et al., 2022). Qian et al. proposed an improved model based on VGG16 to identify apple leaf diseases, in which a global average polarization layer was used instead of a fully connected layer to reduce parameters and a batch normalization layer was added to improve convergence speed. A migration learning strategy is used to avoid long training time. The experimental results show that the overall accuracy of apple leaf classification based on the proposed model could reach 99.01%. Compared with the classical VGG16, the model parameters are reduced with 89%, the recognition accuracy is improved with 6.3%, and the training time is reduced to 0.56% of the original model (Yan et al., 2020).

In apple leaf disease classification and recognition research, scholars have achieved high recognition accuracy using deep learning techniques, however, how to ensure apple leaf disease recognition accuracy while making the model run faster is still the focus of research. Therefore, this paper proposes the LALNet model, in the next section in-depth discussion of the research content of this paper, in the second section, mainly introduces the data set of this paper, the main components of the LALNet network using the multi-branching structure and the depth separable module to design the efficient leaf detection EALD module, in the LALNet in the use of the EALD module stacking and add SE attention module, Finally, in the inference stage using structural re-parameterization technique to improve the running speed of the model. In Section III, the model was trained, validated and tested using publicly available apple leaf disease datasets, and a comparative analysis of this paper's model with state-of-the-art apple leaf classification models was performed to provide a comprehensive evaluation of the model to ensure its reliability. In Sec. IV, the research work of this paper was fully summarized and the limitations of this research and future research directions were discussed. Thus, the proposed LALNet model improves the speed of image recognition while ensuring recognition accuracy, and finally, this research can support intelligent apple leaf spray control.

# 2 Tests and methods

## 2.1 Apple leaf data set

In this study, apple leaf disease images were collected at the apple experimental field of Shandong Agricultural University

(117.12°E,36.20°N) and at the Tianping Lake experimental demonstration base of Shandong Fruit Tree Research Institute, National Apple Engineering Technology Research Center (117.01° E,36.21°N), which were collected several times in July 2022 under favorable weather conditions.

Meanwhile, Baidu public dataset of apple leaf pathology images (Ai Studio poublic datasets, 2023) was used to expand the dataset of this paper. This dataset contains five types of common apple leaf diseases, namely apple mosaic, rust, gray spot, alternaria leaf spot and brown spot. For the convenience of training management, apple mosaic, rust, gray spot, alternaria leaf spot, and brown spot were represented by the numbers 0, 1, 2, 3, and 4, respectively, and some apple leaf disease images are shown in Figure 1. After flipping, panning and contrast enhancement to pre-process the data set of this paper, a total of 25,000 disease images with image size of 224*224 were obtained. In order to use this dataset for training, validation and testing, the data is divided as shown in Table 1, 80% of the images were used for model training, 10% of the images were used for model validation and 10% of the images were used for model testing.

## 2.2 LALNet network model

The LALNet lightweight apple leaf disease identification network model was mainly constructed by referring to the typical ResNet network model and MobileNet network model structure, using depth separable modules in the network and lightweight attention modules to lighten the parameters of the network model, and using structural reparameterization in inference to improve the inference speed of the network model. The flowchart of the LALNet network was shown in Figure 2, which modeled the main components of the ELAD module and the SE Attention Mechanism module.

### 2.2.1 ResNet network

In recent years, Convolutional neural nerve network (CNN) has been continuously evolving and growing, representing one of the prominent architectures in deep learning networks (Kawakura and Shibasaki, 2020; Deepan and Sudha, 2021). However, as the network depth increases, it becomes increasingly difficult to train, leading to the problem of network degradation. To address this problem, in 2015, a research team from Microsoft Research proposed ResNet (Residual Network) (He et al., 2016), a deep learning network that introduced residual connections. These connections made it easier to train deeper networks.

The network structure of ResNet was shown in Figure 3. The network mainly consists of an input layer, convolutional layers, residual modules, pooling layers, and fully connected layers. Input layer: input image data; Convolutional layer: extracts features of the image; Residual block: consists of two or more convolutional layers with residual connections; Pooling layer: reduces the dimensionality of the image; Fully connected layer: connects the outputs of all convolutional layers (Shifang et al., 2021). The network structure of ResNet consists of two main components: the residual blocks and the backbone network (Chunshan et al., 2020). Each residual block contains two or more convolutional layers and a residual connection, whose main function was to pass the residuals of the input data directly to the next residual block, which increases the mobility of the data so that the gradient can remain valid in deeper layers of the network and thus reduce the effect of gradient disappearance. ResNet constructs a deeper network by stacking more and more residual blocks to solve more complex problems.

### 2.2.2 MobileNet network model

Lightweight network design differs from traditional neural networks by placing greater emphasis on compactness of the model structure for running networks on embedded devices. Google proposed MobileNet V1, a classical lightweight network that can be deployed on mobile (Wenjie et al., 2021), which uses deep separable convolution instead of traditional convolution to reduce the network parameters while ensuring network accuracy (Howard et al., 2019). MobileNet V2 further improves the performance of the model by adding inverse residual structure and linear units and using nonlinear activation functions in high-

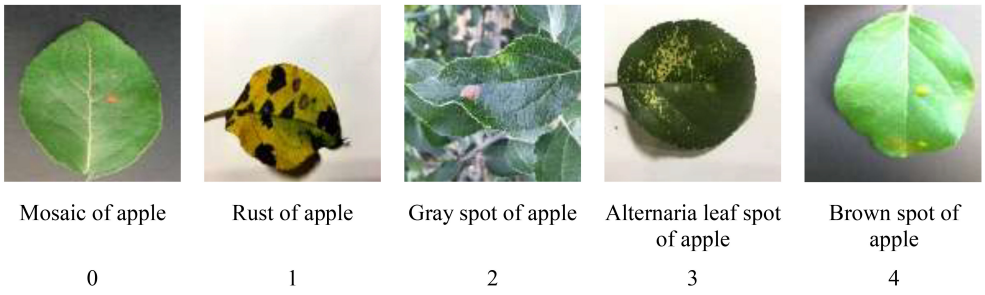



FIGURE 1
Pictures of some fruit leaf diseases.

TABLE 1 Classification of apple leaf data set.

| Type | Disease category | Training set/sheet | Validation set/sheet | Test set/sheet |
|---|---|---|---|---|
| 0 | Mosaic of apple | 4275 | 534 | 534 |
| 1 | Rust of apple | 4525 | 565 | 565 |
| 2 | Gray spot of apple | 3849 | 481 | 480 |
| 3 | Alternaria leaf spot of apple | 3901 | 487 | 487 |
| 4 | Brown spot of apple | 4556 | 569 | 569 |

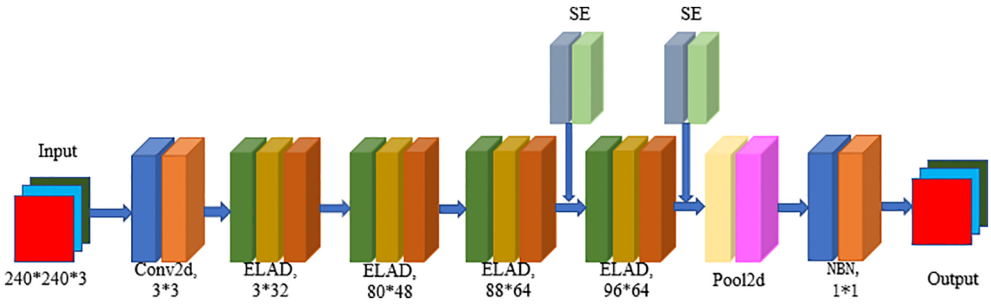



FIGURE 2
Flowchart of LALNet network structure.

dimensional space based on V1. MobileNet V3 (Hu et al., 2020), based on V2, introduces lightweight attention (squeeze and excitation) (Zhou et al., 2022) modules that effectively suppress unnecessary channels, while the model uses the h-swish activation function to reduce the computational cost of applying nonlinear activation functions and achieve better parameter reduction.

Deeply separable convolution (DSC) holds the key to lightweight network design, as shown in Figure 4. This convolution is a decomposable convolutional structure that decomposes standard convolution into deep wise convolution, which is the process of combining features to create feature vectors of new dimensions, and Pointwise convolution, which is the process of filtering the input feature vectors. Compared to traditional convolution, deep separable convolution can reduce the parameters of the model to improve the detection speed. For example, the input feature map size for H×W, the number of input channels for M, the convolution kernel size for K×K, the number of output channels for N, and the output feature map size for OT×OT. The normal convolution computes Nc is.

$$N_c = K \times K \times M \times N \times O_T \times O_T \tag{1}$$

The deeply separable convolution computation Na is.

$$N_a = O_T \times O_T \times K \times K \times M + O_T \times O_T \times M \times N \tag{2}$$

The ratio of computational cost between depth wise separable convolution and regular convolution is.

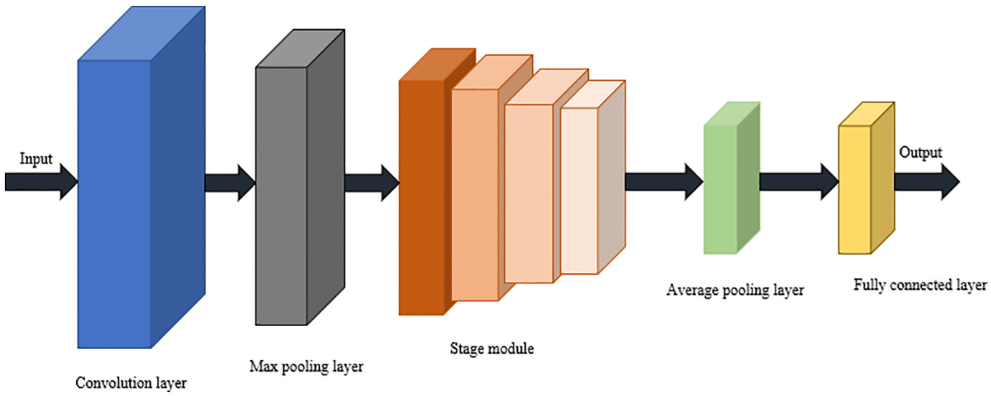



FIGURE 3
ResNet network structure diagram.

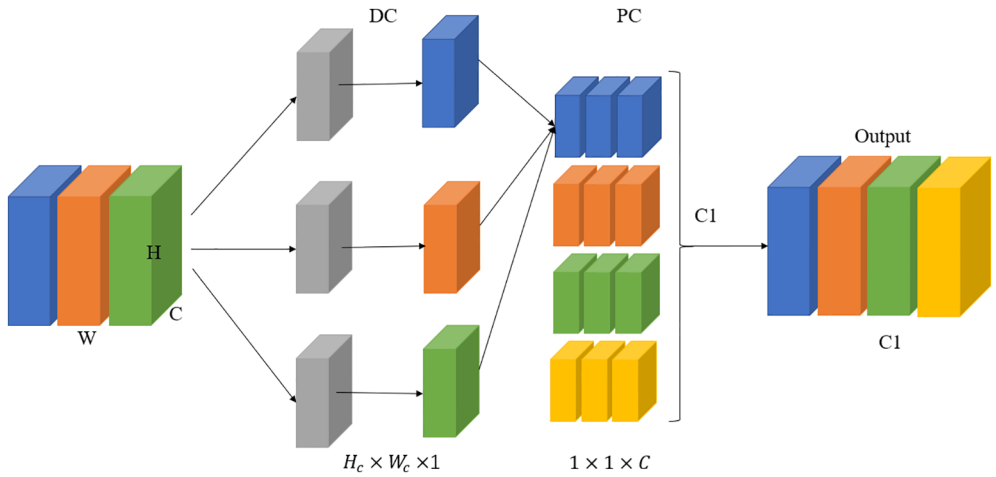



FIGURE 4
Structure of deep separable convolutional network.

$$\frac{N_a}{N_c} = \frac{1}{N} + \frac{1}{k^2} \tag{3}$$

From the ratio of deeply separable convolutional to normal convolutional computation, it is shown that the reduction of deeply separable convolutional computation is related to the number of channels and the size of the convolutional kernel, with the larger the size of the convolutional kernel, the larger the computational reduction.

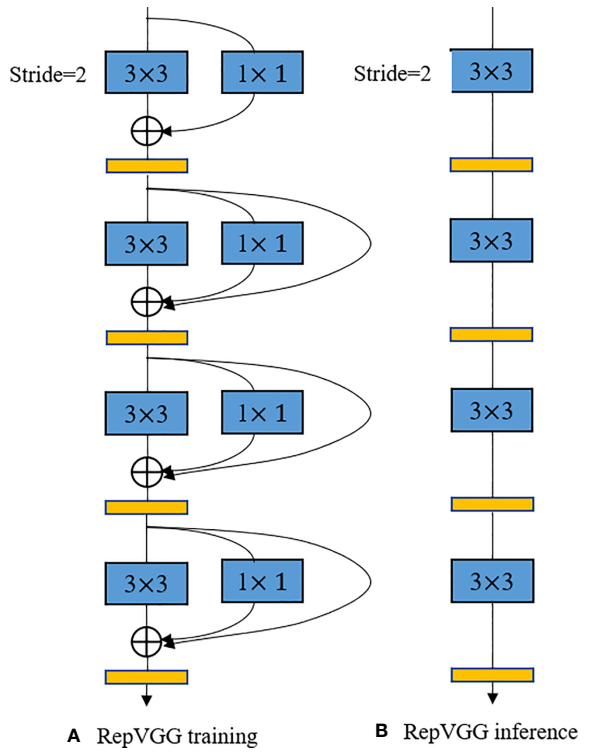



FIGURE 5
RepVGG model training and inference structure diagram.

### 2.2.3 Structural reparameterization

The structural re-parameterization is a technique for optimizing neural network models (Ding et al., 2021). This technique enables efficient training and deployment of deep learning models in scenarios with limited computational resources by using constant parameter transformations to reduce the storage and computational resources of the model through simplification of the network structure. As shown in Figure 5, the earlier RepVGG model uses a simple architecture consisting of stacked 3*3 Conv and ReLU to achieve structural decoupling during training and inference, and uses a multi-branch structure during training, and then uses reparameterization to equivalently transform the multi-branch architecture to a VGG single-way architecture with stacked 3*3 convolutional layers after training was completed, using this structured reparameterization method to enable RepVGG to achieve ImageNet to achieve more than 80% accuracy and run several times faster (Transactions of the Chinese Society of Agricultural Engineering et al., 2021; Hu et al., 2022).

### 2.2.4 LALNet model construction

Inspired by the depth-wise separable convolutions in ResNet and MobileNet, this paper proposes an efficient EALD module. The EALD module, as shown in Figure 6, uses a multi-branch structure and depth-wise separable modules to extract more feature information with fewer parameters and computational complexity. First, the module uses a standard 1x1 convolution kernel for dimensionality reduction, followed by different branches for feature extraction. The first and second branches use 3x3 depth separable modules to extract complex features. In the third branch, a 1x1 standard convolution is used to extract residual information and to enhance the interplay of module features. Then, the outputs of the three branches are summed and the

channel number is restored using a 1x1 pointwise convolution. Finally, channel shuffling was performed to facilitate information fusion between channels, thereby improving the feature recognition capability.

The LALNet lightweight apple leaf disease classification model was stacked using the EALD module, and the network structure of the LALNet model follows in Table 2. First, the initialized feature extraction of three channels of the image was performed in step one using a standard convolution with a convolution kernel of 3*3, which has a step size of two and an output channel number of 16. The EALD module was used for feature extraction in steps two-five with a step size of 1. The SE attention module was added in steps four and five to increase the feature extraction capability. In step six, an adaptive averaging pooling layer was used and then a linear layer with 960 input features and 1280 output features was passed. In step seven, the output of the linear layer was passed through another batch normalization layer so that a linear layer with 1280 input features and number of output classes was applied as the final layer.

While a multi-branch structure reduces the number of parameters in a model, many researchers argue that having too many branches can affect the model's runtime speed during

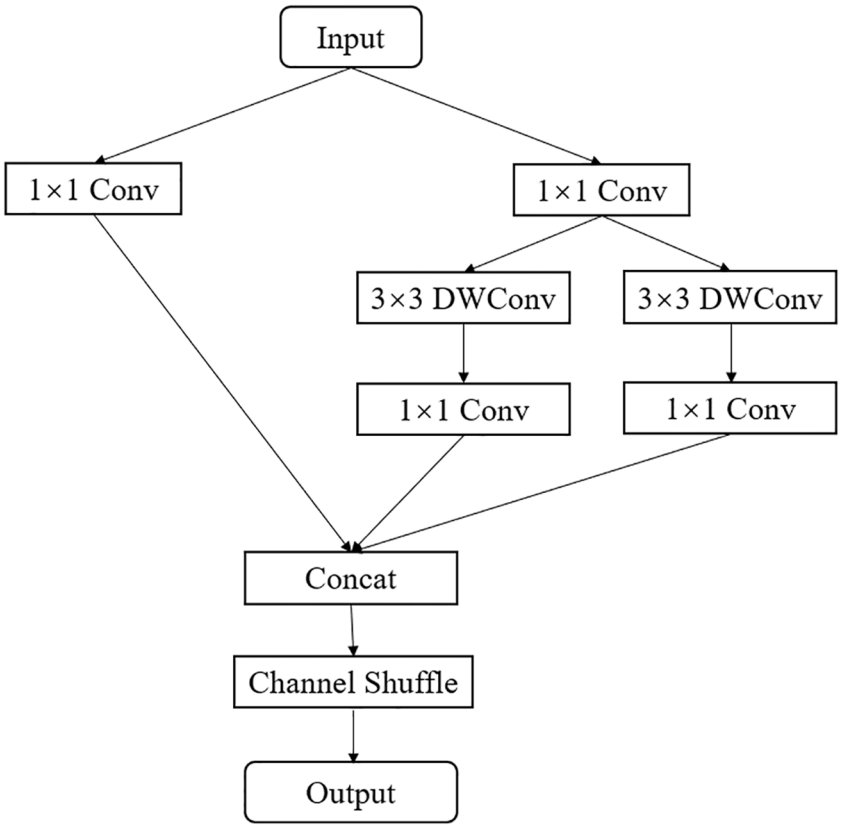



FIGURE 6
EALD module structure diagram.

TABLE 2 network structure of LALNet model.

| Step | Input | Operator | Stride | Output | Attention mechanism |
|---|---|---|---|---|---|
| 1 | 3 | Conv2d, 3×3 | 2 | 16 | – |
| 2 | 3 | EALD,(3,32) | 1 | 80 | – |
| 3 | 80 | EALD, (80, 48) | 1 | 88 | – |
| 4 | 88 | EALD, (88, 64) | 1 | 96 | 1 |
| 5 | 96 | EALD,(96, 64) | 1 | 96 | 1 |
| 6 | 960 | Pool2d | – | – | – |
| 7 | 1280 | 1×1, NBN | | 1280 | – |

inference. Therefore, this paper optimizes the structure of the model during recognition using a re-parameterization strategy. As shown in Figure 7, the convolutional layers with 3x3 depth-wise separable convolutions and their respective batch normalization (BN) layers in the first and second branches are fused. After fusion, a set of 3x3 depth-wise separable convolutional groups is used to represent the common parameters of the two branches, thereby improving the model's recognition speed during inference.

# 3 Results and discussion

## 3.1 Experiment environment

In this study, the hardware experimental environment consisted of a Lenovo laptop (y9000p) with an Intel Pentium i5-12700H processor running at a frequency of 3.5GHz, and a GeForce GTX 3060 6G GPU. The software experimental environment involved a Windows 10 operating system, Python 3.8 as the programming language, PyTorch 1.10.0 as the machine learning library, and CUDA 10.2 as the parallel computing framework.

## 3.2 Evaluative metrics

The following metrics are commonly used when evaluating the performance of classification models:

Accuracy: This is a measure of the overall accuracy of the model's predictions. It indicates the percentage of correct predictions made by the model across all samples.

$$A = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \times 100\,\% \tag{4}$$

Precision: This is a measure of the proportion of actual positive samples for which the model predicts a positive outcome.

$$P = \frac{T_P}{T_P + F_P} \times 100\,\% \tag{5}$$

Recall: This is a measure of the proportion of actual positive samples for which the model is predicted to be positive.

$$R = \frac{T_P}{T_P + F_N} \times 100\,\% \tag{6}$$

F1 value: This is a combined precision and recall metric that measures the overall predictive effectiveness of the model for positive samples.

$$F1 = 2 \times \frac{P \times R}{P + R} \times 100\,\% \tag{7}$$

where: A- Accuracy; P-precision; R-recall rate;

$T_P$-True positive, the number of samples correctly predicted as positive;

$T_N$-True negative, the number of negative samples predicted as negative;

$F_P$-False positive, the number of negative samples predicted as positive;

$F_N$-False negative, the number of positive samples predicted as negative.

## 3.3 Model training, testing parameters

In training and testing the LALNet model, the parameters of the training and testing models were finally selected after several tests and trials to suit the data set and computer performance of this paper as shown in Table 3 below, the image size of the training and testing models was 224*224, the Batch Size was 16 during training, the Batch Size was 16 during testing, the loss function was Cross entropy loss, the optimization function was Adam, the learning rate was 0.001, and the number of training rounds was 100.

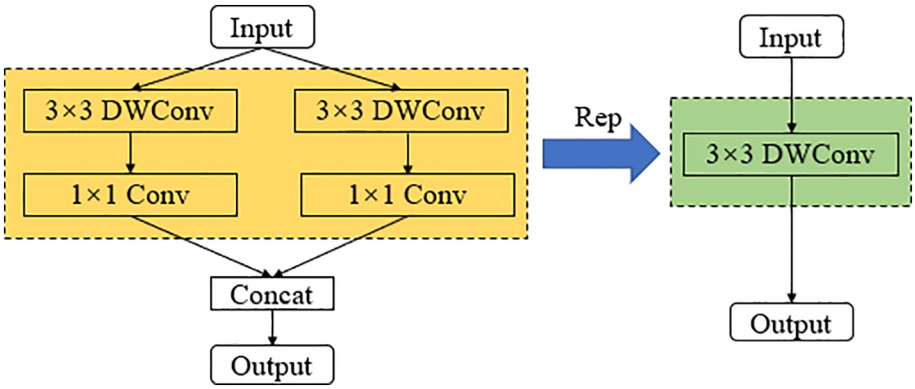



FIGURE 7
Schematic diagram of multi-branch fusion using structural re-parameters.

TABLE 3 LALNet model training and testing parameters.

| Parameter | Training | Testing |
|---|---|---|
| Image size | 224 × 224 | 224 × 224 |
| Batch Size | 16 | 32 |
| Loss function | Cross entropy loss | – |
| Optimization function | Adam | – |
| Learning Rate | 0.001 | – |
| Training epoch | 100 | – |

## 3.4 Accuracy of the model training

In the training process of the model, it is common to evaluate the loss and accuracy of the training and validation sets to assess the performance of the model on the dataset (Pang et al., 2020). In this paper, both the training and validation of the model were performed using the same parameters and number of training epochs. The loss and accuracy of the model on the training and validation sets were monitored. Figure 8A shows the loss graph of the training and validation of the model, while Figure 8B shows the accuracy graph of the training and validation of the model in this paper. From the figures, it can be seen that the model initially had lower accuracy and higher loss values with significant fluctuations. However, as the training progressed, both the accuracy and loss values of the model stabilized. Therefore, the model did not experience overfitting or underfitting problems, indicating a good training performance of the model.

## 3.5 Analysis of structural re-parameterization results

In this study, structural re-parameterization was applied to the EALD module during the model inference phase, with the goal of improving the model's runtime speed during inference. The recognition accuracy and single frame recognition speed of the model with and without structural reparameterization were evaluated on the test set (Yueming et al., 2023). The experimental results, as shown in Table 4, indicate that the parameter size of the model remained almost unchanged after reparameterization. Although there was a slight decrease of 1% in detection accuracy, the model's detection speed improved by 19.03%. Therefore, this re-parameterization method demonstrates its effectiveness in improving the model running speed while maintaining the model performance.

## 3.6 Analysis of model parameters, efficiency

The confusion matrix is a common tool for evaluating the performance of classifiers, which assesses the performance of the model by tracking the relationship between the actual and predicted

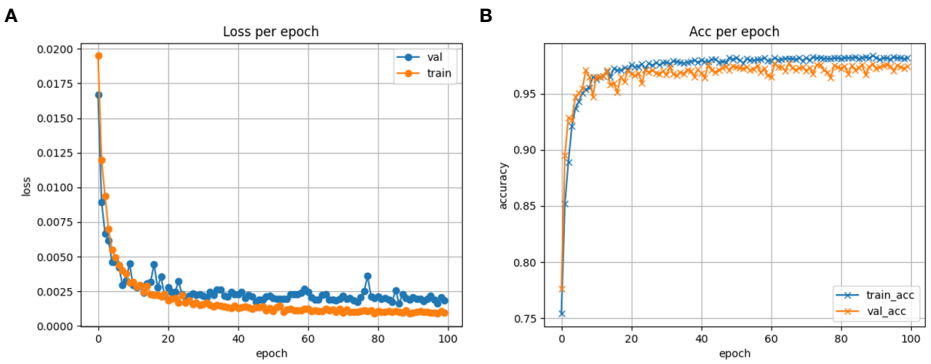



FIGURE 8
Model training monitoring graph. **(A)** Loss plot of model training. **(B)** Accuracy plot of model training.

TABLE 4 Recognition results before and after using structural heavy parameters.

| Model names | Structure heavy parameters | Total accuracy of test set/% | Total precision of test set/% | Recall rate of test set/ % | Test set F1 value/ % | Model Parameters/ MB | Single picture detection speed/ ms |
|---|---|---|---|---|---|---|---|
| LALNet LALNet | Yes | 95.95 | 95.94 | 98.52 | 96.01 | 6.60 | 8.25 |
| | No | 96.07 | 95.98 | 96.02 | 96.06 | 6.61 | 6.68 |

labels of the classifier (Simonyan and Zisserman, 2014; Görtler et al., 2022). The confusion matrix of the LALNet model with MobileNet V3-small model and ShuffleNet V2 model on the test set was shown in Figure 9. From the confusion matrix Figure 9A, it can be seen that the label 0 correctly predicted images of 503, label 1's correctly predicted images of 556, label 2's correctly predicted images of 445, label 3's correctly predicted images of 482, and label 4's correctly predicted images of 545. By comparing Figures 9A, B, it was found that the correctly predicted images of label 4 in the MobileNet V3-small model exceeded the LALNet model, and the rest of the labels were slightly lower than the LALNet model. From the comparison of Figures 9A, C, it was found that the correct predicted images of label 2 in the ShuffleNet V2 model exceeded the LALNet model, and the rest of the labels were slightly lower than the LALNet model. By comparing the three confusion matrices, it was observed that each model recognized different types and numbers of confused labels, which indicated that different models had different recognition of apple leaf diseases. It was also found that the LALNet model integrated the correct label matching slightly better than the other two models, thus indicating the superior design of the LALNet model.

This paper conducts a comparative test on whether the lalnet model uses the attention mechanism. The results are shown in Table 5. It can be seen from Table 5 that when the attention module is not added, the total accuracy of the LALNet model was 95.46%, the total precision was 95.79%, the total F1 was 95.43%, and the single picture detection speed is 6.68ms. When the attention module was added, the total accuracy of lalnet model was 96.07%, and the total F1 was 96.06%. The accuracy and F1 values were improved. At the same time, the single image detection speed was also slightly reduced to 7.58ms. Thus, the attention mechanism can improve the performance of the model to a certain extent, making it more accurate and robust, but it will also affect the detection speed of the model.

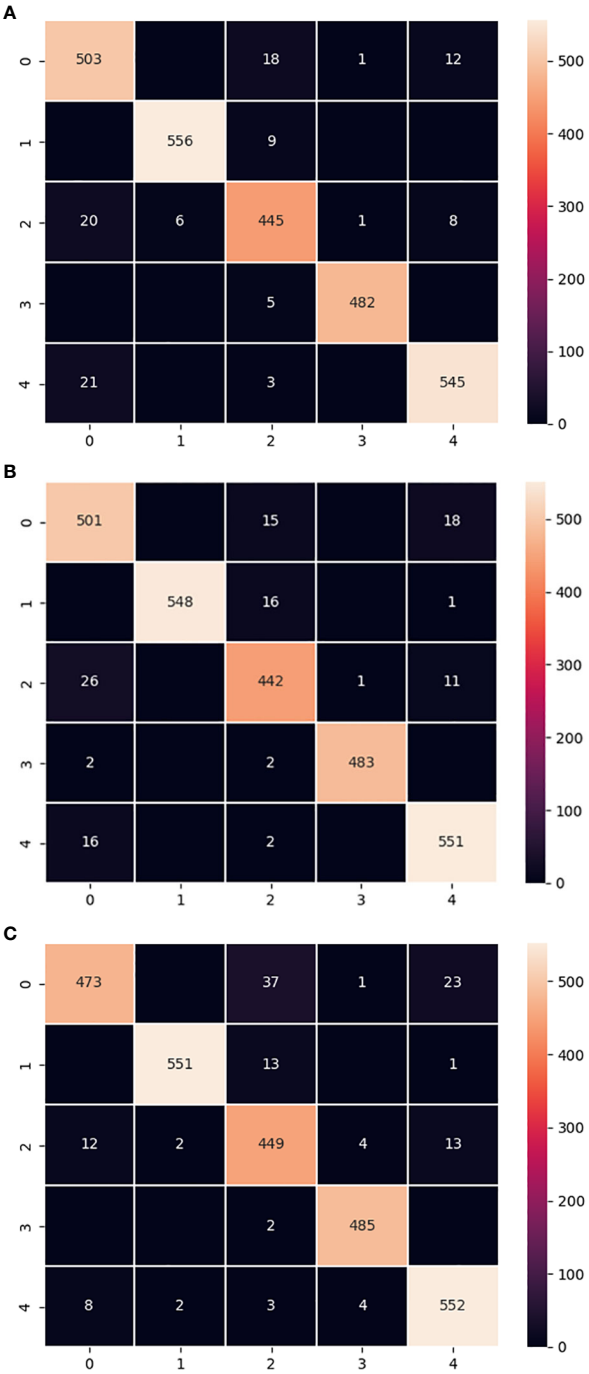



FIGURE 9
Confusion matrix for the 3 models. **(A)** LALNet model confusion matrix. **(B)** MobileNet V3-small model confusion matrix. **(C)** ShuffleNet V2 model confusion matrix.

## 3.7 Comparative analysis of different models in the experimental study

To further validate the performance of the model in classifying different types of apple leaf categories, the model was evaluated using six different network models: LALNet, VGG16 (Ma et al., 2018), ResNet34, MobileNet V2, MobileNet V3-small, and ShuffleNet V2 (Zhong and Zhao, 2020). The experimental results on the test set are shown in Table 6, while the performance metrics for different leaf diseases are shown in Figure 10.

From the data results in Table 6, it could be seen that the LALNet model had an overall accuracy of 96.07% on the test set, which was higher than the other six models. In addition, the total precision, the total recall and total F1 values of the LALNet model were 95.98%,96.05% and 96.06%, respectively, which were also better than the other six models. In terms of detection speed, the single image detection speed of the LALNet model was 6.68ms faster than that of the other six classical models, and the single image detection speed was 16.79% higher than that of the lightweight MobileNet V3-small model, which means that the LALNet model has better real-time detection performance in practical applications.

In the performance evaluation of the different classical models compared, VGG 16 has the lowest total test set accuracy of 94.91%,

10.3389/fpls.2023.1246065

TABLE 5 Recognition results before and after using attention mechanism.

| Name of the model | Category Tags | Attention Module | Test set accuracy/ % | Test set precision/ % | Test set F1/% | Single picture detection speed/ms |
|---|---|---|---|---|---|---|
| LALNet | 0 | No | 94.19 | 94.21 | 93.76 | 7.58 ms |
| | 1 | | 97.52 | 97.50 | 98.31 | |
| | 2 | | 89.58 | 89.58 | 92.18 | |
| | 3 | | 98.36 | 98.38 | 98.97 | |
| | 4 | | 97.01 | 97.11 | 93.96 | |
| LALNet | 0 | Yes | 94.19 | 94.22 | 93.32 | 6.68 ms |
| | 1 | | 98.41 | 98.38 | 98.67 | |
| | 2 | | 92.71 | 92.74 | 92.71 | |
| | 3 | | 98.97 | 98.96 | 99.28 | |
| | 4 | | 98.78 | 98.76 | 96.12 | |

TABLE 6 Recognition results before and after using attention mechanism.

| Name of the model | Test set accuracy/ % | Test set precision/ % | Test set recall rate/ % | Test set F1/ % | Single picture detection speed/ ms |
|---|---|---|---|---|---|
| LALNet | 96.07 | 95.98 | 96.05 | 96.06 | 6.68 |
| VGG 16 | 94.91 | 94.94 | 94.61 | 94.64 | 10.20 |
| ResNet34 | 95.03 | 95.00 | 94.42 | 94.45 | 8.46 |
| MobileNet V2 | 94.93 | 94.91 | 94.51 | 94.64 | 9.68 |
| MobileNet V3-small | 95.82 | 95.83 | 95.87 | 95.80 | 8.05 |
| ShuffleNet V2 | 95.25 | 95.28 | 95.26 | 95.20 | 10.23 |

the lowest total test set F1 value of 94.64%, and the second lowest total test set recall of 94.61%. MobileNet V3-large has a test set total accuracy of 95.93% and a test set total recall of 95.87%. Comparison of the models reveals that the detection accuracy of the lightweight model MobileNet V3-small exceeds the detection accuracy of ResNet34 and VGG16 models while ensuring the detection speed, which indicates that the lightweight structure design is superior in terms of model architecture, but still lacking compared to LALNet. In comparison, it was found that the LALNet model is faster in single image detection while ensuring detection accuracy, so it is more advantageous in apple leaf disease detection application scenarios that require fast response.

As shown in Figure 10, the accuracy, recall and F1 values of different models varied for different leaf disease categories. The LALNet model performed consistently in terms of accuracy, fluctuating around 95% for different disease categories, with the LALNet model achieving the best accuracy for category 1 and category 4, 98.41% and 98.78%, respectively. All models performed the worst accuracy on category 2, with the VGG 16 model having a lower accuracy of only 84.92% on category 2. In terms of recall, LALNet models performed best in category 1 and category 4 on the four disease categories, while the greatest variability in performance was found among the six models in category 2, where the VGG 16 model had about 85% recall on category 2 and ShuffleNet V2 about 94% on category 2. In terms of F1 values, LALNet models had the best F1 values in categories 1 and 4, while all models had F1 values above 95% in categories 0 and 2. The comparison showed that the LALNet models performed consistently in terms of accuracy, recall and F1 values, which achieved better performance for each disease category. Compared to other models, LALNet shows superior recognition accuracy in most disease categories, further validating the reliability and effectiveness of LALNet as an excellent model for apple leaf disease recognition.

To further analyze the performance of this paper's model in apple leaf disease detection, the LALNet model was compared and analyzed with existing state-of-the-art apple leaf disease detection

10.3389/fpls.2023.1246065

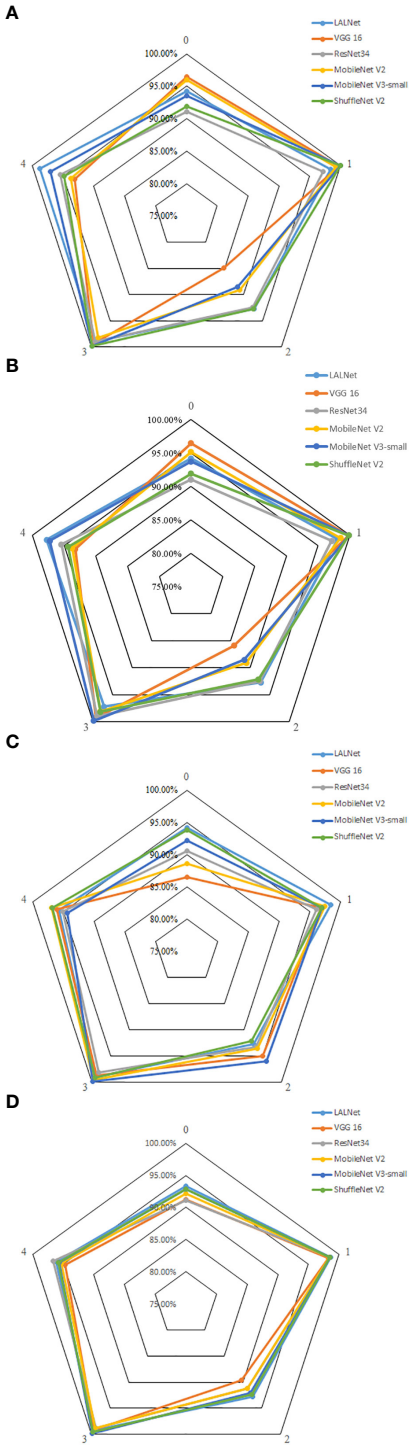



FIGURE 10
Comparison of different apple leaf disease evaluation indexes of the 6 models. **(A)** Accuracy, **(B)** Precision, **(C)** Recall rate, **(D)** F1 values.

methods, as shown in Table 7. It can be seen that the LALNet model achieves 96.07% accuracy on the self-built dataset and the Baidu AI dataset, and this paper's model shows good performance in the disease detection task compared with other methods. In the comparison, it can be found that the detection accuracy of this paper's model is close to or even exceeds some advanced research results, and it also can be found that the overall detection accuracy of seven models exceeds 90%, and Yinping Chen et al. achieved 97.78% on the PlantVillage dataset.However, we should also pay attention to the limitations of different methods due to the experimental environments in which the hardware devices different and the datasets used are also very different, which will affect the test results, especially the detection speed of the model.

## 4 Conclusion and limitations

In this paper, it proposed a fast apple leaf disease detection model LALNet. Firstly, an efficient leaf detection stacking EALD module was designed using multi-branch structure and depth separable modules, which can obtain more accurate identification information with less parameters and computation. Further, the EALD module was used in the LALNet model to stack four layers and add the SE module in the last layer of the model to improve the attention of the network model to focus on important features. Finally, the structural reparameterization technique was used to combine the outputs of two layers of deeply separable convolutions in the branch to improve the speed of the model during the inference phase. The proposed fast apple leaf disease detection model has an overall accuracy of 96.07% in the test set, precision of 95.98%, and F1 score of 96.06%, a model size of 6.61 MB, and a detection speed of 6.68 ms for a single image, thus the model meets the detection accuracy while ensuring its operation speed and is suitable for use on embedded devices.

However, it is important to acknowledge the limitations of this study in order to provide readers with a comprehensive assessment. First, the dataset used in this research has limitations in terms of data collection methods, sample size, and range of disease types covered, which may affect the generalizability of the model. Second, the performance of the model in real-world applications may be affected by factors such as lighting variations, different capture angles, and variations in leaf quality, which may affect its detection performance. Finally, while the focus of this study was on common apple leaf diseases, it does not cover all possible disease types that may be present in practical cultivation. Future research should consider collecting more diverse and comprehensive datasets and further optimizing the model to improve its accuracy and robustness.

TABLE 7 Comparison results of advanced apple leaf classification models.

| Reference | Data sets | Disease type | Model | Accuracy (%) |
|---|---|---|---|---|
| Yong Zhong et al. (Zhong and Zhao, 2020) | AI challenger plant disease recognition | Healthy apple, General apple scab, Serious apple scab, Apple gray spot, General cedar apple rust, Serious cedar apple rust | Densenet-121 | 93.71 |
| Yuxi Gao et al. (2023) | Kaggle dataset + Self built dataset | Healthy, Brown spot, Alternaria leaf spot, Mosaic, Powdery mildew, Rust | BAM-Net | 95.64 |
| Xiaofei Chao et al. (2021) | Self built dataset | Alternaria leaf spot, Rust disea, Brown spot disease, Gray spot disease, Mosaic disease, Healthy leaves | SE_miniXception | 97.01 |
| Cong Xu et al. (Cong et al., 2022) | Self built dataset | Mosaic, Brown spot, Rust, Gray spot, Spotted leaf litter | DCCapsNet | 93.16 |
| Yinping Chen et al. (Yiping et al., 2023) | Plant village | Healthy, Scab, Cedar rust | Inproved CycleGAN | 97.78 |
| Chongke Bi et al. (2022) | Self built dataset | Alternaria leaf, Spot, Rust | MobileNet v3 | 73.50 |
| Helong Yu et al. (2022) | Kaggle dataset | Powdery mildew, Alternaria leaf spot, Scab, Rust, Alternaria leaf spot and rust, Healthy | MSOResNet | 95.7 |
| Proposed method et al. | Self built dataset+ Baidu AI dataset | Mosaic, Rust, Gray spot, Alternaria leaf spot, Brown spot | LALNet | 96.07 |

In future work, we aim to further improve the performance of the LALNet model by addressing the aforementioned limitations. In addition, we plan to use the model in an intelligent tracked apple spraying robot to achieve precision spraying and reduce pesticide use.

## Data availability statement

The original contributions presented in the study are included in the article/supplementary material. Further inquiries can be directed to the corresponding authors.

## Author contributions

RH prepared materials. ZY was responsible for the experiment design. GA and RL performed the program development. GA, and HX analyzed the data, and SY were responsible for writing the manuscript. SY and RL contributed to reviewing the manuscript. All authors contributed to the article and approved the submitted version.

## Funding

This work was the Innovation Team Fund for Fruit Industry of Modern Agricultural Technol-ogy System in Shandong Province (SDAIT-06-12) and equipment post and State Key Laboratory of Mechanical System and Vibration (Grant No. MSV202002).

## Acknowledgments

We are very grateful to all the authors for their support and contribution with the manuscript.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

Ai Studio poublic datasets. (2023). "*Pathological images of apple leaves*." Available at: https://aistudio.baidu.com/aistudio/datasetdetail/11591/0.

Arumugam, K., Swathi, Y., Sanchez, D. T., and Okoronkwo, E. (2022). Towards applicability of machine learning techniques in agriculture and energy sector. *Mater. Today.: Proc.* 51, 2260–2263. doi: 10.1016/j.matpr.2021.11.394

Bi, C., Wang, J., Duan, Y., Fu, D., Kang, J., and Shi, Y. (2022). MobileNet based apple leaf diseases identification. *Mobile. Netw. Appl.* 27, 172–180. doi: 10.1007/s11036-020-01640-1

Chao, X., Hu, X., Feng, J., Zhang, Z., Wang, M., and He, D. (2021). Construction of apple leaf diseases identification networks based on xception fused by SE module. *Appl. Sci.* 11 (10), 4614. doi: 10.3390/app11104614

Chunshan, W., Ji, Z., Huarui, W., Guifa, T., Chunjiang, Z., and Jiuxi, L. (2020). Identification of vegetable leaf diseases based on improved Multi-scale ResNet. *Trans. Chin. Soc. Agric. Eng. (Transactions. CSAE).* 36 (20), 209–217. doi: 10.11975/j.issn.1002-6819.2020.20.025

Cong, X., Xuqi, W., and Shanwen, Z. (2022). Dilated convolution capsule network for apple leaf disease identification. *Front. Plant Sci.* 13, 1002312. doi: 10.3389/FPLS.2022.1002312

Deepan, P., and Sudha, L. R. (2021). "Deep learning algorithm and its applications to ioT and computer vision," In K. G. Manoharan, J. A. Nehru and S. Balasubramanian (eds). *Artificial intelligence and ioT. Studies in Big Data* (Singapore: Springer) 85, 223–244. doi: 10.1007/978-981-33-6400-4_11

Ding, X., Zhang, X., Ma, N., Han, J. G., Ding, G. G., and Sun, J. (2021). "Repvgg: Making vgg-style convnets great again," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (Nashville, TN, USA). 2021, 13733–13742. doi: 10.1109/CVPR46437.2021.01352

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in 2016 *IEEE conference on computer vision and pattern recognition (CVPR)*. (Las Vegas, NV, USA). 2016, 770–778. doi: 10.1109/CVPR.2016.90

Gao, Y., Cao, Z., Cai, W., Gong, G., Zhou, G., and Li, L. (2023). Apple leaf disease identification in complex background based on BAM-net. *Agronomy* 13 (5), 1240. doi: 10.3390/agronomy13051240

Gill, H. S., and Khehra, B. S. (2022). Fruit image classification using deep learning. *Res. Sq.* preprint, 06. doi: 10.21203/rs.3.rs-574901/v1

Görtler, J., Hohman, F., Moritz, D., Wongsuphasawat, K., Ren, D. H., and Rahul, N. (2022). "Neo: Generalizing confusion matrix visualization to hierarchical and multi-output labels," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. (New York, USA: Association for Computing Machinery). 408, 1–13. doi: 10.1145/3491102.3501823

Guo, B., Li, B., Huang, Y., Hao, F. Y., Xu, B. L., and Dong, Y. Y. (2022). Bruise detection and classification of strawberries based on thermal images. *Food Bioprocess. Technol.* 15 (5), 1133–1141. doi: 10.1007/s11947-022-02804-5

Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L. C., Tan, M., et al. (2019). "Searching for mobileNetV3," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), pp. 1314–1324. doi: 10.1109/ICCV.2019.00140

Hu, M., Feng, J., Hua, J., Lai, B., Huang, J. Q., Gong, X. J., et al. (2022). "Online convolutional re-parameterization," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA. 2022, 568–577. doi: 10.1109/CVPR52688.2022.00065

Hu, J., Shen, L., and Sun, G. (2020). "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp, 7132–7141. doi: 10.48550/arXiv.1709.01507

Kamilaris, A., and Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Comput. Electron. Agric.* 147, 70–90. doi: 10.1016/j.compag.2018.02.016

Kawakura, S., and Shibasaki, R. (2020). Deep learning-based self-driving car: Jetbot with nvidia ai board to deliver items at agricultural workplace with object-finding and avoidance functions. *Eur. J. Agric. Food Sci.* 2 (3). doi: 10.24018/ejfood.2020.2.3.45

Li, L. L., Sun, S. J., Mu, Y., Hu, Y., and Gong, T. L. (2022). Lightweight-convolutional neural network for apple leaf disease identification. *Frostierrs. Plant Sci.* 13, 831219. doi: 10.3389/fpls.2022.831219

Ma, N., Zhang, X., Zheng, H. T., and Sun, J. (2018). "Shufflenet V2: Practical guidelines for efficient cnn architecture design," In V. Ferrari, M. Hebert, C. Sminchisescu and Y. Weiss (eds). *Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science()*, Springer, Cham, vol 11218. doi: 10.1007/978-3-030-01264-9_8

Pang, L., Liu, H., Chen, Y., and Miao, J. (2020). Real-time concealed object detection from passive millimeter wave images based on the YOLOv3 algorithm. *Sensors* 20 (6), 1678. doi: 10.3390/s20061678

Sharma, A., Jain, A., Gupta, P., and Chowdary, V. (2020). Machine learning applications for precision agriculture: A comprehensive review. *IEEE Access* 9, 4843–4873. doi: 10.1109/ACCESS.2020.3048415

Shifang, S., Yan, Q., and Yuan, R. (2021). Recognition of grape leaf diseases and mobile application based on transfer learning. *Trans. Chin. Soc. Agric. Eng. (Transactions. CSAE).* 37 (10), 127–134. doi: 10.11975/j.issn.1002-6819.2021.10.015

Shun, Z., Yihong, G., and Jinjun, W. (2019). The development of deep convolutional neural networks and its applications on computer vision. *Chin. J. Comput.* 42 (3), 453–482. doi: 10.11897/SP.J.1016.2019.00453

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *IEEE Conference on Computer Vision and Pattern Recognition* 1, 1–14. doi: 10.48550/arXiv.1409.1556

Wang, D., and Wang, J. (2021). Crop disease classification with transfer learning and residual networks. *Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE)* 37 (4), 199–207. doi: 10.11975/j.issn.1002-6819.2021.4.024

Wenjie, Q. I. U., Jin, Y. E., Liangqing, H. U., Juan, Y. A. N. G., Qili, L. I., Jianyou, M. O., et al. (2021). Distilled-mobilenet model of convolutional neural network simplified structure for plant disease recognition. *Smart. Agric.* 3 (1), 109–117. doi: 10.12133/j.smartag.2021.3.1.202009-SA004

Yan, Q., Yang, B., Wang, W., Wang, B., Chen, P., and Zhang, J.. (2020). Apple leaf diseases recognition based on an improved convolutional neural network. *Sensors* 20 (12), 3535. doi: 10.3390/s20123535

Yiping, C., Jinchao, P., and Qiufeng, W. (2023). Apple leaf disease identification via improved CycleGAN and convolutional neural network. *Soft. Computing.* 27 (14), 9773–9786. doi: 10.1007/S00500-023-07811-Y

Yu, H., Cheng, X., Chen, C., Heidari, A., Liu, J., Cai, Z., et al. (2022). Apple leaf disease recognition method with improved residual network. *Multimed. Tools Appl.* 81, 7759–7782. doi: 10.1007/s11042-022-11915-2

Yueming, W., Meng, Q., Deqing, Z., and Hai, J. (2023). Obfuscation-resilient android malware detection based on graph convolution neural networks. *J. Software.* 34 (6), 0–0. doi: 10.13328/j.cnki.jos.006848

Zhang, Q. Q. (2021). *Study on the evolution and advantage evaluation of Apple production distribution in China* (Shanxi, China: Northwest A&F University).

Zhang, C., and Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. *J. Ind. Inf. Integration.* 23, 100224. doi: 10.1016/j.jii.2021.100224

Zhang, M., Su, H., and Wen, J. (2021). Classification of flower image based on attention mechanism and multi-loss attention network. *Comput. Commun.* 179, 307–317. doi: 10.1016/j.comcom.2021.09.001

Zheng, Y. Y., Kong, J. L., Jin, X. B., Wang, X. Y., and Zou, M. (2019). CropDeep: The crop vision dataset for deep-learning-based classification and detection in precision agriculture. *Sensors* 19 (5), 1058. doi: 10.3390/s19051058

Zhong, Y., and Zhao, M. (2020). Research on deep learning in apple leaf disease recognition. *Comput. Electron. Agric.* 168, 105146. doi: 10.1016/j.compag.2019.105146

Zhou, H., Liu, L., Zhang, H., and He, H. Y. (2022). "A novel structural re-parameterization block without extra training parameters," *2022 International Joint Conference on Neural Networks (IJCNN)*, Padua, Italy, 1–9. doi: 10.1109/IJCNN55064.2022.9892874

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 13 September 2023
DOI 10.3389/fpls.2023.1233241

Check for updates

OPEN ACCESS

EDITED BY
Alvaro Fuentes,
Jeonbuk National University,
Republic of Korea

REVIEWED BY
Fayadh Alenezi,
Jouf University, Saudi Arabia
Laura Falaschetti,
Marche Polytechnic University, Italy
Nitin Goyal,
Central University of Haryana, India

*CORRESPONDENCE
Bibo Lu
lubibo@hpu.edu.cn

†These authors have contributed equally to this work and share first authorship

RECEIVED 01 June 2023
ACCEPTED 18 August 2023
PUBLISHED 13 September 2023

CITATION
Lu B, Lu J, Xu X and Jin Y (2023) MixSeg: a lightweight and accurate mix structure network for semantic segmentation of apple leaf disease in complex environments.
*Front. Plant Sci.* 14:1233241.
doi: 10.3389/fpls.2023.1233241

COPYRIGHT
© 2023 Lu, Lu, Xu and Jin. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# MixSeg: a lightweight and accurate mix structure network for semantic segmentation of apple leaf disease in complex environments

Bibo Lu*†, Jiangwen Lu†, Xinchao Xu and Yuxin Jin

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, China

**Introduction:** Semantic segmentation is effective in dealing with complex environments. However, the most popular semantic segmentation methods are usually based on a single structure, they are inefficient and inaccurate. In this work, we propose a mix structure network called MixSeg, which fully combines the advantages of convolutional neural network, Transformer, and multi-layer perception architectures.

**Methods:** Specifically, MixSeg is an end-to-end semantic segmentation network, consisting of an encoder and a decoder. In the encoder, the Mix Transformer is designed to model globally and inject local bias into the model with less computational cost. The position indexer is developed to dynamically index absolute position information on the feature map. The local optimization module is designed to optimize the segmentation effect of the model on local edges and details. In the decoder, shallow and deep features are fused to output accurate segmentation results.

**Results:** Taking the apple leaf disease segmentation task in the real scene as an example, the segmentation effect of the MixSeg is verified. The experimental results show that MixSeg has the best segmentation effect and the lowest parameters and floating point operations compared with the mainstream semantic segmentation methods on small datasets. On apple alternaria blotch and apple grey spot leaf image datasets, the most lightweight MixSeg-T achieves 98.22%, 98.09% intersection over union for leaf segmentation and 87.40%, 86.20% intersection over union for disease segmentation.

**Discussion:** Thus, the performance of MixSeg demonstrates that it can provide a more efficient and stable method for accurate segmentation of leaves and diseases in complex environments.

KEYWORDS

**MixSeg, CNN, Transformer, MLP, fruit leaf disease, semantic segmentation**

# 1 Introduction

Apples are rich in nutritional value and are one of the most important cash crops in the world. However, apple leaves are often affected by pests and diseases, and failure to identify and prevent diseases promptly can easily lead to a decrease in fruit quality and yield, causing severe economic losses to growers (Bansal et al., 2021; Yogeshwari and Thailambal, 2021; Aggarwal et al., 2023). In reality, the identification of crop diseases usually relies on manual work (Chen et al., 2022; Mu et al., 2022). The manual approach has many drawbacks, such as excessive subjectivity, low efficiency, and high recognition error rate (Barman et al., 2020; Xu et al., 2023). Therefore, computer vision technology can be used to identify diseases quickly and accurately, which plays a crucial role in effective disease prevention and higher apple fruit quality and yield.

In recent years, with the rapid development of computer vision technology, deep learning had gained much attention in crop disease identification. More and more researchers used semantic segmentation methods for crop disease recognition. For example, (Storey et al., 2022) used ResNet-50 and MobileNetV3 as the backbone of Mask-RCNN and then segmented apple disease leaf disease images separately to test their suitability for this task. (Divyanth et al., 2023) researched the advantages and disadvantages of the semantic segmentation networks SegNet, U-Net and DeepLabv3+, and chose U-Net and DeepLabv3+ for segmentation of corn leaves and disease spots, respectively. (Wu et al., 2021) tested the effect of DeepLabv3+ with different backbone networks, Xception-65, Xception-71, ResNet-50, and ResNet-101, on the performance of leaf and disease spot segmentation of hydroponic lettuce, providing a guide for an automatic selection and segmentation device for hydroponic lettuce. (Agarwal et al., 2021) chose ResNet as the encoder of U-Net to achieve effective segmentation of crop disease leaf images. (Yuan et al., 2022) replaced the backbone network of DeepLabv3+ with the lightweight MobileNetV2 to reduce the training time of the model and was able to segment the leaf veins in real-time. Therefore, semantic segmentation methods are suitable for recognizing crop leaves and diseases in complex environments.

Currently, the mainstream semantic segmentation models are mainly based on convolutional neural network (CNN) structure and Transformer structure. CNN achieved a certain degree of offset, scale, and distortion invariance by forcing the capture of local priors using local perceptual fields. For the same target appearing at different locations in the image, all feature representations with some similarity can be extracted by CNN (Zeiler and Fergus, 2014; He et al., 2015). In addition, CNN had a hierarchical learning model, from simple low-level textures to higher-order semantic patterns. This property of CNN allowed for strong robustness and generalization when dealing with problems such as target recognition (Simonyan and Zisserman, 2014). Therefore, CNN was useful for extracting and optimizing local leaf and disease features in apple leaf disease image segmentation. However, the convolution operation focuses only on local regions, which can lead to general limitations of CNN for modeling relationships between distant pixels (Kuo, 2016; Hu et al., 2018). (Simonyan and Zisserman, 2014) extended the ability to extract global features by stacking deeper networks to increase the field of perception. While this alleviated the general limitation of CNN for modeling direct long-range relations, too deep layers introduced the problem of gradient degradation, which led to degradation of the model's performance (He et al., 2016). Some researchers had started using self-attention instead of convolution to extract global features (Hu et al., 2019; Ramachandran et al., 2019).

Transformer (Vaswani et al., 2017) was first applied in NLP and gradually received attention in computer vision due to its excellent performance in processing long text sequences. Unlike CNN, Transformer used self-attention to learn the relationship between pixels and regions. It can perform spatial transformations and extract feature dependencies between distant pixels, so Transformer had more flexibility. (Dosovitskiy et al., 2020) first proposed the visual Transformer ViT, which treated an image as a set of sequences and uses self-attention for global modeling. Subsequently, a series of Transformer-based semantic segmentation models started to appear, such as SegFormer (Xie et al., 2021a), PoolFormer (Yu et al., 2022), and SETR (Zheng et al., 2021). However, the semantic segmentation networks with self-attention as the global modeling paradigm had some problems. For example, self-attention had a quadratic computational complexity for the input token sequence, and therefore, the network was not conducive to high-resolution input. In addition, self-attention had many parameters and was prone to overfitting when the dataset is small (Huang et al., 2019; Wang et al., 2020; Guo et al., 2022). Moreover, self-attention cannot encode the position and cannot recover the target's position information in the decoder. These problems eventually led to less efficient network segmentation and poor segmentation results. Multi-layer perception (MLP) was a typical feedforward neural network consisting of multiple layers of neurons, each fully connected to the previous layer. Compared to self-attention, MLP aimed to establish weights for all features. Thus it possessed a more robust ability than self-attention to extract the dependencies of features that show long-range (Ding et al., 2021). Moreover, MLP can be viewed as a combination of multiple linear transformations and nonlinear functions, and it only needed to learn the weights and biases of each layer, so MLP was less susceptible to noise and erroneous inputs and was more stable in computation. However, the traditional MLP had many parameters, was computationally complex, and was also prone to overfitting. In summary, the correct use of local prior, global dependence and control of computational complexity are the keys to improving the performance of semantic segmentation models.

In response to the above problems of single structured networks, the goal of this study was to design a mix structure semantic segmentation model by extracting the core strengths of CNN, Transformer and MLP architectures to improve the global modeling efficiency of the model while maintaining a strong mastery of detailed features and being more lightweight. Specifically, MixSeg consisted of three key components: Mix Transformer, position indexer (PI), and local optimization module (LOM). Mix Transformer aimed to leverage the local feature injection of CNN, the stability of global modeling

structure of Transformer, and the ability of MLP to establish complete global dependency at a smaller computational cost for extracting more comprehensive global features. The role of the position indexer was to accurately preserve the position information of target leaves during the downsampling process in order to enhance the model's anti-interference capability and segmentation performance. LOM, on the other hand, focused on optimizing the extraction of local features such as disease leaf edges and details. In summary, MixSeg was a mix structure model applied to segment apple leaves and diseases in complex environments. Compared with the single structure model, it was capable of better segmenting target leaves and diseases while being more lightweight and suitable for practical applications. The main contributions were as follows:

(1) A lightweight and accurate mix structure semantic segmentation model (MixSeg) was designed to improve the segmentation performance of apple leaves and disease spots in complex environments with fewer parameters and computational effort.
(2) A novel Mix Transformer was designed, which uses the depthwise separable convolution of the residual for local bias injection, and the designed MMLP to reduce the amount of computation and establish global dependencies.
(3) The PI was designed to dynamically index absolute position information on the feature map and can be independent of variable length inputs.
(4) The LOM was designed to enhance the model's ability to represent local features, optimize the segmentation effect of leaf edges and extract more tiny spots.

The rest of the paper was organized as follows: the dataset and the details of the proposed MixSeg were presented in Section 2. Then, the experimental results were presented and analyzed in Section 3. Next, the limitations of MixSeg were analyzed by discussing the proposed method in Section 4. Finally, the work of this study was summarized in Section 5 and future research directions are envisioned.

# 2 Materials and methods

## 2.1 Data collection and annotation

The data for this work were obtained from Northwest Agriculture and Forestry University manual photpgraphy was adopted to obtain images of apple leaf diseases in real outdoor scenarios. A total of two common apple leaf diseases were collected, namely apple alternaria blotch and apple grey spot. In addition, the weather in which these dataset images were taken includes both sunny and rainy days, which enhanced the dataset's diversity and authenticity.

To improve the training accuracy, we used the professional semantic segmentation labeling software Labelme to label the original images under the guidance of experts. We finely labelled the edges of each leaf and disease by a pixel-by-pixel approach. Then corresponding label files was generated, where the background, leaf, and disease pixel values were set as 0, 1, 2, respectively. The partial images and labels of the two apple leaf datasets were illustrated in Figure 1, where the black, green, and red parts of the label represented the background, leaf and disease area, respectively.

## 2.2 Data augmentation

For deep learning methods, too little data volume tended to lead to model overfitting and poor generalization. (Lee et al., 2019). Therefore, it was necessary to use data augmentation methods to properly extend the two apple leaf datasets. A total of six data enhancement methods were applied: (1) Geometric deformation: the original image and labels were cropped, randomly folded, or rotated simultaneously to obtain new images and labels. (2) Chromaticity change: the original image brightness, contrast, or saturation was changed to obtain a new image, and the labels remain unchanged. These data enhancement methods simulate the changes in shooting angle and illumination during data acquisition,

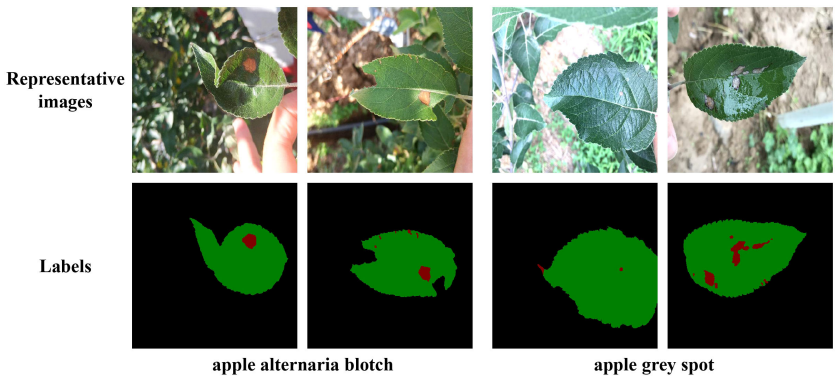



FIGURE 1
Representative images of two types of apple leaf disease spots and their corresponding labels. In the labels, the black, green and red parts represent the background, the leaf and the diseased area, respectively.

which can improve the model's robustness and generalization ability. The apple alternaria blotch disease dataset had 256 original images and 1536 enhanced images, totaling 1792 images. Apple grey spot disease dataset had 162 original images and 972 enhanced images, totaling 1134 images. Taking the apple alternaria blotch leaf image dataset as an example, the enhanced images were shown in Figure 2.

## 2.3 Design for MixSeg

In order to improve the segmentation accuracy and efficiency, a mix structure network MixSeg combining the advantages of CNN, Transformer and MLP was designed, the overall structure was shown in Figure 3. MixSeg adopted an encoding-decoding structure. In the encoder, the network consisted of four sets of Mix stages. In each Mix stage, it consisted of a patch embedding and Mix block. In addition, the core components of Mix block consisted of Mix Transformer, PI, and LOM designed by us. In the decoder, shallow and deep features were fused and accurate segmentation results were output by two 3×3 depthwise separable convolutions.

Specifically, the image size of the input network was 512×512×3. We used a 7×7 patch in Mix stage 1, and stride was set to 4. The token serialization of the input image was performed in an overlapping manner to reduce the parameters of the initial input network. Each token was reshaped into a 147 dimensional vector and mapped to a $C_1$ dimensional embedding through a linear layer. Immediately afterward, the extracted token sequence sets were fed into Mix Transformer for global modeling. In particular, a PI was inserted after each Mix Transformer to prevent the absolute position information of the target from being lost when building global dependencies. Then, the output of the Mix Transformer was optimized by a LOM for local features. After the Mix stage 1, the feature map $F_1$ with dimension $(\frac{H}{4} \times \frac{W}{4} \times C_1)$ was output, and the inter-level quadruple downsampling was completed. In addition, the channel dimensions $C_i$(i=1, 2, 3, 4) of the output feature maps in MixSeg-T and MixSeg-S were [32, 64, 160, 256], and MixSeg-M, MixSeg-L were [64, 128, 256, 512]. Other Mix stages all use 3×3 size patches with stride 2, the inter-level double downsampling was performed, and the multi-scale feature maps $F_2$, $F_3$, $F_4$ with $\{\frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$ of the original input are output. $F_1$ and $F_4$ perform feature fusion after 2x and 16x upsampling, and finally produced fine segmentation results by two depthwise separable convolutions. The network innovations were explained in detail in the remainder of this section. Mix Transformer, PI, and LOM were introduced in Section 2.3.1, 2.3.2, and 2.3.3, respectively.

### 2.3.1 Mix Transformer

Traditional Transformer architecture networks, such as ViT, usually consisted of self-attention and channel MLP to achieve spatial and channel information mixing. When pre-trained on a large dataset, these traditional networks performed well on image recognition tasks. However, in real production situations, data acquisition and labeling were difficult and suffer from insufficient sample size. In addition, in segmentation tasks, the image size of the input network was usually large, and self-attention for global modeling requires dealing with many long-range pixel dependencies. Therefore, semantic segmentation networks used self-attention as the design paradigm were usually more complex, and they were prone to overfitting phenomena on small-scale datasets, which was not conducive to applications in real production. In this subsection, we proposed a new Transformer design paradigm named Mix Transformer. It utilized depthwise separable convolution to inject local bias. It further utilized the designed multi-dimensional hybrid MLP (MMLP) to reduce the computational complexity and built a complete global dependency in multiple dimensions to make the model better match small datasets. The structure of the Mix Transformer was demonstrated in Figure 3.

As shown in Figure 3, the Mix Transformer maintained the basic architecture of the Transformer, which was composed of two main residual blocks. In particular, local bias injection was performed in the first main residual block, which consisted of GroupNorm and 3×3, 5×5 depthwise separable convolution of

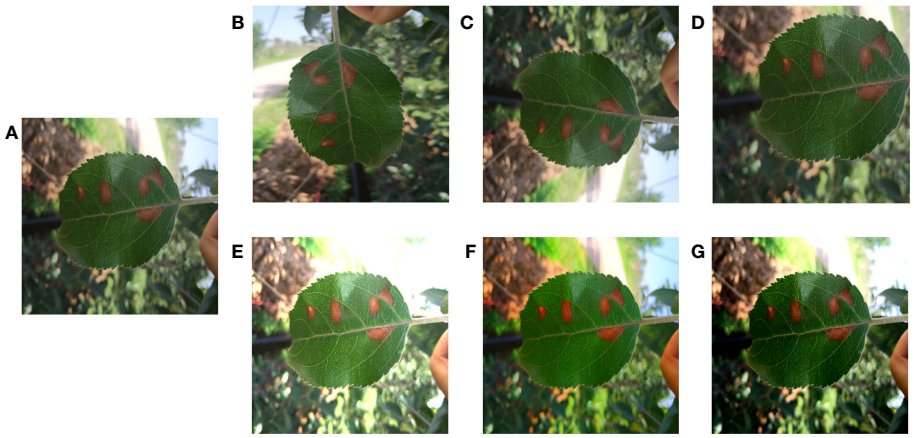



FIGURE 2
Image enhancement. **(A)** Original image. **(B)** Image rotation. **(C)** Image flip. **(D)** Image crop. **(E)** Saturation enhancement. **(F)** Brightness enhancement. **(G)** Contrast enhancement.

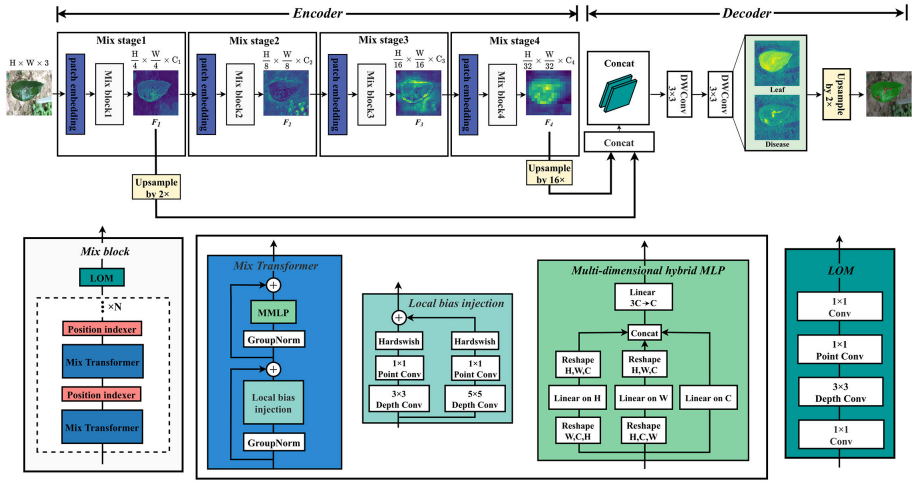



FIGURE 3
MixSeg overall architecture. The encoder of MixSeg consists of four MixStages, each with a patch embedding for feature serialization and a Mix block for feature extraction. Mix transformer, PI, and LOM together form the Mix block. In addition, N represents the number of stacks of Mix Transformer and PI, which is determined by the size of the network designed in this paper. In the decoder, DWConv represents the depthwise separable convolution. The decoder fuses Mix Stage1 and Mix Stage4 output results and outputs the fine-grained segmentation results by two DWConv.

residuals connected. Global feature dependencies were established in the second main residual block, which consisted of GroupNorm and MMLP.

Local biasing was a common technique in CNN, which summed a fixed bias value to each position of the input data convolved by an additive operation (Zhuang et al., 2019). Such an operation allowed each convolutional kernel to learn different features and produced corresponding responses to various input data locations. Self-attention did not have local bias, which made some noise and interference in the input data may affect the computational results. Therefore, we chose depthwise separable 3×3 and 5×5 residual convolutions instead of self-attention to inject local bias to the token sequence set. Specifically, the first main residual block can be expressed as:

$$Y_i = (DWConv_{3\times3}(Norm(X_i)) + DWConv_{5\times5}(Norm(X_i))) + X_i \quad (1)$$

where $X_i$(i=1, 2, 3, 4) was the output of the patch embedding in each Mix stage, $X_i \in R^{(N_i \times C_i)}$. $N$ represented the number of token sequences, $C$ stood for their sequence dimension. Then, $X_i$ was reshaped as $X_i \in R^{(H_i \times W_i \times C_i)}$. Thus after the input local bias injection $Y_i \in R^{(H_i \times W_i \times C_i)}$. In addition, DWConv presented the depthwise separable convolution. 3×3 and 5×5 DWConv required few parameters to inject rich multiscale local bias to the set of token sequences extracted by patch embedding.

Transformer architecture usually used channel MLP for mixing channel information. Channel MLP interacted with all the elements in each token, so the computational complexity was large when the token sequence was long in dimension, and overfitting was likely to occur. In addition, channel MLP can only mix information on the channels. Therefore, in the second main residual block, we designed MMLP to improve the original MLP by reducing the computational effort and jointly establishing global dependencies in multiple dimensions to make the model less prone to overfitting when training on small datasets. Figure 4 illustrated the specific architecture of MMLP.

Specifically, $Y_i$ obtained from Equation 1 was used as input. Taking the branching path in the $H$ dimension as an example, this data was reshaped as $Y_H i \in R^{(W_i \times C_i \times H_i)}$, and then linear feature mapping in the $H$ dimension was performed to output $\hat{Y}_{H^i}$, where the $H$ dimension shared the weight $W_W i \in R^{(H_i \times H_i)}$. The same calculation was used in the $W$, $C$ dimension. Immediately afterward, the outputs of these three branches were fused in the channel dimension. Then the number of channels was adjusted from $3C_i$ to $C_i$. The computation of the second component can be expressed as:

$$\begin{aligned} Y_{H^i} &= Reshape(W_i, C_i, H_i)(Y_i), \forall i \\ Y_{W^i} &= Reshape(H_i, C_i, W_i)(Y_i), \forall i \end{aligned} \quad (2)$$

$$\begin{aligned} \hat{Y}_{H^i} &= Reshape(H_i, W_i, C_i) Linear(H_i, H_i)(Y_{H^i}), \forall i \\ \hat{Y}_{W^i} &= Reshape(H_i, W_i, C_i) Linear(W_i, W_i)(Y_{W^i}), \forall i \\ Y_{C^i} &= Linear(C_i, C_i)(Y_i), \forall i \end{aligned} \quad (3)$$

$$Z_i = Linear(3C_i, C_i)(Concat(\hat{Y}_{H^i}, \hat{Y}_{W^i}, Y_{C^i})), \forall i \quad (4)$$

Such a design of MMLP can significantly reduce the number of parameters and computation, and can establish global dependencies in different dimensions at the same time, and then summarize this global information in the mixing stage to achieve a more complete global dependency establishment. The number of parameters of MMLP module can be calculated as:

$$Params_{MMLP} = H^2 + W^2 + C^2 + 3C^2 \quad (5)$$

where the parameter of $3C^2$ was used in the mixing phase of multi-dimensional information. In contrast, the parameter number of conventional MLP module was:

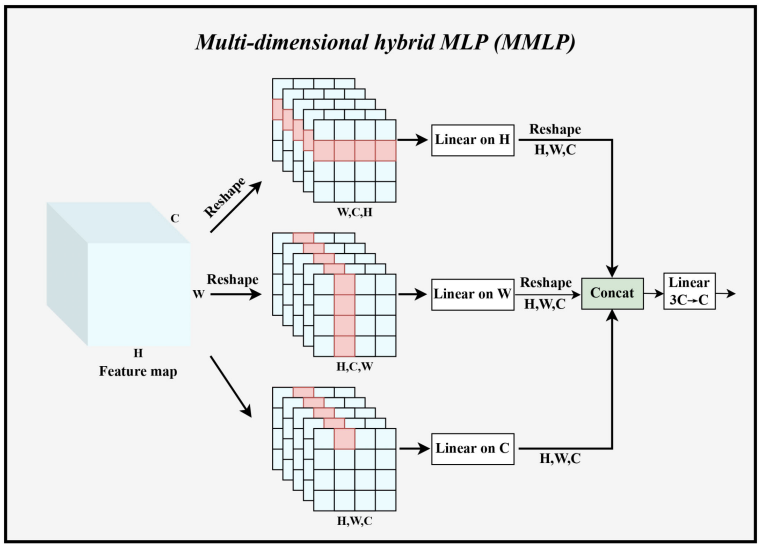



FIGURE 4
The specific implementation process of MMLP. Linear represents linear feature mapping, Concat indicates stitching the fused feature maps in the channel dimension, and Reshape denotes changing the shape of the feature maps to the specified dimension.

$$Params_{MLP} = 2\tau(HW)^2 \tag{6}$$

where 2 represented the two fully connected processes of expansion to compression, $\tau$ being the ratio of the expansion of the MLP layer. In addition, the computational complexity of MMLP module can be expressed as:

$$FLOPs_{MMLP} = HWC(H + W + C) + 3HWC^2 \tag{7}$$

the computational complexity of MLP module can be expressed as:

$$FLOPs_{MLP} = 2\tau(HW)^2C \tag{8}$$

among them, the product of $H$ and $W$ represents the number of tokens, which was denoted as $N$. It can be found that the computational complexity of an MMLP grows with $N\sqrt{N}$, and the MLP increases with $N^2$.

In summary, Mix Transformer was designed to input local bias to the network and model it globally by building a depthwise separable convolution of residuals and MMLP. Compared with the traditional Transformer architecture, Mix Transformer had lower computational complexity, which made the model lighter and, thus, more suitable for small datasets.

### 2.3.2 Position indexer

Traditional Transformer architectures implemented location information awareness by adding absolute location codes to each input token (Dosovitskiy et al., 2020). These location encodings were learnable and efficient but were disrupted when the Transformer processes longer inputs, made it impossible to implement a pyramidal downsampling pattern while preserving absolute location information. For example, ViT used a downsampling rate of 16 times at each stage of the model to maintain the order of these position-encoded vectors, which made it unable to handle multi-scale target features. Therefore, the PI that can dynamically change with the input size was proposed in this subsection, which enabled the encoder to perform multi-scale downsampling while retaining absolute position information. The schematic diagram of the PI was displayed in Figure 5.

In CNN, when adding different amounts of zero-padding to the input data, the convolution can predict the absolute position information of the input by learning the position and amount of zero-padding (Islam et al., 2020). Therefore, as shown in Figure 5, the PI performed dynamic position indexing with the padding around the input token sequence group as the domain condition. Specifically, taking $Z_i$ of Equation 4 as the input, we first added a set of padding around $Z_i$ for learning the position information, and the dimension of this padding was equal to $\frac{k-1}{2}$, and $k$ was the convolution kernel size. Then, group convolution was performed using the 2D convolution of size $k$ to index the position information on each channel dimension and fuse it, where $groups = C_i$, recorded the output result as $\hat{Z}_i$. Thus, the number of parameters of the PI can be calculated as:

$$Params_{PI} = k^2C_i \tag{9}$$

the amount of computation can be calculated as:

$$FLOPs_{PI} = k^2C_iWH \tag{10}$$

In addition, the PI was added after each Mix Transformer to index the absolute position information of the pixel points after establishing the global dependency.

Therefore, the PI indexed the absolute position information of the target dynamically. Such structure did not affect the translational invariance of the model and allowed the encoder to preserve the absolute position information while performing multi-scale downsampling. This helped to recover the position information accurately in the decoder and improve the segmentation performance of the model.

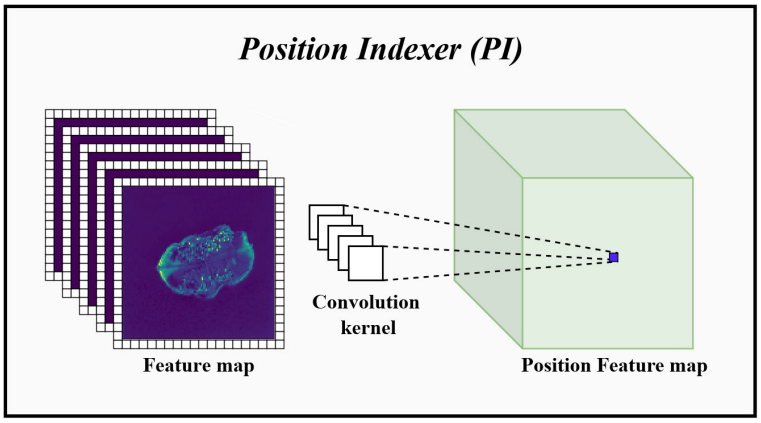



FIGURE 5
The specific implementation process of PI. Convolution kernel represents the convolution kernel group for position information indexing. Before entering the PI, the feature map will add a set of padding for learning position information around it. Then, use these convolution kernels to perform 2D group convolution on the feature map, index the position information in each channel dimension and perform fusion.

### 2.3.3 Local optimization module

In the task of apple leaf disease segmentation, the proportion of disease pixels to the full image pixels was small, making the extraction of small disease features more difficult. In addition, the jagged features of the leaf edges were difficult to extract in complex environments, which can reduce the accuracy of leaf segmentation. The Mix Transformer we proposed in 2.3.1 was capable of global modeling, yet its ability to extract edge features was relatively weak and may miss small spots. Therefore, in this subsection, the LOM was designed to optimize the segmentation of leaf edges and extracted more tiny spots through using CNN's ability to extract local features. The structure of LOM was presented in Figure 3.

As shown in Figure 3, the designed LOM first used a 1×1 convolution to up-dimension, then letted a 3×3 depthwise separable convolution perform local feature extraction in higher dimensions, and finally used a 1×1 convolution to adjust back to the initial dimension. By stacking multiple convolutions, the LOM can gradually capture different levels of texture features of the image and combine them to form a higher-level feature representation. Therefore, the LOM was added to the end of the Mix block to enhance the ability of the model to extract local features. Specifically, the $\hat{Z}_i$ output by the PI was used as input, the LOM can be expressed as:

$$\hat{Z}_i^{'} = Conv_{1\times1}(DWConv_{3\times3}(Conv_{1\times1}(\hat{Z}_i))) \quad (11)$$

By optimizing the local features of the feature layer output from Mix Transformer, LOM can effectively enhance the model's ability to extract detailed features, thus optimizing the segmentation effect of leaf edges and extracting more tiny spots.

## 3 Results

In this section, the experimental environment, hyper-parameter settings and dataset partitioning were presented in subsection 3.1. Then, the rationale and computational formulas for the evaluation metrics used in this study were presented in subsection 3.2. Thereafter, subsection 3.3 conducted ablation experiments to explore the effects of different sizes of MixSeg on segmentation performance. Next, subsection 3.4 compared the segmentation accuracy of different models on two outdoor apple disease leaf image datasets. Then, the segmentation results of different models in real scenes were visualized in subsection 3.5. Subsequently, the effective region of focus was analyzed in subsection 3.6.

## 3.1 Experiment setup

The hardware configuration of the experimental environment for this study was as follows: Intel(R) Core(TM) i5-12400F, 16 G memory, NVIDIA GeForce RTX3060ti, 64-bit Windows operating system. The model was built by the Pytorch framework; the version of PyTorch was 1.10.0. After several trials, the hyperparameters were set as follows: optimizer was Adamw, momentum was 0.9, weight decay was 1e-2, batch size was 8, the initial learning rate was 1e-4, the minimum learning rate was 1e-7, learning rate decay strategy was cos, drop path rate was 0.1, and epoch was 300.

We divided the two apple leaf datasets into a training set and a test set according to the 8:2 ratio. In addition, each dataset will be divided into a training set and a validation set according to 9:1 for cross-validation in the training phase.

## 3.2 Evaluation indicators

The following three evaluation metrics were selected to measure the segmentation effectiveness of the model: intersection over union (IoU), mean pixel accuracy (mPA) and pixel accuracy (PA).

In the segmentation task, IoU represents the ratio of the intersection and union among the predicted results of a category and the actual values of that category. mPA was the cumulative average of the proportion of pixels correctly classified in each

category. PA represented the proportion of all correctly predicted pixels to all pixels. IoU, mPA and PA were calculated as follows:

$$IoU = \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{12}$$

$$mPA = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}} \tag{13}$$

$$PA = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k} \sum_{j=0}^{k} p_{ij}} \tag{14}$$

where $k$ +1 was the number of categories plus background, $p_{ii}$ was the number of correctly predicted pixels, $p_{ij}$ denoted the number of pixels belonging to category $i$ but predicted as category $j$, and $p_{ji}$ denoted the number of pixels belonging to category $j$ but predicted as category $i$.

## 3.3 Ablation studies

In this subsection, ablation experiments were performed on the model structure of MixSeg. The following tests were performed in the same experimental environment and utilized identical training parameters. In addition, total parameters, floating point operations (FLOPs), inference time, frames per second (FPS), IoU, mPA, and PA were used to evaluate the segmentation performance of the different models.

### 3.3.1 Performance comparison between different versions of MixSeg

We designed four versions of MixSeg with different sizes by varying the output dimension of the model and the stacking number of Mix Transformer, which was named MixSeg-T, MixSeg-S, MixSeg-M, and MixSeg-L according to the complexity of the model. The results were used to analyze the effects of different output dimensions of each Mix stage and the number of Mix Transformer stacks on the model's segmentation efficiency and segmentation accuracy. Tables 1 and 2 demonstrated the segmentation performance comparison results of each version of the model.

According to Table 1, increased the channel dimension of the Mix stage and the number of stacks of the Mix Transformer made the network more complex, reduced the efficiency of the network, and therefore it will lead to longer inference time and less FPS. Since the Mix Transformer we design was lightweight, increased the number of stacks of the Mix Transformer had less impact on the network complexity than increasing the channel dimension of the Mix stage. From Table 3, increased the network size can improve the segmentation accuracy on the test set of apple disease leaf images. In addition, the proposed lightweight models MixSeg-T and MixSeg-S were fast and efficient by the combined comparison of Tables 1 and 2, and they took into account the lightweight and maintain competitive performance in segmentation accuracy.

Thus, researchers can deploy MixSeg in different sizes for mobile devices in real leaf disease diagnosis scenarios depending on the specific situation. Generally, a larger MixSeg model will have higher segmentation accuracy but requires more computational resources and inference time. Therefore, researchers can choose a smaller MixSeg model for resource-constrained application scenarios to obtain faster inference speed and less computational resource consumption. In contrast, they can choose a larger MixSeg model for high-precision segmentation scenarios to obtain higher segmentation accuracy.

### 3.3.2 Performance verification of PI and LOM

We designed four sets of experiments to verify the performance of the proposed PI and LOM. Specifically, Test 1 used MixSeg-T as the base framework and removes the PI and LOM, which were set as the baseline. Test 2 and Test 3 were both based on Test 1, which introduced the PI and LOM separately. In addition, Test 4, which was the method proposed in this paper, introduces both the PI and the LOM on top of Test 1. The evaluation results of the different test models were shown in Tables 3, 4 were shown.

As shown in Table 3, by compared Test 1 and Test 2, the model with the introduction of PI outperformed the baseline model in segmentation accuracy. On apple alternaria blotch and apple grey spot test sets, the IoU for leaf segmentation was improved by 2.14% and 2.45%, respectively. The IoU for lesion segmentation was improved by 1.6% and 2.6%, respectively. Compared to Test 1 and Test 3, the segmentation accuracy of the model was improved more significantly compared to the baseline model after the introduction of LOM. The IoU for leaf segmentation was improved by 9.00% and 9.38%, respectively. The IoU for lesion segmentation was improved by 7.71% and 8.96%, respectively. The results indicate that PI helps to accurately recover the position information of the target leaf, lesion in the decoder due to its ability to dynamically index the absolute position information of the target, which in turn improves the segmentation effect of the

TABLE 1 Performance metrics results for different versions of MixSeg.

| Version | Output dimension | Mix Transformer stacking number | Total parameters/ M | FLOPs/ G | Inference time/ ms | FPS |
|---|---|---|---|---|---|---|
| MixSeg-T | [32,64,160,256] | [2,2,2,2] | **1.90** | **4.17** | **8.70** | **114.82** |
| MixSeg-S | [32,64,160,256] | [4,4,4,4] | 2.97 | 5.33 | 12.67 | 78.96 |
| MixSeg-M | [64,128,256,512] | [2,2,2,2] | 6.44 | 14.38 | 18.93 | 52.80 |
| MixSeg-L | [64,128,256,512] | [4,4,4,4] | 10.03 | 17.85 | 27.00 | 37.03 |

The bold font denotes which MixSeg version has the best performance on a particular evaluation metric.

TABLE 2 The results of segmentation accuracy for different versions of MixSeg.

| Version | Apple alternaria blotch | | | | Apple grey spot | | | |
|---|---|---|---|---|---|---|---|---|
| | IoU/% | | mPA | PA | IoU/% | | mPA | PA |
| | Leaf | Disease | | | Leaf | Disease | | |
| MixSeg-T | 98.22 | 87.40 | 97.21 | 99.45 | 98.09 | 86.20 | 97.26 | 99.46 |
| MixSeg-S | 98.39 | 88.55 | 97.54 | 99.50 | 98.32 | 87.98 | 97.62 | 99.53 |
| MixSeg-M | 98.41 | 88.63 | 97.58 | 99.51 | 98.40 | 88.12 | 97.67 | 99.55 |
| MixSeg-L | **98.53** | **89.55** | **97.87** | **99.55** | **98.49** | **88.87** | **97.93** | **99.58** |

The bold font denotes which MixSeg version has the best performance on a particular evaluation metric.

model. And LOM can optimize the segmentation effect of leaf edges and extract more tiny spots, so it can also improve the segmentation effect of the model. In comparison with Test 4 and other tests, the combination of both PI and LOM, introduced on the baseline model, gave the best performance in leaf and spot segmentation. The results demonstrate that the proposed method is designed to maximize the segmentation accuracy of the model as the LOM is able to accurately and efficiently optimize the edge features of the target leaves and diseases after the PI indexes to the absolute position information of the target leaves and spots. In addition, observed Table 4, the effects of PI and LOM on model inference time, total parameters and FLOPs were relatively minor. However, compared Test 1 and Test 4, the proposed method decreased 27.87 in FPS compared to the baseline model due to the increase in network complexity. Due to the complexity of the orchard environment, sacrifice of some FPS to ensure accurate segmentation of apple leaves and diseases was necessary. In summary, the proposed method had good segmentation performance in the task of segmenting apple diseased leaf images.

## 3.4 Comparsion of different models

In this subsection, to demonstrate the advantages of the MixSeg model, MixSeg was compared with popular deep learning semantic segmentation models, including CNN-based and Transformer-based single structure models. These models were PSPNet (Zhao et al., 2017), HRNetV2 (Sun et al., 2019), U-Net (Ronneberger et al., 2015), DeepLabv3+ (Chen et al., 2018), and SegFormer (Xie et al., 2021b). PSPNet used a pyramid pooling module to efficiently exploit contextual information. HRNetV2 provided rich semantic information by simultaneously extracting features at branches of different resolutions and then fusing features from these branches to maintain high resolution while providing rich semantic information. U-Net used a symmetric encoding-decoding structure to achieve multi-scale information fusion by jump linking. DeepLabv3+ used atrous spatial pyramid pooling (ASPP) to efficiently enhance the receptive field and mitigate the information loss problem caused by pooling. SegFormer was a Transformer semantic segmentation-based model, which further improves the segmentation efficiency of the model by reducing the number of parameters of self-attention and decoder. To be fair, all models were placed in the same experimental environment as MixSeg, and both were trained on two outdoor apple disease leaf image datasets. The segmentation accuracy of different models was shown in Table 5.

As illustrated in Table 5, MixSeg-T performed best on both outdoor apple diseased leaf image datasets. Taking the apple alternaria blotch leaf dataset as an example, the same results were shown on another dataset. MixSeg-T was much better than PSPNet in segmentation accuracy. Compared with PSPNet, MixSeg-T achieved 2.75% higher IoU for leaf segmentation and 15.6% higher IoU for lesion segmentation. HRNetV2 had less accuracy than MixSeg-T in both leaf and disease spot segmentation, with the

TABLE 3 The results of segmentation accuracy for different test models.

| Test No. | Model | Apple alternaria blotch | | | | Apple grey spot | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | IoU/% | | mPA | PA | IoU/% | | mPA | PA |
| | | Leaf | Disease | | | Leaf | Disease | | |
| 1 | Baseline | 88.48 | 76.71 | 93.79 | 96.07 | 87.53 | 73.26 | 93.65 | 96.05 |
| 2 | Baseline+PI | 90.62 | 78.31 | 94.28 | 96.88 | 89.98 | 75.86 | 94.19 | 96.92 |
| 3 | Baseline+LOM | 97.48 | 84.42 | 96.48 | 99.21 | 97.36 | 82.22 | 96.18 | 99.25 |
| 4 | Baseline+PI+LOM (Proposed method) | **98.22** | **87.40** | **97.21** | **99.45** | **98.09** | **86.20** | **97.26** | **99.46** |

The bold font indicates which model performs best on a particular evaluation metric.

TABLE 4 Performance metrics results for different test models.

| Test No. | Model | Inference time/ms | FPS | Total parameters/M | FLOPs/G |
|---|---|---|---|---|---|
| 1 | Baseline | **7.00** | **142.69** | **1.70** | **3.99** |
| 2 | Baseline+PI | 7.37 | 135.68 | 1.71 | 4.01 |
| 3 | Baseline+LOM | 8.48 | 120.32 | 1.89 | 4.15 |
| 4 | Baseline+PI+LOM (Proposed method) | 8.70 | 114.82 | 1.90 | 4.17 |

The bold font indicates which model performs best on a particular evaluation metric.

IoU of leaf segmentation being 3.1% lower than that of MixSeg-T and the IoU of disease spot segmentation being 8.17% lower than that of MixSeg-T. Compared to SegFormer, MixSeg-T achieved 1.42% higher IoU for leaf segmentation and 5.18% higher IoU for lesion segmentation. In addition, DeepLabv3+ and U-Net were closer and better than other methods in segmentation accuracy, but both were inferior to MixSeg-T. DeepLabV3+ was 1.37% and 4.85% lower than the proposed method in the IoU of leaf and spot segmentation, respectively. At the same time, U-Net was 2.73% and 3.68% lower in the IoU of leaf and spot segmentation, respectively.

This indicates that the combined CNN, Tranformer and MLP models outperform the single structure model in terms of segmentation accuracy for apple leaves and diseases in outdoor environments. Therefore, the more accurate segmentation capability of MixSeg can help growers better analyze the disease status of their crops, detect pests and diseases in time and take effective preventive measures. It can also help researchers gain a deeper understanding of the characteristics and development patterns of crop diseases and improve the accuracy and efficiency of disease diagnosis.

Table 6 recorded the inference time, FPS, total parameters and FLOPs for all models. According to Table 6, MixSeg-T exhibited a more lightweight performance. MixSeg-T was 30.97ms faster than PSPNet in inference time, and the FPS was 89.62 higher. Moreover, the total parameters and FLOPs of MixSeg-T were only 3.9% and 6.8% of PSPNet. The inference time of HRNetV2 was 20.87ms slower than MixSeg-T, and the FPS was 81.01 less. Compared with the lightweight SegFormer, the inference time of MixSeg-T was 1.84ms faster, the FPS was 20.02 higher, and the total parameters and FLOPs are lower than SegFormer. In contrast to MixSeg-T, the inference time of U-Net was 36.06ms slower than MixSeg-T, and the FPS was 92.49 lower. Moreover, the total parameters and FLOPs of MixSeg-T were only 7.6% and 1.8% of U-Net. In addition, the inference speed and FPS of DeepLabv3+ were close to those of MixSeg-T, but MixSeg-T had fewer total parameters and FLOPs.

In practical scenarios, apple trees in outdoor environments can be widely dispersed, and the diagnosis of leaf diseases needs to be conducted at various locations. This necessitates models with smaller scale and lower computational requirements to be optimally deployed on mobile devices, enhancing the flexibility and efficiency of the diagnosis process. Experimental findings demonstrate that MixSeg outperforms mainstream segmentation models in terms of lightweight characteristics, rendering it more suitable for real-world applications. It can be swiftly deployed on mobile devices such as smartphones, drones, and application robots, enabling efficient diagnosis of leaf diseases. Consequently, this facilitates the timely implementation of effective preventive measures.

## 3.5 Visualization of apple diseased leaf segmentation results

To validate the segmentation performance of MixSeg in complex environments, all models were predicted with two sets of

TABLE 5 The results of segmentation accuracy of different models on two diseased leaf test sets.

| Model | Apple alternaria blotch | | | | Apple grey spot | | | |
|---|---|---|---|---|---|---|---|---|
| | IoU/% | | mPA | PA | IoU/% | | mPA | PA |
| | Leaf | Disease | | | Leaf | Disease | | |
| PSPNet | 95.47 | 71.80 | 93.50 | 98.56 | 94.69 | 64.84 | 91.91 | 98.46 |
| HRNetV2 | 95.12 | 79.23 | 94.06 | 98.45 | 95.75 | 76.85 | 93.92 | 98.77 |
| SegFormer | 96.80 | 82.22 | 95.89 | 99.00 | 96.60 | 78.94 | 94.93 | 99.03 |
| U-Net | 95.49 | 83.72 | 95.71 | 98.57 | 95.46 | 82.23 | 95.99 | 98.70 |
| DeepLabv3+ | 96.85 | 82.55 | 96.74 | 99.01 | 96.47 | 79.41 | 96.88 | 98.98 |
| MixSeg-T | **98.22** | **87.40** | **97.21** | **99.45** | **98.09** | **86.20** | **97.26** | **99.46** |

The bold font indicates which model performs best on a particular evaluation metric.

TABLE 6 The results of performance metrics for different models.

| Model | Inference time/ms | FPS | Total parameters/M | FLOPs/G |
|---|---|---|---|---|
| PSPNet | 39.67 | 25.20 | 49.07 | 61.63 |
| HRNetV2 | 29.57 | 33.81 | 9.64 | 18.66 |
| SegFormer | 10.54 | 94.80 | 3.71 | 6.77 |
| U-Net | 44.76 | 22.33 | 24.89 | 225.85 |
| DeepLabv3+ | **7.96** | **125.52** | 5.81 | 26.44 |
| MixSeg-T | 8.70 | 114.82 | **1.90** | **4.17** |

The bold font indicates which model performs best on a particular evaluation metric.

outdoor apple disease leaf test images. We adjusted the transparency of the predicted result image to 50% and then fused it with the original image to compare the segmentation effect of each method more clearly.

Figures 6 and 7 demonstrated the segmentation results of each model under the test set of apple alternaria blotch leaf images. Compared Figure 6, PSPNet was less effective in segmenting outdoor leaves and was easily disturbed by other leaves. Moreover, PSPNet missed many small disease spots. This poor performance can also be seen in Figure 7. HRNetV2 did not segment both leaves and disease spots well. The leaf area extracted by HRNetV2 is incomplete, and small spots were missed. SegFormer was less effective in outdoor leaf image segmentation, and there was a loss of tiny spots. DeepLabv3 + was not accurate enough in extracting leaf regions and misses small spots. Although the segmentation of U-Net was better and superior to the previously mentioned methods, it also suffered from the interference of overlapping leaves and misses some tiny spots. We can see that both MixSeg-T and MixSeg-L have better segmentation results than U-Net. They can extract the target leaves clearly from the complex environment and the tiny spots missed by other methods.

Figures 8 and 9 demonstrated the segmentation results of each model under the test set of apple grey spot diseased leaf images. Compared Figures 8, 9, PSPNet can usually segment the whole target leaf, but it was vulnerable to other environmental factors. Therefore the target leaf area extracted by PSPNet was not complete enough to segment the serrated shape of the leaf edge. Moreover, the color of the disease spots at the leaf tip was similar to some elements in the background, and PSPNet cannot extract these spots. The leaf areas extracted by HRNetV2 were somewhat more complete than those of PSPNet. However, HRNetV2 also cannot segment the serrated shape of the leaves. Moreover, HRNetV2 also missed the disease spots at the leaf tips. SegFormer had poor segmentation performance on leaves, which were disturbed by other leaves. In addition, SegFormer could not accurately extract the diseased areas at the leaf tips. DeepLabv3+ was also less effective in segmenting the spots at the leaf tips. The segmentation of U-Net was better and superior to the above methods, but not as good as MixSeg-T and MixSeg-L. Although U-Net extracts the spots at the leaf tips, the area of the disease spots was not accurate enough. In contrast, MixSeg-T and MixSeg-L can extract the area of leaf tip spots more accurately. In addition, MixSeg-T and MixSeg-L can better segment the serrations on the leaf edges.

To more comprehensively compare the differences in segmentation performance between the different models, Figure 10 compared the segmentation results. From Figure 10, the comparison with MixSeg, other models had poor segmentation results due to the difference in feature extraction structure. For example, the pyramid pooling and spatial pyramid structure of PSPNet and DeepLabv3+, although expanding the sensory field, tended to miss small spots in the downsampling. SegFormer, although used self-attention for global

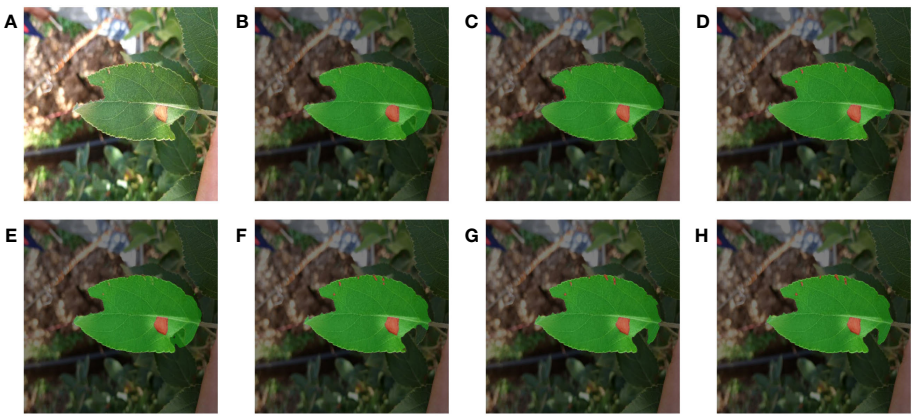



FIGURE 6
Fusion segmentation results of different models for apple alternaria blotch leaf with breakage. **(A)** Image. **(B)** PSPNet. **(C)** HRNetV2. **(D)** SegFormer. **(E)** DeepLabv3+. **(F)** U-Net. **(G)** MixSeg-T. **(H)** MixSeg-L.

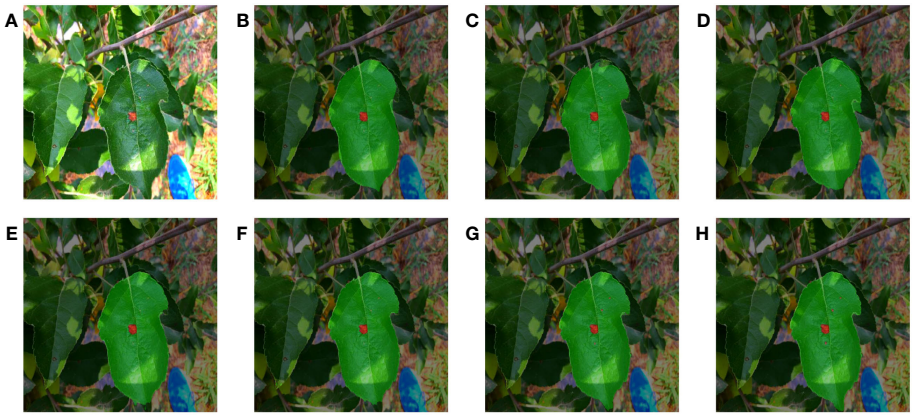



FIGURE 7
The results of fusion segmentation of different models on overlapping apple alternaria blotch leaves. **(A)** Image. **(B)** PSPNet. **(C)** HRNetV2. **(D)** SegFormer. **(E)** DeepLabv3+. **(F)** U-Net. **(G)** MixSeg-T. **(H)** MixSeg-L.

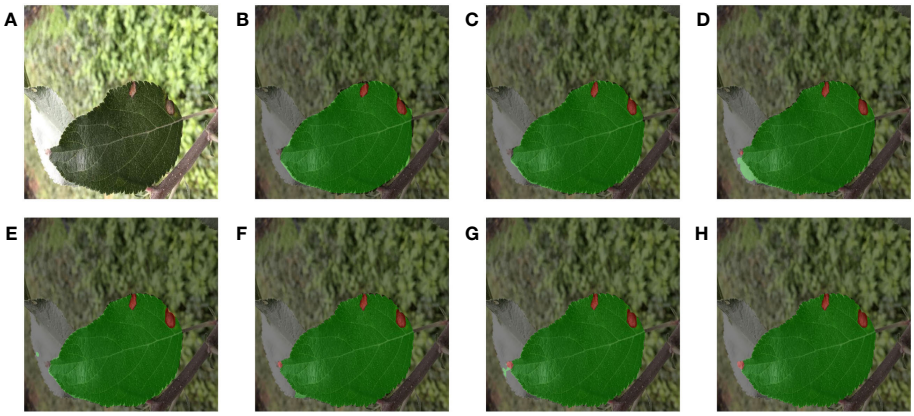



FIGURE 8
The results of different models for fusion segmentation of apple gray spot leaves with lesions present at the leaf tip. **(B)** PSPNet. **(C)** HRNetV2. **(D)** SegFormer. **(E)** DeepLabv3+. **(F)** U-Net. **(G)** MixSeg-T. **(H)** MixSeg-L.

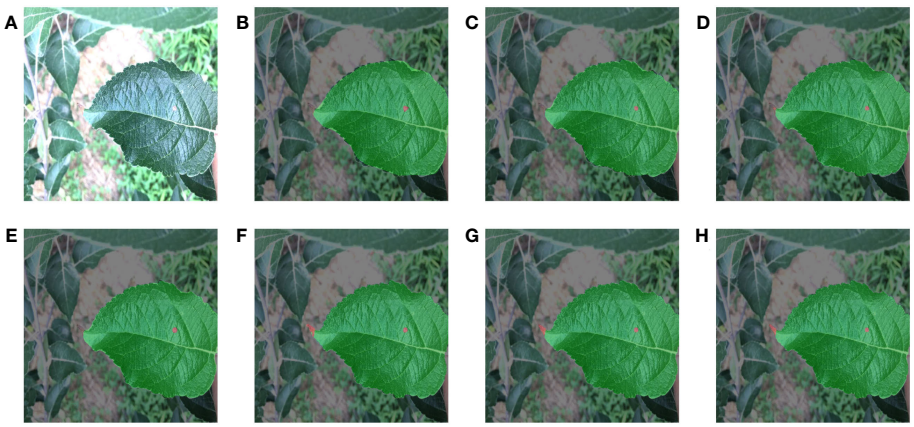



FIGURE 9
The results of fusion segmentation of apple gray spot leaves by different models in overilluminated environments. **(B)** PSPNet. **(C)** HRNetV2. **(D)** SegFormer. **(E)** DeepLabv3+. **(F)** U-Net. **(G)** MixSeg-T. **(H)** MixSeg-L.

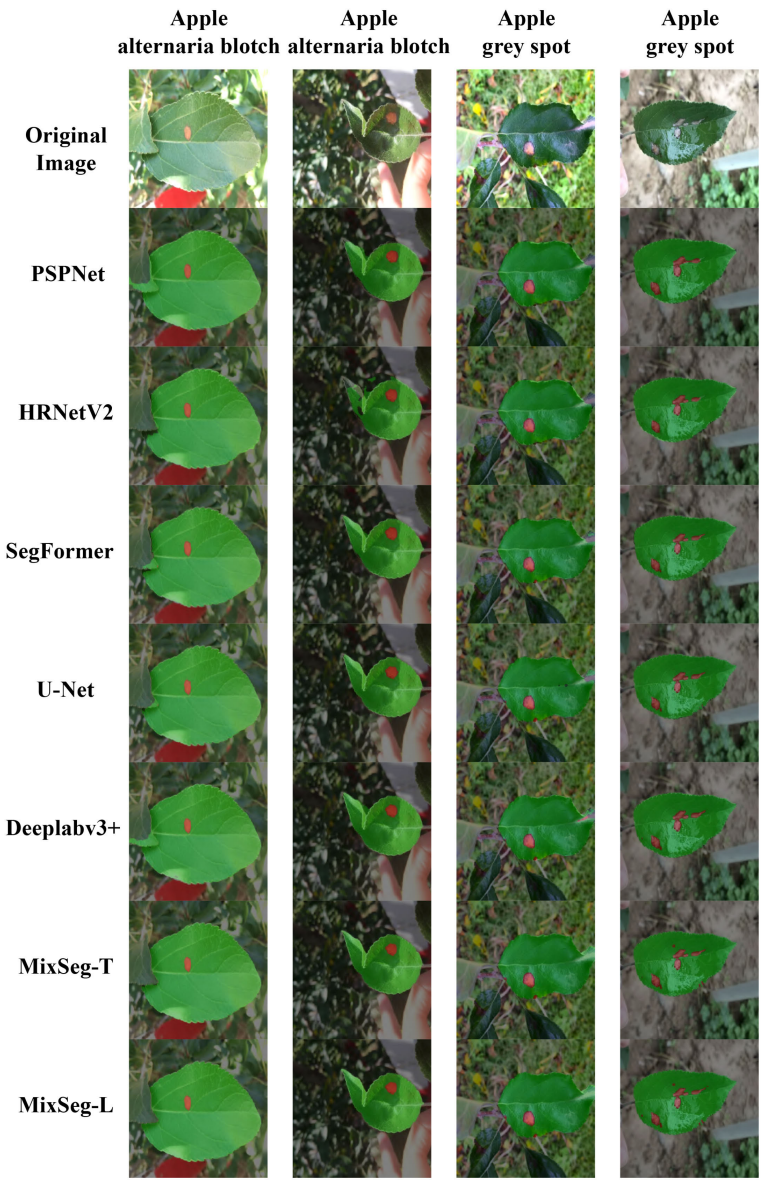



FIGURE 10
Fusion segmentation results of different models for outdoor apple alternaria blotch and grey spot leaf image datasets.

modeling, cannot encode the absolute position it cannot accurately recover the position information of leaves and spots in the upsampling. This led to poor segmentation results. Through a comprehensive comparison, MixSeg outperformed other methods in terms of segmentation effect.

## 3.6 Effective focus on regional analysis

To further demonstrated the effectiveness of the MixSeg architecture, we outputted the heat map of effective attention regions for DeepLabv3+ and MixSeg classification heads used the Grad-CAM technique separately and compared them. DeepLabv3+ was selected because it can be found through various experiments that DeepLabv3+ had the best overall performance except for MixSeg. Grad-CAM explained the parts of the network that are of concern in the decision-making process, and these regions of concern played an essential role in the final prediction. Regions with brighter colors or higher intensities in the heat map were considered more important for the network's decision-making. In contrast, regions with darker or lower intensities were relatively less critical. Based on this, we can better measure the segmentation performance of the network. Figure 11 showed the comparison results of DeepLabv3+ and MixSeg effective attention regions.

Observed the second and third rows of Figure 11, the effective area of focus of MixSeg can accurately focus on the target leaves. However, DeepLabv3+ was not focused enough on the target leaf because it was vulnerable to environmental distracting factors. In addition, through comparing with DeepLabv3+, MixSeg can notice the serrated shape on the leaf edge so that it can obtain better leaf

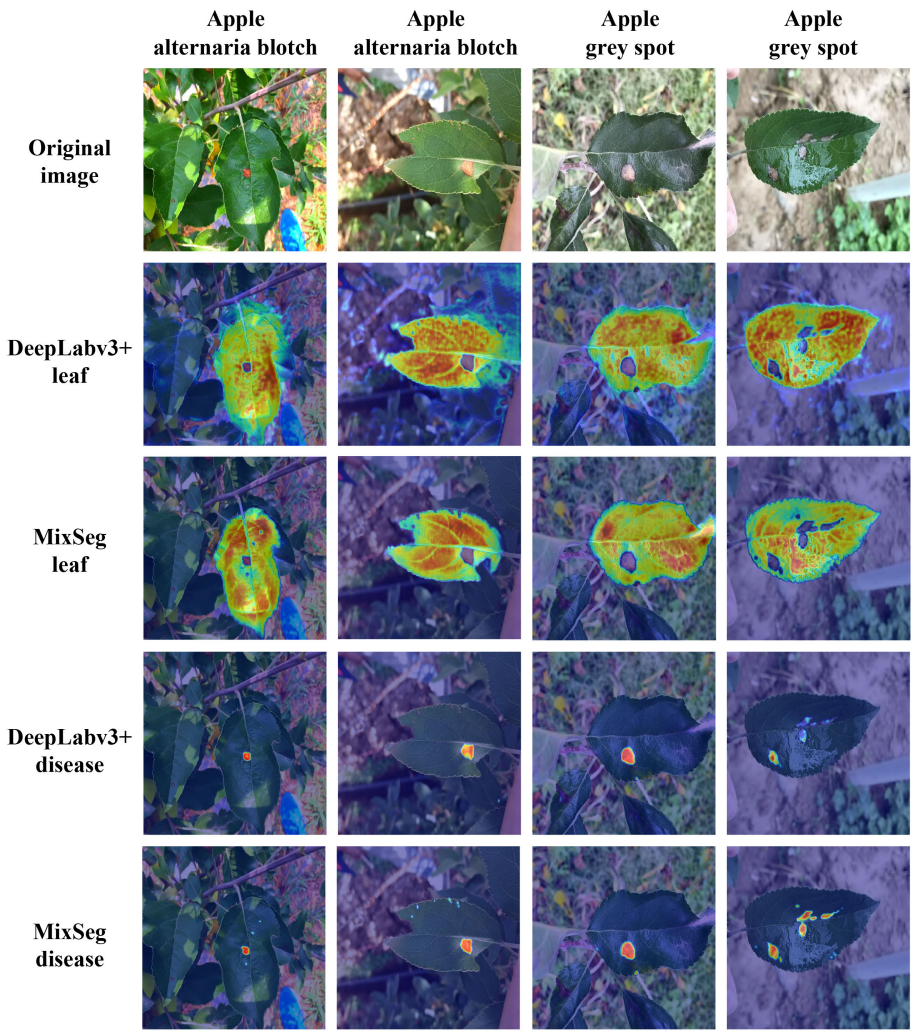



FIGURE 11
The results of DeepLabv3+ and MixSeg for effective focus areas on outdoor leaves and diseases. Areas with brighter colors or higher intensity in the heat map are considered more important for network decision-making, while areas with darker colors or lower intensity are relatively less important. These focus areas play an important role in the final prediction of the network.

segmentation results. Compared with the fourth and fifth rows of Figure 11, the color of the spots was similar to the color of some elements in the background, and DeepLabv3+ was less resistant to interference and therefore misses the attention to some spots. In contrast, MixSeg can focus on the spots missed by DeepLabv3+. Moreover, MixSeg was able to give more substantial attention to the spots. Therefore, MixSeg had a better segmentation effect on the spots in the final prediction. In summary, MixSeg was more resistant to interference and had a more accurate region of attention, which made MixSeg have better segmentation performance.

## 4 Discussion

The MixSeg model proposed in this paper introduced a hybrid structural network model based on CNN, Transformer and MLP for apple leaf and disease segmentation. By comparing with the latest segmentation models, it can be found that MixSeg was the most lightweight with the best segmentation accuracy and segmentation effect. In addition, the segmentation performance of MixSeg with different sizes was verified by ablation experiments. The ability of MixSeg to accurately extract leaves and diseases in complex environments was further demonstrated by analyzing and comparing the effective regions of focus of MixSeg and DeepLabv3+.

In the current work, the feasibility of MixSeg applied to segment apple leaf diseases in real scenarios was deeply experimented and discussed. MixSeg was designed to be lightweight for the requirements of deployment on mobile devices in real scenarios. In addition, MixSeg was resistant to interference and had strong feature representation capability and flexibility to capture and learn the features of target leaves and diseases in complex environments. Although the performance of MixSeg had only been validated on the task of apple leaf spot segmentation in complex environments in this study, the core principles and technical framework were somewhat generalizable and therefore very likely to be applied to other crop leaves. In the subsequent work, we will further add

different varieties of crop disease leaf datasets and continue to train MixSeg in order to improve its generalization ability in practical applications. However, there may be some limitations when applying MixSeg to other crop leaf categories. For instance, different kinds of leaves may have a varying morphology, texture and disease characteristics, which may limit the performance of MixSeg in some cases. Therefore, in future research, when extending MixSeg to other crop leaves, further research and understanding of the characteristics of the target crop leaves and diseases are needed to optimize and adjust accordingly to ensure the accuracy and robustness of the method.

## 5 Conclusion

This work presented a mixed-structure semantic segmentation method, MixSeg, for fast and accurate segmentation of apple leaves and diseases in complex environments. In this model, Mix Transformer was designed to inject multi-scale local biases into the model at a much smaller computational cost and establishes global dependencies between pairs of pixels. The PI was proposed, which was independent of the input length and thus allowed the model to perform multi-scale modeling while extracting learnable absolute position information. LOM was proposed to optimize the effect of the model on local feature extraction. In addition, the advantages of the mix structure model were demonstrated by comparing various experiments of MixSeg with mainstream single-structure semantic segmentation models. The experimental results showed that MixSeg was the most effective in segmenting apple diseased leaf images in complex environments, which made this study a key attempt to advance the research on smart agriculture. However, only apple diseased leaves were selected as experimental objects in the study, and the generalization ability of the model needed to be further improved. In future research, we will increase the variety of datasets used for training and further optimize the model according to the characteristics of different crop diseased leaves and the complex environments in which they grow, so that the mix structure model can be more widely applied to the field of smart agriculture.

## Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

## Author contributions

BL and JL contributed to conception and design of the study. JL, XX, and YJ contributed to the preparation of the equipment and the acquisition of the data. JL wrote the code and tested the method. JL and XX carried out the experiments and verified the methods. JL wrote the manuscript. BL were responsible for checking and revising the manuscript. All authors contributed to the article and approved the submitted version.

## Funding

This work was supported by the National Natural Science Foundation of China (Grant 41773024), Modification mechanism of hydrocarbon reservoirs by Neoproterozoic bioturbation in the northern South China Sea (Grant 42272178) and Henan Province key research and development and promotion special projects (Grant 222102210131).

## Acknowledgments

We are very grateful to all the authors for their support and contribution with the manuscript.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

Agarwal, M., Gupta, S. K., and Biswas, K. K. (2021). "Plant leaf disease segmentation using compressed unet architecture," in *Trends and Applications in Knowledge Discovery and Data Mining*. Eds. M. Gupta and G. Ramakrishnan (Cham: Springer International Publishing), 9–14.

Aggarwal, M., Khullar, V., and Goyal, N. (2023). "Exploring classification of rice leaf diseases using machine learning and deep learning," in *2023 3rd International Conference on Innovative Practices in Technology and Management (ICIPTM)* (Uttar Pradesh, India: IEEE). 1–6. doi: 10.1109/ICIPTM57143.2023.10117854

Bansal, P., Kumar, R., and Kumar, S. (2021). Disease detection in apple leaves using deep convolutional neural network. *Agriculture* 11, 617. doi: 10.3390/agriculture11070617

Barman, U., Choudhury, R. D., Sahu, D., and Barman, G. G. (2020). Comparison of convolution neural networks for smartphone image based real time classification of citrus leaf disease. *Comput. Electron. Agric.* 177, 105661. doi: 10.1016/j.compag.2020.105661

Chen, R., Qi, H., Liang, Y., and Yang, M. (2022). Identification of plant leaf diseases by deep learning based on channel attention and channel pruning. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.1023515

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). "Encoder-decoder with atrous separable convolution for semantic image segmentation," in

*Proceedings of the European conference on computer vision (ECCV)*. (Cham: Springer) 801–818.

Ding, X., Xia, C., Zhang, X., Chu, X., Han, J., and Ding, G. (2021). Repmlp: Re-parameterizing convolutions into fully-connected layers for image recognition. *arXiv*. doi: 10.48550/arXiv.2105.01883

Divyanth, L., Ahmad, A., and Saraswat, D. (2023). A two-stage deep-learning based segmentation model for crop disease quantification based on corn field imagery. *Smart Agric. Technol.* 3, 100108. doi: 10.1016/j.atech.2022.100108

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv*. doi: 10.48550/arXiv.2010.11929

Guo, M.-H., Liu, Z.-N., Mu, T.-J., and Hu, S.-M. (2022). Beyond self-attention: External attention using two linear layers for visual tasks. *IEEE Trans. Pattern Anal. Mach. Intell* 45, 5436–5447. doi: 10.1109/TPAMI.2022.3211006

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 1904–1916. doi: 10.1109/TPAMI.2015.2389824

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. (Las Vegas, NV, USA: IEEE) 770–778.

Hu, J., Shen, L., and Sun, G. (2018). "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. (Radarweg 29: Elsevier) 7132–7141.

Hu, H., Zhang, Z., Xie, Z., and Lin, S. (2019). "Local relation networks for image recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. (Seoul, Korea (South): IEEE) 3464–3473.

Huang, L., Yuan, Y., Guo, J., Zhang, C., Chen, X., and Wang, J. (2019). Interlaced sparse self-attention for semantic segmentation. *arXiv*. doi: 10.48550/arXiv.1907.12273

Islam, M. A., Jia, S., and Bruce, N. D. (2020). How much position information do convolutional neural networks encode? *arXiv*. doi: 10.48550/arXiv.2001.08248

Kuo, C.-C. J. (2016). Understanding convolutional neural networks with a mathematical model. *J. Visual Communication Image Represent.* 41, 406–413. doi: 10.1016/j.jvcir.2016.11.003

Lee, H., Hwang, S. J., and Shin, J. (2019). *Rethinking data augmentation: Self-supervision and selfdistillation*.

Mu, H., Wang, K., Yang, X., Xu, W., Liu, X., Ritsema, C. J., et al. (2022). Pesticide usage practices and the exposure risk to pollinators: A case study in the north China plain. *Ecotoxicol. Environ. Saf.* 241, 113713. doi: 10.1016/j.ecoenv.2022.113713

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., and Shlens, J. (2019). Stand-alone self-attention in vision models. *Adv. Neural Inf. Process. Syst.* 32. doi: 10.48550/arXiv.1906.05909

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference*, [Munich, Germany, October 5-9, 2015. (Cham: Springer)], 234–241.

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv*. doi: 10.48550/arXiv.1409.1556

Storey, G., Meng, Q., and Li, B. (2022). Leaf disease segmentation and detection in apple orchards for precise smart spraying in sustainable agriculture. *Sustainability* 14, 1458. doi: 10.3390/su14031458

Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., et al. (2019). High-resolution representations for labeling pixels and regions. *arXiv*. doi: 10.48550/arXiv.1904.04514

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Adv. Neural Inf. Process. Syst.* 30. doi: 10.48550/arXiv.1706.03762

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. (2020). Linformer: Self-attention with linear complexity. *arXiv*. doi: 10.48550/arXiv.2006.04768

Wu, Z., Yang, R., Gao, F., Wang, W., Fu, L., and Li, R. (2021). Segmentation of abnormal leaves of hydroponic lettuce based on deeplabv3+ for robotic sorting. *Comput. Electron. Agric.* 190, 106443. doi: 10.1016/j.compag.2021.106443

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. (2021a). Segformer: Simple and efficient design for semantic segmentation with transformers. *Adv. Neural Inf. Process. Syst.* 34, 12077–12090.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. (2021b). Segformer: Simple and efficient design for semantic segmentation with transformers. *Adv. Neural Inf. Process. Syst.* 34, 12077–12090.

Xu, L., Cao, B., Zhao, F., Ning, S., Xu, P., Zhang, W., et al. (2023). Wheat leaf disease identification based on deep learning algorithms. *Physiol. Mol. Plant Pathol.* 123, 101940. doi: 10.1016/j.pmpp.2022.101940

Yogeshwari, M., and Thailambal, G. (2021). Automatic feature extraction and detection of plant leaf disease using glcm features and convolutional neural networks. *Mater. Today: Proc.* doi: 10.1016/j.matpr.2021.03.700

Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., et al. (2022). "Metaformer is actually what you need for vision," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. (New Orleans, LA, USA: IEEE) 10819–10829.

Yuan, H., Zhu, J., Wang, Q., Cheng, M., and Cai, Z. (2022). An improved deeplab v3+ deep learning network applied to the segmentation of grape leaf black rot spots. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.795410

Zeiler, M. D., and Fergus, R. (2014). "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014: 13th European Conference*, Zurich, Switzerland, September 6-12, 2014. (Cham: Springer) 818–833.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. (Honolulu, HI, USA: IEEE) 2881–2890.

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., et al. (2021). "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. Nashville, TN, USA: IEEE 6881–6890.

Zhuang, C., Zhai, A. L., and Yamins, D. (2019). "Local aggregation for unsupervised learning of visual embeddings," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. (Seoul, Korea (South): IEEE), 6002–6012.

frontiers | Frontiers in Plant Science

TYPE Perspective
PUBLISHED 22 September 2023
DOI 10.3389/fpls.2023.1225409

Check for updates

OPEN ACCESS

EDITED BY
Huiling Chen,
Wenzhou University, China

REVIEWED BY
Jiaogen Zhou,
Huaiyin Normal University, China
Jun Liu,
Weifang University of Science and Technology, China

*CORRESPONDENCE
Jucheng Yang
jcyang@tust.edu.cn
Sook Yoon
syoon@mokpo.ac.kr

†These authors have contributed equally to this work

RECEIVED 19 May 2023
ACCEPTED 30 August 2023
PUBLISHED 22 September 2023

CITATION
Xu M, Kim H, Yang J, Fuentes A, Meng Y, Yoon S, Kim T and Park DS (2023) Embracing limited and imperfect training datasets: opportunities and challenges in plant disease recognition using deep learning.
*Front. Plant Sci.* 14:1225409.
doi: 10.3389/fpls.2023.1225409

COPYRIGHT
© 2023 Xu, Kim, Yang, Fuentes, Meng, Yoon, Kim and Park. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# Embracing limited and imperfect training datasets: opportunities and challenges in plant disease recognition using deep learning

Mingle Xu[1†], Hyongsuk Kim[1†], Jucheng Yang[2*], Alvaro Fuentes[1], Yao Meng[1], Sook Yoon[3*], Taehyun Kim[4] and Dong Sun Park[1]

[1]Department of Electronic Engineering, Core Research Institute of Intelligent Robots, Jeonbuk National University, Jeonju, Republic of Korea, [2]College of Artificial Intelligence, Tianjin University of Science and Technology, Tianjin, China, [3]Department of Computer Engineering, Mokpo National University, Muan, Republic of Korea, [4]National Institute of Agricultural Sciences, Wanju, Republic of Korea

Recent advancements in deep learning have brought significant improvements to plant disease recognition. However, achieving satisfactory performance often requires high-quality training datasets, which are challenging and expensive to collect. Consequently, the practical application of current deep learning–based methods in real-world scenarios is hindered by the scarcity of high-quality datasets. In this paper, we argue that embracing poor datasets is viable and aims to explicitly define the challenges associated with using these datasets. To delve into this topic, we analyze the characteristics of high-quality datasets, namely, large-scale images and desired annotation, and contrast them with the *limited* and *imperfect* nature of poor datasets. Challenges arise when the training datasets deviate from these characteristics. To provide a comprehensive understanding, we propose a novel and informative taxonomy that categorizes these challenges. Furthermore, we offer a brief overview of existing studies and approaches that address these challenges. We point out that our paper sheds light on the importance of embracing poor datasets, enhances the understanding of the associated challenges, and contributes to the ambitious objective of deploying deep learning in real-world applications. To facilitate the progress, we finally describe several outstanding questions and point out potential future directions. Although our primary focus is on plant disease recognition, we emphasize that the principles of embracing and analyzing poor datasets are applicable to a wider range of domains, including agriculture. Our project is public available at https://github.com/xml94/EmbracingLimitedImperfectTrainingDatasets.

KEYWORDS

**plant disease recognition, AI in Agriculture, deep learning in agriculture, smart agriculture, precision agriculture**

# 1 Introduction

Plant diseases are responsible for significant yield losses (Savary et al., 2019), making their recognition a crucial task in crop cultivation. In the past decade, deep learning, characterized by two essential attributes inherited from classical machine learning methods (Kawasaki et al., 2015; Mohanty et al., 2016; Fuentes et al., 2017), has emerged as a promising approach for this purpose. First, deep learning possesses the remarkable ability to serve as a feature extractor (Singh et al., 2018; Bengio et al., 2021). This stands in contrast to traditional machine learning, which often necessitates human experts to manually design features, such as histograms of oriented gradients for Red-Green-Blue (RGB) images (Fan et al., 2022) and vegetation indices for hyperspectral and multispectral images (Abdulridha et al., 2020). However, designing effective features has proven challenging and often requires diversity for different tasks. Second, deep learning–based methods have demonstrated "decent performance" in numerous studies on plant disease recognition (Singh et al., 2018; Boulent et al., 2019; Abade et al., 2021; Liu and Wang, 2021; Ouhami et al., 2021; Singh et al., 2021; Thakur et al., 2022). Furthermore, the implementation of deep learning on farms offers the enticing advantage of liberating human labor and significantly reducing associated costs. This is particularly valuable in the present century, as the global population is expected to continue increasing while the number of agricultural workers has been steadily declining.

Although deep learning has demonstrated its potential, the requirement for high-quality datasets to achieve satisfactory performance remains a challenge. Unfortunately, collecting such datasets is often prohibitively expensive and extremely challenging in many real-world applications (Xu et al., 2022a; Xu et al., 2023). Conversely, poor datasets are prevalent, and current models may struggle when confronted with them. Recognizing this reality, we contend that embracing poor datasets presents new opportunities to advance plant disease recognition in real-world applications. To further enrich the relevant understanding of this embrace, we analyze the characteristics of the desired high-quality datasets: large-scale and desired annotation. Specifically, large-scale datasets provide a vast quantity of information within the images, whereas desired annotation ensures that the images are annotated in accordance with specific criteria and objectives. More details are discussed in Section 2. In contrast, poor-quality datasets are defined by their deviations from the characteristics. Specifically, a dataset not on a large scale is categorized as limited, whereas a dataset lacking the desired annotation is considered imperfect. Embracing poor datasets, therefore, entails embracing limited and imperfect dataset, each of which is further explored and analyzed in Sections 4 and 5, respectively. The challenges associated with embracing limited and imperfect datasets are explicitly defined within these sections. A novel taxonomy detailing these challenges is conceptually described in Section 3, and Table 1 provides a glimpse of the taxonomy.

This study distinguishes itself from existing survey papers on plant disease recognition using deep learning by adopting a "challenge-oriented" approach instead of a "technique-oriented" one. Whereas previous works such as those by Singh et al. (2018); Boulent et al. (2019); Abade et al. (2021); Liu and Wang (2021); Ouhami et al. (2021); Singh et al. (2021), and Thakur et al. (2022) have focused on summarizing existing techniques and relevant materials, we have identified the key challenges associated within this field. Specifically, we highlight the scarcity of large-scale annotated data and advocate for embracing the concept of limited and imperfect datasets when deploying deep learning in real-world applications.

To conclude, in pursuit of deploying deep learning for plant disease recognition in real-world applications with satisfactory performance, we offer a perspective that embraces limited and imperfect datasets, contrasting with high-quality data. Our main contributions are as follows.

- We *explicitly* argue embracing limited and imperfect datasets for plant disease recognition using deep learning,

TABLE 1 Taxonomy of challenges arising when embracing limited and imperfect datasets.

| Challenge | | | Definition |
|---|---|---|---|
| Limited dataset | Class-level | Few-shot | All classes have similar few annotated images, where trained models may get low performance for all classes. |
| | | Class imbalance | One class has many more annotated images than another class, where trained models may get high performance in the former class but suffer in the latter class. |
| | Dataset-level | Domain shift | The training and test datasets share the same label spaces but are in different distribution spaces, where trained models may get low test performance. |
| | | Unknown class | Unknown (new) classes exist in the test dataset, where trained models will consider the corresponding image into a known class and not distinguish the unknown from known classes. |
| Imperfect dataset | Incomplete annotation | | Training datasets have labeled and unlabeled images simultaneously, where utilizing the unlabeled images may contribute to the test performance. |
| | Inexact annotation | | Training datasets are given with only coarse-grained annotations, where utilizing these annotations is challenging to train models. |
| | Inaccurate annotation | | Some annotations may be inaccurate, where it is challenging to get decent test performance by utilizing these annotations to train models. |

A limited dataset suggests that the training dataset is not on a large scale, whereas an imperfect dataset implies that the annotations are not expected. In addition, the challenges of limited dataset encompass class-level challenges, which involve image variations among different classes within the training dataset, and dataset-level challenges, which pertain to the information gap between the training and test datasets.

motivated by the reality that collecting high-quality datasets is expensive and challenging.

- We analyze the underlying reasons behind the current necessity for high-quality datasets in Section 2.
- We present a *taxonomy* of challenges associated with the embrace in Section 3, with *formal definitions*. A concise overview of existing studies that tackle them is also given as discussed in Sections 4 and 5.
- We provide noteworthy questions and highlight potential directions for further exploration in Section 6.

# 2 Why is high-quality dataset desired?

In general, to achieve promising performance using deep learning models, the training datasets should have two characteristics: large-scale and annotated with desired strategies. This section aims to probe the underlying reasons behind the desired characteristics.

## 2.1 Large-scale dataset

### 2.1.1 Deep learning models require large-scale data

For two reasons, deep learning generally requires a large-scale training dataset to obtain a comparable test performance. First, there are enormous learnable parameters in deep learning–based models that require large-scale data (Krizhevsky et al., 2017; Sarker, 2021). This ensures that a better feature extractor could be learned; otherwise, the training data points could be remembered, resulting in a poor test performance (Xu et al., 2023). Second, the distribution of the training dataset is gradually approaching that of the test dataset when the training dataset becomes larger, supporting a better test performance. For example, a model trained with images captured in laboratories is not expected to be effective when tested with images captured on farms (Guth et al., 2023; Wu et al., 2023).

### 2.1.2 Huge image variation requires large-scale datasets

The requirement for a large-scale training dataset comes from not only deep learning but also the task of plant disease recognition, called image variation (Xu, 2023; Xu et al., 2023). Considering the types of plant diseases, it can be divided into intra-class, differences within the same plant disease, and inter-class, heterogeneity between the two plant diseases. The intra-class image variation, partially illustrated in Figure 1, originates from three main elements. The first one is "plant itself". For example, some plants may have different types, such as different types of tomatoes, with diverse leaf shapes and sizes, as shown in Figure 1A. The same type of plant and plant disease may also occur at various stages with individual visual patterns. For example, Figures 1B–D have shown the different stages of plant disease, flowers, and leaves. Second, the plants may be grown in different "environments", such as fields and greenhouses. Heterogeneous illuminations and backgrounds in the fields are shown in Figures 1E, F, respectively. Third, "imaging processing" is another source of intra-class image variation. Arguably, optical sensors and platforms result in greater diversity, such as RGB and thermal images, phones, and satellites (Oerke et al., 2014; Mahlein, 2016). When the optical sensors and platforms are fixed, the distance between the plants and sensors results in a multiscale challenge, as shown in Figure 1G. In addition, these viewpoints also lead to variations, as shown in Figure 1H.

The inter-class variation assumes that one type of plant disease is visually different from another, such as tomato leaf mold and canker, as shown in Figures 2A, C. We emphasize that this assumption should be considered when formulating the application objectives. For example, early symptoms in RGB images of plant diseases really resemble healthy ones, and, consequently, finding related plant diseases at very early stages may not be reliable. In addition, inter-class image variation can be viewed as a relative challenge. Specifically, the visual differences between the two plant diseases could be larger than those of the counterparts of another pair. As shown in Figure 2, the visual deviations between tomato leaf mold and canker are larger than those between tomato magnesium deficiency and chlorosis virus. A strategy used for this scenario is to collect more data for the close pairs, in which the models required more evidence to make decisions.

## 2.2 Desired annotation strategy

Deep learning is first trained in a "training dataset" and then tested in a "test dataset". A "validation dataset" is usually utilized to select the best-trained model from different hyperparameters and other training settings. The training and validation datasets are accessible at both the training and test stages, whereas the test dataset is only accessible at the test stage. Furthermore, the training and validation datasets for most deep learning–based models are hypothesized to be annotated in the desired manner. A desired annotation strategy, called EEP, has three primary points: exclusion, extensiveness, and precision. This has suggested that every annotation included only one specific visual pattern of plant disease, whereas extensive indicated that every plant disease in the images should have been annotated. The last one requires a precise annotation. For example, in image classification, every image should have included a type of plant disease and be linked to a label, as illustrated in the first row of Figure 3. By contrast, as shown in the second row of Figure 1, object detection generalizes the idea that one image can cover multiple plant diseases. However, every region should be annotated with labels and locations (a bounding box with four values: two for the left point in the horizontal and vertical directions, and two for width and height).

Accordingly, segmentation is a task for recognizing plant disease, in which every pixel should be assigned a label, as suggested in the last row of Figure 3.

# 3 Challenge formulation with limited and imperfect datasets

As previously discussed, achieving satisfactory performance using deep learning often requires training datasets that are both annotated as desired and of large scale (Lu et al., 2022). However,

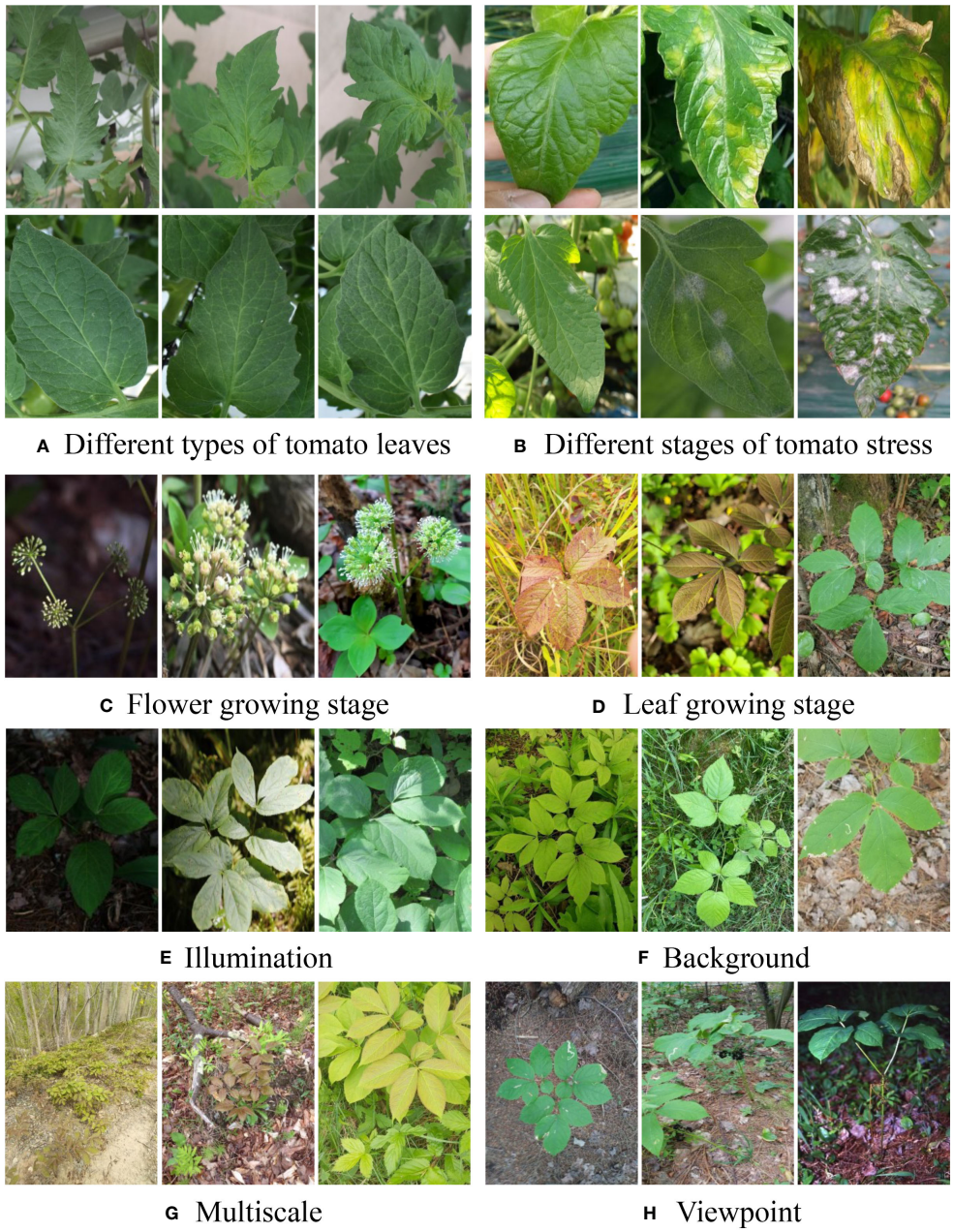



FIGURE 1
Illustration of the intra-class image variation. The images in **(A, B)** are tomato leaves. The images in **(C–H)** stem from a species, Aralia nudicaulis, in the PlantCLEF2022 dataset (Goëau et al., 2022). Every group suggests that the pictures belonging to the same plant disease may have visual diversities.

collecting such datasets is frequently challenging, time-consuming, and costly. Consequently, existing models may encounter limitations when applied to real-world scenarios without access to high-quality datasets. Therefore, a more convincing objective is to secure the satisfactory performance of models using limited and imperfect training datasets. However, this concept remains relatively unexplored within the context of plant disease recognition. The present study aims to shed light on this direction, with the ultimate goal of monitoring plant growth, thereby reducing human intervention and potentially mitigating the issues arising from plant diseases.

To comprehend the challenges that arise when dealing with limited and imperfect training datasets, we propose a novel taxonomy. Specifically, the term "limited dataset" refers to scenarios where the training dataset is not on a large scale, whereas "imperfect dataset" describes situations where the annotations of the training dataset deviate from the expected and desired. The limited dataset can be further divided into two subcategories: class-level, which examines deviations among different classes within the training datasets, and dataset-level, which analyzes the heterogeneity between the training and test datasets. On the other hand, imperfect dataset can be classified into

10.3389/fpls.2023.1225409

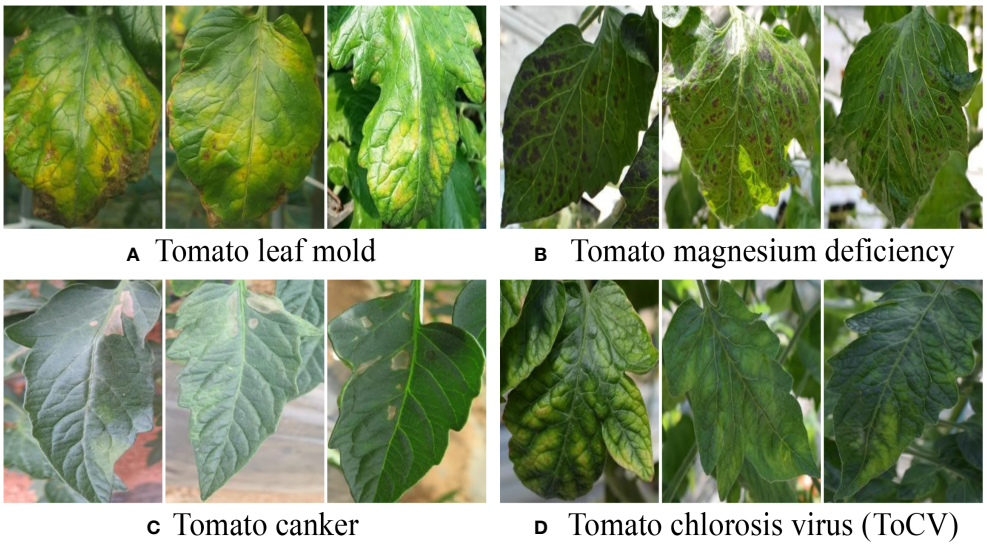



FIGURE 2
Illustration of inter-class image variation. It can be cast to a relative challenge where the visual deviations between **(A, C)** are larger than that between **(B, D)**. A corresponding strategy is to collect more data for **(B, D)** in that models need more evidence to make decisions for hard scenarios.

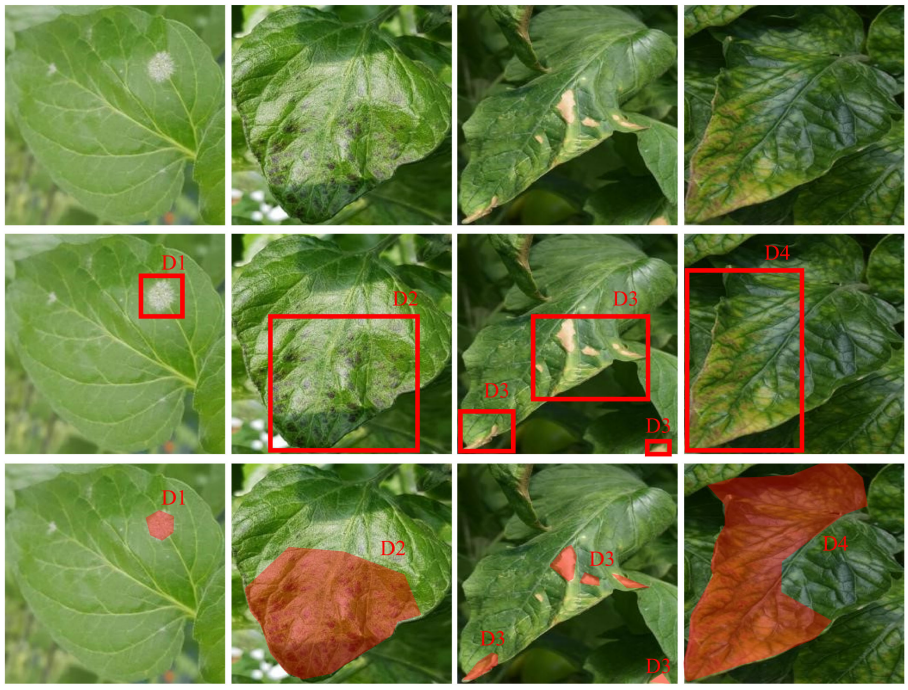



FIGURE 3
Annotation strategies in three primary tasks of plant disease recognition. From the first to the last rows are image classification, object detection, and segmentation, respectively. *Dm* (*m* = 1, 2, 3, 4) denote the types of plant disease, and every column suggests different plant diseases. In the simplest way, image classification refers to assigning a class to one image, whereas, in object detection, classes and their locations (bounding box) are entailed to predict. Segmentation requires class prediction at a pixel level. From the first to the last row, the annotation becomes more complicated and thus more time-consuming. The images in real-world applications tend to be more complex than these examples, such as multiple diseases existing in one leaf and one image including multiple leaves. Furthermore, a desired annotation strategy embraces three primary points: exclusive, extensive, and precise. The exclusive suggests that every annotation just includes one specific visual plant disease pattern, whereas the extensive denotes that every plant disease in the images should be annotated. The precise requires that the images should be annotated precisely. Violating the three points leads to the challenges of imperfect annotation.

three distinct types based on the nature of conflict: incomplete annotation, where a portion of images lacks annotations; inexact annotation, where some classes are annotated in a coarse-grained manner; and inaccurate annotation, where certain images are annotated with inaccuracies or even incorrect labels. Table 1 offers a glimpse into this comprehensive taxonomy.

# 4 Limited dataset

To make the following content easy to understand, several notations are first described. $D_X$ and $D_Y$ denote the training and test datasets, $X$ and $Y$ denoting the training and test domains. In general, the two datasets encompass $n$ plant disease classes: $c^1,c^2,\ldots,c^N$. Let $n_{Xi}$ ($i$ = 1,2,…,$N$) and $n_{Yj}$ ($j$ = 1,2,…,$N$) denote the numbers of annotated images for class $c^i$ in $D_X$ and class $c^j$ in $D_Y$, respectively.

## 4.1 Class-level limited dataset

The class-level limited dataset, confined to the training dataset, is a case in which the number of annotated images for a class is small. Considering the differences across the different classes, they are divided into few-shot and class imbalances.

### 4.1.1 Few-shot

The few-shot challenge assumes that collecting and annotating images are expensive for *every* plant disease with the same number of annotated images. Formally, this challenge is *strictly* defined as $n_{Xi}= M$, where $M \in R^+$ is a small natural number. In general, $M$ may be equal to 5 or 10, which suggests that every plant disease has only 5 or 10 annotated images. Moreover, the few-shot challenge could be *generalized* as

$$n_{X_i} \approx M \tag{1}$$

where every class contain *approximately M* annotated images. The essential issue is that a few annotated images could not provide sufficient evidence to train a deep learning–based model; thus, the trained model could not be generalized in the test dataset for every class. On the basis of this, we further extend the few-shot challenge from a small number of annotated images to a larger case, such as 100 and even 500, where *most deep learning–based models could not obtain good test performance for every plant disease*. This motivation is based on the observation that plant disease may have huge intra-class image variation and relatively low inter-class image variation.

To address this few-shot challenge, image manipulation, a set of traditional image processing methods, such as translation and flipping, is one of the simplest methods. It is hypothesized to retain class information and mimic image variations. For example, image translation changes the positions of objects in an image. This method is utilized to increase the number of training images from 350 to 39,010 for six plant diseases and healthy leaves (Gorad and Kotrappa, 2021). In particular, a new background is fused to the object of plant disease to create diverse backgrounds in the field rather than in the laboratory (Gui et al., 2021). Owing to its simple deployment, image manipulation is leveraged by default with many other advanced methods in the general few-shot cases (Mohanty et al., 2016; Xu et al., 2022c). In addition, the image-generating models provide opportunities. Conditional generative adversarial networks (Mirza and Osindero, 2014) can generate new images, and, given a label, the generated images are assumed to be similar to the original images, but not the same (Abbas et al., 2021). Intuitively, image-generating models aim to learn image variations in the original training dataset and then create new image variations. However, learning an image-generating model requires data, which is often not satisfactory in the few-shot challenge.

Another effective and efficient method is transfer learning, which transfers knowledge for plant disease recognition from another task with a large-scale training dataset, assuming that learning one task with a large-scale training dataset is beneficial to plant disease recognition (Zhuang et al., 2020). In general, the plant disease recognition dataset is called the target dataset and the task is called the target task, whereas another one is termed as source dataset and task, respectively. With this strategy, the first issue is to choose a better model from many deep learning–based models (Mohanty et al., 2016; Kaya et al., 2019; Chen et al., 2020), such as ResNet (He et al., 2016) and DenseNet (Huang et al., 2017). The way to adopt a given pre-trained model is a question, such as freezing most of the models and training the remaining part (Sethy et al., 2020). In addition to models, choosing a better source dataset is another essential issue. A seminal work directly utilized a generic computer vision dataset, ImageNet (Deng et al., 2009). Although it has many variations, it is not that kindly related to plant disease recognition. Simultaneously, a source dataset related to the target dataset is more appealing (Neyshabur et al., 2020; Matsoukas et al., 2022). In this manner, plant-related datasets are considered, such as the AIChallenger2018 (Zhao et al., 2022) and PlantCLEF2017 (Kim et al., 2021) datasets. Furthermore, the image variations inside the source dataset should also be considered. For example, PlantCLEF2017 has more annotated images and higher intra-class variations than AIChallenge2018 where most images are collected in the laboratory. Considering the model and source dataset together is therefore encouraging. Embracing this idea, PlantCLEF2022 (Goëau et al., 2022; Xu et al., 2022b), a large-scale plant-relevant dataset with 2,885,052 annotated images for 80,000 classes, is leveraged with a ViT-based (Dosovitskiy et al., 2020) model rather than convolution neural network (CNN), to achieve versatile plant disease recognition (Xu et al., 2022c). With this strategy, a mean test accuracy of 86.29% over 12 datasets of plant disease recognition in a 20-shot case is achieved with a fast convergence speed, which is 12.76% higher than the current state-of-the-art accuracy of 73.53%.

Although transfer learning has been widely adopted, we argue that more opportunities still exist. First, transfer learning involves more segments beyond the models and source datasets, such as the loss function to pre-train and re-train the model. Considering these segments may provide motivation for improving the performance of plant disease recognition. We argue that the new methods in the general field of computer vision are useful for plant disease recognition. For example, a model could be trained using a source dataset with not only annotated images but also paired text to obtain improved semantic representations (Radford et al., 2021;

Wei et al., 2022). The utilization of transfer learning in practical applications is another issue. For example, current deep learning–based models have numerous parameters and thus should be compressed for embedding systems. In this case, transferring knowledge from a large to a small model is desirable (Abbasi Koohpayegani et al., 2020).

Other possible strategies, such as meta-learning (Huisman et al., 2021) and metric learning (Kaya and Bilge, 2019), have received attention over the last few years. Generic deep learning directly outputs the final results such as the type of plant disease and could be used to learn the relationship between samples and the corresponding ground truth. In contrast, meta-learning aims to learn how to learn, with which the output is rather parameters used to train a new task (Chen et al., 2021b; Li and Yang, 2021; Nuthalapati and Tunga, 2021). One of the primary issues is that current plant disease recognition datasets could not support an enormous number of source tasks, such as more than 110,000 tasks (Chen et al., 2021b). Furthermore, metric learning attempts to learn the differences between samples. In general, it pushes the samples closer within the same class and away from different classes (Afifi et al., 2020; Li and Yang, 2020; Egusquiza et al., 2022). Finally, the aforementioned strategies can be combined to further improve the performance, such as (Afifi et al., 2020) utilizing transfer learning and metric learning simultaneously.

### 4.1.2 Class imbalance

The few-shot challenge assumes that each type of plant disease occurs with a similar frequency, thus resulting in small image variations. However, some plant diseases occur at a higher frequency than others in the natural world. For instance, some plant diseases may appear more often than others, and, even for one specific type, different stages can be observed with diverse frequencies. In this case, one class may have a much higher number of annotated images than another class in the training dataset, termed a "class imbalance". Mathematically, the class imbalance challenge is formalized as $n_{Xi} \gg n_{Xj}$, where $\gg$ denotes much larger. A class with many more annotated images refers to the majority class; otherwise, it refers to the minority class (Xu et al., 2022a). The fundamental challenge is that the trained model tends to assign a high probability to the majority class during the test stage because it contributes more at the training stage (Xu et al., 2023). However, when the minority class also has many annotated images, the models may exhibit acceptable performance. Therefore, we propose a strict definition that is closer to the real applications:

$$\begin{cases} n_{X_i} \gg n_{X_j}, \\ n_{X_j} \leq M. \end{cases} \tag{2}$$

In the strict version, the number of annotated images of the minority class is not only much less than that of the majority class but should also be lower than a specific value. We argue that $M$ should not be fixed for all tasks. By contrast, this value depends on multiple factors, such as intra- and inter-class variations. Essentially, *deep learning–based models may not be able to learn robust features for the minority class in the class-imbalance datasets*. To mitigate this challenge, the primary idea is to increase the performance of the minority class while maintaining that of the majority class.

Theoretically, the majority of strategies designed for few-shot could be utilized in that class imbalance becomes a few-shot challenge when the number of annotated images of the majority class reduces to a certain extent. This subsection introduces the methods aiming to specifically facilitate the minority classes. Compared with the majority class, the lower performance of the minority class is assumed to be due to the lower observation frequency of the models. This assumption inspires balancing the frequency for models by pushing the model to look at the images of the minority class more often (Nafi and Hsu, 2020). Similarly, models can also be punished more by using samples from the minority class (Nafi and Hsu, 2020; Oksuz et al., 2020). In addition, image augmentation aims to increase image variations to facilitate deep learning–based models. The methods belonging to image augmentation for the class imbalance differing from that of the few-shot is the basic motivation that the majority class can be leveraged for the minority. Conditional image-generating models implicitly utilize this insight by training a single model to learn from all classes (Mirza and Osindero, 2014; Abbas et al., 2021). By contrast, translating an image from one class to another directly utilizes the information among these classes (Cap et al., 2020; Nazki et al., 2020; Lu et al., 2022). To further consider the intra-class image variation from the majority class to the minority class, a specific loss is leveraged along with the image translation strategy (Xu et al., 2022a).

## 4.2 Dataset-level limited dataset

The limited dataset at the class level considers situations among the classes of the training dataset, whereas heterogeneity between the training and test datasets appears at the dataset level. It is further categorized into unknown classes and domain shifts. The former suggests that some classes in the test dataset, termed unknown classes, do not appear in the training dataset; whereas the latter emphasizes that the image variations in the test and training datasets are diverse. We emphasize that these two categories focus on specific essences and that their combination may exist at a higher frequency in real-world applications.

### 4.2.1 Unknown class

The class in the training dataset refers to the "known class", whereas a class existing in the test dataset but not in the training dataset is termed an "unknown class" (Geng et al., 2020). In the concept of plant disease, unknown classes may result in a large economic loss, and recognizing them is thus one of the fundamental demands. Simultaneously, collecting all the existing plant diseases is difficult and even impossible for real-world applications. Therefore, assuming the existence of unknown classes in the test dataset is encouraging. In this scenario, the task of plant disease recognition has two-fold; classifying known classes and rejecting unknown classes (Yang et al., 2021). This task refers to open set recognition or out-of-distribution and has witnessed significant developments in the field of computer vision (Geng et al., 2020; Salehi et al., 2021;

Yang et al., 2021). However, it has been rather underdeveloped for recognizing plant disease. In the following paragraphs, three key understandings from the computer vision field are first introduced, followed by a review of the literature on plant disease recognition.

First, thresholds are commonly employed to distinguish unknown classes, such as when an image larger than the threshold is categorized as known. In this case, two things are essential: a method to compute a value for a given image and a method to set a threshold. Currently, the probability (Liang et al., 2018), energy (Zhang et al., 2020), and reconstruction error (Sun et al., 2020) are the three main strategies for a given image. For example, known classes are assumed to have higher probabilities, lower energies, and smaller reconstruction errors than unknown classes. Simultaneously, a fixed threshold tuned in the training dataset is widely employed, such as the accuracy to maintain 95% of the images in the training dataset as known (Huang et al., 2022). This fixed one can be deemed at the dataset level and the class-level threshold has been recently considered in that different known classes probably behave diversely (Wang et al., 2022a).

Second, learning a good classifier with known classes is an effective and efficient strategy, such as utilizing strong image augmentation and longer training times (Vaze et al., 2022). A good classifier requires models to learn a robust feature space to distinguish one known class from another. In general, a robust feature space is tight for a specific class and the distances between the two classes are sufficiently large, with which unknown classes have more possibilities to be recognized. However, known classes with occlusions and unknown classes with features similar to those of known classes trigger problems in this strategy (Dietterich and Guyer, 2022). Finally, the exposure of potential novel classes, not the unknown class in the test dataset, is a convincing strategy because the primary challenge is that models are trained with only known classes and extra information about unknown classes can provide extra information (Zhou et al., 2021; Dietterich and Guyer, 2022). With this paradigm, the potential unknown classes and efficiency of sampling images from unknown classes are essential (Chen et al., 2021a).

In plant disease recognition, an existing work aims to learn a good classifier via metric learning (You et al., 2022), with the inspiration that the distances between two images from the same unknown class should be smaller than those from different known classes. In general, metric learning pushes models to learn robust feature spaces and thus implicitly contributes to the recognition of the unknown class. In addition, an extra probability branch is explicitly utilized to distinguish between known and unknown classes along with a generic classification branch for known classes (Jiang et al., 2022). Simultaneously, images belonging to unknown classes are utilized to train the models, where exposure to unknown classes is beneficial, although unknown classes in the training stage may also appear in the test stage. The ratio between the number of known and unknown classes is formally analyzed (Fuentes et al., 2021a). The experimental results suggest that the performance deteriorates with more unknown classes mainly because of the shortage of useful information in the training dataset.

### 4.2.2 Domain shift

Domain shift is a common problem in deep learning where the training and test data come from different domains. In such cases, the trained model may perform poorly on the test data, resulting in a phenomenon known as a "domain shift". Domain shifts can occur for several reasons such as differences in data distribution, scale, and quality. The general assumptions are as follows:

$$P(c \mid X) \neq P(c \mid Y) \tag{3}$$

where $P(c|X)$ represents the probability distribution of one plant disease $c$ given the input images in the training dataset $X$, and $P(c|Y)$ represents the probability distribution of $c$ given the input images in test dataset $Y$. The inequality sign indicates that the two distributions are not equal, implying that the domain shift can lead to a significant decrease in the performance of the test data, making the model ineffective. The unknown class in the test dataset is a special form of the domain shift challenge, but, in this section, we aim to highlight the domain shift where the set of classes in the test dataset is a proper subset of that in the training dataset.

In the case of plant disease data, symptoms do not have well-defined boundaries and gradually change from healthy normal conditions to diseased regions, making it difficult to create homogeneity in the data (Barbedo, 2018). In addition, the inter- and intra-class image variations, as well as the explicit variations given by the domain used for the data collection, add complexity to the model. A frequent performance drop occurs when a model is trained on a dataset from a particular scenario but is further tested on data from a different scenario. A common scenario, for example, is that the training dataset is collected in one place by one person and the test dataset is collected in another place with different infrastructures and illumination by another person with their individual habit of taking pictures, such as training in the images collected in the laboratory and testing in the real world (Guth et al., 2023; Wu et al., 2023).

To address the domain shift problem, researchers have developed several techniques, such as domain adaptation (Wang and Deng, 2018) and domain generalization (Wang et al., 2022b). Domain adaptation aims to adapt a model to the test domain by modifying the training data or the model itself, whereas domain generalization aims to train a model that can perform well in unseen domains. The goal of generalization is to design a model that can operate efficiently in the same environment or across multiple environments. There are several approaches to domain adaptation for plant disease recognition using deep learning. One approach is to use transfer learning, which involves fine-tuning a pre-trained model on a new dataset. This approach can be effective when the new dataset is similar to the original dataset but may not work as well when there are significant differences between the domains. Another approach is to use domain adaptation techniques such as adversarial training or domain adaptation networks. These techniques involve training a model to recognize features specific to the target domain while also minimizing the differences between the source and target domains. This approach can be effective when there are significant differences between domains but may require more computational resources and training data.

In the literature, this problem has barely been investigated in plant disease recognition; however, it is an important issue for developing a more generalized model. Early work in this area (Fuentes et al., 2021b) shows the benefits of using control classes, such as background and healthy leaves, to lead the learning process toward classes of interest. It exhibited improved performance as an easier-to-adapt model across environments. However, data from different backgrounds and environments are required to achieve this goal. This issue was further investigated by (Fuentes et al., 2021a), where a bounding box detector was trained to obtain the regions of interest. Then, in the second stage, a domain adaption model obtained the features of data from a source farm with known diseases and transferred them to a target farm in which unlabeled data were used to assess the generalization capabilities of the model to recognize regions belonging to known classes or otherwise assigned them as unknown. This scenario showed how implementation could improve the recognition of target diseases and precisely estimate novel information by associating them with an unknown class.

Another important assumption is to address domain shifts across crops and environments. Shibuya et al. (Shibuya et al., 2021) utilized more than 221,000 labeled leaf images from different regions and crops to investigate the performance bias of evaluation within the same farm and the effect of Region-Of-Interest (ROI) detection. They found that even with many training images, the diagnostic performance for images in fields different from the training images is greatly degraded owing to covariate shifts. From this study, two important questions arise: first, what is the importance of data taken in different environments than the training data for evaluation, and, second, how do the pre-detection of regions of interest, including symptoms of diseases, affect the performance? Another essential aspect to investigate in the domain shift is the changes that occur in data collected in the laboratory compared with data collected under field conditions. The generalization capabilities of CNNs are investigated to learn the clear patterns from lab conditions that are to be detected again under new and more complex field conditions while avoiding overfitting (Guth et al., 2023). The important insight derived from this is that, in order to create useful tools for disease detection and classification using deep learning for image analysis, it is crucial to develop a final product that can handle a wide range of images from various crop conditions and locations, including inter-class and intra-class variations (Wu et al., 2023). This requires carefully designed datasets with a large number of image samples that can accommodate the significant variability in crop conditions in different areas.

In summary, domain shift is a critical challenge in plant disease recognition using deep learning. Developing models that can adapt to different domains is essential for building robust and accurate systems that can be used across a wide range of crops and regions. Although there are several approaches to the domain shift, researchers must continue to develop new techniques and datasets for ensuring that these models are effective in real-world scenarios.

# 5 Imperfect dataset

The limited dataset challenge considers annotated images within either a class or dataset in which all images in the training dataset are assumed to be annotated, whereas the imperfect dataset challenge instead hypothesizes that the annotation in the training dataset can be missing and not perfect. The imperfect dataset challenge is categorized into incomplete, inexact, and inaccurate annotations based on the violation of the EEP annotation strategy. Figure 4 provides a quick impression of the changes in performance when the annotations are different from the desired ones.

## 5.1 Incomplete annotation

Incomplete annotation indicates that the training dataset includes annotated and unannotated images, primarily because of economic issues and the annotation requirements of expert knowledge in plant science. The number of annotated images is much lower than that of unannotated images. A straightforward method is to discard unannotated images and just use the annotated images to train the models. By contrast, the use of unannotated images has become an active topic in recent years. One strategy to use unlabeled images is self-supervised learning (SSL), which aims to learn better representations, followed by a fine-tuning process within the annotated images. In SSL, a pretext task should first be defined without using annotation to train a deep learning–based model (Jing and Tian, 2020). Currently, there are many types of pretext tasks, but only a few are utilized to recognize plant disease. In particular, image augmentation does not change one image's type of plant disease and is deemed as a pretext task (Nagasubramanian et al., 2022). Furthermore, advanced image augmentation methods, such as Mixup (Zhang et al., 2018) changing the labels linearly, can also be utilized as a pretext task (Monowar et al., 2022) using the connections before and after image augmentation.

Semi-supervised learning, which is another active topic, attempts to leverage unlabeled and labeled images simultaneously. One branch directly adopts SSL methods along with a supervised loss function such as softmax. Another branch generates pseudo-labels for unlabeled images, where the labeled images can be leveraged to annotate the unlabeled images by first training a classifier (Li and Chao, 2021). In contrast, clustering methods find the similarity without label information and link the images in the test dataset to those in the training dataset (Fang et al., 2021).

Furthermore, active learning aims to select informative images labeled by humans later instead of machines, hoping to annotate fewer images yet obtain a better performance (Ren et al., 2021). Therefore, selecting informative images effectively and efficiently (Nagasubramanian et al., 2021) is essential. Moreover, the involvement of human experts in the training loop is difficult and inconvenient in real-world applications.

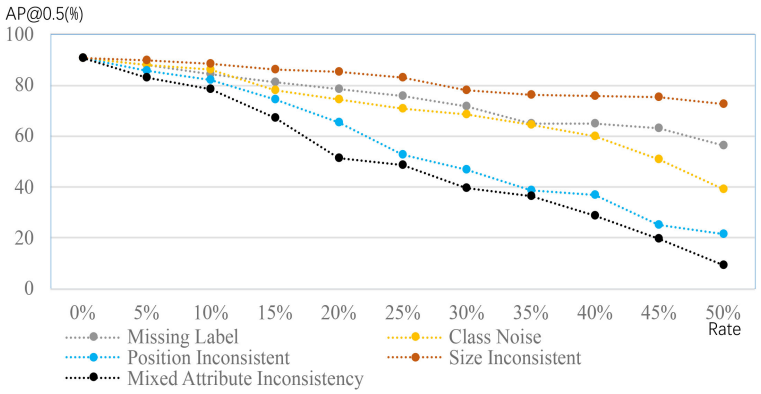



FIGURE 4
Performance comparison of object detection in a plant disease dataset, using different annotations; this figure adapted from Dong et al. (2022). The cases of missing labels and class noise suggest some patterns of plant disease have no labels and wrong labels. The inconsistencies of position and size suggest that the position and size are different from the desired. The mixed case is the combination of previous cases. Detection performance clearly degrades when the deviations from the desired ones are severe.

## 5.2 Inaccurate annotation

Inaccurate annotation, also called noisy annotation, denotes that some annotations in the training dataset may not be correct (Algan and Ulusoy, 2021; Dong et al., 2022). For example, plant disease is incorrectly annotated in the classification case, considering that experts may have conflicting decisions for a given image. Similarly, the bounding box used for object detection may be imprecise. Inaccurate annotation can be mitigated using multiple annotators, but this is expensive. Existing work (Dong et al., 2022) suggests that inaccurate annotation results in worse recognition performance, and different noise magnitudes have diverse impacts. Accordingly, facilitating the training process to reduce its impact is a straightforward approach (Li et al., 2019). Following this idea, new plant diseases are randomly generated for every image, and meta-learning is adopted to obtain consistent predictions (Zhai et al., 2022). In this manner, meta-learning aims to reduce the adverse impact of randomly generated labels. Although inaccurate annotations are facilitated, the corresponding images do not contribute to the trained models. Therefore, we highlight inaccurate annotations and employ relevant images by re-correcting the annotations (Liu et al., 2022; Wang et al., 2022c), although this idea has not yet been leveraged in plant disease recognition.

## 5.3 Inexact annotation

Inexact annotation refers to coarse-grained annotation, and the meaning varies for different tasks (Zhou, 2018; Zhang et al., 2021). For example, only image-level labels are accessible for object detection and segmentation, without bounding boxes and pixel-level classes, respectively. We emphasize that image classification also has a situation of inexact annotation, such as multiple diseases existing in one image but only one disease label. Simultaneously, inexact annotations may appear along with precise annotations. A basic assumption in using inexact annotation is that a deep learning–based model may learn a significant area with coarse-grained annotations. Specifically, the activation value in every layer indicates that the pixels contribute to the final prediction diversely. Therefore, computing the most important pixels in an input image is one way of determining the exact annotation. This strategy has been employed for object detection of crop pests (Bollis et al., 2022) and segmentation of foliar diseases (Yi et al., 2021). However, this challenge currently receives less attention than incomplete and inaccurate annotations.

# 6 Concluding remarks and future perspectives

In this study, we advocated for embracing limited and imperfect training datasets for plant disease recognition using deep learning,

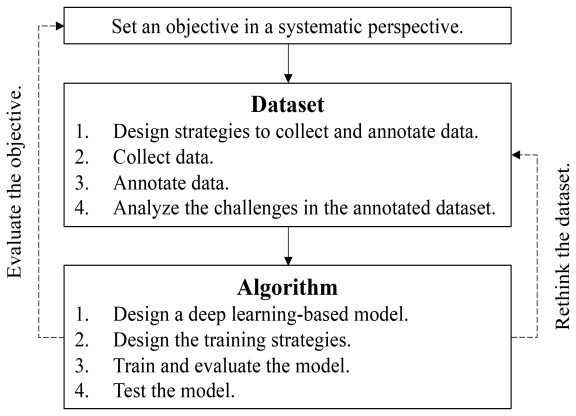



FIGURE 5
Flowchart to deploy deep learning in plant disease recognition. The evaluation of the project objectives and rethinking of the datasets are highlighted.

**BOX 1** Outstanding questions.

- How can efficiently integrate plant science, including plant disease recognition, and artificial intelligence knowledge from collecting data to deploying a deep learning model?
- How to make a reliable dataset for the application-orientated challenges.
- Is there any other challenge to deploying deep learning in plant disease recognition, except the limited and imperfect dataset?
- What are the heterogeneities between plant disease recognition and generic computer vision tasks?
- How to design a preliminary automatic prototype to recognize plant disease as a real-world application?
- Considering the success of the large language models and foundation models, what can be done in plant disease recognition and the plant science field?

**BOX 2** Future directions.

- Inspiration
  - Adopt the commonness between plant disease recognition and general computer vision tasks and then adapt the suitable concepts such as new problem formulations and methods.
  - Distinguish plant disease recognition from general computer vision tasks such as different plants having similar diseases and further leverage the difference.
- Dataset
  - Collect application-orientated datasets, such as for the domain shift.
  - Collect datasets from multiple sensors simultaneously, such as RGB and multispectral images.
  - Collect datasets from multiple observations such as spatial and temporary, inspired by the effectiveness of accumulated evidence.
  - Develop strategies to integrate datasets from the whole community.
- Model and algorithm
  - Develop strategies for specific challenges, such as for open set recognition.
  - Fine-tune large pre-trained models to achieve better performance in plant disease recognition tasks, and design strategies to achieve parameter-efficient fine-tuning (PEFT).
  - Employ small models for the embedding system.
  - Integrate large and small models to have decent performance yet for the embedding system.
- Analysis
  - Analyze the challenges in a dataset quantitatively.
  - Analyze the impacts of strategies to annotate datasets.
  - Analyze the relationship between performance, data quality and amount, computing resources, and model capacity.
- Application
  - Evaluate plant disease quantitatively, such as object detection and segmentation.
  - Deploy deep learning in real-world applications, such as robotic systems.
  - Design versatile plant disease recognition such as multi-plant and multi-dataset, rather than individual models for specific plants and datasets.
  - Consider plant disease recognition with other plant-related tasks, such as plant identification.

acknowledging the practical difficulties, expenses, and challenges associated with collecting high-quality datasets in real-world applications. Although this embrace is more convincing and practical, it also introduces new challenges. To enrich our understanding, we proposed a novel taxonomy of challenges with formal definitions. In addition, we provided a concise overview of strategies to address these challenges. One noteworthy finding is the limited research focused on dataset-level challenges related to limited datasets and imperfect annotation, whereas significant developments have been made concerning class-level limited datasets. Another essential point that we discovered is the severe shortage of benchmark datasets specifically tailored for real-world applications. By highlighting these insights, we aim to contribute to the advancement of deep learning techniques in real-world applications and foster progress in the domain of plant disease recognition. Although the primary focus of this study was on plant disease recognition, we emphasize that the concept of embracing limited and imperfect datasets is applicable to broader fields, such as deep learning in agriculture.

Building upon the challenges posed by limited and imperfect datasets, we propose a process tailored for real-world applications, depicted in Figure 5. Our objective is to emphasize the importance of evaluating and reevaluating the objectives and datasets. A fundamental assumption is that each class exhibits distinguishable visual patterns in the image space. However, certain classes may share remarkably similar patterns, making them challenging to distinguish. In such cases, the objectives of utilizing deep learning or the collected datasets should be polished, possibly by incorporating novel evidence. We also present several outstanding questions in Box 1 and outline potential future directions in Box 2, seeking to foster further research and advancements in the domain.

## Author contributions

MX: conceptualization, formal analysis, writing—original draft, and writing—review and editing. HK: project administration, funding acquisition, and writing—review and editing. JY: writing—review and editing. AF: writing—original draft, writing—review and editing. YM: writing—original draft and writing—review and editing. SY: supervision and writing—review and editing. TK: resources. DP:

supervision, project administration, funding acquisition, and writing—review and editing.

## Funding

This work was partially supported by the Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry (IPET) and Korea Smart Farm R&D Foundation (KosFarm) through the Smart Farm Innovation Technology Development Program, funded by the Ministry of Agriculture, Food and Rural Affairs (MAFRA) and Ministry of Science and ICT (MSIT), Rural Development Administration (RDA) (1545027177). This research was partially supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2019R1A6A1A09031717). This work was partially supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A2C1012174).

## Acknowledgments

We thank Jiuqing Dong (ORCID: 0000-0002-5148-9817) for the early discussions and recommended materials.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2023.1225409/full#supplementary-material

## References

Abade, A., Ferreira, P. A., and de Barros Vidal, F. (2021). Plant diseases recognition on images using convolutional neural networks: A systematic review. *Comput. Electron. Agric.* 185, 106125. doi: 10.1016/j.compag.2021.106125

Abbas, A., Jain, S., Gour, M., and Vankudothu, S. (2021). Tomato plant disease detection using transfer learning with c-gan synthetic images. *Comput. Electron. Agric.* 187, 106279. doi: 10.1016/j.compag.2021.106279

Abbasi Koohpayegani, S., Tejankar, A., and Pirsiavash, H. (2020). Compress: Self-supervised learning by compressing representations. *Adv. Neural Inf. Process. Syst.* 33, 12980–12992.

Abdulridha, J., Ampatzidis, Y., Roberts, P., and Kakarla, S. C. (2020). Detecting powdery mildew disease in squash at different stages using uav-based hyperspectral imaging and artificial intelligence. *Biosyst. Eng.* 197, 135–148. doi: 10.1016/j.biosystemseng.2020.07.001

Afifi, A., Alhumam, A., and Abdelwahab, A. (2020). Convolutional neural network for automatic identification of plant diseases with limited data. *Plants* 10, 28. doi: 10.3390/plants10010028

Algan, G., and Ulusoy, I. (2021). Image classification with deep learning in the presence of noisy labels: A survey. *Knowledge-Based Syst.* 215, 106771. doi: 10.1016/j.knosys.2021.106771

Barbedo, J. G. (2018). Factors influencing the use of deep learning for plant disease recognition. *Biosyst. Eng.* 172, 84–91. doi: 10.1016/j.biosystemseng.2018.05.013

Bengio, Y., Lecun, Y., and Hinton, G. (2021). Deep learning for ai. *Commun. ACM* 64, 58–65. doi: 10.1145/3448250

Bollis, E., Maia, H., Pedrini, H., and Avila, S. (2022). Weakly supervised attention-based models using activation maps for citrus mite and insect pest classification. *Comput. Electron. Agric.* 195, 106839. doi: 10.1016/j.compag.2022.106839

Boulent, J., Foucher, S., Theau, J., and St-Charles, P.-L. (2019). Convolutional neural networks for the automatic identification of plant diseases. *Front. Plant Sci.* 10, 941. doi: 10.3389/fpls.2019.00941

Cap, Q. H., Uga, H., Kagiwada, S., and Iyatomi, H. (2020). Leafgan: An effective data augmentation method for practical plant disease diagnosis. *IEEE Trans. Automation Sci. Eng* 19 (2), 1258-1267. doi: 10.1109/TASE.2020.3041499

Chen, J., Chen, J., Zhang, D., Sun, Y., and Nanehkaran, Y. A. (2020). Using deep transfer learning for image-based plant disease identification. *Comput. Electron. Agric.* 173, 105393. doi: 10.1016/j.compag.2020.105393

Chen, L., Cui, X., and Li, W. (2021b). Meta-learning for few-shot plant disease detection. *Foods* 10, 2441. doi: 10.3390/foods10102441

Chen, G., Peng, P., Wang, X., and Tian, Y. (2021a). Adversarial reciprocal points learning for open set recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 8065–8081. doi: 10.1109/TPAMI.2021.3106743

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. (Miami, FL, USA: IEEE), 248–255.

Dietterich, T. G., and Guyer, A. (2022). The familiarity hypothesis: Explaining the behavior of deep open set methods. *Pattern Recognition* 132, 108931. doi: 10.1016/j.patcog.2022.108931

Dong, J., Lee, J., Fuentes, A., Xu, M., Yoon, S., Lee, M. H., et al. (2022). Data-centric annotation analysis for plant disease detection: Strategy, consistency, and performance. *Front. Plant Sci.* 4937. doi: 10.3389/fpls.2022.1037655

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*. doi: 10.48550/arXiv.2010.11929

Egusquiza, I., Picon, A., Irusta, U., Bereciartua-Perez, A., Eggers, T., Klukas, C., et al. (2022). Analysis of few-shot techniques for fungal plant disease classification and evaluation of clustering capabilities over real datasets. *Front. Plant Sci.* 295. doi: 10.3389/fpls.2022.813237

Fan, X., Luo, P., Mu, Y., Zhou, R., Tjahjadi, T., and Ren, Y. (2022). Leaf image based plant disease identification using transfer learning and feature fusion. *Comput. Electron. Agric.* 196, 106892. doi: 10.1016/j.compag.2022.106892

Fang, U., Li, J., Lu, X., Gao, L., Ali, M., and Xiang, Y. (2021). Self-supervised cross-iterative clustering for unlabeled plant disease images. *Neurocomputing* 456, 36–48. doi: 10.1016/j.neucom.2021.05.066

Fuentes, A., Yoon, S., Kim, S. C., and Park, D. S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* 17, 2022. doi: 10.3390/s17092022

Fuentes, A., Yoon, S., Kim, T., and Park, D. S. (2021a). Open set self and across domain adaptation for tomato disease recognition with deep learning techniques. *Front. Plant Sci.* 12. doi: 10.3389/fpls.2021.758027

Fuentes, A., Yoon, S., Lee, M. H., and Park, D. S. (2021b). Improving accuracy of tomato plant disease diagnosis based on deep learning with explicit control of hidden classes. *Front. Plant Sci.* 12, 682230. doi: 10.3389/fpls.2021.682230

Geng, C., Huang, S.-j., and Chen, S. (2020). Recent advances in open set recognition: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 3614–3631. doi: 10.1109/TPAMI.2020.2981604

Goëau, H., Bonnet, P., and Joly, A. (2022). "Overview of plantclef 2022: Image-based plant identification at global scale," in *CLEF 2022-Conference and Labs of the Evaluation Forum*. (Bologna, Italy: Springer) 3180, 1916–1928.

Gorad, B., and Kotrappa, D. S. (2021). "Novel dataset generation for Indian brinjal plant using image data augmentation," in *IOP Conference Series: Materials Science and Engineering*. (IOP Publishing) 1065, 012041. doi: 10.1088/1757-899X/1065/1/012041

Gui, P., Dang, W., Zhu, F., and Zhao, Q. (2021). Towards automatic field plant disease recognition. *Comput. Electron. Agric.* 191, 106523. doi: 10.1016/j.compag.2021.106523

Guth, F. A., Ward, S., and McDonnell, K. (2023). From lab to field: An empirical study on the generalization of convolutional neural networks towards crop disease detection. *Eur. J. Eng. Technol. Res.* 8, 33–40. doi: 10.24018/ejeng.2023.8.2.2773

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. (Las Vegas, Nevada: IEEE), 770–778.

Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2017). "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. (Honolulu, Hawaii: IEEE), 4700–4708.

Huang, H., Wang, Y., Hu, Q., and Cheng, M.-M. (2022). Class-specific semantic reconstruction for open set recognition. *IEEE Trans. Pattern Anal. Mach. Intell* 45 (4), 4214-4228. doi: 10.1109/TPAMI.2022.3200384

Huisman, M., Van Rijn, J. N., and Plaat, A. (2021). A survey of deep meta-learning. *Artif. Intell. Rev.* 54, 4483–4541. doi: 10.1007/s10462-021-10004-4

Jiang, K., You, J., Dorj, U.-O., Kim, H., and Lee, J. (2022). Detection of unknown strawberry diseases based on openmatch and two-head network for continual learning. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.989086

Jing, L., and Tian, Y. (2020). Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 4037–4058. doi: 10.1109/TPAMI.2020.2992393

Kawasaki, Y., Uga, H., Kagiwada, S., and Iyatomi, H. (2015). "Basic study of automated diagnosis of viral plant diseases using convolutional neural networks," in Advances in Visual Computing: 11th International Symposium, ISVC 2015. (Las Vegas, NV, USA: Springer International Publishing). 638–645.

Kaya, M., and Bilge, H. S. (2019). Deep metric learning: A survey. *Symmetry* 11, 1066. doi: 10.3390/sym11091066

Kaya, A., Keceli, A. S., Catal, C., Yalic, H. Y., Temucin, H., and Tekinerdogan, B. (2019). Analysis of transfer learning for deep neural network based plant classification models. *Comput. Electron. Agric.* 158, 20–29. doi: 10.1016/j.compag.2019.01.041

Kim, B., Han, Y.-K., Park, J.-H., and Lee, J. (2021). Improved vision-based detection of strawberry diseases using a deep neural network. *Front. Plant Sci.* 11, 559172. doi: 10.3389/fpls.2020.559172

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 84–90. doi: 10.1145/3065386

Li, Y., and Chao, X. (2021). Semi-supervised few-shot learning approach for plant diseases recognition. *Plant Methods* 17, 1–10. doi: 10.1186/s13007-021-00770-1

Li, J., Wong, Y., Zhao, Q., and Kankanhalli, M. S. (2019). "Learning to learn from noisy labeled data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (Long Beach, CA, USA: IEEE), 5051–5059.

Li, Y., and Yang, J. (2020). Few-shot cotton pest recognition and terminal realization. *Comput. Electron. Agric.* 169, 105240. doi: 10.1016/j.compag.2020.105240

Li, Y., and Yang, J. (2021). Meta-learning baselines and database for few-shot classification in agriculture. *Comput. Electron. Agric.* 182, 106055. doi: 10.1016/j.compag.2021.106055

Liang, S., Li, Y., and Srikant, R. (2018). "Enhancing the reliability of out-of-distribution image detection in neural networks," in *6th International Conference on Learning Representations*. (Vancouver, BC, Canada: ICLR).

Liu, J., and Wang, X. (2021). Plant diseases and pests detection based on deep learning: a review. *Plant Methods* 17, 1–18. doi: 10.1186/s13007-021-00722-9

Liu, C., Wang, K., Lu, H., Cao, Z., and Zhang, Z. (2022). "Robust object detection with inaccurate bounding boxes," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*. 53–69.

Lu, Y., Chen, D., Olaniyi, E., and Huang, Y. (2022). Generative adversarial networks (gans) for image augmentation in agriculture: A systematic review. *Comput. Electron. Agric.* 200, 107208. doi: 10.1016/j.compag.2022.107208

Mahlein, A.-K. (2016). Plant disease detection by imaging sensors–parallels and specific demands for precision agriculture and plant phenotyping. *Plant Dis.* 100, 241–251. doi: 10.1094/PDIS-03-15-0340-FE

Matsoukas, C., Haslum, J. F., Sorkhei, M., Soderberg, M., and Smith, K. (2022). "What makes transfer learning work for medical images: feature reuse & other factors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (Louisiana: IEEE), 9225–9234.

Mirza, M., and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Mohanty, S. P., Hughes, D. P., and Salathe, M. (2016). Using deep learning for image-based plant disease detection. *Front. Plant Sci.* 7, 1419. doi: 10.3389/fpls.2016.01419

Monowar, M. M., Hamid, M. A., Kateb, F. A., Ohi, A. Q., and Mridha, M. (2022). Self-supervised clustering for leaf disease identification. *Agriculture* 12, 814. doi: 10.3390/agriculture12060814

Nafi, N. M., and Hsu, W. H. (2020). "Addressing class imbalance in image-based plant disease detection: Deep generative vs. sampling-based approaches," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*. (Niteroi, Brazil: IEEE), 243–248.

Nagasubramanian, K., Jubery, T., Fotouhi Ardakani, F., Mirnezami, S. V., Singh, A. K., Singh, A., et al. (2021). How useful is active learning for image-based plant phenotyping? *Plant Phenome J.* 4, e20020.

Nagasubramanian, K., Singh, A., Singh, A., Sarkar, S., and Ganapathysubramanian, B. (2022). Plant phenotyping with limited annotation: Doing more with less. *Plant Phenome J.* 5, e20051. doi: 10.1002/ppj2.20051

Nazki, H., Yoon, S., Fuentes, A., and Park, D. S. (2020). Unsupervised image translation using adversarial networks for improved plant disease recognition. *Comput. Electron. Agric.* 168, 105117. doi: 10.1016/j.compag.2019.105117

Neyshabur, B., Sedghi, H., and Zhang, C. (2020). What is being transferred in transfer learning? *Adv. Neural Inf. Process. Syst.* 33, 512–523.

Nuthalapati, S. V., and Tunga, A. (2021). "Multi-domain few-shot learning and dataset for agricultural applications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2021* (Nashville, Tennessee: IEEE). 1399–1408.

Oerke, E.-C., Mahlein, A.-K., and Steiner, U. (2014). "Proximal sensing of plant diseases," in *Detection and diagnostics of plant pathogens* (Springer), 55–68.

Oksuz, K., Cam, B. C., Kalkan, S., and Akbas, E. (2020). Imbalance problems in object detection: A review. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 3388–3415. doi: 10.1109/TPAMI.2020.2981890

Ouhami, M., Hafiane, A., Es-Saady, Y., El Hajji, M., and Canals, R. (2021). Computer vision, iot and data fusion for crop disease detection using machine learning: A survey and ongoing research. *Remote Sens.* 13, 2486. doi: 10.3390/rs13132486

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., et al. (2021). "Learning transferable visual models from natural language supervision," in *International conference on machine learning* (PMLR), 8748–8763.

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., et al. (2021). A survey of deep active learning. *ACM computing surveys (CSUR)* 54, 1–40.

Salehi, M., Mirzaei, H., Hendrycks, D., Li, Y., Rohban, M. H., and Sabokrou, M. (2021). A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges. *arXiv preprint arXiv:2110.14051*. doi: 10.48550/arXiv.2110.14051

Sarker, I. H. (2021). Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Comput. Sci.* 2, 1–20. doi: 10.1007/s42979-021-00815-1

Savary, S., Willocquet, L., Pethybridge, S. J., Esker, P., McRoberts, N., and Nelson, A. (2019). The global burden of pathogens and pests on major food crops. *Nat. Ecol. Evol.* 3, 430–439. doi: 10.1038/s41559-018-0793-y

Sethy, P. K., Barpanda, N. K., Rath, A. K., and Behera, S. K. (2020). Deep feature based rice leaf disease identification using support vector machine. *Comput. Electron. Agric.* 175, 105527. doi: 10.1016/j.compag.2020.105527

Shibuya, S., Cap, Q. H., Nagasawa, S., Kagiwada, S., Uga, H., and Iyatomi, H. (2021). "Validation of prerequisites for correct performance evaluation of image-based plant disease diagnosis using reliable 221k images collected from actual fields," in *AI for Agriculture and Food Systems* (Arlington, Virginia: AAAI).

Singh, A. K., Ganapathysubramanian, B., Sarkar, S., and Singh, A. (2018). Deep learning for plant stress phenotyping: trends and future perspectives. *Trends Plant Sci.* 23, 883–898. doi: 10.1016/j.tplants.2018.07.004

Singh, A., Jones, S., Ganapathysubramanian, B., Sarkar, S., Mueller, D., Sandhu, K., et al. (2021). Challenges and opportunities in machine-augmented plant stress phenotyping. *Trends Plant Sci.* 26, 53–69. doi: 10.1016/j.tplants.2020.07.010

Sun, X., Yang, Z., Zhang, C., Ling, K.-V., and Peng, G. (2020). "Conditional gaussian distribution learning for open set recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*. (Seattle, Washington: IEEE), 13480–13489.

Thakur, P. S., Khanna, P., Sheorey, T., and Ojha, A. (2022). Trends in vision-based machine learning techniques for plant disease identification: A systematic review. *Expert Syst. Appl.*, 118117. doi: 10.1016/j.eswa.2022.118117

Vaze, S., Han, K., Vedaldi, A., and Zisserman, A. (2022). "Open-set recognition: A good closed-set classifier is all you need," in *International Conference on Learning Representations*. (ICLR).

Wang, M., and Deng, W. (2018). Deep visual domain adaptation: A survey. *Neurocomputing* 312, 135–153. doi: 10.1016/j.neucom.2018.05.083

Wang, S., Gao, J., Li, B., and Hu, W. (2022c). Narrowing the gap: Improved detector training with noisy location annotations. *IEEE Trans. Image Process.* 31, 6369–6380. doi: 10.1109/TIP.2022.3211468

Wang, J., Lan, C., Liu, C., Ouyang, Y., Qin, T., Lu, W., et al. (2022b). *Generalizing to unseen domains: A survey on domain generalization* (IEEE Transactions on Knowledge and Data Engineering).

Wang, H., Li, Z., Feng, L., and Zhang, W. (2022a). "Vim: Out-of-distribution with virtual-logit matching," in *17th European Conference*. (Tel Aviv, Israel: Springer), 4921–4930.

Wei, L., Xie, L., Zhou, W., Li, H., and Tian, Q. (2022). "Mvp: Multimodality-guided visual pretraining," in *Computer Vision – ECCV 2022*. (Cham: Springer-Verlag), 337–353 (Springer-Verlag). doi: 10.1007/978-3-031-20056-420

Wu, X., Fan, X., Luo, P., Choudhury, S. D., Tjahjadi, T., and Hu, C. (2023). From laboratory to field: Unsupervised domain adaptation for plant disease recognition in the wild. *Plant Phenomics* 5, 0038. doi: 10.34133/plantphenomics.0038

Xu, M. (2023). Enhanced plant disease recognition with limited training dataset using image translation and two-step transfer learning (Jeonbuk National University). doi: 10.13140/RG.2.2.27298.91846

Xu, M., Yoon, S., Fuentes, A., and Park, D. S. (2023). A comprehensive survey of image augmentation techniques for deep learning. *Pattern Recognition* 137, 109347. doi: 10.1016/j.patcog.2023.109347

Xu, M., Yoon, S., Fuentes, A., Yang, J., and Park, D. S. (2022a). Style-consistent image translation: a novel data augmentation paradigm to improve plant disease recognition. *Front. Plant Sci.* (Bolonga, Italy: Springer), 12, 3361. doi: 10.3389/fpls.2021.773142

Xu, M., Yoon, S., Jeong, Y., Lee, J., and Park, D. S. (2022b). "Transfer learning with self-supervised vision transformer for large-scale plant identification," in *International conference of the cross-language evaluation forum for European languages*. 2253–2261 (Springer).

Xu, M., Yoon, S., Jeong, Y., and Park, D. S. (2022c). Transfer learning for versatile plant disease recognition with limited data. *Front. Plant Sci.* 13, 4506. doi: 10.3389/fpls.2022.1010981

Yang, J., Zhou, K., Li, Y., and Liu, Z. (2021). Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*. doi: 10.48550/arXiv.2110.11334

Yi, R., Weng, Y., Yu, M., Lai, Y.-K., and Liu, Y.-J. (2021). Lesion region segmentation via weakly supervised learning. *Quantitative Biol* 10 (3), 239. doi: 10.15302/J-QB-021-0272

You, J., Jiang, K., and Lee, J. (2022). Deep metric learning-based strawberry disease detection with unknowns. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.891785

Zhai, D., Shi, R., Jiang, J., and Liu, X. (2022). Rectified meta-learning from noisy labels for robust image-based plant disease classification. *ACM Trans. Multimedia Computing Communications Appl. (TOMM)* 18, 1–17.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*. (Convention center in Vancouver, Canada: ICLR).

Zhang, D., Han, J., Cheng, G., and Yang, M.-H. (2021). Weakly supervised object localization and detection: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 5866–5885.

Zhang, H., Li, A., Guo, J., and Guo, Y. (2020). "Hybrid models for open set recognition," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. 102–117 (Springer).

Zhao, X., Li, K., Li, Y., Ma, J., and Zhang, L. (2022). Identification method of vegetable diseases based on transfer learning and attention mechanism. *Comput. Electron. Agric.* 193, 106703. doi: 10.1016/j.compag.2022.106703

Zhou, Z.-H. (2018). A brief introduction to weakly supervised learning. *Natl. Sci. Rev.* 5, 44–53. doi: 10.1093/nsr/nwx106

Zhou, D.-W., Ye, H.-J., and Zhan, D.-C. (2021). "Learning placeholders for open-set recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. (Nashville, Tennessee: IEEE), 4401–4410.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., et al. (2020). A comprehensive survey on transfer learning. *Proc. IEEE* 109, 43–76. doi: 10.1109/JPROC.2020.3004555

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 28 September 2023
DOI 10.3389/fpls.2023.1199473

Check for updates

OPEN ACCESS

EDITED BY
Yalin Wu,
Chinese Academy of Sciences (CAS), China

REVIEWED BY
Oguz Gungor,
Ankara University, Türkiye
Zhiwei Ji,
Nanjing Agricultural University, China

*CORRESPONDENCE
Hongwei Sun
sunhongwei@hdu.edu.cn
Fengnong Chen
fnchen@hdu.edu.cn

RECEIVED 03 April 2023
ACCEPTED 04 September 2023
PUBLISHED 28 September 2023

CITATION
Zhang F, Sun H, Xie S, Dong C, Li Y, Xu Y, Zhang Z and Chen F (2023) A tea bud segmentation, detection and picking point localization based on the MDY7-3PTB model.
*Front. Plant Sci.* 14:1199473.
doi: 10.3389/fpls.2023.1199473

COPYRIGHT
© 2023 Zhang, Sun, Xie, Dong, Li, Xu, Zhang and Chen. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# A tea bud segmentation, detection and picking point localization based on the MDY7-3PTB model

Fenyun Zhang[1], Hongwei Sun[1]*, Shuang Xie[1], Chunwang Dong[2], You Li[1], Yiting Xu[1], Zhengwei Zhang[1] and Fengnong Chen[1]*

[1]School of Automation, Hangzhou Dianzi University, Hangzhou, China, [2]Tea Research Institute, Shandong Academy of Agricultural Sciences, Jinan, China

**Introduction:** The identification and localization of tea picking points is a prerequisite for achieving automatic picking of famous tea. However, due to the similarity in color between tea buds and young leaves and old leaves, it is difficult for the human eye to accurately identify them.

**Methods:** To address the problem of segmentation, detection, and localization of tea picking points in the complex environment of mechanical picking of famous tea, this paper proposes a new model called the MDY7-3PTB model, which combines the high-precision segmentation capability of DeepLabv3+ and the rapid detection capability of YOLOv7. This model achieves the process of segmentation first, followed by detection and finally localization of tea buds, resulting in accurate identification of the tea bud picking point. This model replaced the DeepLabv3+ feature extraction network with the more lightweight MobileNetV2 network to improve the model computation speed. In addition, multiple attention mechanisms (CBAM) were fused into the feature extraction and ASPP modules to further optimize model performance. Moreover, to address the problem of class imbalance in the dataset, the Focal Loss function was used to correct data imbalance and improve segmentation, detection, and positioning accuracy.

**Results and discussion:** The MDY7-3PTB model achieved a mean intersection over union (mIoU) of 86.61%, a mean pixel accuracy (mPA) of 93.01%, and a mean recall (mRecall) of 91.78% on the tea bud segmentation dataset, which performed better than usual segmentation models such as PSPNet, Unet, and DeeplabV3+. In terms of tea bud picking point recognition and positioning, the model achieved a mean average precision (mAP) of 93.52%, a weighted average of precision and recall (F1 score) of 93.17%, a precision of 97.27%, and a recall of 89.41%. This model showed significant improvements in all aspects compared to existing mainstream YOLO series detection models, with strong versatility and robustness. This method eliminates the influence of the background and directly detects the tea bud picking points with almost no missed detections, providing accurate two-dimensional coordinates for the tea bud picking points, with a positioning precision of 96.41%. This provides a strong theoretical basis for future tea bud picking.

KEYWORDS

**tea bud picking point, multi-attention mechanism, deep learning, DeepLabv3+, YOLOv7, focal loss**

# 1 Introduction

Tea, made from the young shoots and leaves of the tea tree, is one of the most widely consumed beverages in the world (Yu and He, 2018). In 2018, global tea production reached 5.8 million tons, with China accounting for 45% of the total production and ranking first in the world. As tea production continues to increase annually, it poses a great challenge to the labor force. In order to address this trouble, researchers have developed relevant harvesting machines (Han et al., 2014; Du et al., 2018; Xia et al., 2022; Xu et al., 2022). However, manual harvesting remains the primary method for harvesting tea shoots, supplemented by machinery, and it requires a significant amount of time and labor. However, as labor costs increase, specialists become scarce, and tea producers demand higher quality, mechanized tea picking is becoming an inevitable trend for the sustainable development of the tea industry (Zhu et al., 2021). Therefore, using computer vision to quickly and accurately identify tea shoot picking in natural environments will become a key issue for intelligent tea picking.

In order to improve crop quality through mechanized picking, the first and fundamental task is to be able to identify shoots and then locate picking points in complex environments (Li et al., 2021). In fact, many experts and scholars have researched the detection and classification of crops and other plants (Zheng et al., 2017a; Zheng et al., 2017b). For example, Hai et al. (2014) established a tea shoot color distribution model to roughly separate tea shoot regions of interest (ROIs) from complex backgrounds, and then extracted local features around the top buds of tea leaves. These features were put into mean offset clustering to locate the tea shoot picking points. Lin et al. (2019) proposed a detection algorithm that utilizes color, depth, and shape information to detect spherical or cylindrical fruits on plants in natural environments, guiding harvesting robots to pick them automatically. Liu et al. (2019) used color features extracted from blocks to determine candidate regions and adopted the histogram of oriented gradient (HOG) to describe the shape of fruits, which achieved the average values of recall, precision, and F-1 reach 89.80%, 95.12%, and 92.38% respectively. Wang et al. (2019) utilized a natural statistical visual attention model to remove background saliency and combined it with a threshold segmentation algorithm to extract salient binary regions of apple images. Wu et al. (2019) presented a clustering and model segmentation based approach for detecting ripe peaches using RGB-D (red, green, blue, and depth space) cameras combined with color data and 3D contour features. As observed in the aforementioned study, the target objects are distinct from the background objects such as leaves in terms of color and shape, enabling easy recognition and extraction through color and shape features. However, the similarity in color and shape between tea buds and tea leaves, as well as the background of tea buds being tea leaves, makes it difficult for the human eye to distinguish between them. Consequently, traditional detection methods relying on color and shape are inadequate for accurately identifying tea buds within tea leaves.

In recent years, the development of artificial intelligence and 5G networks has had a significant impact on agriculture, as science and technology have continued to evolve. Furthermore, deep learning has opened up new possibilities for researchers in various fields, including target recognition. Wu et al. (2023) devised novel Inner Cascaded U-Net and Inner Cascaded U2-Net as improvements to plain cascaded U-Net for medical image segmentation, achieving better segmentation performance in terms of dice similarity coefficient and hausdorff distance as well as getting finer outline segmentation. Nie et al. (2022) proposed a method named SegNet that was developed and trained with different data groups. Quantitative metrics and clinical-based grading were used to evaluate differences between several groups of automatic contours. Yuan et al. (2022) proposed an enhanced network architecture based on PSPNet, referred to as Shifted Pool PSPNet, which integrates a module called Shifted Pyramid Pool instead of the original Pyramid Pool module to enable the utilization of entire local features for pixels located at the grid edges. Xie et al. (2023) proposed a population-based intelligent algorithm called Salp Swarm Algorithm for Feature Selection (SSAFS) for plant disease detection based on images. This algorithm aims to determine the optimal combination of handcrafted features and reduce the feature dimensionality to improve accuracy. Compared to the state-of-the-art algorithms, SSAFS demonstrates advantages in exploring the feature space and utilizing valuable features for plant disease image classification. Fu et al. (2021) proposed an improved DeepLabv3+ semantic segmentation algorithm for crack detection. The proposed method utilizes a densely connected spaceless pyramidal pooling module in the network structure, which enables the network to obtain more dense pixel sampling and improves the network's ability to extract detailed features. Xie et al. (2022) proposed a new computational framework that combines deep embedding image clustering strategy, weighted distance measurement, and t-random neighbor embedding algorithm. The results indicate that the newly developed framework can identify plant diseases and uncover subtypes effectively, demonstrating excellent clustering performance. Yang et al. (2019) trained a model for detecting tea tree shoots using an improved "You Only Look Once" (YOLO) network. They achieved high precision results for the validation dataset. Xu et al. (2022) proposed a tea shoot recognition model based on a cascade network. First, the YOLOv3 network was used for the initial selection of tea shoot regions, and the resulting recognition outputs were then fed into DenseNet. Subsequently, the recognition results were further processed by the DenseNet-201 tea shoot classification network. The final recognition precision of tea shoots reached 95.71%, which provides a new approach to tea shoot recognition. Chen et al. (2022) proposed a tea-picking point location method based on YOLO-v3 algorithm, semantic segmentation algorithm, skeleton extraction, and minimum bounding rectangle. They designed an intelligent tea-picking machine based on personal computer and microcontroller cooperative control, which solved the problems of complex shadows and easy damage during the picking process. Jiang et al. (2022) proposed an improved attention mechanism YOLOv7 algorithm, named CBAM-YOLOv7, which adds three CBAM (Convolutional Block Attention Module) modules to the backbone network of YOLOv7 to enhance the network's feature extraction capabilities. They also conducted comparison experiments using SE-YOLOv7 and ECA-YOLOv7. Yan et al.

(2022) proposed the MR3P-TS model for tea shoot detection. The model calculates the area of multiple connected domains in the mask to determine the main part of the tea shoot. Then, it calculates the minimum bounding rectangle of the main part to determine the stem axis of the tea tree. Finally, the location coordinates of the tea shoot's pickup point are obtained. Gui et al. (2023) proposed a lightweight tea leaf detection model based on an improved YOLOv5 architecture. The model incorporates the Ghost_conv module to reduce model size and includes the BAM module in the backbone network to suppress irrelevant information. The improved model achieved an average precision increase of 9.66% while reducing parameters by 22.71 M. Lu et al. (2023) proposed a method for segmentation and localization of tea buds. The method initially employs four semantic segmentation algorithms to process the images of tea buds. The position of the tea bud is determined by calculating the centroid and the centroid of the minimum bounding rectangle. This method achieves effective localization of tea buds. Qi et al. (2021) combined the "Maximum Between-Class Variance Method" (Otsu) with the traditional watershed algorithm to determine the threshold for image segmentation, thereby improving the accuracy of segmentation. They also improved the SE module to enhance the performance of deep learning networks and achieved outstanding accuracy on the tea bud dataset.

Therefore, the problem of segmenting, recognizing, and localizing objects with similar background colors and shapes, such as tea buds in complex environments, needs to be addressed, and precision needs to be further improved. In order to achieve precise identification and localization of tea buds, it is necessary to accurately obtain the two-dimensional coordinates of the image. After obtaining the two-dimensional coordinates, a specific coordinate system conversion is performed to ultimately obtain the three-dimensional spatial coordinates of the picking point. Therefore, accurate two-dimensional image coordinates play a crucial role in obtaining future three-dimensional spatial coordinates. To obtain accurate two-dimensional image coordinates, we establish an image dataset of tea buds in their natural growth state and propose a new model called the MDY7-3PTB model. The main contributions are as follows: (1) Focal Loss as a loss function to address data imbalance; (2) the improved DeepLabv3+ model with a fused CBAM attention mechanism for tea shoot segmentation in complex environments, removing interference from the background in tea shoot picking point detection; (3) accurate detection of tea shoot picking points based on YOLOv7;(4) the geometric center of the bounding box to achieve two-dimensional coordinate positioning of tea shoot picking points and restore their backgrounds.

## 2 Related work

### 2.1 DeepLabV3+ network segmentation: one of the top-performing semantic segmentation algorithms currently available.

The DeepLabV3+ network segmentation is formed by building the encoding and decoding structure upon the DeepLabv3 architecture with the addition of concise and effective decoders. The overall network structure is illustrated in Figure 1.

As can be seen from the figure, it is composed of an encoder network and a decoder network, which is similar to the structure of traditional semantic segmentation networks. At the encoder stage, the input image is subjected to initial feature extraction using the Xception backbone network that includes null convolutions to produce a low-order feature map that is 1/4 of the original image size and a high-order semantic feature map that is 1/16 of the original image size. The higher-order semantic feature maps are then fed into the ASPP module, which is composed of null convolutional layers with expansion rates of 6, 12, and 18 to capture the contextual global information of the feature maps. Finally, the ASPP module outputs the feature map for channel stitching. Since the feature map has a high feature channel

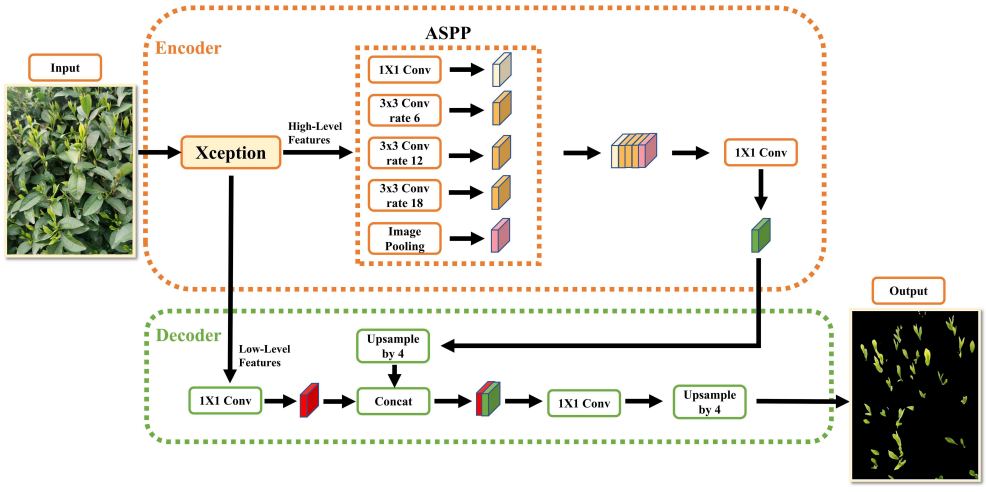



FIGURE 1
The original architecture of the DeepLabv3+ network.

dimension at this point, 1×1 convolution is used to downscale it to reduce network computation, and the downscaled feature map is passed into the decoder network.

In the decoding stage, the bilinear interpolation algorithm is used to upscale the feature map output from the encoder network by a factor of 4 to match the size of the low-level feature map. Then, the two feature maps are concatenated and fused for channel stitching. Finally, the fused feature maps are restored to the original image size using 3x3 convolution and 4 times up-sampling, and the final segmentation result is obtained by applying Softmax probability prediction (Zheng et al., 2022).

## 2.2 Attention mechanism: focusing on selective information

The attention mechanism is an algorithm that imitates the selective observation behavior of the human brain's visual system. Its primary function is to assign larger weights to significant features to highlight their importance and smaller weights to irrelevant features to suppress the interference of irrelevant features during network training, thereby enhancing the learning ability of network models. In the tea shoot recognition task discussed in this paper, background noise can interfere with the recognition of tea shoots due to the relatively small number of shoot features and the lack of contrast with the background color. Thus, incorporating the attention mechanism into the tea shoot segmentation network can decrease the feature weight of background noise and enhance the representation of effective shoot features, thereby improving the network's recognition and localization capability for tea shoots.

During the development of computer vision applications, numerous works on attention mechanisms have been proposed. One of these is the Convolutional Block Attention Module (CBAM), which was introduced by Woo et al. (2018) In 2018, this attention module combines channel attention and spatial attention to enable the network to highlight important features and suppress irrelevant features. The CBAM module differs from other attention modules in that it focuses on both channel and spatial dimensions, thereby achieving better efficiency.

The CBAM attention mechanism first employs the channel attention mechanism to enhance important channel features and suppress irrelevant ones. Then, the enhanced features are passed to the spatial attention mechanism to locate the region with the most informative features (in this case, the tea shoot region). Finally, the processed feature results are output. The formula can be expressed as follows:

$$F' = M_c(F) \otimes F \tag{1}$$

$$F'' = M_s(F') \otimes F' \tag{2}$$

In the formula, $F \in R^{C \times H \times W}$ represents the input feature map, where R represents the set of real numbers and C, H, and W represent the number of channels, height, and width of the feature map, respectively. $M_c \in R^{C \times 1 \times 1}$ represents the channel attention weights obtained by applying the channel attention mechanism, and $M_s(F') \in R^{1 \times H \times W}$ represents the spatial attention weights obtained by applying the spatial attention mechanism to the transformed feature map F'.

In the channel attention module, the input feature map is compressed using a spatial dimension method, and the AvgPool and MaxPool methods are applied simultaneously to effectively calculate the weight attention assigned to the channel dimension. The formula can be expressed as follows:

$$\begin{aligned} M_c(F) &= \sigma\left(\mathrm{MLP}\left(\mathrm{AvgPool}(F)\right) + \mathrm{MLP}\left(\mathrm{MaxPool}(F)\right)\right) \\ &= \sigma\left(W_1\left(W_0\left(F^C_{avg}\right)\right) + W_1\left(W_0\left(F^C_{max}\right)\right)\right) \end{aligned} \tag{3}$$

In the formula, σ represents the Sigmoid function. MLP refers to a multi-layer perceptron with a hidden layer whose operational weights are determined by $W_0$ and $W_1$, where $W_0$ is activated by the ReLU (Rectified Linear Unit) function, $W_0 \in R^{\frac{C}{r} \times C}$, $W_1 \in R^{C \times \frac{C}{r}}$.

The spatial attention module focuses on the location of information in the image and is complementary to the previous module. Computationally, it first applies AvgPool and MaxPool operations on the channel axis and concatenates them into a meaningful feature descriptor. The two pooling operations aggregate channel information of a feature map to generate a two-dimensional map. Finally, a convolution operation is performed by the convolution layer to obtain the corresponding spatial feature map (Chen et al., 2022). The formula can be expressed as follows:

$$\begin{aligned} M_S(F) &= \sigma\left(f^{7\times7}\left(\left[\mathrm{AvgPool}(F); \mathrm{MaxPool}(F)\right]\right)\right) \\ &= \sigma\left(f^{7\times7}\left(\left[F^S_{avg}; F^S_{max}\right]\right)\right) \end{aligned} \tag{4}$$

In the formula, σ represents the Sigmoid function, $\int^{7\times7}$ represents the 7 × 7 convolution kernel and $F^S_{avg}, F^S_{max} \in R^{1\times H\times W}$.

To enhance the detection precision of the network, we incorporated the CBAM attention module into the backbone network structure, building upon its excellent performance. The resulting structure is illustrated in Figure 2.

## 2.3 YOLOv7: an excellent object detection model

In 2015, YOLOv1 (Redmon et al., 2016) was proposed, which introduced single-stage detection algorithms and effectively addressed the problem of slow inference speed in two-stage detection networks while maintaining good detection accuracy. YOLOv2 (Redmon and Farhadi, 2017) was an improvement over YOLOv1, with each convolutional layer being followed by a Batch Normalization layer, and dropout no longer being used. YOLOv3 (Redmon and Farhadi, 2018) was an improved version of the previous work, with its main feature being the introduction of the residual module darknet-53 and the FPN architecture, which predicted objects of three different scales and achieved multi-scale fusion. Based on the YOLOv3 version, YOLOv4 (Bochkovskiy et al., 2020) introduced PANet networks, mosaic data enhancement and CIoU loss function. Subsequently, YOLOv5 (Glenn, 2020) introduced various data enhancement methods, C3 modules,

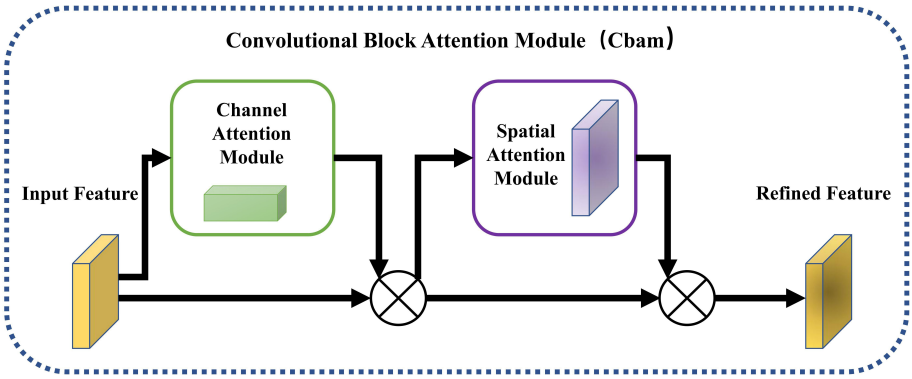



FIGURE 2
The schematic diagram of the CBAM attention mechanism.

SPPF spatial pyramid pooling and Focal Loss loss function. In 2022, YOLOv7 (Wang et al., 2022) was introduced, which is currently one of the best detection models. It innovatively proposed the Extended-ELAN architecture, which can improve the network's self-learning ability without breaking the original gradient path. Additionally, it adopts a model scale-based cascade method, which can generate models of corresponding scales for practical tasks to meet detection requirements. To more clearly express the internal composition of its network, the structure of each part of the YOLOv7 model is shown in Figure 3.

# 3 Materials and methods

## 3.1 Data acquisition

This paper focuses on the study of tea bud images in complex environments, and the tea bud samples used in this study were obtained from the Hangzhou Tea Research Institute in Zhejiang Province in 2022. The images were captured using the high-definition camera of a Xiaomi 10 mobile phone, with parameters shown in Table 1. In order to obtain more authentic and high-quality tea bud images in natural environments, the data was collected in mid-to-late March when the weather was clear and sunny outdoors. During the process of collecting tea bud image data, we changed the angle and distance of the camera multiple times to collect tea bud data from different directions and distances, and manually filtered out highly repetitive and blurred data. The collected tea bud images varied in pose, covering various distances, angles, and directions. Some of the collected tea bud images are shown in Figure 4.

## 3.2 Data preprocessing

### 3.2.1 Correction of category imbalance

Although this paper only focuses on the segmentation of tea buds into two categories (tea bud or background), the small size of the tea buds in the images and their small proportion in the overall image lead to a significant class imbalance, with a large proportion of the background class. This imbalance can result in inefficient training and negatively impact the model's precision. Therefore, it is necessary to correct this class imbalance in the dataset. In this paper, we use the Focal Loss loss function to perform category imbalance correction. The formula can be expressed as follows:

$$FL(P_t) = -\alpha_t(1 - P_t)^{\gamma}\log(P_t) \tag{5}$$

$$P_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise} \end{cases} \tag{6}$$

In the formula, P represents the detection result, y represents the true label, $\alpha_t$ represents the weighting factor, $(1 - P_t)^{\gamma}$ represents the adjustment factor and $\gamma \geq 0$ is the adjustable focusing parameter.

In this paper, we experimentally adjust the parameters $\alpha_t = 0.5, \gamma = 2$.

### 3.2.2 Image annotation

Due to the high density of tea bud picking points and the similarity in color between tea buds and tea leaves, it is challenging to annotate a large number of original images, leading to low prediction results and missed detections during model training and prediction. To overcome these issues, this study proposes segmenting all content other than tea buds as background, which makes the color of tea buds stand out and facilitates the image labeling, model training, and prediction. This approach improves overall efficiency and accuracy.

In the detection phase, the segmented dataset is utilized to train the YOLOv7 model. Image annotation, being the fundamental component, significantly impacts the training and prediction of the model. Therefore, the annotation process is divided into four stages. The initial step involves manual labeling of partially segmented images. Subsequently, the partially labeled and all unlabeled images are uploaded to the EasyData platform for intelligent labeling in the second stage. In the third step, the images that are not fully labeled or unlabeled are identified and grouped with the manually labeled images. The fourth step involves filtering the labeled images to ensure completeness. Finally, the labeled files with annotations are saved in the VOC format. Figure 5 displays the effect of labeled tea bud picking point images.

10.3389/fpls.2023.1199473

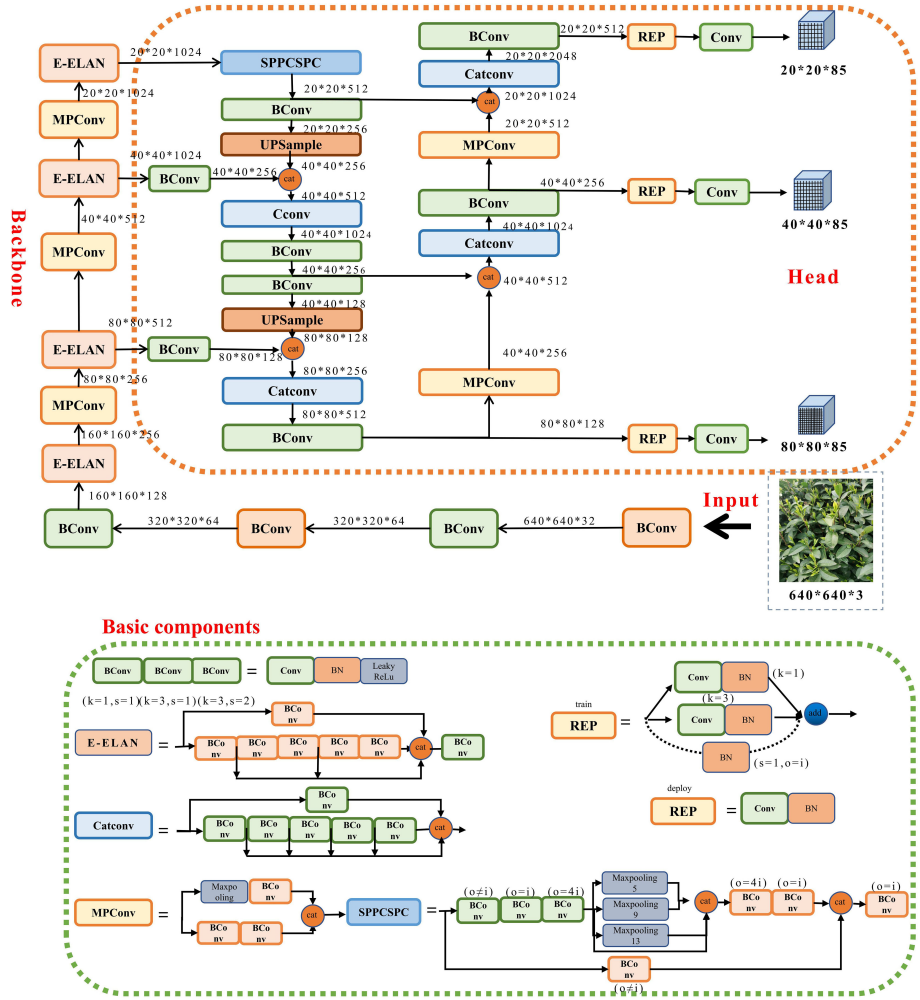



FIGURE 3
YOLOv7 network architecture.

TABLE 1 The camera parameters of Xiaomi 10 mobile phone.

| Parameter | Value |
|---|---|
| Camera manufacturer | Xiaomi |
| Camera Model | Mi 10 |
| Resolution | 3000×4000 |
| Aperture value | f/1.7 |
| Exposure time | 1/1001 sec |
| ISO speed | ISO-50 |
| Exposure compensation | 0 stops aperture |
| Focal length | 7mm |
| Maximum aperture | 1.51 |
| Metering mode | Off-center average |
| Flash mode | No flash, forced |
| Image type | JPG |

### 3.2.3 Data augmentation

In deep learning object recognition networks, the amount of data often affects the final recognition performance, and too little data can easily lead to overfitting. For tea bud images, it is not enough to rely solely on downloading from the internet or collecting them personally, and the collection of tea bud images is also limited to around the Qingming Festival, which has temporal restrictions, and taking tea bud images in the field is also very time-consuming. Therefore, it is necessary to augment the tea bud image data. The specific operation of data augmentation is to expand the existing tea bud image by using relevant data augmentation methods before making the dataset, so as to achieve the effect of increasing the quantity of the dataset. Image brightness adjustment and flipping not only can expand the dataset but also improve the model's robustness, accuracy, and generalization performance. Therefore, this paper used brightness increased, brightness decreased, horizontal flipping and vertical flipping for data augmentation (Xu et al., 2022) and the final dataset reached 2948 images. The image dataset was randomly divided into three groups to form the model training, validation and test datasets, with proportions of 70%, 20%, and 10%, respectively. These datasets

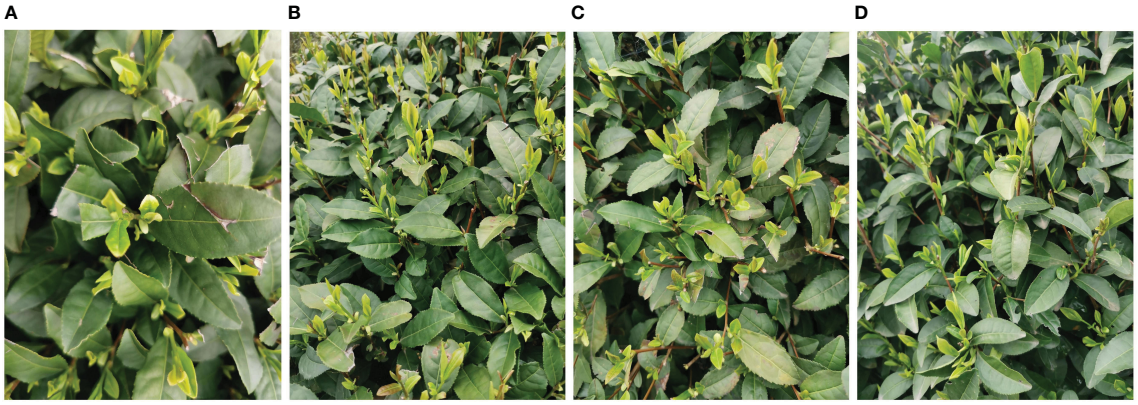



FIGURE 4
Images of tea buds were obtained under different conditions in complex environments. **(A)** close range, **(B)** long range, **(C)** top shot and. **(D)** side shot.

will be used for model training and parameter optimization, and compared with the prediction results to evaluate the model's object detection performance, as shown in Figure 6.

## 3.3 Experimental environment

In this paper, all experiments were conducted on the same computer, and specific information on the computer's hardware and software configuration and model training environment is shown in Table 2.

## 3.4 Training parameters

The training parameters of the training process used in the experiment are shown in Table 3.

## 3.5 Evaluation metrics

### 3.5.1 Performance evaluation of segmentation

In this paper, several commonly used evaluation metrics for segmentation models are selected as segmentation performance evaluation metrics for the tea bud dataset, including pixel precision, mean pixel precision (mPA), intersection over Union (IoU), mean intersection over Union (mIoU), and mean recall (mRecall). Additionally, the number of parameters (Params) and frame rate (FPS) are also used as evaluation metrics for tea bud segmentation performance, taking into consideration the practical needs of tea bud segmentation.

The pixel precision metric represents the proportion of correctly predicted category pixels to the total number of pixels in the segmentation. The formula can be expressed as follows:

$$PA = \frac{\sum_{i=0}^{k} P_{ii}}{\sum_{i=0}^{k} \sum_{j=0}^{k} P_{ij}} \tag{7}$$

The mean pixel precision represents the average value of the sum of the ratio of correctly predicted pixel points to the total pixel points for each category. The formula can be expressed as follows:

$$mPA = \frac{1}{k+1} \sum_{i=0}^{k} \frac{P_{ii}}{\sum_{j=0}^{k} P_{ij}} \tag{8}$$

The Intersection over Union measures the overlap between the predicted and ground truth segmentation masks for each object category. It is defined as the ratio of the intersection between the

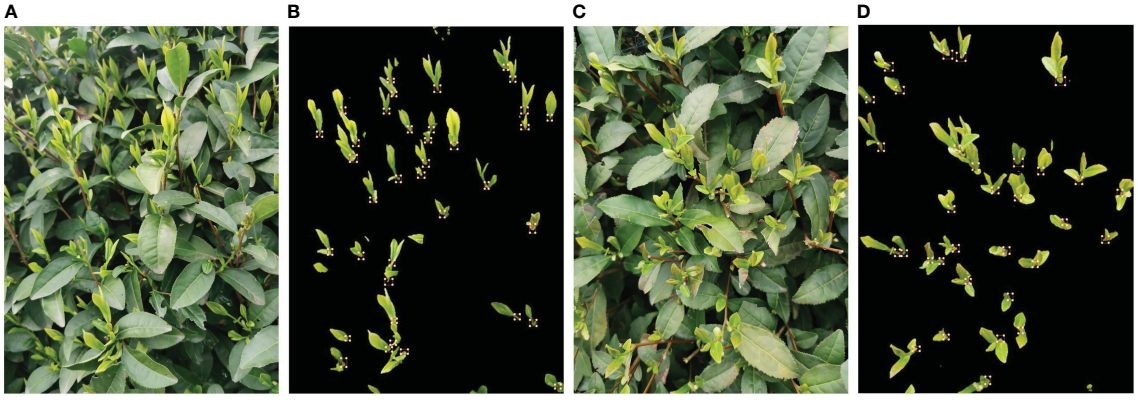



FIGURE 5
Examples of labeling results for tea bud picking points. **(A, B)** and **(C, D)** represents the original image and its annotation process.

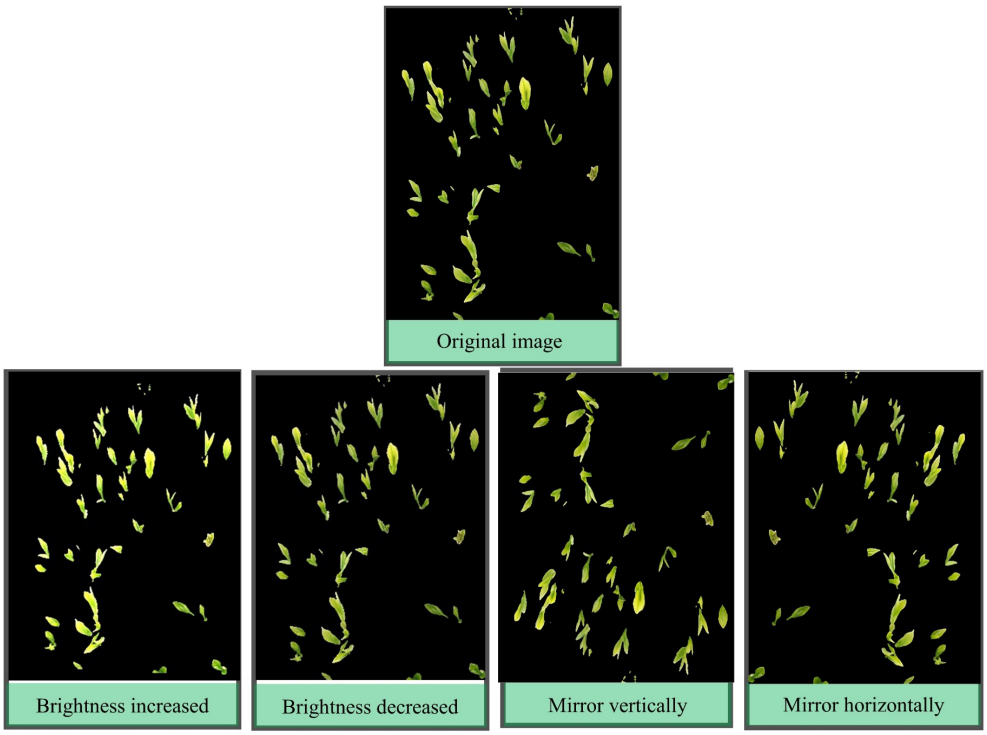



FIGURE 6
Results of data augmentation.

predicted and ground truth masks to their union. The formula can be expressed as follows:

$$Iou = \frac{P_{ii}}{\sum_{j=0}^{N} P_{ij} + \sum_{j=0}^{N} P_{ji} - P_{ii}} \tag{9}$$

The mean intersection over union represents the average of the intersection-over-union ratios between the predicted results and the ground truth labels for each category. It is a commonly used metric to evaluate the segmentation performance of models. The formula can be expressed as follows:

$$mIou = \frac{1}{k+1} \sum_{i=0}^{k} \frac{P_{ii}}{\sum_{j=0}^{k} P_{ij} + \sum_{j=0}^{k} P_{ji} - P_{ii}} \tag{10}$$

TABLE 2 Computer hardware and software configuration and model training environment.

| Environmental parameter | Value |
|---|---|
| CPU | AMD R9 4900H |
| GPU | GeForce 3060Ti |
| RAM | 16GB |
| Video memory | 6GB |
| Operating system | Windows10 |
| Deep learning framework | PyTorch |
| Cudnn | Cudnn10.1 |
| OpenCV | 4.5.2 |

The mean recall is the average ratio of the number of correctly classified pixels in each class to the total number of pixels in that class. The formula can be expressed as follows:

$$mRecall = \frac{1}{k+1} \sum_{i=0}^{k} \frac{P_{ii}}{P_{ii} + \sum_{j=0}^{k} P_{ji}} \tag{11}$$

where $k$ is the total number of categories, $P_{ij}$ represents the number of pixels that belong to class $i$ but are predicted as class $j$, $P_{ii}$ represents the correct number predicted classes, and $P_{ji}$ is false positive or false negative.

### 3.5.2 Performance evaluation of testing

To evaluate the detection model's performance, this study uses precision(P), recall (R), mean average precision(mAP), and F1 score as evaluation metrics. Precision represents the proportion of true positive samples to all samples predicted as positive. The formula can be expressed as follows:

$$Presion = \frac{TP}{TP+FP} \tag{12}$$

Recall represents the proportion of true positive samples to all positive samples in the dataset. The formula can be expressed as follows:

$$Recall = \frac{TP}{TP+FN} \tag{13}$$

The formula shows that TP represents the number of predicted bounding boxes where the tea bud picking point is located, FP represents the number of predicted bounding boxes where the tea bud picking point is not located, and FN represents the number of

TABLE 3 Training parameters.

| Segmentation parameter | Value | Detection parameter | Value |
|---|---|---|---|
| Init_lr | 0.007 | Init_lr | 0.01 |
| Init_Epoch | 0 | Init_Epoch | 0 |
| Freeze_Epoch | 50 | Freeze_Epoch | 50 |
| Freeze_batch_size | 8 | Freeze_batch_size | 8 |
| UnFreeze_Epoch | 330 | UnFreeze_Epoch | 620 |
| Unfreeze_batch_size | 4 | Unfreeze_batch_size | 4 |
| Image Size | 512*512 | Image Size | 640*640 |
| Momentum | 0.9 | Momentum | 0.937 |
| Optimizer | sgd | Optimizer | sgd |
| backbone | MobileNetV2 | Mixup_prob | 0.5 |

missed bounding boxes where the tea bud picking point is located. Therefore, precision represents the proportion of correct predictions among all predicted outcomes, while recall represents the proportion of correct predictions among all true targets, where the values of both precision and recall are between 0 and 1.

The F1 score represents the weighted average of precision and completeness. The formula can be expressed as follows:

$$F1 = \frac{2}{Recall^{-1}+Precision^{-1}} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (14)$$

Precision reflects the ability of a model to correctly classify negative samples. A higher precision indicates a stronger ability of the model to distinguish negative samples. Recall, on the other hand, reflects the ability of the model to correctly identify positive samples. A higher recall indicates a stronger ability of the model to identify positive samples. F1 score is a combination of both precision and recall, where a higher F1 score indicates a more robust model.

The mean average precision represents the average value of the AP sought for all categories. The formula can be expressed as follows:

$$mAP = \frac{\sum_{j=1}^{S} AP(j)}{S} \quad (15)$$

In the formula, S represents the number of all categories, and the numerator represents the sum of APs of all categories. Since this study only tested for tea bud picking points, the mAP can be calculated as mAP = AP.

## 3.6 MDY7-3PTB model

### 3.6.1 *A* general overview *of the MDY7-3PTB* model

The tea bud localization method proposed in this study, named MDY7-3PTB model, as shown in Figure 7. The MDY7-3PTB model combines the high-precision segmentation capability of DeepLabv3+ and the fast detection capability of YOLOv7. The process of locating tea bud picking points consists of three stages. In the first stage, which is the segmentation stage of tea buds, features are extracted from the input image using the MobileNetV2 backbone. Then, the feature maps are passed through the CBAM module to generate attention feature maps in both channel and spatial dimensions. These two feature maps are multiplied with the previous raw input feature map for adaptive feature recalibration, and then sent to the ASPP module. In the second stage, after the segmentation is completed, the detection of tea bud picking positions is performed. The segmented image is fed into the YOLOv7 backbone, which extracts features from the image, and then through the SPPCSPC module. This module addresses image distortion caused by image processing operations and the challenge of extracting repetitive features by convolutional neural networks. This approach is achieved by merging multiple MaxPool operations in a series of convolutions. Additionally, the FPN feature extractor combines high-level semantic information with low-level detail information to improve the detection of small targets. Finally, the fused features are input to box regression for boundary box correction, resulting in the predicted image of tea bud picking points. In the third stage, the predicted image is used to restore the background, and the two-dimensional coordinates are computed from the center of the rectangular boxes to obtain the tea bud picking points.

### 3.6.2 Segmentation using MDY7-3PTB model

DeepLabv3+ is currently the best-performing semantic segmentation model, however, it still has some limitations. For instance, the feature extraction network Xception has a large number of layers and parameters, leaving room for improvement in segmentation precision and operation speed. To enhance the segmentation effect of tea buds, this paper proposes several improvements to the DeepLabV3+ model. Firstly, the unsatisfactory Xception feature extraction network is replaced by the more lightweight MobileNetV2 network, significantly reducing the number of model parameters and improving calculation speed. Multiple fusion channels and the spatial attention mechanism CBAM are introduced before the feature extraction module and feature map input decoder to obtain better image features.

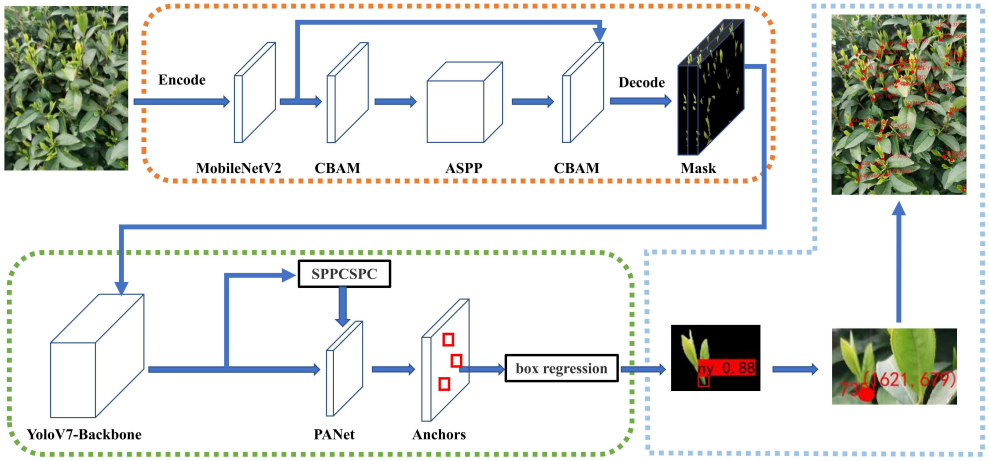



FIGURE 7
Overall overview of MDY7-3PTB model for tea bud segmentation, detection and localization.

Additionally, the weighted loss function is introduced to address the class imbalance problem of the dataset and improve the model's segmentation precision for tea buds. The network structure of the improved DeepLabV3+ algorithm is illustrated in Figure 8.

### 3.6.3 Detection using MDY7-3PTB model

To accurately detect segmented tea bud pictures, the MDY7-3PTB model utilizes the YOLOv7-Backbone as the detection backbone network. The labeled tea bud picking point dataset is fed into the YOLOv7-Backbone backbone network to extract picture features, which are then input into the SPPCSPC template to address issues such as image distortion caused by image processing operations and the problem of repeated features extracted from pictures by convolutional neural networks. After PANet feature fusion, the images are input to box regression for prediction box correction, resulting in the prediction of tea bud picking points. The detection backbone network consists of convolution, E-ELAN module, MPConv module, and SPPCSPC module. The E-ELAN module (Jiang et al., 2022), based on the original ELAN, modifies the computational blocks while retaining the transition layer structure of the original ELAN. It uses the concepts of expand, shuffle, and merge cardinality to achieve enhanced network learning without disrupting the original gradient path.The SPPCSPC module introduces multiple parallel MaxPool operations into a series of convolutions, which helps to prevent image distortion caused by image processing operations and the problem of extracting duplicate image features by convolutional neural network. In the MPConv module, the

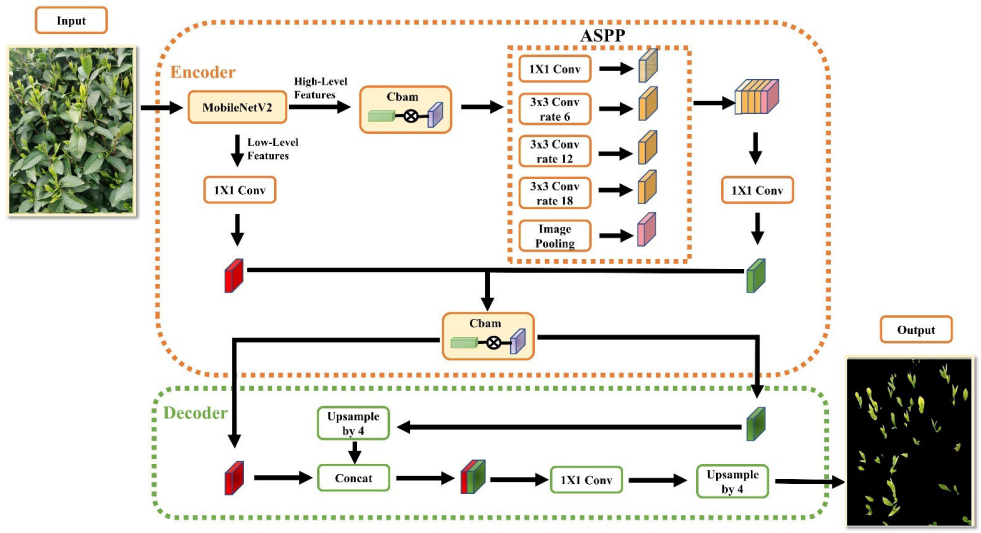



FIGURE 8
Improved DeeplabV3+ algorithmic network architecture.

MaxPool operation expands the receptive field of the current feature layer and fuses it with the feature information after normal convolution processing, which improves the generalization of the network. The PANet module is a top-down and bottom-up bidirectional fusion backbone network with a "shortcut" between the bottom and top layers to shorten the path between layers. It also includes two modules, adaptive feature pooling and full connection fusion. The adaptive feature pooling can be used to aggregate features between different layers to ensure the integrity and diversity of features, and the full-connection fusion can achieve more accurate prediction results.

### 3.6.4 Tea bud picking point positioning using the MDY7-3PTB model

To obtain the two-dimensional coordinates of the tea buds, this study selects the geometric center of the prediction frame as the tea bud picking point coordinates, and applies the Box Regression module, which uses a 4-dimensional vector to represent the window, including the coordinates of the center point, width, and height. Finally, the window is adjusted through translation and scaling to gradually converge to the real value and obtain the exact coordinates of the tea bud picking point. Take the coordinates of A and D as shown in Figure 9. The formula for calculating the geometric center of the rectangle is as follows. $x_0$ and $y_0$ are the horizontal and vertical coordinates of the center point of the rectangular box. $x_1$ and $x_2$ is the horizontal coordinate of the vertex of the rectangle. $y_1$ and $y_2$ is the ordinate of the vertices of the rectangle.

$$x_0 = \frac{x_1 + x_2}{2} \tag{16}$$

$$y_0 = \frac{y_1 + y_2}{2} \tag{17}$$

## 4 Experimental results

### 4.1 Segmentation performance of the MDY7-3PTB model

As the segmentation results of tea shoot images can directly affect the subsequent identification and localization of picking points, segmentation precision is of primary concern. Additionally, the number of model parameters and segmentation speed, which are related to subsequent model deployment, should also be important factors to consider. To address these issues, this paper employs the Focal Loss function for data imbalance correction and proposes an improved DeeplabV3+ model, namely the MDY7-3PTB model, for segmentation. To further validate the segmentation performance of the MDY7-3PTB model, this study selected commonly used crop segmentation models such as PSPNet, Unet, and DeeplabV3+ for comparison. Additionally, different backbones were employed for the three comparative models, resulting in the establishment of six models: PSPNet_MobileNetV2, PSPNet_Resnet50, Unet_VGG, Unet_Resnet50, DeeplabV3+_MobileNetV2, and DeeplabV3+_Xception. In this study, 1000 images were selected for the training and testing of the segmentation model. The comparison results are presented in Table 4. The mIoU, mPA, and mRecall in Table 4 represent the mean and standard deviation of the model's results from 10 tests.

The results in Table 4 demonstrate that the MDY7-3PTB model achieved a mean intersection over union of 86.61%, which outperforms other models such as DeeplabV3+_Xception, DeeplabV3+_MobileNetV2, Unet_Resnet50, Unet_VGG, PSPNet_Resnet50, and PSPNet_MobileNetV2, by 3.55%, 6.18%, 8.61%, 2.96%, 6.68%, and 12.99%, respectively. The MDY7-3PTB model also achieved a mean pixel precision of 93.01%, which increased by 2.10%, 4.12%, 5.30%, 1.79%, 8.09%, and 14.1%,

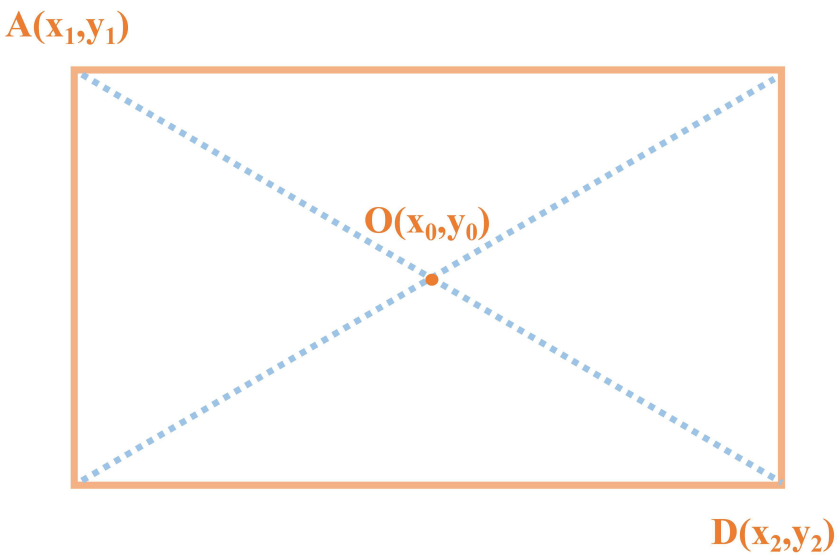



FIGURE 9
Calculation method for obtaining the center point of a rectangular box.

TABLE 4 Model performance table under multiple indicators.

| Network Model | mIoU (%) | mPA (%) | mRecall (%) | Params (MB) | FPS |
|---|---|---|---|---|---|
| PSPNet_MobileNetV2 | 73.62 ± 0.41 | 78.91 ± 0.42 | 87.66 ± 0.23 | 19.3 | 75.63 |
| PSPNet_Resnet50 | 79.93 ± 0.57 | 84.92 ± 0.50 | 91.07 ± 0.38 | 178.5 | 37.78 |
| Unet_VGG | 83.65 ± 0.45 | 91.22 ± 0.37 | 89.64 ± 0.29 | 94.9 | 15.11 |
| Unet_Resnet50 | 78.00 ± 0.36 | 87.71 ± 0.26 | 84.98 ± 0.36 | 167.4 | 23.39 |
| DeeplabV3+_MobileNetV2 | 80.43 ± 0.40 | 88.89 ± 0.40 | 87.26 ± 0.36 | 22.4 | 63.12 |
| DeeplabV3+_Xception | 83.06 ± 0.24 | 90.91 ± 0.36 | 89.15 ± 0.24 | 209.7 | 23.84 |
| MDY7-3PTB | 86.61 ± 0.63 | 93.01 ± 0.25 | 91.78 ± 0.65 | 22.5 | 59.01 |

respectively. Moreover, the MDY7-3PTB model showed a mean recall of 91.78%, which increased by 2.63%, 4.52%, 6.80%, 2.14%, 0.71%, and 4.12%, respectively. Additionally, the proposed method significantly reduced the number of model parameters and improved the FPS, resulting in faster and more accurate segmentation. To comprehensively evaluate the segmentation performance of the MDY7-3PTB model, the segmentation results of tea buds and background was compared and the results are presented in Table 5. The IoU, PA, and Recall in Table 5 represent the mean and standard deviation of the model's results from 10 tests.

To further compare the performance of each method on tea bud segmentation, Table 5 presents the results of intersection over union, pixel precision, and recall for both tea buds and background. It can be observed that the background has the highest values for intersection over union, pixel precision, and recall due to its large proportion in the image and small proportion of tea buds. Overall, MDY7-3PTB has higher values for intersection over union, pixel precision, and recall on both tea buds and background than the other comparison models, indicating that the proposed method is effective in improving the segmentation performance for each category. To provide a more clear and intuitive comparison of the tea bud prediction results of each model, this paper compiled the original image and segmentation maps of each network model. By adjusting parameters such as learning rate, threshold, and iteration, continuous training and optimization were performed to obtain the prediction results of each model. The comparison results are shown in Figure 10. Figure 10 select a representative picture for display.

Based on the comparison chart above, it can be observed that MDY7-3PTB, DeeplabV3+Xception, and Unet_VGG models perform well in the actual tea bud segmentation effect, especially in terms of detailed contour aspects and small tea bud segmentation. Moreover, compared to DeeplabV3+Xception and Unet_VGG models for tea bud contour segmentation, MDY7-3PTB exhibits higher precision in tea bud segmentation, which is closer to the actual number of tea buds in the original image. However, other models have some segmentation defects. PSPNet MobileNetV2 and PSPNet Resnet50 models fail to segment many tea buds accurately, and some of the segmented tea bud backgrounds is not completely removed, leading to low tea bud segmentation precision. Unet_Resnet50 and DeeplabV3+_MobileNetV2 models suffer from segmentation errors, where some tea leaves are erroneously segmented as tea buds, and the background of a few tea buds is not entirely removed.

Based on the comparison of the result data and image analysis above, it can be concluded that the proposed MDY7-3PTB model has significantly improved the segmentation precision, and the model is also lightweight and more suitable for actual tea bud segmentation needs.

TABLE 5 The results of different segmentation models on tea buds and backgrounds under multiple metrics.

| Network Model | IoU (%) | | PA (%) | | Recall (%) | |
|---|---|---|---|---|---|---|
| | Tea Buds | Background | Tea Buds | Background | Tea Buds | Background |
| PSPNet_MobileNetV2 | 50.03 ± 0.76 | 97.21 ± 0.08 | 58.65 ± 0.83 | 99.16 ± 0.02 | 77.30 ± 0.42 | 98.02 ± 0.07 |
| PSPNet_Resnet50 | 61.93 ± 1.08 | 97.92 ± 0.07 | 70.50 ± 0.90 | 99.33 ± 0.03 | 83.56 ± 0.72 | 98.58 ± 0.06 |
| Unet_VGG | 69.11 ± 0.85 | 98.20 ± 0.06 | 83.44 ± 0.72 | 98.99 ± 0.03 | 80.08 ± 0.55 | 99.19 ± 0.04 |
| Unet_Resnet50 | 58.60 ± 0.67 | 97.39 ± 0.08 | 76.94 ± 0.52 | 98.48 ± 0.06 | 71.09 ± 0.71 | 98.88 ± 0.03 |
| DeeplabV3+_MobileNetV2 | 63.09 ± 0.75 | 97.78 ± 0.06 | 79.15 ± 0.77 | 98.76 ± 0.03 | 75.67 ± 0.67 | 98.98 ± 0.05 |
| DeeplabV3+_Xception | 68.02 ± 0.44 | 98.12 ± 0.05 | 82.89 ± 0.74 | 98.94 ± 0.04 | 79.13 ± 0.51 | 99.17 ± 0.04 |
| MDY7-3PTB | 74.66 ± 1.17 | 98.58 ± 0.09 | 86.81 ± 0.48 | 99.21 ± 0.08 | 84.21 ± 1.30 | 99.36 ± 0.03 |

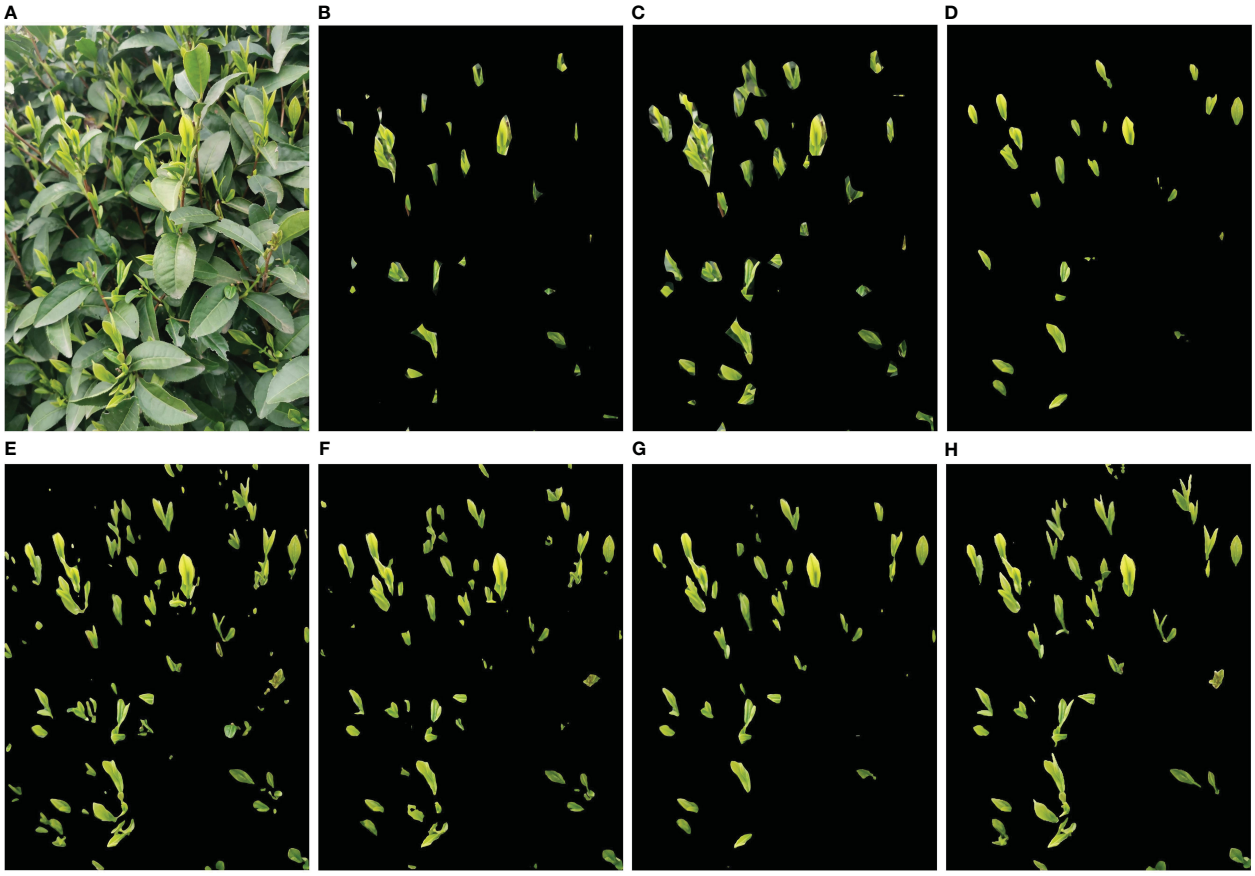



FIGURE 10
Examples of segmentation results for different models. **(A)** Original image; **(B)** PSPNet_MobileNetV2; **(C)** PSPNet_Resnet50; **(D)** Unet_VGG; **(E)** Unet_Resnet50; **(F)** DeeplabV3+_MobileNetV2; **(G)** DeeplabV3+_Xception; **(H)** MDY7-3PTB.

## 4.2 Detection performance of the MDY7-3PTB model

After the dataset in this study is segmented and then input the segmented dataset into YOLOv3, YOLOv4, YOLOv5s, YOLOxm, and YOLOv7 detection models to create the corresponding MDY3-3PTB, MDY4-3PTB, MDY5s-3PTB, MDYxm-3PTB, and MDY7-3PTB models. The YOLOv7 module represents direct detection of the original image. The MDY3-3PTB module combines the high-precision segmentation capability of improved DeeplabV3+ module and the rapid detection of YOLOv3, the MDY4-3PTB module combines the high-precision segmentation capability of improved DeeplabV3+ module and the rapid detection of YOLOv4, and the MDY5s-3PTB module combines the high-precision segmentation capability of improved DeeplabV3+ module and the rapid detection of YOLOv5s, and so on. This paper compares their evaluation metrics such as precision, recall, F1 score, and mAP_(@0.5), as shown in Table 6. The "mAP@0.5" in Table 6 refers to the mAP value calculated at a confidence threshold of 0.5.

After comparing the results presented in Table 6, it is evident that the overall performance of the MDY7-3PTB model proposed in this paper is superior to other models. Specifically, the precision of MDY7-3PTB is 97.27%, which is higher than YOLOv7, MDY3-3PTB, MDY4-3PTB, MDY5s-3PTB, MDYxm-3PTB, and MDY7-3PTB models by 11.63%, 23.05%, 27.15%, 7.85%, and 1.76% respectively. The recall of MDY7-3PTB is 89.41%, which represents an improvement of 17.91%, 21.22%, 36.15%, 27.48%, and -1.74% respectively. In terms of F1 score, MDY7-3PTB achieves a value of 93.17%, which is higher than YOLOv7, MDY3-3PTB, MDY4-3PTB, MDY5s-3PTB, and MDYxm-3PTB by 15.24%, 22.09%, 32.63%, 19.99%, and -0.11% respectively. Moreover, in terms of the mean average precision, MDY7-3PTB is 93.52%, which represents an improvement of 13.7%, 30.04%, 39.77%, 18.66%, and 1.9% respectively.

To facilitate a clearer and more intuitive comparison of the tea bud picking point prediction results from each model, this paper presents a collation of the prediction results from each network model. By adjusting parameters such as learning rate, threshold, and iteration, continuous training and optimization were conducted, and ultimately, the prediction results of each model were obtained. The comparison of their effects is presented in Figure 11. Figure 11 select representative pictures for display.

Based on the comparison chart above, it can be concluded that the MDY7-3PTB model has the best performance in actual tea bud picking point detection. The MDY7-3PTB model has no missed or erroneous detections and high confidence, while the other models have their own limitations. The MDYxm-3PTB model has a detection performance similar to that of MDY7-3PTB, but with a

TABLE 6 Comparison of segmentation performance under multiple metrics.

| Network Model | P | R | F1 | $mAP_{@0.5}$ |
|---|---|---|---|---|
| YOLOv7 | 85.64 | 71.50 | 77.93 | 79.82 |
| MDY3-3PTB | 74.22 | 68.19 | 71.08 | 63.48 |
| MDY4-3PTB | 70.12 | 53.26 | 60.54 | 53.75 |
| MDY5s-3PTB | 89.42 | 61.93 | 73.18 | 74.86 |
| MDYxm-3PTB | 95.51 | 91.15 | 93.28 | 91.62 |
| MDY7-3PTB | 97.27 | 89.41 | 93.17 | 93.52 |

few missed detections and a lower confidence level. The MDY5s-3PTB, MDY4-3PTB, and MDY3-3PTB models have worse performance in all aspects, with many missed detections and tea bud picking points detected with relatively low confidence. The YOLOv7 model, which directly detects tea buds from the original images, is much less effective than MDY7-3PTB and other models, and shows significant missed detection, which can be attributed to the similar color of tea buds and tea leaves in complex environments. Therefore, it is better to remove the background and perform data enhancement before detection. Furthermore, from the predicted images of each model, it is observed that the picking points of tea buds captured from side views are easier to detect, while the picking points of tea buds captured from top views are more challenging to detect, which requires further investigation in future studies.

This paper presents experimental data on the training conditions of a model for tea bud picking point detection, using the change of mean average precision (mAP) and loss function as indicators. A scatter diagram is used to visualize the results. The proposed MDY7-3PTB model is compared to five mainstream YOLO detection models, and its advantages in detection precision are demonstrated. The other models, including MDY3-3PTB, MDY4-3PTB, MDY5s-3PTB, MDYxm-3PTB, and YOLOv7, have various shortcomings in detection. As the iteration times increase, the mAP of MDY7-3PTB significantly outperforms that of other models. Compared to the original YOLOv7, the MDY7-3PTB model converges much faster due to the early removal of background interference and the addition of the CBAM attention mechanism, which enhances the feature distribution weights of objects in both the spatial and channel dimensions and eliminates the interference of irrelevant features. Although MDYxm-3PTB is second only to MDY7-3PTB in all aspects of metrics, it has slower convergence and larger early fluctuations. These findings are shown in Figures 12, 13.

In summary, based on the result data and image comparisons presented above, it is evident that the MDY7-3PTB model proposed in this paper has significantly improved detection precision. This provides a solid theoretical foundation for future developments in tea bud picking.

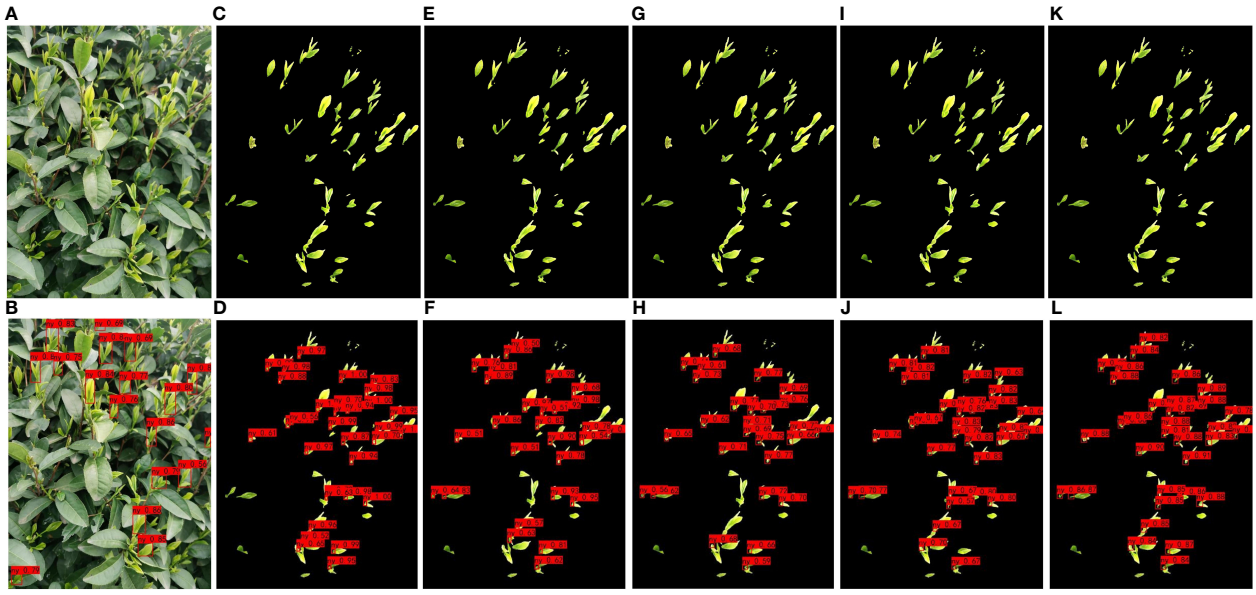



FIGURE 11
Examples of detection results of different models for segmented datasets; **(A, B)** YOLOv7; **(C, D)** MDY3-3PTB; **(E, F)** MDY4-3PTB; **(G, H)** MDY5s-3PTB; **(I, J)** MDYxm-3PTB; **(K, L)** MDY7-3PTB.

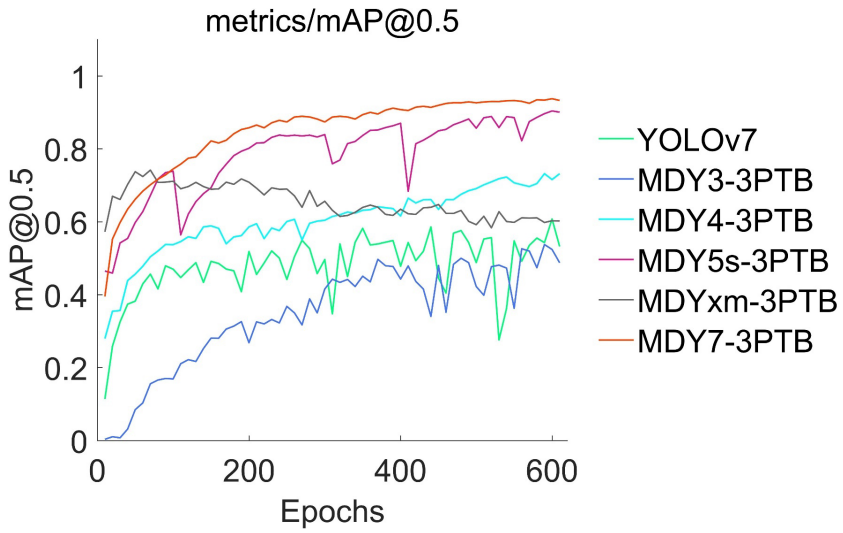



FIGURE 12
Accuracy variation of six object detectors.

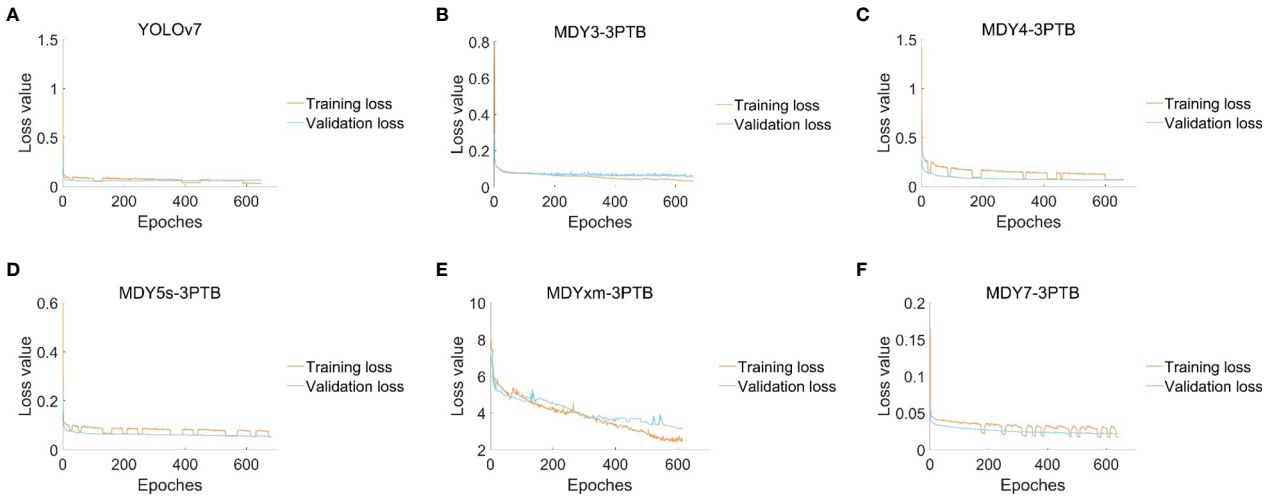



FIGURE 13
**(A)** The loss changes of the YOLOv7 model; **(B)** The loss changes of the MDY3-3PTB model; **(C)** The loss changes of the MDY4-3PTB model; **(D)** The loss changes of the MDY5s-3PTB model; **(E)** The loss changes of the MDYxm-3PTB model; **(F)** The loss changes of the MDY7-3PTB model.

## 4.3 Positioning performance of the MDY7-3PTB model

To provide a theoretical basis for the tea-picking robot, this paper uses MDY7-3PTB to segment and detect the tea bud picking points, and outputs the coordinates of the picking points by taking the geometric center of the rectangular frame and restoring their background, achieving accurate localization of the tea bud picking points. The two-dimensional coordinate positioning is shown in Figure 14, where the set in parentheses indicates the position coordinates of the tea bud picking point relative to the original image. Excluding the tea buds that were not segmented due to focusing issues, the number of tea bud picking points marked after segmentation was 167, with 161 correct positioning and a positioning precision of 96.41%. This greatly improved the precision compared to direct detection and subsequent positioning, thereby enhancing the precision of subsequent tea bud picking.

# 5 Discussion

As is well known, some researchers have used the YOLO series detection algorithm for detecting and classifying tea buds, as well as other crops such as citrus (Chen et al., 2022), apple (Sun et al., 2022), grape (Yang et al., 2022), lychee (Xie et al., 2022), and camellia oleifera fruit (Wu et al., 2022). These crops have

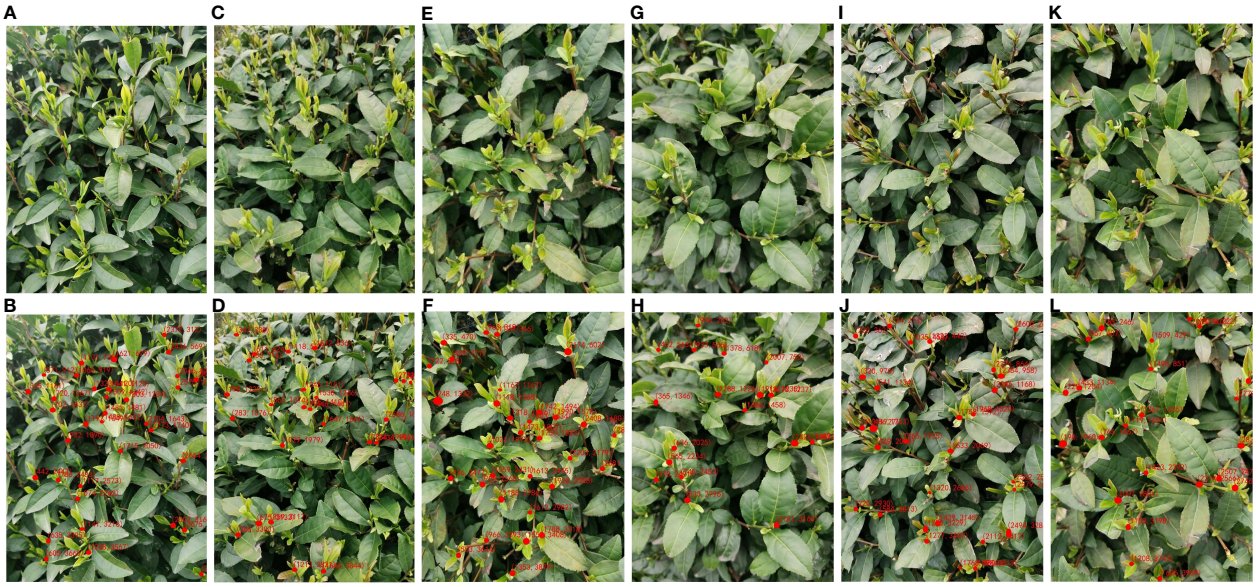



FIGURE 14
Example of results of MDY7-3PTB model for locating picking points of tea buds in complex environments; **(A, C, E, G, I, K)** for original images; **(B, D, F, H, J, L)** for tea bud picking points identified.

significantly different colors from their backgrounds, making it easy to accurately locate their positions. In addition, these crops are round in shape and different from the surrounding background, such as leaves, making it possible to directly use the detection algorithm for object detection with high accuracy. However, tea buds have similar colors and shapes to their background, so using the detection algorithm directly for detection may result in problems such as missed detection, false detection, low detection speed, and low accuracy.

Therefore, to eliminate background interference and improve the accuracy of tea bud localization, this study first performs segmentation on the tea buds, separating them from the background. Then, object detection is applied to the segmented tea buds to identify the picking points, followed by the final localization of the tea buds. The addition of an attention mechanism module is an effective way to enhance model performance (Chen et al., 2022; Khan et al., 2022). The proposed improved DeepLabV3+ model based on attention mechanism can effectively accomplish the segmentation task. Furthermore, compared to directly performing tea bud object detection (Li et al., 2023) or segmentation alone, the proposed approach of combining object segmentation and detection in this study achieves better results (Guo et al., 2023; Meng et al., 2023). However, there are still a very few tea buds whose coordinates have not been correctly identified, which can be attributed to the following reasons summarized as follows: Firstly, the tea buds in the images are excessively blurred, leading to segmentation or detection failures. Secondly, several tea buds overlap with each other, causing incomplete segmentation or detection. Thirdly, some tea buds only partially appear in the images, resulting in recognition errors.

Currently, Zhang et al. (2021) use a binocular depth camera to capture images of fruit in the field and design a fruit spatial positioning system. Yan et al. (2022) use the MR3P-TS model to locate tea picking points and provide cutting angle recommendations. Xu et al. (2022) proposed a combined multi-point picking scheme and selectively designed the size of the tip of the bud picker. Chen et al. (2022) apply robot technology and deep learning to develop a computer vision system for intelligent tea picking, providing theoretical support for the intelligent picking of tea buds. The above research indicates that achieving automated picking of tea buds requires particular emphasis on the localization of picking points. The novel method proposed in this study for tea bud picking point localization can effectively accomplish this task, providing a theoretical foundation for the development of tea-picking robots.

## 6 Conclusion

The tea plantation background is complex, and manual picking requires a lot of manpower and resources. In this paper, we propose a MDY7-3PTB model based on the high-precision segmentation ability of DeepLabv3+ and the fast detection ability of YOLOv7, which is mainly used for tea leaf detection and localization in tea plantations. The model consists of three stages. The first stage uses the Focal Loss function to correct the class imbalance in the original dataset, and then uses the CBAM attention mechanism and lightweight DeepLabv3+ network to segment the original tea leaf dataset. The second stage uses YOLOv7 to detect the tea picking points after segmentation. The third stage uses the method of taking the center of the rectangular box to accurately locate the two-dimensional coordinates of the tea picking points. In testing, the proposed MDY7-3PTB model was compared with other object segmentation or detection models. The model achieved an

average intersection over union (IoU) of 86.61%, an average pixel accuracy of 93.01%, and an average recall of 91.78% on the tea shoot segmentation dataset. In addition, for tea plucking point recognition and localization, the model showed significant improvements compared to existing mainstream detection models, with a mean average precision of 93.52%, a weighted average of precision and recall of 93.17%, a precision of 97.27%, and a recall of 89.41%. These results provide a strong theoretical foundation for the future of tea plucking, demonstrating significant advancements in all aspects of the detection process. In future research, the goal is to develop a system for annotating the three-dimensional coordinates of tea picking points using a binocular depth camera, which can be combined with mechanical structures to provide further theoretical basis for intelligent tea picking and achieve fine tea picking. Additionally, Future research will focus on improving the YOLOv7 model by adding attention mechanism modules, reducing the number of parameters, and increasing the detection speed.

## Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

## Author contributions

FZ: Methodology, Analysis, Suggestion, Writing-review. SX: Conceptualization, Investigation, Writing-original draft, Writing-review and editing. CD: provided the tea buds data. LY: Methodology, Analysis, Writing-review. YX and ZZ: Picture Drawing and Artistic Design. HS and FC: Validation, Resources, Supervision, Project administration, Writing-review and editing, Funding acquisition. All authors contributed to the article and approved the submitted version.

## Funding

The Project was supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. LQ21C130007, and the Open Fund of Key Laboratory of Transplanting Equipment and Technology of Zhejiang Province under Grant No. 2023E10013-05.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

Bochkovskiy, A., Wang, C., and Liao, H. (2020). YOLOv4: optimal speed and accuracy of object detection. *arXiv* 2020, arXiv:2004.10934. doi: 10.48550/arXiv.2004.10934

Chen, C., Lu, J., Zhou, M., Yi, J., Liao, M., and Gao, Z. (2022). A YOLOv3-based computer vision system for identification of tea buds and the picking point. *Comput. AND Electron. IN Agric.* 198, 107116. doi: 10.1016/j.compag.2022.107116

Du, Z., Hu, Y., and Wang, S. (2018). Simulation and experiment of reciprocating cutter kinematic of portable tea picking machine. *Trans. Chin. Soc Agricult. Machinery* 49, 221–226. doi: 10.6041/j.issn.1000-1298.2018.S0.029

Fu, H., Meng, D., Li, W., and Wang, Y. (2021). Bridge crack semantic segmentation based on improved deeplabv3+. *J. OF Mar. Sci. AND Eng.* 9 (6), 671. doi: 10.3390/jmse9060671

Glenn, J. (2020) *yolov5. Git Code*. Available at: https://github.com/ultralytics/yolov5 (Accessed 20 November 2022).

Gui, Z., Chen, J., Li, Y., Chen, Z., Wu, C., and Dong, C. (2023). A lightweight tea bud detection model based on Yolov5. *Comput. Electron. Agric.* 205, 107636. doi: 10.1016/j.compag.2023.107636

Guo, S., Yoon, S.-C., Li, L., Wang, W., Zhuang, H., Wei, C., et al. (2023). Recognition and positioning of fresh tea buds using YOLOv4-lighted + ICBAM model and RGB-D sensing. *Agriculture* 13 (3), 518. doi: 10.3390/agriculture13030518

Hai, V., Le, T., Hai, T., and Thuy, N. (2014). A vision-based method for automatizing tea shoots detection. *IEEE Int. Conf. Image Process.* 2013, 3775–3779. doi: 10.1109/ICIP.2013.6738778

Han, Y., Xiao, H., Qin, G., Song, Z., Ding, W., and Mei, S. (2014). Developing situations of tea plucking machine. *Engineering* 06, 268–273. doi: 10.4236/Eng.2014.66031

Jiang, K., Xie, T., Yan, R., Wen, X., Li, D., Jiang, H., et al. (2022). An attention mechanism-improved YOLOv7 object detection algorithm for hemp duck count estimation. *AGRICULTURE-BASEL* 12 (10), 1659. doi: 10.3390/agriculture12101659

Khan, R., Wong, W., Ullah, I., Algarni, F., Ul Haq, M., Barawi, b., et al. (2022). Evaluating the efficiency of CBAM-resnet using Malaysian sign language. *CMC-COMPUTERS MATERIALS CONTINUA* 71 (2), 2755–2772. doi: 10.32604/cmc.2022.022471

Li, Y., He, L., Jia, J., Lv, J., Chen, J., Qiao, X., et al. (2021). In-field tea shoot detection and 3d localization using an rgb-d camera. *Comput. Electron. Agric.* 185, 106149. doi: 10.1016/j.compag.2021.106149

Li, Y., Ma, R., Zhang, R., Cheng, Y., and Dong, C. (2023). A tea buds counting method based on YOLOv5 and kalman filter tracking algorithm. *Plant Phenomics* 5, 0030. doi: 10.34133/plantphenomics.0030

Lin, G., Tang, Y., Zou, X., Xiong, J., and Fang, Y. (2019). Color-, depth-, and shape-based 3d fruit detection. *Precis. Agriculture.* 21 (1), 1–17. doi: 10.1007/s11119-019-09654-w

Liu, X., Zhao, D., Jia, W., Ji, W., and Sun, Y. (2019). A detection method for apple fruits based on color and shape features. *IEEE Access.* 7, 67923–67933. doi: 10.1109/ACCESS.2019.2918313

Lu, J., Yang, Z., Sun, Q., Gao, Z., and Ma, W. (2023). A machine vision-based method for tea buds segmentation and picking point location used on a cloud platform. *Agronomy* 13 (6), 1537. doi: 10.3390/agronomy13061537

Meng, J., Wang, Y., Zhang, J., Tong, S., Chen, C., Zhang, C., et al. (2023). Tea bud and picking point detection based on deep learning. *Forests* 14 (6), 1188. doi: 10.3390/f14061188

Nie, S., Wei, Y., Zhao, F., Dong, Y., Chen, Y., Li, Q., et al. (2022). A dual deep neural network for auto-delineation in cervical cancer radiotherapy with clinical validation. *Radiat. Oncol.* 17 (1), 1–9. doi: 10.1186/s13014-022-02157-5

Qi, F., Xie, Z., Tang, Z., and Chen, H. (2021). Related study based on otsu watershed algorithm and new squeeze-and-excitation networks for segmentation and level classification of tea buds. *Neural Process Lett.* 53, 2261–2275. doi: 10.1007/s11063-021-10501-1

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. (IEEE), 779–788.

Redmon, J., and Farhadi, A. (2017). "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. (IEEE), 7263–7271.

Redmon, J., and Farhadi, A. (2018). YOLOv3: an incremental improvement. *arXiv* 2018, arXiv:1804.02767. doi: 10.48550/arXiv.1804.02767

Sun, L., Hu, G., Chen, C., Cai, H., Li, C., Zhang, S., et al. (2022). Lightweight apple detection in complex orchards using YOLOV5-PRE. *HORTICULTURAE* 8 (12), 1169. doi: 10.3390/horticulturae8121169

Wang, C., Bochkovskiy, A., and Liao, H. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv Prepr.* 2022, arXiv:2207.02696. doi: 10.48550/arXiv.2207.02696

Wang, D., He, D., Song, H., Liu, C., and Xiong, H. (2019). Combining SUN-based visual attention model and saliency contour detection algorithm for apple image segmentation. *MULTIMEDIA Tools AND Appl.* 78 (13), 17391–17411. doi: 10.1007/s11042-018-7106-y

Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S. (2018). *CBAM:Convolutional block attention module* Vol. 2018 (Cham, Switzerland: Springer), 3–19.

Wu, D., Jiang, S., Zhao, E., Liu, Y., Zhu, H., Wang, W., et al. (2022). Detection of camellia oleifera fruit in complex scenes by using YOLOv7 and data augmentation. *Appl. SCIENCES-BASEL* 12 (22), 11318. doi: 10.3390/app122211318

Wu, W., Liu, G., Liang, K., and Zhou, H. (2023). Inner cascaded U2-net: an improvement to plain cascaded U-net. *CMES-COMPUTER MODELING IN Eng. Sci.* 134 (2), 1323–1335. doi: 10.32604/cmes.2022.020428

Wu, G., Zhu, Q., Huang, M., Guo, Y., and Qin, J. (2019). Automatic recognition of juicy peaches on trees based on 3d contour features and colour data. *Biosyst. ENGINEERING.* 188, 1–13. doi: 10.1016/j.biosystemseng.2019.10.002

Xia, F., Xie, X., Wang, Z., Jin, S., Yan, K., and Ji, Z. (2022). A novel computational framework for precision diagnosis and subtype discovery of plant with lesion. *Front. Plant Sci.* 12. doi: 10.3389/fpls.2021.789630

Xie, J., Peng, J., Wang, J., Chen, B., Jing, T., Sun, D., et al. (2022). Litchi detection in a complex natural environment using the YOLOv5-litchi model. *AGRONOMY-BASEL* 12 (12), 3054. doi: 10.3390/agronomy12123054

Xie, X., Xia, F., Wu, Y., Liu, S., Yan, K., Xu, H., et al. (2023). A novel feature selection strategy based on Salp swarm algorithm for plant disease detection. *Plant Phenomics.* 5, 39. doi: 10.34133/plantphenomics.0039

Xu, L., Xie, Y., Chen, X., Chen, T., Kang, Z., Huang, P., et al. (2022). Design of an efficient combined multipoint picking scheme for tea buds. *Front. IN Plant Sci.* 13, , 1042035. doi: 10.3389/fpls.2022.1042035

Xu, W., Zhao, L., Li, J., Shang, S., Ding, X., and Wang, T. (2022). Detection and classification of tea buds based on deep learning. *Comput. Electron. Agric.* 192, 106547. doi: 10.1016/j.compag.2021.106547

Yan, L., Wu, K., Lin, J., Xu, X., Zhang, J., Zhao, X., et al. (2022). Identification and picking point positioning of tender tea shoots based on MR3P-TS model. *Front. IN Plant Sci.* 13, 962391. doi: 10.3389/fpls.2022.962391

Yang, H., Chen, L., Chen, M., Ma, Z., and Li, X. (2019). Tender tea shoots recognition and positioning for picking robot using improved yolo-v3 model. *IEEE ACCESS.* 7, 180998–181011. doi: 10.1109/ACCESS.2019.2958614

Yang, R., Hu, Y., Yao, Y., Gao, M., and Liu, R. (2022). Fruit target detection based on BCo-YOLOv5 model. *MOBILE Inf. Syst* 2022, 8457173. doi: 10.1155/2022/8457173

Yu, X., and He, Y. (2018). Optimization of tea-leaf saponins water extraction and relationships between their contents and tea (Camellia sinensis) tree varieties. *Food Sci. AND Nutr.* 6 (6), 1734–1740. doi: 10.1002/fsn3.724

Yuan, W., Wang, J., and Xu, W. (2022). Shift pooling PSPNet: rethinking PSPNet for building extraction in remote sensing images from entire local feature pooling. *Remote Sens.* 14 (19), 4889. doi: 10.3390/rs14194889

Zhang, X., Gao, Q., Pan, D., Cao, P., and Huang, D. H. (2021). Research on spatial positioning system of fruits to be picked in field based on binocular vision And Ssd model. *J. Phys. Conf. Ser.* 1748, 42011. doi: 10.1088/1742-6596/1748/4/042011

Zheng, B., Shen, Y., Luo, Y., Fang, X., Zhu, S., Zhang, J., et al. (2022). Automated measurement of the disc-fovea angle based on DeepLabv3+. *Front. IN Neurol.* 13, 949805. doi: 10.3389/fneur.2022.949805

Zheng, H., Wang, R., Yu, Z., Wang, N., Gu, Z., and Zheng, B. (2017a). Automatic plankton image classification combining multiple view features via multiple kernel. *BMC Bioinf.* 18, 1–18. doi: 10.1186/s12859-017-1954-8

Zheng, H., Wang, N., Yu, Z., and Zheng, B. (2017b). Robust and automatic cell detection and segmentation from microscopic images of non-setae phytoplankton species. *IET IMAGE PROCESSING.* 11 (11), 1077–1085. doi: 10.1049/iet-ipr.2017.0127

Zhu, Y., Wu, C., Tong, J., Chen, J., He, L., Wang, R., et al. (2021). Deviation tolerance performance evaluation and experiment of picking end effector for famous tea. *AGRICULTURE-BASEL* 11 (2), 128. doi: 10.3390/agriculture11020128

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 02 October 2023
DOI 10.3389/fpls.2023.1243822

Check for updates

OPEN ACCESS

EDITED BY
Yongliang Qiao,
University of Adelaide, Australia

REVIEWED BY
Preeta Sharan,
The Oxford College of Engineering, India
Jun Liu,
Shandong Provincial University Laboratory for Protected Horticulture, China
Catarina Silva,
University of Coimbra, Portugal

*CORRESPONDENCE
Hyongsuk Kim
hskim@jbnu.ac.kr
Sook Yoon
syoon@mokpo.ac.kr

RECEIVED 21 June 2023
ACCEPTED 13 September 2023
PUBLISHED 02 October 2023

CITATION
Dong J, Fuentes A, Yoon S, Kim H, Jeong Y and Park DS (2023) A New Deep Learning-based Dynamic Paradigm Towards Open-World Plant Disease Detection.
*Front. Plant Sci.* 14:1243822.
doi: 10.3389/fpls.2023.1243822

COPYRIGHT
© 2023 Dong, Fuentes, Yoon, Kim, Jeong and Park. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# A New Deep Learning-based Dynamic Paradigm Towards Open-World Plant Disease Detection

Jiuqing Dong[1,2], Alvaro Fuentes[1,2], Sook Yoon[3]*, Hyongsuk Kim[1,2]*, Yongchae Jeong[1] and Dong Sun Park[1,2]

[1]Department of Electronic Engineering, Jeonbuk National University, Jeonju, Republic of Korea, [2]Core Research Institute of Intelligent Robots, Jeonbuk National University, Jeonju, Republic of Korea, [3]Department of Computer Engineering, Mokpo National University, Muan, Republic of Korea

Plant disease detection has made significant strides thanks to the emergence of deep learning. However, existing methods have been limited to closed-set and static learning settings, where models are trained using a specific dataset. This confinement restricts the model's adaptability when encountering samples from unseen disease categories. Additionally, there is a challenge of knowledge degradation for these static learning settings, as the acquisition of new knowledge tends to overwrite the old when learning new categories. To overcome these limitations, this study introduces a novel paradigm for plant disease detection called open-world setting. Our approach can infer disease categories that have never been seen during the model training phase and gradually learn these unseen diseases through dynamic knowledge updates in the next training phase. Specifically, we utilize a well-trained unknown-aware region proposal network to generate pseudo-labels for unknown diseases during training and employ a class-agnostic classifier to enhance the recall rate for unknown diseases. Besides, we employ a sample replay strategy to maintain recognition ability for previously learned classes. Extensive experimental evaluation and ablation studies investigate the efficacy of our method in detecting old and unknown classes. Remarkably, our method demonstrates robust generalization ability even in cross-species disease detection experiments. Overall, this open-world and dynamically updated detection method shows promising potential to become the future paradigm for plant disease detection. We discuss open issues including classification and localization, and propose promising approaches to address them. We encourage further research in the community to tackle the crucial challenges in open-world plant disease detection. The code will be released at https://github.com/JiuqingDong/OWPDD.

KEYWORDS

**plant disease detection, incremental learning, open-world detection, out-of-distribution detection, dynamic paradigm**

# 1 Introduction

Accurate and timely detection and diagnosis of plant diseases are crucial for preserving crop health and increasing agricultural productivity. However, traditional methods of plant disease detection primarily rely on skilled agricultural professionals who diagnose diseases based on visual symptoms and pathologic characteristics of pathogens. These methods suffer from limitations such as subjectivity, prolonged diagnosis time, and dependence on experienced experts (Dong et al., 2022). To address these limitations of traditional methods, plant disease detection based on image analysis and artificial intelligence has emerged as a hot research topic (Shoaib et al., 2023; Xu et al., 2023). This emerging approach utilizes images captured from various plant parts such as leaves and stems, followed by computer algorithms for image analysis and recognition, enabling automated detection and diagnosis of plant diseases. This method not only enhances the accuracy and efficiency of detection but also allows non-experts to participate in plant disease monitoring and diagnosis (Panchal et al., 2023).

A substantial body of published work attests to the success of deep learning in plant disease detection tasks (Fuentes et al., 2018; Li et al., 2020; Nazki et al., 2020; Singh et al., 2020; Fenu and Malloci, 2021; Fuentes et al., 2021; Qiao et al., 2022b). However, existing studies focus on fixed disease categories of specific species with all available annotations during the training phase. This training strategy is known as closed-set learning (Xiong et al., 2019). In this case, the model is more likely to classify suspicious regions as one of the categories it has already learned, rather than indicating the presence of an abnormal disease type (Du et al., 2022b). We show the potential risks associated with closed-set learning in Figure 1A. Note that "known classes" refer to the classes present in the training dataset, while "unknown classes" refer to the classes that exist in real-world scenarios but are either absent or unannotated in the training dataset.

In the concept of plant stress, unknown diseases may result in a large economic loss, and recognizing them is thus one of the fundamental demands (Geng et al., 2020). Therefore, unknown disease detection is more useful in most practical scenarios. The learning paradigm that can detect unknown classes is known as open-set learning (Vaze et al., 2021). Figure 1B illustrates the open-set learning paradigm, which allows the model to detect instances that are currently unknown to the model. Developmental psychology (Livio, 2017) has revealed that the ability to recognize the unknown is crucial for stimulating curiosity, which in turn fuels the desire to learn new things. In the open-set learning paradigm, when the model detects unknown diseases and provides feedback to domain experts, it is important for the domain experts to pay attention to these disease samples and assign them appropriate category labels. This allows the model to further learn about these new diseases.

To learn these new diseases, one naive learning strategy is to combine the new and old data together and let the model learn again. However, as the number of tasks increases, the accumulated data volume becomes significantly large, resulting in high training costs. This approach may be feasible in the short term but is not sustainable as a long-term training strategy. Another learning method is to fine-tune the old model using new data. In this way, the model will quickly adapt to the new task but there is a risk of losing the ability to detect previously known classes. This prompts us to propose a new challenge: a new paradigm should be capable of recognizing instances of unknown diseases as unknown and gradually learning these unknown categories through incremental learning. Figure 1C illustrates the workflow of this new paradigm.

Plant growth is a dynamic process, and plant disease dynamics are more complex than we imagined. During the plant growth cycle monitoring, unexpected diseases and pests are likely to emerge. Simultaneously, collecting all the existing plant diseases is difficult and even impossible for real-world applications (Xu et al., 2023). Given the dynamic nature of our world, the setup of open-world plant disease detection is more aligned with real-world applications compared to existing closed-set learning and open-set learning settings. Therefore, we need to introduce a new paradigm to continuously learn these unknown diseases instead of learning them all at once. In this paradigm, the model can detect unknown diseases and provide feedback to domain experts. Then, experts will label these unknown diseases. As and when more information about the identified unknown classes becomes available, the system should be able to incorporate them into its existing knowledge base. This iterative learning process will cycle throughout the model's lifecycle. In this paper, we propose an open-world detector for plant disease detection, aiming to achieve this goal.

The key contributions of our work as follows:

1. We introduce the concept of open-world problem formulation into plant disease detection for the first time, enabling a closer simulation of real-world application scenarios. Unlike all existing plant disease detectors, it dynamically expands the learned categories and actively responds to unknown diseases.
2. We introduce an unknown-aware region proposal network (UA-RPN) and conducted pre-training on various datasets. We find that the model pre-trained on LVIS (Large Vocabulary Instance Segmentation) (Gupta et al., 2019) dataset can exhibit superior performance across different experimental setups. Additionally, we propose a class-agnostic region of interest (ROI) head, which significantly improved the recall rate for unknown classes. Interestingly, the model trained on a dataset of tomato leaf diseases could even detect diseases in paprika fruit.
3. Our method also achieves class incremental detection of plant diseases. Additionally, we discuss the open issues associated with open-world plant disease detection and provide promising solutions. We believe that this open-world and dynamically updated detection method can become a new paradigm for future plant disease detection, and we encourage the research community to explore and address these open challenges.

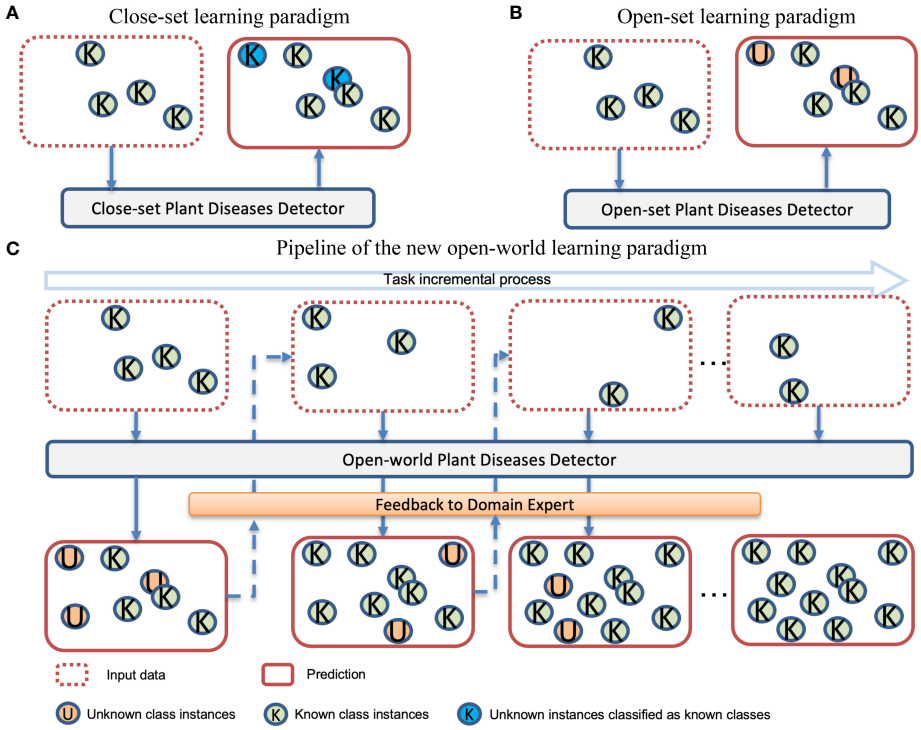



FIGURE 1
Comparison of three different learning paradigms. **(A)** Closed-set-based models detect unknown diseases as known diseases; **(B)** Open-set based models can detect unknown diseases but do not learn them; **(C)** Open-world detector learns the known diseases and also autonomously detects unknown diseases. The identified unknown diseases are then provided as feedback to domain experts, who annotate these newly discovered labels. This valuable information is incorporated into the model during subsequent tasks, allowing it to adaptively update itself with new knowledge.

Section 2 provides a detailed review of the deep learning techniques employed for plant anomaly detection and existing open-set and open-world deep learning approaches. Section 3 comprehensively describes the problem formulation, methodology, and evaluation framework of the novel paradigm we have introduced. In Section 4, experimental results are presented to demonstrate the effectiveness and expandability of our proposed approach. We have observed that the proposed method achieves cross-species disease detection. Furthermore, we discuss the open challenges concerning plant disease detection in the context of open-world detection. In the final section, we provide several conclusions to guide future researchers. In summary, this work establishes the foundation for open-world detection in intelligent agriculture and advocates for increased attention to incremental learning and unknown target detection within the community.

# 2 Related works

In this section, we provide a brief overview of recent studies relevant to our proposed approach. Firstly, we delve into existing deep learning-based methods employed in plant disease detection. Furthermore, considering the limitations of the latest advancements in plant disease recognition, no previous work specifically addresses open-world detection. Consequently, we explore two closely related avenues: open-set detection and open-world detection.

## 2.1 Deep learning technics in plant disease detection

In recent years, various deep learning-based object detection algorithms have been applied in plant disease detection task (Qiao et al., 2022a; Shoaib et al., 2023). In the two-stage plant disease detection methods, Fuentes et al. (2017) first used Faster RCNN (Ren et al., 2015) to accurately locate tomato diseases and pests in a dataset consisting of 4800 images with 11 different classes. When using deep feature extractors like VGG-Net and ResNet, the mean average precision (mAP) was calculated as 88.66%. Liu and Wang (2021) suggested modifying the Faster RCNN (Ren et al., 2015) framework to automatically detect beet spot diseases by changing the parameters of the CNN model. Priyadharshini and Dolly (2023) provided a comparative investigation on tomato leaf disease detection and classification using RCNN (Girshick et al., 2014), Fast RCNN (Girshick, 2015) and Faster RCNN (Ren et al., 2015). Murugeswari et al. (2022) trained a model using 1500 images of healthy and diseased sugarcane leaves and deployed the model in an android application. Seetharaman and Mahendran (2022) proposed using a convolutional recurrent neural network for banana leaf disease detection. Alruwaili et al. (2022) proposed real-time faster region convolutional neural network (RTF-RCNN) for the real-time detection of tomato leaf diseases in video streams.

In the application of single-stage networks, Zhang et al. (2019) proposed a new method for detecting small agricultural pests by

combining an improved version of the YOLOv3 algorithm with spatial pyramid pooling. This method addresses the low accuracy caused by the varying poses and scales of crop pests by applying deconvolution, oversampling, and convolution operations. Mathew and Mahesh (2022) used YOLOv5 to detect bell pepper leaf disease. Wang et al. (2022) optimized the lightweight YOLOv5 model for detecting peanut diseases. Additionally, Dong et al. (2022) evaluated the performance of different annotation strategies based on the YOLOv5 model.

During the training process, the aforementioned methods have access to all labels. However, they cannot locate and classify unknown diseases. In the task of plant disease classification, Fuentes et al. (2021) proposed an approach based on the concept of open-set domain adaptation to the task of plant disease recognition to allow existing systems to operate in new environments with unseen conditions and farms. To the best of our knowledge, there is currently no relevant work on detecting unknown diseases in plant disease detection tasks.

## 2.2 Out-of-distribution detection

The class in the training dataset refers to the 'known class' while a class existing in the test dataset but not in the training dataset is termed an 'unknown class'. Determining whether inputs are out-of-distribution (OOD) is an essential building block for safely deploying machine learning models in the open world. OOD detection is crucial for ensuring the reliability and usability of systems in the real world. Hendrycks and Gimpel (2016) proposed a baseline for OOD detection that relies on softmax confidence scores. However, such methods can be influenced by overconfidence in the posterior distribution of OOD data. Liu et al. (2020) demonstrated mathematically that the softmax confidence score is a biased scoring function that is not aligned with the density of the inputs and hence is not suitable for OOD detection.

The energy-based model maps each input to a single scalar that is lower for observed data and higher for unobserved ones (Lecun et al., 2006). Liu et al. (2020) first proposed a unified framework for OOD detection using energy scores. Unlike softmax confidence scores, energy scores are theoretically aligned with the probability density of the input and are less susceptible to issues of overconfidence. Joseph et al. (2021) were the first to apply energy-based OOD detection to object detection. In this paper, we follow the setup of (Joseph et al., 2021) and maintain a validation set to learn the energy distribution of both known and unknown classes.

## 2.3 Open-world object detection

Open-world object detection is an emerging topic in computer vision and has attracted extensive attention due to its practicability in the real world. Unlike OOD tasks that only focus on the identification of unknown classes, open-world tasks require models to learn new classes and recognize old classes. This learning process is also known as incremental learning. To our best knowledge, there have been only a few relevant works published in top-tier conferences and journals (Joseph et al., 2021; Gupta et al., 2022; Wu et al., 2022; Ma et al., 2023a; Ma et al., 2023b; Zohar et al., 2023). Based on network architecture, these works can be categorized into methods based on Region Proposal Network (RPN) (Joseph et al., 2021; Wu et al., 2022; Ma et al., 2023b) and methods based on Transformer (Gupta et al., 2022; Ma et al., 2023a; Ma et al., 2023b; Zohar et al., 2023).

To endow the model with the capacity of detecting unknown objects, Joseph et al. (2021) proposed the Open World Object Detection (ORE) method, in which an unknown auto-labeling RPN is designed to generate pseudo labels for unknown instances. Gupta et al. (2022) and Zohar et al. (2023) employed an attention mechanism to score candidate bounding boxes, enhancing the network's perception capability for unknown objects. Ma et al. (2023a) proposed a method that combines selective search and attention mechanisms to further enhance the retrieval capability for unknown objects. The underlying logic behind these methods is to enhance the proposal quality for unknown objects in order to obtain stronger weak supervision signals. However, methods based on attention mechanisms and selective search tend to be complex. Optimizing the perception capability for unknown objects through a simpler approach is indeed more desirable in practical engineering scenarios. Therefore, we improve the proposal quality of the network for unknown objects by using a pre-trained region proposal network (RPN), thereby enhancing the performance of open-world plant disease detection.

# 3 Methods

## 3.1 Challenges of real-world plant disease detection

Plant disease detection is a complex field that possesses distinct characteristics and challenges, particularly when considering the influence of diverse domains such as greenhouse conditions. Incremental learning serves as a crucial tool to address these challenges and enhance the accuracy and adaptability of disease detection systems.

### 3.1.1 Characteristics of plant disease detection

The process of plant disease detection is marked by several unique characteristics. Unlike some other domains, plant health is influenced by an intricate interplay of factors. Variations in features across plant species, genetic diversity, and environmental conditions lead to a diverse range of disease symptoms. These symptoms can be subtle, ranging from changes in leaf color and texture to wilting and necrosis. Additionally, the progression of diseases can vary widely, making it challenging to predict the trajectory and severity of an infection.

### 3.1.2 Challenges in diverse domains and greenhouse conditions

Diverse domains, such as greenhouse environments, introduce a set of challenges that impact plant disease detection. Greenhouses provide controlled conditions for plant growth, which can

accelerate disease progression due to the close proximity of plants, regulated temperature, and humidity. The dynamic interactions between plants, pathogens, and the environment within greenhouses contribute to complex disease patterns that traditional, static models might struggle to capture. Moreover, the controlled environment can lead to rapid mutations in pathogens, adding further complexity to disease identification.

In a domain characterized by diverse symptoms, feature variations, environmental factors, and disease progression, previous models to detect plant disease can fall short. Our proposed approach, however, enables models to evolve alongside the evolving disease landscape. The adaptive nature of our approach allows models to incorporate new information, adapt to feature variations, and account for changing environmental conditions. As the disease patterns shift and pathogens mutate, incremental learning ensures that the detection system remains up-to-date and effective. This is particularly critical in greenhouse conditions, where rapid disease spread demands real-time monitoring and rapid response.

## 3.2 Problem formulation

In this section, we provide a formal definition of Open World Object Detection. In a closed-setting approach, a model is trained on a specific set of known classes and then tested on data collected from the same or similar environment such as $D_{train}^{closed-set} = \{(x_i, y_i)\}_{i=1}^{M} \subset X \times C$, and $D_{test}^{closed-set} = \{(x_i, y_i)\}_{i=1}^{N} \subset X \times C$, where $X$ denotes the image samples in the dataset $D$, and $C$ indicates the number of classes. However, real-world scenarios often involve new environments and the presence of unknown diseases that the model has not encountered before. Consequently, when tested on such data, the model may fail to perform accurately. In this context, the test dataset is $D_{test}^{open-world} = \{(x_i, y_i)\}_{i=1}^{N} \subset X \times (C\&U)$, where $U$ denotes unknown classes in training phase. Therefore, in open-world disease detection, the primary target is to detect these unknown diseases.

After achieving the primary target, the model becomes capable of identifying diseases that were not part of the initial training set (unknown diseases). Our objective is for the model to learn these new classes in subsequent learning tasks while retaining its recognition ability for the classes learned int eh previous tasks. We define the initial training task as Task 1 and subsequent tasks as Task 2, Task 3, and so on. In Task 1, the training dataset, denoted as $D_{T1}$, consists of labeled samples for a number of $C_{T1}$ disease classes. However, during the inference process, the model may encounter instances of unknown diseases that were not seen during training. To address this, the model needs to accurately locate these unknown disease types and assign them the label 'unknown'. These unknown disease instances will be presented to domain experts for annotation and will be used for training in Task 2. In Task 2, the number of new disease classes is denoted as $C_{T2}$. After completing Task 2, the set of known classes is updated to the previously known classes $C_{T1}$ along with the newly learned classes $C_{T2}$. However, during the inference process, the model still may encounter unknown diseases that do not belong to the known classes ($C_{T1} + C_{T2}$). Therefore, in addition to detecting the known classes, the model will continue to identify unknown diseases and assign them the label 'unknown'. These unknown disease instances will be learned in Task 3.

This cycle of updating the model's knowledge continues throughout the entire lifecycle of the detector. In each task, the detector acquires new knowledge without forgetting the previously learned classes. This allows the model to continuously adapt and improve its detection capabilities by incorporating new information in a progressive manner.

## 3.3 Datasets and splits

After defining the open-world problem, it is necessary to search for suitable datasets to evaluate our method. In this study, we extended the tomato dataset used in previous works (Fuentes et al., 2018; Fuentes et al., 2021) to include 15 different classes, which were learned in Task 1, Task 2, and Task 3, respectively. To ensure a balanced distribution, we divided the classes equally, with 5 different classes assigned to each task. In Task 1, instances belonging to the classes of Task 2 and Task 3 were not available. Additionally, we aimed to investigate the performance of our model in cross-species training. For this purpose, we incorporated the paprika disease detection dataset (Dong et al., 2022) in Task 4. The tomato dataset originally consisted of 15 classes, while the paprika dataset contained 5 classes. To ensure the dataset's representation of real-world scenarios and to introduce complexity, we excluded images collected in a laboratory setting. This approach prevents potential overestimation of the model's performance and enhances the dataset's ability to simulate real-world conditions.

For each task, we employed a random selection process to designate 20% of the integrated dataset (combining tomato and paprika data) as the validation data. This allowed us to learn the distribution of known and unknown samples within this subset. Additionally, we randomly chose 20% of the data as the test set, which was used across all tasks. Here we aim to address the question: why do we test diseases from different species together? There are several reasons for this approach. Firstly, evaluating the performance of our model on different species' diseases allows us to assess its generalization capability across species. In real-world scenarios, plant disease detection systems encounter various species and their associated diseases. By testing different species' diseases together, we can effectively assess how well our model handles the challenges of detecting diseases across multiple species. This includes dealing with variations in symptoms, visual appearances, and disease patterns. Such evaluation helps us gain insights into the robustness and effectiveness of our model in practical applications where encounters with a diverse range of plant species are expected. Furthermore, successful detection of diseases from different species indicates that our model has acquired solid features of diseases as a concept. It demonstrates that the model's learning transcends species-specific information and can be effectively applied to diverse plant species. The dataset split and more specific details are presented in Table 1. Unless otherwise specified, the training order of all experiments in this paper follows the sequence shown in Table 1.

## 3.4 Architecture

In their study, Dhamija et al. (2020) found that two-stage networks outperform single-stage networks when it comes to detecting unknown objects. Motivated by this finding, we have chosen to implement our open-world detection model using the classic Faster RCNN (Ren et al., 2015), which is a two-stage network architecture. To enhance the representation of multi-scale features, we have incorporated the feature pyramid network (FPN) (Lin et al., 2017).

In Figure 2, we present an illustration of the Faster RCNN with the FPN network. Please note that our method, unlike the standard Faster RCNN, can detect unknown classes. This capability is achieved through a well-trained unknown perception Region Proposal Network (RPN) and a class-agnostic localization head. The unknown perception RPN is designed for automatic labeling of unknown objects, while the class-agnostic localization head is responsible for accurately localizing these unknown objects. Each of these components is explained in detail in the following subsections, providing a coherent understanding of their roles in our model.

## 3.5 Well-trained unknowns-aware RPN

In the context of object detection tasks, the objective is to identify and localize objects of interest within an image. Traditional object detection models are typically trained on datasets that consist of known classes, assuming that all objects can be classified into predefined categories. However, real-world scenarios often present instances where the model encounters objects belonging to unknown or unseen classes.

To address the challenge of detecting unknown diseases, we introduce an additional "unknown" class during the training process. This class is assigned as a pseudo label 'unknown' to proposals that have a high objectness score but do not overlap with any ground-truth objects. To generate high-quality proposal boxes, we directly train the detector on the object detection dataset to obtain well-initialized parameters. A well-trained RPN can generate highly accurate proposals or candidate object regions within an image. These proposals effectively filter out cluttered or background regions, enabling the model to focus solely on relevant object proposals. This capability helps in reducing false positives and improving overall detection accuracy. Additionally, a well-trained RPN can effectively handle objects of different sizes and shapes. It learns to generate proposals that encompass objects with varying aspect ratios, ensuring comprehensive coverage of the object space. This enables the model to effectively handle novel or unseen objects, thereby enhancing its performance and robustness in open-world scenarios. We further compare the performance of different pre-trained datasets in open-world plant disease detection.

## 3.6 Class-agnostic ROI head

Locating unknown diseases is an important issue in open-world detection tasks. Standard detectors are primarily designed for localizing objects of known classes, as they employ class-specific localization methods. For instance, detectors like Faster RCNN (Ren et al., 2015) and Mask RCNN (He et al., 2017) generate class-specific bounding boxes for each known class when the proposals enter their prediction heads.

To address the localization of novel objects, we introduce a class-agnostic Region of Interest (ROI) head in our object detection models. The class-agnostic ROI head treats region-based feature extraction and classification tasks independently of specific object classes. Unlike class-specific ROI heads that are designed to predict object classes for each region, the class-agnostic ROI head focuses solely on generating accurate bounding box regression outputs without considering the object categories. This makes it well-suited for open-world object detection scenarios where unknown or novel classes may appear.

Inspired by the learned objectness (Kuo et al., 2023), we utilize class-agnostic box regression heads instead. We have observed that class-agnostic ROI heads exhibit better generalization to unseen classes during inference. They are not biased towards specific object

TABLE 1 Task composition and data split in the proposed open-world plant disease detection protocol.

| Task sequence | Task 1 | Task 2 | Task 3 | Task 4 |
|---|---|---|---|---|
| Species | Tomato | Tomato | Tomato | Paprika |
| Training Classes | 5 | 5 | 5 | 5 |
| Categories | magnesium deficiency, gray mold, leaf mold, yellow leaf curl virus, physical damage | canker, plague, leaf miner, white fly, white fly egg | wilt, chlorosis virus, stress, powdery mildew, old leaf | blossom end rot, gray mold, powdery mildew, spider mite, spotting disease |
| Training images | 3236 | 1728 | 1647 | 2049 |
| Validation images | 2491 | 2491 | 2491 | 2491 |
| Test images | 2493 | 2493 | 2493 | 2493 |
| Known Classes | 5 | 10 | 15 | 20 |
| Unknown Classes | 15 | 10 | 5 | 0 |

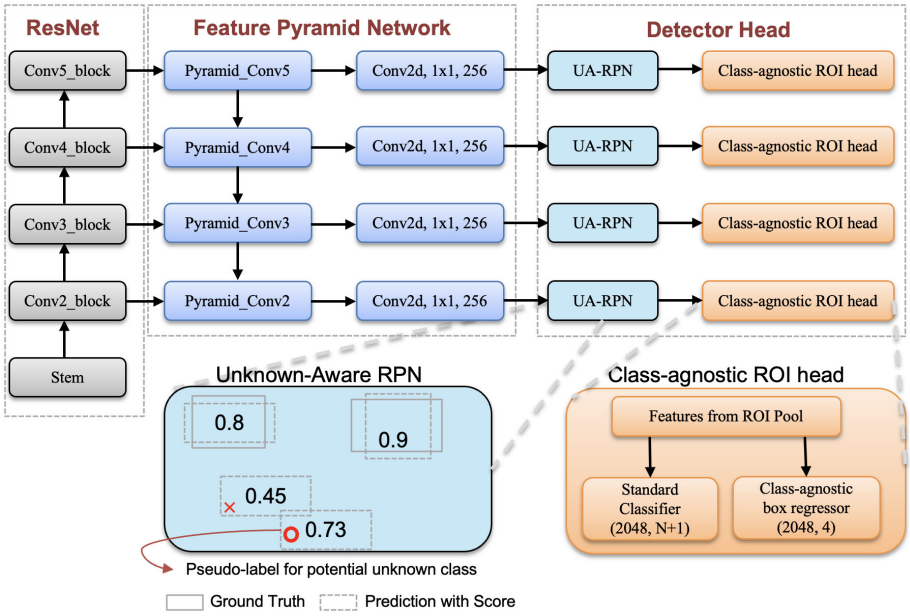



FIGURE 2
Overview of our model where ResNet and FPN are constructed following the default approach in detectron2 (Wu et al., 2019). We illustrate the unknown-aware RPN and class-agnostic ROI head in the diagram. Unknown aware RPN modifies the labels of background candidate boxes with the highest object scores to 'unknown'. The class-agnostic head focuses on regressing bounding boxes for disease regions without considering the disease category.

categories, allowing the model to adapt to new classes without the need for retraining or fine-tuning. Additionally, by removing the class-specific classification branch, the overall architecture becomes simpler and more streamlined. This modification not only reduces the computational complexity and memory requirements of the model but also enables more efficient handling of unknown classes.

## 3.7 Alleviating forgetting

Catastrophic forgetting (Hayes et al., 2020) refers to the phenomenon observed in incremental learning, where a model trained on new data gradually loses or forgets the knowledge acquired from previously learned tasks or classes. This occurs when the new data heavily influences the model's parameters, leading to the overwriting or disrupting of previously learned information. To address catastrophic forgetting, several techniques have been proposed, such as parameter isolation (Prabhu et al., 2020), regularization (Li and Hoiem, 2017), and sample replay (Rebuffi et al., 2017). These techniques reinforce the model's memory of previous tasks or classes by incorporating previously observed samples during training. In this way, the model can maintain its performance on old tasks while learning new ones.

Sample replay is relatively straightforward compared to other techniques like parameter isolation or complex regularization strategies. It periodically included old samples in the training dataset, making integrating them into existing training pipelines easy. The simplest form of sample replay is randomly retaining training samples. This paper follows the sample replay strategy proposed by Joseph et al. (2021), which is the simplest way of sample replay. After each incremental step, a balanced set of samples is stored randomly, and the model is fine-tuned. To ensure an adequate representation of each class, we guarantee a minimum of $N_{samples}$ instances for each class in the sample set. Generally, a larger $N_{samples}$ tends to result in better fine-tuning performance (an extreme case being the use of the entire dataset). However, this contradicts the original intention of dynamic learning in an open-world setting. To ensure a fair comparison among the models, we set $N_{samples} = 25$ for fine-tuning the model.

## 3.8 Evaluation metrics

We present a comprehensive evaluation protocol to assess the performance of an open-world detector in various aspects: identifying unknown classes, detecting known classes, and progressively learning new classes when labels are available for some unknown samples.

### 3.8.1 Mean average precision score

mAP is the area under the precision-recall curve calculated for all classes. To evaluate the detection performance of known classes, we utilize the standard mean average precision (mAP) metric with an intersection over union (IoU) threshold of 0.5 [mAP@50, consistent with the existing literature (Joseph et al., 2021; Gupta et al., 2022; Wu et al., 2022; Ma et al., 2023a; Ma et al., 2023b; Zohar et al., 2023)].

$$AP = \frac{1}{11}\sum_{r\in[0,0.1,\ldots,0.9,1]} P(r) \quad (1)$$

$$\mathbf{P}(\mathbf{r}) = \max_{\tilde{r}:\tilde{r}\geq r} p(\tilde{r}) \quad (2)$$

where, $P(r)$ is the maximum precision for any recall values greater than r, and $p(\tilde{r})$ is the measured precision at recall $\tilde{r}$. Since the problem setting of open-world detectors is different from that of standard detectors, there are three forms of mAP, which are current classes mAP, previous classes mAP, and known classes mAP.

### 3.8.2 Unknown recall

We employ recall as the main metric for unknown object detection instead of the commonly used mAP. This is because all possible unknown object instances in the dataset are not annotated. Unknown recall is widely used in open-world object detection (Gupta et al., 2022; Ma et al., 2023a; Ma et al., 2023b; Zohar et al., 2023).

$$U - R = \frac{TPU}{AU} \quad (3)$$

where, $TPU$ is the true positive of unknown instances, and AU denotes all unknown instances for the current task.

### 3.8.3 Absolute open-set error

In addition, we employ the Absolute open-set error (A-OSE) (Miller et al., 2018) metric to report the number of unknown objects that are misclassified as any of the known classes. This metric implicitly measures how effective the model is in handling unknown objects.

To facilitate readability, we use the abbreviations listed in Table 2 to denote the evaluation metrics. The metrics include Unknown Recall and A-OSE, which assess the performance of the unknown classes, and Mean Average Precision (mAP), which evaluates the model's ability to detect the known classes. By employing these metrics, we can comprehensively evaluate and compare the model's performance across both known and unknown classes, providing a comprehensive assessment of its detection capabilities.

TABLE 2 Abbreviation and meaning of the evaluation metrics.

| Abbreviation | Meaning | Others |
|---|---|---|
| P | mean average precision score of previous task classes | ↑ |
| C | mean average precision score of current task classes | ↑ |
| K | mean average precision score of all known classes | ↑ |
| A | Absolute open-set error | ↓ |
| U-R | Unknown-Recall | ↑ |

Arrows indicate expected trends. Up means that the larger the value, the better, and vice versa.

# 4 Results

## 4.1 Implementation details

In the training task sequence, the model can only access the data from the current task. Known classes are defined as the classes in the current task as well as the previous tasks, while other classes are defined as unknown classes. For each image, the model generates only one unknown instance. We adopted the contrastive clustering loss proposed by ORE (Joseph et al., 2021) and used stochastic gradient descent to optimize the model, with a batch size set to 4. For each training task, we iterated 18,000 times, and for each fine-tuning task, we iterated 4,000 times. We used ResNeXt101 (Xie et al., 2017) as the final backbone. The entire training process for the project, conducted on 4 NVIDIA GeForce RTX 3090 GPUs, was completed in less than 12 hours. For more details, please refer to our code.

## 4.2 Overall results

Table 3 compares our method with Faster RCNN (Ren et al., 2015) and ORE (Joseph et al., 2021) using the proposed open-world evaluation protocol. The 1-3 row in Table 3 showcases the result obtained by the standard Faster-RCNN. Note that we used the ResNet50 backbone on the ImageNet1K dataset as a pretraining backbone. We provide a brief overview of the training approach for Faster RCNN. Row 1: We trained Faster-RCNN using a static closed-set training strategy for a fair comparison. As anticipated, Faster-RCNN trained with the closed-set strategy demonstrated optimal results in closed-set evaluation metrics, because the model retrained with all known datasets for each task. However, the model's focus remains limited to known categories, incapable of identifying unknown targets, which contradicts the open-world setting. This experimental set allows researchers to grasp the upper-performance limits of the model in known-category recognition tasks. Hence, we employ 'Upper' to denote the results of this experiment. Row 2: We trained the standard Faster-RCNN on Task 1, followed by Task 2, Task 3, and Task 4. After completing each task, the model's performance was evaluated through testing. In this scenario, the model was also unable to identify unknown diseases. We observed a significant decline in detection performance for previous classes during subsequent task learning with the standard Faster RCNN, which indicates that new knowledge quickly replaced old knowledge throughout the training process. In contrast, our method can successfully detect unknown classes and continuously learn new categories without the need to train from scratch. Row 3: We employed a sample replay strategy to train Faster-RCNN dynamically. This experimental set allows researchers to understand how much sample replay preserves the model's memory capabilities. We denote the results of this experiment as 'Faster-RCNN*' in Table 3.

Furthermore, our four variants, labeled as Ours (a), Ours (b), Ours (c), and Ours (d) utilized the ResNet-50-FPN backbone, but

TABLE 3 Overall results of our method compared with the baseline approach.

| Methods | Task 1 | | | Task 2 | | | | | Task 3 | | | | | Task 4 | | | Parameter |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | C | A | U-R | P | C | K | A | U-R | P | C | K | A | U-R | P | C | K | |
| Upper | 63.9 | 2044 | – | 69.20 | 60.54 | 64.91 | 1706 | – | 67.06 | 48.66 | 60.93 | 705 | – | 63.88 | 84.40 | 69.02 | 33M |
| Faster RCNN | 63.9 | 2044 | – | 6.6 | 42.9 | 24.8 | – | – | 3.6 | 25.5 | 10.9 | – | – | 1.8 | 54.3 | 14.9 | 33M |
| Faster RCNN* | 63.9 | 2044 | – | 62.8 | 40.3 | 51.6 | 2386 | – | 50.7 | 32.2 | 44.5 | 1226 | – | 44.7 | 62.2 | 49.1 | 33M |
| ORE | 63.5 | 2002 | 14.2 | 62.6 | 39.1 | 50.9 | 2303 | 4.7 | 48.2 | 31.0 | 42.5 | 1228 | 5.1 | 42.9 | 62.7 | 47.9 | 33M |
| Ours (a) | 65.4 | 2124 | 22.6 | 63.2 | 43.2 | 53.2 | 2190 | 12.4 | 50.6 | 29.3 | 43.5 | 1572 | 13.1 | 44.5 | 61.5 | 48.8 | 41M |
| Ours (b) | 60.7 | 1827 | 24.0 | 63.1 | 46.3 | 54.7 | 3144 | 8.9 | 52.4 | 28.2 | 44.4 | 1638 | **38.3** | 45.3 | 67.0 | 50.7 | 41M |
| Ours (c) | 62.4 | 1932 | **25.3** | 62.7 | 44.9 | 53.8 | 1961 | 13.0 | 52.7 | 29.5 | 45.0 | 1211 | 13.0 | 47.1 | 67.1 | 52.1 | 41M |
| Ours (d) | 65.2 | 1880 | 23.8 | 63.2 | 47.0 | 55.1 | 1960 | **13.7** | 54.4 | 31.3 | 46.7 | 1418 | 15.7 | 46.8 | 66.9 | 51.8 | 41M |
| Ours* | **66.3** | **1559** | 22.7 | **64.0** | **48.9** | **56.5** | **1854** | **13.4** | **54.5** | **35.8** | **48.3** | **1123** | 10.8 | **48.2** | **70.0** | **53.6** | 104M |

For the notation of evaluation metrics, please refer to Table 2. Bold font represents the optimal results. "Upper" represents training using the entire dataset, which theoretically serves as the performance upper bound. "Faster RCNN*" represents training using a dynamic paradigm.
Ours* is a larger model than ours (d), other settings are the same.
'-' indicates that the evaluation metric is not applicable to the current experimental setup.

were pretrained on different datasets. Specifically, Ours (a) used Imagenet-1k (Deng et al., 2009), Ours (b) used COCO (Lin et al., 2014), Ours (c) used Object-365-v2 (Shao et al., 2019), and Ours (d) used the LVIS (Gupta et al., 2019) dataset. These experiments demonstrate that our method consistently outperforms the ORE (Joseph et al., 2021) baseline across all evaluation metrics. Additionally, we explored the ResNeXt101(Xie et al., 2017) architecture, an extension of ResNet, which introduced cardinality to enhance feature representation, making it potentially more powerful in capturing complex patterns and achieving better performance compared to ResNet101. To further improve the model's performance, we trained the ResNeXt-101-FPN on the LVIS dataset. The final row in the table shows the results of our method using the ResNeXt-101-FPN backbone pre-trained on the LVIS dataset, denoted as Ours* in Table 3. Note that A-OSE scores and unknown recalls cannot be measured for Task 4 because of the absence of unknown ground truths. For a visual comparison with the baseline, we present the detection results for Task 1 in Figure 3. Our model outperformed ORE in terms of known disease detection, demonstrating higher accuracy in Figures 3A, B. Furthermore, when it comes to unknown diseases, our model excelled in reducing false positives as seen in Figures 3C, D. Additionally, our model achieved precise localization for unknown diseases, as evident in Figure 3E–H.

Furthermore, in Figure 4, we present additional qualitative results, showcasing a batch of images that were tested on our model across three tasks. Case A and Case E highlight the model's ability to remember previously learned classes, accurately classifying and locating diseases learned in Task 1. Cases B, C, and D demonstrate the model's capability to detect unknown diseases and progressively learn them. Although these instances were unknown in Task 1, the model gradually learned them in Task 2 and Task 3. Additionally, we include a set of failed cases where the model started to exhibit confusion in localizing old classes as new knowledge is introduced. These challenges will be addressed in future studies.

## 4.3 Ablation experiment

To analyze the individual contributions of each component in our method, we conducted meticulous ablation experiments, and the results are presented in Table 4.

### 4.3.1 Backbone

We compared the FPN module with the C4 module on ResNet-50 (Row 1 and Row 2). The inclusion of FPN significantly enhanced the model's learning ability and memory capacity, as evidenced by improved performance in Task 1 (63.55% vs. 67.67%) and Task 3 (48.27% vs. 51.50%). Based on this observation, all subsequent experiments were performed using ResNet with FPN as the backbone network instead of the C4 structure.

### 4.3.2 Class-agnostic head

Our class-agnostic head played a crucial role in the model's performance. By not assigning specific class labels to detected objects, the class-agnostic head enabled the model to treat all objects as potential unknown classes. This means that if a detected object does not match any known class, it is more likely to be classified as an unknown object rather than misclassified into a known class. Consequently, the class-agnostic head improved the model's ability to recognize and recall unknown objects, thus enhancing overall performance in open-world scenarios. Table 4 demonstrates that the class-agnostic head significantly improved the recall of unknown classes across different pretraining data. Moreover, Table 5 indicates that the class-agnostic head remains effective even when used with larger networks.

### 4.3.3 Pretraining datasets

In order to investigate the influence of different pre-training datasets on our model, we conducted a series of experiments as outlined in Table 6. Our findings reveal that the model trained on the Imagenet-1k dataset exhibited better performance on the initial




FIGURE 3
Visualization results comparison between ORE and our model, both trained on Task 1. We present eight pairs of examples **(A-H)**. Best view in color.

tasks. However, as the tasks progressed, this advantage gradually diminished. On the other hand, the model trained on the LVIS dataset showed an advantage in terms of unknown recall, with no significant drop in performance (mAP) for known class detection. Similarly, the model trained on the COCO dataset exhibited a similar trend, albeit with slightly lower performance.

We attribute the benefits brought by the LVIS-based pre-trained models to two main factors. Firstly, the consistency of pre-training objectives played a significant role. The LVIS-based pre-trained models utilize training objectives that align closely with the target detection task, encompassing multi-label classification and bounding box regression. In contrast to ImageNet pre-trained models, these objectives are better suited for the target detection task, resulting in improved performance. Secondly, the richness of the data is a contributing factor. LVIS encompasses over 1,200 categories, whereas COCO only includes 80 categories. The significantly larger number of categories in LVIS provided a more diverse and comprehensive representation of objects across various domains. Consequently, this allowed the LVIS-based pretrained models to learn more comprehensive features and contextual information for different categories. Based on these observations, we argue that the pre-training model based on LVIS exhibited greater potential for subsequent tasks due to the alignment of training objectives and the broader representation of object categories.

Furthermore, we also trained and released these three models on the Object365 dataset using the Detectron2 framework. The Object365-v2 dataset (Shao et al., 2019) contains nearly 2 million images with over 10 million annotated bounding boxes. In terms of scale, Object365-v2 contains a greater number of instances compared to LVIS. However, we observed that pre-trained on the Object365-v2 dataset significantly boosts the performance of the COCO dataset in open-world evaluation settings, but its performance on plant disease datasets is slightly lower than the model pre-trained on LVIS dataset. As a result, we opted for the LVIS-based pre-trained model as the final choice for our work. Please note that fine-tuning the COCO dataset results using the Object365-v2 dataset is not the focus of this paper. We presented these results in our code repository.

Additionally, we performed experiments using larger models to enhance the performance of our model, as presented in Table 5. It was challenging to improve all performance metrics across all tasks simultaneously. However, in general, employing larger models, leveraging well-pretrained Region Proposal Networks (RPNs), and incorporating class-agnostic heads tended to yield better results.

## 4.4 Sensitivity analysis on training order

In the context of incremental learning tasks, the order in which tasks are presented to the model can significantly impact its performance and the overall learning process. The learning sequence plays a crucial role in addressing challenges such as

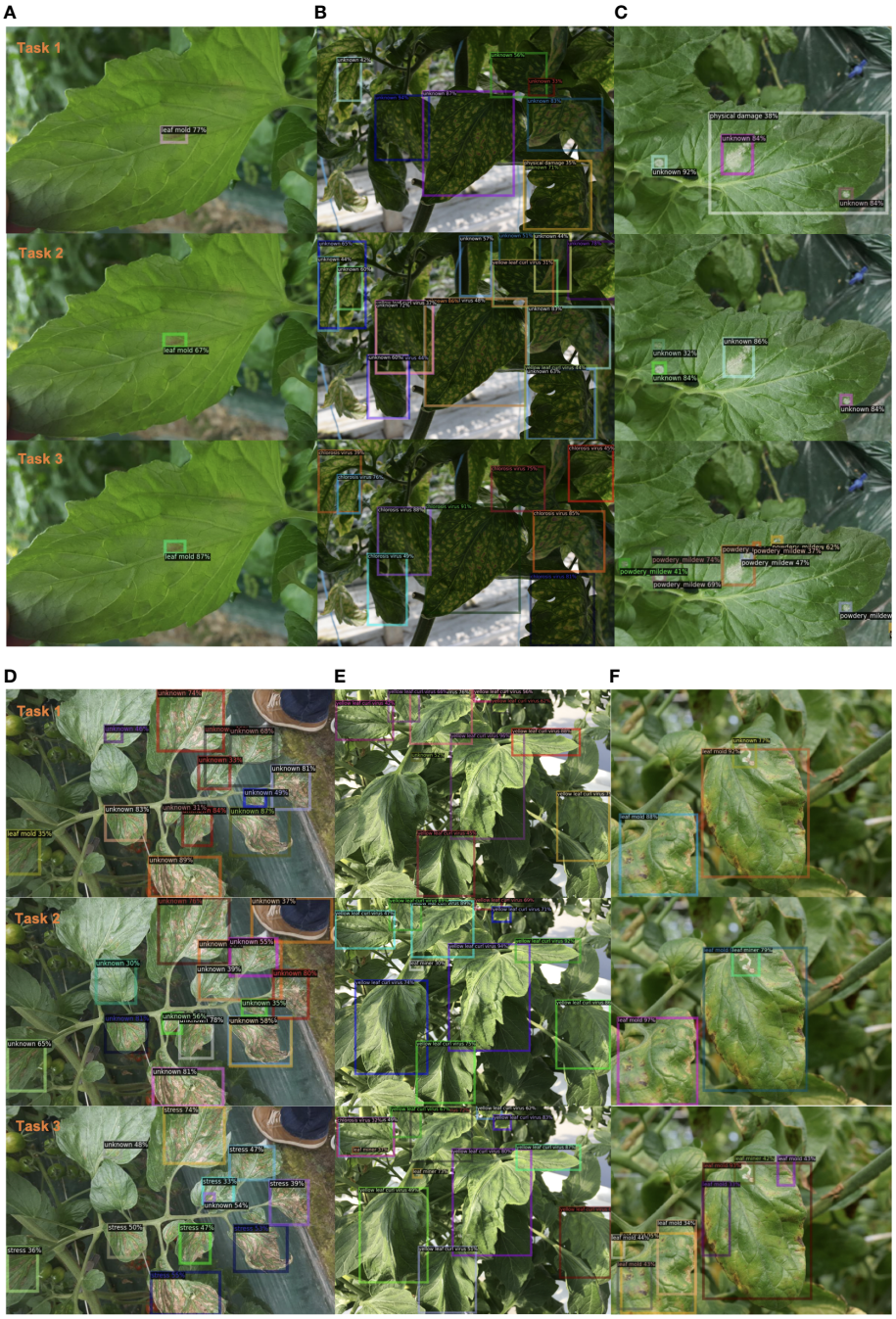



FIGURE 4
Qualitative results of our method on example images from our plant disease dataset. We present six groups of examples **(A–F)** from Task 1 to Task 3. Best view in color.

knowledge forgetting, conflicting information, and fluctuations in performance. Recognizing the importance of the learning order, we conducted an investigation to understand the model's sensitivity to different training sequences.

By analyzing the results in Table 3, we observed that Faster RCNN achieved the highest performance on Task 1 and the lowest on Task 3 when detecting tomato diseases. This observation led us to infer that Task 1 might be relatively simpler, while Task 3 could pose more challenges in disease detection.

Following the principle of human learning from easy to difficult, we believe that the model should start learning from simple tasks. Therefore, in previous experimental settings, the default learning order was from Task 1 to Task 3. After learning the diseases of one species, the model continued to learn the cross-species detection task (Task 4).

However, to explore the sensitivity of our research model to the training order, we decided to deviate from the default sequence and adopt a different approach. We opted to initiate the learning process

TABLE 4 Ablation results. PTD and CAH denote pre-trained dataset and class-agnostic head, respectively.

| Architecture | PTD | CAH | Task 1 | | | Task 2 | | | | | Task 3 | | | | | Task 4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | C | A | U-R | P | C | K | A | U-R | P | C | K | A | U-R | P | C | K |
| R-50-C4 | IN1k | x | 63.5 | 2002 | 14.2 | 62.6 | 39.1 | 50.9 | 2303 | 4.7 | 48.2 | 31.0 | 42.5 | 1228 | 5.1 | 42.9 | 62.7 | 47.9 |
| R-50-FPN | IN1k | x | 67.6 | 2173 | 14.7 | 63.6 | 42.9 | 53.3 | 2176 | 13.2 | 51.5 | 30.0 | 44.3 | 1688 | 7.8 | 44.8 | 65.7 | 50.0 |
| R-50-FPN | IN1k | √ | 65.4 | 2124 | 22.6 | 63.2 | 43.2 | 53.2 | 2190 | 12.4 | 50.6 | 29.3 | 43.5 | 1572 | 13.1 | 44.5 | 61.5 | 48.8 |
| R-50-FPN | COCO | x | 59.0 | 1906 | 12.4 | 60.3 | 44.9 | 52.6 | 2075 | 10.4 | 52.9 | 28.1 | 44.6 | 1468 | 4.1 | 45.6 | 67.3 | 51.0 |
| R-50-FPN | COCO | √ | 60.7 | 1827 | 24.0 | 63.1 | 46.3 | 54.7 | 3144 | 8.9 | 52.4 | 28.2 | 44.4 | 1638 | 38.3 | 45.3 | 67.0 | 50.7 |
| R-50-FPN | Object365 | x | 62.0 | 2066 | 10.5 | 62.1 | 47.1 | 54.6 | 1971 | 9.8 | 53.2 | 36.4 | 47.6 | 1300 | 5.1 | 47.8 | 69.1 | 53.1 |
| R-50-FPN | Object365 | √ | 62.4 | 1932 | 25.3 | 62.7 | 44.9 | 53.8 | 1961 | 13.0 | 52.7 | 29.5 | 45.0 | 1211 | 13.0 | 47.1 | 67.1 | 52.1 |
| R-50-FPN | LVIS | x | 64.6 | 2036 | 13.0 | 61.9 | 46.9 | 54.4 | 2156 | 13.5 | 55.0 | 32.9 | 47.6 | 1566 | 9.1 | 47.1 | 69.1 | 52.6 |
| R-50-FPN | LVIS | √ | 65.2 | 1880 | 23.8 | 63.2 | 47.0 | 55.1 | 1960 | 13.7 | 54.4 | 31.3 | 46.7 | 1418 | 15.7 | 46.8 | 66.9 | 51.8 |

IN1k denotes the Imagenet-1k dataset. For the notation of evaluation metrics, please refer to Table 2.
'√' and 'x' respectively indicate model with or without the class-agnostic head.

with the more difficult Task 3. By doing so, we aimed to observe how the model adapts and performs when confronted with the most challenging task from the start. Therefore, in this study, we rearranged the task sequence as follows: Task 3, Task 2, Task 1, and finally, Task 4.

This alternative task sequence enabled us to investigate the model's ability to learn and transfer knowledge in a non-conventional order, offering insights into its adaptability and potential for early tackling of more complex tasks.

The detection results of the model on the tomato disease dataset and the paprika disease dataset under different training orders are presented in Table 7. Due to different task sequences, we can only compare the model's performance in detecting known classes after learning 15 tomato diseases. We also compared the model's memory ability to capture disease patterns and learning ability in cross-species detection tasks such as paprika. The memory ability is reflected in the mAP of previous classes (P), while the learning ability is reflected in the mAP of current classes (C). As expected, learning from more challenging task orders led to a slight performance degradation in the model for all aspects, even though the impact is not significant. This finding serves as a reminder to practitioners that the learning order of models should follow a progression from simpler to more difficult tasks in order to achieve optimal performance.

## 4.5 Cross-species detection

Our research has uncovered a fascinating discovery regarding the capabilities of our model, particularly in the context of cross-species disease detection. Despite being trained solely on a dataset of tomato diseases, our model exhibited the remarkable ability to identify and provide an initial assessment of affected regions in paprika fruit diseases. This intriguing finding is illustrated through two specific cases showcased in Figure 5A, namely Case 1 and Case 2.

Please note that our model has never been exposed to or trained on any tomato fruit diseases, only leaves, making its performance in detecting paprika fruit diseases all the more intriguing. The fact that the model can generalize its knowledge and effectively apply it to a different species highlights its versatility and potential for cross-species disease detection, which also demonstrates that our method learns the fundamental features of disease.

Furthermore, we conducted a similar experiment in which we trained a separate model using a paprika disease dataset and evaluated its performance on a test dataset consisting of tomato plants. The results were equally compelling. Our paprika-trained model successfully detected pests present on tomato leaves, as demonstrated by Case 8 in Figure 5B. This further reinforces the model's ability to transfer its learned knowledge across species boundaries and adapt it to different contexts.

TABLE 5 Results of our method using larger model.

| Architecture | PTD | CAH | Task 1 | | | Task 2 | | | | | Task 3 | | | | | Task 4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | C | A | U-R | P | C | K | A | U-R | P | C | K | A | U-R | P | C | K |
| R-50-C4 | IN1k | x | 63.5 | 2002 | 14.2 | 62.6 | 39.1 | 50.9 | 2303 | 4.7 | 48.2 | 31.0 | 42.5 | 1228 | 5.1 | 42.9 | 62.7 | 47.9 |
| R-50-FPN | IN1k | x | 67.6 | 2173 | 14.7 | 63.6 | 42.9 | 53.3 | 2176 | 13.2 | 51.5 | 30.0 | 44.3 | 1688 | 7.8 | 44.8 | 65.7 | 50.0 |
| R-50-FPN | LVIS | √ | 65.2 | 1880 | 23.8 | 63.2 | 47.0 | 55.1 | 1960 | 13.7 | 54.4 | 31.3 | 46.7 | 1418 | 15.7 | 46.8 | 66.9 | 51.8 |
| R-101-FPN | LVIS | √ | 64.8 | 1815 | 24.0 | 65.4 | 44.1 | 54.7 | 2448 | 9.3 | 54.2 | 34.5 | 47.6 | 1418 | 7.6 | 48.3 | 71.8 | 54.1 |
| X-101-FPN | LVIS | √ | 66.3 | 1559 | 22.7 | 64.0 | 48.9 | 56.5 | 1854 | 13.4 | 54.5 | 35.8 | 48.3 | 1123 | 10.8 | 48.2 | 70.0 | 53.6 |

For a fair comparison, we show the results of the baseline (R50-C4) and Resnet-50 with FPN (R50-FPN).
'√' and 'x' respectively indicate model with or without the class-agnostic head.

TABLE 6 Results of different pre-training data in the open-world disease detection tasks.

| (A) | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Architecture | PTD | CAH | Task 1 | | | Task 2 | | | | | Task 3 | | | | | Task 4 | | |
| | | | C | A | U-R | P | C | K | A | U-R | P | C | K | A | U-R | P | C | K |
| R-50-FPN | IN1k | x | **67.6** | 2173 | **14.7** | **63.6** | 42.9 | 53.3 | 2176 | 13.2 | 51.5 | 30.0 | 44.3 | 1688 | 7.8 | 44.8 | 65.7 | 50.0 |
| R-50-FPN | COCO | x | 60.9 | **1950** | 11.8 | 59.4 | 44.0 | 51.7 | 2238 | 9.4 | 51.5 | 27.6 | 43.5 | 1731 | 1.3 | 44.9 | 66.3 | 50.2 |
| R-50-FPN | Object365 | x | 62.0 | 2066 | 10.5 | **62.1** | **47.1** | **54.6** | **1971** | 9.8 | 53.2 | **36.4** | **47.6** | **1300** | 5.1 | **47.8** | **69.1** | **53.1** |
| R-50-FPN | LVIS | x | 64.6 | 2036 | 13.0 | 61.9 | 46.9 | 54.4 | 2156 | 13.5 | **55.0** | 32.9 | 47.6 | 1566 | **9.1** | 47.1 | 69.1 | 52.6 |
| (B) | | | | | | | | | | | | | | | | | | |
| R-101-FPN | IN1k | x | **66.7** | 2042 | **14.9** | 63.2 | 44.0 | 53.6 | **2038** | **13.2** | 52.1 | 31.0 | 45.0 | 1538 | 3.7 | 44.9 | 62.4 | 49.2 |
| R-101-FPN | COCO | x | 61.5 | **1842** | 10.5 | 61.9 | 45.1 | 53.5 | 2367 | 6.7 | 52.1 | 28.4 | 44.2 | 1538 | 2.7 | 45.5 | 69.3 | 51.4 |
| R-101-FPN | Object365 | x | 62.6 | 2077 | 11.6 | 62.5 | 44.1 | 53.3 | 2035 | 10.0 | 53.7 | 31.8 | 46.4 | 1546 | **8.2** | 47.7 | 65.4 | 52.1 |
| R-101-FPN | LVIS | x | 64.1 | 1910 | 12.7 | **63.8** | **45.9** | **54.8** | 2338 | 5.5 | **53.7** | **32.2** | **46.6** | **1583** | 4.8 | 47.6 | **72.1** | **53.8** |
| (C) | | | | | | | | | | | | | | | | | | |
| X-101-FPN | IN1k | x | **69.0** | 1832 | **18.1** | **65.0** | 46.4 | 55.7 | 1894 | 8.7 | 52.8 | 37.3 | 47.6 | 1231 | 1.1 | 45.9 | 66.8 | 51.1 |
| X-101-FPN | COCO | x | 58.0 | **1594** | 11.5 | 60.6 | 47.4 | 54.0 | 2029 | 4.1 | 52.5 | 33.5 | 46.2 | 1542 | 6.3 | 45.7 | 66.7 | 51.0 |
| X-101-FPN | Object365 | x | 61.0 | 1774 | 15.9 | 63.6 | 47.3 | 55.5 | **1864** | **9.6** | 51.8 | 33.6 | 45.7 | **1140** | **7.8** | 46.5 | 69.1 | 52.2 |
| X-101-FPN | LVIS | x | 63.5 | 1668 | 12.7 | 62.9 | **48.4** | 55.6 | 1907 | 5.5 | **54.4** | **36.1** | **48.3** | 1398 | 4.5 | **48.8** | **71.8** | **54.6** |

We compared three different network models: ResNet50 with FPN, ResNet101 with FPN, and ResNetX101 with FPN. Bold font indicates the best results among each comparison group.
'x' denotes the model without the class-agnostic head.

To provide a comprehensive visualization of the model's cross-species detection capabilities, Figure 5 presents qualitative results of these experiments. These visual examples offer a glimpse into the model's ability to identify diseases and pests in species it has not been explicitly trained on, demonstrating its potential for broader applicability and practical use in real-world scenarios.

TABLE 7 Sensitivity analysis on the task training order.

| Architecture | PTD | Order | Task 1~3 (Tomato) | | Task 4 (cross-species task) (Paprika) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | K | Δ | P | Δ | C | Δ | K | Δ |
| R-50-C4 | IN1k | Task 1, Task 2, Task 3, Task 4 | 42.52 | – | 42.96 | – | 62.78 | – | 47.92 | – |
| R-50-C4 | IN1k | Task 3, Task 2, Task 1, Task 4 | 41.37 | -1.15 | 40.8 | -2.16 | 63.03 | 0.25 | 46.4 | -1.52 |
| R-50-FPN | IN1k | Task 1, Task 2, Task 3, Task 4 | 44.35 | – | 44.80 | – | 65.70 | – | 50.03 | – |
| R-50-FPN | IN1k | Task 3, Task 2, Task 1, Task 4 | 42.38 | -1.97 | 42.37 | -2.43 | 58.64 | -7.06 | 46.44 | -3.59 |
| R-50-C4 | COCO | Task 1, Task 2, Task 3, Task 4 | 44.24 | – | 45.90 | – | 64.24 | – | 50.48 | – |
| R-50-C4 | COCO | Task 3, Task 2, Task 1, Task 4 | 41.46 | -2.78 | 41.33 | -4.57 | 62.16 | -2.08 | 46.54 | -3.94 |
| R-50-FPN | COCO | Task 1, Task 2, Task 3, Task 4 | 43.58 | – | 44.92 | – | 66.31 | – | 50.27 | – |
| R-50-FPN | COCO | Task 3, Task 2, Task 1, Task 4 | 42.04 | -1.54 | 41.84 | -3.08 | 65.97 | -0.34 | 47.87 | -2.40 |
| R-50-FPN | LVIS | Task 1, Task 2, Task 3, Task 4 | 47.69 | – | 47.19 | – | 69.12 | – | 52.67 | – |
| R-50-FPN | LVIS | Task 3, Task 2, Task 1, Task 4 | 43.17 | -4.52 | 43.9 | -3.29 | 68.23 | -0.89 | 49.98 | -2.69 |
| R-101-FPN | LVIS | Task 1, Task 2, Task 3, Task 4 | 46.63 | – | 47.68 | – | 72.13 | – | 53.80 | – |
| R-101-FPN | LVIS | Task 3, Task 2, Task 1, Task 4 | 44.48 | -2.15 | 44.37 | -3.31 | 68.17 | -3.96 | 50.32 | -3.48 |
| X-101-FPN | LVIS | Task 1, Task 2, Task 3, Task 4 | 48.31 | – | 48.87 | – | 71.86 | – | 54.61 | – |
| X-101-FPN | LVIS | Task 3, Task 2, Task 1, Task 4 | 47.26 | -1.05 | 46.56 | -2.31 | 71.57 | -0.29 | 52.81 | -1.80 |

We alternately show the performance of different training orders in the same model. Δ represents the differences caused by the order of learning. For the notation of evaluation metrics, please refer to Table 2.
'-' indicates that the difference is not applicable to the current situation.




FIGURE 5
Qualitative results on cross-species detection study. **(A)**. Training on tomato dataset and test on paprika dataset. **(B)**. Training on paprika dataset and test on tomato dataset. The sample number is indicated in the top left corner of each subplot. Best view in color.

## 5 Discussion

The task of object detection is typically divided into two subtasks: classification and localization. In this section, we discuss the limitations of our method in these two subtasks, including open issues. Finally, we present several potential avenues for future research.

### 5.1 Localization

Addressing the localization problem of unknown objects is a key challenge in open-world object detection. The main difficulty lies in the lack of prior knowledge about the unknown classes in the model. As a result, it is challenging to directly learn their features and location information from the training data. Our method improved the model's ability to detect unknown diseases. However, qualitative experimental results showed that the unknown recall is still below 30%. A unified unknown detection evaluation protocol is even more difficult than finding unknown diseases. As shown in Figure 6, these unknown detection results are treated as false positive boxes under the current ground truth, even though our model has already localized these suspicious regions.

The controversy surrounding the evaluation criteria for unknown class localization stems from the lack of consistent standards and consensus. This controversy is formed when

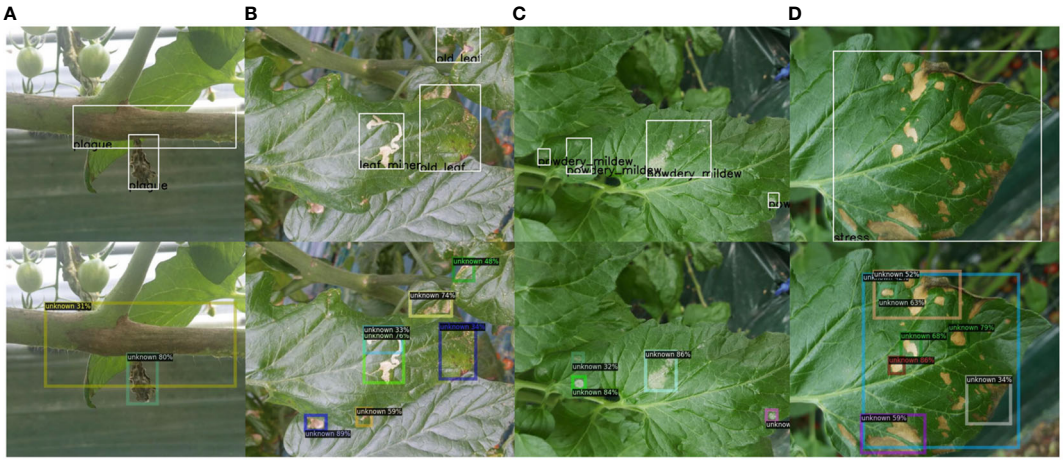



FIGURE 6
Qualitative results for unknown instances from our dataset. We present four pairs of examples **(A–D)**. The first row displays the image and annotations, while the second row represents our detection results. Best view in color.

annotating datasets. Our previous work (Dong et al., 2022) discussed how to efficiently label plant diseases. We believe that different disease symptoms should adopt different labeling strategies, and we verified this scheme's effectiveness through several experiments. However, these annotation strategies and evaluation schemes were designed for known categories. To the best of our knowledge, no related work discusses the localization of unknown classes in plant disease detection tasks. Additionally, the definition and scope of unknown classes also introduce subjectivity and uncertainty, further contributing to the controversy of evaluation criteria. Therefore, further research and consensus-building are needed to establish consistent and fair evaluation criteria for assessing the localization performance of unknown classes.

## 5.2 Classification

We developed a dynamic open-world detector since plant growth is a dynamic process. However, plant disease dynamics are more complex than we imagined. Some diseases may exhibit different symptoms at different stages of growth, leading to a challenging feature expression. Additionally, different diseases may also exhibit similar symptoms at different stages, which can be due to different pathogens (such as bacteria, fungi, viruses, etc.) or environmental factors. We list some common examples of tomato diseases with similar symptoms:

Yellowing symptoms: Yellowing is a common symptom of many plant diseases, including viral infections, fungal diseases, and nutrient deficiencies. Different pathogens or causes may lead to yellowing of plant leaves or other tissues, but their pathological processes and treatment methods may differ completely.

Leaf spot diseases: Many pathogens can cause similar leaf spot diseases, such as fungal and bacterial leaf spots. They produce similar spots or patches on the leaves, but the pathogens and pathogenic mechanisms behind them are different.

Rotting symptoms: Rotting is a common symptom caused by various diseases or pathogens, including bacterial soft rot, fungal rot, and rotting caused by certain environmental factors. Although they manifest as the decay of plant tissues, the specific causes may be different.

Similar symptoms may also occur in the detection of cross-species diseases. In addition, the leaves of different plants are different in a healthy state. However, diseases may force the leaves of different species to deform to the same symptom at the final state. In this case, even experts also struggle to distinguish them. Therefore, deep learning models may still face the same challenges in accurately differentiating them. Figure 7 shows some cases with similar symptoms but different species. When testing for diseases on paprika leaves using a model trained on the tomato dataset, all suspicious regions should have been detected as unknown. However, some unknown regions are mistakenly detected as gray mold due to similar symptoms. Although the category is correct, these instances of gray mold are treated as false positives.

These pieces of evidence prove that deep learning models can offer advantages in distinguishing similar disease symptoms but are not infallible. Domain expertise and collaboration with experts remain critical in evaluating and validating the model's predictions. The model's success still depends on the availability of quality training data and the complexity of the differentiation task. Another limitation we encountered is the challenge of obtaining additional high-quality datasets for plant disease detection to validate generalizability further. Despite this constraint, we have conducted validation using the COCO dataset to showcase the method's performance. For more details, please refer to our code repository.

## 5.3 Future works

We present some promising approaches to tackle classification problems. Recently, Du et al. (2022a) introduced spatial-temporal unknown distillation (STUD), a model designed to detect unknown objects in videos by establishing spatial-temporal context. STUD (Du et al., 2022a) utilizes time series features to evaluate the relationship between the current frame and the reference frame, reducing the occurrence of classification errors. In real-world agricultural practices, the same species is commonly planted in one area. Therefore, considering the spatial-temporal context to determine the species category can effectively narrow down the range of disease classifications. Another intriguing direction is utilizing large visual language models (Radford et al., 2021), renowned for their impressive zero-shot detection capabilities, making them highly suitable for identifying unknown categories. A recent study (Wortsman et al., 2022) demonstrated that fine-tuning a large-scale visual language model through weight integration performs well not only on specific downstream tasks but also maintains its ability to recognize unknown targets. Consequently, embedding a large language-vision model into open-world detection tasks has the potential to enhance the model's robustness. We encourage the community to pay attention to these promising methods and apply them to plant disease detection tasks.

# 6 Conclusions

In this study, we introduced a new paradigm called open world plant disease detector. This novel detection paradigm enables the detection of unknown diseases and allows for the dynamic updating of new knowledge. This paradigm breaks the closed-set, static open-set settings of conventional plant disease detectors. We observed that detectors trained on complex object detection datasets can enhance the detection performance for unknown classes, and the category-agnostic head further improved the recall rate for unknown diseases. Additionally, cross-species disease detection experiments have demonstrated that our model can comprehend the concept of diseases and successfully detect them across different species. Extensive ablation experiments validated the effectiveness



FIGURE 7
Qualitative results for unknown instances from our dataset. These instances of gray mold should have been detected as unknown. The sample number is indicated in the top left corner of each subplot. Best view in color.

of our proposed method. Furthermore, we thoroughly discussed the existing open challenges in plant disease detection and offered insightful perspectives. We strongly encourage researchers and practitioners to address the current challenges that remain.

## Data availability statement

The original contributions presented in the study are included in the article/supplementary files. Further inquiries can be directed to the corresponding authors.

## Author contributions

JD designed the method, performed the experiments, and wrote the manuscript. KH and SY advised in the design of the system and analysed the annotation strategies to find the best method for efficient plant disease detection. AF and DP provided support in the data collection and proofreading article. JY presents some conceptual suggestions and domain knowledge. All authors contributed to the article and approved the submitted version.

## Funding

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (No. 2019R1A6A1A09031717); by the Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry(IPET) and Korea Smart Farm R&D Foundation (KosFarm) through Smart Farm Innovation Technology Development Program, funded by Ministry of Agriculture, Food and Rural Affairs(MAFRA) and Ministry of Science and ICT(MSIT), Rural Development Administration(RDA)(1545027569); and by the National Research Foundation of Korea(NRF) grant funded by the Korea government (MSIT). (NRF-2021R1A2C1012174).

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

# References


Alruwaili, M., Siddiqi, M. H., Khan, A., Azad, M., Khan, A., and Alanazi, S. (2022). RTF-RCNN: An architecture for real-time tomato plant leaf diseases detection in video streaming using Faster-RCNN. *Bioengineering* 9, 565. doi: 10.3390/bioengineering9100565

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Miami, FL, USA. 248–255.

Dhamija, A., Gunther, M., Ventura, J., and Boult, T. (2020). "The overlooked elephant of object detection: Open set," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, Snowmass Village, CO, USA. 1021–1030.

Dong, J., Lee, J., Fuentes, A., Xu, M., Yoon, S., Lee, M. H., et al. (2022). Data-centric annotation analysis for plant disease detection: Strategy, consistency, and performance. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.1037655

Du, X., Wang, Z., Cai, M., and Li, Y. (2022b). Vos: Learning what you don't know by virtual outlier synthesis. In *International Conference on Learning Representations* (OpenReview, Online) (arXiv preprint arXiv). doi: 10.48550/arXiv.2202.01197

Du, X., Wang, X., Gozum, G., and Li, Y. (2022a). "Unknown-aware object detection: Learning what you don't know from videos in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, Louisiana, USA. 13678–13688.

Fenu, G., and Malloci, F. M. (2021). DiaMOS plant: A dataset for diagnosis and monitoring plant disease. *Agronomy* 11, 2107. doi: 10.3390/agronomy11112107

Fuentes, A., Lee, J., Lee, Y., Yoon, S., and Park, D. S. (2017). "Anomaly detection of plant diseases and insects using convolutional neural networks," in *Proceedings of the International Society for Ecological Modelling Global Conference*, Jeju Island, South Korea.

Fuentes, A., Yoon, S., Kim, T., and Park, D. S. (2021). Open set self and across domain adaptation for tomato disease recognition with deep learning techniques. *Front. Plant Sci.* 12. doi: 10.3389/fpls.2021.758027

Fuentes, A. F., Yoon, S., Lee, J., and Park, D. S. (2018). High-performance deep neural network-based tomato plant diseases and pests diagnosis system with refinement filter bank. *Front. Plant Sci.* 9. doi: 10.3389/fpls.2018.01162

Geng, C., Huang, S.-J., and Chen, S. (2020). Recent advances in open set recognition: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 3614–3631. doi: 10.1109/TPAMI.2020.2981604

Girshick, R. (2015). "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, Santiago, Chile. 1440–1448.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Columbus, OH, USA. 580–587.

Gupta, A., Dollar, P., and Girshick, R. (2019). "Lvis: A dataset for large vocabulary instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Long Beach, California, USA. 5356–5364.

Gupta, A., Narayan, S., Joseph, K., Khan, S., Khan, F. S., and Shah, M. (2022). "Ow-detr: Open-world detection transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, Louisiana, USA. 9235–9244.

Hayes, T. L., Kafle, K., Shrestha, R., Acharya, M., and Kanan, C. (2020). "Remind your neural network to prevent catastrophic forgetting," in *Computer Vision–ECCV 2020: 16th European Conference*, Glasgow, UK, August 23–28, 2020. 466–483.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, Venice, Italy. 2961–2969.

Hendrycks, D., and Gimpel, K. (2016). A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations* (San Juan, Puerto Rico: arXiv preprint arXiv). doi: 10.48550/arXiv.1610.02136

Joseph, K., Khan, S., Khan, F. S., and Balasubramanian, V. N. (2021). "Towards open world object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, United States. 5830–5840.

Kuo, W., Cui, Y., Gu, X., Piergiovanni, A., and Angelova, A. (2023). "Open-vocabulary object detection upon frozen vision and language models," in *The Eleventh International Conference on Learning Representations*, Kigali, Rwanda.

Lecun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. (2006). A tutorial on energy-based learning. *Predicting structured Data* (Massachusetts USA: MIT Press) 1.

Li, Z., and Hoiem, D. (2017). Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 2935–2947. doi: 10.1109/TPAMI.2017.2773081

Li, K., Lin, J., Liu, J., and Zhao, Y. (2020). Using deep learning for Image-Based different degrees of ginkgo leaf disease classification. *Information* 11, 95. doi: 10.3390/info11020095

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, HI, USA. 2117–2125.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). "Microsoft coco: Common objects in context," in *European conference on computer vision*, Zurich, Switzerland. 740–755.

Liu, J., and Wang, X. (2021). Plant diseases and pests detection based on deep learning: a review. *Plant Methods* 17, 1–18. doi: 10.1186/s13007-021-00722-9

Liu, W., Wang, X., Owens, J., and Li, Y. (2020). Energy-based out-of-distribution detection. *Adv. Neural Inf. Process. Syst.*, 21464–21475.

Livio, M. (2017). *Why?: What makes us curious* (New York, United States: Simon and Schuster).

Ma, Y., Li, H., Zhang, Z., Guo, J., Zhang, S., Gong, R., et al. (2023b). "Annealing-based label-transfer learning for open world object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, Canada. 11454–11463.

Ma, S., Wang, Y., Wei, Y., Fan, J., Li, T. H., Liu, H., et al. (2023a). "CAT: LoCalization and identificAtion cascade detection transformer for open-world object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, Canada. 19681–19690.

Mathew, M. P., and Mahesh, T. Y. (2022). Leaf-based disease detection in bell pepper plant using YOLO v5. *Signal Image Video Process.* 16, 841–847. doi: 10.1007/s11760-021-02024-y

Miller, D., Nicholson, L., Dayoub, F., and Sünderhauf, N. (2018). "Dropout sampling for robust object detection in open-set conditions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia. 3243–3249.

Murugeswari, R., Anwar, Z. S., Dhananjeyan, V. R., and Karthik, C. N. (2022). "Automated sugarcane disease detection using faster RCNN with an android application," in *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India. 1–7.

Nazki, H., Yoon, S., Fuentes, A., and Park, D. S. (2020). Unsupervised image translation using adversarial networks for improved plant disease recognition. *Comput. Electron. Agric.* 168, 105117. doi: 10.1016/j.compag.2019.105117

Panchal, A. V., Patel, S. C., Bagyalakshmi, K., Kumar, P., Khan, I. R., and Soni, M. (2023). Image-based plant diseases detection using deep learning. *Materials Today: Proc.* 80, 3500–3506. doi: 10.1016/j.matpr.2021.07.281

Prabhu, A., Torr, P. H., and Dokania, P. K. (2020). "Gdumb: A simple approach that questions our progress in continual learning," in *Computer Vision–ECCV 2020: 16th European Conference*, Glasgow, UK, August 23–28, 2020. 524–540.

Priyadharshini, G., and Dolly, D. R. J. (2023). "Comparative investigations on tomato leaf disease detection and classification using CNN, R-CNN, fast R-CNN and faster R-CNN," in *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, TamilNadu, INDIA. 1540–1545.

Qiao, Y., Valente, J., Su, D., Zhang, Z., and He, D. (2022a). AI, sensors and robotics in plant phenotyping and precision agriculture. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.1064219

Qiao, Y., Zhang, Z., Guo, Y., and He, D. (2022b). "Deep learning based grape mildew disease severity classification," in *2022 ASABE Annual International Meeting*, Houston, Texas, USA. 1.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., et al. (2021). "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. 8748–8763.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. (2017). "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA. 2001–2010.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, Montreal Convention Center, Montreal, Canada.

Seetharaman, K., and Mahendran, T. (2022). Leaf disease detection in banana plant using gabor extraction and region-based convolution neural network (RCNN). *J. Institution Engineers (India): Ser. A* 103, 501–507. doi: 10.1007/s40030-022-00628-2

Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., et al. (2019). "Objects365: A large-scale, high-quality dataset for object detection," in *Proceedings of the IEEE/CVF international conference on computer vision* (Seoul, Korea). 8430–8439.

Shoaib, M., Shah, B., Ei-Sappagh, S., Ali, A., Ullah, A., Alenezi, F., et al. (2023). An advanced deep learning models-based plant disease detection: A review of recent research. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1158933

Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., and Batra, N. (2020). "PlantDoc: a dataset for visual plant disease detection," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD* (Hyderabad, India).

Vaze, S., Han, K., Vedaldi, A., and Zisserman, A. (2021). Open-set recognition: A good closed-set classifier is all you need. In *International Conference on Learning Representations*. doi: 10.48550/arXiv.2110.06207

Wang, H., Shang, S., Wang, D., He, X., Feng, K., and Zhu, H. (2022). Plant disease detection and classification method based on the optimized lightweight YOLOv5 model. *Agriculture* 12, 931. doi: 10.3390/agriculture12070931

Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., et al. (2022). "Robust fine-tuning of zero-shot models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, Louisiana, USA. 7959–7971.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. (2019). Detectron2. https://github.com/facebookresearch/detectron2.

Wu, Z., Lu, Y., Chen, X., Wu, Z., Kang, L., and Yu, J. (2022). "UC-OWOD: unknown-classified open world object detection," in *Computer Vision–ECCV 2022: 17th European Conference*, Tel Aviv, Israel, October 23–27, 2022. 193–210.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, HI, USA. 1492–1500.

Xiong, H., Lu, H., Liu, C., Liu, L., Cao, Z., and Shen, C. (2019). "From open set to closed set: Counting objects by spatial divide-and-conquer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, Korea. 8362–8371.

Xu, M., Kim, H., Yang, J., Fuentes, A., Meng, Y., Yoon, S., et al. (2023). Embrace limited and imperfect training datasets: opportunities and challenges in plant disease recognition using deep learning. *Front. Plant Science* 14, 1225409. doi: 10.48550/arXiv.2305.11533

Zhang, S., Huang, W., and Zhang, C. (2019). Three-channel convolutional neural networks for vegetable leaf disease recognition. *Cogn. Syst. Res.* 53, 31–41. doi: 10.1016/j.cogsys.2018.04.006

Zohar, O., Wang, K.-C., and Yeung, S. (2023). "PROB: Probabilistic objectness for open world object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, Canada. 11444–11453.

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 13 October 2023
DOI 10.3389/fpls.2023.1238722

Check for updates

OPEN ACCESS

EDITED BY
Jakub Nalepa,
Silesian University of Technology, Poland

REVIEWED BY
Hitesh Mohapatra,
KIIT University, India
Pushkar Gole,
University of Delhi, India

*CORRESPONDENCE
Sook Yoon
syoon@mokpo.ac.kr
Hyongsuk Kim
hskim@jbnu.ac.kr

RECEIVED 12 June 2023
ACCEPTED 22 September 2023
PUBLISHED 13 October 2023

CITATION
Dong J, Fuentes A, Yoon S, Kim H and Park DS (2023) An iterative noisy annotation correction model for robust plant disease detection. *Front. Plant Sci.* 14:1238722. doi: 10.3389/fpls.2023.1238722

COPYRIGHT
© 2023 Dong, Fuentes, Yoon, Kim and Park. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# An iterative noisy annotation correction model for robust plant disease detection

Jiuqing Dong[1,2], Alvaro Fuentes[1,2], Sook Yoon[3]*, Hyongsuk Kim[1,2]* and Dong Sun Park[1,2]

[1]Department of Electronic Engineering, Jeonbuk National University, Jeonju, Republic of Korea, [2]Core Research Institute of Intelligent Robots, Jeonbuk National University, Jeonju, Republic of Korea, [3]Department of Computer Engineering, Mokpo National University, Muan, Republic of Korea

Previous work on plant disease detection demonstrated that object detectors generally suffer from degraded training data, and annotations with noise may cause the training task to fail. Well-annotated datasets are therefore crucial to build a robust detector. However, a good label set generally requires much expert knowledge and meticulous work, which is expensive and time-consuming. This paper aims to learn robust feature representations with inaccurate bounding boxes, thereby reducing the model requirements for annotation quality. Specifically, we analyze the distribution of noisy annotations in the real world. A teacher-student learning paradigm is proposed to correct inaccurate bounding boxes. The teacher model is used to rectify the degraded bounding boxes, and the student model extracts more robust feature representations from the corrected bounding boxes. Furthermore, the method can be easily generalized to semi-supervised learning paradigms and auto-labeling techniques. Experimental results show that applying our method to the Faster-RCNN detector achieves a 26% performance improvement on the noisy dataset. Besides, our method achieves approximately 75% of the performance of a fully supervised object detector when 1% of the labels are available. Overall, this work provides a robust solution to real-world location noise. It alleviates the challenges posed by noisy data to precision agriculture, optimizes data labeling technology, and encourages practitioners to further investigate plant disease detection and intelligent agriculture at a lower cost. The code will be released at https://github.com/JiuqingDong/TS_OAMIL-for-Plant-disease-detection.

KEYWORDS

**annotation correction, plant disease detection, auto-labeling, noisy labels, teacher-student model**

# 1 Introduction

According to the United Nations, the world population reached 8 billion in mid-November 2022 (Pison, 2022). Meanwhile, hunger-related fatalities rose and reached 4 million in 2020, 10 times the number of COVID-19 fatalities in the same period (He and Krainer, 2020). Given those antecedents, it is essential to find ways to feed a growing population while limiting environmental damage and improving the yield and quality of agricultural products (Smith et al., 2022). Nonetheless, this endeavor presents a formidable challenge given the susceptibility of crops to afflictions and stressors, both of which have the potential to engender detrimental economic repercussions and reductions in production output. Consequently, the timely identification of preliminary indications of disease and stress factors in vegetation assumes paramount significance in instituting optimal conditions conducive to crop cultivation.

Recent frontiers in non-invasive sensor technology and image processing methodologies provide potential remedies for the aforementioned challenges. Deep learning methods have shown great success in various tasks, such as plant state monitoring (Xu et al., 2021a; Bhise et al., 2022; Wang et al., 2022a; Dong et al., 2023; Shoaib et al., 2023; Tomaszewski et al., 2023), medical diagnosis (Yao et al., 2022; Nalepa et al., 2023), cell variation (Rahman et al., 2021), and flora (Evangelisti et al., 2021; Ganesh et al., 2022). These achievements frequently hinge upon the extraction of visual cues from images and the provision of precise annotations. Nevertheless, annotating these domain-specific datasets is not as simple as identifying cats and dogs. Acquiring an accurately annotated dataset relies on expert knowledge, which is only sometimes feasible. Deploying current deep learning-based methods in real-world applications may suffer primarily from limited and imperfect data (Xu et al., 2023). In a real scenario, practitioners without computer vision knowledge lack experience in annotating high-quality boxes, and annotators without domain knowledge have difficulties in annotating accurate object boxes. Annotation cost would be significantly high if domain experts were to annotate the entire dataset. Embracing these imperfect annotations is a promising strategy that has not received sufficient attention (Dubel et al., 2023). We need to consider the practical problem of whether it is worth spending more on the expert and computational costs to get a better performance or applying techniques to mitigate these issues.

We attempt to answer this question in the general case. As illustrated in previous work (Dong et al., 2022), enhancing performance through increased computational expenditure is prevalent in computer vision techniques. For instance, as observed in (Liu et al., 2022c), the image classification accuracy on ImageNet only improved by 6%, while the number of parameters increased from 88 million to 30 billion. Compared to computational cost, labor cost for annotation is more substantial. Particularly, researchers refine the label set multiple times just for a weak improvement in a specific task. Therefore, refinement labeling is high-cost and low-reward, while spending a considerable cost for a slight improvement in practical applications is unwise.

Compared to class noise, fully supervised object detector (FSOD) performance is more susceptible to inaccurate localization (Dong et al., 2022). Note that FSOD means that each instance is assigned an accurate label during the training phase. Moreover, localization noise is almost unavoidable compared to class noise, because not all researchers can afford such colossal labor or time costs and organize a professional processing line for annotation as the COCO team (Lin et al., 2014) has done. As a result, researchers are often faced with dealing with a noisy label set. In this study, we address these challenges by considering a method capable of handling noisy annotations to mitigate their impact. This allows us to relax the strict annotation standards, which in turn benefits intelligent agricultural practitioners by reducing the threshold for their involvement.

Based on the above-mentioned, in this paper, we propose an annotation correction strategy based on the teacher-student learning paradigm, which is effective in two training settings. Our method improves the model's performance in supervised learning tasks by correcting for noisy localization noise. While in semi-supervised learning tasks, it can act as an automatic annotator, generating accurate pseudo-labels. Figure 1 demonstrates two main application scenarios of our method.

Regarding the correction of class noise, there are already a few methods for training accurate Deep Neural Networks (DNNs) under noisy labels (Li et al., 2020; Mao et al., 2021; Xu et al., 2021c; Song et al., 2022; Huang et al., 2023; Zhang et al., 2023). A significant line of research focuses on the classification task, which develops various techniques to deal with noisy labels, such as sample selection (Xia et al., 2021), robust regularization (Gudovskiy et al., 2021; Huang et al., 2023), and robust loss

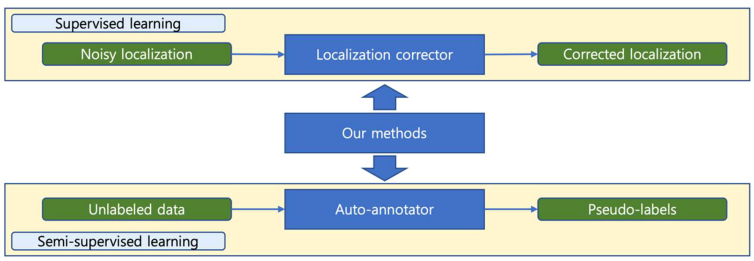



FIGURE 1
Two application scenarios of our method. In supervised learning, our method can correct noisy locations, while in semi-supervised learning, it can generate accurate pseudo-labels from unlabeled data. Our method plays different roles in different settings.

functions (Jiang et al., 2021). In contrast, there are significantly fewer studies focused on addressing localization noise. Recently, some efforts have extended the accumulated experience in classification to object detection tasks, such as class noise correction (Xu et al., 2021c), missing label correction (Xu et al., 2019), and noisy localization correction (Liu et al., 2022a). However, these methods mainly focus on general vision datasets, such as MS-COCO (Lin et al., 2014), PASCAL VOC (Everingham et al., 2010), and ImageNet (Deng et al., 2009), rather than domain-specific datasets. Although, there are a large number of frontier works in plant disease detection (Fuentes et al., 2018; Nazki et al., 2020; Fuentes et al., 2021; Xu et al., 2021b; Khakimov et al., 2022; Priyadharshini and Dolly, 2023), which can achieve 90% or even higher performance on their respective datasets, these methods assume in advance that they are trained on well-annotated datasets. Unlike previous approaches, we emphasize that our model is trained on noisy annotations.

Our paper further studies the distribution of localization noise and the noise synthesis rules. Intuitively, the localization of a small object noise seems more severe than large objects. Based on this motivation, this paper provides an insightful analysis of the distribution of location noise and the relationship between noise distribution and bounding box size. Previous research work generally synthesized noise by perturbing clean bounding boxes (Liu et al., 2022a), while synthesized noise follows a uniform distribution $relative_{noise} \sim U(-r, r)$. The $r$ is a parameter to control the noise level. We argue that such a noise synthesis rule without considering object size is unreasonable. Unlike (Liu et al., 2022a), our method is trained on a dataset with real-world noise and synthesized noise following the real-world distribution.

To build a robust detector, we expect the model learns from corrected labels rather than noisy labels. Therefore, we introduce an iterative teacher-student learning framework on the correction network (Liu et al., 2022a). Teacher-student learning is a learning paradigm, introduced in knowledge distillation (Gou et al., 2021), where knowledge is usually distilled from a teacher network to improve the feature representation of students (Wang et al., 2022b). Typically, a teacher model is more complicated than a student, but a simple student network can achieve comparable performance to a teacher. Unlike most existing teacher-student algorithms (Hu et al., 2022), teacher and student networks, in this work, hold the same architecture but different parameters. We train a teacher network on noisy datasets and the corrected annotations are used as a supervised signal for the student network. In addition, we found that labels are still noisy after being corrected by OA-MIL (Liu et al., 2022a) (refer to the red line in Figure 2). Therefore, a teacher-student learning paradigm is adopted to correct noisy labels iteratively. In other words, the performance improvement of the student model comes from the optimized label set, which can be regarded as the additional knowledge of the teacher model. Our method avoids manually refining noisy labels, thereby reducing annotation costs.

To further reduce annotation costs, semi-supervised and unsupervised learning algorithms are usually used to learn the feature representation from unlabeled data. However, unsupervised learning is often used to tackle more complex tasks such as domain adaptation (Liu et al., 2022b) and learning from compression (He et al., 2022), which requires a large-scale dataset. In addition, unsupervised learning cannot provide an intuitive category label like semi-supervised learning, which is not friendly to plant disease detection. Therefore, semi-supervised learning is more suitable for handling general plant disease detection tasks. Semi-supervised object detection tasks aim to train an object detector with many image-level annotations and a few box-level annotations (Li et al., 2022) and generate many instance-level pseudo-labels. Suppose the pseudo-labels generated during the semi-supervised learning process are regarded as a collection of noisy labels. Our method can naturally extend to semi-supervised learning tasks, requiring only limited box-level annotations and no extra image-level annotations. To the best of our knowledge, this is the first semi-supervised learning method in plant disease detection, even though it is not explicitly designed for the semi-supervised learning task.

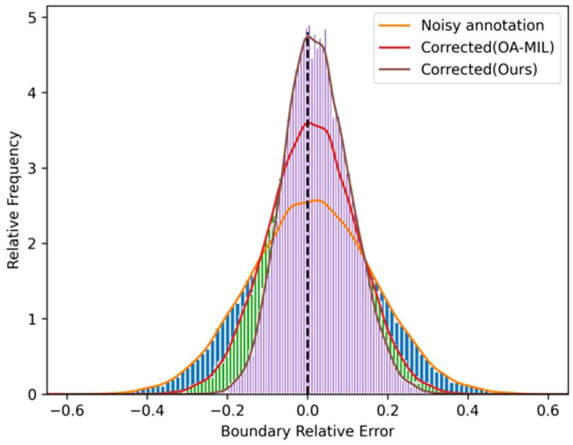



FIGURE 2
Distributions of relative boundary coordinate errors for noisy annotations and our corrected ones.

Overall, our primary contributions can be summarized as follows:

1. We investigate the distribution of location noise in real-world plant data annotation and provide insightful analyses.
2. An annotation correction network based on the iterative teacher-student learning paradigm is proposed to offset the impact of noise on model performance by correcting imprecise labeled boxes.
3. Our method can be easily extended to semi-supervised learning tasks and automatic labeling. For instance, we achieve approximately 75% of the performance of fully supervised learning methods using only 1% of accurate labels.
4. Our approach lowers the labeling quality requirements for practitioners, by improving the robustness of the model to localization noise. Additionally, it is anticipated to advance applications related to location-based tasks.

The remaining sections of the paper are organized as follows. Section 2 introduces the dataset used for noise analysis, the results of the noise analysis, and the correction methods. Section 3 presents the experimental results, including the correction results, experimental results in the semi-supervised setting, and ablation experiments. Section 4 discusses relevant topics, limitations, and future work. Section 5 provides a conclusion for the paper. We have included all qualitative results of the experiments in the appendix, denoted as Figure A in the main text, where A represents the appendix.

# 2 Materials and methods

## 2.1 Datasets

The paprika disease dataset (Dong et al., 2022) was used to evaluate our methods. As previously noted, researchers often engage in iterative model optimization and label set refinement throughout the task processing to enhance the ultimate performance. Indeed, the dataset in (Dong et al., 2022) was refined through multiple mutual verifications between plant experts and artificial intelligence experts to achieve a well-annotated dataset finally. Therefore, based on the assumption that the paprika disease dataset is clean, we conducted experiments and analyses. Please note that the unrefined raw label set is called the real-world noisy dataset, and the well-annotated label set is called the clean dataset. Both followed the split strategy of the dataset in (Dong et al., 2022). The Paprika disease dataset consists of five disease categories, with 5,928 images.

If the pseudo-labels generated during the semi-supervised learning process are considered as noisy labels, our method can be employed to correct location noise. However, besides positional noise, pseudo-labels also come with class noise. Therefore, it is essential to evaluate the performance of positional correction using a single-class object detection dataset to avoid the influence of multi-class classification problems on our model. Unlike the paprika disease dataset, the Global Wheat Head Detection (GWHD2021) (David et al., 2020; David et al., 2021) dataset only requires the distinction of wheat heads and does not involve multi-class classification. Thus, we evaluated our method in a semi-supervised setting using GWHD2021. Please note that GWHD2021 is only used for evaluating the performance of our method when extended to the semi-supervised setting. While handling the labels, we found that some images had no annotation information, totaling 128. Therefore, we used a total of 6,387 images that had annotations. We followed their dataset split scheme. More details about the dataset are presented in Table 1.

## 2.2 Preliminaries on object detection

Most object detectors follow the dense prediction paradigm, usually with an ingenious loss function to guide them to predict correctly. Generally, a loss function in object detection mainly consists of two parts: classification loss and regression loss. Classification loss is used to distinguish categories, while regression loss is designed to localize objects, which can be abstracted as Equation 1:

$$L(\hat{C}_i, \hat{B}_i) = \lambda_{cls} \sum_i L_{cls}(\hat{C}_i,\ C_i) + \lambda_{reg} \sum_i \mathbb{I}_{\{C_i \neq 0\}} L_{reg}(\hat{B}_i,\ B_i) \quad (1)$$

where $\lambda$ denotes the normalization and reweighting factor. $L_{cls}$ and $L_{reg}$ are classification loss and regression loss, $\hat{C}_i$ and $\hat{B}_i$ denote the prediction results of category and location, $C_i$ and $B_i$ denote the ground truth of category and location. $\mathbb{I}_{\{C_i \neq 0\}}$ is the binary function, being one if $C_i \neq 0$ (foreground) and zero otherwise (background).

In fully supervised object detection, the default for evaluating the dataset is treated as accurate ground truth labels without considering possible human errors, which means that the regression object $B_i$ in Equation 1 is considered accurate. Due to the high cost to elaborate annotation, some publicly available datasets may satisfy the above criteria. However, some private and self-collected datasets usually do not meet these conditions. To explore the impact of noise in real-world scenarios, we studied the form of the distribution of the noisy label set.

TABLE 1 Details of datasets and splits.

| (A)Paprika dataset (Dong et al., 2022) | | | | |
|---|---|---|---|---|
| Category | Training | Validation | Test | Total |
| Blossom end rot | 933 | 117 | 133 | 1183 |
| Gray mold | 355 | 43 | 43 | 441 |
| Powdery mildew | 334 | 37 | 45 | 416 |
| Spider mite | 347 | 33 | 40 | 420 |
| Spotting disease | 2773 | 367 | 328 | 3468 |
| (B)GWHD2021 (David et al., 2020; David et al., 2021) | | | | |
| Category | Train | Validation | Test | Total |
| Wheat Head | 3605 | 1448 | 1334 | 6387 |

## 2.3 Location noise analysis

To quickly obtain a label set with real-world noise, we used the raw label set that has yet to be refined. Furthermore, our real-world noisy dataset shares almost the same class noise and missing labels as the clean ones to ensure a fair comparison. The only difference between noisy and clean label sets is that the location of bounding boxes in the noisy label set is more inaccurate. Therefore, we analyze real-world localization noise by traversing and matching boxes of the same category in two different versions of the annotation files. Specifically, in the clean and noisy label pairs, we compute the intersection between each clean bounding box and all noisy bounding boxes under the same category, selecting the largest intersection as the matching box. Then, we calculate the difference between this box and its corresponding matching box. For analysis, the absolute error is defined as the difference between each boundary of the clean bounding box and the corresponding boundary of its noise bounding box. The relative error is defined as the ratio of the absolute to the width or height of the clean bounding box. We analyzed the location noise distribution from the noisy label set, leading to three interesting observations: 1. The absolute error is proportional to the corresponding bounding box width or height; 2. The relative error related to the bounding box width and height follows a Gaussian distribution; 3. The mean of relative error is smaller than 0 in objects with a large width or height, while it is greater than 0 with a small width or height.

Figure 3A defines four boundaries and absolute error in a bounding box, where $(H, W)$ denotes image width and height, and $(h, w)$ denotes clean bounding box width and height. For small, middle, and large bounding box sizes, we refer to the definition of the COCO dataset (Lin et al., 2014). For example, $h$ is defined as small if $h/H \in (0, 0.1]$ , while $h$ is defined as large if $h/H \in (0.3, 1]$. Figure 3B shows the scatter plot of the relationship between the absolute error of four boundaries and the bounding box width or height.

Figures 4 shows the relative error distribution, the mean of absolute error by size, and correlation analysis for four types of boundary noises. Figure 4A, B indicate that boundary noises follow the normal distribution. We introduce the root mean square relative error $\gamma$ to measure the noise level. By computing the location noise for all boundaries, we get $\gamma = 0.15$ in a real-world noisy label set. The root mean square relative error is calculated by Equation 2:

$$\gamma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\Delta_i}{S_i}\right)^2} \tag{2}$$

where $\Delta_i$ represents the absolute error of the noise and $S_i$ represents the width or height of the bounding box. Assuming that the absolute error standard deviation is $\sigma$ , then $\sigma$ can be expressed by Equation 3:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\Delta_i)^2} \tag{3}$$

From (2) and (3), the relationship between $\sigma$ and $\gamma$ can be obtained by Equation 4:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[(\frac{\Delta_i}{S_i})^2 \cdot S_i^2]} \approx \gamma \cdot \sqrt{E(S_i)^2} \tag{4}$$

Figure 4C shows the relationship between $\sigma$ in Equation3 and bounding box size, where each point represents the local standard deviation within a specific range. For example, the abscissa of the first point represents the mean value of the bounding box size between 0 and 64, and the ordinate represents the $\sigma$ in Equation3 of the corresponding range. The slope of the fitted line is approximately the same as $\gamma$ , as expressed in Equation 4. Besides, Figure 4D shows that the noise distribution of the four boundaries is weakly correlated, thus, $\gamma$ can share on the four boundaries to synthesize the noise.

Overall, the relative error distribution follows a normal distribution with mean 0. To obtain the synthetic noise label set, we add absolute error $\Delta_i$ to the four boundaries of each bounding box, $\Delta_i \sim N(0, (\gamma \cdot S_i)^2)$ . We generated synthetic noise datasets by changing $\gamma$ and used them to train Faster-RCNN (Ren et al., 2015) detector. Figure 5 shows the performance according to the datasets used in the training. We can see that the performance of the

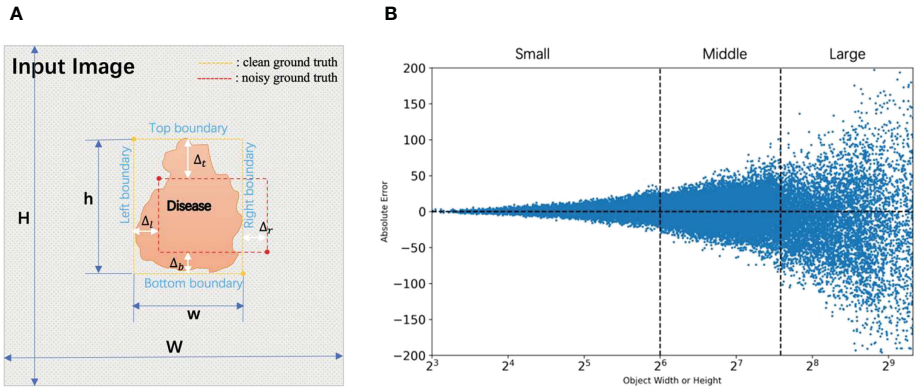



FIGURE 3
The definition and distribution of the boundary noise. **(A)** An example of the noise of four boundaries. Four white gaps correspond to four boundary noises ($\Delta_*$). **(B)** Scatter diagram of absolute error for each boundary coordinate with respect to corresponding bounding box width or height. Width and height are scaled to 640 x 640.

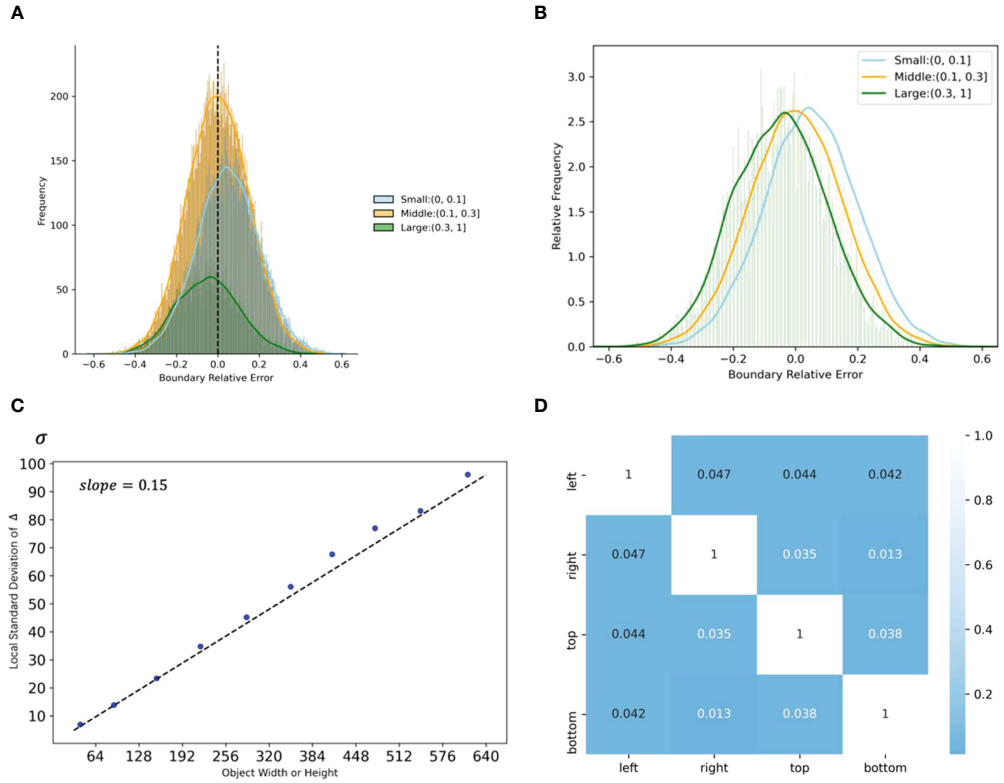



FIGURE 4
Analyses of location noise in the real-world noisy dataset. **(A)** The distribution of relative errors for different sizes of width and height. **(B)** Relative frequency of relative errors by size. **(C)** The standard deviation by sizes. **(D)** Correlation coefficients between relative errors of different boundaries. Please note that width and height are normalized.

detector trained with a real noisy dataset with $\gamma = 0.15$ and the detector trained with a synthetic dataset made with $\gamma = 0.15$ are relatively similar, indicating that our noise synthesis method and analysis are reasonable and realistic.

## 2.4 Robust detector via iterative noisy annotation correction model

Researchers often face a dataset with noisy location annotations due to limited labor or time budgets. Therefore, it is worth exploring how to utilize these noisy annotations effectively, as using them directly can lead to notable performance degradation. Liu et al. (2022a) propose an object-aware multi-instance learning (OA-MIL) approach that jointly optimizes an instance selector, an instance classifier, and an instance generator. They tried to generate high-quality proposals guided by noisy annotations. Specifically, in OA-MIL, an instance generator is used to generate multiple proposal boxes near each noisy annotated bounding box. These proposal boxes are generated to capture potential instances of the target object. Subsequently, an instance classifier is employed to assess the probability of containing the target object within these proposal boxes. The instance classifier predicts whether the proposal boxes contain the desired target. Then, the instances selector considered the proposals and the noisy ground truth to select the corrected bounding box. The corrected bounding box was used as a supervised signal to update network parameters. Finally, OA-MIL merges proposal boxes and the noisy ground truth, which is formulated as follows:

$$B_i^{correct} = \lambda \cdot B_i^{*} + (1 - \lambda) \cdot B_i \tag{5}$$

where $B_i^{correct}$ denotes the corrected bounding box. $B_i^{*}$ and $B_i$ denote the best proposal and noisy bounding box, respectively. $\lambda$ denotes the normalized weight factor.

Inaccurate ground truth dominates strong object localization priors. In some cases, poor instance initialization could render failure during training, resulting in no performance gains or even worse. Although the corrected bounding boxes have lower $\gamma$ than the noisy annotations, only relying on the annotations corrected to provide appropriate supervision still cannot achieve the optimal performance of the model.

To address the above issues, we employed a teacher-student learning framework in the training process. Specifically, we utilized OA-MIL (Liu et al., 2022a) as the teacher model, trained on the noisy dataset. Once the training of the teacher model was completed, we froze its parameters and employed it to perform inference on the training set, thereby generating relatively corrected labels. These corrected labels were then utilized to train the student model. The student model shared the same structure as the teacher

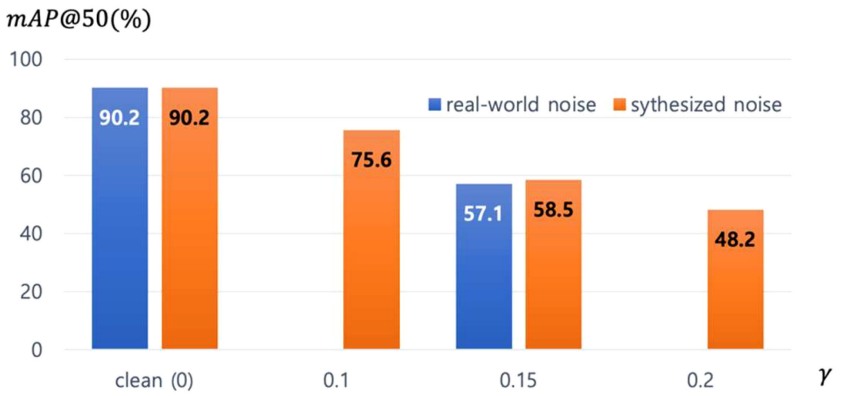



FIGURE 5
The impact of real-world noise versus synthesized noise on Faster-RCNN. The $\gamma$ equals 0.15 in real-world noise.

model but benefited from the more accurate corrected labels, enabling further refinement of the bounding box positions.

Once the student model was well-trained with the correct annotations, it served as the teacher detector to further refine the annotations. This iterative process of teacher-student learning and annotation refinement helped improve the accuracy and reliability of the final annotations. This process is repeated $N$ times. We have observed that increasing the number of iterations does not lead to a sustained improvement in model performance. In our experiments, we found that the label set typically reached its peak performance after three iterations. Beyond this point, the model's performance tended to plateau, indicating diminishing returns in terms of performance gains with further iterations. In this paper, the default setting for $N$ was three. The corrected annotations were used as a supervised signal for the student detector, which improved the model's performance by a continuously optimized label set. The localization performance of the student detector was thus improved. Compared to Faster-RCNN, our model does not incur any additional computation during the inference phase, resulting in no extra latency. The overall framework is shown in Figure 6.

## 2.5 Extension to semi-supervised learning

Semi-supervised learning aims to train object detectors with a large scale of image-level annotations and some box-level annotations (Liu et al., 2022a) (Li et al., 2022). Generally, there are mainly three steps in the semi-supervised learning paradigm. Firstly, training the model on limited labeled data; then, pseudo-labels are generated for the unlabeled data by using the output of the previous step; finally, optimizing the quality of the pseudo-labels to perform semi-supervised learning tasks. We argue that pseudo-

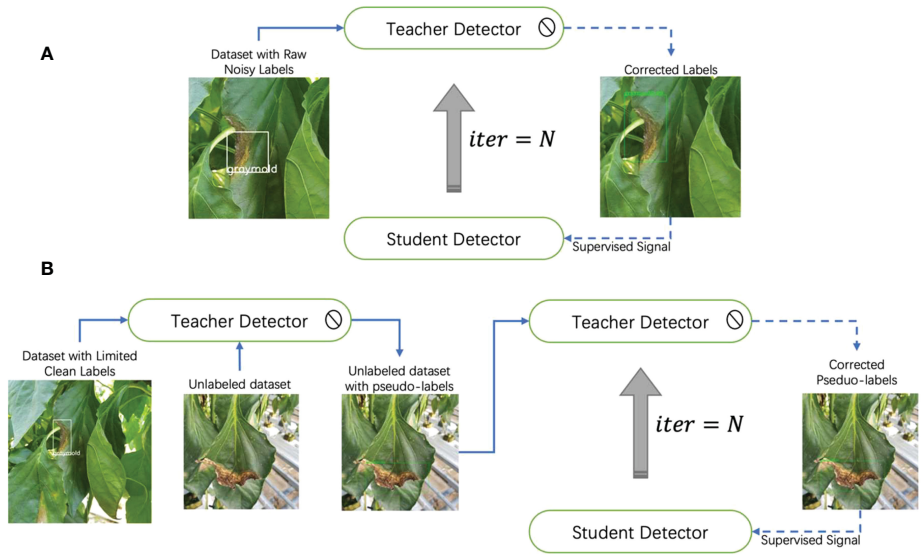



FIGURE 6
Iterative teacher-student learning framework. **(A)** The teacher detector corrects the noisy annotation. The corrected annotation is used as a supervised signal to train a student detector. Once the student detector is well-trained, it will become a new teacher. **(B)** The pipeline for extending our method to semi-supervised learning tasks. We use OA-MIL as teacher and student detector.

labels can be treated as inaccurate instance-level annotations. In other words, we transform the optimization problem of pseudo-labels into a label correction problem. Therefore, extending our method naturally to the semi-supervised learning paradigm is meaningful and feasible. However, the model's performance is limited by annotated data scale, thereby sufficient mislabeled or wrongly localized box predictions are selected as pseudo-ground truth, resulting in a sub-optimal solution of detection performance. As shown in Figure A.2, the model may generate predictions of two classes for one symptom.

We observed that without cleaning up redundant labels during the iterative process, it increased the number of false positive boxes in subsequent iterations. To overcome this issue, we implemented a post-processing step for the pseudo-labels using prior knowledge. We iterated through all the predicted pseudo-labels and set an intersection-over-union (IoU) threshold 0.3. For a pair of predicted bounding boxes with different categories, we retained only the one with the higher confidence score as the pseudo-label for the next iteration, discarding the lower-scoring one. While for a pair of predicted bounding boxes with the same category, we retained the big one. This post-processing step helped refine the pseudo-labels and improve the overall quality of the iterative process. The second row in Figure A.2 showcases the pseudo-labels retained after applying the label post-processing method.

Additionally, the model paid special attention to some samples' healthy, unknown, and background regions, which is exacerbated when the available labels are limited. As shown in Figure A.3, the model generated many false positive bounding boxes in these regions. To address this issue, we introduced additional control class labels such as background class, unknown class, and healthy label. These control class labels were added to improve the model's decision-making capability and mitigate the generation of false positive predictions.

During the data annotation process, we can initially annotate a subset (e.g. 1%) and then use our proposed method to infer and save pseudo-labels. The process of preserving pseudo-labels is commonly referred to as auto-labeling (Zhang et al., 2021). Through auto-labeling, researchers can obtain a large number of pseudo-labels. These pseudo-labels have relatively high quality and require minimal further data cleaning to obtain high-quality labels. Auto-labeling enables researchers to efficiently annotate data, accelerating the annotation process and facilitating the development of models with improved performance.

## 2.6 Evaluation metric

Intersection-over-Union metric (IoU): We utilized a threshold of 0.5 to capture true positive detections generated by the model, as:

$$IoU = \left| \frac{A \cap B}{A \cup B} \right| \tag{6}$$

where A and B represent the ground truth and predicted box, respectively.

Mean Average Precision score (mAP): mAP is the area under the precision-recall curve calculated for all classes. We utilize the standard mean average precision (mAP) metric with an intersection over union (IoU) threshold of 0.5 (mAP@50).

$$AP = \frac{1}{11} \sum_{r \in [0, 0.1, \ldots, 0.9, 1]} P(r) \tag{7}$$

$$\mathrm{P}(\mathbf{r}) = \max_{\tilde{r}:\tilde{r} \geq r} p(\tilde{r}) \tag{8}$$

where, $P(r)$ is the maximum precision for any recall values greater than r, and $p(\tilde{r})$ is the measured precision at recall $\tilde{r}$.

# 3 Experiments and results

## 3.1 Implementation details

### 3.1.1 Noise analysis

We utilized the LXML library to read clean annotation and noisy annotation files. For each label pair, we computed the intersection between each clean bounding box and all noisy bounding boxes under the same category by using IOU function. Then we selected the largest intersection as the matching box.

### 3.1.2 Detector

We benchmarked Faster-RCNN (Ren et al., 2015), a representative two-stage detector. All experiments are based on the popular open-source code libraries mmdetection (Chen et al., 2019) and OA-MIL (Liu et al., 2022a) with default settings. A distributed training method was adopted with a batch size of 8 per GPU 3090. The learning rate was set to 0.02. ResNet-50 (He et al., 2016) is used as a default backbone and initialized with weights pre-trained on ImageNet.

### 3.1.3 Annotation correction

The model was trained for 12 epochs in one iteration, and the learning rate was reduced by a factor of 0.1 at the 8th and 11th epochs. In the experiments regarding label correction, no offline data augmentation was employed. Instead, only horizontal image flipping was employed as online training data augmentation. We estimate the time complexity of the model using training duration. Over 4742 training images, our model requires only 1.42 hours to complete training, approximately 1.2 times faster than Faster-RCNN (1.17 hours). The overall time complexity depends on the number of iterations, and with the number of iterations set to 3, the training duration is approximately four times that of Faster-RCNN (including label correction and updates).

### 3.1.4 Semi-supervised learning

When 1% labeled data was available, the model was trained for 200 epochs, and the learning rate was decreased by 0.1 at the 150th and 180th epochs. When 10% labeled data was available, the model was trained for 100 epochs, and the learning rate was decreased by

0.1 at the 75th and 90th epochs. Horizontal image flipping, vertical flipping, 90-degree rotation, and random scaling were used as offline training data augmentation for labeled data. Following the completion of the initial training, the model was utilized to infer over all unannotated images within the training set. We used the LXML library to save pseudo-label files for automatic annotation. Subsequent iterations were conducted using the default settings for label correction. While this approach may require three times the training duration compared to Faster-RCNN, it offers significant cost savings regarding manual annotation.

### 3.1.5 Inference

Only the final student detector was needed in the inference stage, so there was no additional computational cost. This indicates that our model has the same inference speed as the baseline model Faster R-CNN. All configurations at this stage can be performed according to the default settings without additional modification. The IoU threshold for non-maximum suppression was set to 0.5, and the score threshold was 0.5 which is higher than the default setting of 0.05 in most works.

## 3.2 Annotation correction results

For a fair comparison, all models were evaluated on the well-annotated paprika disease test set in (Dong et al., 2022). The root mean square relative error $\gamma$was set to 0.1, 0.15, and 0.2, respectively, to generate datasets with synthesized noise and evaluate the method's effectiveness. The performance of Faster-RCNN in various settings was used as a benchmark. Table 2 shows the result on real-world noise datasets and synthesized noisy datasets. Note that we performed validation experiments on synthetic noise. The results presented in Table 2 are the mean of two experiments. Surprisingly, our method achieved a slight improvement even on clean datasets.

Our method can enhance the model's robustness to location noise. Under real-world noise, the performance improved from 57.1% to 83.1%. With synthetic noise at $\gamma = 0.15$ , our model improved from 58.5% to 82.1%, which is almost the same gain trend as real-world noise. Therefore, it is reasonable to describe the noise level by the root mean square relative error $\gamma$ . Compared with OA-MIL, our model improved by 5.3%. It indicated that inaccurate ground truth dominates strong object localization priors, thereby misleading the correction module in OA-MIL. Thus, optimizing a model by itself by only learning corrected labels cannot achieve optimal performance since corrected labels still contain noise. Our method used a teacher-student learning framework to correct the noisy labels and finally trained a more robust model. Extensive experiments verify the effectiveness of our method. To benchmark against state-of-the-art detectors, we provide the performance on YOLOv8-L (Jocher et al., 2023), comparable in parameter count to Faster R-CNN. Additionally, we compare our approach with the noise correction method NDet (Wang et al., 2022b) based on Faster R-CNN. The results indicate that these methods may perform better on clean datasets or datasets with lower noise levels. However, our approach demonstrates a clear advantage as the $\gamma$ increases.

As shown in Figure 7, the performance according to the number of training epochs and the number of iterations was compared using a clean and real-world noisy paprika dataset. Faster-RCNN was used as the baseline detector. Figure 7A shows the mAP comparison of different datasets used for training according to the number of training epochs. Performance increases as the epoch increases in all datasets. Using clean datasets for training shows the best performance, and using real noisy datasets shows the worst performance. The proposed method shows better performance of the two types of label correction methods. The graph for the proposed method with an iterative process shows the results in the training period of the student detector in the last iterative process. Figure 7B shows the change of $\gamma$ over epochs. Although noisy bounding boxes corrected by OA-MIL (Liu et al., 2022a) have a lower $\gamma$ than the noisy annotations, only relying on the annotations corrected to provide appropriate supervision still cannot achieve the optimal performance of the model. Figure 7C shows how $\gamma$ and mAP change with iterations of the proposed method. Our method can further correct noisy labels, thereby iteratively improving model performance. We can see that $\gamma$ decreases and mAP increases as the iteration proceeds to some extent, which shows that the proposed iteration method is effective.

Besides, Figure 7D shows the comparison of performance change rates of OA-MIL (Liu et al., 2022a) and ours over Faster-RCNN. Note that our method achieved more remarkable performance gain at higher noise levels, which undoubtedly alleviated the adverse effects of location noise, thereby narrowing the performance gap caused by noise. We can even conclude that if

TABLE 2 Performance comparison on the paprika disease test dataset. $\gamma$ denotes the synthesized noise level.

| Methods (mAP@50(%)) | Noise Type | | | | |
|---|---|---|---|---|---|
| | Clean ($\gamma = 0$) | Real-world ($\gamma \approx 0.15$) | Synthesized ($\gamma = 0.1$) | Synthesized ($\gamma = 0.15$) | Synthesized ($\gamma = 0.2$) |
| Faster-RCNN (Ren et al., 2015) | 90.2 | 57.1 | 75.6 | 58.5 | 48.2 |
| YOLO-v8-L (Jocher et al., 2023) | **91.0** | 67.2 | 79.6 | 66.8 | 53.9 |
| NDet (Wang et al., 2022b) | 90.6 | 78.7 | 81.4 | 78.2 | 73.8 |
| OA-MIL (Liu et al., 2022a) | 90.6 (+0.4) | 77.8 (+20.7) | 80.2 (+4.6) | 76.5 (+18.0) | 71.6 (+23.4) |
| Ours | 90.6 (+0.4) | **83.1 (+26.0)** | **84.3 (+8.7)** | **82.1 (+23.6)** | **78.4 (+30.2)** |

Parentheses indicate the performance improvement of our method compared to the Faster RCNN. The best performance is in boldface. We use mAP@50(%) as performance metric to evaluate our model.

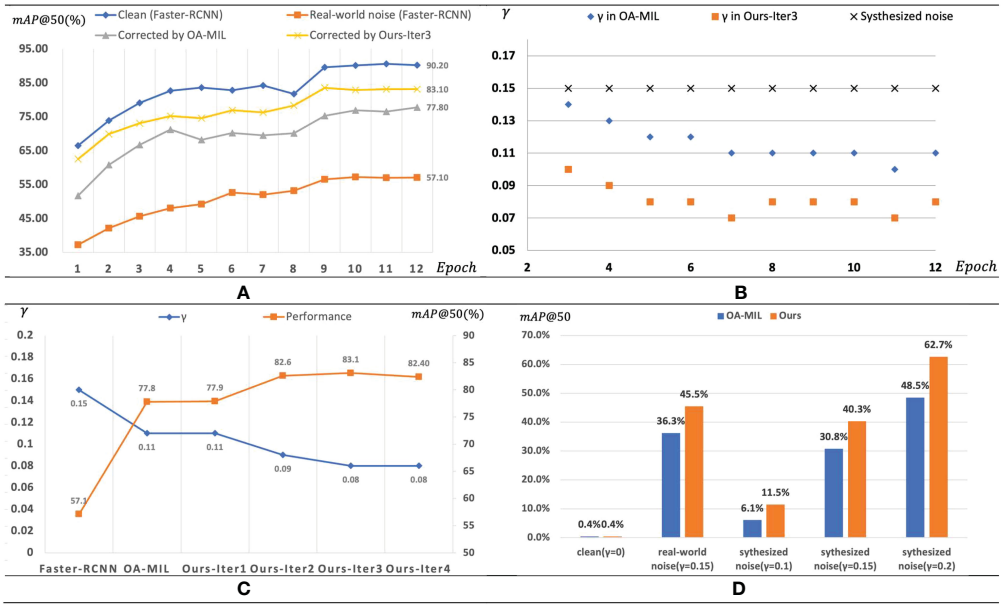



FIGURE 7
Training process. **(A)** mAP evaluated on paprika disease test set at each epoch for different training scenarios. **(B)** The root mean square relative error $\gamma$ of training samples in these training scenarios. We assume $\gamma$ equals 0 in the clean dataset. **(C)** Comparison of our method with OA-MIL. **(D)** Comparison of performance change rates of OA-MIL and Ours over Faster-RCNN.

the performance of Faster-RCNN differs greatly from our method, the dataset may contain severe noise interference.

## 3.3 Semi-supervised learning results

The pseudo-labels generated during semi-supervised learning also contain localization noise. Therefore, our method can be easily generalized to the semi-supervised learning paradigm naturally. We randomly chose 1%, 10%, and 50% of the clean training label set as available labeled data and the remaining data as unlabeled data. Table 3 shows the corresponding experimental results. For the paprika dataset, our method achieved approximately 75% of the performance of a fully supervised object detector when 1% of labels are available. In comparison, with 10% available labels, our method achieved 86% of the performance equivalent to fully supervised object detectors.

Furthermore, despite our post-processing techniques and adding control categories, we observed that classification errors persist in the pseudo-labels generated by this method. We attribute this to the limited data volume and multi-class classification problem. In other words, our method pays more attention to annotation localization noise rather than class noise. The GWHD2021 dataset only contains annotations for wheat heads and no other categories. For rigor, the GWHD2021 dataset was also used to validate the method's performance in a semi-supervised learning setting. Our method achieved 90.2% of the performance of a fully supervised object detector when 10% of the labels are available, reflecting our method's efficiency. Therefore, the annotation cost can be significantly reduced by deploying our method on semi-supervised learning tasks.

## 3.4 Ablation study

This section explores the impact of different components or design choices in our approach. Noise-corrected ablation experiments are based on Faster-RCNN with synthetic Gaussian noise ($\gamma = 0.15$ ). As mentioned in Section 2.4, relying only on the predictions produced by the detector itself as a source of supervision may not lead to optimal solutions. Therefore, we propose a teacher-student learning framework. Table 4 shows the number of iterative trainings versus model performance. We observed that too many iterations could lead to model performance degradation. Therefore, we iterated three times to stop for the final result. Unless otherwise specified, all experimental results in the paper are reported based on the results obtained after three iterations.

In semi-supervised learning tasks, we added control class labels, pseudo-label post-processing, and data augmentation to the noisy dataset to improve the discriminative power of the model. It is essential to adopt the post-label processing process for the iterative training of teachers and students in the later stage. Otherwise, the model will generate many overlapping or wrong labels and cause the task to fail. Table 5 shows the result of ablation experiments on semi-supervised learning. We have presented the qualitative results of model post-processing and class control in Figures A.2, A.3 in the appendix.

## 3.5 Visualization

The distributions of relative boundary coordinate errors for noisy annotations and our corrected ones are shown in Figure 2.

TABLE 3 The result of semi-supervised learning scenario setting on different datasets. The last row is the ratio of ours to Faster-RCNN*.

| | Paprika disease (mAP@50) (%) | | | GWHD2021 (mAP@50) (%) | | |
|---|---|---|---|---|---|---|
| Faster-RCNN* | 90.2 | | | 56.0 | | |
| Labeled data | 1% | 10% | 50% | 1% | 10% | 50% |
| Faster-RCNN | 54.6 | 69.9 | 75.2 | 33.6 | 46.0 | 48.2 |
| Ours | 67.5 | 77.6 | 82.9 | 42.1 | 50.5 | 53.4 |
| (Efficiency) | (74.8%) | (86.0%) | (91.9%) | (75.2%) | (90.2%) | (95.3%) |

* denotes the result of training Faster-RCNN on the full (100%) clean dataset.

The relative error of noisy annotations became smaller after being corrected by our method. Besides, we also visualize the experimental results in Appendix. In Figure A.1, each triplet contains input annotations, OA-MIL corrected results, and our method corrected results. The annotation correction results of our method cover the actual object more tightly than OA-MIL's. Figure A.2 demonstrates that post-processing of pseudo-label can remove low-confidence annotations and overlapping labels. Figure A.3 presents that the model reduces the misjudgment of suspicious regions after adding the control category. Figure A.4 displays the pseudo-labels generated by our method in the semi-supervised setting, where the last two rows show some failed predictions. The model is more prone to misclassify instances with similar symptoms, which may be due to the limited dataset. Even with misclassifications, the labels still closely match the actual objects, demonstrating the robustness of our method to locations. Figure A.5 shows the pseudo-labels generated by our method on the unlabeled dataset in the wheat head classification dataset GWHD2021. Our method can locate most of the wheat heads accurately.

# 4 Discussion

## 4.1 Does human cost equal intelligence?

In computer vision, there is an old saying highlighting the importance of labeling in deep learning methods: "As much human cost, there is as much machine intelligence." Labeled datasets have played a crucial role in the rapid development of deep learning. Over the years, researchers have proposed methods to reduce computation costs and improve intelligence, such as novel feature extractors, optimized loss functions, and efficient augmentation strategies. These methods have significantly contributed to the advancement of related industries.

Nevertheless, in specific domains, inaccuracies in annotations pose substantial hurdles for these methodologies. Our approach tackles this challenge by embracing a semi-supervised learning framework, which facilitates automatic annotation and fortifies the model against localization errors. This dual benefit notably slashes the costs associated with manual annotation. Figure 8 depicts the confusion matrix of the pseudo-labels for unlabeled samples when only 1% of the training set is labeled. The results show that backgrounds and some unseen instances are often misclassified as known classes, while some instances are overlooked due to limited diversity in the labeled data. Regarding automatic labeling, further corrections for false positives and negatives are required to achieve comparable performance to fully supervised learning methods. Manual annotation is unnecessary for well-learned instances as it would be a waste of resources.

## 4.2 Limitations and future work

In our previous studies (Dong et al., 2022), we established that localization noise exerts a more pronounced impact on model performance than class noise. Therefore, the core objective of this research was to address the challenge of correcting inaccurate

TABLE 4 The choice of the number of iterations for teacher-student learning trained on the paprika disease dataset. $\gamma$ denotes the synthesized noise level.

| Methods | Clean | Real-world noise | Synthesized noise | | |
|---|---|---|---|---|---|
| | $\gamma$ / 0 | $\gamma \approx 0.15$ | $\gamma$ / 0.1 | $\gamma$ / 0.15 | $\gamma$ / 0.2 |
| Faster-RCNN | 90.2 | 57.1 | 75.6 | 58.5 | 48.2 |
| Ours(iter-1) | 90.6 | 77.9 | 79.2 | 76.5 | 71.6 |
| Ours(iter-2) | – | 82.6 | 84.1 | 81.6 | 78.5 |
| Ours(iter-3) | – | **83.1** | **84.3** | **82.1** | **78.4** |
| Ours(iter-4) | – | 82.4 | 83.5 | 81.6 | 77.9 |
| Ours(iter-5) | – | 82.7 | 83.2 | 81.5 | 77.5 |

We used mAP50 to validate the performance at different iteration times. The best performance is in boldface.

TABLE 5 Ablation studies in a semi-supervised learning setting.

| Faster-RCNN | Data Augmentation* | Post-process | Control Class | Teacher-student | Paprika Disease mAP@50(%) | |
|---|---|---|---|---|---|---|
| | | | | | 1% | 10% |
| √ | √ | | | | 54.6 | 69.9 |
| √ | √ | | | iter-1 | 55.8 | 61.0 |
| √ | √ | √ | | iter-1 | 56.1 | 61.5 |
| √ | √ | √ | √ | iter-1 | 57.3 | 64.2 |
| √ | √ | √ | √ | iter-2 | 66.9 | **78.4** |
| √ | √ | √ | √ | iter-3 | **67.5** | 77.6 |
| √ | √ | √ | √ | iter-4 | 67.4 | 77.9 |
| √ | √ | √ | √ | iter-5 | 67.1 | 77.4 |

The models are trained on the paprika disease clean dataset. The best performance is in boldface. The * denotes only performing offline data augmentation on the labeled set.

bounding box annotations caused by localization noise. It is important to note that while our emphasis is on localization noise, it does not suggest the absence of class noise within real-world label sets. In practice, both types of noise can coexist, posing significant hurdles in training accurate models for object detection tasks. More research should focus on the patterns of real-world noise, based on which more effective methods can be proposed and further improve the detector performance. However, quantifying class noise is very difficult due to the diversity of datasets. Meanwhile, adding class noise in a random perturbation way does not match the actual distribution of class noise. Therefore, before the methodology is proposed, how to construct a class noise dataset is a question worth considering.

Recently, large-scale vision-language models (Zhang et al., 2022; Kim et al., 2023) have been applied to localization tasks. These models can locate objects in images using only textual labels, without the need for explicit training. This process is known as zero-shot inference (Cao et al., 2023). In the semi-supervised setting of this paper, we require pre-labeling a portion of the data to train the teacher model for automatic annotation. In contrast, it may be possible to achieve fully automatic annotation by correcting the zero-shot inference results of these large-scale vision-language models. Therefore, correcting the zero-shot inference results of these models is an interesting research topic that we plan to explore in future work.

## 5 Conclusion

This paper investigated the impact of location noise on detector performance in real-world environments. We observed that relative location noise in real scenarios follows Gaussian distribution and is dependent on object size, which guides how to synthesize location noise. Furthermore, we proposed an annotation correction method

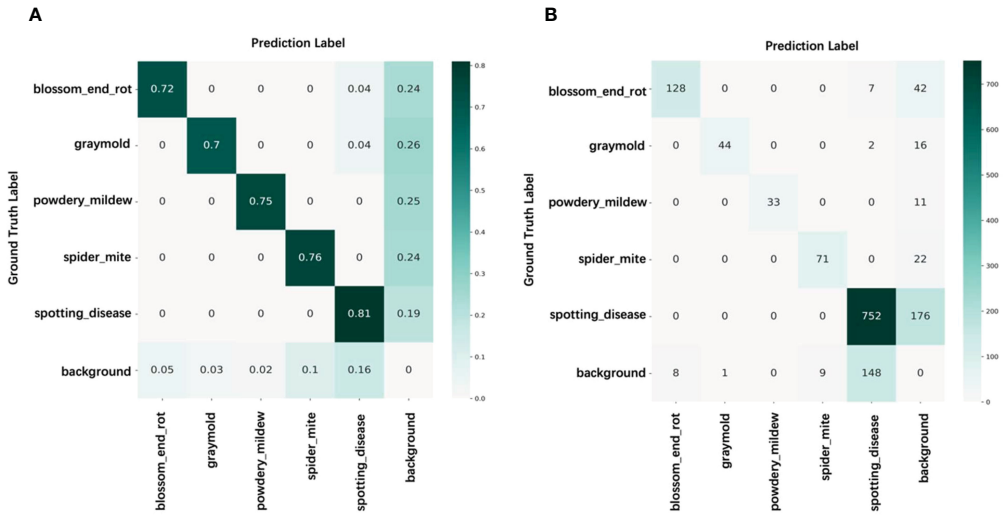



FIGURE 8
Confusion matrix of pseudo-label on paprika disease training set with 1% labeled data. **(A)** Percentage count results. **(B)** Count results.

based on the teacher-student learning paradigm, which significantly narrows the performance gap caused by noise. Utilizing our method is crucial for performance improvement if the labeling budget is limited or constrained. Our method also supports the correction of imprecise pseudo-labels generated in a semi-supervised learning task, implying that our method can be extended to semi-supervised learning tasks. In summary, our method is suitable for handling datasets with low-quality annotations, thus reducing the annotation cost and improving traditional labeling methods.

## Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

## Author contributions

JD designed the method, performed the experiments, and wrote the manuscript. AF and DP provided support in the data collection, plant disease knowledge, and proofreading article. SY and HK advised in the design of the system and analyzed the annotation strategies to find the best method for efficient plant disease detection. All authors contributed to the article and approved the submitted version.

## Funding

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (No. 2019R1A6A1A09031717); by the Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry(IPET) and Korea Smart Farm R&D Foundation(KosFarm) through Smart Farm Innovation Technology Development Program, funded by Ministry of Agriculture, Food and Rural Affairs(MAFRA) and Ministry of Science and ICT(MSIT), Rural Development Administration (RDA)(1545027569); and by the National Research Foundation of Korea(NRF) grant funded by the Korea government (MSIT). (NRF-2021R1A2C1012174).

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2023.1238722/full#supplementary-material

SUPPLEMENTARY FIGURE 1
Comparison of our method with OA-MIL results. The first row displays labels with location noise. The second and third rows show the correction results using OAMIL and our method (iter-3). The labels in the third row are more closely aligned with the actual disease locations.

SUPPLEMENTARY FIGURE 2
Example results of post-processing the pseudo-labels. We use red color to highlight false positive bounding boxes. Better displayed on the screen.

SUPPLEMENTARY FIGURE 3
The model reduces the number of false positive labels for suspicious regions by adding control classes. When the data volume is low, the model is prone to classifying backgrounds, unknown regions, and healthy areas as diseases. The red color highlights false positive bounding boxes. Better displayed on the screen.

SUPPLEMENTARY FIGURE 4
Pseudo-labels generated in a semi-supervised learning setting when only 1% of the label set is available on the paprika disease dataset. The last two lines present some failure cases, including classification failures, localization failures, and missed detections.

SUPPLEMENTARY FIGURE 5
Pseudo-labels (after three iterations) generated in a semi-supervised learning setting when only 1% of the label set is available on the GWHD dataset.

## References

Bhise, D., Kumar, S., and Mohapatra, H. (2022) in *2022 6th International Conference On Electronics, Communication And Aerospace Technology*, Tamil Nadu, India. 1106–1111 (New York: IEEE).

Cao, W., Wu, Y., Sun, Y., Zhang, H., Ren, J., Gu, D., et al. (2023). A review on multimodal zero-shot learning. *Wiley Interdiscip. Reviews: Data Min. And Knowledge Discovery* 13, E1488. doi: 10.1002/Widm.1488

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., et al. (2019). *Mmdetection: open mmlab detection toolbox and benchmark* (Arxiv Preprint Arxiv). *1906.07155*. doi: 10.48550/Arxiv.1906.07155

David, E., Madec, S., Sadeghi-Tehran, P., Aasen, H., Zheng, B., Liu, S., et al. (2020). Global wheat head detection (Gwhd) dataset: A large and diverse dataset of high-resolution rgb-labelled images to develop and benchmark wheat head detection methods. *Plant Phenomics.* 3521852. doi: 10.34133/2020/3521852

David, E., Serouart, M., Smith, D., Madec, S., Velumani, K., Liu, S., et al. (2021). Global wheat head detection 2021: an improved dataset for benchmarking wheat head detection methods. *Plant Phenomics* 2021, 9846158. doi: 10.34133/2021/984615

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009) in *2009 Ieee Conference On Computer Vision And Pattern Recognition*, Miami, Fl, Usa. 248–255 (New York: IEEE).

Dong, J., Fuentes, A., Yoon, S., Kim, H., Jeong, Y., and Park, D. S. (2023). A new deep learning-based dynamic paradigm towards open-world plant disease detection. *Front. In Plant Sci.* 14. doi: 10.3389/Fpls.2023.1243822

Dong, J., Lee, J., Fuentes, A., Xu, M., Yoon, S., Lee, M. H., et al. (2022). Data-centric annotation analysis for plant disease detection: strategy, consistency, and performance. *Front. In Plant Sci.* 13. doi: 10.3389/Fpls.2022.1037655

Dubel, R., Wijata, A. M., and Nalepa, J. (2023) in *International Conference On Computational Science*, Prague, Czech Republic. 111–119 (New York: Springer).

Evangelisti, E., Turner, C., Mcdowell, A., Shenhav, L., Yunusov, T., Gavrin, A., et al. (2021). Deep learning-based quantification of arbuscular mycorrhizal fungi in plant roots. *New Phytol.* 232, 2207–2219. doi: 10.1111/Nph.17697

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (Voc) challenge. *Int. J. Of Comput. Vision* 88, 303–338. doi: 10.1007/S11263-009-0275-4

Fuentes, A., Yoon, S., Kim, T., and Park, D. S. (2021). Open set self and across domain adaptation for tomato disease recognition with deep learning techniques. *Front. In Plant Sci.* 12. doi: 10.3389/Fpls.2021.758027

Fuentes, A. F., Yoon, S., Lee, J., and Park, D. S. (2018). High-performance deep neural network-based tomato plant diseases and pests diagnosis system with refinement filter bank. *Front. In Plant Sci.* 9. doi: 10.3389/Fpls.2018.01162

Ganesh, R. S., Sivakumar, S., Dinesh, R., and Harish, M. (2022) in *2022 3rd International Conference On Smart Electronics And Communication (Icosec)*, Trichy, India. 1187–1192 (New York: IEEE).

Gou, J., Yu, B., Maybank, S. J., and Tao, D. (2021). Knowledge distillation: A survey. *Int. J. Of Comput. Vision* 129, 1789–1819. doi: 10.1007/S11263-021-01453-Z

Gudovskiy, D., Rigazio, L., Ishizaka, S., Kozuka, K., and Tsukizawa, S. (2021) in *Proceedings Of The Ieee/Cvf Conference On Computer Vision And Pattern Recognition*, Nashville, United States. 16601–16610. (New York: IEEE)

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022) in *Proceedings Of The Ieee/Cvf Conference On Computer Vision And Pattern Recognition*, New Orleans, Louisiana, Usa. 16000–16009. (New York: IEEE)

He, S., and Krainer, K. M. C. (2020). Pandemics of people and plants: which is the greater threat to food security? *Mol. Plant* 13, 933–934. doi: 10.1016/J.Molp.2020.06.007

He, K., Zhang, X., Ren, S., and Sun, J. (2016) in *Proceedings Of The Ieee Conference On Computer Vision And Pattern Recognition*, Las Vegas, Nevada. 770–778. (New York: IEEE)

Hu, C., Li, X., Liu, D., Chen, X., Wang, J., and Liu, X. (2022). *Teacher-student architecture for knowledge learning: A survey* (Arxiv Preprint Arxiv). *2210.17332*.doi: 10.48550/Arxiv.2210.17332

Huang, Z., Zhang, J., and Shan, H. (2023) in *Proceedings Of The Ieee/Cvf Conference On Computer Vision And Pattern Recognition*, Vancouver, Canada. 11661–11670. (New York: IEEE)

Jiang, Z., Zhou, K., Liu, Z., Li, L., Chen, R., Choi, S.-H., et al. (2021) in *International Conference On Learning Representations*.

Jocher, G., Chaurasia, A., and Qiu, J. (2023) *Yolo by ultralytics*Available at: https://github.com/ultralytics/ultralytics

Khakimov, A., Salakhutdinov, I., Omolikov, A., and Utaganov, S. (2022) in *Iop Conference Series: Earth And Environmental Science*. 012002 (Bristol, England: IOP Publishing).

Kim, D., Park, J., Lee, J., Park, S., and Sohn, K. (2023) in *Proceedings Of The Ieee/Cvf Winter Conference On Applications Of Computer Vision*, Waikoloa, Hawaii. 2539–2548. (New York: IEEE)

Li, H., Wu, Z., Shrivastava, A., and Davis, L. S. (2022) in *Proceedings Of The Aaai Conference On Artificial Intelligence*, Arlington, Virginia. 1314–1322. (Stanford California: AAAI)

Li, J., Xiong, C., Socher, R., and Hoi, S. (2020). *Towards noise-resistant object detection with noisy annotations* (Arxiv Preprint Arxiv). doi: 10.48550/Arxiv.2003.01285

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014) in *European Conference On Computer Vision*, Zurich, Switzerland. 740–755 (Springer, New York).

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., et al. (2022c) in *Proceedings Of The Ieee/Cvf Conference On Computer Vision And Pattern Recognition*, New Orleans, Louisiana, Usa. 12009–12019. (IEEE: New York)

Liu, C., Wang, K., Lu, H., Cao, Z., and Zhang, Z. (2022a) in *European Conference On Computer Vision*, Tel Aviv, Israel. 53–69 (New York: Springer).

Liu, X., Yoo, C., Xing, F., Oh, H., El Fakhri, G., Kang, J.-W., et al. (2022b). Deep unsupervised domain adaptation: A review of recent advances and perspectives. *Apsipa Trans. On Signal And Inf. Process.* 11. doi: 10.1561/116.00000192

Mao, J., Yu, Q., and Aizawa, K. (2021). Noisy localization annotation refinement for object detection. *Ieice Trans. On Inf. And Syst.* 104, 1478–1485. doi: 10.1587/Transinf.2021edp7026

Nalepa, J., Kotowski, K., Machura, B., Adamski, S., Bozek, O., Eksner, B., et al. (2023). Deep learning automates bidimensional and volumetric tumor burden measurement from mri in pre-and post-operative glioblastoma patients. *Comput. In Biol. And Med.* 154, 106603. doi: 10.1016/J.Compbiomed.2023.106603

Nazki, H., Yoon, S., Fuentes, A., and Park, D. S. (2020). Unsupervised image translation using adversarial networks for improved plant disease recognition. *Comput. And Electron. In Agric.* 168, 105117. doi: 10.1016/J.Compag.2019.10511

Pison, G. (2022). World population: 8 billion today, how many tomorrow? *Population Societies* 604, 1–4. doi: 10.3917/Popsoc.604.0001

Priyadharshini, G., and Dolly, D. R. J. (2023) in *2023 9th International Conference On Advanced Computing And Communication Systems (Icaccs)*, Tamilnadu, India. 1540–1545 (New York: IEEE).

Rahman, A., Zunair, H., Reme, T. R., Rahman, M. S., and Mahdy, M. R. C. (2021). A comparative analysis of deep learning architectures on high variation malaria parasite classification dataset. *Tissue And Cell* 69, 101473. doi: 10.1016/J.Tice.2020.101473

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-cnn: towards real-time object detection with region proposal networks. *Adv. In Neural Inf. Process. Syst.* 39 (06), 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Shoaib, M., Shah, B., Ei-Sappagh, S., Ali, A., Ullah, A., Alenezi, F., et al. (2023). An advanced deep learning models-based plant disease detection: A review of recent research. *Front. In Plant Sci.* 14. doi: 10.3389/Fpls.2023.1158933

Smith, A. G., Han, E., Petersen, J., Olsen, N. A. F., Giese, C., Athmann, M., et al. (2022). Rootpainter: deep learning segmentation of biological images with corrective annotation. *New Phytol.* 236, 774–791. doi: 10.1111/Nph.18387

Song, H., Kim, M., Park, D., Shin, Y., and Lee, J.-G. (2022). Learning from noisy labels with deep neural networks: A survey. *IEEE Trans. On Neural Networks And Learn. Systems*.doi: 10.1109/Tnnls.2022.3152527

Tomaszewski, M., Nalepa, J., Moliszewska, E., Ruszczak, B., and Smykała, K. (2023). Early detection of solanum lycopersicum diseases from temporally-aggregated hyperspectral measurements using machine learning. *Sci. Rep.* 13, 7671. doi: 10.1038/S41598-023-34079-X

Wang, S., Gao, J., Li, B., and Hu, W. (2022b). Narrowing the gap: improved detector training with noisy location annotations. *IEEE Trans. On Image Process.* 31, 6369–6380. doi: 10.1109/Tip.2022.3211468

Wang, P., Meng, F., Donaldson, P., Horan, S., Panchy, N. L., Vischulis, E., et al. (2022a). High-throughput measurement of plant fitness traits with an object detection method using faster R-cnn. *New Phytol.* 234, 1521–1533. doi: 10.1111/Nph.18056

Xia, X., Liu, T., Han, B., Gong, M., Yu, J., Niu, G., et al. (2021). *Sample selection with uncertainty of losses for learning with noisy labels* (Arxiv Preprint Arxiv). *2106.00445*. doi: 10.48550/Arxiv.2106.00445

Xu, M., Bai, Y., Ghanem, B., Liu, B., Gao, Y., Guo, N., et al. (2019). Missing labels in object detection. *Cvpr Workshops* 5. Long Beach, Ca, Usa.

Xu, H., Blonder, B., Jodra, M., Malhi, Y., and Fricker, M. (2021a). Automated and accurate segmentation of leaf venation networks *via* deep learning. *New Phytol.* 229, 631–648. doi: 10.1111/Nph.16923

Xu, M., Kim, H., Yang, J., Fuentes, A., Meng, Y., Yoon, S., et al. (2023). Embrace Limited and Imperfect Training Datasets: Opportunities and Challenges in Plant Disease Recognition Using Deep Learning. *Front. Plant Sci.* 14, 1225409. doi: 10.3389/fpls.2023.1225409

Xu, M., Yoon, S., Fuentes, A., Yang, J., and Park, D. S. (2021b). Style-consistent image translation: A novel data augmentation paradigm to improve plant disease recognition. *Front. In Plant Sci.* 12. doi: 10.3389/Fpls.2021.773142

Xu, Y., Zhu, L., Yang, Y., and Wu, F. (2021c). Training robust object detectors from noisy category labels and imprecise bounding boxes. *IEEE Trans. On Image Process.* 30, 5782–5792. doi: 10.1109/Tip.2021.3085208

Yao, Y., Yu, W., Gao, Y., Dong, J., Xiao, Q., Huang, B., et al. (2022). W-transformer: accurate cobb angles estimation by using A transformer-based hybrid structure. *Med. Phys.* 49, 3246–3262. doi: 10.1002/Mp.15561

Zhang, S., Jafari, O., and Nagarkar, P. (2021). *A survey on machine learning techniques for auto labeling of video, audio, and text data* (Arxiv Preprint Arxiv). *2109.03784*. doi: 10.48550/Arxiv.2109.03784

Zhang, Y., Lin, F., Mi, S., and Bian, Y. (2023). Self-label correction for image classification with noisy labels. *Pattern Anal. And Appl.*, 1–10. doi: 10.1007/s10044-023-01180-w

Zhang, H., Zhang, P., Hu, X., Chen, Y.-C., Li, L., Dai, X., et al. (2022). Glipv2: unifying localization and vision-language understanding. *Adv. In Neural Inf. Process. Syst.* 35, 36067–36080. (New Orleans, LA, USA).

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 16 October 2023
DOI 10.3389/fpls.2023.1258658

Check for updates

OPEN ACCESS

EDITED BY
Tonghai Liu,
Tianjin Agricultural University, China

REVIEWED BY
M. F. Mridha,
American International University, Bangladesh
Xiujun Zhang,
Chinese Academy of Sciences (CAS), China

*CORRESPONDENCE
Wenping Jiang
jiangwenping@sit.edu.cn

RECEIVED 14 July 2023
ACCEPTED 18 September 2023
PUBLISHED 16 October 2023

CITATION
Liang J and Jiang W (2023) A ResNet50-DPA model for tomato leaf disease identification.
*Front. Plant Sci.* 14:1258658.
doi: 10.3389/fpls.2023.1258658

COPYRIGHT
© 2023 Liang and Jiang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# A ResNet50-DPA model for tomato leaf disease identification

Jin Liang and Wenping Jiang*

School of Electrical and Electronic Engineering, Shanghai Institute of Technology, Shanghai, China

Tomato leaf disease identification is difficult owing to the variety of diseases and complex causes, for which the method based on the convolutional neural network is effective. While it is challenging to capture key features or tends to lose a large number of features when extracting image features by applying this method, resulting in low accuracy of disease identification. Therefore, the ResNet50-DPA model is proposed to identify tomato leaf diseases in the paper. Firstly, an improved ResNet50 is included in the model, which replaces the first layer of convolution in the basic ResNet50 model with the cascaded atrous convolution, facilitating to obtaining of leaf features with different scales. Secondly, in the model, a dual-path attention (DPA) mechanism is proposed to search for key features, where the stochastic pooling is employed to eliminate the influence of non-maximum values, and two convolutions with one dimension are introduced to replace the MLP layer for effectively reducing the damage to leaf information. In addition, to quickly and accurately identify the type of leaf disease, the DPA module is incorporated into the residual module of the improved ResNet50 to obtain an enhanced tomato leaf feature map, which helps to reduce economic losses. Finally, the visualization results of Grad-CAM are presented to show that the ResNet50-DPA model proposed can identify diseases more accurately and improve the interpretability of the model, meeting the need for precise identification of tomato leaf diseases.

KEYWORDS

**tomato leaf image, disease identification, deep learning, convolutional neural network, feature extraction**

## 1 Introduction

In 2021, China's tomato production reached 67.63 million tons, accounting for 35% of the world's total output, which is a very popular garden plant (Bhatkar et al., 2021; Jing et al., 2023; Liu et al., 2023). However, plant disease resistance needs to be faced by all crops in large-scale production, and various diseases can damage crops in any given growing season, resulting in reduced yield or lower quality. For example, early blight, a fungal disease that is a common type of disease in field-grown tomatoes, causes round or elongated brown lesions to form on infected plant tissue (Paul et al., 2019; Hu and Zhang, 2023). Late blight causes infected leaves to turn pale green to brown and eventually to wither and die (Arafa et al., 2022). Therefore, it is very important to quickly and

accurately identify tomato leaf disease types and provide timely targeted treatment. The traditional research on disease recognition is to preprocess crop disease pictures first, and then use manual recognition methods to classify diseases. However, this method is highly dependent on professional knowledge and the cost of expert diagnosis is high (Manavalan, 2021; Li et al., 2022; Thakur et al., 2022). Meanwhile, it is time-consuming, labor-intensive and highly subjective, which is prone to misclassification. Therefore, a method that can accurately identify tomato leaf diseases is urgently needed.

With the rapid development of computer vision and artificial intelligence, the study on crop disease recognition based on image data has received extensive attention (Al-Wesabi et al., 2022), and the recognition accuracy has also been greatly improved. According to the extraction method of disease features in images, it can be divided into method based on machine learning and method based on deep learning (Saleem et al., 2019; Vishnoi et al., 2020). In terms of machine learning, Qin et al. (2017) used the K-median clustering algorithm and the segmentation method of linear discriminant analysis to obtain the image of the diseased area, and then extracted texture for detecting alfalfa leaf disease by SVM model. Finally, the identification accuracy rate is 80%. Sghair et al (Sghair and Naisbitt, 2017). segmented leaf disease areas by filtering and thresholding color features, and then detected rice leaf disease types, with an accuracy rate of 96.6%. However, the leaf texture and shape characteristics are not discussed, resulting in the loss of leaf information. Yao et al. (2009) used the GLCM algorithm to extract the characteristics of rice leaf area, perimeter, contrast, and leaf length and width. Then the SVM algorithm is employed for classification and recognition, with an accuracy rate of 97.2%. While, disease types with similar textures are prone to misclassification. Wang et al. (2012) studied the grape and wheat diseases identification, where the K-means clustering algorithm is applied to segment the disease area, and the backpropagation network are designed as the classifier. The results showed that the accuracy of grape and wheat disease recognition is 97.14% and 100%, respectively. However, the features extracted in (Wang et al., 2012) have been combined many times to achieve the best recognition effect. How to find a suitable feature combination set is still a great challenge. Compared with manual recognition, the recognition accuracy of the machine learning-based method has been significantly improved, and this method is more interpretable for the model. However, when the amount of data is large, the performance of machine learning-based method is not satisfactory. Meanwhile, manual design on features is inevitable, which requires the empirical knowledge of domain experts and relies heavily on feature engineering. the above problems can be effectively solved by the deep learning-based methods with higher computational performance, especially on large-scale datasets. This method uses an optimized loss function to learn rules, avoiding manual design of rules, which is highly portable, and can be compatible with multiple platforms. Therefore, many scholars apply it to the identification of leaf diseases, taking advantage of its real-time, automatic feature learning and high accuracy, in order to take control measures in time to protect the growth of crops. For example, Elhassouny et al (Elhassouny and Smarandache, 2019). applied the MobileNet neural network to identify tomato leaf diseases, and proposed an embedded intelligent application for the algorithm, which greatly reduced the required computing resources, and the accuracy rate reached 88.4%. However, the algorithm has a small amount of data and only uses more than 7,000 images for recognition, thus, how to avoid overfitting is a potential problem. Liu et al. (2020) first used the Inception structure to enhance feature extraction performance, and then introduced a dense connection strategy to enhance feature propagation, and obtained a new CNN model DICNN. Compared with the traditional neural network, the recognition accuracy of the DICNN model has increased by 2.97%, which provides a new idea for the identification of grape leaf diseases. Zhao et al. (2021) introduced the attention mechanism, SENet, into ResNet50 to identify tomato leaf disease. Then the average recognition accuracy rate reached 96.81%, which strengthened the ability to extract information. However, the SENet module only focused on the channel attention mechanism, which caused the loss of image texture information during the transfer process. Liu et al. (2022) applied the ROI feature extraction algorithm to the DenseNet classification model, which showed that the ROI algorithm can highlight the lesion area of rice leaves and improve the recognition accuracy. Finally, the accuracy rate of rice leaf disease recognition reached 96% in (Liu et al., 2022). Ahmad et al. (2021) used the gray-level co-occurrence matrix to extract the features of plant diseased leaves, and strengthened the extraction of image texture information. The research results showed that the classification accuracy rate reached 98.79%. Faisal et al. (2023) proposed a plant leaf disease classification algorithm DFNet that combines dual pre-training models (MobileNetV2, NASNetMobile), whose recognition accuracy for corn leaf disease and coffee leaf disease was 97.53% and 94.65%, respectively. Zhou et al. (2021) proposed the RRDN model for the disease identification of tomato leaves, which combined the deep residual network and the dense network to improve the calculation accuracy and reduce the number of parameters. The results show that the average identification accuracy rate reached 95%. Zhao et al. (2022) introduced the residual structure as a convolutional block into the Inception network, which alleviated the problem of gradient disappearance. Simultaneously (Zhao et al., 2022), embedded an attention mechanism in the model to enhance the feature extraction performance of plant leaves, and experimental results show that the algorithm has higher recognition accuracy and fewer model parameters.

To sum up, the convolutional neural network is conducive to the accurate identification of plant leaf diseases, liberates the labor force, and realizes the development of smart agriculture (Chin et al., 2023). Moreover, many scholars have improved the recognition accuracy by strengthening the model's feature extraction performance. However, the effect still needs to be further improved. In this regard, this paper studies the tomato leaf disease recognition problem, where the novel ResNet50-DPA model is proposed, which effectively enhances the ability to extract image features and improves the recognition accuracy. The main contributions of the paper are summarized as follows:

(1) In ResNet50-DPA model, the improved ResNet50 is obtained by using cascaded atrous convolution instead of

the first layer of convolutions in the ResNet50 residual network, for capturing tomato leaf characteristics with different scales.

(2) In this model, a novel DPA mechanism is proposed, which contains a channel attention module and a spatial attention module where the stochastic pooling is innovatively introduced. Meanwhile, two convolutions with one dimension are employed to replace the MLP layer in DPA. This mechanism helps to extract information of important areas in the image, and effectively reduces the damage to the leaf information.

(3) The enhanced tomato leaf feature map is obtained by inserting the above DPA into the residual module of improved ResNet50, which facilitates obtaining more comprehensive information, and then quickly and accurately identifying tomato leaf disease types and reducing crop losses.

The rest of this paper is as follows: Section 2 introduces the data set used and describes the proposed model in detail. In Section 3, grouping experiments are carried out for the proposed model, the experimental results are analyzed, and the model is visualized. Section 4 summarizes and explains the future work.

# 2 Methodology

## 2.1 Data preprocessing

The quality of images has a significant impact on the image processing, feature extraction, and ultimately tomato leaf disease recognition. Plant Village, an open data set on the Internet, used in this paper, which a total of 14 plant varieties, each of which contained different leaf disease types. Among them, tomato leaf disease images accounted for 26% of the total data set, with a total of 10 categories, including 1 healthy category and 9 diseased category images, which was much larger than the data of other plant species. Therefore, tomato leaf disease images were selected as the research object in this paper. Figure 1 shows all categories of tomato leaf disease sample images included in this dataset.

In deep learning, increasing the amount of data used by the training neural network has an important impact on the final recognition effect. The model trained after enriching the data set has strong generalization ability and can ensure the diversity of the data to avoid overfitting during the training process. Therefore, this paper adopts imgaug image enhancement library to expand the data set. The emboss enhancement outlines the outline of the image, highlights the changing part of the image, and fade the same gray level part to reflect the depth. Image equalization in image enhancement is denoising by using the residual information prevalent in natural images. In this paper, the specific operation is to make the pixel values originally distributed in a concentrated way evenly distributed to all possible ranges, so that the contrast and brightness of the image can be improved and the details of the tomato leaf image can be better observed (Tang and Isa, 2017). In Figure S1, is the histogram before image equalization processing, and the height in the figure represents the total frequency of this pixel value. It can be seen that most of the pixel values before processing are concentrated in a region, and Figure S1 is the histogram after image equalization processing, where the frequency of the pixel value 70 decreases significantly, while the frequency of the other pixel values increases, and is evenly distributed everywhere. Figure 2 below is the effect of the randomly selected tomato leaf sample image enhanced by the above data.

After image enhancement and equalization processing, the total data reached 22,930 images. 20% of the total data set was divided into test sets, and the detailed values of the final data set were shown in Table 1. A total of 10 tomato leaf disease categories were included in the dataset, including 18,345 images in the training set and 4585 images in the test set.

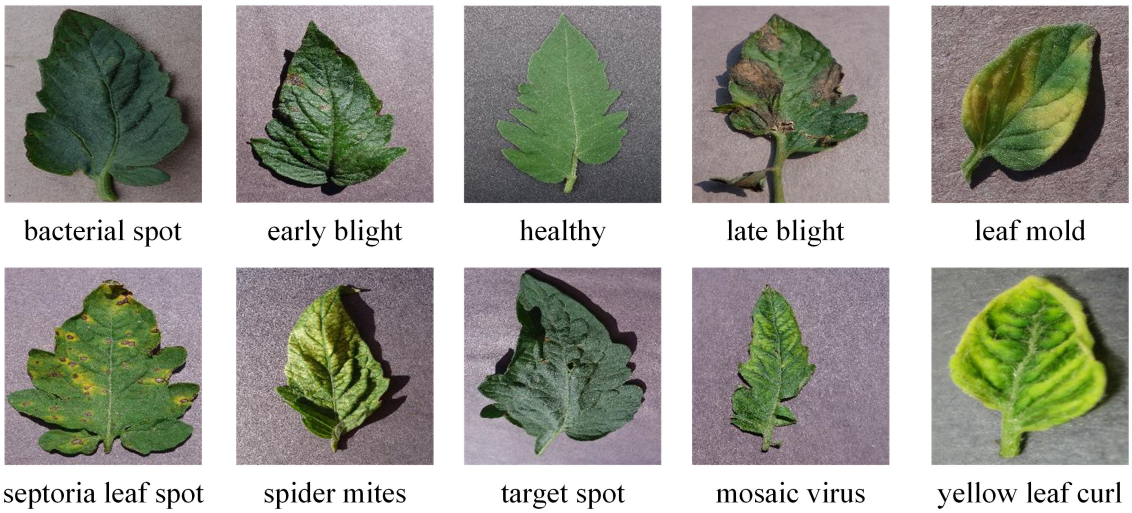



FIGURE 1
Sample image of tomato leaf diseases.

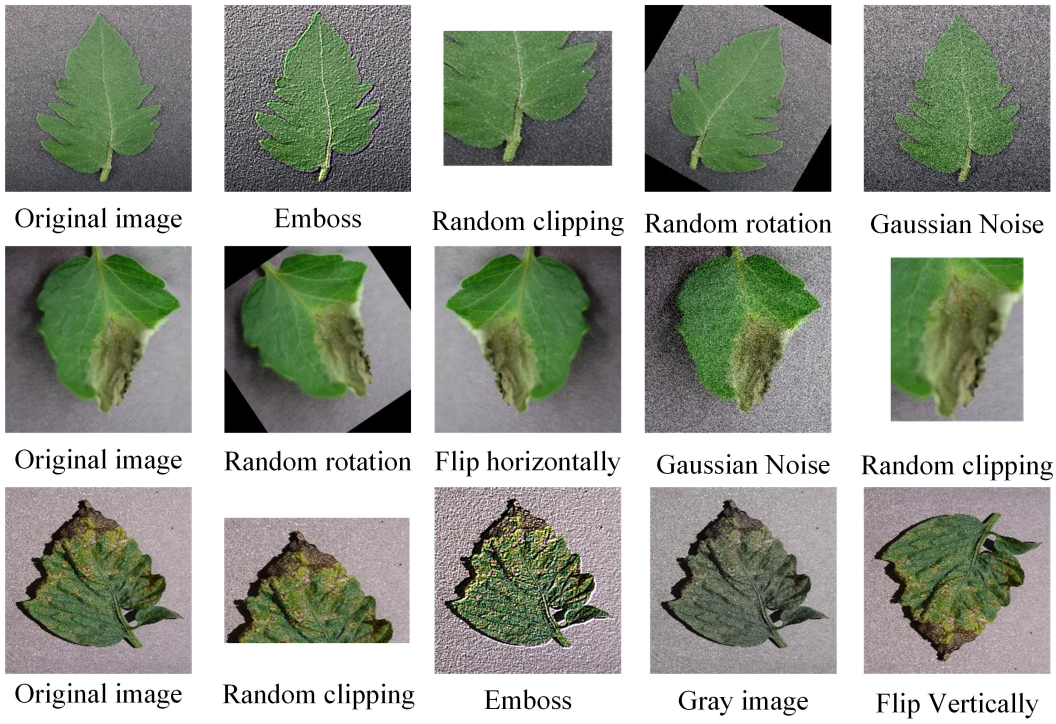



FIGURE 2
Sample image enhancement example.

## 2.2 ResNet50-DPA model

Problem statement:

For leaf disease recognition, most of the existing studies improve the recognition accuracy by deepening the number of network layers or adjusting parameters. However, by observing the data set, we found that there were only slight differences between different types of tomato leaf diseases, so strengthening the feature extraction capability is our focus to improve the recognition performance. To this end, firstly, On the one hand, the atrous cavity convolution is added to the input layer of the model to obtain information of different scales; Then, on the other hand, the dual-path attention mechanism is introduced into the residual module of the residual network to make the extracted information more comprehensive.

TABLE 1 Detailed information on tomato leaf disease data set.

| Tomato leaf category | Total | Train | Test |
|---|---|---|---|
| bacterial spots | 2127 | 1702 | 425 |
| early blight | 2400 | 1920 | 480 |
| healthy | 2407 | 1926 | 481 |
| late blight | 2314 | 1851 | 463 |
| leaf mold | 2352 | 1882 | 470 |
| septoria leaf spot | 2181 | 1745 | 436 |
| spider mites | 2176 | 1741 | 435 |
| target spot | 2284 | 1827 | 457 |
| mosaic virus | 2238 | 1790 | 448 |
| yellow leaf curl | 2451 | 1961 | 490 |

Model introduction:

The ResNet50-DPA model proposed in the paper consists of two parts: DPA and improved ResNet50. The dual-path attention mechanism is divided into two parts: channel attention module and spatial attention module, where the stochastic pooling is innovatively introduced. The addition of spatial attention module can effectively reduce the problem of image texture information loss caused by using only channel attention module. Different from the existing studies, the stochastic pool method is introduced in the two attention modules of channel and space, which can effectively eliminate the influence of non-maximum value in the feature graph. At the same time, the MLP layer is replaced by two one-dimensional convolution layers, which reduces the computation and avoids the destruction of tomato leaf information. In addition, based on the ResNet50, the model proposed in this paper innovatively replaces the first layer of 7×7 convolution in the original model with a cascade of atrous convolution, which can help us obtain tomato leaf features at different scales, facilitate the extraction of more comprehensive information, and make the final recognition more accurate. The following sections expand on DPA and the improved ResNet50.

### 2.2.1 The DPA mechanism

Attention mechanism is a technique used to enhance the model's attention to different parts of the input image, which improves the performance and generalization ability of the model

by giving different weights to each area of the image. SENet, published in 2018, is a representative work that applies attention mechanism to channel dimension. Based on the SENet module, this paper proposes a novel attention mechanism, the DPA mechanism. Different from the SENet module, which only uses global average pooling, resulting in the loss of image texture information in the process of passing spatial information to the channel. On this basis, this paper introduces maximum pooling and stochastic pooling in DPA. The introduction of maximum pooling effectively reduces the problem of information loss (Chen et al., 2021), while stochastic pooling can eliminate the influence of non-maximum values and improve the model's ability to perceive details, thus improving the model's performance. Stochastic pooling uses the feature graph values in each channel divided by their sum to get a probability matrix. Random sampling and selection are performed according to the size of the probability values. The sampling results are positively correlated with the size of the values, and the probability of being selected is also greater if the element value is larger, instead of always taking the maximum value as in maximum pooling. The specific calculation process of stochastic pooling is as follows:

Firstly, the element values in each square in the pooled area are divided by the sum of the element values in all squares to obtain the probability values of each element. See Equation 1, where k represents the index of the element values.

$$P = (p_i) = \frac{X_i}{\sum_{k \in R_j} X_k} \tag{1}$$

Then, position l is selected in the pooling region according to probability P, as shown in Equation 2.

$$A_j = X_l, l \sim P(p_1, \ldots, p_{|R_j|}) \tag{2}$$

Finally, the final value is obtained according to the position comparison. Figure 3 below shows the results obtained by using the three pooling operations respectively. It can be seen that the average and maximum pooling results are 1.11 and 3. The stochastic pooling first generates the probability graph, and then determines the position l. The final output result is 2, and the non-maximum value is selected.

The structure diagram of the channel attention module in DPA is shown in Figure 4 below. The addition of stochastic pool is conducive to eliminating non-maximum values, and at the same time, the model has stronger generalization ability and better regularization effect. Then, two one-dimensional convolution layers are used to replace the MLP layer. Since the one-dimensional convolution does not involve dimensionality reduction operation, damage to tomato leaf feature information can be avoided. Finally, the three spatial information descriptors generated are added and activated by the sigmoid to obtain the final channel attention module, as shown in Equation 3.

$$M_c(F) = \sigma\left(conv(Max(F)) + conv(Sto(F)) + conv(Avg(F))\right) \tag{3}$$

The channel attention module in DPA mainly focuses on the relationship between channels of features, while the spatial attention module pays more attention to which part of the graph is more influential in the whole feature graph (Cao and Yuan, 2022). Similar to the channel attention mechanism, the spatial attention module of DPA also uses stochastic pooling operations, as shown in Figure 5. The main idea is to carry out global average pooling, global maximum pooling and stochastic pooling along channel dimensions, and then splicing the obtained results, using the 7×7 convolution layer to reduce dimension, and then obtaining the attention feature map of the spatial domain through sigmoid activation function. See the following Equation 4 for detailed description.

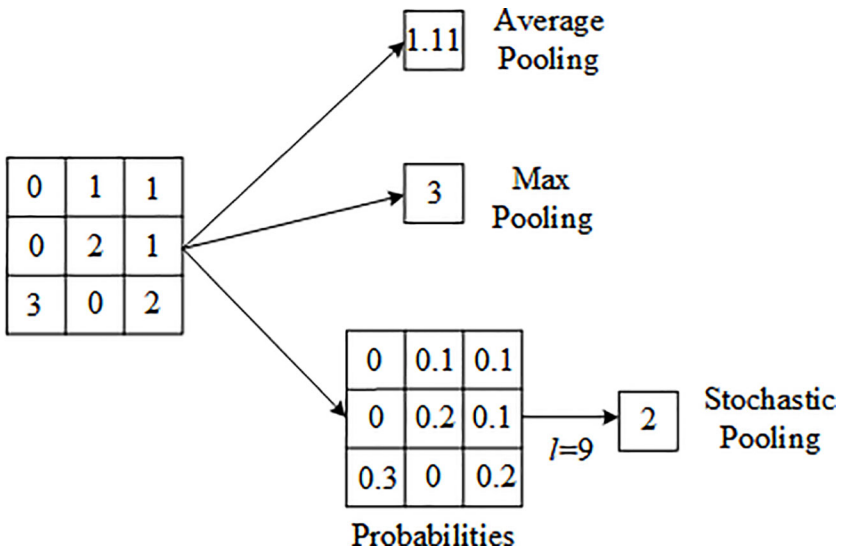



FIGURE 3
Examples of different pooling operations.

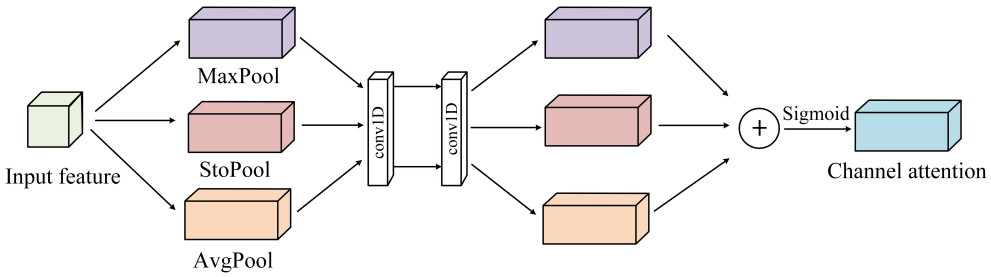



FIGURE 4
Channel attention module in DPA.

$$M_s(F) = \sigma(f^{7\times7}([Avg(F); Max(F); Sto(F)])) = \sigma\left(f^{7\times7}([F^s_{avg}; F^s_{max}; F^s_{sto}])\right) \quad (4)$$

In the above two formulas, $\sigma$ represents sigmoid operation. For the specific formula, see Equation 5. Avg represents global average pooling, Max represents global maximum pooling, Sto represents stochastic pooling, and f represents convolution operation.

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (5)$$

### 2.2.2 Improved ResNet50 residual network

With the continuous development and progress of deep learning theory and computer hardware, more and more applications have been made in various fields of our lives, such as face recognition, language recognition, automatic driving technology, etc. (Kute et al., 2019), but existing problems have also slowly emerged. In deep learning, the more convolutional layers, the better the classification effect will be. Therefore, in order to obtain deeper features, the number of convolutional layers is constantly deepening. At the beginning, LeNet network only has 7 convolutional layers, AlexNet has 8 layers, and the later VggNet network contains 19 layers. With the passage of time, the depth exploration of neural network is constantly carried out, but when it reaches a certain degree, the accuracy rate decreases, and the recognition accuracy of neural network is gradually saturated, resulting in the problem of gradient disappearance. Subsequently, the residual network ResNet was proposed by He et al. By constantly stacking residual structures, the problem of gradient disappearance during model training was effectively alleviated (He et al., 2016). In this paper, ResNet50 was selected as the basic model in consideration of the experimental effect and calculation amount, and its network structure was shown in Table S1.

The structure diagram of the residual block is shown in Figure S2, where X represents the input feature matrix. After two 3×3 convolution operations, the output is F(x), which is called the residual function, and the final output F(x)+X is passed into the next residual module. However, in ResNet50, the residual function F(x) is changed to three-layer convolution, as shown in (B) in Figure S2. A 3×3 convolution is retained, and a 1×1 convolution layer is added before and after the convolution layer to achieve the functions of reduction and dimension increase so that the expression features of the network are better and the corresponding detection or classification performance is stronger when the number of layers is increased. In addition, 1×1 convolution is used in the residual network, which reduces the number of parameters in the network and also reduces the amount of computation to some extent.

In order to avoid the loss of spatial information when extracting feature graphs in the residual network model, this paper chooses atrous convolution replace the first layer 7×7 convolution in the ResNet50 network, which can enlarge the receptive field without increasing the number of parameters. Figure 6 shows a schematic diagram of the atrous convolution with different expansion rates. In figure, the atrous convolution with the expansion rate of 1, 3×3 convolution kernel size has the same effect as the ordinary 3×3 convolution. For the atrous convolution with an expansion rate of 2,

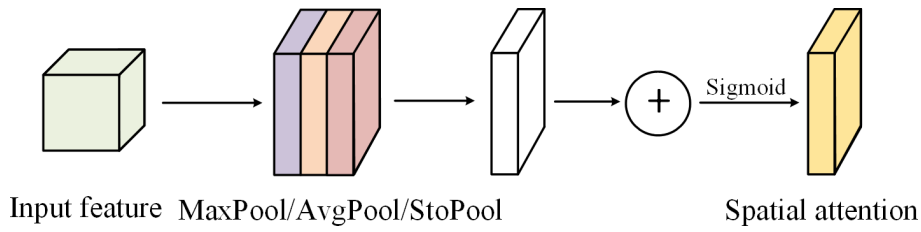



FIGURE 5
Spatial attention module in DPA.

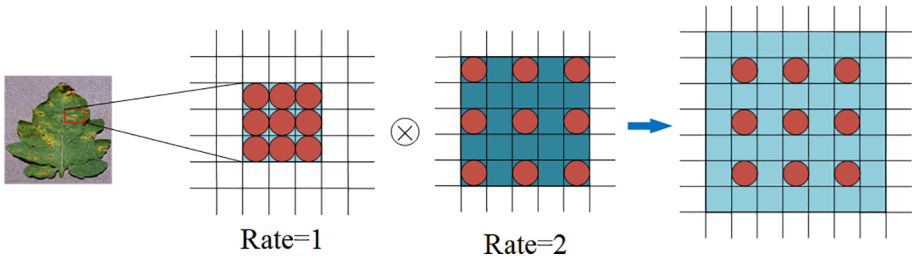



FIGURE 6
Extraction process of atrous convolution feature with different expansion rates.

although the size of the convolution kernel does not change, the receptive field of the convolution has increased to 5×5, as shown in the dark blue part in Figure 6. The effect of ordinary 7×7 convolution nuclei can be achieved by concatenating these two 3×3 convolution nuclei with different expansion rates, as shown in the light blue part in Figure 6. Moreover, due to the different expansion rates of the two convolution nuclei, the sensitivity fields are also different, that is, multi-scale information is obtained, which helps to observe tomato leaf characteristics at different scales and extract more comprehensive information. The identification of subsequent tomato leaf diseases is also more accurate.

Finally, this paper adds the aforementioned DPA mechanism to the improved ResNet50 residual block structure, which can enlarge the weight of effective channels in the element layer, reduce the influence of redundant features with the help of DPA mechanism. It is worth mentioning that the 7×7 convolution in the first layer is replaced with a cascaded atrous convolution, which can extract tomato leaf features at different scales. Meanwhile, in this paper, the maximum pooling layer is removed and the soft pooling is chosen, which helps to retain more details in the original image. The final ResNet50-DPA structure diagram is shown in Figure 7.

## 2.3 Overall design for tomato leaf disease image recognition

Figure 8 below is the overall design diagram of this paper. At the input layer of the model, the imgaug image enhancement library is first used to expand the data set. Due to the problem of random clipping, the data set at this time is RGB three-channel tomato leaf disease images with different sizes, and their sizes were uniformly adjusted to 224×224 as the model input. After atrous convolution and soft pooling operations, the DPA module was introduced into the residual module of the residual network to obtain the enhanced tomato leaf feature map. At last, after the normalization layer, the 1×1 global average pooling layer and the fully connected layer, the probability values of tomato leaves corresponding to different disease categories were calculated by the Softmax function, and the category of the maximum value is output as the classification result. Then, the classification and recognition of the final tomato leaf disease image is complete.

# 3 Experiment and data analysis

## 3.1 Experimental platform

The experiment is based on Windows 11 operating system, and the simulation software uses jupyter notebook compiler based on Python3.8 programming under Anaconda. The deep learning framework adopts Tensorflow architecture, which can support the model deployment on multiple GPUs, and the processor uses an i7 processor. RTX3050 graphics card set up the model proposed in this paper.

## 3.2 Evaluation criteria

In this paper, the accuracy rate, precision rate, recall rate and F1 score are as indicators to evaluate the performance of the proposed ResNet50-DPA model. In the following expressions, true positive (TP) represents the number of correctly classified positive samples, false positive (FP) represents the number of incorrectly classified negative samples, true negative (TN) represents the number of correctly classified negative samples, and false negative (FN) represents the number of incorrectly classified positive samples (Wang et al., 2017).

Accuracy: Accuracy is defined as the proportion of the number of correctly classified samples to the total number of samples, as shown in Equation 6.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\,\% \tag{6}$$

Precision: Precision is defined as the proportion of the number of positive samples with correct classification to the number of positive samples determined by the classifier, as shown in Equation 7.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

Recall: Recall rate is defined as the proportion of the number of correctly classified positive samples to the number of real positive samples, as shown in Equation 8.

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

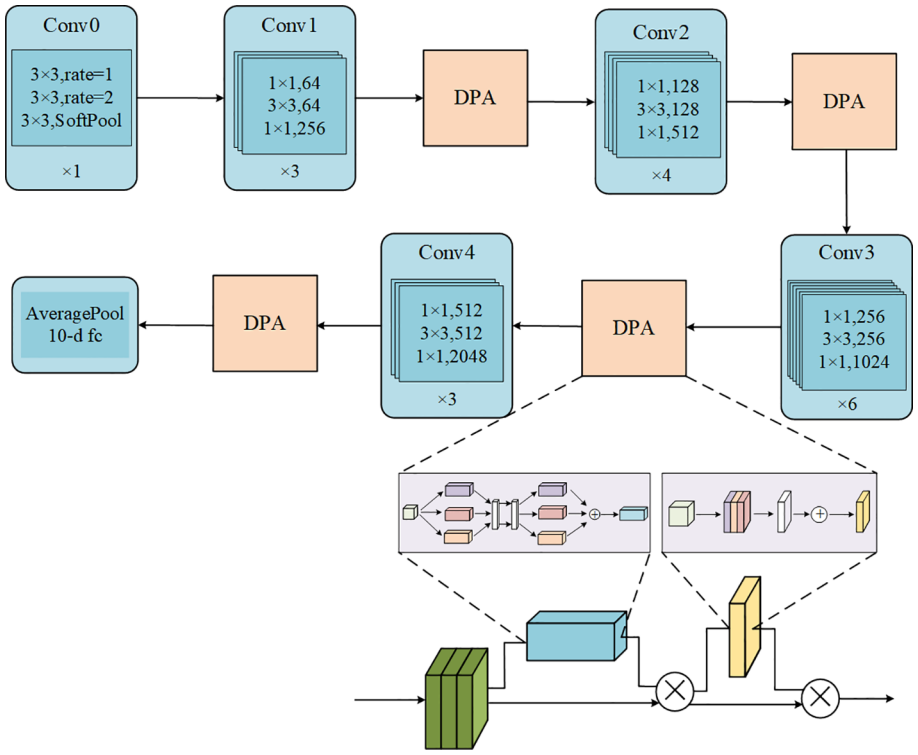



FIGURE 7
ResNet50-DPA network structure.

F1 value: refers to the harmonic average of precision rate and recall rate, see Equation 9.

$$F1 = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \tag{9}$$

## 3.3 Comparative analysis of hyperparameters

Hyperparameters are variables that define the network structure. In order to further improve the classification

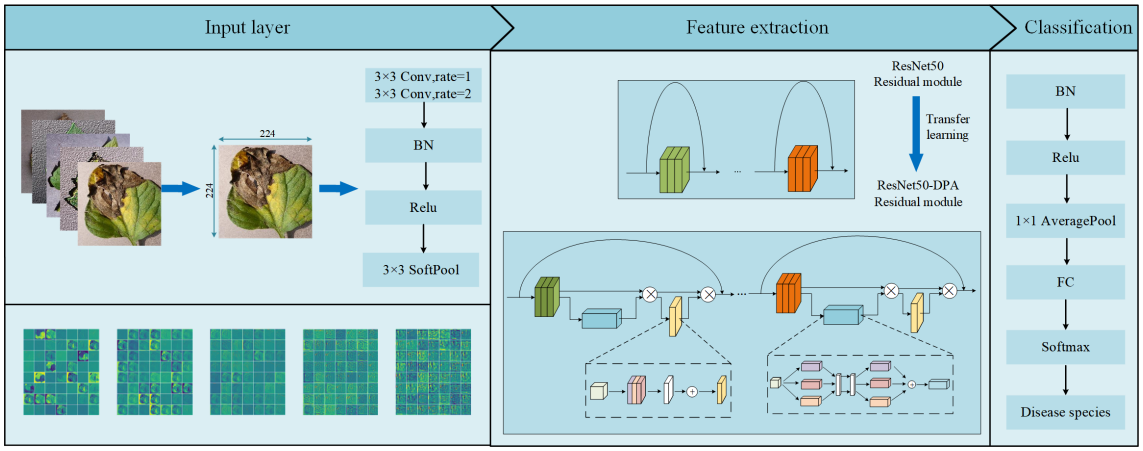



FIGURE 8
General design drawing.

performance of the proposed model, it is necessary to discuss the values of some necessary parameters, such as batch size, loss function, and optimizer selection. In this section, we test within 50 epochs and employ the early stop technique. When the accuracy did not continuously improve within 5 epochs, the experiment was stopped to observe the effect of different parameter values on the classification effect. Meanwhile, three experiments are conducted to take the mean value of each parameter as the result. The values of the final major parameters are shown in Table 2. The following is a discussion and analysis of the specific parameters.

### 3.3.1 Batch size

Training batch size refers to the number of samples taken from the training set during each iteration training. The larger the value, the faster the model is trained, but too large a value makes it take up more memory. However, too small batch will lengthen the training time, and seriously may cause gradient shock, making the model convergence slow or unable to converge. Multiple literatures related to image classification are reviewed, and experimental results are observed with the common batch size. The accuracy of leaf disease recognition obtained with different values of batch size was shown in Figure 9A. With the increase of Batch size, the classification accuracy increased slowly, but the accuracy decreases as the value continued to increase. The classification effect is best when the value is 32.

### 3.3.2 Loss function

The loss function is a function used to measure the gap between the real value and the predicted value, which is used to guide the model optimization direction and improve the prediction accuracy. When different loss functions are used, the model performance is shown in Figure 9B. In this paper, the cross-entropy loss function is finally selected, and compared with other loss functions, the cross-entropy loss function has better performance.

$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i -[y_i log(p_i) + (1 - y_i) log(1 - p_i)] \quad (10)$$

### 3.3.3 Learning rate and optimizer

The learning rate can be used to control the updating speed of the weight of the neural network, usually with the order of 10, to search for the best learning rate on a logarithmic scale. However, it is impractical to continuously adjust the learning rate manually, so this paper uses typical learning rate values to test the classification performance, that is, from $10^{-2}$ to $10^{-5}$ (0.01, 0.001, 0.0001, 0.00001), At the same time with a number of different optimizer (SGD/Adam/Adagrad/Adadelta/RMSprop) are combined, and different combination of tomato leaf disease recognition results are shown in Table S1 and Figure 9C, and finally set the model the optimizer to high calculation efficiency, low memory requirements of Adam optimizer, vector for 0.001.

TABLE 2 Parameter value.

| Parameter | Value |
|---|---|
| Batch size | 32 |
| Image size | 224×224 |
| Loss function | Cross-entropy |
| Optimization algorithm | Adam |
| Learning rate | 0.001 |
| Number of epochs | 200 |

## 3.4 Model performance comparison

In order to verify the recognition performance of the ResNet50-DPA model for tomato leaf diseases proposed in this paper, three comparison experiments are presented as follows.

Comparison experiment 1:

The first set of experiments is trained with different deep learning models under the same data set and experimental environment. The compared models include AlexNet, MobileNetV2 and DenseNet121. The change curve of accuracy of each model in the training process is shown in Figure 10.

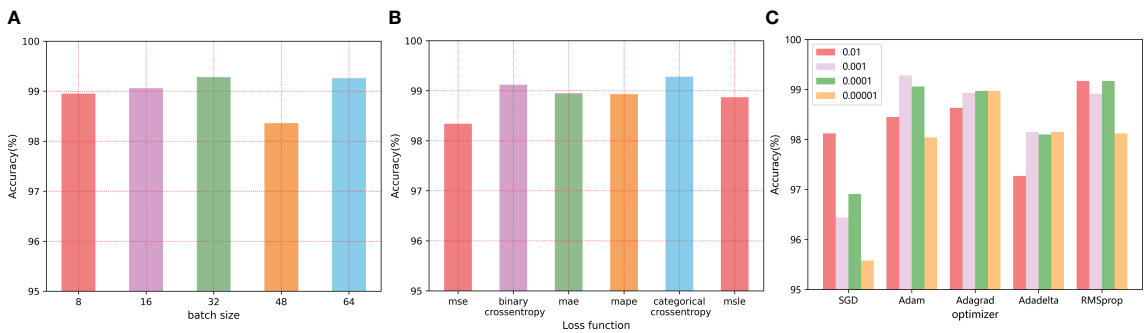



FIGURE 9
The influence of different parameter values on the final recognition effect: **(A)** The batch size **(B)** Loss function value **(C)** Optimizer selection.

It can be seen from the curve trend in the Figure 10 that with the increase of iterations, the prediction accuracy of each model has improved, among which AlexNet model has the slowest convergence speed and ResNet50-DPA model has the fastest convergence speed. Detailed experimental results are shown in Table 3. By observing the data in the table, we can see that the recognition accuracy of AlexNet, MobileNetV2 and DenseNet121 network models are 74.15%, 95.00% and 97.97%, respectively, while the recognition accuracy of the model proposed in this paper is 99.28%, which is significantly better than the previous three models.

Comparison experiment 2:

The second group of comparison experiments takes ResNet series residual network as the benchmark to observe whether the improved model has improved. Figure 11 below shows the change curve of accuracy of the ResNet series residual network, and it can be seen that the proposed model still reaches convergence fastest. In Table 4, the recognition accuracy of the ResNet series residual network is 96.97%, 97.43% and 97.60% respectively, and ResNet101 has the best recognition effect, while the accuracy of ResNet50-DPA model reaches 99.28%, its classification performance is the best among all models, and its accuracy is 99.29%. The value of recall rate and F1 are both 99.28%, which is also better than other models, indicating that the DPA enhances the model's ability to obtain information and can dig deeper features.

Comparison experiment 3:

The third group of comparison experiments is compared with tomato leaf disease recognition algorithms proposed by different scholars in recent years, and the results are shown in Table 5. According to the Table 5, when the amount of data is relatively small, the classification algorithm proposed by Trivedi et al. has an

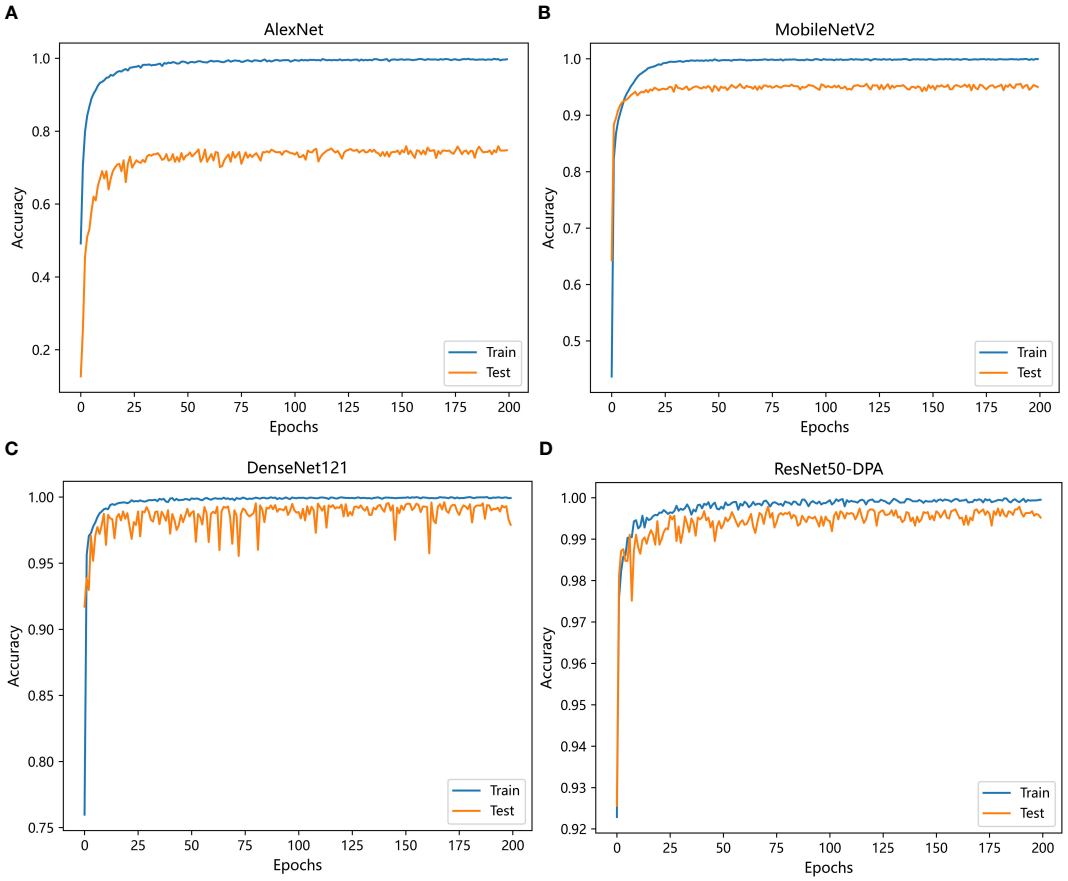



FIGURE 10
Variation curve of accuracy: **(A)** AlexNet **(B)** MobileNetV2 **(C)** DenseNet121 **(D)** ResNet50-DPA.

TABLE 3 Comparison of disease recognition performance of different network models.

| Model | Accuracy/% | Precision/% | Recall/% | F1/% |
|---|---|---|---|---|
| AlexNet | 74.15 | 80.19 | 74.15 | 74.54 |
| MobileNetV2 | 95.00 | 95.07 | 95.01 | 95.02 |
| DenseNet121 | 97.97 | 98.03 | 97.97 | 97.97 |
| ResNet50-DPA | 99.28 | 99.29 | 99.28 | 99.28 |

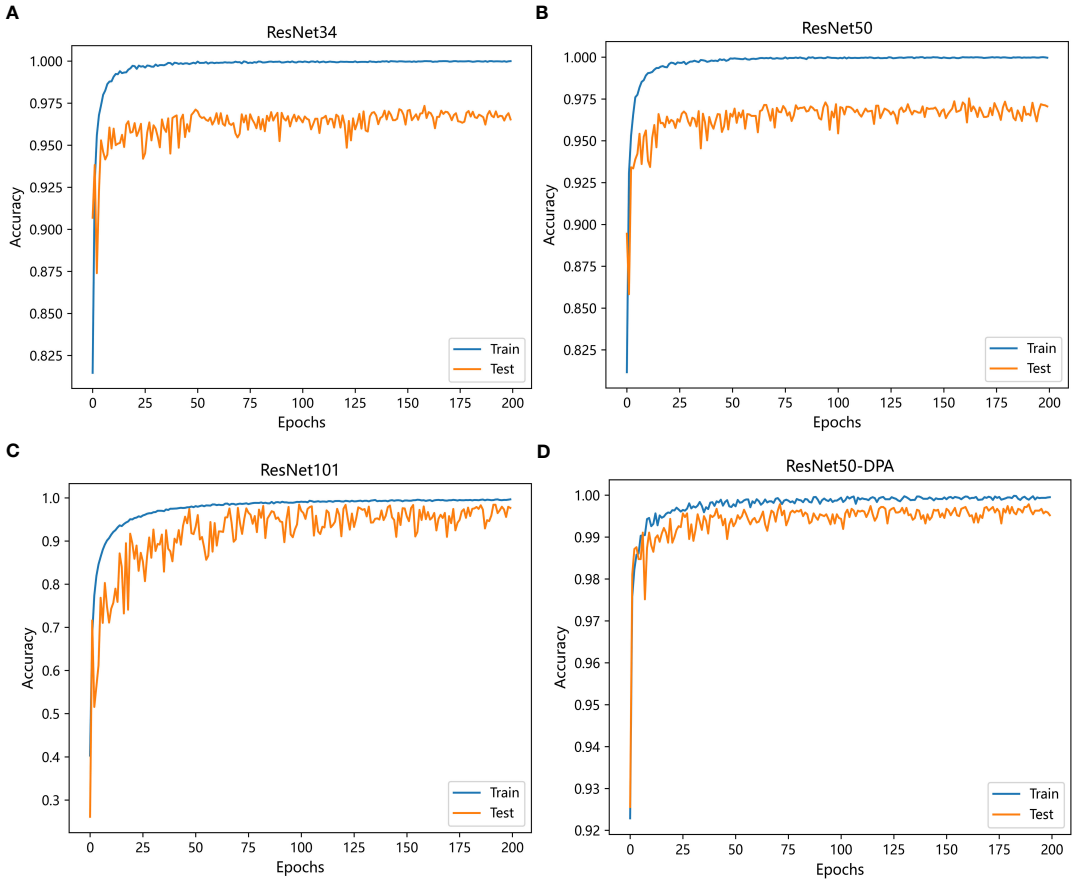



FIGURE 11
Change curve of ResNet series residual network accuracy: **(A)** ResNet34 **(B)** ResNet50 **(C)** ResNet101 **(D)** ResNet50-DPA.

TABLE 4 Comparison of disease recognition performance of ResNet series residual networks.

| Model | Accuracy/% | Precision/% | Recall/% | F1/% |
|---|---|---|---|---|
| ResNet34 | 96.97 | 96.98 | 96.97 | 96.96 |
| ResNet50 | 97.43 | 97.43 | 97.43 | 97.42 |
| ResNet101 | 97.60 | 97.69 | 97.60 | 97.60 |
| ResNet50-DPA | 99.28 | 99.29 | 99.28 | 99.28 |

accuracy of 98.49%. However, how to avoid overfitting is a potential problem. When the amount of data is similar, the model proposed in this paper has a higher accuracy due to obtaining multi-scale information by using cascades of atrous convolution when inputting images. Meanwhile, stochastic pooling is introduced into the dual path attention mechanism and one-dimensional convolution is used instead of MLP layer to reduce information loss, which is conducive to achieving better recognition accuracy.

Except for the accuracy, this paper also verifies the ability of the proposed model to identify tomato leaf diseases from different perspectives. The classification results observed using the confusion matrix are shown in Figure 12. The horizontal and vertical coordinates in the figure represent the 10 types of tomato leaf diseases in the data set, and the elements on the main diagonal represent the number of correctly classified tomato leaf disease samples. It can be seen that the ResNet50-DPA model has only a few sample graph classification errors, which can achieve a good recognition effect for different disease types, while the number of classification errors of other models is much larger than that of the model proposed in this paper.

In addition, the detailed values of the classification results of the proposed model for different types of tomato Leaf diseases are presented in Table 6. It can be seen from the data in the table that the identification accuracy of the model for Leaf mold and Mosaic virus

TABLE 5 Compared with the existing methods of tomato leaf disease identification.

| References | Year | Dataset | Images | Methods | Accuracy(%) |
|---|---|---|---|---|---|
| Zhao et al. (Vishnoi et al., 2020) | 2021 | Tomato leaf disease | 18160 | ResNet50+SeNet | 96.81 |
| Zhou et al. (Yao et al., 2009) | 2021 | Tomato leaf disease | 13185 | RRDN | 95 |
| Zhao et al. (Wang et al., 2012) | 2022 | Tomato leaf disease | 18160 | Inception + CBAM | 95.2 |
| Moussafir et al. (2022) | 2022 | Tomato leaf disease | 14526 | ResNet50 + EfficientNet | 98.01 |
| Anandhakrishnan et al. (Anandhakrishnan and Jaisakthi, 2022) | 2022 | Tomato leaf disease | 18160 | CNN | 98.40 |
| Theodora et al. (2023) | 2023 | Tomato leaf disease | 18160 | VGGNet | 99.23 |
| Rangarajan et al. (2018) | 2018 | Tomato leaf disease | 13262 | VGG16 | 97.29 |
| Anandaet et al. (Ananda and Vandana, 2022) | 2020 | Tomato leaf disease | 14421 | VGG16 | 95.71 |
| Ujawe et al. (Ujawe and Nirkhi, 2023) | 2023 | Tomato leaf disease | 13262 | AlexNet | 97.49 |
| Proposed | 2023 | Tomato leaf disease | 22930 | ResNet50 + DPA | 99.28 |

leaf diseases reaches 100%, while the accuracy of Leaf spot is only 97.09%. The model wrongly identified it as the type of spider mites, and the identification accuracy of other disease types is above 99%. The results shows that the research method proposed in this paper could achieves better recognition effect for different disease types, and could complete the accurate identification of tomato leaf diseases.

## 3.5 Ablation experiments

In order to further verify the role of key modules in the model proposed in this paper, a set of ablation experiments are set up. The experimental results are shown in Table 7. It can be seen that the accuracy rate of the benchmark model ResNet50 is 97.43%, and the accuracy rate is increased by 0.42% after adding atrous convolution. When the channel attention mechanism is added to the residual module, the recognition accuracy rate is 98.21%; when the spatial attention mechanism is continued, the accuracy rate is 98.42%; when the DPA mechanism proposed in this paper is added, the accuracy rate can reach 99.28%, an increase of 1.85%, which is significantly improved compared with the basic model. These results show that the DPA module plays a crucial role in the overall framework, while the introduction of atrous convolution plays a role in the overall framework, but with limited effect. At the same time, the complexity of the model is measured by the number of parameters. It can be seen that the benchmark model ResNet50 has the smallest number of parameters. The model proposed in this paper increases the number of layers of the network by adding dual

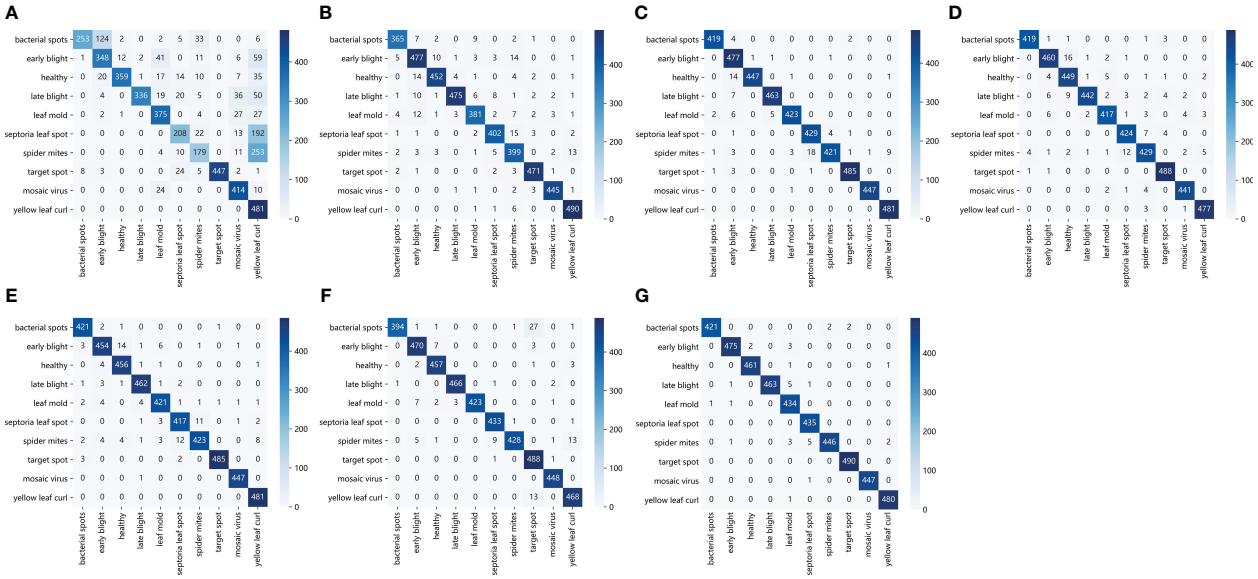



FIGURE 12
Confusion matrix: **(A)** AlexNet **(B)** MobileNetV2 **(C)** DenseNet121 **(D)** ResNet34 **(E)** ResNet50 **(F)** ResNet101 **(G)** ResNet50-DPA.

TABLE 6 Identification effect of the model for different disease types.

| Disease type | Accuracy/% | Precision/% | F1/% | support |
|---|---|---|---|---|
| Bacterial spot | 99.76 | 99.06 | 99.41 | 425 |
| Early blight | 99.37 | 98.96 | 99.16 | 480 |
| Late blight | 99.57 | 99.57 | 99.57 | 463 |
| Leaf mold | 100.00 | 98.51 | 99.25 | 470 |
| Leaf spot | 97.09 | 99.54 | 98.30 | 436 |
| spider mites | 98.42 | 1.00 | 99.20 | 435 |
| Target spot | 99.55 | 97.59 | 98.56 | 457 |
| Yellow leaf curl | 99.59 | 1.00 | 99.80 | 490 |
| Mosaic virus | 100.00 | 99.78 | 99.89 | 448 |
| healthy | 99.38 | 99.79 | 99.59 | 481 |
| Macro avg | 99.27 | 99.28 | 99.27 | 4585 |
| Weighted avg | 99.29 | 99.28 | 99.28 | 4585 |

TABLE 7 Results of ablation experiments.

| Description | Accuracy/% | Precision/% | Recall/% | F1/% | Params/M |
|---|---|---|---|---|---|
| ResNet50(baseline) | 97.43 | 97.43 | 97.43 | 97.42 | 25.56 |
| ResNet50 + atrous | 97.85 | 97.48 | 97.79 | 97.63 | 25.58 |
| ResNet50 + SENet | 98.21 | 98.23 | 98.43 | 98.33 | 28.19 |
| ResNet50 + CBAM | 98.42 | 98.38 | 98.43 | 98.41 | 28.36 |
| ResNet50 + DPA | 99.28 | 99.29 | 99.28 | 99.28 | 30.78 |

path attention mechanism and atrous convolution to the benchmark model ResNet50, resulting in an increase in the number of model parameters. However, from the final recognition accuracy of leaf disease, the effect of this model is ideal.

## 3.6 Visualization of experimental results

In this paper, the output feature maps of different convolutional layers in the ResNet50-DPA model are visualized, as shown in Figure S3. It can be seen that with the increase in the number of convolutional layers, the extracted information became more and more abstract, macro information replaced the detailed information of the image, and the model learned more abundant features.

Finally, in order to see more directly which region the model focused on during the training process, gradient class activation mapping (Grad-CAM) is used to draw the heat map of the input tomato leaf disease image in this paper. The idea of Grad-CAM algorithm is to use a specific convolutional neural network architecture to generate a visual heat map (Selvaraju et al., 2017). The key is to obtain the weight by obtaining the partial derivative of the category confidence of the network output to the feature graph. The formula is shown in the following Equation 11, where c is an image category, $y^c$ is the gradient corresponding to the category, $A^k$ is the feature activation value, and the weight $\alpha_k^c$ of neurons can be obtained by calculation. The final heat map is obtained by weighted summation of the weight values of the categories corresponding to all feature graphs. See Equation 12 for the specific formula.

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{11}$$

$$L_{Grad-CAM}^c = ReLU\left(\sum_k \alpha_k^c A^k\right) \tag{12}$$

The heap map results obtained after Grad-CAM processing are shown in Figure 13. Some tomato leaf disease images are randomly selected, as shown in Figure 13A. The visualization results obtained by using different models are shown in Figures 13B–H. The visualization features obtained by ResNet50-DPA model are shown in Figure 13H. It can be seen that the model proposed in this paper pays more accurate attention to the disease region, and the Grad-CAM explains the ability of the model to identify leaf disease, which shows the region of interest of the model to the input image.

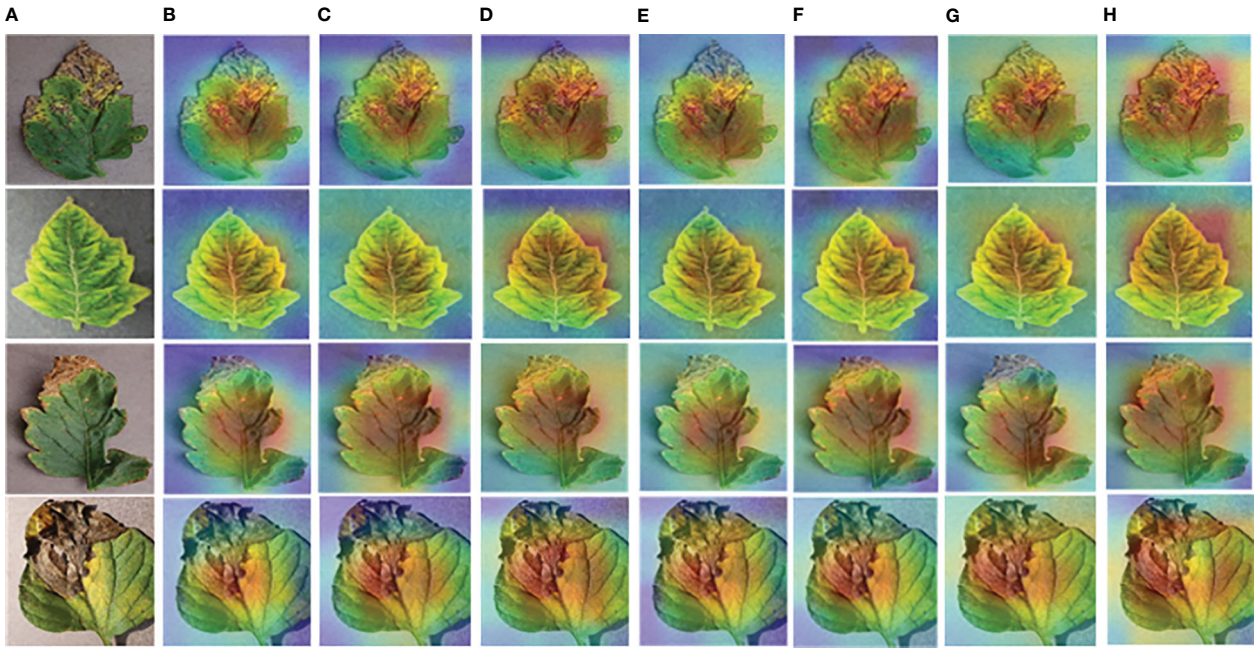



FIGURE 13
Visual heat map: **(A)** Original Image **(B)** AlexNet **(C)** MobileNetV2 **(D)** DenseNet121 **(E)** ResNet34 **(F)** ResNet50 **(G)** ResNet101 **(H)** ResNet50-DPA.

## 4 Discuss

In the identification of tomato leaf disease, a novel detection method based on ResNet50-DPA was proposed in this paper. How the proposed model accurately identifies disease types can be explained from the following two aspects.

On the one hand, the analysis is carried out according to the proposed detection method. The data enhancement technique is used to expand the data set to ensure the diversity of the data set and prevent the overfitting phenomenon. By replacing the first layer of convolution with a cascade of atrous convolution, atrous convolution can expand the range of receptive field and facilitate the extraction of features at different scales, instead of simply extracting blade features at the same scale. DPA is inserted into the residual module to enhance the extraction of important features. Three pooling operations are used in DPA, which not only reduces the problem of image information loss, but also eliminates the influence of non-maximum values in the model. The replacement of two one-dimensional convolution MLP layers also reduces the loss of information because it does not involve dimensionality reduction operations.

On the other hand, the analysis is based on the obtained results. Firstly, three sets of comparison experiments are carried out, and the experimental results show that the proposed model converges faster and has higher accuracy. Secondly, the confusion matrix are used to observe the classification performance of different disease categories. It can be seen that ResNet50-DPA has a good recognition effect on 10 tomato leaf disease types, and only a few samples are classified wrong. Finally, the Grad-CAM was used for visualization analysis. The Grad-CAM could highlight the most concerned regions of the model in the feature extraction process. The results showed that the proposed model could better focus on the disease regions of tomato leaves compared with other models, which verified the effectiveness of the proposed model.

In summary, although the current application scenarios are relatively limited, the models proposed in this paper can effectively identify different leaf disease types in limited scenarios, and the interpretability of the models is improved through visualization. The method proposed in this paper can help us to manage early disease, maintain crop growth, take control measures in time and reduce economic losses.

## 5 Conclusion

In this paper, a novel ResNet50-DPA model is proposed for tomato leaf disease recognition, which solves the problem of image information loss when extracting image features from conventional convolutional neural networks. In this method, the proposed DPA mechanism is introduced into the residual module of the residual network ResNet50 to enhance the ability of the network to extract details of tomato leaves. Then, cascades of atrous convolution are used to replace the first layer of convolution in ResNet50 to obtain leaf features at different scales and further improve the performance of the model. Finally, compared with the traditional methods, the experimental results show that the proposed model is superior to the common deep learning model, reaching an accuracy of 99.28%, which verifies the effectiveness of the proposed model for tomato leaf disease classification. In addition, the generalization ability and computing speed of the model proposed in the paper need to be further improved, which will be our future research.

## Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/spMohanty/PlantVillage-Dataset.

## Author contributions

WJ: Funding acquisition, Project administration, Supervision, Writing – review & editing. JL: Methodology, Visualization, Writing – original draft.

## Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This research was supported by the Foundation of Shanghai Institute of Technology (Grant number: XTCX2021-10).

## Acknowledgments

The authors wish to thank all the reviewers who participated in the review during the preparation of this manuscript.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2023.1258658/full#supplementary-material

SUPPLEMENTARY FIGURE 1
Image equalization processing: **(A)** Histogram before image equalization processing **(B)** Histogram after image equalization processing.

SUPPLEMENTARY FIGURE 2
Residual block structure: **(A)** The basic residual block **(B)** ResNet50 residual block.

SUPPLEMENTARY FIGURE 3
Feature map in visual convolution.

## References

Ahmad, N., Asif, H. M. S., Saleem, G., Younus, M. U., Anwar, S., and Anjum, M. R. (2021). Leaf image-based plant disease identification using color and texture features. *Wireless Pers. Commun.* 121, 1139–1168. doi: 10.1007/s11277-021-09054-2

Al-Wesabi, F. N., Albraikan, A. A., Hilal, A. M., Eltahir, M. M., Hamza, M. A., and Zamani, A. (2022). Artificial intelligence enabled apple leaf disease classification for precision agriculture. *Computers Materials Continue* 70, 6223–6238. doi: 10.32604/cmc.2022.021299

Ananda, S. P., and Vandana, B. M. (2022). Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG. *Artif. Intell. Agric.* 6, 23–33. doi: 10.1016/j.aiia.2021.12.002

Anandhakrishnan, T., and Jaisakthi, S. M. (2022). Deep Convolutional Neural Networks for image based tomato leaf disease detection. *Sustain. Chem. Pharm.* 30, 100793. doi: 10.1016/j.scp.2022.100793

Arafa, R. A., Kamel, S. M., Taher, D. I., Solberg, S. O., and Rakha, M. T. (2022). Leaf extracts from resistant wild tomato can be used to control late blight (Phytophthora infestans) in the cultivated tomato. *Plants* 11, 1824. doi: 10.3390/plants11141824

Bhatkar, N. S., Shirkole, S. S., Mujumdar, A. S., and Thorat, B. N. (2021). Drying of tomatoes and tomato processing waste: a critical review of the quality aspects. *Drying Technol.* 39, 1720–1744. doi: 10.1080/07373937.2021.1910832

Cao, Z., and Yuan, R. (2022). Real-time detection of mango based on improved YOLOv4. *Electronics* 11, 3853. doi: 10.3390/electronics11233853

Chen, L., Tian, X., Chai, G., Zhang, X., and Chen, E. A. (2021). New CBAM-P-net model for few-shot forest species classification using airborne hyperspectral images. *Remote Sens.* 13, 1269. doi: 10.3390/rs13071269

Chin, R., Catal, C., and Kassahun, A. (2023). Plant disease detection using drones in precision agriculture. *Precis. Agric.* 24, 1663–1682. doi: 10.1007/s11119-023-10014-y

Elhassouny, A., and Smarandache, F. (2019). "Smart mobile application to recognize tomato leaf diseases using Convolutional Neural Networks," in *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*, Agadir, Morocco. 1–4.

Faisal, M., Leu, J. S., Avian, C., Prakosa, S. W., and Koppen, M. (2023). DFNet: Dense fusion convolution neural network for plant leaf disease classification. *Agron. J.*, 21341. doi: 10.1002/agj2.21341

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA. 770–778.

Hu, Z., and Zhang, Y. (2023). Effect of dimensionality reduction and noise reduction on hyperspectral recognition during incubation period of tomato early blight. *Spectrosc. Spectral Anal.* 43, 744–752. doi: 10.3964/j.issn.1000-0593(2023)03-0744-09

Jing, J. P., Li, S. F., Qiao, C., Li, K. Y., Zhu, X. Y., and Zhang, L. X. (2023). A tomato disease identification method based on leaf image automatic labeling algorithm and improved YOLOv5 model. *J. Sci. Food Agric.* doi: 10.1002/jsfa.12793

Kute, R. S., Vyas, V., and Anuse, A. (2019). Component-based face recognition under transfer learning for forensic applications. *Inf. Sci.* 476, 176–191. doi: 10.1016/j.ins.2018.10.014

Li, D. L., Song, Z. Y., Quan, C. Q., Xu, X. B., and Liu, C. (2022). Recent advances in image fusion technology in agriculture. *Comput. Electron. Agric.* 191, 106491. doi: 10.1016/j.compag.2021.106491

Liu, B., Ding, Z., Tian, L., He, D., Li, S., and Wang, H. (2020). Grape leaf disease identification using improved deep convolutional neural networks. *Front. Plant Sci.* 11, 1082. doi: 10.3389/fpls.2020.01082

Liu, W., Yu, L., and Luo, J. (2022). A hybrid attention-enhanced DenseNet neural network model based on improved U-Net for rice leaf disease identification. *Front. Plant Sci.* 13, 922809. doi: 10.3389/fpls.2022.922809

Liu, S. H., Zhai, B. L., Zhang, J. T., Yang, L. P., Wang, J. Z., Huang, K., et al. (2023). Tomato detection based on convolutional neural network for robotic application. *J. Food Process Eng.* 46. doi: 10.1111/jfpe.14239

Manavalan, R. (2021). Automatic identification of diseases in grains crops through computational approaches: A review. *Comput. Electron. Agric.* 178, 105802. doi: 10.1016/j.compag.2020.105802

Moussafir, M., Chaibi, H., Saadane, R., Chehri, A., Rharras, A. E., and Jeon, G. (2022). Design of efficient techniques for tomato leaf disease detection using genetic algorithm-based and deep neural networks. *Plant Soil* 479, 251–266. doi: 10.1007/s11104-022-05513-2

Paul, R. K., Vennila, S., Bhat, M. N., Yadav, S. K., Sharma, V. K., Nisar, S., et al. (2019). Prediction of early blight severity in tomato (Solanum lycopersicum) by machine learning technique. *Indian J. Agric. Sci.* 89, 169–175. doi: 10.56093/ijas.v89i11.95344

Qin, F., Liu, D. X., Sun, B. D., Ruan, L., Ma, Z. H., and Wang, H. G. (2017). Identification of alfalfa leaf diseases using image recognition technology. *PLoS One* 11, e0168274. doi: 10.1371/journal.Pone.0168274

Rangarajan, A. K., Purushothaman, R., and Ramesh, A. (2018). Tomato crop disease classification using pre-trained deep learning algorithm. *Proc. Comput. Sci.* 133, 1040–1047. doi: 10.1016/j.procs.2018.07.070

Saleem, M. H., Potgieter, J., and Arif, K. M. (2019). Plant disease detection and classification by deep learning. *Plants* 8, 468. doi: 10.3390/plants8110468

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). "Grad-CAM: visual explanations from deep networks via gradient-based," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy. 618–626.

Sghair, M. E., and Naisbitt, J. (2017). An algorithm for plant diseases detection based on color features. *Int. J. Agric. Sci.* 2.

Tang, J. R., and Isa, N. A. M. (2017). Bi-histogram equalization using modified histogram bins. *Appl. Soft Computing* 55, 31–43. doi: 10.1016/j.asoc.2017.01.053

Thakur, P. S., Khanna, P., Sheorey, T., and Ojha, A. (2022). Trends in vision-based machine learning techniques for plant disease identification: A systematic review. *Expert Syst. Appl.* 208, 118117. doi: 10.1016/j.eswa.2022.118117

Theodora, S., Argyrios, S., Maria, V. S., and Minas, D. (2023). Tomato leaf disease identification via two–stage transfer learning approach. *Smart Agric. Technol.* 5, 100275. doi: 10.1016/j.atech.2023.100275

Ujawe, P., and Nirkhi, S. (2023). Comparative study of tomato crop disease detection system using deep learning techniques. *Intelligent Communication Technol. Virtual Mobile Networks* 131, 493–499. doi: 10.1007/978-981-19-1844-5_39

Vishnoi, V. K., Kumar, K., and Kumar, B. (2020). Plant disease detection using computational intelligence and image processing. *J. Plant Dis. Prot.* 128, 19–53. doi: 10.1007/s41348-020-00368-0

Wang, H., Li, G., Ma, Z., and Li, X. (2012). "Image recognition of plant diseases based on backpropagation networks," in *2012 5th International Congress on Image and Signal Processing*, Chongqing, China. 894–900.

Wang, L., Zhang, K., Liu, X., Long, E., Jiang, J., An, Y., et al. (2017). Comparative analysis of image classification methods for automatic diagnosis of ophthalmic images. *Sci. Rep.* 7, 41545. doi: 10.1038/srep41545

Yao, Q., Guan, Z., Zhou, Y., Tang, J., Hu, Y., and Yang, B. (2009). "Application of support vector machine for detecting rice diseases using shape and color texture features," in *2009 International Conference on Engineering Computation*, Hong Kong, China. 79–83.

Zhao, S., Peng, Y., Liu, J., and Wu, S. (2021). Tomato leaf disease diagnosis based on improved convolution neural network by attention module. *Agriculture* 11, 651. doi: 10.3390/agriculture11070651

Zhao, Y., Sun, C., Xu, X., and Chen, J. (2022). RIC-Net: A plant disease classification model based on the fusion of Inception and residual structure and embedded attention mechanism. *Comput. Electron. Agric.* 193, 106644. doi: 10.1016/j.compag.2021.106644

Zhou, C., Zhou, S., Xing, J., and Song, J. (2021). Tomato leaf disease identification by restructured deep residual dense network. *IEEE Access* 9, 28822–28831. doi: 10.1109/ACCESS.2021.3058947

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 23 October 2023
DOI 10.3389/fpls.2023.1256545

Check for updates

OPEN ACCESS

EDITED BY
Jucheng Yang,
Tianjin University of Science and Technology, China

REVIEWED BY
Zhaoyu Zhai,
Nanjing Agricultural University, China
Arpit Jain,
QpiAI India Pvt. Ltd., India

*CORRESPONDENCE
Jianhua Zhang
zhangjianhua@caas.cn
Lin Hu
hulin@caas.cn

RECEIVED 11 July 2023
ACCEPTED 20 September 2023
PUBLISHED 23 October 2023

CITATION
Pan P, Guo W, Zheng X, Hu L, Zhou G and Zhang J (2023) Xoo-YOLO: a detection method for wild rice bacterial blight in the field from the perspective of unmanned aerial vehicles.
*Front. Plant Sci.* 14:1256545.
doi: 10.3389/fpls.2023.1256545

COPYRIGHT
© 2023 Pan, Guo, Zheng, Hu, Zhou and Zhang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# Xoo-YOLO: a detection method for wild rice bacterial blight in the field from the perspective of unmanned aerial vehicles

Pan Pan[1,2], Wenlong Guo[3], Xiaoming Zheng[2,3], Lin Hu[1,2]*, Guomin Zhou[1,2,4] and Jianhua Zhang[1,2]*

[1]National Agriculture Science Data Center, Agricultural Information Institute, Chinese Academy of Agricultural Sciences, Beijing, China, [2]National Nanfan Research Institute (Sanya), Chinese Academy of Agricultural Sciences, Sanya, China, [3]Institute of Crop Science, Chinese Academy of Agricultural Sciences, Beijing, China, [4]Farmland Irrigation Research Institute, Chinese Academy of Agricultural Sciences, Xinxiang, China

Wild rice, a natural gene pool for rice germplasm innovation and variety improvement, holds immense value in rice breeding due to its disease-resistance genes. Traditional disease resistance identification in wild rice heavily relies on labor-intensive and subjective manual methods, posing significant challenges for large-scale identification. The fusion of unmanned aerial vehicles (UAVs) and deep learning is emerging as a novel trend in intelligent disease resistance identification. Detecting diseases in field conditions is critical in intelligent disease resistance identification. In pursuit of detecting bacterial blight in wild rice within natural field conditions, this study presents the Xoo-YOLO model, a modification of the YOLOv8 model tailored for this purpose. The Xoo-YOLO model incorporates the Large Selective Kernel Network (LSKNet) into its backbone network, allowing for more effective disease detection from the perspective of UAVs. This is achieved by dynamically adjusting its large spatial receptive field. Concurrently, the neck network receives enhancements by integrating the GSConv hybrid convolution module. This addition serves to reduce both the amount of calculation and parameters. To tackle the issue of disease appearing elongated and rotated when viewed from a UAV perspective, we incorporated a rotational angle (theta dimension) into the head layer's output. This enhancement enables precise detection of bacterial blight in any direction in wild rice. The experimental results highlight the effectiveness of our proposed Xoo-YOLO model, boasting a remarkable mean average precision (mAP) of 94.95%. This outperforms other models, underscoring its superiority. Our model strikes a harmonious balance between accuracy and speed in disease detection. It is a technical cornerstone, facilitating the intelligent identification of disease resistance in wild rice on a large scale.

KEYWORDS

**wild rice, UAV, bacterial blight, disease detection, deep learning, YOLOv8**

# 1 Introduction

Wild rice, which is closely related to cultivated Asian rice (referred to as "rice" hereafter), serves as a valuable parental material for rice breeding. It shares the same chromosome group as rice and harbors exceptional genes that are either absent or lost in rice. Consequently, wild rice acts as a fundamental germplasm resource for genetic enhancement in rice (Yuan, 1986; Fan et al., 2023; Shao et al., 2023). As the genetic background for disease resistance in rice becomes increasingly limited, the screening of superior disease resistance genes in wild rice, which have been lost during the domestication of rice, and utilizing them in breeding emerges as an effective and cost-efficient approach to mitigating rice diseases. The identification of disease resistance in wild rice is one of the key aspects of breeding disease-resistant varieties of rice. In the process of identifying new genes with broad-spectrum and durable disease resistance traits and resolving their molecular mechanisms of resistance, different wild rice resources need to be identified and validated for disease resistance in order to select high-quality germplasm resources and breeding materials with high disease resistance. The identification of disease resistance in wild rice is instrumental in bolstering the construction, conservation, and utilization of wild rice germplasm resources, as well as in breeding superior rice varieties with enhanced disease resistance (Yun and Han, 2014).

Rice bacterial blight, caused by *Xanthomonas oryzae* pv. *oryzae*, is a highly detrimental bacterial disease that significantly affects the growth of rice. This disease not only leads to a decrease in yield but also negatively impacts the quality of rice (Chen et al., 2021). In comparison to the utilization of fungicides, the utilization of bacterial blight resistance genes from wild rice and the development of genetic varieties with broad-spectrum resistance to bacterial blight offer economic, environmentally friendly, and safe alternatives. This approach is currently a prominent research focus in the field of plant immunity and a crucial objective for crop breeding worldwide (Xu et al., 2019). In the conventional process of disease resistance identification, researchers are required to visually observe and manually test the presence of diseases in wild rice within relatively distant experimental fields, which is, however, both time-consuming and subjective. Given the growing demand for large-scale identification of disease resistance in wild rice, there is a pressing need to develop a high-throughput and intelligent approach for accurately identifying disease resistance in wild rice on a large scale, which would enable automated disease resistance identification in the field.

The method of field-based wild rice disease detection enables precise localization of diseases affecting wild rice from an unmanned aerial vehicle (UAV) perspective. It serves as a crucial component in the intelligent identification of wild rice disease resistance, providing fundamental support for post-processing tasks such as wild rice disease segmentation, disease spot measurement, and disease resistance identification. Deep learning algorithms possess the ability to autonomously learn and represent features, and they can partially replace manual disease detection with their high robustness and accuracy (Liu and Wang, 2021; Shao et al., 2022). It has been extensively utilized in the detection of diseases in maize (Khan et al., 2023), potatoes (Dai et al., 2022), strawberries (Lee et al., 2022), citrus (Qiu et al., 2022), and other crops (Dai and Fan, 2022). In recent years, researchers have employed deep learning techniques to detect rice bacterial blight. For instance, Jia et al. (2023) utilized MobileNetV3 to substitute the original YOLOv7 algorithm's backbone network. They integrated the coordinate attention (CA) module into the feature fusion layer of YOLOv7 to impart richer semantic information. Additionally, they incorporated SIoU to bolster precision and robustness, mitigating overfitting concerns. This enhancement led to an impressive average precision (AP) of 98%. To address the challenges associated with the undefined value of "k" in the k-means clustering algorithm, which often leads to suboptimal solutions, Zhou et al. (2019) proposed the FCM-KM algorithm. This innovative approach employs maximum and minimum distances to determine both the optimal "k" value and the central positions for clustering. To enhance the detection of rice disease, they integrated the FCM-KM algorithm with the Faster R-CNN model. This fusion yielded impressive results, with a detection accuracy of 97.53% and a processing time of 0.62 s. In a similar vein, Prasomphan (2023) utilized the YOLOv3 model, achieving a notable AP of 89.6%. Kumar et al. (2023) introduced a multi-scale YOLOv5 detection network. This innovation in detection accuracy is achieved through the integration of the DAIS segmentation and Bi-FAPN networks. Their approach also effectively reduces computational costs by employing the principled pruning technique. Remarkably, their model achieves a mAP of 82.8% on the RLD dataset. On the other hand, Haque et al. (2022) employed the YOLOv5 model, attaining an AP of 65%.

The efforts of the aforementioned researchers have certainly propelled the progress of rice bacterial blight detection. They have contributed valuable insights in areas such as dataset enhancement and optimization of detection algorithms. However, there remains room for further improvement in accurately detecting small and densely clustered targets within the intricate field conditions. Moreover, considering various angles of the disease under the UAV viewpoint and acknowledging the subtle disparities in visual features between wild and cultivated rice diseases, the current methodologies are inadequate to meet the demands of disease-resistant breeding applications for wild rice bacterial blight. Furthermore, in the realm of object detection models, achieving real-time wild rice bacterial blight detection from UAV necessitates the integration of algorithms with swift inference capabilities. The faster R-CNN model (Ren et al., 2017), representing a two-stage detection model, demonstrates higher accuracy but slower speed. On the other hand, the YOLO model, representing a one-stage detection model, offers a significant advantage in speed compared to the two-stage model, making it better suited for real-time detection requirements (Pan et al). Hence, this study opts for the latest YOLO series algorithm, YOLOv8, as the baseline model.

With a specific focus on wild rice bacterial blight, the research introduces a novel approach built on the Xoo-YOLO model for detecting bacterial blight in wild rice from the viewpoint of UAV within field environments. This method is designed to address challenges associated with low accuracy in detection under the perspective of UAV in fields. It also addresses the inability of the

horizontal bounding box detection to detect various angles of the disease detected from the UAV viewpoint. The overarching aim is to enhance the precision and robustness of bacterial blight detection in wild rice when viewed from the UAV's perspectives within field conditions.

The important contributions of this paper are as follows:

(1) Images of wild rice bacterial blight were collected as a dataset from the perspective of UAV in natural field environments. These images were utilized for training, validating, and testing the model.

(2) A method for wild rice bacterial blight detection in the field from the UAV perspective, which is based on the Xoo-YOLO model, is proposed to address the issue of poor detection performance for dense or small-object disease targets and balancing both the detection accuracy and speed. This method meets the demand for wild rice bacterial blight detection in breeding disease resistance. The backbone network was introduced into the Large Selective Kernel Network (LSKNet) to better achieve the detection of disease targets under the UAV viewpoint by dynamically adjusting its large spatial receptive field. Simultaneously, the neck network is enhanced by introducing the hybrid convolution module of the GSConv to reduce the amount of calculation and parameters of the model.

(3) An oriented bounding box detection method with rotating angles (theta dimension) is proposed to address the issue of inaccurate detection brought on by the wild rice bacterial blight, which presents arbitrary angles from the UAV viewpoint. This method achieves localized detection of wild rice disease spots in arbitrary directions while reducing the interference brought about by too much background information introduced and improving the network's ability to extract disease.

(4) Through experiments, the proposed Xoo-YOLO model has been validated on the dataset of wild rice bacterial blight. Across metrics such as accuracy, recall, and *F*1 score, this approach consistently outperforms. Notably, it achieves this superiority while effectively reducing model parameters and computational complexity. Striking a balance between accuracy and efficiency, it is better poised to cater to real-time detection requirements under resource-constrained settings, such as UAV and other edge devices.

# 2 Materials and methods

## 2.1 Materials

### 2.1.1 Material preparation

Wild rice germplasm resources from the Institute of Crop Science, Chinese Academy of Agricultural Sciences, were planted at the Potianyang Base, Yazhou District, Sanya City, Hainan Province, China (N: 18°39′84.13″, E: 109°17′51.68″). On 1 March 2023, a total of 120 wild rice samples were artificially inoculated with the pathogen of rice bacterial blight. Following the "The technique rules for identification of rice variety resistance against bacterial blight (*Xanthomonas oryzae* pv. *oryzae*)," the inoculation was performed using the pathogenic strain PXO99A of *Xanthomonas oryzae* pv. *oryzae*, which is known to cause rice bacterial blight. The inoculation method involved manually cutting leaves during the tillering stage (Administration, A.P.M.S, 2017; Tang et al., 2017).

### 2.1.2 Image acquisition and processing

The camera-equipped UAV was utilized to acquire RGB images of wild rice at different times after infection with bacterial blight. The image acquisition method of wild rice bacterial blight is shown in Figure 1. The type of UAV equipment is Dji Mini2 (DJI Inc, Shenzhen, China), the flying altitude is 0.6–1.5 m above the surface of the wild rice field, the camera head pitch angle is −90° to −60°, and the output image resolution is 1,920 × 1,080, which is saved in JPG format. Image acquisition was conducted at different time intervals, specifically on the fifth, seventh, ninth, 12th, 14th, 16th, and 18th days after infection. The acquisition periods were from 8:00 AM to 11:00 AM and from 4:30 PM to 6:30 PM. The weather conditions during image acquisition included sunny, cloudy, and overcast days, with environmental temperatures ranging from 22°C to 30°C. All images were captured in the natural field environment, utilizing natural lighting without the use of flash. The images contained various types of interference, such as different levels of occlusion, water surface reflections, and overexposure, as well as weeds, withered leaves, bird droppings, and field debris. Each image contained one-fourth to two wild rice plants. In total, 750 images of infected wild rice with bacterial blight were collected under the UAV viewpoint. The days since inoculation, figures, and image samples in the wild rice bacterial blight dataset are shown in Figure 2. These images were further identified and confirmed by two experts specializing in wild rice germplasm identification. The open-source software roLabelImg (cgvict, 2017) was used to manually annotate wild rice leaf blight spot areas, and the annotation information was saved as a file in.xml format using this software. The dataset was divided and split in the ratio of 6:2:2, with 450 images randomly selected as the training set, 150 as the validation set, and 150 as the test set. Furthermore, the inclusion of mosaic augmentation is notable during training. Mosaic augmentation involves the random extraction of four images, where each image contributes only a portion of its content and corresponding detection box information. These fragments are then combined into a single image, serving as input for the network. This technique significantly diversifies the training data, effectively guarding against overfitting by introducing greater variability into the learning process.

## 2.2 Methods

### 2.2.1 Overall model

Given the intricate backgrounds inherent in wild rice paddies—encompassing water, shadows, and reflections—and the dynamic

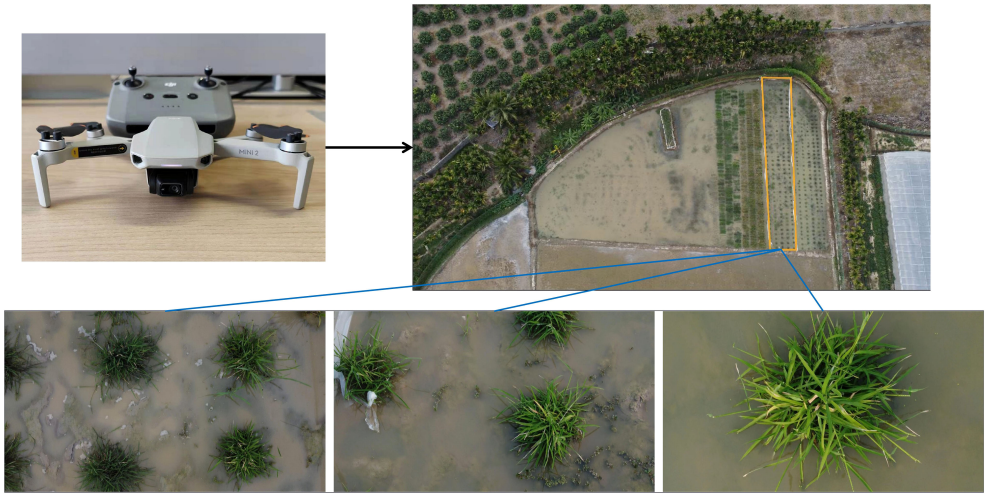


FIGURE 1
Using an unmanned aerial vehicle (UAV) to acquire image data of wild rice infected with bacterial blight.

alterations in the manifestation of visual disease attributes due to the interplay of UAV propeller airflow and variable weather conditions, the imperative for real-time detection underscores our decision to employ YOLOv8 as the baseline model for this research. In this context, Xoo-YOLO emerges as an enhanced iteration built upon YOLOv8, specifically tailored for the task of detecting wild rice bacterial blight from the UAV perspective in the field. The architecture of Xoo-YOLO is visualized in Figure 3.

The model comprises four principal components: input, backbone, neck, and head. The enhancements incorporated are outlined as follows:

(1) Integration of LSKNet into the backbone: To better achieve the detection of disease targets under the UAV viewpoint, we introduced LSKNet into the backbone network, which adopts the dynamic adjustment of its large spatial receptive field, allowing the model to adaptively use different large kernels and adjust the receptive field for each target in space as needed.

(2) Construction of a lightweight model: In pursuit of a more streamlined design, the neck network is enhanced by introducing the hybrid convolution module of the GSConv to reduce the number of calculations and parameters of the model.

(3) Inclusion of an oriented bounding box detection method: To detect the wild rice bacterial blight under the UAV viewpoint with arbitrary angles, we introduce an oriented bounding box detection method. This approach integrates angle information (theta) within the head layer and loss function, reducing the interference brought about by too much background information and improving the network's ability to extract disease.

By amalgamating these advancements, Xoo-YOLO is equipped to address the intricacies associated with wild rice bacterial blight detection from the UAV viewpoint, offering improved accuracy and robustness. The whole algorithm is summarized in the pseudo-lcode in Algorithm 1.

```
Input: Image I, Confidence Threshold T
Output: Detected objects with oriented bounding boxes and labels
1. Scaling and normalizing the input image I
2. Feed image I into the network to obtain the output feature map
3. Prediction for each grid unit using the feature map:
   a. Predict the presence of an object in each grid unit using the BCE function
   b. Predict the category of objects in each grid unit using the BCE function
   c. Predict the location and size of objects in each grid unit using the KLD function
4. Post-processing of predicted bounding boxes:
   a. Remove bounding boxes with confidence lower than threshold T
   b. Apply non-maximum suppression to remove overlapping bounding boxes
5. Output final prediction results: Category, confidence, and location information of wild rice bacterial blight
```

Algorithm 1. Pseudocode of Xoo-YOLO Algorithm.

### 2.2.2 YOLOv8

YOLOv8, a prominent object detection algorithm, was introduced by Ultralytics in January 2023. This algorithm has exhibited commendable outcomes in both speed and accuracy (Terven and Cordova-Esparza, 2023). The backbone of YOLOv8 centers around the C2f module, drawing inspiration from the ELAN

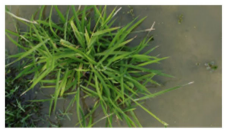
| Days since inoculation | Figures | Image |
|---|---|---|
| Day5 | 55 |  |
| Day7 | 104 |  |
| Day9 | 120 |  |
| Day12 | 110 |  |
| Day14 | 132 |  |
| Day16 | 125 |  |
| Day18 | 104 |  |
| **Total** | **750** | |

FIGURE 2
The days since inoculation, figures, and image samples in the wild rice bacterial blight dataset.

module. The C2f module is constructed from two Conv modules and multiple Darknet BottleNeck modules, interconnected by Split and Concat modules. Additionally, YOLOv8 incorporates the Spatial Pyramid Pool Fusion (SPPF) module. This innovation transforms feature maps of varying sizes into fixed-size feature vectors, effectively preserving the original image's features and positional information to the maximum extent. The YOLOv8's neck network incorporates the Path Aggregation Network (PANet), similar to YOLOv7 (Wang et al., 2023), which fuses three effective feature layers obtained from the backbone network across layers of features, with the three effective feature layers located in the middle, lower middle, and bottom layers of the backbone network.

### 2.2.3 Improved backbone network structure

Owing to the intricate setting of wild rice paddies, encompassing water, mud, algae, weeds, bird droppings, shadows, and other intricate backgrounds, coupled with the relatively diminutive size of wild rice bacterial blight as viewed from UAV, relying on limited contextual information often triggers incorrect detection. For instance, it may lead to the misinterpretation of white streaks of debris within the wild rice paddies as instances of wild rice bacterial blight. Simultaneously, the varying viewpoints and distances of the UAV introduce distinct contextual information requirements for accurate detection. However, introducing an excess of contextual information can inadvertently obscure target features' specifics while exacerbating the model's complexity.

To address the aforementioned challenges, the presented Xoo-YOLO model incorporates the LSKNet (Li et al., 2023). LSKNet dynamically adjusts its large spatial receptive field, enabling the model to flexibly employ varying large kernels and modify the receptive field according to the specific spatial requirements of each target. This adaptability is especially vital for detecting wild rice bacterial blight amid the UAV viewpoint and intricate conditions of the field.

LSKNet is implemented through a spatial selection mechanism, which is implemented by effectively weighting the features processed by a sequence of large depth-wise convolutional kernels and spatially merging them. The weights of this kernel are dynamically determined based on the input, allowing the model to adaptively use different large kernels and adjust the receptive field for each target in space as needed.

LSKNet mainly consists of two sub-blocks, Large Kernel Selection (LK Selection) and Feed-Forward Network (FFN). FFN is used for channel mixing and feature refinement and consists of a sequence of a fully connected layer, a depth-wise convolution, a GELU activation, and another fully connected layer; LK Selection consists of a fully connected layer, LSK sub-block, a GELU activation, and another fully connected layer.

The central component of LSKNet is the LSK sub-block, which comprises a series of large kernel convolutions alongside a spatial kernel selection mechanism. Larger kernel convolution constructs by explicitly decomposing it into a sequence of depthwise convolutions with a large growing kernel and increasing dilation. To elaborate, the expansion of various parameters, such as kernel size "$k$," dilation rate "$d$," and the receptive field "RF," within the $i$th depthwise convolution within the sequence, is characterized by the following definitions:

$$\begin{aligned} &k_{i-1} \le k_i; d_1 = 1, d_{i-1} < d_i \le \mathrm{RF}_{i-1}, \\ &\mathrm{RF}_1 = k_1, \mathrm{RF}_i = d_i(k_i - 1) + \mathrm{RF}_{i-1}\,. \end{aligned} \tag{1}$$

The increasing kernel size and dilation rate ensure that the receptive field expands quickly enough, and at the same time, an upper bound on the dilation rate is set to avoid the dilation convolution introducing gaps between feature maps. This approach makes the later kernel selection easier and also reduces the number of parameters significantly. Simultaneously, a series of decomposed depthwise convolutions with different receptive fields are used to obtain features with contextual information at different ranges, allowing channel blending for each spatial feature vector. The calculation is shown in Eqs. (2) and (3).

$$\mathbf{U}_0 = \mathbf{X}, \mathbf{U}_{i+1} = \mathcal{F}_i^{dw}(\mathbf{U}_i) \tag{2}$$

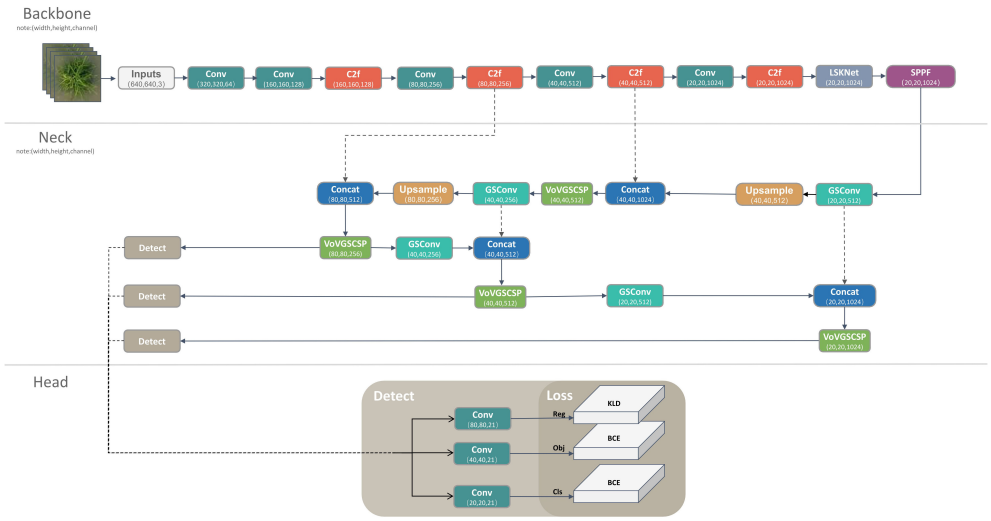



FIGURE 3
Overall model architecture diagram.

$$\widetilde{\mathbf{U}}_i = \mathcal{F}_i^{1\times1}(\mathbf{U}_i),\ \text{for } i \text{ in } [1, N] \tag{3}$$

Based on the above, LSKNet employs a spatial kernel selection mechanism to enhance the network's ability to focus on the most relevant spatial context regions, spatially selecting feature mappings from large convolutional kernels of different scales.

The outlined implementation steps are as follows:

(1) Concatenate features obtained from different kernels with different ranges of receptive fields:

$$\widetilde{\mathbf{U}} = [\widetilde{\mathbf{U}}_1; \ldots; \widetilde{\mathbf{U}}_i] \tag{4}$$

(2) Efficient spatial relation extraction using channel-based average pooling and maximum pooling:

$$\mathbf{SA}_{\text{avg}} = \mathcal{P}_{\text{avg}}\left(\widetilde{\mathbf{U}}\right)\ \mathbf{SA}_{max} = \mathcal{P}_{max}\left(\widetilde{\mathbf{U}}\right) \tag{5}$$

ally pooled features and use a convolutional layer $\mathcal{F}^{2\to N}(\cdot)$ to transform the pooled features (with two channels) into $N$ spatial attention maps:

$$\widehat{\mathbf{SA}} = \mathcal{F}^{2\to N}([\mathbf{SA}_{\text{avg}}; \mathbf{SA}_{\text{max}}]) \tag{6}$$

(4) A sigmoid activation function is used to obtain a separate spatial selection mask for each decomposed kernel:

$$\widetilde{\mathbf{SA}_i} = \sigma(\widehat{\mathbf{SA}}_i) \tag{7}$$

(5) The features in the decomposed macronucleus sequence are weighted with the corresponding spatially selective masks and fused by a convolutional layer $\mathcal{F}(\cdot)$ to obtain the attention feature $\mathbf{S}$:

$$S = \mathcal{F}\left(\sum_{i=1}^{N}(\widetilde{\mathbf{SA}_i} \cdot \tilde{\mathbf{U}}_i)\right) \tag{8}$$

(6) The final output of the LSK module is an element-by-element product between input features $X$ and $S$:

$$\mathbf{Y} = \mathbf{X} \cdot \mathbf{S} \tag{9}$$

LSKNet fulfills the requirement for detecting bacterial blight in wild rice from the perspective of UAV in the field while addressing the necessity for a wider and adaptable contextual understanding without bells and whistles. Therefore, we add LSKNet to the backbone network to enhance the feature extraction capability of this model. The structural diagram of LSKNet is illustrated in Figure 4.

### 2.2.4 Improved neck network structure

The standard convolution (SC) module used in YOLOv8 applies different convolutional kernels to multiple channels simultaneously, which leads to an increase in the number of parameters required and high FLOP. On the other hand, although lightweight networks using depthwise separable convolutions (DSC) such as MobileNet (Howard et al., 2017) and ShuffleNet (Zhang et al., 2018) can effectively solve this problem and greatly improve the detection performance, DSC separates channel information from the input image during computation, leading to a significant reduction in the feature extraction and fusion capabilities and resulting in a decrease in the detection performance of the model. It cannot meet the real-time requirements of disease resistance identification for the detection of wild rice bacterial blight.

To enhance computation speed without compromising detection accuracy, the Xoo-YOLO model integrates the GSConv hybrid convolution module (Li et al., 2022). This module incorporates Shuffle to infuse information produced by SC into the information generated by DSC. In contrast to DSC, the strength of GSConv lies in its ability to maintain hidden connections while operating with reduced complexity. This approach adeptly strikes a model equilibrium between accuracy and speed, ensuring a judicious trade-off between the two.

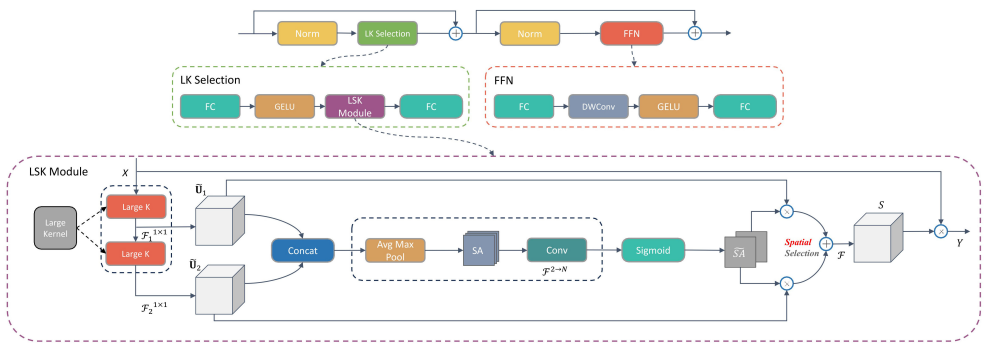



FIGURE 4
Structural diagram of LSKNet.

The GSconv module is primarily composed of Conv, DWConv, Concat, and Shuffle operations. This configuration is illustrated in Figure 5A and is constructed as follows:

(1) The input feature map has a total of C1 channels.
(2) DSC is applied to half of the channels, while SC is applied to the remaining half.
(3) The resulting two output feature maps are concatenated along the channel dimension.
(4) The concatenated feature map is then subjected to a shuffle operation, yielding the final output.
(5) The final output feature map possesses a total of C2 channels.

VoVGSCSP is an iterative integration of the GS bottleneck utilizing the foundation of GSConv (Xu et al., 2023). The process involves segmenting the input feature map's channel count into two segments. The initial portion traverses through convolution (Conv) for processing, following which the features undergo extraction through consecutively stacked GS bottleneck modules. On the other hand, the remaining segment is utilized as residuals, engaging in a single convolution operation. This module aptly harmonizes the model's accuracy and speed, resulting in a reduction of both computational and complexity. Simultaneously, it sustains a commendable degree of accuracy and significantly augments the reutilization rate of extracted features. The structural representation of VoVGSCSP can be observed in the diagram depicted in Figure 5B.

Ultimately, the neck network underwent refinement through the amalgamation of GSConv and VoVGSCSP. These enhancements served to diminish the model's overall computational overhead, resulting in swifter network operation and reduced information processing time. This harmonious adjustment better balances the trade-off between detection speed and accuracy. The augmented

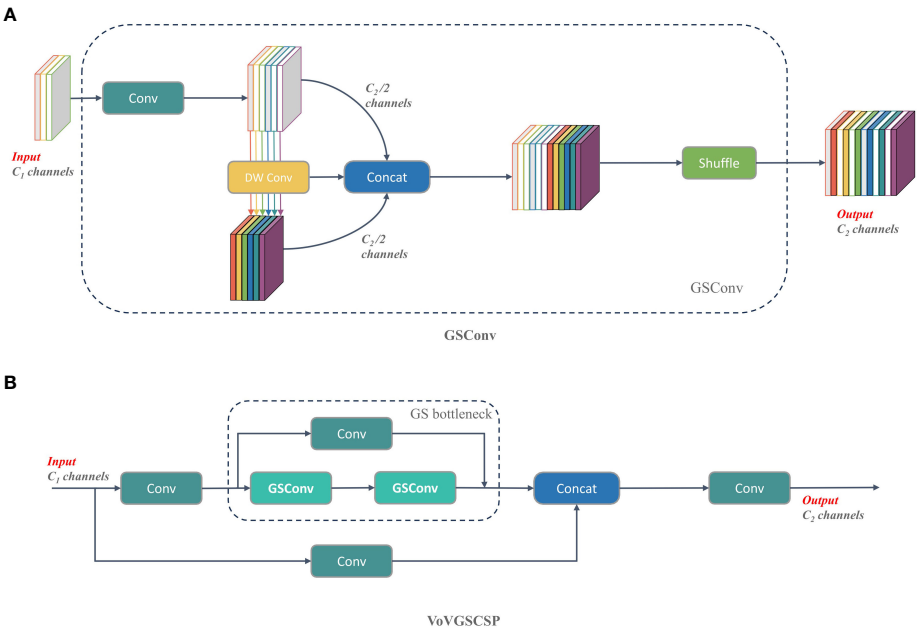



FIGURE 5
Structural diagram of GSConv and VoVGSCSP module. **(A)** GSConv module. **(B)** VoVGSCSP module.

configuration of the neck network is visually presented in the provided Figure 6.

### 2.2.5 Oriented bounding box

The UAV was taken from above, and the target in its images usually appears at any angle. The wild rice leaves are striped and lanceolate in shape, and the bacterial blight is also striped on the leaves. This makes the horizontal bounding box of the original YOLOv8 model overlap too much and inevitably introduces too much background information into the horizontal bounding box of adjacent targets, which not only leads to the phenomenon of missed detection and wrong detection at the same time but also increases the difficulty of extracting features from the network (Zhang et al., 2021). Recently, oriented bounding boxes, which include an angular dimension, have been utilized to represent objects with different orientations (Liao et al., 2018). For instance, Zhang et al. (2023) utilized oriented bounding boxes to detect *Fusarium* head blight (FHB) in wheat and achieved notably high accuracy and robustness in predicting FHB levels. The use of oriented bounding boxes for detecting crop diseases from a UAV perspective has proven to be effective. In light of this, the model proposed in this study adopts oriented bounding box detection. This approach involves angular regression (Ren et al., 2015) and employs the long-edge definition method (Ma et al., 2018) to regress the minimum bounding rectangle of the target. This enhancement is crucial for ensuring reliable and accurate wild rice bacterial blight detection in field conditions.

The horizontal bounding box parameters are expressed as ($x$, $y$, $w$, and $h$). The four parameters indicate the horizontal and vertical coordinates of the center of the horizontal bounding box, width, and height, respectively. The oriented bounding box contains five parameters ($x$, $y$, $w$, $h$, and $\theta$), with $\theta$ indicating the angle of rotation (Li et al., 2020). The structural diagram of the head is shown in Figure 7. To prevent duality, the long-edge definition method (Figure 8) is used, i.e., $\theta$ is between −90° and 90°, $w$ is the longest edge, its neighbor is $h$, and $\theta$ represents the range of angles through which the $x$-axis is rotated to $w$. The model proposed in this paper adds a new rotation angle prediction channel to the head structure to implement the detection of the oriented bounding box, with a dimension of $3 \times (5 + 1 + C)$, where 3 represents that each grid will be predefined with three predicted boxes of various aspect ratios, 5 represents that each predicted box will predict the parameter ($x$, $y$, $w$, $h$, and $\theta$) of the border, 1 is used to determine whether each grid contains the object, and the final $C$ parameters is used to determine the type of object each grid contains.

The loss function of the model proposed in this paper consists of three parts: Reg part, Obj part, and Cls part. The Reg part is the regression parameter judgment of the feature points, the Obj part is the judgment of whether the feature points contain objects, and the Cls part is the kind of objects contained in the feature points. In order to avoid the differences between the angular parameter $\theta$ and other parameters to bring difficulties to the training of the model, the Reg part uses KLD as the loss function (Yang et al., 2021), whose core idea is to convert the oriented bounding box into a two-dimensional Gaussian distribution, i.e. ($x$, $y$, $w$, $h$, and $\theta$) into a two-dimensional Gaussian distribution of $\mathcal{N}_{(\mu,\Sigma)}$, and the conversion is shown in Eq. (10). The KLD between the Gaussian distribution is calculated as the regression, the loss is calculated as shown in Eq. (11), where the subscripts $p$ and $t$ denote the predicted distribution results and the actual results respectively. The KLD loss function is scale-invariant and can dynamically adjust the gradient weights of the angle parameters according to the aspect ratio of the object, and this self-modulation optimization mechanism effectively promotes the accuracy of oriented bounding box detection. The Obj and Cls parts, on the other hand, adopt a binary cross-entropy loss function to reduce the computational complexity of training.

$$\begin{aligned}
\boldsymbol{\mu} &= (x, y)^{\top} \\
\Sigma^{1/2} = \mathbf{RAR}^{\top} &= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \frac{w}{2} & 0 \\ 0 & \frac{h}{2} \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \\
&= \begin{pmatrix} \frac{w}{2}\cos^2\theta + \frac{h}{2}\sin^2\theta & \frac{w-h}{2}\cos\theta\sin\theta \\ \frac{w-h}{2}\cos\theta\sin\theta & \frac{w}{2}\sin^2\theta + \frac{h}{2}\cos^2\theta \end{pmatrix}
\end{aligned} \tag{10}$$

$$\begin{aligned}
D_{kl}(\mathrm{N}_{p'} \,\|\, \mathrm{N}_{t'}) &= \tfrac{1}{2}(\mu_p - \mu_t)^T \Sigma_t^{-1}(\mu_p - \mu_t) + \tfrac{1}{2}\mathrm{Tr}(M^T(M^T)^{-1}\Sigma_t^{-1}M^{-1}M\Sigma_P) \\
&\quad + \ln\frac{|\Sigma_t|}{\Sigma_P} - 1 \\
\mathcal{L}_{\mathrm{KLD}} &= 1 - \frac{1}{\tau + f(\mathrm{D})}, \tau \geq 1
\end{aligned} \tag{11}$$

## 3 Experiments and analysis of results

### 3.1 Training procedures

The operating environment for this experimental was a Dell tower workstation (Dell, Inc) with an operating system environment of Windows 11, a 12th-Gen Intel® Core™ i5-12500 3.00 GHz processor, 32 G of on-board running memory, a 1-TB solid-state drive, and an NVIDIA GeForce RTX 3080 graphics card with 10 GB of video memory and using GPU-accelerated

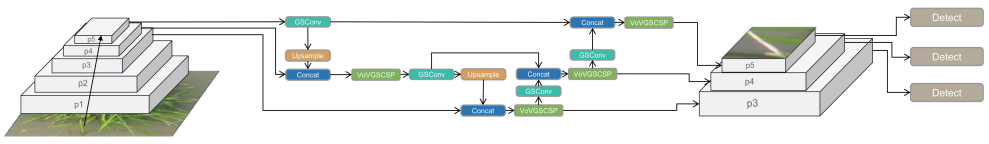



FIGURE 6
Structural diagram of the neck.

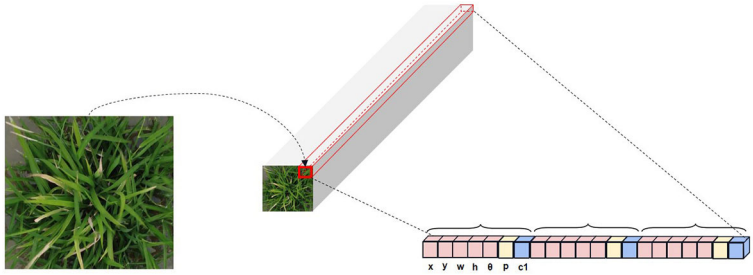



FIGURE 7
Structural diagram of the head.

computing. (Software environment: Python 3.7.16, PyTorch 1.7.0, Torchvision 0.8.2, CUDA 11.0.)

The number of iterations in this experiment was 700, batch_size was set to 2, and Adam was used as the optimizer. The initial learning rate of the models was 1*e*−3, the maximum learning rate was 1*e*−5, the momentum was 0.937, the weight decay was 0, and the input image resolution was 640 × 640. The same training parameters and dataset were used for all models during training.

## 3.2 Performance evaluation

To accurately evaluate the effectiveness of the method proposed in the previous section, this paper uses several evaluation metrics, including precision (*P*), recall (*R*), *F*1 score, mean average precision (mAP), speed (FPS), number of parameters (Params), and GFLOPs. These metrics are utilized to assess the performance of the model in terms of its detection accuracy and efficiency.

The precision refers to the proportion of correctly classified positive samples out of all the samples predicted. It is calculated using the formula presented in Eq. (12).

$$\textbf{Precision} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FP}} \tag{12}$$

Where TP represents the number of true-positive samples (correctly predicted positive samples), and FP represents the number of false-positive samples (negative samples incorrectly predicted as positive).

The recall quantifies the proportion of positive samples that are correctly identified by the model out of the total number of actual positive samples. It is calculated using the Eq. (13).

$$\textbf{Recall} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FN}} \tag{13}$$

The *F*1 score combines both precision and recall into a single value. The equation for calculating the *F*1 score is shown in Eq. (14).

$$\textbf{F1} = 2 \cdot \frac{\boldsymbol{P} \cdot \boldsymbol{R}}{\boldsymbol{P} + \boldsymbol{R}} \tag{14}$$

The mAP is calculated based on the precision–recall (PR) curve, which only needs to detect a single disease; mAP is equivalent to AP. The equation for calculating mAP is shown in Eq. (15).

$$\boldsymbol{mAP} = \boldsymbol{AP} = \int_0^1 \boldsymbol{P(R)dR} \tag{15}$$

The Params reflects the model's complexity and capacity to learn and represent features. The equation for calculating Params is shown in Eq. (16).

$$\textbf{Params} = [\mathbf{i} \cdot (\mathbf{k} \cdot \mathbf{k}) \cdot \mathbf{o}] + \mathbf{o} \tag{16}$$

Where *i* is the input size, *k* is the convolution kernel size, and *o* is the output size.

Speed is measured in frames per second (FPS). The equation for calculating speed is shown in Eq. (17).

$$\textbf{Speed} = \textbf{frames} \,/\, \textbf{time} \tag{17}$$

GFLOPS is the speed of the model based on computation costs. The formula for calculating GFLOPS is shown in Eq. (18).

$$\textbf{GFLOPS} = \mathbf{H} \cdot \mathbf{W} \cdot \textbf{params} \tag{18}$$

Where $H \times W$ is the size of the outputted feature map.

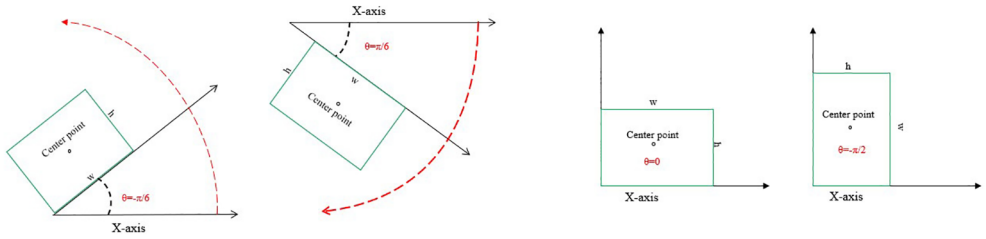



FIGURE 8
Long-edge definition method.

## 3.3 Ablation experiment

To offer a more comprehensive understanding of the effectiveness of the proposed enhancement technique applied to the Xoo-YOLO model, a series of ablation experiments were conducted. YOLOv8 was employed as the baseline model for comparison, and the results are detailed in Table 1.

(1) Effects of LSKNet: a comparative analysis between YOLOv8 and YOLOv8+LSKNet highlights the efficacy of integrating LSKNet. Notably, the addition of LSKNet leads to a notable enhancement in model accuracy. The metrics mAP@0.5, precision, and recall show improvements of 2.26%, 4.60%, and 4.27%, respectively. This substantiates that LSKNet indeed contributes to improved model performance, attributed to its dynamic adaptation of a large spatial receptive field.

(2) Effects of GSConv: a comparison between YOLOv8 and YOLOv8 + GSConv reveals that the incorporation of GSConv contributes to a reduction in computational costs, with GFLOPS and Params experiencing reductions of 45.89% and 53.15%, respectively. Simultaneously, feature extraction capabilities receive a modest boost, as evidenced by increases of 0.18% in mAP@0.5, 3.80% in precision, and 4.42% in recall. While the GSConv module was integrated into the neck with careful consideration, it was deliberately excluded from the backbone to prevent an excessive presence of GSConv modules. This decision aimed to circumvent the overcomplication of the network architecture, which could hinder spatial information flow and significantly elongate inference times.

(3) Effects of both together: Xoo-YOLO harmoniously amalgamates the strengths of both LSKNet and GSConv. The result is a model with a 45.43% reduction in parameter count, a 51.39% decrease in computational demand, a 10.96% enhancement in precision, a 4.34% improvement in recall, and a noteworthy 7.80% advancement in mAP@0.5 when compared to YOLOv8.

Collectively, Xoo-YOLO exemplifies a well-rounded synergy between accuracy enhancement and model lightweightness, thus affirming the significance of our proposed enhancements.

## 3.4 Comparative experiment

To validate the advantages of the model, the wild rice bacterial blight dataset, which includes 450 images in the training set, 150 images in the validation set, and 150 images in the test set, was used to evaluate the model's performance in terms of precision, recall, *F*1 score, mAP, speed (FPS), Params, and GFLOPs. All experiments were conducted under identical experimental conditions to ensure a fair comparison. Comparative experiments were conducted on the proposed model with the YOLOv7 and YOLOv7-added Swin Transformer module. The comparison results are shown in Figure 9 and Table 2. The detection performance of the three networks is different, and the mAP@0.5 of the Xoo-YOLO proposed in this paper is 94.95%, which is 9.13% and 6.13% higher than the original YOLOv7 and YOLOv7-added Swin Transformer module.

Furthermore, the Xoo-YOLO model outperforms the other models in various metrics. There are two main reasons contributing to this phenomenon. Firstly, in comparison to the ELAN module in YOLOv7, the C2f module in YOLOv8 incorporates a parallel concatenating operation of the bottleneck module, which allows for more branching quadratic links and thus richer gradient flow information and thus possesses enhanced feature extraction and fusion capabilities. Secondly, the LSKNet module dynamically adjusts the spatial receptive field as needed, effectively mitigating instances of false positives and false negatives. These factors collectively reinforce the efficacy of the Xoo-YOLO model in wild rice bacterial blight detection. In order to further verify the performance of the proposed Xoo-YOLO model, we randomly selected some detection results under different environmental conditions from all testing samples, as shown in Figure 10.

In terms of speed and model size, the Xoo-YOLO model exhibits a significant reduction in parameter count and computational complexity. This reduction is primarily attributed to the introduction of the GSConv module, which has a lower computation intensity. This module effectively accelerates feature fusion while decreasing computational complexity. Additionally, when compared to the YOLOv7 model, the C2f module and SPPF module utilized in YOLOv8 are more lightweight, providing advantages in terms of parameter and computational complexity. Comparative results are presented in Figure 11.

To analyze the performance differences between the proposed oriented box detection method and the original horizontal box

TABLE 1 Comparisons of ablation experiments.

| Models | Precision (%) | Recall (%) | mAP@0.5 (%) | GFLOPS (G) | Params (M) |
|---|---|---|---|---|---|
| YOLOv8 | 87.94 | 75.66 | 87.15 | 54.317 | 24.352 |
| YOLOv8 + LSKNet | 90.20 | 79.93 | 91.75 | 54.662 | 24.786 |
| YOLOv8 + GSConv | 88.12 | 79.46 | 91.57 | 29.347 | 11.404 |
| **YOLOv8 + GSConv + LSKNet (proposed model)** | **97.60** | **80.00** | **94.95** | **29.692** | **11.838** |

Bold values represent the model proposed in this paper, namely the Xoo-YOLO model.

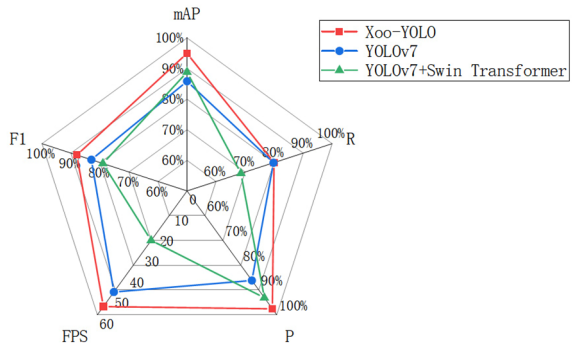



FIGURE 9
Comparison of detection performance between different models.

detection method for wild rice bacterial blight, 750 images were annotated by the horizontal bounding box method and fed into the same network for training. Comparative experiments were conducted on the trained models, and the results are presented in Table 3.

Compared to the original method of using horizontal bounding boxes for detection, the model proposed in this study demonstrates improvements in terms of recall, precision, and mAP. When dealing with wild rice bacterial blight cases characterized by large aspect ratios and varying orientations, the utilization of oriented bounding box detection provides a better fit for the diseases. This approach reduces the influence of the background and facilitates more accurate feature extraction. Conversely, employing horizontal bounding boxes for detection can lead to visual disturbances and result in more pronounced instances of missed and false detections. Such an approach is inadequate for addressing the requirements of detecting wild rice bacterial blight from the UAV's perspective in field conditions. Furthermore, it significantly hampers subsequent research involving disease segmentation and disease resistance identification for wild rice bacterial blight. The comparative results of detection performance using both methods are illustrated in Figure 12.

TABLE 2 Comparison of detection performance of different models.

| Models | Precision (%) | Recall (%) | *F*1 | mAP@0.5 (%) | FPS |
|---|---|---|---|---|---|
| YOLOv7 | 86.15 | 79.81 | 0.83 | 85.82 | 49 |
| YOLOv7 + Swin Transform | 93.13 | 68.59 | 0.79 | 88.82 | 24 |
| **Proposed model** | **97.6** | **80.00** | **0.88** | **94.95** | **56** |

Bold values represent the model proposed in this paper, namely the Xoo-YOLO model.

TABLE 3 Comparison of the horizontal bounding box and oriented bounding box detection performance.

| Models | Precision (%) | Recall (%) | mAP (%) |
|---|---|---|---|
| HBB | 73.3 | 78.5 | 79.5 |
| **Proposed model** | **97.60** | **80.00** | **94.95** |

Bold values represent the model proposed in this paper, namely the Xoo-YOLO model.

## 4 Discussion

Extensive research on rice bacterial blight detection using deep learning has been conducted. However, prior studies by Haque et al. (2022); Jia et al. (2023); Kumar et al. (2023), and Prasomphan (2023) did not fully consider the complexities of field conditions and the diverse angles at which diseases appear under the UAV viewpoint. This study addresses these specific needs. Moreover, there has been a scarcity of studies involving wild rice bacterial blight detection utilizing deep learning methods. To our knowledge, this is pioneering research to detect wild rice bacterial blight under UAV viewpoints in field settings. The results underscore the considerable potential of the Xoo-YOLO model in disease detection.

The advantages of the Xoo-YOLO model are as follows:

1. Efficiency and speed: Xoo-YOLO boasts lightweight characteristics and enhanced processing speed, making it suitable for deployment on UAV or edge devices.
2. Balance of accuracy and efficiency: The model strikes a harmonious equilibrium between lightweight design and detection accuracy, outperforming other common deep learning models in detection accuracy.
3. Rotated bounding box detection: The Xoo-YOLO model introduces a method for detecting wild rice bacterial blight through oriented bounding boxes, leading to more accurate disease detection and localization. This approach minimizes interference caused by excessive background information under UAV viewpoints, thus establishing a robust foundation for subsequent disease segmentation and measurement efforts.

However, the recall of the Xoo-YOLO model for detecting wild rice bacterial blight stands at 80.0%, suggesting room for improvement. This lower recall could be attributed to factors such as wind interference and motion blur caused by UAV propellers, as well as instances of closely clustered diseases leading to missed detections. To address these, future experiments will explore the use of adversarial generative networks. Additionally, future endeavors should place a priority on including a wide range of wild rice varieties in the research. This approach is essential to ensuring the robustness and generalizability of the proposed method.

While Xoo-YOLO has its limitations, it serves as a valuable technical reference for detecting wild rice bacterial blight in field environments under the UAV viewpoint. The application of the Xoo-YOLO model to an intelligent assessment platform for wild rice diseases holds the promise of validating its reliability.

10.3389/fpls.2023.1256545




FIGURE 10
Prediction results of the proposed method. **(A)** Under dense disease conditions. **(B)** Under the conditions of image blurriness generated during the UAV flight collection process. **(C)** Under complex backgrounds such as weeds and debris in the field.

## 5 Conclusion

Wild rice disease detection is a crucial step in screening and cultivating highly disease-resistant rice varieties. To achieve rapid and accurate detection of bacterial blight in wild rice under natural field conditions, this study establishes a dataset for field-based disease detection. Addressing the unique characteristics of detecting wild rice bacterial blight from UAV viewpoints, this research builds upon the YOLOv8 model, introduces enhancements, and proposes the Xoo-YOLO network architecture. This is achieved by incorporating the LSKNet network in the backbone, integrating the GSConv module in the neck, and adopting oriented bounding box detection. These improvements enable real-time automatic detection of bacterial blight in wild rice under UAV perspectives.

Experimental results reveal that the proposed model achieves an impressive mAP@0.5 of 94.95%. It outperforms comparative models in terms of precision, recall, and *F*1 score. The model demonstrates superior computational complexity, parameter, and detection time with values of 29.692 G, 11.838 M, and 17.78 ms, respectively. These improvements compared to classic object detection models like YOLOv7 and YOLOv8 are significant. The model is well-suited for subsequent research focusing on disease segmentation and disease resistance identification in wild rice bacterial blight.

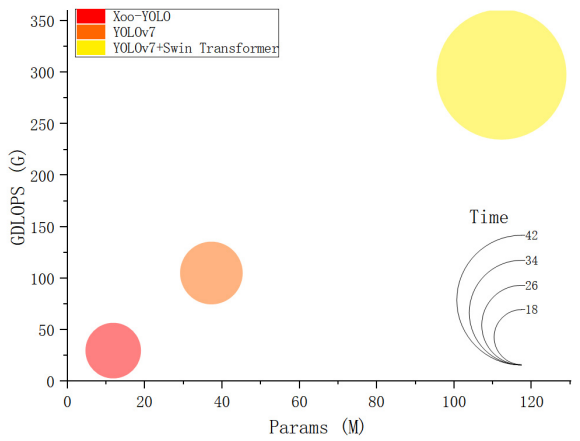



FIGURE 11
Comparison of different models in terms of computational complexity, parameter, and detection time.

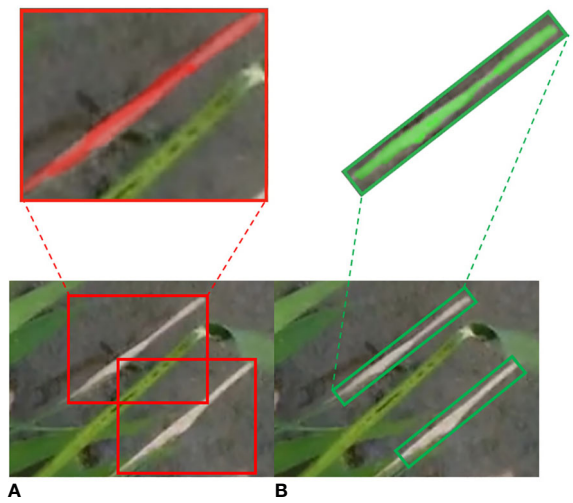



FIGURE 12
Comparison of detection performance between horizontal bounding boxes (HBB) and oriented bounding boxes (OBB). **(A)** HBB. **(B)** OBB.

Considering the requirements of disease resistance identification standards, factors beyond disease detection, such as disease length, need integration with other models like disease segmentation. This study solely focuses on disease detection. However, in future work, we plan to delve into disease segmentation and disease resistance identification to enhance the efficiency of disease resistance gene exploration in wild rice.

## Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

## Author contributions

PP: Conceptualization, Methodology, Project administration, Software, Validation, Writing – original draft, Writing – review & editing. WG: Investigation, Methodology, Validation, Writing – review & editing. XZ: Methodology, Resources, Supervision, Writing – review & editing. LH: Methodology, Resources, Supervision, Writing – review & editing. GZ: Methodology, Resources, Supervision, Writing – review & editing. JZ: Conceptualization, Methodology, Resources, Supervision, Writing – original draft, Writing – review & editing.

## Funding

The authors declare financial support was received for the research, authorship, and/or publication of this article. This research was funded by the National Natural Science Foundation of China (No. 31971792, No. 32160421), the Project of Sanya Yazhou Bay Science and Technology City (No. SCKJ-JYRC-2023-45), the National Key Research and Development Program of China (No. 2022YFF0711805), the Innovation Project of the Chinese Academy of Agricultural Sciences (No. CAAS-ASTIP-2023-AII, No. ZDXM23011), the Special Fund of Chinese Central Government for Basic Scientific Research Operations in Commonweal Research Institutes (No. JBYW-AII-2023-06, No.Y2022XK24, No. Y2022QC17, No. JBYW-AII-2022-14), and the Nanfan special project, CAAS Grant Nos. YBXM2312, YDLH01, YDLH07, and YBXM10.

## Acknowledgments

Appreciations are given to the editors and reviewers of the journal Frontiers in Plant Science.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

Administration, A.P.M.S. (2017). *The Technique Rules for Identification Rice Varieties Resistance Against Bacterial Blight (Xanthomonas oryzae pv. oryzae)*. Available at: https://dbba.sacinfo.org.cn/stdDetail/50e3a8a7d337f3f78f316304f9835ae5a0b76a25d6355f3b062aa69e3eb2e047 (Accessed 9 July 2023).

cgvict (2017) *roLabelImg [Online]. GitHub*. Available at: https://github.com/cgvict/roLabelImg (Accessed 1 April 2023).

Chen, X., Liu, P., Mei, L., He, X., Chen, L., Liu, H., et al. (2021). Xa7, a new executor R gene that confers durable and broad-spectrum resistance to bacterial blight disease in rice. *Plant Commun.* 2 (3), 100143. doi: 10.1016/j.xplc.2021.100143

Dai, G., and Fan, J. (2022). An Industrial-Grade Solution for Crop Disease Image Detection Tasks. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.921057

Dai, G., Hu, L., and Fan, J. (2022). DA-ActNN-YOLOV5: Hybrid YOLO v5 Model with Data Augmentation and Activation of Compression Mechanism for Potato Disease Identification. *Computational Intelligence and Neuroscience* 2022, 6114061. doi: 10.1155/2022/6114061

Fan, W., Liu, Z., Yun, Y., Tang, Q., Zhou, S., Xiao, X., et al. (2023). Collection and Preliminary Identification of Germplasm Resources Resistant to Bacterial Blight of Wild Rice from Hainan Province. *J. Plant Genet. Resour.* 24 (01), 117–125. doi: 10.13430/j.cnki.jpgr.20220830002

Ershadul Haque, M., Rahman, A., Junaeid, I., Hoque, S. U., and Paul, M. (2022). Rice leaf disease classification and detection using yolov5. *arXiv: 2209.01579*. doi: 10.48550/arXiv.2209.01579

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv: 1704.04861*. doi: 10.48550/arXiv.1704.04861

Jia, L., Wang, T., Chen, Y., Zang, Y., Li, X., Shi, H., et al. (2023). MobileNet-CA-YOLO: An Improved YOLOv7 Based on the MobileNetV3 and Attention Mechanism for Rice Pests and Diseases Detection. *Agriculture* 13, 7, 1285. doi: 10.3390/agriculture13071285

Khan, F., Zafar, N., Tahir, M. N., Aqib, M., Waheed, H., and Haroon, Z. (2023). A mobile-based system for maize plant leaf disease detection and classification using deep learning. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1079366

Kumar, V. S., Jaganathan, M., Viswanathan, A., Umamaheswari, M., and Vignesh, J.J.E.R.C. (2023). Rice leaf disease detection based on bidirectional feature attention pyramid network with YOLO v5 model. *Environmental Research Communications* 5 (6), 065014. doi: 10.1088/2515-7620/acdece

Lee, S., Arora, A. S., and Yun, C. M. (2022). Detecting strawberry diseases and pest infections in the very early stage with an ensemble deep-learning model. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.991134

Li, Y., Hou, Q., Zheng, Z., Cheng, M.-M., Yang, J., and Li, X. (2023). Large Selective Kernel Network for Remote Sensing Object Detection. *arXiv: 2303.09030*. doi: 10.48550/arXiv.2303.09030

Li, H., Li, J., Wei, H., Liu, Z., Zhan, Z., and Ren, Q. (2022). Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles. *arXiv: 2206.02424*. doi: 10.48550/arXiv.2206.02424

Li, C., Luo, B., Hong, H., Su, X., Wang, Y., Liu, J., et al. (2020). Object detection based on global-local saliency constraint in aerial images. *Remote Sensing* 12 (9), 1435. doi: 10.3390/rs12091435

Liao, M., Zhu, Z., Shi, B., Xia, G.-s., and Bai, X. (2018). Rotation-sensitive regression for oriented scene text detection. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5909–5918. doi: 10.1109/CVPR.2018.00619

Liu, J., and Wang, X. (2021). Plant diseases and pests detection based on deep learning: a review. *Plant Methods* 17 (1), 22. doi: 10.1186/s13007-021-00722-9

Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., et al. (2018). Arbitrary-oriented scene text detection *via* rotation proposals. *IEEE Transactions on Multimedia* 20 (11), 3111–3122. doi: 10.1109/TMM.2018.2818020

Pan, P., Jianhua, Z., Xiaoming, Z., Guomin, Z., Lin, H., Quan, F., et al. (2023). Research progress of deep learning in intelligent identification of disease resistance of crops and their related species. *Acta Agriculturae Zhejiangensis* 35 (08), 1993–2012. doi: 10.3969/j.issn. 1004-1524. 20236105

Prasomphan, S. (2023). Rice Bacterial Infection Detection Using Ensemble Technique on Unmanned Aerial Vehicles Images. *Comput. Syst. Sci. Eng.* 44 (2), 991–1007. doi: 10.32604/csse.2023.025452

Qiu, R.-Z., Chen, S.-P., Chi, M.-X., Wang, R.-B., Huang, T., Fan, G.-C., et al. (2022). An automatic identification system for citrus greening disease (Huanglongbing) using a YOLO convolutional neural network. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.1002606

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (6), 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Shao, W., Fan, G., He, Z., and Song, Z. (2023). Phenotypic plasticity and local adaptation of Oryza rufipogon revealed by common garden trials. *Biodiversity Sci.* 31 (03), 109–116. doi: 10.17520/biods.2022311

Shao, M., Zhang, J., Fen, Q., Chai, X., Zhang, N., and Zhang, W. (2022). Research progress of deep learning in detection and recognition of plant leaf diseases. *Smart Agric.* 4 (1), 29–46. doi: 10.12133/j.smartag.SA202202005

Tang, Z., Liu, Q., Xiao, X., and Niu, X. (2017). Selection of Optimized Candidate Reference Genes for Qrt-pcr Normalization in Rice during Xanthomonas Oryzae Pv. Oryzae Infection. *Mol. Plant Breed.* 15 (1), 300–306. doi: 10.13271/j.mpb.015.000300

Terven, J., and Cordova-Esparza, D. (2023). A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *arXiv*. 2304.00501 doi: 10.48550/arXiv.2304.00501

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2023). "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7464–7475. doi: 10.1109/CVPR52729.2023.00721

Xu, C., Wang, Z., Du, R., Li, Y., Li, D., Chen, Y., et al. (2023). A method for detecting uneaten feed based on improved YOLOv5. *Computers and Electronics in Agriculture* 212, 108101. doi: 10.1016/j.compag.2023.108101

Xu, Z., Xu, X., Gong, Q., Li, Z., Li, Y., Wang, S., et al. (2019). Engineering broad-spectrum bacterial blight resistance by simultaneously disrupting variable TALE-binding elements of multiple susceptibility genes in rice. *Mol. Plant* 12 (11), 1434–1446. doi: 10.1016/j.molp.2019.08.006

Yang, X., Yang, X., Yang, J., Ming, Q., Wang, W., Tian, Q., et al. (2021). Learning high-precision bounding box for rotated object detection via kullback-leibler divergence. *Advances in Neural Information Processing Systems* 34, 18381–18394. doi: 10.48550/arXiv.2106.01883

Yuan, L. (1986). Hybrid Rice in China. *Chin. J. Rice Sci.* 1 (1), 8–18.

Yun, Y., and Han, Y. (2014). Research Progress on Identifying and Utilizing Disease-resistance Resources in Chinese Wild Rice. *J. Plant Genet. Resour.* 15 (03), 472–476 +482. doi: 10.13430/j.cnki.jpgr.2014.03.003

Zhang, D.-Y., Luo, H.-S., Cheng, T., Li, W.-F., Zhou, X.-G., Gu, C.-Y., et al. (2023). Enhancing wheat Fusarium head blight detection using rotation Yolo wheat detection network and simple spatial attention network. *Computers and Electronics in Agriculture* 211, 107968. doi: 10.1016/j.compag.2023.107968

Zhang, S., Wang, X., Li, P., Wang, L., Zhu, M., Zhang, H., et al. (2021). "An improved YOLO algorithm for rotated object detection in remote sensing images," in *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC): IEEE*. 840–845. doi: 10.1109/IMCEC51613.2021.9482265

Zhang, X., Zhou, X., Lin, M., and Sun, J. (2018). "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6848–6856. doi: 10.1109/CVPR.2018.00716

Zhou, G., Zhang, W., Chen, A., He, M., and Ma, X. (2019). Rapid detection of rice disease based on FCM-KM and faster R-CNN fusion. *IEEE Access* 7, 143190–143206. doi: 10.1109/IMCEC51613.2021.9482265

frontiers | Frontiers in Plant Science

TYPE Review
PUBLISHED 01 November 2023
DOI 10.3389/fpls.2023.1269371

Check for updates

OPEN ACCESS

EDITED BY
Tonghai Liu,
Tianjin Agricultural University, China

REVIEWED BY
Milind B. Ratnaparkhe,
ICAR Indian Institute of Soybean Research, India
Inam Ullah Khan,
King's College London, United Kingdom

*CORRESPONDENCE
Teng Zhou
zhouteng@hainanu.edu.cn

RECEIVED 29 July 2023
ACCEPTED 18 October 2023
PUBLISHED 01 November 2023

CITATION
Ning H, Liu S, Zhu Q and Zhou T (2023) Convolutional neural network in rice disease recognition: accuracy, speed and lightweight.
*Front. Plant Sci.* 14:1269371.
doi: 10.3389/fpls.2023.1269371

COPYRIGHT
© 2023 Ning, Liu, Zhu and Zhou. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author (s) and the copyright owner (s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# Convolutional neural network in rice disease recognition: accuracy, speed and lightweight

Hongwei Ning[1], Sheng Liu[2], Qifei Zhu[2] and Teng Zhou[3]*

[1]College of Information and Network Engineering, Anhui Science and Technology University, Bengbu, Anhui, China, [2]Information Network Security College, Yunnan Police College, Kunming, Yunnan, China, [3]Mechanical and Electrical Engineering College, Hainan University, Haikou, Hainan, China

There are many rice diseases, which have very serious negative effects on rice growth and final yield. It is very important to identify the categories of rice diseases and control them. In the past, the identification of rice disease types was completely dependent on manual work, which required a high level of human experience. But the method often could not achieve the desired effect, and was difficult to popularize on a large scale. Convolutional neural networks are good at extracting localized features from input data, converting low-level shape and texture features into high-level semantic features. Models trained by convolutional neural network technology based on existing data can extract common features of data and make the framework have generalization ability. Applying ensemble learning or transfer learning techniques to convolutional neural network can further improve the performance of the model. In recent years, convolutional neural network technology has been applied to the automatic recognition of rice diseases, which reduces the manpower burden and ensures the accuracy of recognition. In this paper, the applications of convolutional neural network technology in rice disease recognition are summarized, and the fruitful achievements in rice disease recognition accuracy, speed, and mobile device deployment are described. This paper also elaborates on the lightweighting of convolutional neural networks for real-time applications as well as mobile deployments, and the various improvements in the dataset and model structure to enhance the model recognition performance.

KEYWORDS

**rice disease, convolutional neural network, lightweight, model compression, convolution operation**

## 1 Introduction

In recent years, rice planting has developed quickly and the mechanization degree has been gradually improved (Kabir et al., 2021). However, rice disease has always been a huge obstacle to the further development of rice planting (Azim et al., 2021). The disease has always been an important factor restricting rice growth, high and stable yield. Rice disease

**Abbreviations:** GPU, Graphics Processing Unit; SLR, Single Lens Reflex Camera; YOLO, You Only Look Once; VGG, Visual Geometry Group; R-CNN, Region-based Convolutional Neural Network.

affects the total grain loss of up to 10% to 30% in the world every year (Agrawal and Agrawal, 2020). Therefore, rapid and accurate identification of rice diseases is very important for ensuring rice production and maintaining global food security (Daniya and Vigneshwari, 2022a).

The traditional rice disease recognition generally relies on the experience accumulation of farmers in the actual production process (Yakkundimath et al., 2022). This method has high professional requirements for practitioners, consumes a lot of workforce and costs. But the judgment outcomes are highly biased, with large errors even (Sony, 2019). It is challenging to achieve accurate disease identification, simple to lose the best time for disease preclusion and control. At the same time, it is difficult to meet current needs for real-time monitoring and prediction of a wide range of diseases (Chen J. et al., 2020). Therefore, it is significant to investigate automatic recognition methods of rice typical diseases for early detection, diagnosis, and treatment of rice diseases to reduce loss and increase yield (Jadhav et al., 2021; Jiang et al., 2021).

With the enhancement of computing power of computer hardware and the explosive growth of data, deep learning technology has achieved very good results and been broadly applied in many areas such as speech recognition, image processing, and natural language processing (Domingues et al., 2022; Ning et al., 2022). The convolutional neural network is a very representative deep learning technology (Priyangka and Kumara, 2021). Its performances in computer vision missions such as image semantic segmentation, object detection, and classification recognition are far superior to other traditional methods. It has been extensively used in face recognition, automatic driving, and other engineering fields (Brahimi et al., 2018; Sethy et al., 2020a).

In terms of rice disease recognition, convolutional neural networks have also obtained very good results and been widely utilized (Li et al., 2020; Sharma et al., 2020) (Figure 1). In Figure 1, the CNN backbone is a common convolutional neural network such as ResNet50,VGG16 etc. which is mainly responsible for extracting various features from images. The RPN searches for candidate regions where rice diseases may be present from the extracted features by traversing them at a time. The ROI pooling converts all candidate regions into a format that is uniform in length and width. The final fully connected layer unifies all the candidate regions and outputs the type of rice disease and its location in the image. With deepening application, the recognition of rice disease has put forward new requirements for convolutional neural networks. In the paper, the achievements of convolutional neural networks in rice disease recognition in recent years are summarized, and the recognition accuracy optimization, recognition speed improvement, and lightweight revision are discussed respectively. The contributions of the paper are the following:

- Theory of convolutional neural networks is explained detailed, especially various measures that could reduce the complexity of convolutional neural networks.
- Methods that can improve the accuracy of rice disease identification based on convolutional neural networks are summarized.
- Existing research results for speeding up convolutional neural networks to identify rice diseases are summarized.
- The deployment of convolutional neural networks to mobile devices for real-time rice disease identification is summarized.
- The limitations of rice disease recognition based on convolutional neural networks are given and the future research priorities are envisioned.

The structure of this paper is organized as follows: the first part is the introduction; the second part discusses the basic principles and lightweight measures of convolutional neural networks; the third part summarizes the method of convolutional neural network to improve the recognition accuracy in the process of rice disease

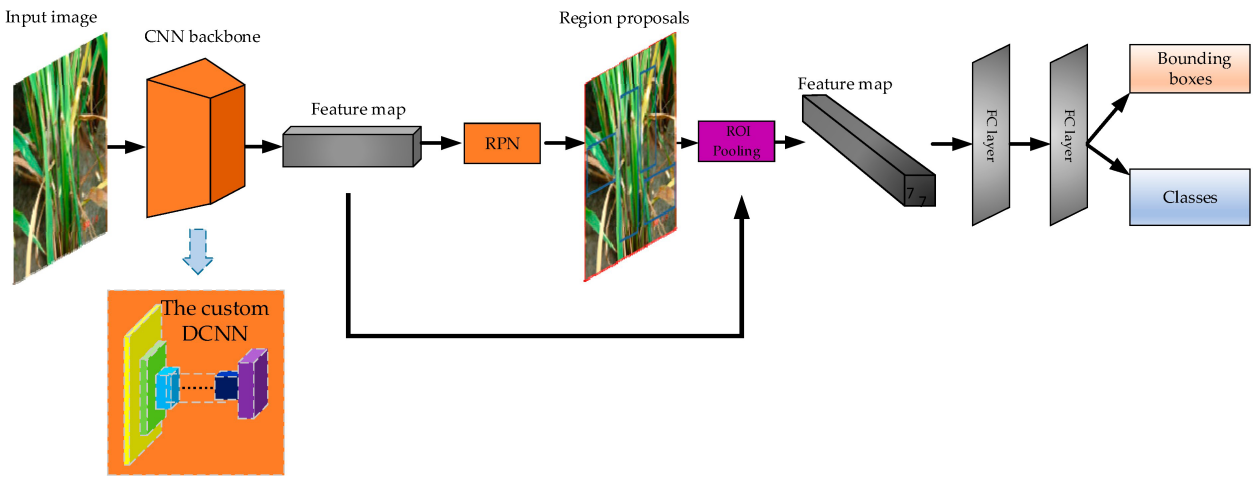



FIGURE 1
Rice disease detection model diagram. (Reprinted from ref. (Li et al., 2020) under the terms of the Creative Commons CC-BY license).

recognition; in the fourth part, the measures to deploy convolutional neural networks on mobile devices after lightweight revision in the process of rice disease recognition are discussed; the fifth part summarizes the approaches to optimize the recognition speed; the sixth part lists limitations of convolutional neural networks applied in rice disease identification and future research directions. Finally, a summary of the paper is presented.

## 2 Theory of convolutional neural network

The convolutional neural network is one of the classical ways of deep learning with multi-layers structure and convolution computation (Gu et al., 2018). It can automatically extract image properties and advanced features that cannot be extracted manually (Yamashita et al., 2018). It has obvious advantages in image recognition and can be applied to large-scale data training (Senan et al., 2020). The lightweight operation of a convolutional neural network is also a research hotspot recently (Haque et al., 2021).

### 2.1 Framework of convolutional neural network

The structure of a convolutional neural network can be split into the convolution layer, pooling layer, and fully connected layer according to different calculation ways (Zhang et al., 2020) (Figure 2). The feature map is generated by extracting local features of input data through the convolution layer. The feature map dimension is reduced by the pooling layer. The processed feature map is input into the fully connected layer and the outcome according to different tasks is output (Lu et al., 2017; N et al., 2021).

Convolution layer. The convolution layer collects local attributes of the training data by convolution operations (Atole and Park, 2018). The input data and weight parameters are combined with the offset value after convolution operations, and the result is the input of the activation function. The size of the final output feature graph is related to the setting of relevant parameters of the convolutional neural network. The formula for calculating convolution is:

$$x_{ij}^{(l)} = f\left(\sum_{m=0}^{k}\sum_{n=0}^{k} x_{i=m,j+n}^{(l-1)} w_{mn}^{(l)} + b^{(l)}\right)$$

$x$ is the image pixel, $k$ is the length or width of the convolution kernel, $w$ is the weight vector, $b$ is the bias, and $f$ is the activation function. The image is input to the convolutional neural network as a data matrix, and the convolution kernel is essentially a local weight matrix as well. The convolution operation is to slide the convolution kernel over the input data in a certain number of steps, multiplying and summing element-by-element for each position to obtain a new 2D feature matrix. The convolution operation can effectively extract the local spatial information in the input data.

Pooling layer. Similar to the principle of convolution operations, pooling is realized by moving the sliding window on the feature graph (Jiang et al., 2020). According to the step size, each move on the feature map will get the presentation value of the region. After pooling operations, the size of the feature map will be reduced. The calculation formula for the pooling operation is the following:

$$x^t = R(\beta^t . P(x^{t-1}) + b^t)$$

$R$ is the activation function, $P$ is the pooling function, $\beta$ is the weight coefficient, and $b$ is the bias. The pooling operation divides the input feature mapping matrix into non-overlapping regions and then performs aggregation of each region by calculating the average or taking the maximum value. The pooling operation aggregates or counts the values within each window by performing a sliding window process on this matrix and outputs them to the next layer as input data. The pooling operation is usually immediately followed by the convolution operation, which reduces the feature dimensions of the output of the convolutional layer by pooling, effectively preserving the important feature information.

Fully connected layer. The fully connected layer is positioned in the last part of the convolutional neural network. The fully connected mode is adopted to map the two-dimensional feature map into a one-dimensional vector, and finally map to the sample space according to different tasks. The preceding convolution layer

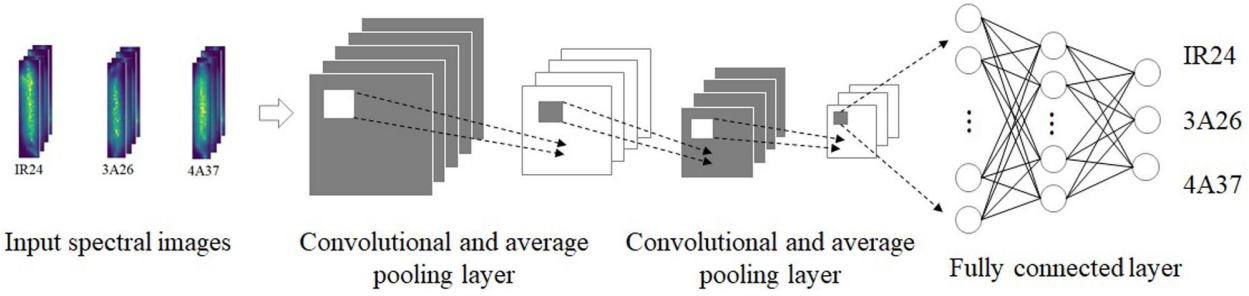



FIGURE 2
A convolutional neural network framework. (Reprinted from ref. (Zhang et al., 2020) under the terms of the Creative Commons CC-BY license).

and pooling layer map the input data to the feature representation space, and then the fully connected layer maps the feature representation space to the sample's label space to achieve the final classification or regression task (Islam et al., 2021).

## 2.2 Lightweight measures of convolutional neural network

Traditional convolutional neural networks have the problem of too many parameters. If the model is improved to obtain better classification performance, a lot of sample sets are needed. However, in practical engineering applications, it is hard to get a lot of sample sets to promote the classification precision. There is an increasing demand for convolutional neural network deployments on mobile terminals, and in-depth lightweight research on convolutional neural networks is required. Under the premise of not reducing model performance, architecture size and the computational amount should be reduced as far as possible to get a balance between performance and overhead (Hui et al., 2020). At present, the lightweight of convolutional neural networks is mainly realized through model compression and adjustment of the convolution operation.

The commonly model compression methods are network pruning, parameter quantization, low-rank approximation, and knowledge distillation.

Network pruning. Pruning, as a classical technique in the field of model compression, has been widely used in the post-processing of various algorithms. Network pruning is an important technique which could reduce network complexity and prevent network overfitting (Yeom et al., 2021). It is widely employed in machine learning and convolutional neural networks (Figure 3). Network pruning can remove redundant connections in convolutional neural networks, decrease model complexity, and reduce the amount of computation. Moreover, it can effectually avoid overfitting and optimize the generalization of the architecture. Network pruning usually has three steps: training connection and measuring the importance of network neurons; removing unimportant neurons; retraining the network and fine-tuning the network (Chen R. et al., 2020).

Parameter quantization. Quantization is operations of replacing all the original parameters with part parameters, which greatly reduces the storage overhead. Parameter quantization, as a common back-end compression technique, can obtain a large reduction in model volume but little performance loss (Zhang and Chung, 2021). The demerit is that the quantified network is fixed, making it hard to change (Figure 4). On the other hand, the generalization of this way is poor, resulting in high maintenance costs. One of the simplest quantization algorithms is scalar quantization (Zhang et al., 2022).

Low-rank approximation. It decomposes the huge dense weight matrix into several small-scale matrices, the original weight matrix can be approximately reconstructed. The operation achieves the purpose of reducing storage and calculation (Lee et al., 2021). The basic computational mode of a convolutional neural network is convolution operations (Figure 5). In the actual implementation,

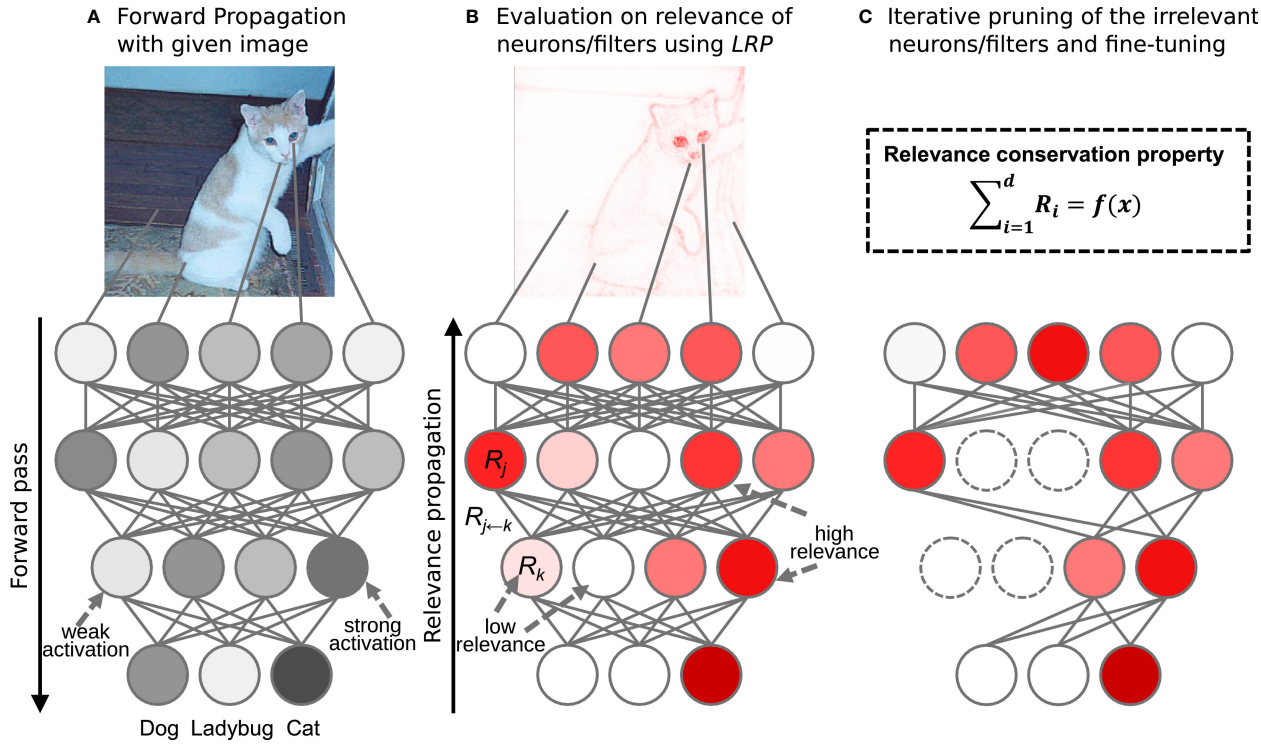



FIGURE 3
Schematic illustration of network pruning. **(A)** Original network; **(B)** Weight measure for each node; **(C)** Network after pruning. (Reprinted from ref. (Yeom et al., 2021) under the terms of the Creative Commons CC-BY license).

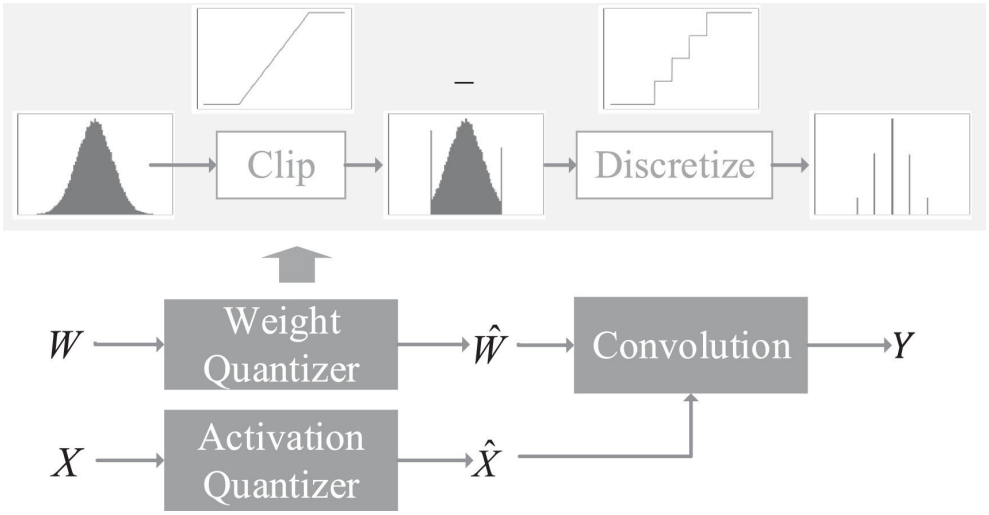



FIGURE 4
Quantization of a convolutional layer. (Reprinted with permission from ref. (Zhang and Chung, 2021) copyright 2021 Elsevier).

the convolution operation is completed by matrix multiplication. However, the weight matrix tends to be dense and large in general, which brings huge overhead in computation and storage. An intuitive idea to solve the problem is that if the dense matrix can be approximately reconstructed through a few smaller matrices, then the storage and computation costs can be reduced effectively (Anvarjon and Kwon, 2020).

Knowledge distillation. It is a transfer learning method, and its target is the convert of the knowledge learned from the complex network framework to the compact small model over certain approaches so that a tiny scheme can also obtain a similar capability as the complex model (Lee et al., 2021; Wang and Du, 2021; Chen W. et al., 2022) (Figure 6). In the framework of knowledge distillation, two basic elements play a decisive role: first, what is "knowledge"? That is, how to extract the knowledge in the model. The second is how to "distillation"? That is, how to complete the task of knowledge transfer (Sahu et al., 2018).

It is an effective method to adjust standard convolution, decrease convolution parameters and speed up convolution operation. Although the model compression takes a good part in the process of convolutional neural network lightweight, its process is too complex, and it usually requires detailed repeated training to achieve a similar performance to the original model. Therefore, some researchers begin to directly design lightweight convolutional neural network frameworks to control the number of parameters and computation by group convolution, depth-wise separable convolution, depth-wise convolution, and pointwise convolution (Sunija et al., 2021).

Group convolution. It divides the dimension of feature channels into several equal parts, then convolves them separately, and piles

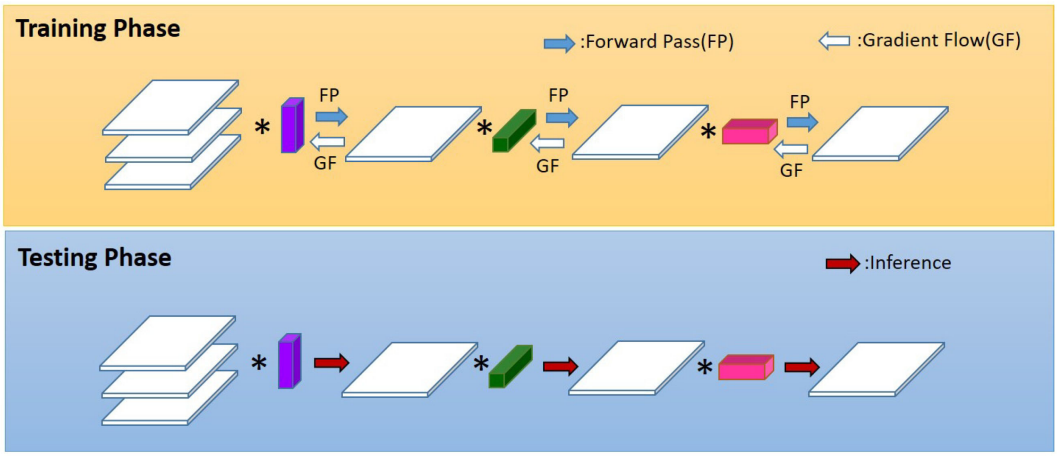



FIGURE 5
Rank-1 approximation. (Reprinted from ref. (Lee et al., 2021) under the terms of the Creative Commons CC-BY license).

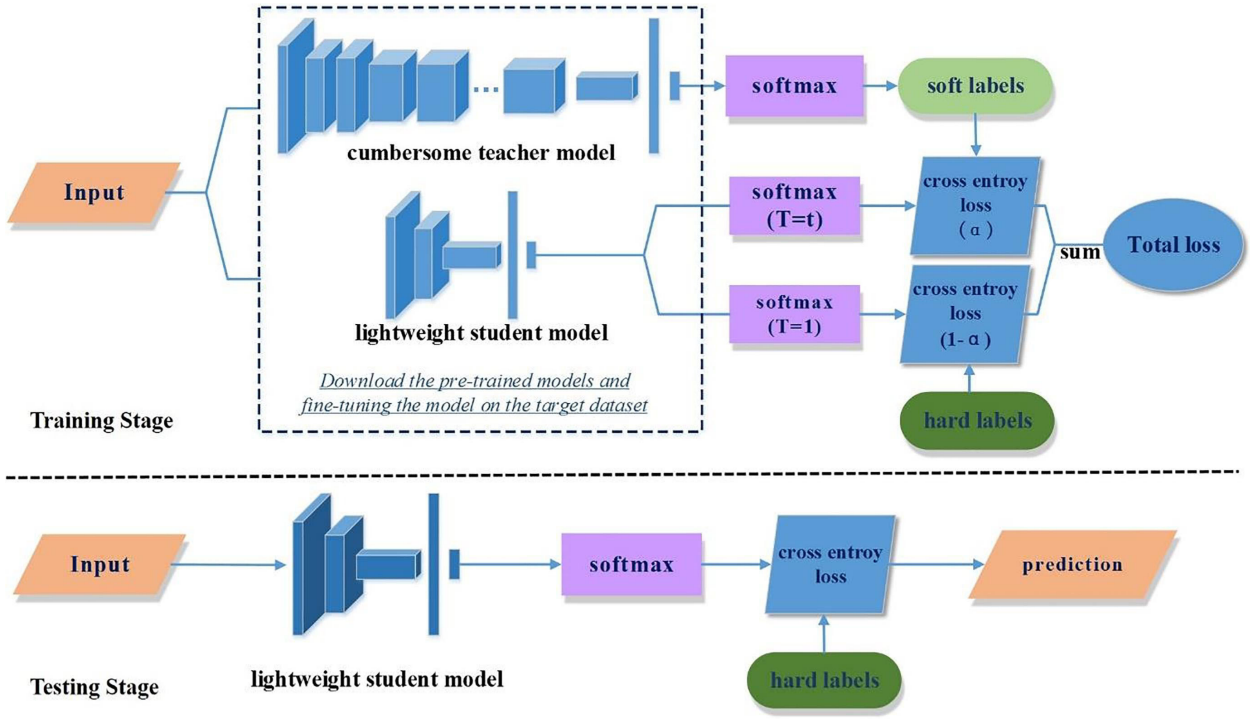



FIGURE 6
An example of knowledge distillation. (Reprinted with permission from ref. (Chen W. et al., 2022) copyright 2022 Elsevier).

up the results. The idea of group convolution is widely used in network design. Besides reducing the number of parameters, it can also be regarded as a structured sparse approach, equivalent to a regularization manner. As the number of filter groups raises, the model parameters decrease, and the framework becomes more efficient. Since the convolution is partitioned into multiple paths and each path can be handled separately through a respective GPU, the architecture can be trained on several GPUs in parallel, and the training speed of the mode is greatly accelerated (Huang et al., 2019).

Depth-wise separable convolution. The merit of depth-wise separable convolution is that the more attributes that need to be extracted, the more parameters can be saved, reducing the amount of computation. Deep-wise separable convolution is actually a kind of decomposable convolution operation, which involves spatial dimension, but also deals with depth dimension (Prottasha and Reza, 2022) (Figure 7). It can be decomposed into two smaller actions: deep convolution and point-by-point convolution.

Depth-wise convolution. It is a kind of packet convolution in which the number of packets is equal to the amount of feature channels, and the convolution kernel is one-to-one corresponding to the channels. Therefore, the depth of the output feature map is the same as that of the input. For the multi-channel characteristic graphs from the previous layer, all of them are first split into feature graphs of a single channel, and single-channel convolution is performed on them respectively and then stacked together. Different from the standard convolution activities, deep-wise convolution splits the convolution kernel into several channels and carries out convolution operations on each channel without changing the depth of the input property image. In this way, the output attribute graph with the same amount of channels as the input feature graph is acquired (Ullah et al., 2021).

Pointwise convolution. The deep-wise convolution only adjusts the dimension of the property graph from the previous layer, while the number of channels does not change, which demands to be modified by pointwise convolution. The pointwise convolution is a 1*1 convolution. Since deep-wise convolution does not merge information between channels, it needs to be applied together with pointwise convolution. It could raise or reduce dimension of feature graphs. The operation of pointwise convolution is similarly to the conventional convolution operation, which is to produce a new attribute map by weighted addition of the property map of the former stage in the depth direction. Each filter exports one feature map, so several channels require multiple filters (Gayathri et al., 2020).

## 3 Convolutional neural network for better recognition accuracy

Accuracy is the basis of all applications. Only with better recognition accuracy can convolutional neural networks be widely utilized in rice disease recognition. Small and scattered rice disease sites in the images can lead to poor identification accuracy. To solve this problem, many researchers have started to improve the models in terms of their feature extraction capability and training methods. To enhance the feature extraction capability of the model, multi-scale features of rice diseases are often extracted from the images

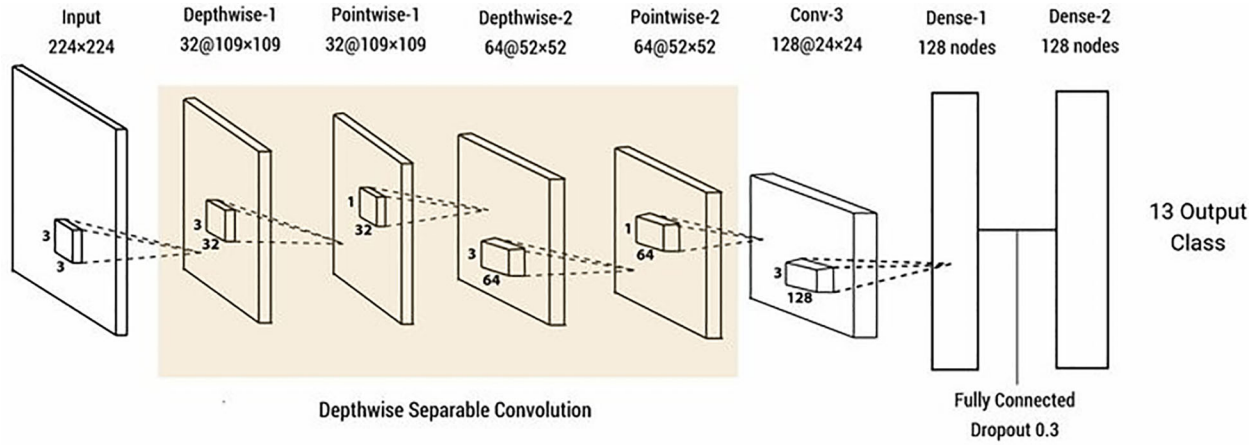



FIGURE 7
Schematic illustration of a convolutional neural network with depth-wise separable convolution. (Reprinted from ref. (Prottasha and Reza, 2022) under the terms of the Creative Commons CC-BY license).

and merged by introducing an attention mechanism. The improvement of the model training method is mainly implemented by using ensemble learning.

By introducing the attention mechanism into a model, it can better understand the correlation between the input data, and improve its prediction ability. Shuai Feng et al. studied the detection of rice blast based on spectral characteristics (Feng S. et al., 2022). The convolutional neural network employed in the method combined the attention mechanism with a residual network to determine the optimal frontal characteristic wavelength. Guided gradient weighted class activation mapped spectral data to guided gradient weighted heat maps and finally determined the appropriate characteristic wavelength. Statistical analysis proved that this way could effectively identify the spectral characteristics of rice diseases and give a high precision recognition. The team also combined spectral features with vegetation index and texture features into the convolutional neural network to identify rice leaf blast (Feng S. et al., 2021). The scheme was adjusted based on a residual network. The spectral images of rice obtained by spectrometers in a paddy field in Shenyang, China, were increased from 145 to 4930 after data enhancement. The experimental comparison with several existing models verified that the neural network with multiple features was much better than the mode with only one feature. Shuo Chen et al. added an attention mechanism and multi-scale feature integration technique to the convolutional neural network to recognize rice diseases (Chen S. et al., 2021). The attention mechanism helped the model to find the key attributes, while the multi-scale feature integration technology could integrate the features of different scales for comprehensive analysis. The dataset consisted of 109 images taken by digital cameras in fields in Xuzhou, China and annotated by the LabelMe software. Data enhancement operations, such as rotation, stretching, sharpening, etc., led to a final dataset of 1,199 images. Experiments confirmed that the precision of rice disease recognition was improved. Mehdhar S. A. M. Al-Gaashani et al. introduced a self-attention mechanism to achieve high accurate rice disease recognition. The attention module embedded kernel attention and implemented contextual information extraction of features in this way. Experiments showed that the model had an accuracy of 98.71%. Afis Julianto et al. compared the accuracy of six typical convolutional neural networks: InceptionV3, ResNet50, InceptionResnetV2, DenseNet201, MobileNet, and EfficientNetB3 for rice disease recognition (Julianto and Sunyoto, 2021). Mendeley Data was used in the dataset, with 5,932 images covering four disease categories: white leaf blight, rice blast, brown spot, and tungro disease. The dataset was enhanced by zooming in, rotating, flipping, etc., and the image amount in the dataset was increased by six times. The experimental results reported that the InceptionResnetV2 network had the highest accuracy, but its training time was also the longest. Md. Mafiul Hasan Matin et al. modified the classical convolutional neural network AlexNet to identify rice diseases (Matin et al., 2020). Their proposed neural network had 5 convolutional layers, 3 fully connected layers, and finally 3 layers of customized network structure for rice disease recognition. The 120 images in the dataset came from Kaggle's online data, and the number of images increased to 900 after data enhancement, involving leaf blight, brown spots, and smut of rice. The experiment verified that the accuracy of this method reached 99%. Wanjie Liang et al. also studied rice disease recognition based on the AlexNet network (Liang et al., 2019). They mainly investigated the automatic identification of rice blast disease. A total of 5,808 images were collected from the Institute of Plant Protection, Jiangsu Academy of Agricultural Sciences. The stochastic gradient descent algorithm was utilized to update the weights in the network during model training. The evaluation results revealed that the rice disease features extracted by this method had a better identification degree than those extracted by traditional manual methods, and the identification effect of rice disease was also better. Based on high-resolution images, Lele Wei et al. detected rice with different health states using the deep neural network YOLOv4 (Wei et al., 2022). All the images were taken by an unmanned aerial vehicle and processed by histogram equalization, color space conversion, and so on. The stochastic gradient

algorithm based on momentum acceleration was employed to train the network mode. Compared with the conventional image segmentation algorithm, the results indicated that this technique was more efficient and stable. Susant Bhujel et al. combined cyclic learning rate fine-tuning and pre-training model in convolutional neural networks to recognize rice diseases (Bhujel and Shakya, 2022). The cyclic learning rate fine-tuning ensured a faster convergence, and the pre-training model could prevent the error signal amplification in the process of model training and speed up training (Figure 8). The original dataset contained 2,092 images from Kaggle's online data, which was augmented to 8,368 images. Experimental data demonstrated that this method was effective in recognition.

Multiscale feature fusion can help a model locate the position of the target more accurately. By fusing features with different sensory fields, the model can find the boundary and detailed information of the target, thus improving the accuracy of target detection and localization. Ching-Ling Wang et al. conducted a series or parallel operation of multiple convolutional neural networks to find the best network structure for rice disease recognition (Wang et al., 2022). Each independent convolutional neural network extracted different features, and these features would be fused in the decision stage and output the final consequence. To further improve the recognition accuracy, they also carried out two-stage data enhancement on the training data. The experimental results indicated that the parallel multi-convolutional neural network structure had the highest recognition accuracy, but also could preserve the micro spot disease feature on rice leaves. He Liu et al. improved the accuracy of rice disease recognition by advance processing of training data and continuous tuning of hyperparameters of convolutional neural networks. All the images used for training the model were standardized to the same size and the dataset was augmented by data enhancement techniques. Parameters such as learning rate, batch size, and number of iterations were continuously adjusted during the model training process, and a framework with a recognition accuracy of 98.64 was finally obtained. T. Daniya et al. proposed a neural network architecture for high precision identification of bacterial leaf blight in rice. To ensure high recognition accuracy, the framework needed to carry out a range of actions on the input images (Daniya and Vigneshwari, 2022b). Firstly, noise suppression, pixel normalization, and segmentation would be carried out on the images. Then, feature extraction to the segmented images and selected effective recognition features was conducted. Finally, the identification result was output. Experimental results reported that this method had higher recognition accuracy than other mainstream ways. Narendra Pal Singh Rathore et al. researched a sequential convolutional neural network to detect diseases of rice and they obtained an accuracy of 99.1% (Rathore and Prasad, 2020). The training model utilized a dataset of 1,000 images, all from Kaggle. All images were compressed to remove redundant pixels. To speed up the training process, images were rotated, shifted, and clipped in the pre-processing stage. This model fully demonstrated the great potential of convolutional neural networks in rice disease recognition. N. V. Raja Reddy Goluguri et al. combined the convolutional neural network with support vector machine, artificial neural network, and long short-term neural network to test the recognition effect of the new model on rice disease (Goluguri et al., 2021). In the process of network parameter training, particle swarm optimization algorithm, artificial fish swarm algorithm, and efficient artificial fish swarm algorithm were utilized to optimize the weight of each neuron. The experimental data validated that the method optimized by the efficient artificial fish swarm algorithm combined with the convolutional neural network and the long short-term memory neural network had the highest accuracy in identifying rice diseases, reaching 97.5%. Ancy Stephen et al. performed high-precision rice disease identification by combining a generative adversarial network with a convolutional neural network. All training data was from Kaggle and uniformed as 224*224*3. The texture, color,

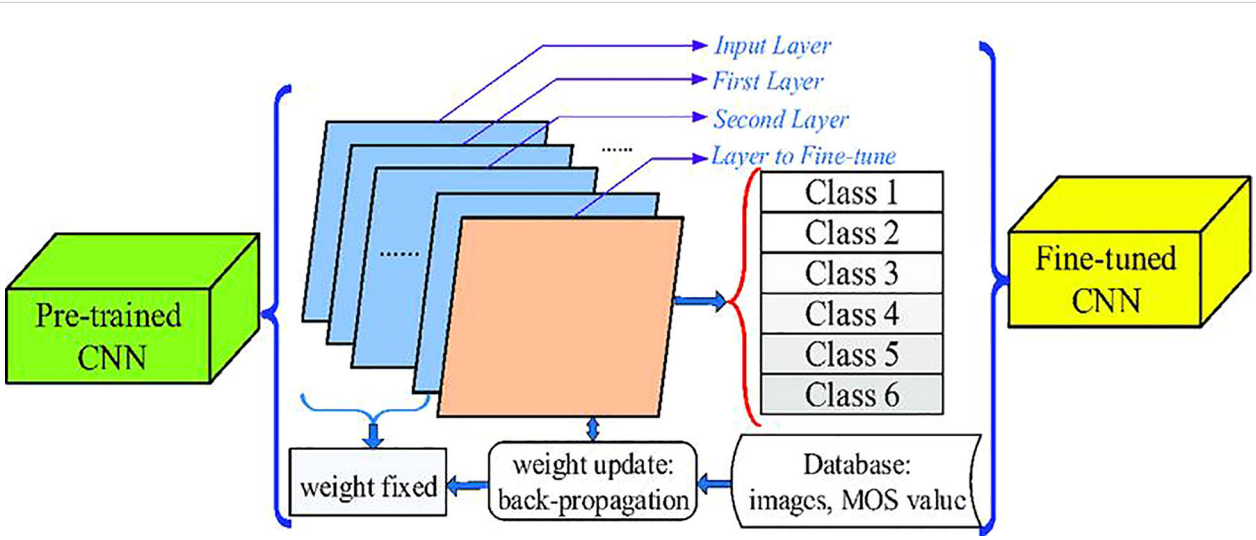



FIGURE 8
Fine-tuned CNN. (Reprinted from ref. (Bhujel and Shakya, 2022) under the terms of the Creative Commons CC-BY license).

and shape features in the target area were extracted in both grayscale and color values. The improved backtracking search algorithm then optimized the layers and nodes for generative adversarial neural network. The model achieved a final recognition accuracy of 98.7%.

Through transfer learning, a model can borrow features, relationships, or patterns from the source domain and apply them to the target task, thereby improving the model's performance and generalization on the target task. Sudhesh K.M et al. applied transfer learning technology and dynamic pattern recognition decomposition method to identify rice diseases (K.M et al., 2023). Among multiple convolutional neural networks trained based on 3416 images, the DenseNet121 network had the best recognition effect on rice disease. Training other machine learning algorithms with the rice disease features extracted by the DenseNet121 network could often achieve better recognition results. In addition, dynamic pattern decomposition driven by an attention mechanism was added to the architecture to locate rice disease areas faster and more accurately. Experimental data showed that the recognition accuracy of this method reached 94.33%. Debaniranjan Mohapatra et al. applied the transfer learning technique to train the AlexNet network to recognize rice diseases (Mohapatra and Das, 2023). The dataset, which included 1,732 images from Kaggle's online data, covered three rice diseases: rice leaf blight, brown spot, and smut. Since the AlexNet network had been trained in advance, the number of pictures in the rice disease dataset was much lower. Experiments data confirmed that the rice disease recognition accuracy of the method was up to 98.8%. Ghazanfar Latif et al. modified the convolutional neural network VGG-19 to identify rice diseases (Latif et al., 2022). The VGG-19 network had been trained on other datasets, and rice disease identification could be carried out just by fine-tuning parameters, which could not only accelerate the model training speed, but also ensure the effect. If there was too little data in the dataset, it was easy to overfit the framework. Therefore, the authors enhanced the images in the dataset through fine-tuning to add images, thus the effect of model training was ensured (Figure 9). In Figure 9, A rice disease dataset originally contained a small number of images, but it was expanded through data enhancement techniques such as rotation, scaling, and panning to bring the number of images within the dataset to 2,465. A portion of the data was used to train the convolutional neural network and other part of the data was used to validate the performance of the trained model. For example, there were 371 images of healthy rice, of which 297 images were used to train the network model while the remaining 74 images were used to validate the performance of the network model. The recognition results demonstrated that the method was more powerful than other existing ways.

## 4 Convolutional neural network for lightweight recognition

Rice disease recognition based on convolutional neural networks has obtained very good consequences in the laboratory, and the accuracy is particularly high. However, so far, the application in the process of rice cultivation is still relatively small. One of the most important reasons is that the convolutional neural network employed in the recognition process is still too complex, has too many parameters and high requirements on hardware. Therefore, further lightweight research on convolutional neural networks is needed to make it run on mobile devices such as smartphones.

It is very effective to process images through the deep neural network model to complete the classification and regression tasks. However, the parameters of the deep neural network model are generally tens of millions, even hundreds of millions. Such a complex model places high demands on both hardware and software computing resources during training and deployment.

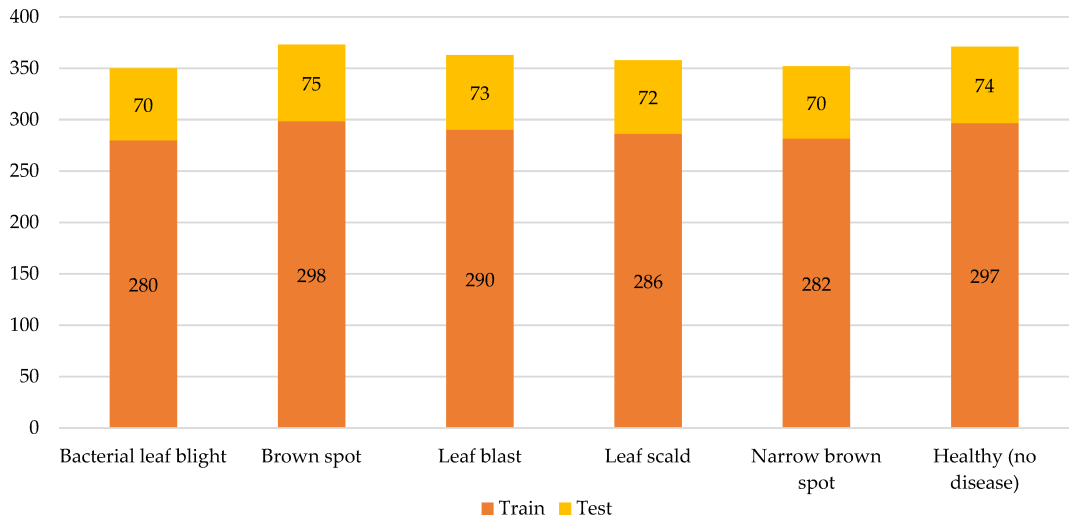



FIGURE 9
Distribution of training images and validation images of each rice disease and healthy rice leaf in the dataset. (Reprinted from ref. (Latif et al., 2022) under the terms of the Creative Commons CC-BY license).

This is the reason that there are fewer examples of neural network technology application of rice disease identification. Currently, the lightweight operations of neural network models in rice disease identification process are achieved by adjusting the neural network structure.

Igor V. Arinichev et al. tested the performance of four relatively lightweight classical convolutional neural networks for rice disease identification: GoogleNet, ResNet-18, SqueezeNet-1.0, and DenseNet-121 (Arinichev et al., 2021). AlexNet and VGG networks were usually not considered in lightweight operations due to their complex structure and large number of parameters. The dataset contained 4,287 images related to brown spots and leaf blast in rice. The test results showed that the accuracy of these models all reached more than 95%. This indicated that convolutional neural networks played a very important role in rice disease recognition, and lightweight networks could also be deployed on mobile devices. Md. Sazzadul Islam Prottasha et al. proposed a relatively lightweight neural network with 2.4 million parameters to recognize rice diseases (Prottasha and Reza, 2022). To ensure the recognition effect under the premise of limited parameters, the Adam algorithm was applied to optimize and adjust parameters in the process of model training. The images in the dataset were from multiple rice fields in Bangladesh and the total number of images enhanced by the original data reached 13415. Experiments revealed that although the size of the model was smaller than the existing 8 most advanced convolutional neural networks, it still had a good recognition effect.

Yibin Wang et al. added the attention mechanism and Bayesian optimization algorithm to a convolutional neural network to reduce its volume and finally realized the application of rice disease recognition on mobile devices (Wang et al., 2021). The attention mechanism allowed the convolutional neural network to understand long distance information within the mode. Bayesian optimization applied a posterior function to decrease the parameters of the convolutional neural network. The dataset contained 2370 pieces of data, including both diseased and healthy rice samples. The extracted features were evaluated visually based on activation mapping and filter visualization techniques. Cross-validation classification experiments showed that the precision of the architecture achieved 94.85%. Chowdhury R. Rahman et al. presented a two-stage mini-convolutional neural network for offline recognition of rice diseases (Rahman et al., 2020). Compared with hundreds of millions of parameters in traditional convolutional neural networks, the network only needed 800,000 parameters, which made the memory efficiency very high and suitable for application in mobile scenarios. A total of 1426 images were included in the dataset for the training of the framework, involving five rice disease categories: pseudosmut, smut, sheath blight, health, and others. All the images were taken with four different types of cameras in different situations, such as summer and winter. Experiments indicated that the model achieved a good balance between memory efficiency and recognition accuracy. Junde Chen et al. designed a convolutional neural network called MobInc-Net for rice disease recognition (Chen J. et al., 2022). To reduce the complexity of the model, the original convolution methods were replaced by deep convolution and point convolution. A total of 1,000 images were collected into the dataset, covering 12 categories, through live photography and crawling on the web. The two-stage transfer learning technique used in mode training further developed the function of the method. Experimental data verified that the procedure had satisfactory efficiency and accuracy even in the complex background of rice disease recognition. The team also studied another outstanding lightweight convolutional neural network to identify minor lesions of rice diseases (Chen J.et al., 2021). This scheme belonged to the category of transfer learning. Training was started on a convolutional neural network that had been trained on ImageNet, which not only reduced the training cost but also improved the training effect. In addition, the framework also introduced an attention module, which could extract more critical details during feature extraction. The dataset consisted of 1,100 images, 660 of which were got online and other 440 were taken in paddy fields. Experiments based on both public and local datasets demonstrated the validity and reliability of the proposed method.

Poornima Singh Thakur et al. investigated a lightweight convolutional neural network with 7 convolution layers for rice disease recognition named VGG-ICNN (Thakur et al., 2023). The network had about 6 million parameters, much less than traditional convolutional neural networks, but its multi-scale feature extraction ability was still very good. The mode had been tested on several public datasets such as PlantVillage, and the results validated that it had a very good recognition effect. The network not only had a good recognition effect on rice diseases but also had perfect performance on maize, apple, and other plant diseases. The performance was also as well as other lightweight convolutional neural networks such as EfficientNet B0. Changguang Feng et al. incorporated deep feature extraction and the attention mechanism into a convolutional neural network to recognize rice blast disease (Feng C. et al., 2022). The depth feature extraction module could simultaneously extract the deep and shallow features of rice blast, and the attention mechanism module could achieve the lightweight goal of this neural network. The dataset contained 800 images, selected from the open dataset, and annotated with the LabelMe software (Figure 10). Figure 10 showed an image which contained rice disease labeled by the software LableMe. The background in the labeled images was standardized to black, the rice leaves were standardized to red, and the portion that was standardized to green represented the location of the lesion. The experiment showed that the method had good universality ability, strong anti-interference ability, and achieved a good balance between recognition speed and accuracy. At the same time, the method supplied a good reference for the deployment of rice disease recognition programs on the Internet of Things and mobile devices.

Ruoling Deng et al. developed a deep learning program based on a convolutional neural network that could run on a smartphone and identify rice diseases (Figure 11). The method integrated several sub-modules to distinguish different rice diseases as much as possible (Deng et al., 2021). The training dataset had 33,026 images covering six rice diseases: leaf blast, pseudosmut, neck blast, sheath blight, bacterial stripe, and brown spot. Instead of starting from scratch, a mature scheme already trained on ImageNet was taken to the dataset for training, greatly reducing



FIGURE 10
Original images and labeled images: the black part means background, the red part means healthy rice leaf, while the green part means rice disease spot. (Reprinted from ref. (Feng C. et al., 2022) under the terms of the Creative Commons CC-BY license).

the training time. The test verified that the recognition accuracy of the model could still reach 95%. Dengshan Li et al. used deep learning techniques to examine rice pests and diseases in the videos (Li et al., 2020). The video was first divided into frames, which conducted rice diseases detectionand then resynthesized into video. The dataset contained 5,320 photos taken by mobile phones in rice paddies in several Chinese provinces, which related to rice grain blight, borers, and brown spots. In the experiment, the framework had achieved satisfactory results and the recognition results of the unprocessed original video were also very good, even though the video was fuzzy and the recognized object was irregular. Jiapeng Cui et al. presented an optimized lightweight convolutional neural network to identify rice diseases, which could identify not only the species of rice diseases, the location of rice diseases (Cui and Tan, 2023). They transposed convolution upsampling in the neural network structure and expanded convolution downsampling to achieve better extraction of rice disease features. The team utilized a digital camera to capture 500 images about rice leaf blight, stripe blight, white spot, stripe blight, and leaf blast. 2,500 images formed the dataset for training the neural network. The results revealed that the method was effective in identifying the five rice diseases.

## 5 Convolutional neural network for higher recognition speed

It is greatly significant to improve the speed of rice disease recognition and realize real-time applications. A lot of researchers have done much work on this aspect and have achieved good results. Rice disease identification based on specific types of images and the use of specific algorithms in the identification process can improve the speed of rice disease identification.

Many studies have demonstrated that identification based on rice disease images obtained by hyperspectral techniques and terahertz imaging is often faster. Lei Feng et al. introduced hyperspectral imaging technology and transfer learning into the process of rice disease detection and realized rapid recognition (Feng L. et al., 2021). Hyperspectral imaging was used to obtain images of rice diseases (Figure 12). The convolutional neural network trained on the original dataset was adjusted and utilized for rice disease recognition. Methods of adjustment included fine-tuning, depth-dependent alignment, and depth-domain obfuscation. The results showed that the model had high detection efficiency and generalization performance, and the scheme could detect different kinds of rice disease. Jinnuo Zhang et al. studied white leaf blight at the stage of rice breeding, using terahertz imaging and near-infrared spectral imaging (Zhang et al., 2020). The recognition models of one-dimensional spectral image and two-dimensional spectral image were respectively constructed with a convolutional neural network. One-dimensional spectral images contained spectral information of samples, while two-dimensional spectral images included both spectral information and spatial information of samples. To better investigate the recognition process, the team also performed a visual analysis. They finally confirmed that the combination of terahertz absorption spectra and convolutional neural network could perform the detection quickly.

Of course, the time required for identification can also be reduced if the common rice disease images are properly processed. Santosh Kumar Upadhyay et al. devised a detection architecture with a convolutional neural network to obtain rapid

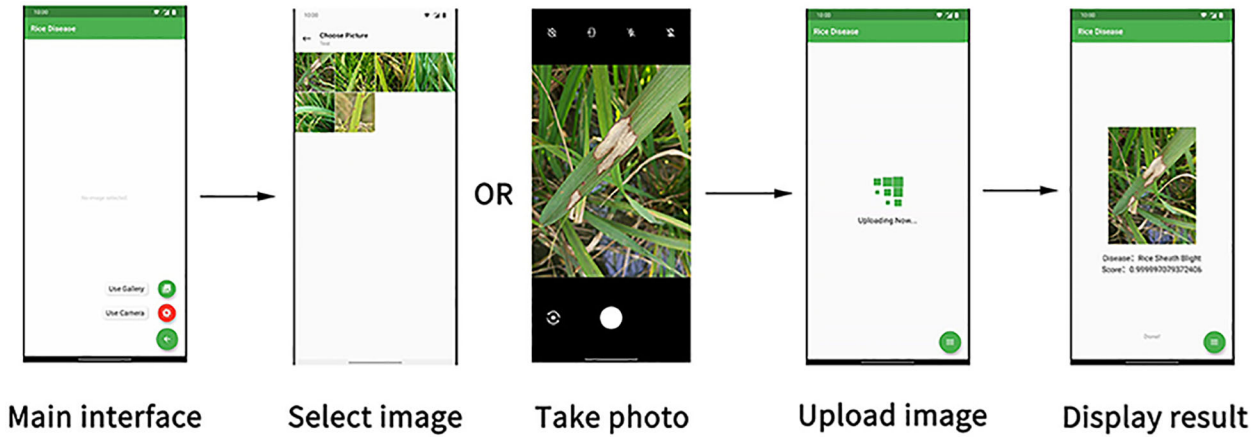



FIGURE 11
The interface of the rice disease recognition APP. (Reprinted from ref. (Deng et al., 2021) under the terms of the Creative Commons CC-BY license).

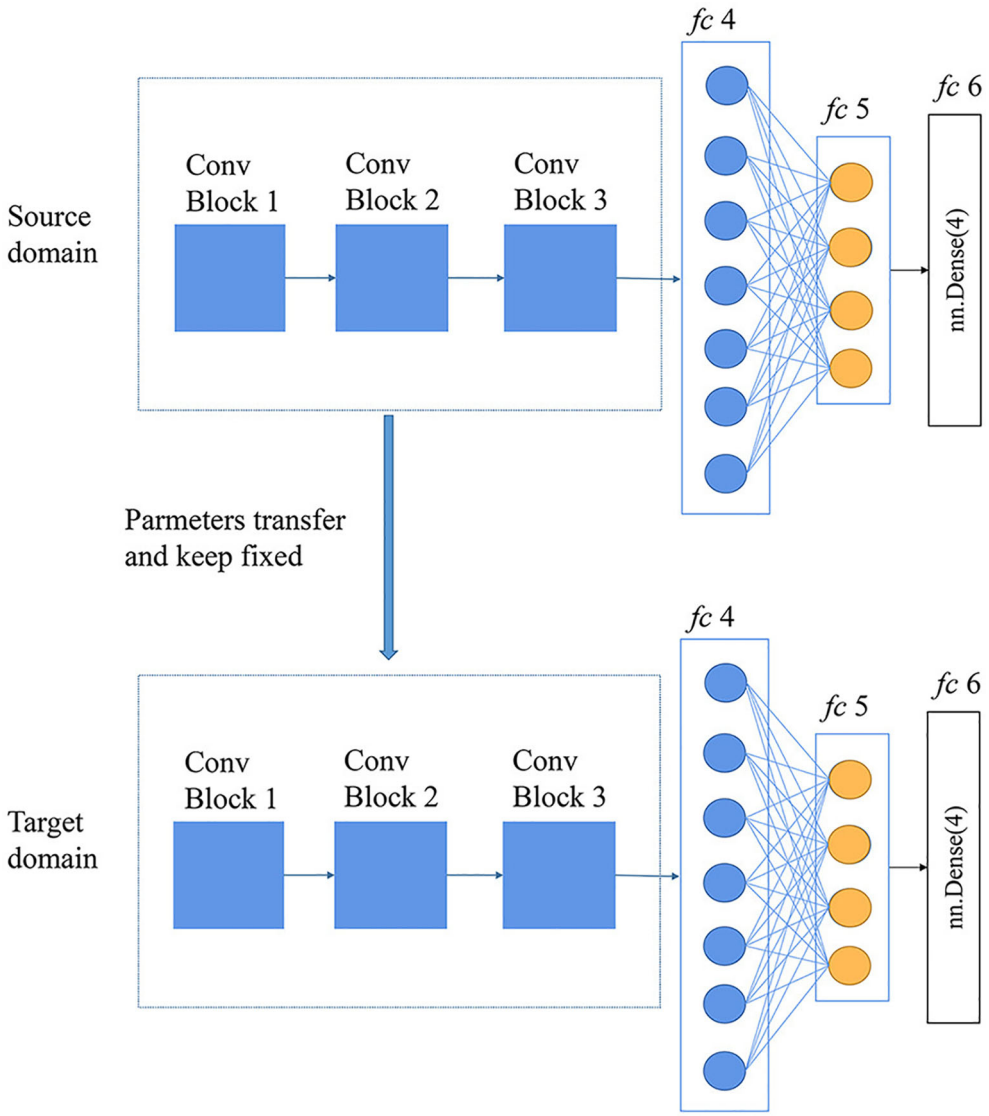



FIGURE 12
Schematic diagram of the fine-tuning algorithm. (Reprinted from ref. (Feng L. et al., 2021) under the terms of the Creative Commons CC-BY license).

diagnosis of rice diseases (Upadhyay and Kumar, 2022). From the input image to output result, the model contained 22 layers. Three rice diseases could be identified by the method: leaf smut, black spot, and leaf blight. Each disease had 4000 pictures in the dataset, of which 3200 pieces were the training set and 800 pieces were the verification set. All images were adjusted to a size of 64 by 64, then the background noise was removed and then grayed, and binarized. The advantage of the operations was that fast identification was guaranteed. The results revealed that the method was fast and effective, and the accuracy gained 99.7%.

Researchers also found that the recognition model based on the Faster R-CNN architecture was better than other algorithmic models in terms of rice disease recognition speed. An area based convolutional neural network was proposed by Bifta Sama Bari et al. to achieve real-time recognition of rice disease (Bari et al., 2021). The enhancement of the regional proposal network improved the detection speed of rice disease greatly. The dataset included both public data on the Internet and photos taken by the authors in the field, and there were 2400 pictures in the dataset. Augmentations to the images in the dataset further promoted the recognition speed and efficiency of the architecture. The results showed that the framework could realize real-time recognition of common rice diseases with very high accuracy. In addition, it could recognize healthy rice leaves. Prabira Kumar Sethy et al. reported a mode to recognize false smut of rice which combined a region proposal network and a convolutional neural network (Sethy et al., 2020b). The region proposal network could assist the model quickly locating the area of disease in the image to speed up the identification activities. All the images in the dataset were obtained by the camera on the smartphone and annotated in the MATLAB software. The test confirmed that the algorithm was effective. But the test may be invalid in some cases. The reason for invalid detection may be that the model provided multiple proposal regions, therefore the system performance needed to be further optimized. Taha Hussain et al. designed a rapid rice disease detection framework based on a convolutional neural network (Hussain et al., 2021). To improve the detection speed, the maximum pooling strategy was adopted in the pooling layer. The 2000 training images and 589 verification images in the dataset were captured by SLRS. All the images were processed into regular patterns and the color images were converted to gray level to ensure the speed of training and recognition. The experimental consequences demonstrated that the framework could complete the recognition procedure in only 0.05 seconds when the recognition accuracy reached 95%, which was very fast.

# 6 Limitations and prospect

Although convolutional neural networks have achieved a lot of results in rice disease recognition, there are still some problems that restrict the development of this research. It is expected that the key work of the research in the future will focus on solving these difficulties.

## 6.1 Limitations of convolutional neural network in rice disease recognition

Although rice disease recognition based on convolutional neural networks has achieved good results, it still faces the problems of excessive training data, limited accuracy, and high computer computation consumption in practical applications (Joseph et al., 2023).

Rice disease recognition based on convolutional neural networks and images requires a large amount of data to train the model and validate its performance. Different types of rice diseases may behave similarly on images, and a large number of images must be available to ensure the effectiveness of recognizing different diseases (Ahmed et al., 2023). The convolutional neural network, as a deep learning model, has a complex structure and contains many parameters, so the demand for data during training is relatively large (Aggarwal et al., 2023; Dogra et al., 2023). Therefore, the rice disease recognition work based on convolutional neural network must have enough rice disease images to ensure the final recognition effect. With the increasing requirement of rice disease recognition accuracy in real-world applications, it becomes more and more difficult to accomplish the work using convolutional neural networks alone. Many researchers have worked on optimizing the identification of rice diseases by modifying the structure of convolutional neural networks. Initially, they achieved good results, but as the optimization work progressed, the improvement in the accuracy of rice disease recognition became less effective (Lamba et al., 2023). To train and obtain a convolutional neural network that is effective in recognizing rice diseases, massive computational resources are often required (Haridasan et al., 2022; Bhuyan et al., 2023). The method relays significantly on the hardware platform, thus greatly limiting the real-time and mobile applications of convolutional neural networks in the rice disease identification process (Patil and Kumar, 2022; Senthil Pandi et al., 2022). Convolutional neural network-based rice disease recognition has to solve this problem if it aims to get out of the laboratory and gain wide application.

## 6.2 Prospect of convolutional neural network in rice disease recognition

Rice disease recognition based on convolutional neural networks still has vast research space and application value in the future. First, rice disease identification based on small sample data may be a hot spot in the future research. Enhancing the model's ability of extracting rice disease features by modifying the structure of convolutional neural networks can improve the training speed and extend the application range. Expanding the small sample dataset by data augmentation techniques or employing transfer learning methods to enhance the extraction ability of specific features by convolutional neural networks are the two feasible approaches (Lu et al., 2023; Ramkumar Raja et al., 2023). Second, coupling convolutional neural networks with other machine

learning models through an ensemble learning approach may further enhance the recognition of rice diseases (Ahad et al., 2023; Yang et al., 2023b). Different types of machine learning models are not sensitive to the features in the dataset, and integrating multiple machine learning models together can provide more comprehensive and effective recognition of various features. Therefore, it will improve the accuracy of recognition. Third, real-time identification and mobile deployment of rice diseases will also be the focus of the following research. Only when real-time identification of rice diseases is realized on mobile devices such as smartphones can a wide range of applications be truly realized. Currently, there are still many factors constraining the real-time rice disease identification and mobile deployment, such as complex network structure and large number of training data. Therefore, this research will be the focus of the next long period of time (Yang et al., 2023a).

## 7 Summary

Rice is one of basic foods in the world, and it is greatly significant to ensure its growth and yield security. From growth to harvest, rice is threatened by a variety of diseases, which poses a serious challenge to human food security. To prevent and control rice diseases, human beings began to study and control rice diseases from a very early time. Traditionally, farmers judge rice disease types based on the existing experience, which takes a lot of trouble and effort, and there are serious defects in recognition accuracy and large-scale promotion. As a very successful branch of deep learning technology, the convolutional neural network has made remarkable achievements in image-based classification, recognition, and segmentation. Automatic recognition of rice disease based on convolutional neural networks has also gained many effects in recent years. This paper summarizes the excellent achievements of convolutional neural networks in rice disease recognition in recent years and focuses on its research progress in recognition accuracy improvement, recognition speed optimization, and mobile terminal deployment. The current research on rice disease recognition based on convolutional neural networks mainly focuses on the improvement of accuracy, and the recognition accuracy in the laboratory has reached a high level through data preprocessing and network structure optimization. To extend the application of convolutional neural networks for rice disease recognition in practice, it is necessary to improve its recognition speed and the ability of deploying in mobile devices. Therefore, the simplification of the network structure is also the focus of the research. As convolutional neural networks require a large amount of sample data to support the training process, the needs for computer computation power are also relatively high, which limits the application of convolutional neural networks in rice disease identification. In the future, the convolutional neural network training based on small sample data combined with transfer learning technology, performance optimization of convolutional neural networks combined with ensemble learning technology, and simplification of the network structure for mobile deployment and real-time detection are the focus of convolutional neural networks in rice disease recognition. It is believed that this paper is important for further boosting the capability of convolutional neural networks in rice disease recognition and expanding the utilization of convolutional neural network in rice disease recognition.

## Author contributions

HN: Writing – original draft, Writing – review & editing. SL: Writing – original draft, Writing – review & editing. QZ: Writing – review & editing. TZ: Conceptualization, Writing – review & editing.

## Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was funded by the National Natural Science Foundation of China (No. 52075138) and Hainan Province Science and Technology Special Fund (No. ZDYF2022SHFZ301), the Science and Technology Plan Project of Anhui Province (Grant No. 202104f06020019), the Natural Science Foundation for the Higher Education Institutions of Anhui Province (Grant No. 2022AH051631), and the Natural Science General Project of Anhui Science and Technology University (Grant No. 2021zryb26).

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2023.1269371/full#supplementary-material

# References


Aggarwal, M., Khullar, V., Goyal, N., Singh, A., Tolba, A., Thompson, E. B., et al. (2023). Pre-trained deep neural network-based features selection supported machine learning for rice leaf disease classification. *Agriculture* 13, 936–959. doi: 10.3390/agriculture13050936

Agrawal, M. M., and Agrawal, S. (2020). Rice plant diseases detection & classification using deep learning models: a systematic review. *J. Crit. Rev.* 7, 4376–4390.

Ahad, M. T., Li, Y., Song, B., and Bhuiyan, T. (2023). Comparison of CNN-based deep learning architectures for rice diseases classification. *Artif. Intell. Agric.* 9, 22–35. doi: 10.1016/j.aiia.2023.07.001

Ahmed, T., Rahman, C. R., and Abid, M. F. M. (2023). Rice disease detection based on dual-phase convolution neural network. *Geographical Res. Bull.* 2, 128–143. doi: 10.50908/grb.2.0_128

Anvarjon, T., and Kwon, S. (2020). Deep-net: A lightweight CNN-based speech emotion recognition system using deep frequency features. *Sensors* 20, 5212. doi: 10.3390/s20185212

Arinichev, I. V., Polyanskikh, S. V., Volkova, G. V., and Arinicheva, I. V. (2021). Rice fungal diseases recognition using modern computer vision techniques. *Int. J. Fuzzy Logic Intelligent Syst.* 21, 1–11. doi: 10.5391/IJFIS.2021.21.1.1

Atole, R. R., and Park, D. (2018). A multiclass deep convolutional neural network classifier for detection of common rice plant anomalies. *Int. J. Adv. Comput. Sci. Appl.* 9, 67. doi: 10.14569/IJACSA.2018.090109

Azim, M. A., Islam, M. K., Rahman, M. M., and Jahan, F. (2021). An effective feature extraction method for rice leaf disease classification. *TELKOMNIKA (Telecommunication Computing Electron. Control)* 19, 463–470. doi: 10.12928/telkomnika.v19i2.16488

Bari, B. S., Islam, M. N., Rashid, M., Hasan, M. J., Razman, M. A. M., Musa, R. M., et al. (2021). A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework. *PeerJ Comput. Sci.* 7, e432. doi: 10.7717/peerj-cs.432

Bhujel, S., and Shakya, S. (2022). Rice leaf diseases classification using discriminative fine tuning and CLR on efficientNet. *J. Soft Comput. Paradigm* 4, 172–187. doi: 10.36548/jscp.2022.3.006

Bhuyan, P., Singh, P. K., Das, S. K., and Kalla, A. (2023). SE_SPnet: Rice leaf disease prediction using stacked parallel convolutional neural network with squeeze-and-excitation. *Expert Syst.* 40 (7), e13304. doi: 10.1111/exsy.13304

Brahimi, M., Arsenovic, M., Laraba, S., Sladojevic, S., Boukhalfa, K., and Moussaoui, A. (2018). Deep learning for plant diseases: detection and saliency map visualisation. *Hum. Mach. Learning: Visible Explainable Trustworthy Transparent* 2018, 93–117. doi: 10.1007/978-3-319-90403-0_6

Chen, J., Chen, W., Zeb, A., Yang, S., and Zhang, D. (2022). Lightweight inception networks for the recognition and detection of rice plant diseases. *IEEE Sensors J.* 22, 14628–14638. doi: 10.1109/JSEN.2022.3182304

Chen, J., Zhang, D., Nanehkaran, Y. A., and Li, D. (2020). Detection of rice plant diseases based on deep transfer learning. *J. Sci. Food Agric.* 100, 3246–3256. doi: 10.1002/jsfa.10365

Chen, J., Zhang, D., Zeb, A., and Nanehkaran, Y. A. (2021). Identification of rice plant diseases using lightweight attention networks. *Expert Syst. Appl.* 169, 114514. doi: 10.1016/j.eswa.2020.114514

Chen, R., Li, X., and Li, S. (2020). A lightweight CNN model for refining moving vehicle detection from satellite videos. *IEEE Access* 8, 221897–221917. doi: 10.1109/ACCESS.2020.3040977

Chen, W., Gao, L., Li, X., and Shen, W. (2022). Lightweight convolutional neural network with knowledge distillation for cervical cells classification. *Biomed. Signal Process. Control* 71, 103177. doi: 10.1016/j.bspc.2021.103177

Chen, S., Zhang, K., Zhao, Y., Sun, Y., Ban, W., Chen, Y., et al. (2021). An approach for rice bacterial leaf streak disease segmentation and disease severity estimation. *Agriculture* 11, 420–437. doi: 10.3390/agriculture11050420

Cui, J., and Tan, F. (2023). Rice plaque detection and identification based on an improved convolutional neural network. *Agriculture* 13, 170. doi: 10.3390/agriculture13010170

Daniya, T., and Vigneshwari, S. (2022a). Deep neural network for disease detection in rice plant using the texture and deep features. *Comput. J.* 65, 1812–1825. doi: 10.1093/comjnl/bxab022

Daniya, T., and Vigneshwari, S. (2022b). A novel Moore-Penrose pseudo-inverse weight-based Deep Convolution Neural Network for bacterial leaf blight disease detection system in rice plant. *Adv. Eng. Software* 174, 103336. doi: 10.1016/j.advengsoft.2022.103336

Deng, R., Tao, M., Xing, H., Yang, X., Liu, C., Liao, K., et al. (2021). Automatic diagnosis of rice diseases using deep learning. *Front. Plant Sci.* 12, 701038. doi: 10.3389/fpls.2021.701038

Dogra, R., Rani, S., Singh, A., Albahar, M. A., Barrera, A. E., and Alkhayyat, A. (2023). Deep learning model for detection of brown spot rice leaf disease with smart agriculture. *Comput. Electr. Eng.* 109, 108659. doi: 10.1016/j.compeleceng.2023.108659

Domingues, T., Brandão, T., and Ferreira, J. C. (2022). Machine learning for detection and prediction of crop diseases and pests: A comprehensive survey. *Agriculture* 12, 1350–1372. doi: 10.3390/agriculture12091350

Feng, S., Cao, Y., Xu, T., Yu, F., Zhao, D., and Zhang, G. (2021). Rice leaf blast classification method based on fused features and one-dimensional deep convolutional neural network. *Remote Sens.* 13, 3207. doi: 10.3390/rs13163207

Feng, C., Jiang, M., Huang, Q., Zeng, L., Zhang, C., and Fan, Y. (2022). A lightweight real-time rice blast disease segmentation method based on DFFANet. *Agriculture* 12, 1543–1554. doi: 10.3390/agriculture12101543

Feng, L., Wu, B., He, Y., and Zhang, C. (2021). Hyperspectral imaging combined with deep transfer learning for rice disease detection. *Front. Plant Sci.* 12, 693521. doi: 10.3389/fpls.2021.693521

Feng, S., Zhao, D., Guan, Q., Li, J., Liu, Z., Jin, Z., et al. (2022). A deep convolutional neural network-based wavelength selection method for spectral characteristics of rice blast disease. *Comput. Electron. Agric.* 199, 107199. doi: 10.1016/j.compag.2022.107199

Gayathri, S., Gopi, V. P., and Palanisamy, P. (2020). A lightweight CNN for Diabetic Retinopathy classification from fundus images. *Biomed. Signal Process. Control* 62, 102115. doi: 10.1016/j.bspc.2020.102115

Goluguri, N. V. R. R., Devi, K. S., and Srinivasan, P. (2021). Rice-net: an efficient artificial fish swarm optimization applied deep convolutional neural network model for identifying the Oryza sativa diseases. *Neural Comput. Appl.* 33, 5869–5884. doi: 10.1007/s00521-020-05364-x

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., et al. (2018). Recent advances in convolutional neural networks. *Pattern recognition* 77, 354–377. doi: 10.1016/j.patcog.2017.10.013

Haque, W. A., Arefin, S., Shihavuddin, A. S. M., and Hasan, M. A. (2021). DeepThin: A novel lightweight CNN architecture for traffic sign recognition without GPU requirements. *Expert Syst. Appl.* 168, 114481. doi: 10.1016/j.eswa.2020.114481

Haridasan, A., Thomas, J., and Raj, E. D. (2022). Deep learning system for paddy plant disease detection and classification. *Environ. Monit. Assess.* 195, 120. doi: 10.1007/s10661-022-10656-x

Huang, K., Liu, X., Fu, S., Guo, D., and Xu, M. (2019). A lightweight privacy-preserving CNN feature extraction framework for mobile sensing. *IEEE Trans. Dependable Secure Computing* 18, 1441–1455. doi: 10.1109/TDSC.2019.2913362

Hui, T.-W., Tang, X., and Loy, C. C. (2020). A lightweight optical flow CNN—Revisiting data fidelity and regularization. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 2555–2569. doi: 10.1109/TPAMI.2020.2976928

Hussain, T., Gill, D., and Pal Singh, R. (2021). A novel design of rice leaf disease detection model using machine learning. *Int. J. Innovative Res. Eng. Manage. (IJIREM) ISSN* 2021, 2350–0557. doi: 10.21276/ijirem.2021.8.6.16

Islam, A., Islam, R., Haque, S. M. R., Islam, S. M. M., and Khan, M. A. I. (2021). Rice leaf disease recognition using local threshold based segmentation and deep CNN. *Int. J. Intell. Syst. Appl.* 13, 35–45. doi: 10.5815/ijisa.2021.05.04

Jadhav, S. B., Udupi, V. R., and Patil, S. B. (2021). Identification of plant diseases using convolutional neural networks. *Int. J. Inf. Technol.* 13, 2461–2470. doi: 10.1007/s41870-020-00437-5

Jiang, Z., Dong, Z., Jiang, W., and YANG, Y. (2021). Recognition of rice leaf diseases and wheat leaf diseases based on multi-task deep transfer learning. *Comput. Electron. Agric.* 186, 106184. doi: 10.1016/j.compag.2021.106184

Jiang, F., Lu, Y., Chen, Y., Cai, D., and Li, G. (2020). Image recognition of four rice leaf diseases based on deep learning and support vector machine. *Comput. Electron. Agric.* 179, 105824. doi: 10.1016/j.compag.2020.105824

Joseph, D. S., Pawar, P. M., and Pramanik, R. (2023). Intelligent plant disease diagnosis using convolutional neural network: a review. *Multimed. Tools Appl.* 82, 21415–21481. doi: 10.1007/s11042-022-14004-6

Julianto, A., and Sunyoto, A. (2021). A performance evaluation of convolutional neural network architecture for classification of rice leaf disease. *IAES Int. J. Artif. Intell.* 10, 1069. doi: 10.11591/ijai.v10.i4.pp1069-1078

Kabir, M. M., Ohi, A. Q., and Mridha, M. F. (2021). A multi-plant disease diagnosis method using convolutional neural network. *Comput. Vision Mach. Learn. Agric.* 2021, 99–111. doi: 10.1007/978-981-33-6424-0_7

K.M, S., V, S., P, S. K., and O.K, S. (2023). AI based rice leaf disease identification enhanced by Dynamic Mode Decomposition. *Eng. Appl. Artif. Intell.* 120, 105836. doi: 10.1016/j.engappai.2023.105836

Lamba, S., Baliyan, A., and Kukreja, V. (2023). A novel GCL hybrid classification model for paddy diseases. *Int. J. Inf. Technol.* 15, 1127–1136. doi: 10.1007/s41870-022-01094-6

Latif, G., Abdelhamid, S. E., Mallouhy, R. E., Alghazo, J., and Kazimi, Z. A. (2022). Deep learning utilization in agriculture: detection of rice plant diseases using an improved CNN model. *Plants* 11, 2230–2246. doi: 10.3390/plants11172230

Lee, S., Kim, H., Jeong, B., and Yoon, J. (2021). A training method for low rank convolutional neural networks based on alternating tensor compose-decompose method. *Appl. Sci.* 11, 643–665. doi: 10.3390/app11020643

Li, D., Wang, R., Xie, C., Liu, L., Zhang, J., Li, R., et al. (2020). A recognition method for rice plant diseases and pests video detection based on deep convolutional neural network. *Sensors* 20, 578–598. doi: 10.3390/s20030578

Liang, W.-J., Zhang, H., Zhang, G.-F., and Cao, H.-X. (2019). Rice blast disease recognition using a deep convolutional neural network. *Sci. Rep.* 9, 2869. doi: 10.1038/s41598-019-38966-0

Lu, Y., Tao, X., Jiang, F., Du, J., Li, G., and Liu, Y. (2023). Image recognition of rice leaf diseases using atrous convolutional neural network and improved transfer learning algorithm. *Multimed. Tools Appl.* 2023, 1–19. doi: 10.1007/s11042-023-16047-9

Lu, Y., Yi, S., Zeng, N., Liu, Y., and Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. *Neurocomputing* 267, 378–384. doi: 10.1016/j.neucom.2017.06.023

Matin, M. M. H., Khatun, A., Moazzam, M. G., and Uddin, M. S. (2020). An efficient disease detection technique of rice leaf using AlexNet. *J. Comput. Commun.* 8, 49. doi: 10.4236/jcc.2020.812005

Mohapatra, D., and Das, N. (2023). A precise model for accurate rice disease diagnosis: a transfer learning approach. *Proc. Indian Natl. Sci. Acad.* 89, 162–171. doi: 10.1007/s43538-022-00149-3

N, K., Narasimha Prasad, L. V., Pavan Kumar, C. S., Subedi, B., Abraha, H. B., and V E, S. (2021). Rice leaf diseases prediction using deep neural networks with transfer learning. *Environ. Res.* 198, 111275. doi: 10.1016/j.envres.2021.111275

Ning, H., Li, R., and Zhou, T. (2022). Machine learning for microalgae detection and utilization. *Front. Mar. Sci.* 9, 947394. doi: 10.3389/fmars.2022.947394

Patil, R. R., and Kumar, S. (2022). Rice-fusion: A multimodality data fusion framework for rice disease diagnosis. *IEEE Access* 10, 5207–5222. doi: 10.1109/ACCESS.2022.3140815

Priyangka, A., and Kumara, I. M. S. (2021). Classification of rice plant diseases using the convolutional neural network method. *Lontar Komput. J. Ilm. Teknol. Inf* 12, 123. doi: 10.24843/LKJITI.2021.v12.i02.p06

Prottasha, S. I., and Reza, S. M. S. (2022). A classification model based on depthwise separable convolutional neural network to identify rice plant diseases. *Int. J. Electrical Comput. Eng. (2088-8708)* 12, 3642. doi: 10.11591/ijece.v12i4.pp3642-3654

Rahman, C. R., Arko, P. S., Ali, M. E., Khan, M. A. I., Apon, S. H., Nowrin, F., et al. (2020). Identification and recognition of rice diseases and pests using convolutional neural networks. *Biosyst. Eng.* 194, 112–120. doi: 10.1016/j.biosystemseng.2020.03.020

Ramkumar Raja, M., Shajin, F. H., and Roopa Devi, E. M. (2023). Radial basis function Neural Network optimized with Salp Swarm algorithm espoused paddy leaf disease classification. *Biomed. Signal Process. Control* 86, 105038. doi: 10.1016/j.bspc.2023.105038

Rathore, N. P. S., and Prasad, L. (2020). Automatic rice plant disease recognition and identification using convolutional neural network. *J. Crit. Rev.* 7, 6076–6086.

Sahu, P., Yu, D., Dasari, M., Hou, F., and Qin, H. (2018). A lightweight multi-section CNN for lung nodule classification and Malignancy estimation. *IEEE J. Biomed. Health Inf.* 23, 960–968. doi: 10.1109/JBHI.2018.2879834

Senan, N., Aamir, M., Ibrahim, R., Taujuddin, N. M., and Muda, W. H. N. W. (2020). An efficient convolutional neural network for paddy leaf disease and pest classification. *Int. J. Adv. Comput. Sci. Appl.* 11, 116. doi: 10.14569/IJACSA.2020.0110716

Senthil Pandi, S., Senthilselvi, A., Gitanjali, J., Arivuselvan, K., Gopal, J., and Vellingiri, J. (2022). Rice plant disease classification using dilated convolutional neural network with global average pooling. *Ecol. Model.* 474, 110166. doi: 10.1016/j.ecolmodel.2022.110166

Sethy, P. K., Barpanda, N. K., Rath, A. K., and Behera, S. K. (2020a). Deep feature based rice leaf disease identification using support vector machine. *Comput. Electron. Agric.* 175, 105527. doi: 10.1016/j.compag.2020.105527

Sethy, P. K., Barpanda, N. K., Rath, A. K., and Behera, S. K. (2020b). Rice false smut detection based on faster R-CNN. *Indonesian J. Electr. Eng. Comput. Sci.* 19, 1590–1595. doi: 10.11591/ijeecs.v19.i3.pp1590-1595

Sharma, A., Jain, A., Gupta, P., and Chowdary, V. (2020). Machine learning applications for precision agriculture: A comprehensive review. *IEEE Access* 9, 4843–4873. doi: 10.1109/ACCESS.2020.3048415

Sony, A. (2019). Prediction of rice diseases using convolutional neural network (in rstudio). *Int. J. Innov. Sci. Res. Technol.* 4, 595–602.

Sunija, A. P., Kar, S., Gayathri, S., Gopi, V. P., and Palanisamy, P. (2021). Octnet: A lightweight cnn for retinal disease classification from optical coherence tomography images. *Comput. Methods programs biomedicine* 200, 105877.

Thakur, P. S., Sheorey, T., and Ojha, A. (2023). VGG-ICNN: A Lightweight CNN model for crop disease identification. *Multimedia Tools Appl.* 82, 497–520. doi: 10.1007/s11042-022-13144-z

Ullah, H., Muhammad, K., Irfan, M., Anwar, S., Sajjad, M., Imran, A. S., et al. (2021). Light-DehazeNet: a novel lightweight CNN architecture for single image dehazing. *IEEE Trans. Image Process.* 30, 8968–8982. doi: 10.1109/TIP.2021.3116790

Upadhyay, S. K., and Kumar, A. (2022). A novel approach for rice plant diseases classification with deep convolutional neural network. *Int. J. Inf. Technol.* 14, 185–199. doi: 10.1007/s41870-021-00817-5

Wang, C.-L., Li, M.-W., Chan, Y.-K., Yu, S.-S., Ou, J.-H.., Chen, C.-Y., et al. (2022). Multi-scale features fusion convolutional neural networks for Rice leaf disease identification. *J. Imag. Sci. Technol.* 66, 1–12.

Wang, Y., Wang, H., and Peng, Z. (2021). Rice diseases detection and classification using attention based neural network and bayesian optimization. *Expert Syst. Appl.* 178, 114770. doi: 10.1016/j.eswa.2021.114770

Wang, Z.-R., and Du, J. (2021). Joint architecture and knowledge distillation in CNN for Chinese text recognition. *Pattern Recognition* 111, 107722. doi: 10.1016/j.patcog.2020.107722

Wei, L., Luo, Y., Xu, L., Zhang, Q., Cai, Q., and Shen, M. (2022). Deep convolutional neural network for rice density prescription map at ripening stage using unmanned aerial vehicle-based remotely sensed images. *Remote Sens.* 14, 46. doi: 10.3390/rs14010046

Yakkundimath, R., Saunshi, G., Anami, B., and Palaiah, S. (2022). Classification of rice diseases using convolutional neural network models. *J. Institution Engineers (India): Ser. B* 103, 1047–1059. doi: 10.1007/s40031-021-00704-4

Yamashita, R., Nishio, M., Do, R. K. G., and Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging* 9, 611–629. doi: 10.1007/s13244-018-0639-9

Yang, L., Yu, X., Zhang, S., Long, H., Zhang, H., Xu, S., et al. (2023a). GoogLeNet based on residual network and attention mechanism identification of rice leaf diseases. *Comput. Electron. Agric.* 204, 107543. doi: 10.1016/j.compag.2022.107543

Yang, L., Yu, X., Zhang, S., Zhang, H., Xu, S., Long, H., et al. (2023b). Stacking-based and improved convolutional neural network: a new approach in rice leaf disease identification. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1165940

Yeom, S.-K., Seegerer, P., Lapuschkin, S., Binder, A., Wiedemann, S., Müller, K.-R., et al. (2021). Pruning by explaining: A novel criterion for deep neural network pruning. *Pattern Recognition* 115, 107899. doi: 10.1016/j.patcog.2021.107899

Zhang, R., and Chung, A. C. S. (2021). MedQ: Lossless ultra-low-bit neural network quantization for medical image segmentation. *Med. Image Anal.* 73, 102200. doi: 10.1016/j.media.2021.102200

Zhang, J., Yang, Y., Feng, X., Xu, H., Chen, J., and He, Y. (2020). Identification of bacterial blight resistant rice seeds using terahertz imaging and hyperspectral imaging combined with convolutional neural network. *Front. Plant Sci.* 11, 821. doi: 10.3389/fpls.2020.00821

Zhang, X., Zhou, J., Sun, W., and Jha, S. K. (2022). A lightweight CNN based on transfer learning for COVID-19 diagnosis. *Computers Materials Continua* 72 (1), 1123–1137. doi: 10.32604/cmc.2022.024589

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 04 January 2024
DOI 10.3389/fpls.2023.1283055

Check for updates

OPEN ACCESS

EDITED BY
Yalin WU,
Lushan Botanical Garden (CAS), China

REVIEWED BY
Jianxin Wang,
Beijing Forestry University, China
Xin-jun Du,
Tianjin University of Science and Technology, China
Guangcan Tao,
Guiyang University, China
Chang Kuo-Chi,
Fujian University of Technology, China

*CORRESPONDENCE
Boyuan Wang
2109853pia30001@student.must.edu.mo

RECEIVED 25 August 2023
ACCEPTED 28 November 2023
PUBLISHED 04 January 2024

CITATION
Wang B (2024) Zero-exemplar deep continual learning for crop disease recognition: a study of total variation attention regularization in vision transformers.
*Front. Plant Sci.* 14:1283055.
doi: 10.3389/fpls.2023.1283055

COPYRIGHT
© 2024 Wang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# Zero-exemplar deep continual learning for crop disease recognition: a study of total variation attention regularization in vision transformers

Boyuan Wang*

School of Computer Science and Engineering, Faculty of Innovation Engineering, Macau University of Science and Technology, Macao, Macao SAR, China

With the increasing integration of AI technology in the food industry, deep learning has demonstrated its immense potential in the domain of plant disease image recognition. However, there remains a gap in research between models capable of continual learning of new diseases and addressing the inherent catastrophic forgetting issue in neural networks. This study aims to comprehensively evaluate various learning strategies based on advanced computer vision models for multi-disease continual learning tasks in food disease recognition. To cater to the benchmark dataset requirements, we collected the PlantDiseaseCL dataset, sourced from the internet, encompassing diverse crop diseases from apples, corn, and more. Utilizing the Vision Transformer (ViT) model, we established a plant disease image recognition classifier, which, in joint learning, outperformed several comparative CNN architectures in accuracy (0.9538), precision (0.9532), recall (0.9528), and F1 score (0.9560). To further harness the potential of ViT in food disease defect recognition, we introduced a mathematical paradigm for crop disease recognition continual learning. For the first time, we proposed a novel ViT-TV architecture in the multi-disease image recognition scenario, incorporating a Total Variation (TV) distance-based loss (TV-Loss) to quantify the disparity between current and previous attention distributions, fostering attention consistency and mitigating the catastrophic forgetting inherent in ViT without prior task samples. In the incremental learning of the PlantDiseaseCL dataset across 3-Steps and 5-Steps, our strategy achieved average accuracies of 0.7077 and 0.5661, respectively, surpassing all compared Zero-Exemplar Approaches like LUCIR, SI, MAS, and even outperforming exemplar-based strategies like EEIL and ICaRL. In conclusion, the ViT-TV approach offers robust support for the long-term intelligent development of the agricultural and food industry, especially showcasing significant applicability in continual learning for crop disease image recognition.

KEYWORDS

**smart agriculture, plant disease detection, deep learning, continual learning, vision transformer**

# 1 Introduction

Plant diseases reduce the yield and quality of food, leading to significant economic losses and reducing food safety at the national and global levels (Savary et al., 2019). Plant disease surveillance is critical for preventing disease spread (Jones, 2021; Ristaino et al., 2021). However, current monitoring approaches rely on regular field identification by agroforestry specialists or farmers' knowledge of plant diseases. This manual inspection-based technique is time-consuming and costly, and it also necessitates specialists' a high level of field knowledge. Therefore, the development of smart agriculture requires a detection system that can automatically identify the type of plant disease and the exact location of the lesion.

With the advancement of AI technologies, researchers have utilized machine learning and image processing techniques to develop systems to automatically detect plant diseases such as apple disease(Chakraborty et al., 2021), wheat disease (Nema and Dixit, 2018), cotton disease (Bhimte and Thool, 2018), and corn disease (Kusumo et al., 2018). Color, shape, and texture information are used to construct feature vectors, which are then classified using random forest (Mekha and Teeyasuksaet, 2021), support vector machines (SVM) (Banerjee and Madhumathy, 2022), etc. However, traditional machine learning and image processing methods rely heavily on prior knowledge and require human design based on disease characteristics, making it difficult to use big data to discover feature patterns automatically (Liu and Wang, 2021). The essence of these techniques is by manually designing features and developing classifiers (or rules) and using computer image processing methods such as image segmentation methods (Prewitt, Sobel), feature extraction methods (SIFT, HOG) and classification methods (SVM). When the environment or the type of plant disease changes, it is always required to change the threshold or redesign the algorithm, which is inefficient for detection in real and complex natural environments (Liu and Wang, 2021). Therefore, the classification performance is low, the model lacks stability, and the adaptability is poor.

After the emergence of deep learning technology, an important branch of AI, models for end-to-end disease detection by learning features from different fields, scenarios, and scales have become a research hotspot in the field of smart agriculture and food industry. Deep learning techniques can automatically learn features from massive amounts of data and cope with specific complex changes in the natural environment (Boulent et al., 2019; Liu and Wang, 2021). Models for plant disease image recognition based on deep learning techniques belong to deep neural networks (DNN), including the classic convolutional neural network (CNN) (Albawi et al., 2017) and the latest ViT (Dosovitskiy et al., 2020), etc. The CNN architectures include VGG16 (Simonyan and Zisserman, 2015), ResNet (He et al., 2016), NASNet (Zoph et al., 2018), Inception V3 (Szegedy et al., 2016), MobileNet (Howard et al., 2017), EfficientNet (Tan and Le, 2019), etc. All these models are very deep neural networks formed by stacking multiple convolutional layers. All above models have been applied to the study of plant disease identification.

In reference to Sultana et al. 's study (Habiba and Islam, 2021), they utilized the VGG16 model for identifying diseased tomatoes through transfer learning. The study focused on ten different categories of tomato leaf images from the Plant Village dataset, namely: a) Target Spot, b) Yellow leaf, c) Mosaic Virus, d) Bacterial Spot, e) Early Blight, f) Leaf Mold, g) Late Blight, h) Septoria Leaf spot, i) Spider Mites, and j) Healthy Leaf. The dataset consists of a varying number of photos per class, ranging between 1500 and 3000. To ensure proper training, validation, and testing, the dataset was divided into 60% for training data, 20% for validation data, and 20% for test data. The results showed satisfactory classification performance with an accuracy of about 95.5%. Brahmaji et al. (Godi et al., 2022a) used the ResNet-152 V2 model for automatic disease identification on a tomato leaf image dataset containing ten different diseases. The processing flow designed mainly consisted of Pre-processing of leaf structure, leaf feature extraction, leaf analysis and segmentation, and leaf classification process. After training, The ResNet-152 V2 model achieved 95% detection accuracy. Yang et al. (Yang et al., 2020) developed a plant disease image classification model based on NASNet's extended neural network and attention mechanism. Their study used a dataset consisting of 58,200 crop leaf images, including 37 different classes of healthy/diseased crops. The results show that the fine-grained NASNet Large neural network model based on the attention mechanism achieves excellent classification performance with 95.62% accuracy, which is well suited for automatically detecting crop diseases. Haque et al. (Haque et al., 2022) collected 5939 images of maize crops from experimental fields located in three maize growing areas, including three types of diseases: Maydis leaf blight, Turcicum leaf blight, and Banded leaf and sheath blight, as well as healthy ones. They used the basic architecture of the advanced CNN model "Inception-v3" network to build three models on the maize dataset, viz. flatten layer with fully connected layer (Inception-V3_flatten-FC), global average pooling layer (Inception-v3_GAP) and global average pooling layer with fully connected layer (Inception-V3_GAP-FC). Of these, Inception-v3_GAP achieved the highest accuracy of 95.99% in a separate test set and was efficient in learning relevant features of the disease and predicting the correct category in unseen data. Rajbongshi et al. (Rajbongshi et al., 2020) used the MobileNet model with a transfer learning approach to detect rose plant diseases on an image dataset of powdery mildew, black spot, rust, and dieback diseases. They used 1600 data images to train the model and 400 data images to test the model. As a result, the MobileNet model with the transfer learning method obtained an accuracy of about 95.63%. Vijayalata et al. (Vijayalata et al., 2022) focused their research on identifying four diseases affecting cassava yield: Cassava Bacterial Blight, Cassava Brown Streak Disease, Cassava Mosaic Disease, and Cassava Green Mottle. They used the EfficientNet-B0 model for the early detection of these diseases. A total of 21,367 cassava images comprised the original image dataset, which was divided into 20 test cases and 80% of the training data, and 20% of the validation data. An accuracy of 92.6% was achieved after the model was applied to the test cases.

Zhuang (Zhuang, 2021) suggested a ViT model-based method for identifying viral diseases in cassava leaf images. The image dataset of cassava leaves was provided by Makerere Artificial Intelligence Lab in a Kaggle competition, including four subtypes of diseases and healthy cassava leaves. After applying the K-Fold

cross-validation method, their model achieved a classification accuracy of 90.02% on the test set. Zhang et al. (Zhang et al., 2021) proposed a new rice disease recognition method based on the Swin Transformer architecture (a new variant of ViT), including sliding window operation and hierarchical design. The proposed model was trained with images of five rice diseases (bacterial blight, rice blast, rice false smut, brown spot, and sheath blight) in the field environment and achieved a classification accuracy of 93.4% on the test set, which is about 4.1% higher than that of traditional machine learning models. Li et al. (Li and Li, 2022) proposed a lightweight ViT-based disease detection model, ConvViT, for apple disease identification in complex environments. ConvViT includes a convolutional structure and a Transformer structure, and the detection accuracy result (96.85%) is comparable to the performance of the current state-of-the-art Swin-Tiny. The parameters and FLOPs are only 32.7% and 21.7% of Swin-Tiny, significantly ahead of CNN models such as MobilenetV3 and Efficientnet-b0.

Both CNN and Transformer architectures have demonstrated exceptional capabilities in detecting plant diseases, surpassing the expertise of agroforestry professionals in certain tasks. Nonetheless, the majority of these models are anchored in static datasets and unchanging settings, overlooking the fact that information often unfolds progressively. As a result, they struggle to assimilate and adapt to fresh insights. On occasion, they might completely break down or exhibit pronounced deterioration in tasks they once mastered, culminating in profound issues of catastrophic forgetting (Hadsell et al., 2020). This phenomenon, where neural networks lose prior knowledge, was first pinpointed by McCloskey and Cohen in 1989 (McCloskey and Cohen, 1989). When juxtaposed with these artificial models, the human aptitude for learning is rooted in a diverse array of neurocognitive processes and brain memory systems. Such complexities underpin our ability to hone skills and embed memories for the long haul, as detailed by German I. Parisi et al. in 2019 (Parisi et al., 2019).

Drawing from the principles of cognitive science, the realm of continual learning, as articulated by Lesort et al. (Lesort et al., 2020) endeavors to confront the aforementioned limitations in artificial intelligence. To achieve a balance between preserving old knowledge and learning new knowledge, continual learning algorithms face a trade-off known as the stability-plasticity dilemma (Abraham and Robins, 2005; Wu et al., 2021; Araujo et al., 2022). In the traditional static learning, data follows independent and identically distributed (IID) distributions, where data is sampled according to the same probability distribution. In typical IID data sets $D$, we have $D \sim P(x, y)$, where $P(x, y)$ denotes the joint probability distribution of the data generation. However, in a continual learning environment, the data probability distribution of the data set $D$ is no longer a typical IID probability distribution but is instead divided into several distinct subsets $D_t$. Let $D = \cup_{t=1}^{T} D_t$, where each subset represents a single task and is sampled from $T$ different IID probability distributions $P_t(x, y)_{t=1}^{T}$. In a continual learning environment, the dataset $D$ can be represented as $D = \cup_{t=1}^{T} D_t$ with $D_t \sim P_t(x, y)$, where $D_t$ represents the subset corresponding to the $t$ -th task and $P_t(x, y)$ represents the probability distribution of the $t$ -th task. The characteristic of continual learning is that it learns from dynamic data distributions, allowing for more flexible and adaptive machine learning systems.

At present, the food industry sees limited exploration and utilization of Continual Learning techniques in AI-driven smart solutions. To bridge this gap, we undertook this investigation, outlining our primary findings below.

The potential of the ViT model in the food industry, particularly in food disease defect recognition, necessitates its enhancement and evaluation for continual learning capabilities. Establishing advanced continual learning visual models in the domain of food and crop disease prevention is imperative. Addressing these challenges, this study was undertaken, and the following key contributions were made:

1. To meet the benchmark dataset requirements for multi-disease continual learning classification tasks, we curated the PlantDiseaseCL dataset from the internet, encompassing diverse food diseases from apples, corn, and more. Using the ViT-S/16 model, we developed a food disease image recognition classifier. In joint learning evaluations, the ViT-S/16 outperformed several other CNN architectures in metrics such as accuracy, precision, recall, and F1 score.
2. Beyond just model performance, we delved into the model's feature learning capability using the t-SNE method. Visualization of feature vectors learned by different models revealed that the ViT-S/16 demonstrated superior classification outcomes in feature distribution, excelling in inter-class separability in feature embeddings.
3. To maximize the potential of ViT in food disease defect recognition, we introduced a mathematical paradigm for continual learning of crop disease defects. We proposed the novel ViT-TV architecture for multi-disease image recognition, incorporating a Total Variation distance (Rudin et al., 1992; Bhojanapalli et al., 2021) loss (TV-Loss) to quantify disparities between current and previous attention distributions.
4. By optimizing the overall loss function of ViT with TV-Loss and Cross-Entropy Loss, we balanced model stability and plasticity, maintaining attention consistency during the learning process of new and old tasks, thereby mitigating the catastrophic forgetting inherent in ViT without the need for storing samples from previous tasks. This offers a new attention alignment method for ViT in multi-disease continual learning scenarios.
5. To validate our proposed ViT-TV, we designed 3-stage and 5-stage continual learning processes on the PlantDiseaseCL dataset. We assessed various continual learning methods from perspectives such as attention alignment, global importance parameter regularization, and knowledge transfer between teacher-student networks for new and old tasks.
6. Further comparisons of different attention alignment method variants were made, juxtaposing our ViT-TV with the original ViT and other methods like Jensen–Shannon divergence, Hellinger distance, and

Bhattacharyya distance, thereby confirming the efficacy of our approach.

These contributions establish a novel framework for continual learning in image classification tasks for food disease recognition. The proposed Zero-Exemplar approach ViT-TV method fosters advancements in multi-disease recognition technology, enhancing the model's capability to continuously learn new diseases, and underpinning the long-term intelligent evolution of the food industry.

The remainder of this paper is organized as follows. The sources and construction methods for training, verifying, and testing datasets are described in Section II. Section III describes our proposed approach, ViT-TV and performance evaluation metrics. Model parameter settings for the experimental study are discussed in Section IV. The experimental results and discussion are presented in Section V. Section VI conclude the paper with comments on future work.

# 2 Materials and methods

## 2.1 Datasets

To validate our proposed methodology, we collected the PlantDiseaseCL dataset, specifically designed for continual learning evaluations. This dataset comprises 30,863 disease images of various foods, including apples, corn, pepper, and potatoes, all of which were collected from the Internet. Each image is standardized to a resolution of 256 × 256 pixels. For structured evaluation, the dataset has been segmented into training, validation, and testing subsets, detailed further in Table 1. For the broader research community's benefit, we have made the PlantDiseaseCL dataset publicly available on the Kaggle platform. It can be accessed at https://www.kaggle.com/datasets/gabrielwang01/leaf-disease-must (last accessed on 18 August 2023).

## 2.2 Methods

### 2.2.1 Multi-disease continual learning paradigm for crops and foods

Continual learning for multi-disease detection in crops plays a significant role in improving agricultural productivity. The ability to detect, differentiate, and act upon a growing variety of diseases over time can significantly impact the crop yield and the overall food supply chain. The mathematical paradigm outlined for multi-disease medical image recognition can similarly be adapted to design a theoretical framework for crops.

Let's define our dataset for crop disease recognition as $C = \{(x_j, z_j)\}_{\{j=1\}}^{m}$, where $x_j$ denotes the sample feature (e.g., an image of a plant or crop's leaf, hyperspectral data) within the feature space $X$ and $z_j$ is the corresponding label within the label space $Z$. Here, $(X)$ represents the input space detailing the features (or symptoms) exhibited by crops due to diseases or other external factors. In contrast, $Z$ is the output space indicating the type of crop disease or health status.

**Incremental Learning over Time**: In real-world scenarios, new crop diseases may emerge, or previously studied diseases might show new symptoms. Therefore, similar to the multi-disease medical paradigm, our dataset $C$ will be split into $K$ stages, each containing $m_k$ data points, where $m_k \in M$ and $\sum_{k=1}^{K} m_k = m$. The parameter set or vector M serves as a means to control the distribution of data points across stages, allowing for flexibility and adaptation based on specific requirements.

**The Learning Objective**: For the crop multi-disease scenario, at each stage, our goal is to generate a model that can recognize all the diseases observed so far without forgetting the previously learned diseases. Mathematically, the objective at the $k+1$ stage is expressed as shown in Equation 1.

$$\theta_{k+1} = \arg\min_{\theta} L(g_{k+1}(\cdot, \theta_k), C_{k+1}) \tag{1}$$

where $g_{k+1}$ represents the model at the $k+1$ stage, θ is the parameter set or vector of the model, and $C_{k+1} = C'_{k+1} \cup \{(x, z) \in C : x \in X_k\}$ is the combined dataset of the prior $k+1$ tasks and the specific crop disease at task $k$.

**Dealing with Catastrophic Forgetting**: To avoid catastrophic forgetting, we need to integrate techniques such as Regularization-based Approach, which adds a penalty to the loss function to ensure the weight changes for a new task do not drastically deviate from the learned weights for previous tasks.

Given: Original loss function: $\mathcal{L}$, Weights of the neural network: $W$, Previously learned weights: $W_{prev}$,

The new loss function, incorporating the regularization term, can be expressed as shown in Equation 2.

$$\mathcal{L}_{new} = \mathcal{L} + \lambda \sum_{i} (W_i - W_{prev,i})^2 \tag{2}$$

where $\lambda$ is a regularization parameter. The summation is taken over all weights in the network. The continual learning approach, when correctly applied to crops, can lead to proactive disease management, better yields, and a more resilient food system. The mathematical paradigm above sets the foundation for building AI systems that can evolve with changing disease landscapes in agriculture.

### 2.2.2 ViT-TV: aligning ViT attention using total variation distance

In our study, we enhanced the original ViT model (Dosovitskiy et al., 2020) to address the challenges of continual learning across multiple diseases. The core concept behind Vision Transformers (ViT) revolves around processing based on image patches.

**For the original ViT**: Given an image $I$ of dimensions $(H \times W \times C)$, where $(H)$ and $(W)$ represent the height and width of the image, and $C$ denotes the number of channels, we partition the image into $n$ patches, each of size $(P \times P)$. Thus, $n = \frac{H \times W}{P \times P}$.

Each image patch $i$ can be linearly embedded into a vector $v_i$ of dimension $D$, as shown in Equation 3.

$$v_i = MA[patch_i] \tag{3}$$

TABLE 1 Training, validation, and test sets for the PlantDiseaseCL dataset.

| Class | Images | Dataset | | |
|---|---|---|---|---|
| | | Training | Validation & Testing | Total |
| Apple healthy |  | 1506 | 1004 | 2510 |
| Apple Black Rot |  | 1490 | 994 | 2484 |
| Apple rust |  | 1320 | 880 | 2200 |
| Apple scab |  | 1512 | 1008 | 2520 |
| Corn healthy |  | 1394 | 930 | 2324 |
| Corn common rust |  | 1430 | 954 | 2384 |
| Corn gray leaf spot |  | 1232 | 820 | 2052 |
| Corn northern leaf blight |  | 1431 | 954 | 2385 |
| Pepper healthy |  | 1491 | 994 | 2485 |
| Pepper bacterial spot |  | 1435 | 956 | 2391 |
| Potato healthy |  | 1368 | 912 | 2280 |
| Potato early blight |  | 1454 | 970 | 2424 |
| Potato late blight |  | 1454 | 970 | 2424 |
| Total | | 18517 | 12346 | 30863 |

where $MA$ is an embedding matrix with dimensions $D \times (P \times P \times C)$. In this context, $patch_i$ serves as an index to select a specific row from the embedding matrix $MA$. These embeddings are then processed through $L$ Transformer layers. Each Transformer layer consists of two primary components: Multi-Head Self-Attention (MHSA) and a Multi-Layer Perceptron (MLP). The design of MHSA aims to capture information in parallel across different representational subspaces. Briefly describing its operation, this structure first projects the input data into multiple representational spaces, each having its unique set of queries, keys, and values.

For $h$ heads, each head has its distinct set of projection matrices: $\{P_i^Q, P_i^K, P_i^V\}$, where $i$ denotes the $i^{th}$ head. These matrices project the original input data into their respective subspaces, as shown in Equation 4.

$$Q_i = \text{Input} \cdot P_i^Q, \quad K_i = \text{Input} \cdot P_i^K, \quad V_i = \text{Input} \cdot P_i^V \tag{4}$$

In their respective subspaces, for each head $i$, a standard attention operation is executed, as illustrated in Equation 5.

$$A_i = \text{Softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_i}}\right) V_i \tag{5}$$

Where $d_i$ represents the dimension of the $i^{th}$ head. Finally, the outputs from all heads are concatenated and passed through a shared output transformation, resulting in the final outcome, as depicted in Equation 6.

$$\text{Output} = \text{Concat}(A_1, \ldots, A_h) \cdot P^O \tag{6}$$

Where ($P^O$) is the weight matrix of the output transformation. This multi-head structure enables the model to capture various features and dependencies in parallel across multiple subspaces, enhancing the model's expressive capability.

**Global Total Variation Distance Regularization**: In continuous recognition of crop diseases, as time progresses, new diseases might emerge, or the manifestations of known diseases may evolve. Thus, we can represent the continuous disease recognition tasks as described in Equation 7.

$$[T_{disease1}, T_{disease2}, \ldots, T_{diseasen}] \tag{7}$$

For each disease task ($T_{disease(i)}$), there exists a unique data distribution: [$P(image, label\ T_{disease(i)})$], where image represents the image data of crop leaves, and  denotes the disease label. Prior to Equation 9, the assumption is made that the covariance between two diseases is zero, denoted as $\Sigma_{disease(i),j} = 0$. This assumption implies that the parameter distributions of different diseases are statistically independent. Mathematically, it can be expressed as represented in Equation 8.

$$\Sigma_{disease(i,j)} = E\left[(\theta - \mu_{disease(i)})(\theta - \mu_{disease(j)})^T\right] = 0 \tag{8}$$

Here, $\Sigma_{disease(i,j)}$ represents the covariance matrix between disease $T_{disease(i)}$ and $T_{disease(j)}$, where θ denotes the model parameters, and $\mu_{disease(i)}$ and $\mu_{disease(j)}$ represent the mean parameters for diseases $T_{disease(i)}$ and $T_{disease(j)}$, respectively.

This assumption signifies that the learning of parameters for one disease does not influence the parameters of other diseases. By assuming independence between disease-specific parameter distributions, we establish a foundation for further derivation and utilization of Equation 9 in addressing continual learning tasks.

To learn on a specific disease, we typically aim to maximize the following likelihood function:

$$L(\theta|T_{disease(i)}) = \sum_{(image,label)\in D_{disease(i)}} \log P(label|image; \theta, T_{disease(i)}) \tag{9}$$

Where $D_{disease(i)}$ is the dataset for disease $T_{disease(i)}$ and θ represents the model parameters.

Probability Distribution Shift: In the continuous recognition tasks of crop diseases, as new diseases emerge or known disease manifestations change, the model needs to be updated continuously. Suppose the model parameter distribution after disease ($T_{disease(i)}$) is ($P(\theta|T_{disease(i)})$). When encountering a new disease ($T_{disease(j)}$), we desire the model parameter distribution to be ($P(\theta|T_{disease(j)})$).

We employ Bayesian updating to describe this process, as expressed in Equation 10.

$$P(\theta|T_{disease(j)}, D_{disease(j)}) \propto P(D_{disease(j)}|\theta, T_{disease(j)}) \\ P(\theta|T_{disease(i)}, D_{disease(i)}) \tag{10}$$

Where ($P(\theta|T_{disease(j)}, D_{disease(j)})$) is the posterior distribution, representing the distribution of the model parameters (θ) given the new disease ($T_{disease(j)}$) and its associated data ($D_{disease(j)}$). ($P(D_{disease(j)}|\theta, T_{disease(j)})$) is the likelihood function, indicating the probability of observing the data ($D_{disease(j)}$) given the model parameters (θ) and the disease ($T_{disease(j)}$). ($P(\theta|T_{disease(i)}, D_{disease(i)})$) is the prior distribution, which describes our belief about the distribution of the model parameters (θ) before considering the disease ($T_{disease(i)}$) and its data ($D_{disease(i)}$).

In continuous recognition tasks of crop diseases, maintaining knowledge from historical learning is crucial. Like other continual learning tasks, when introducing new disease categories or encountering new data distributions, we might face the risk of "catastrophic forgetting", where the process of acquiring new knowledge might disrupt what has been previously learned.

To effectively address this issue, we introduced the TV distance (Rudin et al., 1992; Bhojanapalli et al., 2021) as a regularization technique for the first time in continuous recognition of crop diseases. The TV distance provides us with a means to evaluate the parameter changes in the model across continuous tasks.

The TV distance provides a measure to gauge the difference between two probability distributions associated with the parameters of neural networks. In the context of Bayesian, we treat the weights and biases of the neural network as random variables, effectively viewing the entire set of parameters as a probability distribution. Given two such distributions, ($p(\theta|D_{disease})$) and ($q(\theta|D_{disease})$), which represent the distributions of the whole neural network parameters under two different disease conditions, the TV distance between them is defined as shown in Equation 11.

$$\text{TV}(\text{p},\text{q}) = \frac{1}{2}\int |p(\theta|D_{disease}) - q(\theta|D_{disease})|\text{d}\theta \tag{11}$$

For discrete distributions, this formula can be written as expressed in Equation 12.

$$TV(p,q) = \frac{1}{2}\sum_{\theta} |p(\theta|D_{disease}) - q(\theta|D_{disease})| \tag{12}$$

The core idea behind TV distance is to describe the maximum deviation of two probability distributions for the same event in the context of crop diseases. Considering the posterior distribution of parameters for old tasks ($q(\theta|D_{disease(i)})$) and the posterior distribution of parameters based on new disease data ($p(\theta|D_{disease(i)+1})$), our optimization objective can be expressed as represented in Equation 13.

$$L_{T_{disease(i)+1}}(\theta) = -\sum_{(x,y)\in D_{disease(i)+1}} \log P(yx; \theta, T_{disease(i)+1}) \\ + \lambda TV(p(\theta|D_{disease1:i+1}), q(\theta|D_{disease1:i})) \tag{13}$$

Where:

- $D_{disease1:i}$ and $D_{disease1:i+1}$ represent the dataset up to disease $i$ and the dataset up to disease $i+1$, respectively.
- $p(\theta|D_{disease1:i+1})$ is the posterior distribution of the parameters $\theta$ based on the new disease data (up to disease $i+1$).
- $q(\theta|D_{disease1:i})$ denotes the posterior distribution of the parameters $\theta$ based on the old disease data (up to disease $i$).
- $TV(p,q)$ stands for the Total Variation distance, which assesses the difference between two probability distributions $p$ and $q$ specific to crop diseases.
- $\theta$ encapsulates the model's parameter set, defining its structure and behavior, and is typically adjusted during training to minimize the loss function.
- $(x,y)$ is a sample pair, with $x$ being the input (e.g., a crop image) and $y$ the corresponding label (e.g., disease category).
- $\lambda$ is a regularization coefficient, determining the weight of the TV distance in the overall loss. Adjusting $\lambda$ allows for a balance between the loss for task $i+1$ and the change in parameter distribution. Setting $\lambda$ too high might render the model overly conservative, hindering adaptation to the new task, while a value too low might cause an overemphasis on the new task, risking the forgetting of old tasks.

**ViT-TV Attention Alignment**: In this study, we introduce an enhanced model, termed ViT-TV, that aligns attention based on the TV distance, as depicted in Figure 1. The ViT model aims to integrate the continual learning recognition task of crop diseases by leveraging attention mechanisms. Given an input feature matrix $(X)$ with dimensions $B \times N \times C$, where $B$ denotes batch size, $N$ represents sequence length, and $C$ signifies feature dimensions, the model initially undergoes a linear transformation to obtain a combined representation for $Query(Q)$, $Key(K)$, and $Value(V)$, as shown in Equation 14:

$$QKV = XW_{qkv} \tag{14}$$

Here, $W_{qkv}$ is a weight matrix. After this transformation, the combined representation $QKV$ is reshaped and permuted to separate out the individual representations for $Q$, $K$, and $V$. Specifically, $QKV$ is reshaped to dimensions $(B, N, 3, H, \frac{C}{H})$, where $B$ denotes the batch size, $N$ represents the sequence length, and $H$ stands for the number of attention heads. The tensor is then permuted to rearrange these dimensions, resulting in individual tensors for $Q$, $K$, and $V$. Subsequently, attention scores ATTN are computed, as illustrated in Equation 15.

$$ATTN = (Q \times K^T) \times \frac{1}{\sqrt{d}} \tag{15}$$

Where $(d)$ is the dimension size of each attention head, utilized to scale the dot product. For each score, the softmax function is applied to ensure the sum equals 1 across the last dimension, as demonstrated in Equation 16.

$$ATTN' = \text{Softmax}(ATTN) \tag{16}$$

The computed attention weights are then dot-multiplied with the Value matrix (V), as represented in Equation 17:

$$Z = ATTN' \times V \tag{17}$$

Finally, $Z$ undergoes another linear transformation followed by a dropout layer to produce the model's output. When training the model on a new crop disease recognition task, while ensuring it retains knowledge from previous tasks, we employ attention disparities to compute the regularization loss. Each attention matrix is reshaped from $(B \times h \times w)$ to $((B \times w) \times h)$, where $h$

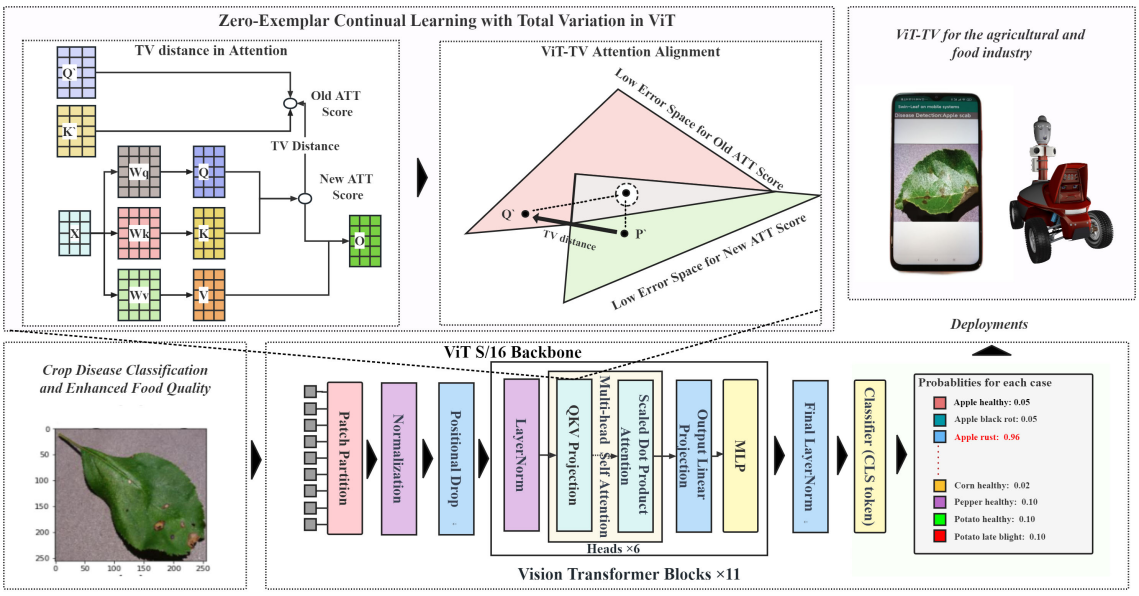



FIGURE 1
Framework for crop disease continual learning with ViT-TV attention alignment.

and $w$ are the height and width of the attention matrix, respectively. To ensure the sum of weights in each attention matrix equals 1, normalization is applied, as represented in Equation 18.

$$P' = \frac{|P|}{\sum_{j=1}^{h} P_{ij}}, \quad Q' = \frac{|Q|}{\sum_{j=1}^{h} Q_{ij}} \tag{18}$$

Where $P$ denotes the attention matrix associated with the previous task, encapsulating the model's attention distribution during that phase. Conversely, $Q$ signifies the attention matrix pertinent to the current task, illustrating the model's attention distribution for the new task at hand. The matrices $P'$ and $Q'$ represent their normalized counterparts, ensuring a standardized attention distribution across the matrix dimensions.

The absolute value operation ensures all values are positive, and normalization ensures the sum of each row equals 1. The difference between the two normalized attention matrices is then computed using the Total Variation distance, as demonstrated in Equation 19.

$$\mathrm{TV}(P', Q') = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{h} \left| P'_{ij} - Q'_{ij} \right| \tag{19}$$

For all attention matrices, the TV distances are accumulated to compute the overall loss, as indicated in Equation 20.

$$\text{Total_Loss} = \sum_{i=1}^{len(\text{attention_list})} \mathrm{TV}(P'_i, Q'_i) \tag{20}$$

To effectively balance the learning of the new crop disease recognition task and the retention of knowledge from previous tasks, we introduce a composite loss consisting of two components:

Cross-Entropy Loss: For the new crop disease recognition task, we compute the cross-entropy loss between the model's predictions and the actual labels, as expressed in Equation 21.

$$L_{\text{cross-entropy}} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)] \tag{21}$$

Where $y_i$ is the actual label, and $\hat{y}_i$ is the model's prediction.

Attention Regularization Loss: Based on the aforementioned description, we have computed the TV distance between two attention matrices, which serves as the regularization loss, as expressed in Equation 22.

$$L_{\text{attention}} = \text{TV_Loss} \tag{22}$$

This loss ensures that during training on a new crop disease recognition task, the model doesn't drastically alter its attention weights from previous tasks. Ultimately, these two losses are combined into a total loss, where λ is a hyperparameter to balance the two, as shown in Equation 23.

$$L_{\text{total}} = L_{\text{cross-entropy}} + \lambda \times L_{\text{attention}} \tag{23}$$

Our objective is to adjust the model parameters to minimize the TV distance, ensuring that predictions on new disease data closely align with the true distribution while maintaining consistency with old disease data. By optimizing this composite loss, the ViT-TV model can retain knowledge of previous tasks while learning new crop disease recognition tasks, even without sample replay.

### 2.2.3 Evaluation metrics

Accuracy is the ratio of correctly predicted samples to the total number of samples, as represented in Equation 24.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Negatives} \tag{24}$$

Precision is the ratio of true positives to the sum of true positives and false positives, as expressed in Equation 25.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positive} (25) \tag{25}$$

Sensitivity is the ratio of true positives to the sum of true positives and false negatives, as shown in Equation 26.

$$Sensitivity = \frac{True\ Positives}{True\ Positives + False\ Negative} \tag{26}$$

F1-score is a measure that combines precision and sensitivity into a single metric, as illustrated in Equation 27.

$$F1 - score = \frac{2 \times (Sensitivity \times Precision)}{Sensitivity + Precision} \tag{27}$$

These metrics are used to evaluate the performance of classification models. Precision measures the proportion of true positive predictions among all positive predictions. Sensitivity measures the ability of the model to identify true positive samples. The F1-score provides a balanced assessment of precision and sensitivity. AUC provides a comprehensive evaluation of model performance across different thresholds.

To evaluate the CL capability of a model $M_t$ that has learned a set of tasks up to time t, denoted as $T_{1:t} = \{T_1, T_2, \ldots, T_t\}$, several metrics have been introduced to assess the degree of continual learning.

Average Accuracy (David Lopez-Paz, 2017), measures the average test accuracy of the model $M_t$ on task j after completion of task $T_t$. It is calculated as shown in Equation 28.

$$Average\ \ Accuracy(M_t) = \frac{1}{T_t} \sum_{j=1}^{T_t} acc(M_t, j) \tag{28}$$

where $T_t$ is the number of tasks completed by the model at time t and $acc(M_t, j)$ is the accuracy of the model on the test set after completing task j. The Average Accuracy metric reflects the extent to which the model has mastered the current task after learning task $T_t$.

## 2.3 Experimental setup

In this study, we compared the performance of several advanced deep learning models as backbone networks for a multi-disease continual learning image classifier. The models used for comparison include:

1. Efficientnet-Lite0 (Tan and Le, 2019), A lightweight convolutional neural network optimized for mobile and edge devices.
2. Regnetx-02 (Radosavovic et al., 2020): A new network design paradigm focusing on parametrizing populations of networks, with the RegNet design space providing simple

and fast networks that outperform EfficientNet models while being faster on GPUs.

3. ConvNeXt-S (Liu et al., 2022): A pure ConvNet model family that competes with Transformers in accuracy and scalability for computer vision tasks, achieving high performance on ImageNet classification and outperforming Swin Transformers on various benchmarks.
4. ViT-S/16 (Dosovitskiy et al., 2020): The model we used, which segments the image into fixed-size blocks and processes them using the Transformer architecture.

In evaluating the classification capabilities of the ViT-S/16 model and other models, we conducted joint learning experiments where all image categories were trained simultaneously on the Training set. During the training process, evaluations were carried out on the Validation set, and tests were conducted on the Testing set. In this setup, each model underwent a certain number of iterations, termed "epochs." In our joint learning experiments, the models were trained for 20 epochs. The training process of the model involved updating its parameters to minimize the discrepancy between the predicted and actual outputs. This procedure employed an optimization algorithm. For our joint learning, we utilized the Adaptive Moment Estimation with Decoupled Weight Decay (AdamW) optimizer, while the AdaMax optimizer was used for incremental learning. The Adam optimizer is a commonly used optimization algorithm in deep learning. Furthermore, we conducted a series of class incremental learning experiments on PlantDiseaseCL using the ViT-S/16 model. In the incremental learning process, the entire dataset was divided into training and testing sets, and segmented into 3-steps and 5-steps learning processes, as shown in Figure 2. The model learned the training data of each phase in sequences of 10 epochs and evaluated the Average Accuracy of all learned categories on the testing set after each training step. The experimental design of the 3-steps and 5-steps learning processes assessed our proposed ViT-TV method against other continual learning methods. The considered continual learning methods include:

**Baseline:**

1. Finetuning: An approach where the model is retrained on new data without using any continual learning methods, which may lead to catastrophic forgetting.
2. Freezing: A method that preserve prior task-related knowledge by halting the weight updates of specific layers after the completion of a designated task. In this context, for instance, upon concluding the first task (identified as task index 0), the principal component of the model—often the backbone or lower-level feature extractor—will cease to undergo weight updates. Subsequently, when initiating training for the second task (indexed as task 1), the frozen components will remain unaltered, without any further updates. Nevertheless, the head portion of the model—typically the classifier component—will continue to undergo weight updates to accommodate the requirements of the new task.

**Exemplar Replay Approach:**

1. Class implementing the End-to-end Incremental Learning (EEIL) (Castro et al., 2018): An approach to learn deep neural networks incrementally using new data and a small exemplar set from old classes, integrating distillation measures and cross-entropy loss.
2. Class implementing the Incremental Classifier and Representation Learning (iCaRL) (Rebuffi et al., 2017): a training strategy that enables learning an increasing number of concepts over time from a stream of data in a class-incremental manner. It learns robust classifiers and data representations concurrently, allowing it to gradually acquire knowledge of numerous classes over an extended period, in contrast to alternative strategies that tend to falter quickly.
3. Class implementing the Class Incremental Learning With Dual Memory (IL2M) (Belouadah and Popescu, 2019): A class incremental learning method using fine-tuning and a dual memory system to mitigate catastrophic forgetting, introducing a second memory to store past class statistics.

**Zero-Exemplar Approach:**

1. Learning a Unified Classifier Incrementally via Rebalancing (Lucir) (Hou et al., 2019): A learning method that

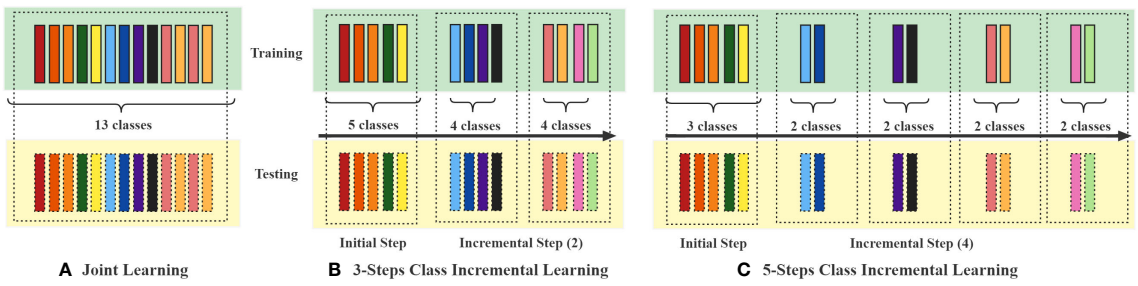



FIGURE 2
Schematic representation of the experimental setups for Joint learning and 3-steps and 5-steps class-incremental learning. Each coloured block represents a specific crop disease. In **(A)** Joint learning, all categories are trained simultaneously, whereas in the **(B)** 3-steps and **(C)** 5-steps setups, the 13 diseases are divided into 3 and 5 phases for class-incremental learning, respectively.

rebalances the loss function to balance the learning of new and old tasks.

2. Memory Aware Synapses (MAS) (Aljundi et al., 2018): A method that protects prior task knowledge by measuring parameter importance.
3. Synaptic Intelligence (SI) (Zenke et al., 2017): A method that protects prior task knowledge by measuring the importance of each synapse (i.e., connection) in the neural network.
4. Riemannian Walk (RWalk) (Chaudhry et al., 2018): A random walk method used to explore the parameter space and preserve important features.
5. Learning without Forgetting (LwF) (Li and Hoiem, 2017): A method based on the idea of global model function regularization, preserving knowledge through knowledge distillation.
6. Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017): A method based on the idea of elastic weight sharing, using the Fisher matrix to store importance parameters for balancing learning between new and old tasks.
7. ViT-TV: Our proposed approach that preserves prior task knowledge by minimizing the TV distance between the attention matrices of new and old tasks, promoting consistent attention regularization.

The algorithmic improvements and assessments are based on the Towards Exemplar-Free Continual Learning in Vision Transformers study (Pelosin et al., 2022), conducted on CIFAR-100 and ImageNet datasets, and benchmarked using the FACIL continual learning evaluation framework (Masana et al., 2023). Experiments were carried out on an NVIDIA V100 GPU utilizing the PyTorch framework, a renowned open-source deep learning platform celebrated for its ease in training and deploying deep learning models.

# 3 Results

## 3.1 Joint training results

### 3.1.1 Accuracy results

After conducting an analysis of the joint training results for various models, it is evident that there are significant differences in their performance. Table 2 presents these findings, with each model evaluated based on important metrics such as Precision, Recall, F1-score, and Accuracy, all expressed in percentage terms.

The ViT-S/16 model emerges as a paragon of excellence, demonstrating superior performance when juxtaposed with other models. It achieves a precision of 95.60%, a recall of 95.32%, an F1-score of 95.28%, and an accuracy of 95.38%. The foundational model, Efficientnet b0 Lite, lags considerably across all metrics. ViT-S/16 surpasses it by a remarkable margin: 19.37% in precision, 20.00% in recall, 20.19% in F1-score, and 19.94% in accuracy. While Regnetx-02 manages to outdo Efficientnet b0 Lite, it remains in the shadow of ViT-S/16's prowess. ViT-S/16 outshines Regnetx-02 by 3.02% in precision, 3.11% in recall, 3.09% in F1-score, and 3.04% in accuracy. ConvNeXt-S, despite performing closely to ViT-S/16, still falls short. ViT-S/16 retains a lead with an advantage of 1.85% in precision, 1.71% in recall, 1.66% in F1-score, and 1.73% in accuracy. As further evidenced by the confusion matrix depicted in Figure 3, ViT-S/16 exhibits the lowest error rate in recognizing each category.

### 3.1.2 Various diseases classification results

In the experiment of plant disease classification, the ViT-S/16 model's prowess, as detailed in Table 3, is marked by its precision across diverse leaf species and their respective diseases. For apple leaves, the model excels in distinguishing healthy ones with a precision of 98.22%, a sensitivity of 99.20%, and an F1-score of 98.71%. Black Rot's detection closely mirrors this performance, albeit slightly lower in precision at 98.01%. Rust and Scab categories exhibit comparable efficacy, with F1-scores of 98.07% and 98.71%, respectively. Corn leaves present an interesting spread: while healthy leaves and Common rust achieved near-perfect scores with F1-scores at 99.79% and 99.37%, the Gray Leaf Spot and Northern Leaf Blight categories recorded 95.62% and 96.89%, respectively. The model's proficiency extends to pepper leaves, where it identifies healthy leaves with an F1-score of 95.13% and Bacterial Spot at 97.60%. Potato leaves classification emphasizes the model's capability, especially in the Early Blight category, which stands out with a stellar F1-score of 99.38%.

Deep learning models inherently have the capability to autonomously distill representative features from images. The caliber of these extracted features fundamentally influences the ensuing classification performance. To rigorously assess the feature quality, we procured models from three predominant deep learning image classification paradigms: CNN and ViT. We extracted the penultimate feature vectors by tapping into the last layer of each model's feature extractor, producing multidimensional vectors. These vectors were subsequently projected onto a two-dimensional plane employing the t-SNE dimensionality reduction technique (Van Der Maaten and Hinton, 2008).

TABLE 2 Comparison of the backbone models of joint training.

| Backbone Model | Evaluation Metrics | | | |
|---|---|---|---|---|
| | Precision | Recall | F1-score | Accuracy |
| Efficientnet b0 Lite | 0.7623 | 0.7532 | 0.7509 | 0.7544 |
| Regnetx-02 | 0.9258 | 0.9221 | 0.9219 | 0.9234 |
| ConvNeXt-S | 0.9375 | 0.9361 | 0.9362 | 0.9365 |
| ViT-S/16 | 0.9560 | 0.9532 | 0.9528 | 0.9538 |

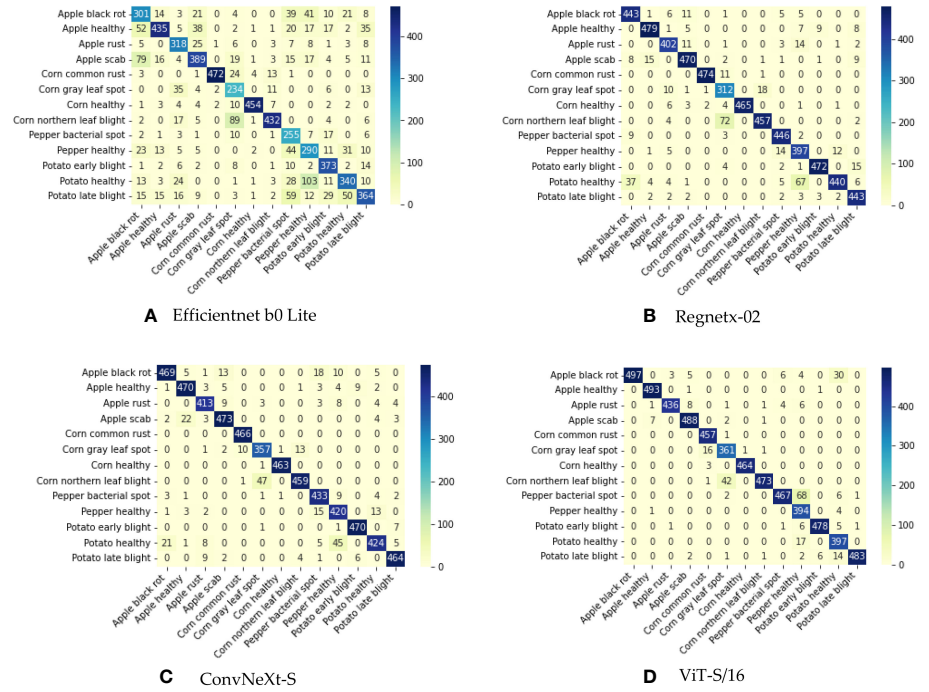



FIGURE 3
Confusion matrix illustrating the outcomes of joint training. **(A)** Confusion matrix for EfficientNet B0 Lite. **(B)** Confusion matrix for RegNetX-02. **(C)** Confusion matrix for ConvNeXt-S. **(D)** Confusion matrix for ViT-S/16.

Figure 4 graphically represents the t-SNE outcomes for various models, with distinct colorations symbolizing different disease categories. Analyzing these t-SNE feature distribution plots proffers enlightening conclusions. The scatter plots derived from the Efficientnet b0 Lite, Regnetx-02, and ConvNeXt-S models manifest an overlap, delineating an absence of discernible boundaries between different classes. Such intertwined high-dimensional features potentially complicate the task for subsequent classifiers, leading to subpar classification accuracy.

In contrast, the feature distribution from ViT-S/16 stands out. There's a clearer distinction between different classes of features. This striking separation highlights the ability of ViT-S/16 not only

TABLE 3 Results of ViT-S/16 classification for different diseases.

| Leaf Species | Disease Type | Evaluation Metrics | | |
|---|---|---|---|---|
| | | Precision | Sensitivity | F1-score |
| Apple | Healthy | 0.9822 | 0.9920 | 0.9871 |
| | Black Rot | 0.9801 | 0.9920 | 0.9860 |
| | Rust | 0.9818 | 0.9795 | 0.9807 |
| | Scab | 0.9842 | 0.9901 | 0.9871 |
| Corn | Healthy | 0.9957 | 1.0000 | 0.9979 |
| | Gray Leaf Spot | 0.9287 | 0.9854 | 0.9562 |
| | Common rust | 0.9979 | 0.9895 | 0.9937 |
| | Northern Leaf Blight | 0.9934 | 0.9455 | 0.9689 |
| Pepper | Healthy | 0.9392 | 0.9638 | 0.9513 |
| | Bacterial Spot | 0.9730 | 0.9791 | 0.9760 |
| Potato | Healthy | 0.9815 | 0.9320 | 0.9561 |
| | Early Bight | 1.0000 | 0.9876 | 0.9938 |
| | Late Blight | 0.9814 | 0.9814 | 0.9814 |

to reduce intra-class variability, but also to effectively separate feature embeddings.

## 3.2 Continual learning results

### 3.2.1 Average accuracy

In our continual learning research, we compared the average accuracy of class-incremental learning based on the ViT-S/16 model on the PlantDiseaseCL dataset under various strategies, as shown in Table 4. For baseline strategies, we explored two primary methods:

1. Fine-tuning, which eschews any continual learning techniques and solely relies on the original ViT model for continual learning. This approach achieved average accuracies of 0.4531 and 0.3799 for 3-Steps and 5-Steps learning, respectively.
2. The freezing strategy, which exhibited slightly inferior performance, with average accuracies of 0.3301 and 0.2000, respectively.

When employing the Exemplar Replay Approach, the system can retain a certain number of samples for subsequent learning. In our experiments, the maximum number of stored samples for all these methods was set at 20. Among them, ICaRL led the pack with scores of 0.6488 and 0.5216, marking an improvement of 8.39% and 4.3% over its counterparts, EEIL and IL2M strategies, respectively.

However, the most salient results were observed under our proposed Zero-Exemplar Approach utilizing the TV method. Remarkably, despite not necessitating the storage of any exemplar samples, this approach achieved average accuracies of 0.7077 and 0.5661 for 3-Steps and 5-Steps learning, respectively. Not only did this significantly outperform other strategies that don't employ exemplar replay (compared to LUCIR, the TV strategy improved by 37.54% and 36.63% for 3-Steps and 5-Steps, respectively; and when juxtaposed with SI, MAS, EWC, and LwF strategies, the gains were 25.20%, 24.19%, 23.66%, and 11.88% for 3-Steps, and 15.31%, 16.71%, 15.53%, and 25.18% for 5-Steps, respectively), but more notably, the TV strategy, even without using exemplar samples, outperformed some strategies that did. For instance, compared to ICaRL, the TV strategy improved by 5.89% in 3-Steps learning. This

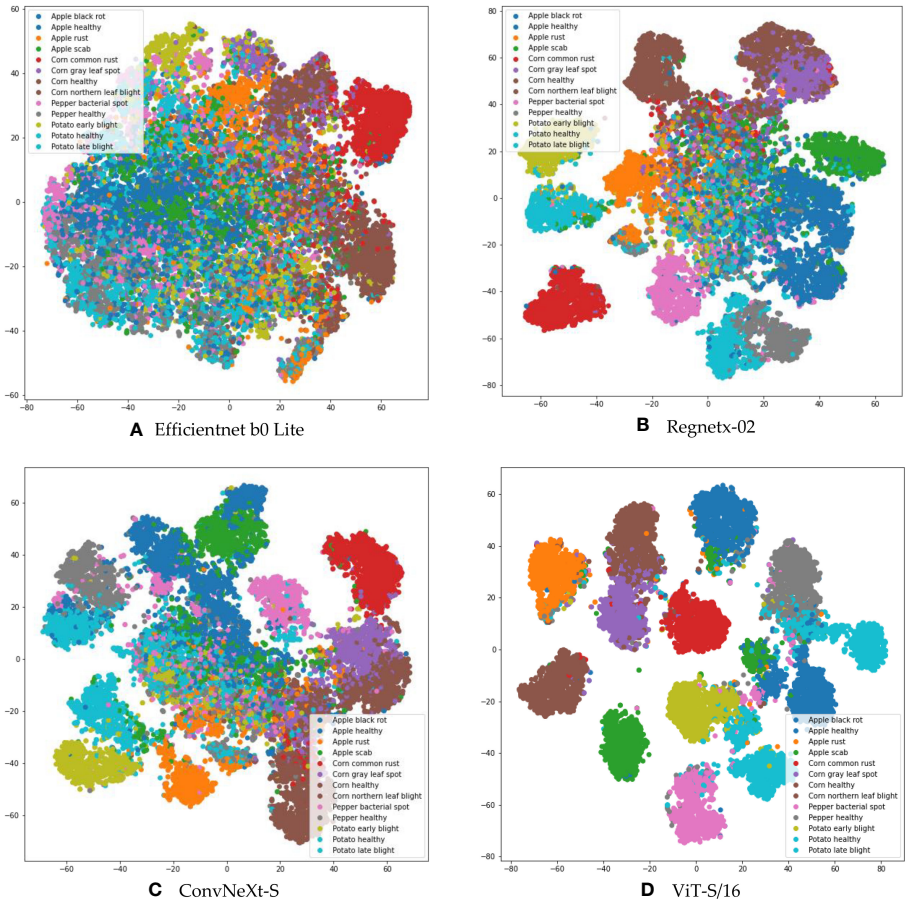



FIGURE 4
Feature space visualization of various models, depicting **(A)** t-SNE analysis for EfficientNet B0 Lite, **(B)** t-SNE analysis for RegNetX-02, **(C)** t-SNE analysis for ConvNeXt-S, and **(D)** t-SNE analysis for ViT-S/16.

TABLE 4 Average accuracy results for class-incremental learning on PlantDiseaseCL (Based on ViT-S/16).

| Strategy | Methods | Examples | Average Accuracy | |
|---|---|---|---|---|
| | | | 3-Steps | 5-Steps |
| Baseline | Freezing | 0 | 0.3301 | 0.2000 |
| | Finetuning | 0 | 0.4531 | 0.3799 |
| Exemplar Replay Approach | EEIL | 20 | 0.5649 | 0.3479 |
| | IL2M | 20 | 0.6058 | 0.4990 |
| | ICaRL | 20 | 0.6488 | 0.5216 |
| Zero-Exemplar Approach | LUCIR | 0 | 0.3323 | 0.1998 |
| | SI | 0 | 0.4557 | 0.4130 |
| | MAS | 0 | 0.4658 | 0.3990 |
| | EWC | 0 | 0.4741 | 0.4108 |
| | RWalk | 0 | 0.4819 | 0.3961 |
| | LwF | 0 | 0.5889 | 0.3143 |
| | TV(ours) | 0 | 0.7077 | 0.5661 |

is a significant finding as, conventionally, strategies employing exemplar samples in class-incremental learning tend to exhibit superior continual learning performance.

### 3.2.2 Incremental learning processes results

In the realm of continual learning, ensuring consistent performance improvement during the incremental learning phase stands as one of the foremost challenges, especially when evaluating against diverse benchmarks. To delve deeper into this process, we employ the ViT-S/16 model and present the evolution of class-incremental learning performance on the PlantDiseaseCL dataset.

Figure 5 reveals subtle distinctions among various continual learning strategies during the 3-Steps and 5-Steps learning phases. The left panel represents the 3-Steps evaluation, unveiling pronounced disparities in strategy effectiveness. Likewise, the right panel encapsulates a broader 5-Steps progression, reinforcing these observations.

Upon a more thorough analysis of these metrics, although most strategies exhibit near-identical accuracy levels in the initial phase, the trajectory diverges thereafter. Approaches such as Freezing and LUCIR appear to respond inadequately to the challenges of continual learning, with accuracy sharply declining as steps progress. Conversely, strategies like IL2M and ICaRL manifest a more gradual decline. However, even within these methods, the rate of degradation varies.

Next, we turn to our proposed strategy, TV (ours). Notably, it not only maintains its momentum but can be argued to exhibit the slowest rate of average accuracy decline at each stage. As stages progress, TV (ours) consistently excels in retaining knowledge and adapting to new information. In the 3-Steps scenario, it achieves an

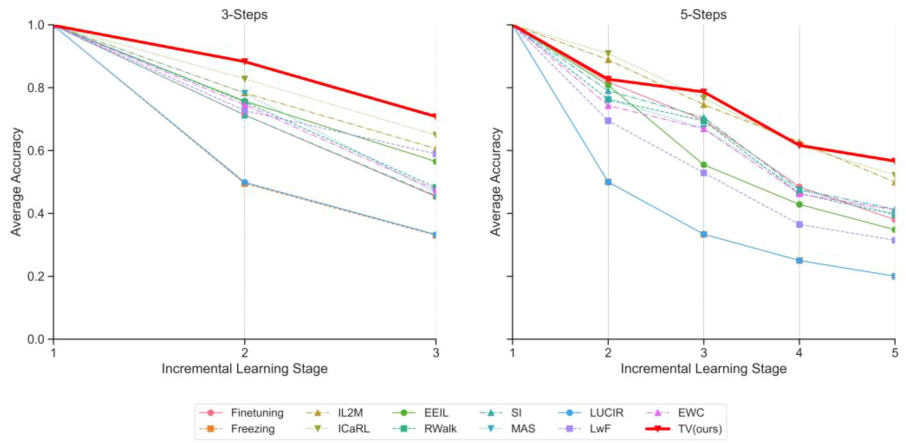



FIGURE 5
Class-incremental learning performance evolution on PlantDiseaseCL: Reported Top-1 average accuracy after each stage learning. Left figure shows evaluation with 3 steps, while the right figure shows evaluation with 5 steps (Based on ViT-S/16).

admirable accuracy of 0.7077 in the third stage, surpassing its closest competitor by a substantial margin. In the 5-Steps evaluation, the TV approach similarly takes the lead, achieving the highest final average accuracy score of 0.5661.

In summary, our TV strategy demonstrates exceptional performance across stages. Its prowess is particularly evident in the achieved final average accuracy scores, outperforming competitors significantly in both 3-Steps and 5-Steps evaluations.

### 3.2.3 Comparative experiments on variants of multiple attention alignment methods

In the Continual Learning segment, assessing the efficacies of various attention alignment methods, especially under consistent ViT conditions, is of paramount importance. The results shed light on how different attention alignment techniques, when based on diverse distance metrics, influence the overall learning performance as shown in Table 5. Interestingly, all variants of attention alignment methods commence their journey from a nearly similar starting point, but the divergence becomes evident in subsequent stages. A notable observation is the performance of the ' Original ViT'. Despite being the foundational model, its average accuracy is only 0.4531 and 0.3799 for 3-Steps and 5-Steps respectively, which points towards the significance of integrating additional distance metrics for enhanced performance.

Notably, methods employing 'JS Divergence' and 'Bhattacharyya Distance' exhibited significant improvements. The principle behind the 'JS Divergence' method is rooted in the Jensen-Shannon divergence metric. For two given probability distributions, P and Q, it first calculates their relative entropy with respect to their average distribution, yielding a measure of divergence for each distribution from the average. The average of these two relative entropies is then taken as the Jensen-Shannon divergence, serving as a measure of the difference between P and Q.

The 'Bhattacharyya Distance', on the other hand, is a metric designed to gauge the similarity between two probability distributions. It involves taking the square root of each element of the two distributions, multiplying them pairwise, and then summing up all the products. The negative logarithm of this sum is then taken. This value, which essentially represents the cross-entropy between the two distributions, quantifies the amount of information shared between them. A smaller Bhattacharyya Distance indicates greater similarity between the distributions, and vice versa. Our experimental results underscored the efficacy of both the JS Divergence and Bhattacharyya Distance methods. Particularly, the 'Bhattacharyya Distance' method manifested a significant accuracy enhancement of 16.05% and 16.28%.

Furthermore, in the 'ViT + Hellinger Distance' method, we utilized the Hellinger distance, which measures the similarity between two probability distributions by calculating the Euclidean distance of their square roots. During computation, we introduced a normalization factor of $\left(\frac{1}{\sqrt{2}}\right)$. Surprisingly, this method exhibited a declining trend in performance across two distinct steps, registering drops of 12.03% and 17.99% respectively when compared to the Original ViT. This suggests that not all attention alignment techniques universally yield positive outcomes in such contexts.

However, the true standout is our proposed 'ViT+TV Distance' method. Demonstrating consistent superiority over other techniques, it achieved an average accuracy of 0.7077 for 3-Steps and an impressive 0.5661 for 5-Steps. These figures not only highlight the robustness and supremacy of the TV Distance in attention alignment but also accentuate its potential in striking an optimal balance between accuracy and adaptability in continual learning environments.

TABLE 5 Comparative results on variants of multiple attention alignment methods.

| Methods | 3-Steps Average Accuracy | 5-Steps Average Accuracy |
|---|---|---|
| Original ViT | 0.4531 | 0.3799 |
| ViT + JS Divergence | 0.6106 | 0.5280 |
| ViT + Hellinger Distance | 0.3328 | 0.2000 |
| ViT + Bhattacharyya Distance | 0.6136 | 0.5427 |
| ViT+TV Distance | 0.7077 | 0.5661 |

## 4 Discussion

In this study, we introduce a novel mathematical paradigm for continual learning in the domain of crop disease defect recognition. By proposing the innovative ViT-TV framework, we further amplify our contribution, addressing the challenges of multi-disease image recognition in crops within the ViT architecture. We employ the Total Variation distance loss (TV-Loss) to quantify the disparity between current and prior attention distributions, fostering attention consistency and mitigating the catastrophic forgetting inherent to ViT in the absence of prior task samples. With this new framework, we offer a solution for continual learning in intricate scenarios like crop disease recognition.

Distinctively, the ViT-TV method bridges the gap between stability and plasticity in model learning. By incorporating TV-Loss into its internal architecture and co-optimizing TV-Loss with cross-entropy loss, it ensures attention consistency when assimilating new tasks, allowing the model to adapt and learn without significantly compromising previously acquired knowledge. Retaining historical knowledge is paramount for accurate and reliable disease recognition in crops, marking a significant stride forward.

Compared to established Zero-Exemplar Approach types of continual learning techniques: SI focuses on safeguarding synaptic weights to alleviate catastrophic forgetting, EWC protects vital knowledge by regularizing the network's global weights, and LwF relies on knowledge transfer techniques from the theory of knowledge distillation, ViT-TV stands out by addressing attention consistency. Maintaining attention consistency becomes crucial in the domain of food and crop disease image recognition, especially when confronted with subtle variations in different disease manifestations. The ViT-TV framework, grounded on TV

distance and attention alignment, offers superior average accuracy metrics in 3-step and 5-step class incremental learning experiments on PlantDiseaseCL by holistically considering attention consistency, stability, and knowledge preservation, presenting a theoretically robust and practically effective approach to maintaining model stability when recognizing multiple diseases.

Further juxtaposing the ViT-TV framework with exemplar replay methods (e.g., ICaRL) accentuates the superiority of our approach. While ICaRL adeptly uses exemplar samples to combat forgetting, the ViT-TV framework obviates the need for sample storage. The philosophy underpinning our method posits that attention consistency based on Total Variation distance plays a pivotal role in memory retention and transfer across tasks. Unlike methods predominantly reliant on archiving exemplar samples to counteract forgetting, ViT-TV captures the attention distribution of prior tasks, amalgamating it with the attention from new learning, effectively mitigating the risk of catastrophic forgetting.

In this research, we also delve deeper into how different distance metrics can be employed to regularize attention maps, optimizing model performance. The intrinsic value of attention mechanisms lies in enabling the model to focus on pivotal parts of the input, thereby capturing salient information. However, these focal points may vary with task or model iterations. Thus, selecting an apt distance metric to accentuate or diminish these differences is crucial. Integrating the TV distance into the ViT's attention mechanism offers a potent strategy for addressing the continual learning recognition challenges of multiple diseases in food and crops.

## 5 Conclusions

In summary, our ViT-TV framework establishes a pioneering approach to address the continual learning challenges in the domain of crop disease defect recognition. By adeptly leveraging attention consistency and the Total Variation distance loss, our method contributes to the intelligent evolution of the agricultural industry, ensuring that AI models possess sustainable growth and augmented disease recognition capabilities.

## Data availability statement

The original contributions presented in the study are included in the article/supplementary material. Further inquiries can be directed to the corresponding author.

## Author contributions

BW: Methodology, Writing – original draft, Writing – review & editing.

## Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This research was funded by the National Key Research and Development Program of China (No. 2022YFC2010105), and the Research on Intelligent Dietary Nutrition Assessment and Analysis System based on Deep Learning and Mobile Devices, Guangdong Provincial Medical Research Fund Project, grant number A2023159.

## Conflict of interest

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

Abraham, W. C., and Robins, A. (2005). Memory retention–the synaptic stability versus plasticity dilemma. *Trends Neurosci.* 28 (2), 73–78. doi: 10.1016/j.tins.2004.12.003

Albawi, S., Mohammed, T. A., and Al-Zawi, S. (2017). "Understanding of a convolutional neural network," in *Proceedings of the 2017 International Conference on Engineering and Technology (ICET)* (Antalya, Turkey: Institute of Electrical and Electronics Engineers (IEEE)), 21-23..

Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., and Tuytelaars, T. (2018). "Memory aware synapses: Learning what (not) to forget," in *Computer vision – ECCV 2018. Lecture notes in computer science*. Eds.V. Ferrari, M. Hebert, C. Sminchisescu and Y. Weiss. (Cham:Springer), 11207. doi: 10.1007/978-3-030-01219-9_9

Araujo, V., Hurtado, J., Soto, A., and Moens, M.-F. (2022). "Entropy-based stability-plasticity for lifelong learning," in *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (New Orleans, LA, USA: Institute of Electrical and Electronics Engineers (IEEE)), 3720-3727. doi: 10.1109/CVPRW56347.2022.00416

Banerjee, I., and Madhumathy, P. (2022). "IoT based agricultural business model for estimating crop health management to reduce farmer distress using SVM and machine learning," in *Internet of things and analytics for agriculture*, vol. Volume 3 . Eds. P. K. Pattnaik, R. Kumar and S. Pal (Singapore: Springer Singapore), 165–183. doi: 10.1007/978-981-16-6210-2_8

Belouadah, E., and Popescu, A. (2019). "IL2M: class incremental learning with dual memory," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. (Seoul, Korea (South): Institute of Electrical and Electronics Engineers (IEEE)), 583-592. doi: 10.1109/ICCV.2019.00067

Bhimte, N. R., and Thool, V. R. (2018). "Diseases detection of cotton leaf spot using image processing and SVM classifier," in *Proceedings of the Second International Conference on Intelligent Computing and Control Systems (ICICCS 2018)* (Madurai, India: Institute of Electrical and Electronics Engineers (IEEE)), 340-344. doi: 10.1109/ICCONS.2018.8662906

Bhojanapalli, S., Chakrabarti, A., Veit, A., Lukasik, M., Jain, H., Liu, F., et al. (2021). Leveraging redundancy in attention with Reuse Transformers. *CoRR*. abs/2110.06821. doi: 10.48550/arXiv.2110.06821

Boulent, J., Foucher, S., Théau, J., and St-Charles, P.-L. (2019). Convolutional neural networks for the automatic identification of plant diseases. *Front. Plant Sci.* 10. doi: 10.3389/fpls.2019.00941

Castro, F. M., Marín-Jiménez, M. J., Guil, N., Schmid, C., and Alahari, ,. K. (2018). "End-to-end incremental learning," in *Computer vision – ECCV 2018*. Eds. V. Ferrari, M. Hebert, C. Sminchisescu, Y. and Weiss.. (Munich, Germany: Springer, Cham), 2018.

Chakraborty, S., Paul, S., and Rahat-uz-Zaman, M. (2021). "Prediction of apple leaf diseases using multiclass support vector machine," *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)* (Dhaka: Institute of Electrical and Electronics Engineers (IEEE)), 2021, pp. 147-151.

Chaudhry, A., Dokania, P. K., Ajanthan, T., and Torr, P. H.S. (2018). "Riemannian walk for incremental learning: understanding forgetting and intransigence," in *Computer Vision – ECCV 2018. Lecture notes in computer science, vol 11215*. Eds. V. Ferrari, M. Hebert, C. Sminchisescu, Y. and Weiss.. (Munich, Germany: Springer Cham). doi: 10.1007/978-3-030-01252-6_33

David Lopez-Paz, M. A. R. (2017). "Gradient episodic memory for continual learning," in *Advances in neural information processing systems (NIPS 2017)* (Long Beach, California, USA: Curran Associates Inc.), 6470–6479.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Gelly, T. U. M. D. M. G. H. S., et al. (2020). "An image is worth 16x16 words: transformers for image recognition at scale," *9th International Conference on Learning Representations, ICLR 2021* (Virtual Event, Austria: OpenReview.net) abs/2010.11929. https://arxiv.org/abs/2010.11929

Godi, B., Muttipati, A. S., Rao, M. P., Satyanarayana, G. V., Padmaja, G. M., and Gadiraju, S. R. (2022a). "ResNet model to forecast plant leaf disease," *Proceedings of the 2022 International Conference on Computing, Communication and Power Technology (IC3P)* (Visakhapatnam, India: Institute of Electrical and Electronics Engineers (IEEE)), 38-43. doi: 10.1109/IC3P52835.2022.00017

Habiba, S. U., and Islam, M. K. (2021). "Tomato plant diseases classification using deep learning based classifier from leaves images," in *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)* (Dhaka, Bangladesh: Institute of Electrical and Electronics Engineers (IEEE) Conference), 82-86. doi: 10.1109/ICICT4SD50815.2021.9396883

Hadsell, R., Rao, D., Rusu, A. A., and Pascanu, R. (2020). Embracing change: continual learning in deep neural networks. *Trends Cognit. Sci.* 24 (12), 1028–1040. doi: 10.1016/j.tics.2020.09.004

Haque, M. A., Marwaha, S., Deb, C. K., Nigam, S., Arora, A., Hooda, K. S., et al. (2022). Deep learning-based approach for identification of diseases of maize crop. *Sci. Rep.* 12 (1), 6334. doi: 10.1038/s41598-022-10140-z

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV, USA: Institute of Electrical and Electronics Engineers (IEEE)), 770-778. doi: 10.1109/CVPR.2016.90.

Hou, S., Pan, X., Loy, C. C., Wang, Z., and Lin, D. (2019). "Learning a unified classifier incrementally via rebalancing," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (Long Beach, CA, USA: IEEE), 831-839. doi: 10.1109/CVPR.2019.00092

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). MobileNets: efficient convolutional neural networks for mobile vision applications. *ArXiv* abs/1704.04861. doi: 10.48550/arXiv.1704.04861

Jones, R. A. C. (2021). Global plant virus disease pandemics and epidemics. *Plants* 10 (2), 233. doi: 10.3390/plants10020233

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci.* 114 (13), 3521–3526. doi: 10.1073/pnas.1611835114

Kusumo, B. S., Heryana, A., Mahendra, O., and Pardede, H. F. (2018). "Machine learning-based for automatic detection of corn-plant diseases using image processing," in *2018 International Conference on Computer, Control, Informatics and its Applications (IC3INA)* (Tangerang, Indonesia: IEEE), 93-97. doi: 10.1109/IC3INA.2018.8629507

Lesort, T., LoMonaco, V., Stoian, A., Maltoni, D., Filliat, D., and Díaz-Rodríguez, N. (2020). Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Inf. Fusion* 58, 52–68. doi: 10.1016/j.inffus.2019.12.004

Li, X., and Li, S. (2022). Transformer help CNN see better: A lightweight hybrid apple disease identification model based on transformers. *Agriculture* 12 (6), 884. doi: 10.3390/agriculture12060884

Li, Z., and Hoiem, D. (2017). Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* 40 (12), 2935–2947. doi: 10.1109/TPAMI.2017.2773081

Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., and Xie, S. (2022). "A convNet for the 2020s," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (New Orleans, LA, USA: IEEE), 11966-11976. doi: 10.1109/CVPR52688.2022.01167

Liu, J., and Wang, X. (2021). Plant diseases and pests detection based on deep learning: a review. *Plant Methods* 17 (1), 22. doi: 10.1186/s13007-021-00722-9

Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A. D., and Weijer, (2023). Class-incremental learning: survey and performance evaluation on image classification. *IEEE Trans. Pattern Anal. Mach. Intell* 45, 5513-5533. doi: 10.1109/TPAMI.2022.3213473

McCloskey, M., and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: the sequential learning problem. *Psychol. Learn. Motivation* Vol. 24, 109–165. doi: 10.1016/S0079-7421(08)60536-8

Mekha, P., and Teeyasuksaet, N. (2021). "Image classification of rice leaf diseases using random forest algorithm," in *2021 Joint international conference on digital arts, media and technology with ecti northern section conference on electrical, electronics, computer and telecommunication engineering*. (Cha-am, Thailand: IEEE), 165-169. doi: 10.1109/ECTIDAMTNCON51128.2021.9425696

Nema, S., and Dixit, A. (2018). "Wheat leaf detection and prevention using support vector machine," in *2018 international conference on circuits and systems in digital enterprise technology (ICCSDET)* (Kottayam, India: IEEE), 1-5. doi: 10.1109/ICCSDET.2018.8821098

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks* 113, 54–71. doi: 10.1016/j.neunet.2019.01.012

Pelosin, F., Jha, S., Torsello, A., Raducanu, B., and Weijer, J. (2022). "Towards exemplar-free continual learning in vision transformers: an account of attention, functional and weight regularization," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (New Orleans, LA, USA: IEEE), 3819-3828. doi: 10.1109/CVPRW56347.2022.00427.

Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., and Dollár, P. (2020). "Designing network design spaces," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Seattle, WA, USA: IEEE), 10425-10433. doi: 10.1109/CVPR42600.2020.01044

Rajbongshi, A., Sarker, T., Ahamad, M. M., and Rahman, M. M. (2020). "Rose diseases recognition using mobileNet," in *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)* (Istanbul, Turkey: IEEE), 1-7. doi: 10.1109/ISMSIT50672.2020.9254420

Rebuffi, S. A., Kolesnikov, A., Sperl, G., and Lampert, C. H. (2017). "iCaRL: incremental classifier and representation learning," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI, USA: IEEE) 2017, pp. 5533-5542. doi: 10.1109/CVPR.2017.587

Ristaino, J. B., Anderson, P. K., Bebber, D. P., Brauman, K. A., Cunniffe, N. J., Fedoroff, N. V., et al. (2021). The persistent threat of emerging plant disease pandemics to global food security. *Proc. Natl. Acad. Sci.* 118 (23), e2022239118. doi: 10.1073/pnas.2022239118

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena* 60 (1), 259–268. doi: 10.1016/0167-2789(92)90242-F

Savary, S., Willocquet, L., Pethybridge, S. J., Esker, P., McRoberts, N., and Nelson, A. (2019). The global burden of pathogens and pests on major food crops. *Nat. Ecol. Evol.* 3 (3), 430–439. doi: 10.1038/s41559-018-0793-y

Simonyan, K., and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *CoRR*. abs/1409.1556. doi: 10.48550/arXiv.1409.1556

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). "Rethinking the inception architecture for computer vision," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV, USA: IEEE), 2818-2826. doi: 10.1109/CVPR.2016.308

Tan, M., and Le, Q. (2019). "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning, Proceedings of Machine Learning Research* (Long Beach, California, USA: MLResearchPress), 6105-6114. doi: 10.48550/arXiv.1905.11946

Van Der Maaten, L., and Hinton, G. (2008). Viualizing data using t-SNE. *J. Mach. Learn. Res.* 9, 2579–2605.

Vijayalata, Y., Billakanti, K., Veeravalli, A. D. R. N., and Kota, L. (2022). "Early detection of casava plant leaf diseases using efficientNet-B0," *2022 IEEE Delhi Section Conference (DELCON)* (New Delhi, India: Institute of Electrical and Electronics Engineers (IEEE)), 2022 1-5. doi: 10.1109/DELCON54057.2022.9753210

Wu, G., Gong, S., and Li, P. (2021). "Striking a balance between stability and plasticity for class-incremental learning," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. (Montreal, QC, Canada: IEEE) 2021, pp. 1104-1113. doi: 10.1109/ICCV48922.2021.00116

Yang, G., He, Y., Yang, Y., and Xu, B. (2020). Fine-grained image classification for crop disease based on attention mechanism. *Front. Plant Sci.* 11. doi: 10.3389/fpls.2020.600854

Zenke, F., Poole, B., and Ganguli, S. (2017). "Continual learning through synaptic intelligence," in *Proceedings of the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research* (International Convention Centre, Sydney, Australia: MLResearchPress), 3987-3995.

Zhang, Z., Gong, Z., Hong, Q., and Jiang, L. (2021). "Swin-transformer based classification for rice diseases recognition," in *2021 International Conference on Computer Information Science and Artificial Intelligence (CISAI)* (Kunming, China: IEEE) 2021, pp. 153-156. doi: 10.1109/CISAI54367.2021.00036

Zhuang, L. (2021). "Deep-learning-based diagnosis of cassava leaf diseases using vision transformer," in *Proceedings of the 2021 4th Artificial Intelligence and Cloud Computing Conference* (Kyoto, Japan: IEEE). doi: 10.1145/3508259.3508270

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). "). Learning transferable architectures for scalable image recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT, USA: IEEE),8697–8710. doi: 10.1109/CVPR.2018.00907

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 08 January 2024
DOI 10.3389/fpls.2023.1280671

Check for updates

OPEN ACCESS

EDITED BY
Jucheng Yang,
Tianjin University of Science and Technology, China

REVIEWED BY
Guoxiong Zhou,
Central South University Forestry and Technology, China
Aibin Chen,
Central South University Forestry and Technology, China

*CORRESPONDENCE
Quan Gao
gaoq@ynau.edu.cn

RECEIVED 21 August 2023
ACCEPTED 28 November 2023
PUBLISHED 08 January 2024

CITATION
He Y, Zhang G and Gao Q (2024) A novel ensemble learning method for crop leaf disease recognition.
*Front. Plant Sci.* 14:1280671.
doi: 10.3389/fpls.2023.1280671

COPYRIGHT
© 2024 He, Zhang and Gao. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# A novel ensemble learning method for crop leaf disease recognition

Yun He[1,2], Guangchuan Zhang[2,3] and Quan Gao[1,2]*

[1]School of Big Data, Yunnan Agricultural University, Kunming, China, [2]Key Laboratory for Crop Production and Intelligent Agriculture of Yunnan Province, Yunnan Agricultural University, Kunming, China, [3]School of Mechanical and Electrical Engineering, Yunnan Agricultural University, Kunming, China

Deep learning models have been widely applied in the field of crop disease recognition. There are various types of crops and diseases, each potentially possessing distinct and effective features. This brings a great challenge to the generalization performance of recognition models and makes it very difficult to build a unified model capable of achieving optimal recognition performance on all kinds of crops and diseases. In order to solve this problem, we have proposed a novel ensemble learning method for crop leaf disease recognition (named ELCDR). Unlike the traditional voting strategy of ensemble learning, ELCDR assigns different weights to the models based on their feature extraction performance during ensemble learning. In ELCDR, the models' feature extraction performance is measured by the distribution of the feature vectors of the training set. If a model could distinguish more feature differences between different categories, then it receives a higher weight during ensemble learning. We conducted experiments on the disease images of four kinds of crops. The experimental results show that in comparison to the optimal single model recognition method, ELCDR improves by as much as 1.5 (apple), 0.88 (corn), 2.25 (grape), and 1.5 (rice) percentage points in accuracy. Compared with the voting strategy of ensemble learning, ELCDR improves by as much as 1.75 (apple), 1.25 (corn), 0.75 (grape), and 7 (rice) percentage points in accuracy in each case. Additionally, ELCDR also has improvements on precision, recall, and F1 measure metrics. These experiments provide evidence of the effectiveness of ELCDR in the realm of crop leaf disease recognition.

KEYWORDS

crop disease, recognition, ensemble learning, weight, feature extraction performance

## 1 Introduction

Crops face continuous threats from different diseases during planting, making disease control a long-standing and crucial challenge for farmers. Early detection of crop diseases is an essential task in agriculture (Applalanaidu and Kumaravelan, 2021). In the early stage, farmers and experts relied on their knowledge and experience to diagnose crop diseases.

However, this approach is inefficient, expensive, and characterized by low accuracy. With the development of information technology, researchers began to apply machine learning (ML) and deep learning (DL) technologies for crop disease recognition. ML and DL technologies offer the potential to automate and enhance the accuracy of crop disease recognition. In recent years, DL has become the mainstream technology in the field of crop disease recognition due to its automated feature extraction, high accuracy, and user-friendliness. Many researchers tried to apply more advanced DL models to recognize diseases in different crops. For instance, Fuentes et al. (2018) used faster R-CNN to recognize tomato diseases, achieving a recognition rate of approximately 96%. Nachtigall et al. (2016) proposed a technique for apple disease recognition based on AlexNet, achieving an accuracy of 96.6%. Jiang et al. (2020) applied convolution neural networks (CNNs) to recognize four different rice diseases, achieving an accuracy of 96.8%. There are also many recognition studies on crops based on DL models, including grape (Xie et al., 2020), mango (Singh et al., 2019), millet (Coulibaly et al., 2019), olive (Cruz et al., 2017), and cucumber (Kawasaki et al., 2015). The DL models used in these studies include Xception, MCNN, VGG, LeNet, and custom CNN. In these studies, different DL models are applied by the researchers who expect that better models will bring better recognition accuracy.

There are many kinds of crops and crop diseases. While the leaves of different crops may exhibit distinct morphological features, the symptoms caused by different diseases can often appear visually similar (Ngugi et al., 2021). This brings a great challenge to the generalization performance of the recognition model. Different DL models have their own feature extraction mechanisms, resulting in different recognition performances in different crops' disease recognition. This variability also makes it difficult to build a common model that can achieve optimal recognition performance across all crop types and diseases. To solve this problem, we proposed a novel ensemble learning method for crop leaf disease recognition (named ELCDR), which can integrate different DL models to improve the generalization performance of the recognition method.

The contributions of this paper include the following key aspects:

1) We proposed a novel ensemble learning method for crop leaf disease recognition (named ELCDR). Compared with the crop disease recognition methods that are based on a single model, ELCDR demonstrates better recognition performance and generalization performance.
2) We proposed an innovative ensemble learning strategy and deployed it within ELCDR. Compared with the traditional voting strategy, our strategy can realize more reasonable ensemble learning. By this, ELCDR can achieve better recognition performance and generalization performance than the methods based on the voting strategy.
3) We executed experiments on the dataset that includes four different crop types, and the results showed the effectiveness of our methods.

## 2 Related works

The recognition of crop leaf disease images essentially constitutes an image classification. In the past, many researchers tried to achieve automated crop disease image recognition using traditional machine learning technologies. Support vector machine (SVM) is the most widely used machine learning algorithm in the research field of crop disease image recognition. Raza et al (Shan-E-Ahmed et al., 2015). proposed an SVM-based method that can detect tomato powdery mildew with an accuracy of more than 90%. Islam et al. (2017) suggested an SVM-based approach for recognizing two potato diseases, with an accuracy of 95%. Additionally, Pantazi et al. (2019) proposed a multiple crop disease recognition system based on SVM, also achieving an accuracy of 95%. On the other hand, Kaur et al. (2018) applied SVM to recognize various diseases of soybean, with the highest accuracy reaching 62.53%. Furthermore, the k-nearest neighbor (KNN) (Hossain et al., 2019), k-means (Prakash et al., 2017), transductive support vector machine (TSVM) (Ahmed et al., 2019), and multiple linear regression (MLR) (Sun et al., 2018) are also the traditional machine learning technologies that are widely applied in crop leaf disease recognition. All of these recognition methods based on traditional machine learning need to select image features manually or by using other selection algorithms. The quality of feature selection significantly impacts the performance of recognition. This leads the traditional machine learning methods to have a certain threshold for use and may have an unstable performance.

In recent years, due to developments in deep learning technologies, researchers have been incorporating deep learning models into the realm of crop disease image recognition. Deep learning models are proficient at automating feature selection and extraction, allowing for end-to-end deployment. This has led deep learning models to gradually become the mainstream methods in the field of crop disease image recognition. For instance, Jiang et al. (2019) achieved real-time disease recognition for apples using the VGG model and attained an accuracy of 78.8%. Additionally, some studies have applied VGG to recognize other crops (Paymode and Malode, 2021) (Bhagat et al., 2023) (Kundu et al., 2021). VGG is a widely applied deep learning model in crop disease recognition studies because it has a simple network structure and a smaller convolutional kernel. Researchers also introduced other deep learning models to crop leaf disease recognition, such as ResNet (Stephen et al., 2023), MobileNet (Chen et al., 2021), AlexNet (Chen et al., 2022), and GoogLeNet (Yang et al., 2023). However, the diverse array of crops and diseases poses a great challenge to the generalization performance of recognition models. This challenge

makes it very difficult to build a unified model that can achieve optimal recognition performance on all kinds of crops and diseases. When applying deep learning models to a new kind of crop or disease, researchers often need to optimize the model to adapt to the unique characteristics of that specific crop and disease (Ganesan and Chinnappan, 2022) (Reddy et al., 2023). Otherwise, the models may fail to achieve their optimal recognition performance.

In order to improve the generalization performance of the recognition method, researchers have introduced ensemble learning to image-based crop disease recognition (Li et al., 2021). Ensemble learning (Ganaie et al., 2022) is an effective way to improve the generalization performance of the recognition method. The authors of refs (Chaudhary et al., 2020) (Mathew et al., 2022) (Palanisamy and Sanjana, 2023). introduced the voting strategy of ensemble learning to crop disease recognition and observed improvements in recognition accuracy in the experiments. When we apply the voting strategy to recognize an image, each of the models costs a vote for a particular category. Then, the image is assigned to the category which receives most of the votes. Voting is the simplest and most effective ensemble learning strategy, but it treats all models as equally important. Even if a model fails to extract effective features, its vote still has significant weight. This is obviously unreasonable. Furthermore, if each model votes for a different category, it becomes challenging to determine which category should the image belong to, rendering the voting strategy ineffective.

To solve this problem, we have proposed a novel ensemble learning method for crop leaf disease recognition method, named ELCDR. Different from the traditional voting strategy, it assigns varying weights to the models during ensemble learning. These weights are determined based on the feature extraction performance of each model, which can be measured by examining the distribution of feature vectors. Using this approach, ELCDR can achieve more accurate and stable disease recognition performance across different corps. Figure 1 shows the primary differences between the voting strategy and our proposed strategy when applied in ensemble learning.

## 3 A novel ensemble learning method for crop leaf disease recognition

The basic flow of ELCDR is shown in Figure 1. There are four stages in ELCDR, and they are represented by arrows of different colors in Figure 2.

1) Stage 1 is represented by the gray arrows. We build a training set and multiple DL models at the beginning of this stage. Then, we use the training set to train each of the DL models, respectively. If there are $n$ DL models, then we will get $n$ trained models at the last of this stage.

2) Stage 2 is represented by the blue arrows. This stage's main purpose is to measure each model's feature extraction performance, and then calculate the ensemble learning weight of each model. We input the training set into each of the trained models to get the feature vector distribution of each trained model, respectively. Then, we can get each model's feature extraction performance by measuring the vector distribution of all the trained models. At last, we calculate the ensemble learning weight for each model based on their feature extraction performance. The details about how to measure the feature extraction performance of models are introduced in Section 3.1, and the details about how to calculate the ensemble learning weight are also introduced in Section 3.1. If there are $n$ trained models, then we will get their ensemble learning weight as $\omega_1$ to $\omega_n$ by this stage.

3) Stage 3 is represented by the red arrows. Once a new image is input into the trained models, we can get the softmax function output

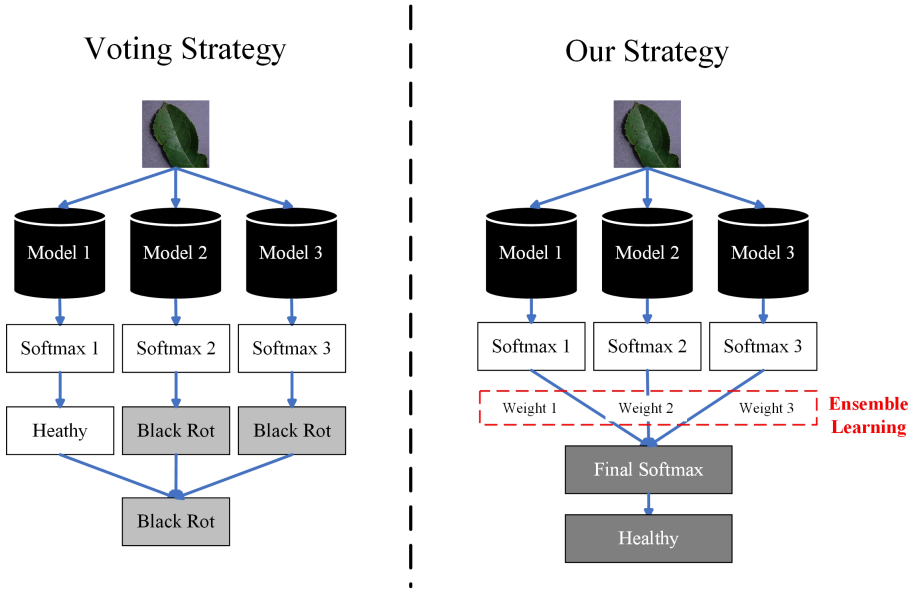



FIGURE 1
Comparison of the voting strategy and our strategy.

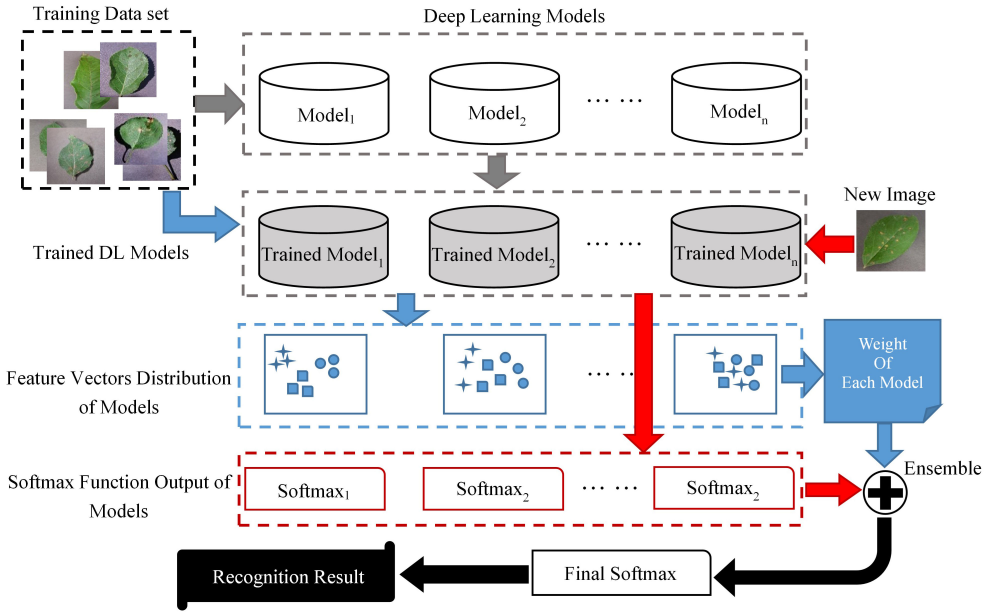



FIGURE 2
The basic flow of ELCDR.

of each model. If there are $n$ trained models, then we can extract their softmax function output as $sf_1$ to $sf_n$.

4) Stage 4 is represented by the black arrows. Based on the ensemble learning weight and the image's softmax function output, we can calculate the final softmax function output by the following formula:

$$sf_{final} = \sum_{i=1}^{n} \omega_i * sf_i \,. \tag{1}$$

In Formula (1), $n$ means the number of trained models. Lastly, we can get the new image's recognition result based on the final softmax function output. The details regarding how to calculate the final softmax function output are provided in Section 3.2.

## 3.1 Weight calculation by measuring the feature extraction performance of the models

The traditional ensemble learning method generally uses a voting weighting strategy, but we find this strategy to be irrational. Different DL models may exhibit different feature extraction performances on the same dataset. If a model could extract a more effective feature, it should be assigned more weight during the ensemble process. Otherwise, the model with less effective feature extraction should be assigned less weight. By this thought, we have introduced a novel weight calculation method that measures the feature extraction performance of the models. This method uses the vectors' distribution of the training set to measure a model's feature extraction performance. If there are $t$ images in the training set, they will be divided into $k$ categories. Once we input the training set into the DL model, we can get $t$ feature vectors corresponding to the images. Figure 3 shows an example of feature vector distribution in a two-dimensional space. In this space, there are 18 feature vectors represented by spots. All of the vectors can be divided into three categories, which we use different colors to represent. Each category has a category centroid represented by a star. The category centroid is an average vector of all vectors in the category. For a specific $category_p$, if there are total $m$ vectors in $category_p$, its centroid $cen_p$ can be calculated by the following formula:

$$cen_p = \sum_{i=1}^{m} vector_i^p \,. \tag{2}$$

We believe that a model's feature extraction performance can be measured by considering the in-category distance and the between-categories distance of the training set feature vectors.

The in-category distance in a vector space means the average distance of all vectors to their respective category centroid. We use $icD$ to represent the in-category distance, and $icD$ can be calculated using the following formula:

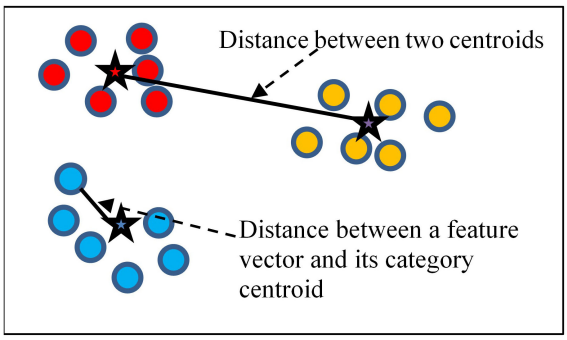



FIGURE 3
Example of feature vector distribution in a two-dimensional space.

10.3389/fpls.2023.1280671

$$icD = \sum_{p=1}^{k}\sum_{i=1}^{m} f_{ED}(cen_p, vector_i^p)/t\,. \quad (3)$$

In Formula (3), $k$ means the total number of categories, and the function $f_{ED}(vector_x, vector_y)$ means the Euclidean distance between two vectors.

The between-categories distance means the average distance between all category centroids. We use $bcD$ to represent the in-category distance, and $bcD$ can be calculated using the following formula:

$$bcD = \sum_{p=1}^{k}\sum_{q=p+1}^{k} f_{ED}(cen_p, cen_q)/C_k^2\,. \quad (4)$$

In Formula (4), $C_k^2$ is the combination number formula.

If a model could effectively extract the images' feature, then the feature vectors that belong to the same category should be closely distributed around their category centroid. Additionally, the category centroid of different categories should be distributed far apart from each other. As shown in Figure 4, there are two different models' feature vector spaces ($Model_A$ and $Model_B$). $Model_B$ obviously has evidently extracted a more effective feature because it is very easy to determine the category of an image in the vector space of $Model_B$. In contrast, the vectors of $Model_A$ are closely distributed together, making it very challenging to distinguish the category of an image. Therefore, we can conclude that $Model_B$ has a higher feature extraction performance than $Model_A$. From this perspective, we should assign a higher weight to $Model_B$ than to $Model_A$ during ensemble learning.

The distribution of vectors in the same category can be measured by $icD$, and the distribution of different categories can be measured by $bcD$. So, if a model has a bigger $bcD$ and a smaller $icD$, then we could consume it to have better feature extraction performance. We use $FEP_g$ to represent the feature extraction performance of $model_g$, and it can be calculated by the following formula:

$$FEP_g = {}^{bcD_g}\Big/_{icD_g}\,. \quad (5)$$

We should give the model that has a higher $FEP$ more weight when performing ensemble learning. So, the weight formula is defined as follows:

$$w_g = \frac{FEP_g}{\sum_{i=1}^{n} FEP_i}\,. \quad (6)$$

In this formula, $w_g$ means the weight assigned to $model_g$, and $n$ means the total number of models used during ensemble learning.

In ELCDR's stage 2, we input the training set into the trained models and then calculate their weights by measuring their feature extraction performance.

## 3.2 Ensemble learning strategy of ELCDR

Once a new image is input to the trained DL models, we can get a softmax function output from each model. The softmax function output is the probability distribution of this image belonging to each category. For example, if we input $img_x$ into a trained DL model, we get a softmax function output as [0.2, 0.6, 0.2]. It means that there are three categories in total, and $img_x$ has the most probability that belongs to the second category. If we input the $img_x$ to multiple trained DL models, we would get the corresponding multiple softmax function output. Then, we can use the weighting strategy in Section 3.2 to integrate them into a final softmax function output $sf_{final}$. $sf_{final}$. is calculated as follows:

$$sf_{final} = \sum_{i=1}^{n} w_i * sf_i\,. \quad (7)$$

In this formula, $sf_i$ means the softmax output of the $i_{\text{th}}$ model, $\omega_i$ means the weight of the $i_{\text{th}}$ model, and there are $n$ models in total to ensemble learning. At last, we choose the category that has the most probability in the $sf_{final}$ as the final crop disease recognition result.

## 3.3 The basic steps of ELCDR

The basic flowchart of ELCDR is shown in Figure 5, comprising seven basic steps. In this section, we will introduce the details of each step.

Step 1. In this step, we need to decide how many and which deep learning models for deployment in ensemble learning are necessary. If we aim to deploy $n$ models, then we need to build these models and set their hyperparameters. Generally, we can choose the models that are widely used in crop disease recognition to deploy, such as VGG, ResNet, and MobileNet. After completing this step, we can have $n$ models: $M_1$, $M_2$,…, and $M_n$.

Step 2. After the step of building the model, we can use the training set images to train the models one by one. Each image in

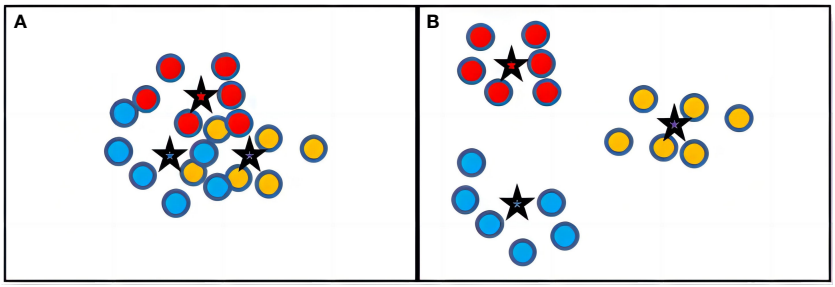



FIGURE 4
Example of feature vector distribution of different models. **(A)** Feature Vectors Distribution of $Model_A$. **(B)** Feature Vectors Distribution of $Model_B$.

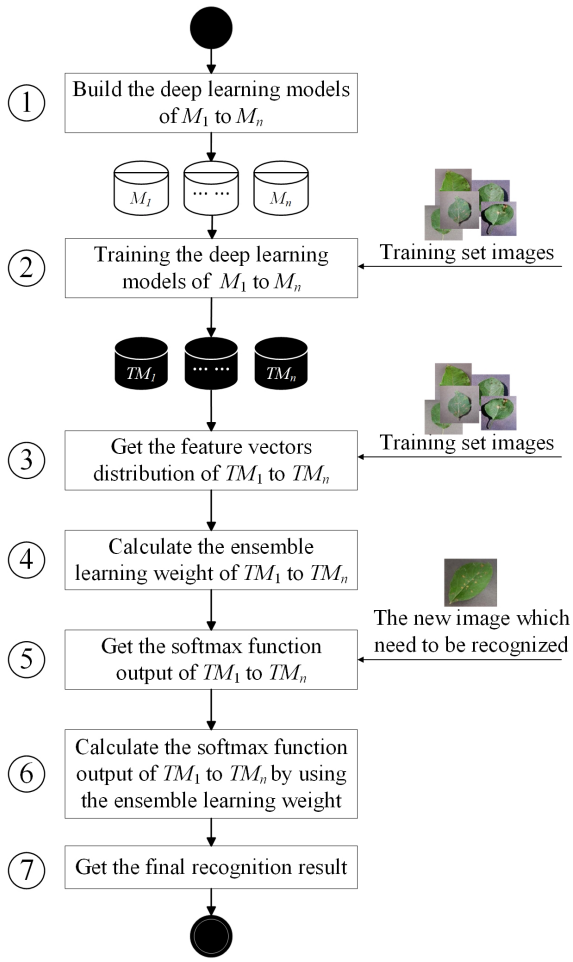



FIGURE 5
Basic flowchart of ELCDR.

the training set is labeled to indicate its crop disease category. If there are $h$ images in the training set, they could be divided into $k$ crop disease categories. This means that for each $image_i \in \{image_1, image_2, \ldots, image_h\}$, it will belong to a specific $category_j \in \{category_1, category_2, \ldots, category_k\}$. After the training process, we can get $n$ trained models: $TM_1$, $TM_2$,…, and $TM_n$.

Step 3. In this step, we need to input the training set images to the trained models one by one and extract the feature vectors in the pooling layer of the trained models. For each $image_i \in \{image_1, image_2, \ldots, image_h\}$, if we input it to $TM_g$, then we could extract its feature vector in the pooling layer of $TM_g$. Since there are $h$ images, we can get $h$ feature vectors in the feature vector space of $TM_g$. Then, we can calculate the feature extraction performance of $TM_g$ by Formulae (2–5) in Section 3.2. For each $TM_g \in \{TM_1, TM_2, \ldots, TM_n\}$, we can calculate its feature extraction performance $FEG_g$ by this way. So, we can get the feature extraction performance of all the models as $\{FEG_1, FEG_2, \ldots, FEG_n\}$.

Step 4. Since we have obtained the feature extraction performance of every model, we can calculate the ensemble learning weight of each model by using Formula (6). Then, we can get the ensemble learning weight of all the models as $\{w_1, w_2, \ldots, w_n\}$.

Step 5. Once we want to recognize a new crop disease image, we need to input it into the trained models one by one. Then, we can get the softmax function output of each model. After this step, we can get $n$ softmax function output. We use $sf_i$ to represent the softmax function output of $TM_i$ as follows:

$$sf_i = \{s_1, s_2, \ldots, s_k\}. \tag{8}$$

In Formula (8), si means that from the perspective of $TM_i$, the possibility of the new image belonging to $category_j$ is $s_i$.

Step 6. After getting the softmax function output of each trained model, we need to calculate the final softmax function output by Formula (7). Then, we can get the final softmax function output $sf_{final}$.

Step 7. After completing the previous step, we can have the final softmax function output as follows:

$$sf_{final} = \{sc_1, sc_2, \ldots, sc_k\}. \tag{9}$$

In Formula (9), $sc_j$ means the final possibility of the new image belonging to $category_j$. So, we will generally choose the category that has the maximum possibility value as the final recognition result.

# 4 Experiments

To verify the effectiveness of ELCDR, our experiments were performed on a dataset that includes four kinds of crops. We will mainly address three research questions (RQ) as follows:

1) RQ 1: Can ELCDR achieve better crop disease recognition performance than single model methods?
2) RQ 2: Can our weighting strategy achieve better crop disease recognition performance than the voting and average weighting strategies?
3) RQ 3: Is our feature extraction performance metric effective?

## 4.1 Dataset

As shown in Table 1, we have built a dataset which includes four kinds of crops, and the details of the dataset are shown in Table 1. In this dataset, each crop category has four image categories. The datasets for apple, corn, and grape were taken from the PlantVillage dataset (Hughes and Salathe, 2015), and the images were captured against a simple background. The rice leaf images were taken from the Sambalpur University's dataset (Sethy et al., 2020), and these images were captured in a natural environment with complex backgrounds. The images in Figure 6 are the example images of the dataset. The dataset has been uploaded to the Kaggle website and can be accessed by the following website address: https://www.kaggle.com/datasets/zhangguangchuan/crop-disease-dataset.

TABLE 1 Dataset.

| Crop | Number of images | Number of categories | Training: test |
|---|---|---|---|
| Apple | 2,000 | 4 | 3:2 |
| Corn | 2,000 | 4 | |
| Grape | 2,000 | 4 | |
| Rice | 2,000 | 4 | |

## 4.2 Performance metrics

We used accuracy, precision, recall, and confusion matrix to measure the performance of crop disease recognition methods in our experiments. These metrics are the most common metrics in the research field of image recognition. For their specific definition, please refer to our previous article (He et al., 2022).

## 4.3 Experiment details

We deployed three DL models in ELCDR, which are widely used in crop disease recognition methods, specifically VGG11, ResNet18, and MobileNet_v3. Their network structure is shown in Figure 7. The main idea of the VGG model is to construct a deep network model by reusing simple foundation blocks (Simonyan and Zisserman, 2014). VGG uses small convolutional kernels and pooling layers, with deeper layers and more channels. It hopes to extract more features by increasing the number of channels. VGG11 has 11 parameter layers, consisting of 8 convolutional layers and 3 fully connected layers. ResNet is a deep residual network developed by Microsoft Research Asia (He et al., 2016). It uses residual blocks and residual connections to construct the network, which allows for training deeper networks and avoids gradient vanishing problem. It can achieve better classification performance by continuously increasing the network depth. ResNet18 has 18 parameter layers, including 8 residual blocks. MobileNet_V3 was proposed by Google in 2019 and is the third-generation network of MobileNet (Howard et al., 2019). MobileNet_V3 is a lightweight model and can construct a very small, low latency, and low consumption model by only setting two hyperparameters. MobileNet_V3 mainly consists of 11 bottleneck layers. All of these models are widely applied in crop disease recognition systems and research studies. Thus, we chose them to test the effectiveness of our method in the experiment.

To answer RQ 1, we tested ELCDR using each of the crops from the dataset, and then we compared its performance against that of VGG11, ResNet18, and MobileNet_v3, respectively.

To answer RQ 2, we compared the performance of ELCDR with the voting and average weighting strategies.

To answer RQ 3, we calculated the ensemble learning weight and recognition performance for each model separately to investigate whether our weighting strategy effectively reflects the feature extraction performance of different models.

The hyperparameter settings of the models in the experiment are shown in Table 2. The number of epochs was set as 60 in the experiment, as we found that the models' loss function had basically converged after 50 training epochs in the experiment.

The models' training loss and accuracy in the training process on different datasets are shown in Figure 8. We can find that as the

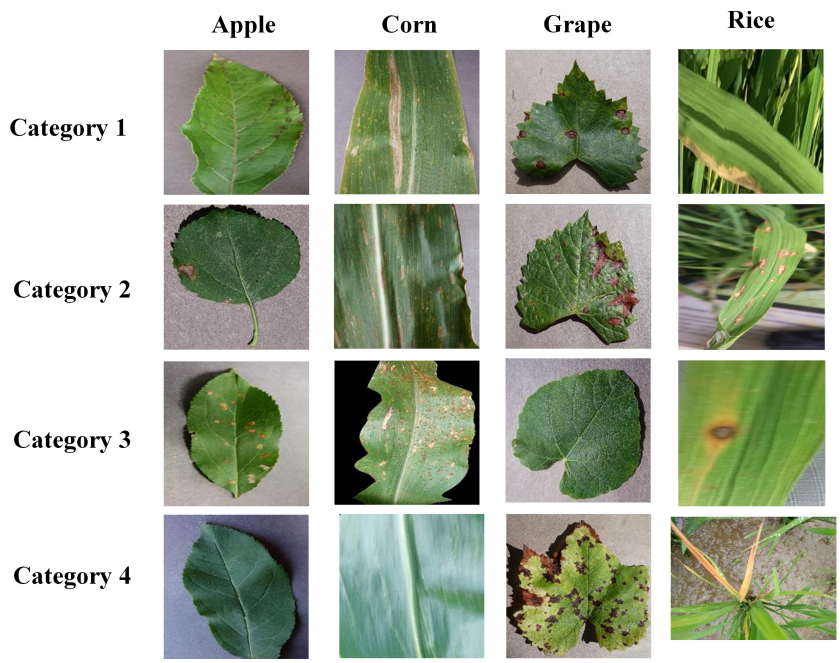



FIGURE 6
Samples of the dataset.

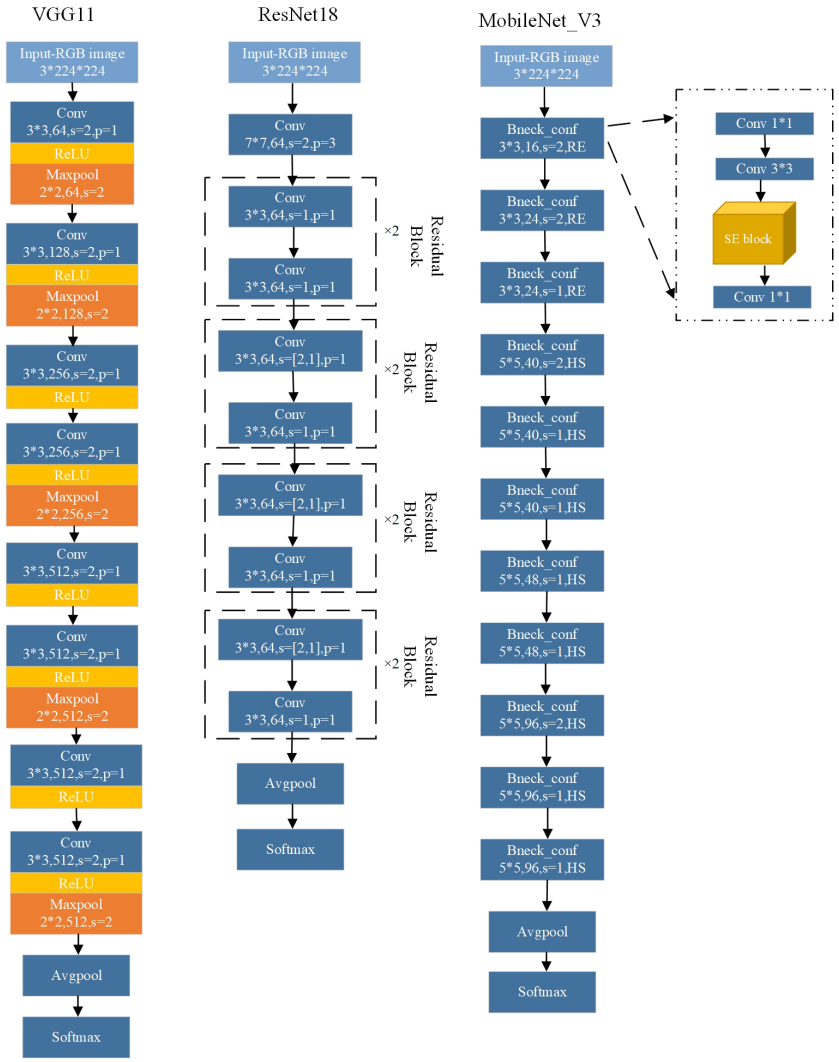



FIGURE 7
The basic network structure of deep learning models that are deployed in the experiment.

number of training epochs increases, the loss gradually decreases while the accuracy gradually improves. After over 50 training epochs, the models' loss and accuracy basically no longer show significant changes. During the experiments, we extracted the output from the pooling layer to serve as the feature vectors when calculating the feature extraction performance of a model.

TABLE 2 Hyperparameter settings.

| Hyperparameters | VGG11 | ResNet18 | MobileNet_v3 |
|---|---|---|---|
| Learning rate | 0.001 | 0.001 | 0.001 |
| Batch size | 32 | 32 | 32 |
| Number of epochs | 60 | 60 | 60 |
| Optimizer | Adam | Adam | Adam |
| Loss function | Cross-entropy | Cross-entropy | Cross-entropy |

## 4.4 Experiment results

### 4.4.1 Comparison of recognition performance between ELCDR and different single models

In order to answer RQ 1, we compared the recognition performance of ELCDR with VGG11, ResNet18, and MobileNet_v3. The results are shown in Table 3 and Figure 9.

From Table 3 and Figure 9, we can find that the accuracy, precision, recall, and F1 measure of ELCDR are the highest across all crop datasets. The single model that has the best recognition performance is ResNet18, while the single VGG11 model has the worst recognition performance. Compared with ResNet18, ELCDR improves by as much as 1.5 (apple), 0.88 (corn), 2.25 (grape), and 1.5 (rice) percentage points in accuracy in each case. As observed in Table 3, ELCDR also has improvements in precision, recall, and F1 measure over the single ResNet18 model.

In Figure 10, it can be found that ELCDR recognizes the greatest number of correct images on each category of the apple, corn, and

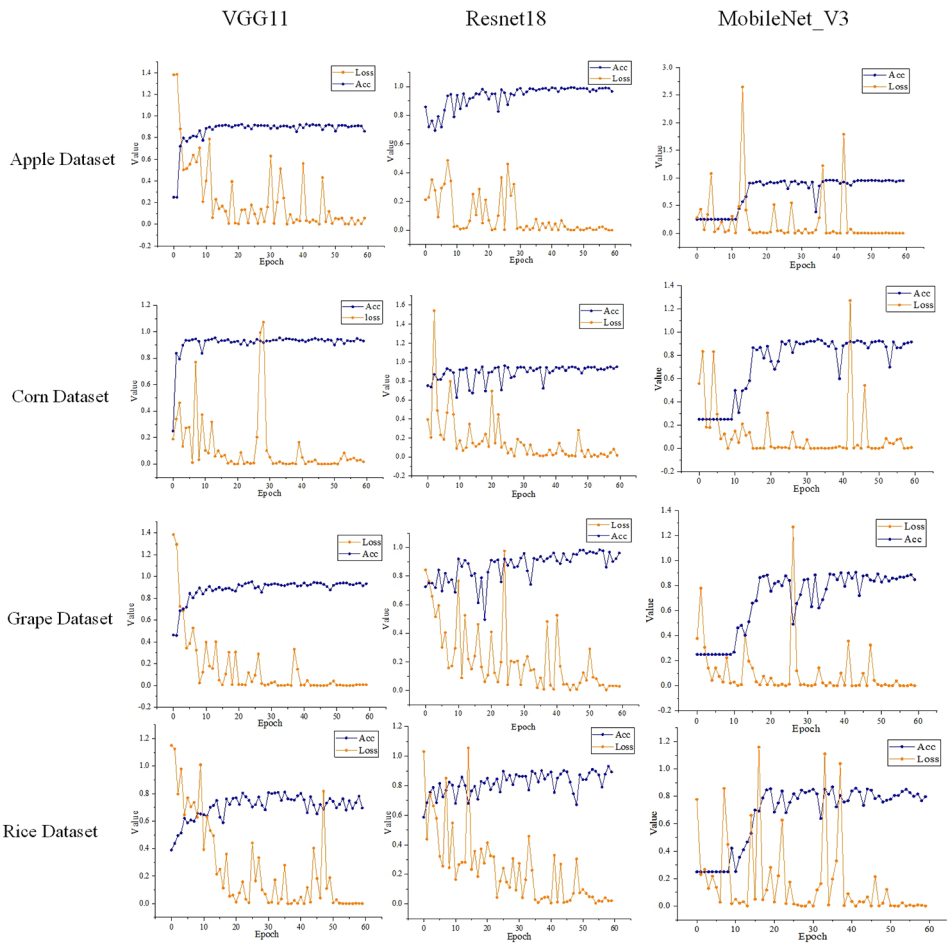



FIGURE 8
Loss and accuracy of different models on different datasets.

rice datasets. In the case of the grape dataset, while ELCDR recognizes a smaller number of correct images than VGG11 in the category of "black rot," it still recognizes the greatest number of correct images in total.

Based on these results, we can answer research question 1: compared with the single model methods, ELCDR can achieve better crop disease recognition performance. Table 3 and Figure 9 show that ELCDR attains higher accuracy, precision, recall, and F1 measure than the methods relying on a single deep learning model. Additionally, Figure 10 shows that ELCDR can recognize more correct images than the methods that are based on the single deep learning model. These findings have proven that the ensemble learning strategy of ELCDR is effective in achieving a better crop disease recognition performance than the single model methods.

### 4.4.2 Comparison of the recognition performance between ELCDR and other ensemble learning strategies

To answer research question 2, we also investigated the performance of the ensemble learning strategy of voting and average weighting. The results are shown in Table 4 and Figure 11. The average weighting strategy calculates the final softmax function output as follows:

$$sf_{final} = \frac{\sum_{i=1}^{n} sf_i}{n}. \quad (10)$$

In Formula (10), sfi means the softmax function output of model i, and n means the number of the models. We can see in Table 4 and Figure 11, ELCDR consistently achieves the best recognition performance on the experiment dataset. On the apple, corn, and grape datasets, the average weighting strategy achieves better performance than the voting strategy. On the rice dataset, the voting strategy has better performance than the average weighting strategy. However, ELCDR consistently achieves the best recognition performance regardless of the dataset. Compared with the voting strategy, ELCDR improves by as much as 1.75 (apple), 1.25 (corn), 0.75 (grape), and 7 (rice) percentage points in accuracy in each case. Compared with the average weighting strategy, ELCDR improves by as much as 1.13 (apple), 1 (corn), 0.5 (grape), and 5.37 (rice) percentage points in accuracy in each case. Especially on the rice dataset, the voting strategy and the average weighting strategy both achieve the worse performance than the single ResNet18 model, and the performance improvement of ELCDR is most evident in this case. This might be because the

TABLE 3 Comparison of the recognition performance between ELCDR and different single models.

| Crop | Model | Accuracy | Precision | Recall | F1 measure |
|---|---|---|---|---|---|
| Apple | VGG11 | 85.88 | 87.07 | 85.88 | 86.47 |
| | ResNet18 | 96.63 | 96.70 | 96.63 | 96.66 |
| | MobileNet_v3 | 94.88 | 94.96 | 94.88 | 94.92 |
| | **ELCDR** | **98.13** | **98.14** | **98.13** | **98.13** |
| Corn | VGG11 | 91.13 | 92.26 | 91.13 | 91.69 |
| | ResNet18 | 95.00 | 95.04 | 95.00 | 95.02 |
| | MobileNet_v3 | 91.50 | 91.81 | 91.50 | 91.65 |
| | **ELCDR** | **95.88** | **96.00** | **95.88** | **95.94** |
| Grape | VGG11 | 84.75 | 70.70 | 84.75 | 77.09 |
| | ResNet18 | 96.13 | 96.24 | 96.13 | 96.18 |
| | MobileNet_v3 | 93.38 | 93.34 | 93.38 | 93.36 |
| | **ELCDR** | **98.38** | **98.39** | **98.38** | **98.38** |
| Rice | VGG11 | 69.50 | 73.34 | 69.50 | 71.37 |
| | ResNet18 | 89.25 | 91.21 | 89.25 | 90.22 |
| | MobileNet_v3 | 79.75 | 84.12 | 79.75 | 81.88 |
| | **ELCDR** | **90.75** | **91.86** | **90.75** | **91.30** |

Bold values highlight ELCDR performance.

images in the rice dataset have a complex background, making it challenging for the models to extract the efficient feature, while the voting strategy and the average weighting strategy cannot determine which model has extracted the most efficient features. The recognition performance of the voting strategy and the average weighting becomes worse. In contrast, ELCDR can determine which model has extracted the most efficient features and assigns more weight during ensemble learning, consistently achieving better recognition performance. We will further discuss this hypothesis in research question 3.

In Figure 12, we can see that ELCDR consistently recognizes the greatest number of correct images on each dataset. When recognizing an image using the voting strategy, if each model votes for a different category, the voting strategy is considered invalid for that image. We can see that in Figure 12, there are consistently some images for which the voting strategy is invalid.

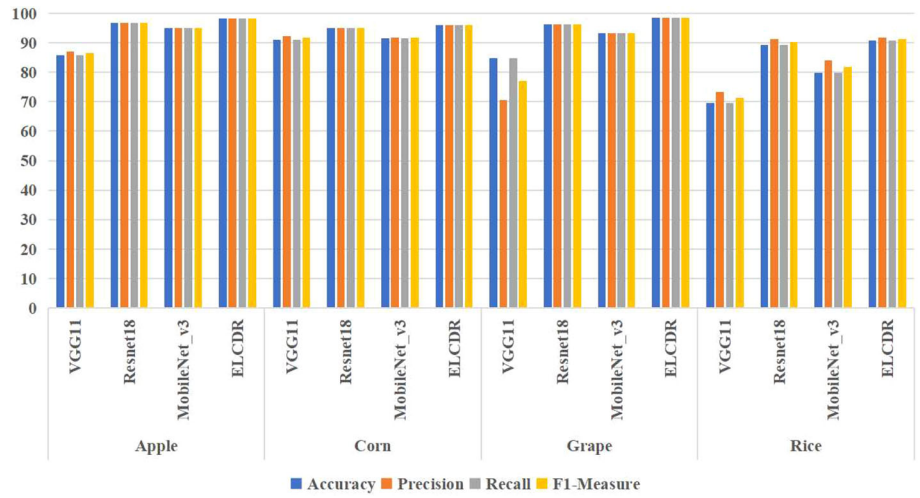



FIGURE 9
Comparison of the recognition performance between ELCDR and different single models.

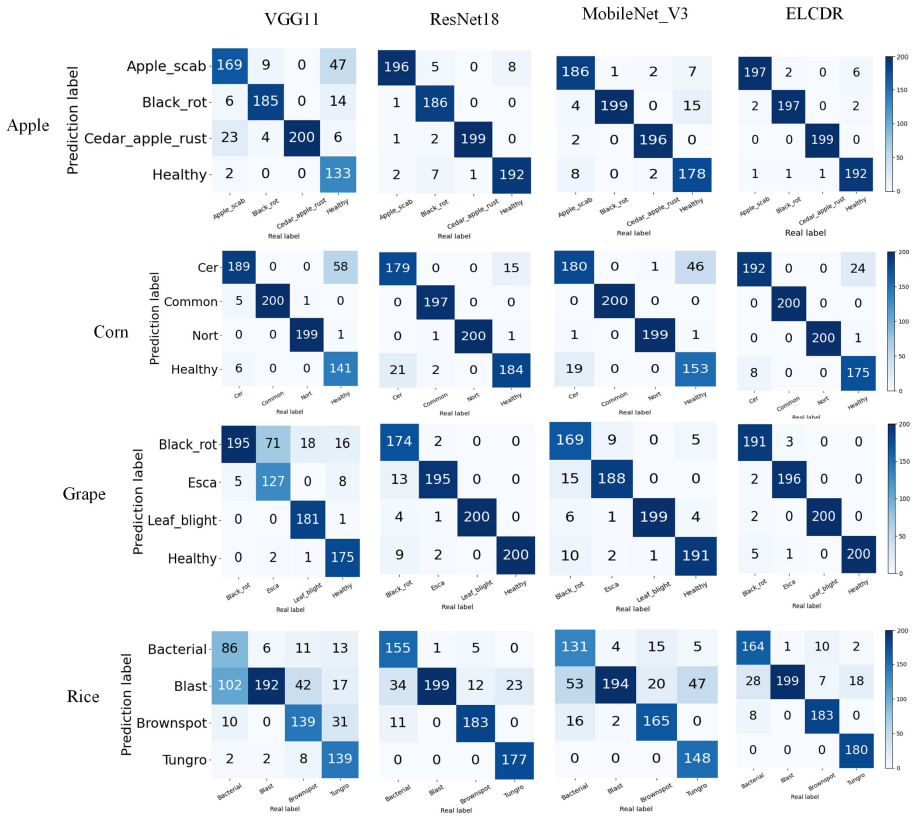



FIGURE 10
Different models' confusion matrix on different datasets.

This is also the main reason why the voting strategy achieves the worse performance than the other ensemble learning strategies.

By this, we can answer research question 2: the weighting strategy of ELCDR achieves better recognition performance than the voting strategy and average weighting strategy. Table 4 and Figure 11 show that ELCDR can achieve better accuracy, precision, recall, and F1 measure than the voting strategy and average weighting strategy. Figure 12 shows that ELCDR can recognize more correct images than the voting strategy and average weighting strategy. These results have proven that the ensemble learning strategy of ELCDR is more effective than the voting strategy and average weighting strategy.

TABLE 4 Comparison of the recognition performance between ELCDR and other ensemble learning strategies.

| Crop | Method | Accuracy | Precision | Recall | F1 measure |
|---|---|---|---|---|---|
| Apple | Voting | 96.38 | 97.03 | 96.38 | 96.70 |
| | Average weighting | 97.00 | 97.05 | 97.00 | 97.02 |
| | **ELCDR** | **98.13** | **98.14** | **98.13** | **98.13** |
| Corn | Voting | 94.63 | 94.86 | 94.63 | 94.77 |
| | Average weighting | 94.88 | 95.16 | 94.88 | 95.02 |
| | **ELCDR** | **95.88** | **96.00** | **95.88** | **95.94** |
| Grape | Voting | 97.63 | 98.25 | 97.63 | 97.94 |
| | Average weighting | 97.88 | 98.24 | 97.88 | 98.13 |
| | **ELCDR** | **98.38** | **98.39** | **98.38** | **98.38** |
| Rice | Voting | 83.75 | 89.79 | 83.75 | 86.66 |
| | Average weighting | 85.38 | 88.49 | 85.38 | 86.91 |
| | **ELCDR** | **90.75** | **91.86** | **90.75** | **91.30** |

Bold values highlight ELCDR performance.

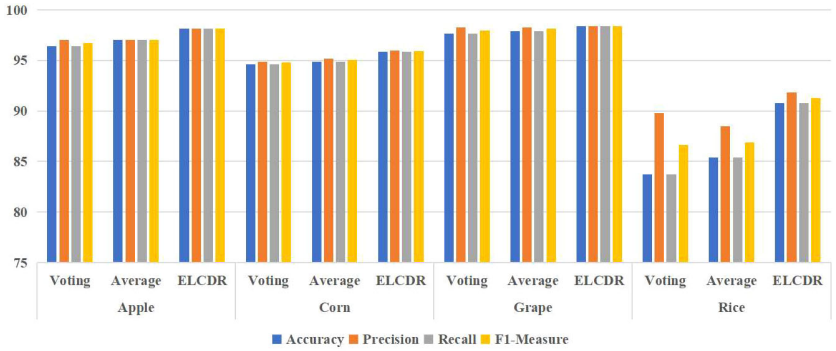



FIGURE 11
Comparison of the recognition performance between ELCDR and other ensemble learning strategies.

### 4.4.3 Comparison of the ensemble learning weights of different models

In order to answer research question 3, we calculated the in-category distance (icD), between-categories distance (bcD), feature extraction performance (FEP), weight, and accuracy performance for each model integrated into ELCDR. The results are shown in Table 5 and Figure 13. We can find that the ResNet18 model consistently achieved the best accuracy performance on each of the datasets, resulting in the highest FEP and ensemble learning weight. Conversely, the VGG11 model consistently demonstrated the worst accuracy performance on each of the datasets, resulting in the lowest FEP and ensemble learning weight. So, the FEP and weight distribution of the models in ELCDR are consistent with their recognition performance. It means that the model that has better

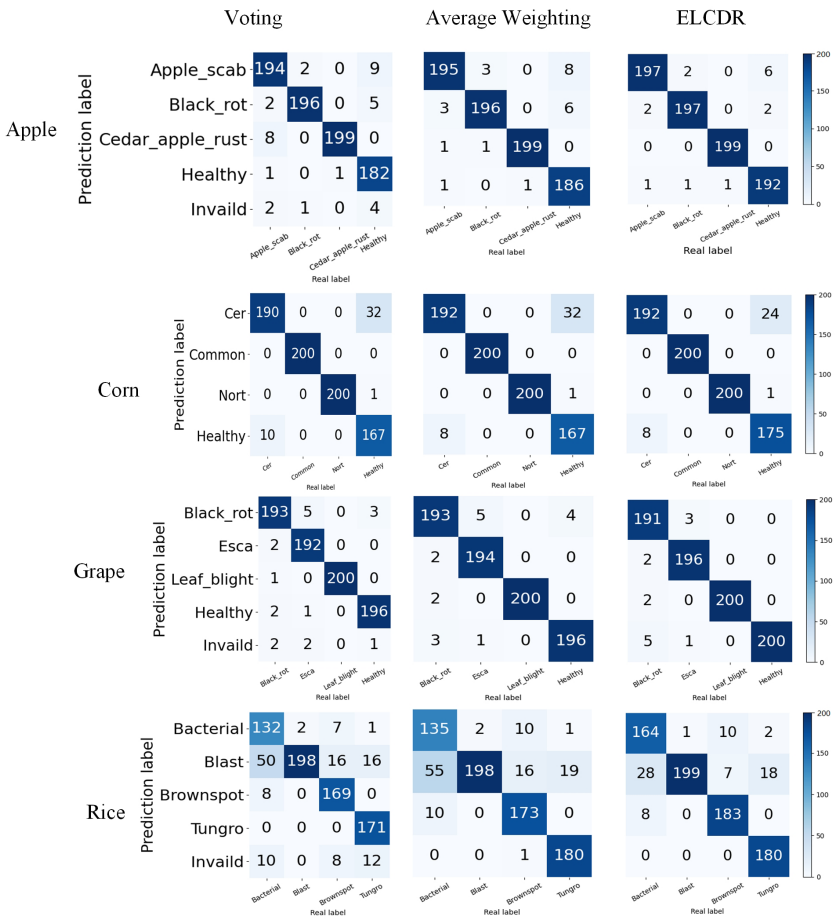



FIGURE 12
Different ensemble learning strategies' confusion matrix on different datasets.

TABLE 5 Comparison of the ensemble learning weights of different models.

| Crop | Model | icD | bcD | FEP (bcD/icD) | Weight (%) | Accuracy (%) |
|---|---|---|---|---|---|---|
| Apple | VGG11 | 2.051201 | 1.807103 | 0.880997 | 14.371 | 85.88 |
| | ResNet18 | 8.251477 | 23.11970 | 2.801886 | **45.706** | **96.63** |
| | MobileNet_v3 | 6.055690 | 14.82056 | 2.447378 | 39.923 | 94.88 |
| Corn | VGG11 | 1.335760 | 2.818918 | 2.110348 | 20.023 | 91.13 |
| | ResNet18 | 4.639691 | 20.89895 | 4.504383 | **42.738** | **95.00** |
| | MobileNet_v3 | 4.316191 | 16.94026 | 3.924816 | 37.239 | 91.50 |
| Grape | VGG11 | 1.756650 | 1.635504 | 0.931035 | 13.523 | 84.75 |
| | ResNet18 | 6.790820 | 20.96166 | 3.086765 | **44.833** | **96.13** |
| | MobileNet_v3 | 3.948179 | 11.32035 | 2.867232 | 41.644 | 93.38 |
| Rice | VGG11 | 2.27075 | 0.979885 | 0.431525 | 10.061 | 69.50 |
| | ResNet18 | 11.52429 | 22.49176 | 1.951684 | **45.504** | **89.25** |
| | MobileNet_v3 | 8.528134 | 16.25342 | 1.905859 | 44.435 | 79.75 |

Bold values highlight ELCDR performance.

feature extraction and recognition performance receives greater weight during ensemble learning with ELCDR, while those with lower performance receive lower weight.

By this, we can answer research question 3 that the feature extraction performance metric of ELCDR is effective. We can use it to measure the model's feature extraction performance and calculate ensemble learning weight efficiently.

We also compared the feature maps of different models in the experiment. The original image was input into VGG11, ResNet18, and MobileNet_V3, respectively. Then, we extracted the feature maps from their convolutional layer. The feature maps are shown in Figure 14. We can find that the feature map of ResNet18 has the most texture detail features and lesion features, while the feature map of VGG11 is the blurriest. This means that ResNet18 has extracted the most effective features, and VGG11 has extracted the least. MobileNet_V3 is a lightweight model, so it cannot extract as many effective features as ResNet18. However, we also can find that the feature map of MobileNet_V3 has more texture detail features and lesion features than the VGG11. Therefore, we can suggest that different models have different feature extraction performance, which is the main reason why different weights are assigned to these models in ensemble learning.

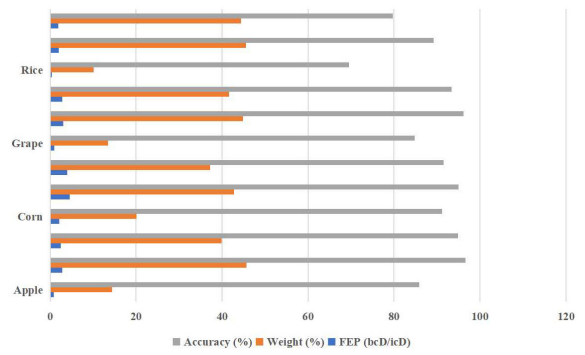



FIGURE 13
Comparison of the ensemble learning weights of different models.

## 4.5 Discussion

In this section, we conducted experiments to assess the crop leaf recognition performance of ELCDR. The experimental results show that ELCDR can achieve better recognition performance than the methods that are based on a single model. It can also achieve better recognition performance than the traditional ensemble learning methods that rely on the voting or average weighting strategies. This is because we applied a novel weight calculation method, which can measure different models' feature extraction performance through the distribution of feature vectors. With this weight calculation method, we assign more weight to the models with better recognition performance when conducting ensemble learning for crop leaf disease recognition. Otherwise, the models with poorer performance receive less weight. All of the experimental results verify the effectiveness of the ELCDR as proposed in this paper.

# 5 Conclusions

Compared with the crop leaf disease recognition methods based on single models, ensemble learning methods have the advantage of integrating multiple single learning models to get more accurate, stable, and robust results. This advantage stems from the fact that different models can extract image features from various perspectives. Ensemble learning can combine these features effectively to get a more powerful integrated model.

Traditional ensemble learning methods generally use the voting or average weight strategies, treating all integrated models as equally important. However, different models may have different feature extraction ability. When using ensemble learning, it is essential to assign more weight to the models that have better feature extraction

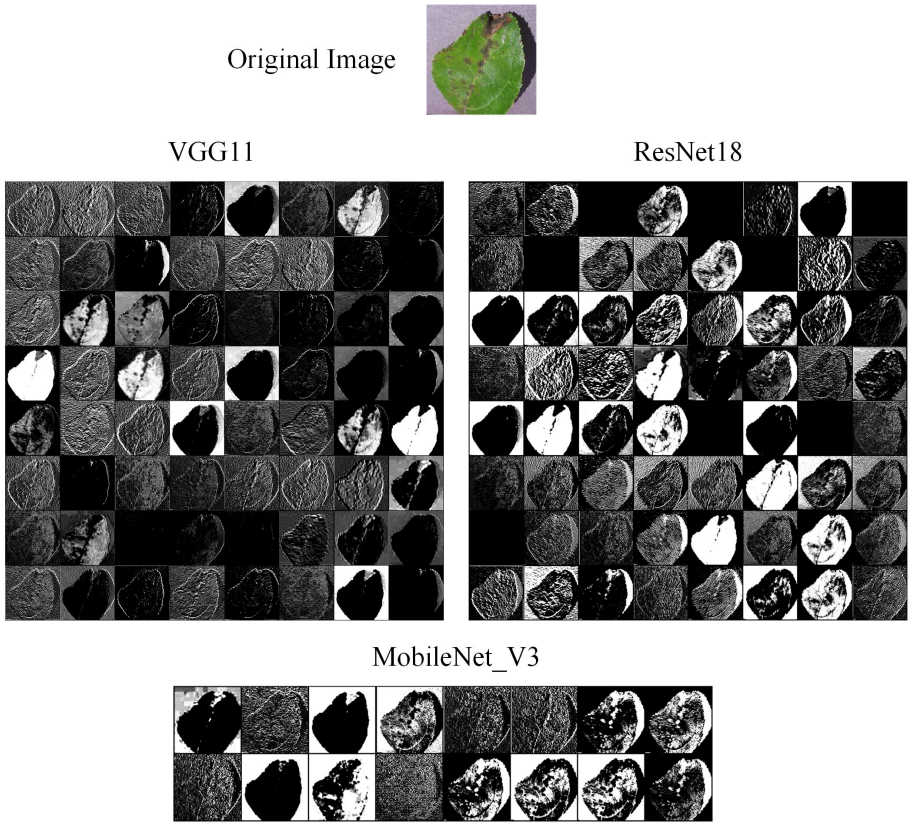



FIGURE 14
The feature map comparison of the different models.

ability and less weight to those that have a weaker feature extraction ability. To solve this problem, we have introduced a novel ensemble learning method for crop leaf disease recognition, named ELCDR. This approach measures each model's feature extraction ability by calculating the model's feature vector distribution and calculates the ensemble learning weight for each model based on the model's feature extraction ability. Through this approach, ELCDR can integrate more effective features from different models, to obtain more accurate, stable, and robust crop leaf disease recognition results. In order to verify the recognition performance of ELCDR, we compared its performance with the recognition methods which are based on a single model, voting strategy, and average weighting strategy in the experiments. The experimental results clearly demonstrate that ELCDR can achieve better accuracy, recall, precision, and F1 measure performance than the recognition methods which are based on a single model, voting strategy, and average weighting strategy. Compared with the VGG11 model, ELCDR improves by as much as 12.25 (apple), 4.75 (corn), 13.63 (grape), and 21.25 (rice) percentage points in accuracy in each case. Compared with the ResNet18 model, ELCDR improves by as much as 1.5 (apple), 0.88 (corn), 2.25 (grape), and 1.5 (rice) percentage points in accuracy in each case. Compared with the MobileNet_V3 model, ELCDR improves by as much as 3.25 (apple), 4.38 (corn), 5 (grape), and 11 (rice) percentage points in accuracy in each case. Compared with the voting strategy, ELCDR improves by as much as 1.75 (apple), 1.25 (corn), 0.75 (grape), and 7 (rice) percentage points in accuracy in each case. Compared with the average weighting strategy, ELCDR improves by as much as 1.13 (apple), 1 (corn), 0.5 (grape), and 5.37 (rice) percentage points in accuracy in each case. These experimental results validate that ELCDR consistently has better recognition performance than the methods that are based on a single model or traditional voting strategy.

We have successfully verified the effectiveness of our proposed feature extraction ability metric in the experiments. However, our new method currently only completes the recognition task in a small range of scenarios. We still face some challenges as follows:

1) The effectiveness of ELCDR on more complex datasets, which may involve a greater variety of crops and harsh environmental conditions, still needs further verification.
2) It remains to be determined how the number of integrated models in ELCDR impacts the recognition performance.
3) Identifying the optimal combination of models to achieve the best recognition performance for ELCDR is another area of potential research.

In the future, we aim to compare the potential benefits and limitations of the existing crop leaf disease recognition methods and explore a robust and accurate crop leaf disease recognition segmentation method.

## Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

## Author contributions

YH: Conceptualization, Writing – original draft, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software. GZ: Validation, Visualization, Writing – review & editing. QG: Conceptualization, Data curation, Supervision, Writing – review & editing.

## Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This study was supported by the National Natural Science Foundation of China (No. 32101611), the Youth Project of Basic Research Program of Yunnan Province (No. 202101AU070096), the Major Project of Science and Technology of Yunnan Province (Nos. 202202AE090021 and 202302AE090020), and the Open Research Program of State Key Laboratory for Conservation and Utilization of Bio-Resource in Yunnan (No. GZKF2021009).

## Acknowledgments

Appreciations are given to the editor and reviewers of the journal.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

Ahmed, M. H., Islam, T., and Ema, R. R. (2019). "A new hybrid intelligent GAACO algorithm for automatic image segmentation and plant leaf or fruit diseases identification using TSVM classifier," in *Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Cox's Bazar, Bangladesh. (ECCE), 1–6. doi: 10.1109/ECACE.2019.8679219

Applalanaidu, M. V., and Kumaravelan, G. (2021). "A review of machine learning approaches in plant leaf disease detection and classification," in *Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, Tirunelveli, India. (ECCE), 716–724. doi: 10.1109/ICICV50876.2021.9388488

Bhagat, M., Kumar, D., and Kumar, S. (2023). Bell pepper leaf disease classification with LBP and VGG-16 based fused features and RF classifier. *Int. J. Inf. Technol.* 15 (1), 465–475. doi: 10.1007/s41870-022-01136-z

Chaudhary, A., Thakur, R., Kolhe, S., and Kamal, R. (2020). A particle swarm optimization based ensemble for vegetable crop disease recognition. *Comput. Electron. Agricult.* 178, 105747. doi: 10.1016/j.compag.2020.105747

Chen, H.-C., Widodo, A. M., Wisnujati, A., Rahaman, M., Lin, J. C.-W., Chen, L., et al. (2022). AlexNet convolutional neural network for disease detection and classification of tomato leaf. *Electronics* 11, 951. doi: 10.3390/electronics11060951

Chen, J., Zhang, D., Suzauddola, M., and Zeb, A. (2021). Identifying crop diseases using attention embedded MobileNet-V2 model. *Appl. Soft Computing* 113, 107901. doi: 10.1016/j.asoc.2021.107901

Coulibaly, S., Kamsu-Foguem, B., Kamissoko, D., and Traore, D. (2019). Deep neural networks with transfer learning in millet crop images. *Comput. Industry* 108, 115–120. doi: 10.1016/j.compind.2019.02.003

Cruz, A. C., Luvisi, A., Bellis, L. D., and Ampatzidis, Y. (2017). X-FIDO: an effective application for detecting olive quick decline syndrome with deep learning and data fusion. *Front. Plant Sci.* 8. doi: 10.3389/fpls.2017.01741

Fuentes, A. F., Yoon, S., Lee, J., and Park, D. S. (2018). High-Performance deep neural network-Based tomato plant diseases and pests diagnosis system with refinement filter bank. *Front. Plant Sci.* 9. doi: 10.3389/fpls.2018.01162

Ganaie, M., Hu, M., Malik, A., Tanveer, M., and Suganthan, P. (2022). Ensemble deep learning: A review. *Eng. Appl. Artif. Intelligence.* 115, 105151. doi: 10.1016/j.engappai.2022.105151

Ganesan, G., and Chinnappan, J. (2022). Hybridization of ResNet with YOLO classifier for automated paddy leaf disease recognition: An optimized model. *J. Field Robotics* 39 (7), 1085–1109. doi: 10.1002/rob.22089

He, Y., Gao, Q., and Ma, Z. (2022). Crop leaf disease image recognition method based on bilinear residual networks. *Math. Problems Eng.* 2022, 2948506. doi: 10.1155/2022/2948506

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA. (IEEE), 770–778. doi: 10.1109/CVPR.2016.90

Hossain, E., Hossain, M. F., and Rahaman, M. A. (2019). "A color and texture based approach for the detection and classification of plant leaf disease using KNN classifier," in *Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Cox's Bazar, Bangladesh. (IEEE), 1–6. doi: 10.1109/ECACE.2019.8679247

Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L. C., Tan, M., et al. (2019). "Searching for mobilenetv3," in *International Conference on Computer Vision 2019 (ICCV 2019)*, Seoul, South Korea. (ICCV). doi: 10.48550/arXiv.1905.02244

Hughes, D. P., and Salathe, M. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. *Computer Sci.* 2015, 1–13. doi: 10.48550/arXiv.1511.08060

Islam, M., Dinh, A., Wahid, K., and Bhowmik, P. (2017). "Detection of potato diseases using image segmentation and multiclass support vector machine," in *Proceedings of the 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Windsor, ON, Canada. (CCECE), 1–4. doi: 10.1109/CCECE.2017.7946594

Jiang, P., Chen, Y., Liu, B., He, D., and Liang, C. (2019). Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. *IEEE Access* 7, 59069–59080. doi: 10.1109/ACCESS.2019.2914929

Jiang, F., Lu, Y., Chen, Y., Cai, D., and Li, G. (2020). Image recognition of four rice leaf diseases based on deep learning and support vector machine. *Comput. Electron. Agricult.* 179 (2), 105824. doi: 10.1016/j.compag.2020.105824

Kaur, S., Pandey, S., and Goel, S. (2018). Semi-automatic leaf disease detection and classification system for soybean culture. *IET Image Processing* 12 (6), 1038–1048. doi: 10.1049/iet-ipr.2017.0822

Kawasaki, Y., Uga, H., Kagiwada, S., and Lyatomi, H. (2015). "Basic study of automated diagnosis of viral plant diseases using convolutional neural networks," in *Proceedings of the International Symposium on Visual Computing*, ISVC 2015. Lecture Notes in Computer Science, Vol. 9475 (Springer, Cham). doi: 10.1007/978-3-319-27863-6_59

Kundu, N., Rani, G., Dhaka, V. S., Gupta, K., Nayak, S. C., Verma, S., et al. (2021). IoT and interpretable machine learning based framework for disease prediction in pearl millet. *Sensors* 21, 5386. doi: 10.3390/s21165386

Li, H., Jin, Y., Zhong, J., and Zhao, R. (2021). A fruit tree disease diagnosis model based on stacking ensemble learning. *Complexity* 2021, 6868592. doi: 10.1155/2021/6868592

Mathew, A., Antony, A., Mahadeshwar, Y., Khan, T., and Kulkarni, A. (2022). Plant disease detection using GLCM feature extractor and voting classification approach. *Materials Today: Proc.* 58, Part 1, 407–415. doi: 10.1016/j.matpr.2022.02.350

Nachtigall, L. G., Araujo, R. M., and Nachtigall, G. R. (2016). "Classification of apple tree disorders using convolutional neural networks," in *Proceedings of the 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, San Jose, CA, USA. (ICTAI), 472–476. doi: 10.1109/ICTAI.2016.0078

Ngugi, L. C., Abelwahab, M., and Abo-Zahhad, M. (2021). Recent advances in image processing techniques for automated leaf pest and disease recognition – A review. *Inf. Process. Agricult.* 8 (1), 27–51. doi: 10.1016/j.inpa.2020.04.004

Palanisamy, S., and Sanjana, N. (2023). "Corn leaf disease detection using genetic algorithm and weighted voting," in *Proceedings of the 2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, Coimbatore, India. (ICAECA), doi: 10.1109/ICAECA56562.2023.10200196

Pantazi, X. E., Moshou, D., and Tamouridou, A. A. (2019). Automated leaf disease detection in different crop species through image features analysis and One Class Classifiers. *Comput. Electron. Agricult.* 156, 96–104. doi: 10.1016/j.compag.2018.11.005

Paymode, A. S., and Malode, V. B. (2021). Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG. *Artif. Intell. Agricult.* 6, 23–33. doi: 10.1016/j.aiia.2021.12.002

Prakash, R. M., Saraswathy, G. P., Ramalakshmi, G., Mangaleswari, K. H., and Kaviya, T. (2017). "Detection of leaf diseases and classification using digital image processing," in *Proceedings of the 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India. (ICIIECS), 1–4. doi: 10.1109/ICIIECS.2017.8275915

Reddy, S. R. G., Varma, G. P. S., and Davuluri, R. L. (2023). Resnet-based modified red deer optimization with DLCNN classifier for plant disease identification and classification. *Comput. Electrical Engineering* 105, 108492. doi: 10.1016/j.compeleceng.2022.108492

Sethy, P. K., Barpanda, N. K., Rath, A. K., MTech, S. K. B. (2020). Deep feature based rice leaf disease identification using support vector machine. *Comput. Electron. Agricult.* 175, 105527. doi: 10.1016/j.compag.2020.105527

Shan-E-Ahmed, R., Gillian, P., Clarkson J, P., and Rajpoot, N. M. (2015). Automatic detection of diseased tomato plants using thermal and stereo visible light images. *PloS One* 10 (4), e0123262. doi: 10.1371/journal.pone.0123262

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *ImageNet Large Scale Visual Recognition Challenge* 2014, 1–14. doi: 10.48550/arXiv.1409.1556

Singh, U. P., Chouhan, S. S., Jain, S., and Jain, S. (2019). Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease. *IEEE Access* 7, 43721–43729. doi: 10.1109/ACCESS.2019.2907383

Stephen, A., Punitha, A., and Chandrasekar, A. (2023). Designing self attention-based ResNet architecture for rice leaf disease classification. *Neural Comput. Applic.* 35, 6737–6751. doi: 10.1007/s00521-022-07793-2

Sun, G., Jia, X., and Geng, T. (2018). Plant diseases recognition based on image processing technology. *J. Electrical Comput. Eng.* 2018, 1–7. doi: 10.1155/2018/6070129

Xie, X., Ma, Y., Liu, B., He, J., Li, S., and Wang, H. (2020). A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks. *Front. Plant Sci.* 11. doi: 10.3389/fpls.2020.00751

Yang, L., Yu, X., Zhang, S., Long, H., Zhang, H., Xu, S., et al. (2023). GoogLeNet based on residual network and attention mechanism identification of rice leaf diseases. *Comput. Electron. Agricult.* 204, 107543. doi: 10.1016/j.compag.2022.107543

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 15 January 2024
DOI 10.3389/fpls.2023.1337467

Check for updates

OPEN ACCESS

EDITED BY
Jucheng Yang,
Tianjin University of Science and Technology, China

REVIEWED BY
Zhaoyu Zhai,
Nanjing Agricultural University, China
Soumi Mitra,
Baylor College of Medicine, United States

*CORRESPONDENCE
Wei Zhan
zhanwei814@yangtzeu.edu.cn

RECEIVED 13 November 2023
ACCEPTED 18 December 2023
PUBLISHED 15 January 2024

CITATION
Zhou T, Zhan W and Xiong M (2024) A series of methods incorporating deep learning and computer vision techniques in the study of fruit fly (Diptera: Tephritidae) regurgitation. *Front. Plant Sci.* 14:1337467. doi: 10.3389/fpls.2023.1337467

COPYRIGHT
© 2024 Zhou, Zhan and Xiong. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# A series of methods incorporating deep learning and computer vision techniques in the study of fruit fly (Diptera: Tephritidae) regurgitation

Tongzhou Zhou[1], Wei Zhan[2*] and Mengyuan Xiong[2]

[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, Hong Kong SAR, China, [2]School of Computer Science, Yangtze University, Jingzhou, China

In this study, we explored the potential of fruit fly regurgitation as a window to understand complex behaviors, such as predation and defense mechanisms, with implications for species-specific control measures that can enhance fruit quality and yield. We leverage deep learning and computer vision technologies to propose three distinct methodologies that advance the recognition, extraction, and trajectory tracking of fruit fly regurgitation. These methods show promise for broader applications in insect behavioral studies. Our evaluations indicate that the I3D model achieved a Top-1 Accuracy of 96.3% in regurgitation recognition, which is a notable improvement over the C3D and X3D models. The segmentation of the regurgitated substance via a combined U-Net and CBAM framework attains an MIOU of 90.96%, outperforming standard network models. Furthermore, we utilized threshold segmentation and OpenCV for precise quantification of the regurgitation liquid, while the integration of the Yolov5 and DeepSort algorithms provided 99.8% accuracy in fruit fly detection and tracking. The success of these methods suggests their efficacy in fruit fly regurgitation research and their potential as a comprehensive tool for interdisciplinary insect behavior analysis, leading to more efficient and non-destructive insect control strategies in agricultural settings.

KEYWORDS

**fruit fly, behavior recognition, semantic segmentation, object tracking, regurgitation**

## 1 Introduction

In recent years, the harm caused by fruit flies has been aggravated year by year in orchards worldwide and has a great impact on the quality and yield of fruits, even affecting economic growth. Therefore, fruit fly control and the study of their behavior are important.

Since 2017, China's fruit planting area has gradually increased. The industry has become an important part of China's agriculture and plays an important role in improving

agricultural development and reaching residents' income. China's fruit production will reach 300 million tons in 2021, which is 4.5% year-on-year. Among them, apple production is 4,579.34 tons, up 4.3% year-on-year; citrus production is 5,595.61 tons, up 9.2% year-on-year; pear production is 1,887.59 tons, up 6% year-on-year; banana production is 1,172.42 tons, up 1.8% year-on-year. These figures reflect the large production volume of the fruits. However, when growing them, fruits are particularly vulnerable to fruit fly, as Vayssieres et al. (2009) indicated in a 2010 study: fruit fly undermines the quality of mango fruit in Benin, which leads to significant production losses. The maximum loss can exceed 70% and has a significant impact on the economy. According to Badii et al. (2015), fruits and vegetables such as citrus, pineapple, papaya, bananas, and tomatoes are consumed in large amounts every year and create great economic value, but more than 50% of the production is infested by insects. Therefore, many countries and regions are investing heavily in controlling fruit flies, but the results are unsatisfactory, expensive, and inefficient. For example, the use of insecticides for control will be phased out because of the increasingly strict restrictions on the use of insecticides and increasing demand for healthy food worldwide (Dias et al., 2018). Therefore, a green and efficient solution to control fruit flies is urgently needed. Current research on fruit flies is moving in this direction. Ant et al. indicated in a 2012 study, using insect sterility techniques to control fruit fly (Ant et al., 2012). Navarro-Llopis et al. (2014) and Lasa et al. (2013) proposed to apply mass trapping techniques, and trapping devices to trap fruit fly in large scale. The above program is effective in controlling insects and confirms that it is important to study how to control fruit flies from the aspect of physiological habits.

Regurgitation is one of the typical physiological behaviors of fruit fly. Many experts are interesting in it because it contains a variety of behavioral information. Many insects have the behavior as well. For example, herbivorous insects regurgitate at the injury part of plants, and their regurgitation liquid contains inducers that trigger different plant responses. Plants will use the inducers to distinguish mechanical damage and herbivorous insects' damage so that they can adopt different responses of defense. However, the insects will confuse the plants by creating the wrong kind of inducers so as to suppress plants' defenses (Timilsena and Mikó, 2017). Regurgitation can also be toxic to vertebrate predators and impose an impact on them (Sword, 2001). Dipteran pests will regurgitate and die without injuring other crops and insects when fed different concentrations of polyols, which provides support for achieving specific insecticide (Díaz-Fleischer et al., 2019). After feeding food the fruit fly, we found that regurgitation can also play an important role in capturing bacteria in the environment and potentially help adult fruit fly to eliminate ingested toxic substances (Guillén et al., 2019). Regurgitation by fly-like insects was also found in a study of Cáceres et al. (2019). According to Wasala et al. (2011), house fly regurgitation spots may be a source of *E. coli* O157:H7 contamination of leafy greens. Therefore, regurgitation liquid can be extracted to detect *E. coli* O157:H7 and other bacteria on plants so that controlling measures can then be taken. In summary, it is important to both study the c behaviors of fruit fly and extract the regurgitated liquids, which will provide practical tools for agricultural experts to do insect's research.

Current research on regurgitation behavior in fruit flies primarily employs two main methodologies. One involves the detection of regurgitation components using chemical reagents, while the other entails anatomical studies to explore the relationship between regurgitation in fruit flies and their internal structures. It is noteworthy that, as of now, systematic research on the regurgitation behavior of fruit flies using artificial intelligence methods has not been undertaken. With the rapid advancement of deep learning technology and computer vision, these two techniques have found extensive applications in agriculture and insect research, including plant detection, disease identification, and pest control. The introduction of deep learning models facilitates expeditious and efficient conduct of insect and plant research, allowing for tasks such as insect quantification, and the behavioral recognition of various anatomical segments of the citrus fruit fly can be carried out in a straightforward manner. She et al. realized continuous dynamic monitoring of orchards through artificial intelligence technology, helping researchers and fruit farmers grasp disease and insect data in time, reduce the use of artificial and pesticide, and realize scientific early warning and control of diseases and pests (She et al., 2022). Hong et al. (2022) detected the grooming behavior of multi-target insects, providing a research basis for the study of insect-related behaviors.

In this study, we combined deep learning and computer vision techniques and selected three different models for three types of problems in fruit fly regurgitation: behavior recognition, regurgitation liquid extraction, and trajectory tracking. We conducted a comprehensive assessment and comparison of these models to address specific issues, aiming to reveal the underlying mechanisms of regurgitation behavior through an analysis of fruit fly motion trajectories and regurgitation patterns. Our goal was to provide observable and quantifiable data for relevant biological and neuroscientific research. This effort not only contributes to a deeper understanding of fruit fly biological behavior but also establishes a scientific foundation for researchers to delve into the physiological characteristics of fruit flies. Furthermore, we explored the feasibility of applying these methods to other insect regurgitation studies, thereby expanding the potential avenues of research on insect behavior.

# 2 Materials and methods

## 2.1 Overview

The method proposed in this paper to study the regurgitation behavior of fruit flies is divided into three main parts, as follows:

1. Detect and recognize the regurgitation behavior of fruit flies using a behavior recognition network.
2. Use Unet network combined with CBAM attention mechanism and other networks to segment the regurgitated liquid, regurgitated liquid can be extracted

precisely, and the area of each liquid can be calculated by OpenCV, so that the total amount of regurgitated liquid can be estimated.

3. To conduct a more comprehensive study of insect regurgitation, we used the Deepsort and Yolov5 method to track the moving trajectory of insects so that their number and trajectory during regurgitation could be recorded simultaneously.

## 2.2 Experimental equipment and environment

The computer equipment used for the behavior recognition experiments is Intel (R) Core (TM) i9-9,900 K CPU @ 3.60 GHz, NVIDIA GeForce RTX 2080Ti with 11G video memory, and the software development environment used was Ubuntu 20.04.1, Python 3.7, and Cuda 11.3. The software development environment used is Ubuntu 20.04.1 operating system, Python environment is 3.7, Cuda 11.3, deep learning framework is PyTorch 1.10.0.

The computer equipment used for the regurgitated liquid extraction and insect trajectory tracking experiments was an 11th Gen Intel (R) Core (TM) i5-11,400 H @ 2.70 GHz 2.69 GHz, and the graphics card was an NVIDIA GeForce RTX 3060 with 6 G of video memory. The Chinese version, python environment is 3.8, Cuda11.5, and deep learning framework is PyTorch 1.10.0.

## 2.3 Model performance metrics

The first part of the behavior recognition experiment was about classification Top-1 Accuracy was used to evaluate the model accuracy. Top-1 Accuracy and Top-5 Accuracy are important metrics used to evaluate the accuracy of the classification model. Top-1 Accuracy refers to tracking the category with the highest probability among the prediction labels as the prediction category, and if the prediction result is the same as the actual result, it is judged to be correct. Top-5 Accuracy refers to taking the top five categories with the highest probability in the prediction labels as the prediction categories, and if one of the categories is the same as the actual result, it is judged to be correct. In the behavior recognition experiment, only fruit fly regurgitation behavior and other behaviors were recognized; therefore, Top-1 Accuracy was chosen as the evaluation index.

In the second part of the regurgitation liquid extraction experiment, the main purpose is to evaluate the results of the semantic segmentation experiment and calculate the Miou of semantic segmentation (Miou is Mean Intersection over Union). In semantic segmentation, the intersection and merge ratio of a single category is the ratio of the intersection and merge of the true label to the predicted value of that category (Figure 1).

Here the positive cases refer to regurgitated liquids and the negative cases refer to non-regurgitated liquids.

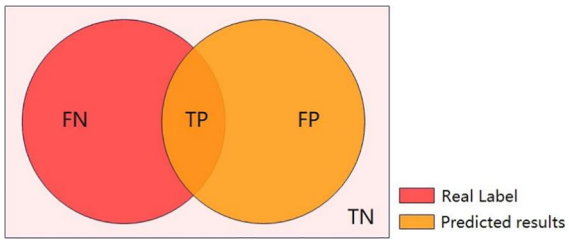



FIGURE 1
Where true positive (TP) represents that the model is predicted to be a positive case and is actually a positive case. False positive (FP) represents that the model is predicted to be a positive case but is actually a negative case. False negative (FN) represents that the model is predicted to be a negative case but is actually a positive case. True negative (TN) represents that the model is predicted to be a negative case and is actually a negative case.

MIoU is the average of the cross-merge ratio for each type of label in this data set. The calculation formula is as follows.

$$MIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}+\sum_{j=0}^{k}p_{ji}-p_{ii}} \quad (1)$$

where i denotes the true value, j denotes the predicted value, and p_ij denotes the prediction of i to j. Also equivalent to

$$MIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{TP}{FN+FP+TP} \quad (2)$$

In the third part of the trajectory tracking experiment, two metrics, precision and recall, were used to evaluate the effectiveness of Yolov5 in detecting fruit flies. Precision is a measure of accuracy that describes the number of predicted positive cases that are true positive cases. Here positive cases refer to fruit flies and negative cases refer to non-fruit flies, and are expressed as follows:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

Recall is a coverage metric that describes how many positive examples are selected from a true outcome perspective, with the following expression.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4)$$

The loss function serves as another evaluation metric for each model training, and is used to estimate the degree of inconsistency between the predicted and true values of the model. This is a non-negative real-valued function. The smaller the loss function, the better is the robustness of the model.

## 2.4 Data collection and processing

During the process of data collection, this study mainly targeted *Bactrocera minax* and *Bactrocera tau*. Both species were photographed at the Insect Ecology Laboratory at the College of Agriculture, Yangtze University. *B. minax* affects almost all fruits of the genus Citrus in the family Rutaceae, and its individuals are

relatively large; *B. tau* is smaller in size compared to *B. minax*, and mainly affects squash, cucumber, tomato and other fruits.

To enable the fruit fly to regurgitate, *B. minax* was fed honey water with a concentration of 5% and then placed in closed petri dishes one by one. A video of regurgitation behavior and other actions of the fruit fly was obtained by vertical filming using a Sony video camera (FDR-AX60) with a filming resolution of 1,920 × 1,080 pixels and a filming frame rate of 50 fps.

In the behavioral recognition experiment, the video was edited using several clips. In this study, the video of *B. minax* regurgitation was edited into 50 10 s clips, and that of other actions was edited into 50 10 s clips (other actions include various grooming behaviors and resting states).The 100 video clips collected were divided into training and validation sets at a ratio of 8:2.

In the semantic segmentation experiment, one image was extracted from every 20 frames of the regurgitation video as the dataset for semantic segmentation, and 200 images were obtained. The 200 images of spit water were divided into training and validation sets in 8:2.

In the trajectory tracking experiments, videos containing insects at rest, walking, and while regurgitating were selected, one image was extracted every 50 frames, and 300 images were obtained. The 300 images of the spit-water trajectories were divided into training and validation sets at an 8:2 ratio. This part of the experiment used *B. tau*, which is much smaller than *B. minax*. Because of its small size, it is more difficult to track and test whether the network meets the criteria for tracking insects. Additionally, the Petri dish is limited in size, so smaller-sized fruit flies will more random, which will make the experimental results more precise.

## 2.5 Regurgitation behavior recognition experiment

The behavior recognition task involves identifying different behavioral actions from the video, and the actions can occur continuously or intermittently. Behavior recognition seems to be an extension of the image classification task to multiframe detection, and then aggregating the predictions for each frame. Traditional behavior recognition focuses on feature extraction from videos. It extracts local high-dimensional visual features of video regions, combines them into fixed-size video-level descriptions, and uses classifiers for final prediction. With the development of deep-learning technology, 2D convolutional neural network (2DCNN) has been applied to behavior recognition. 2DCNN is a two-dimensional matrix of input; therefore, the input video is transformed into images, and the sliding window operation can only be performed on one frame of a single channel. This approach cannot consider inter-frame motion information in the time dimension; therefore, the application of 2DCNN in behavior recognition is not satisfactory. However, with a 3D convolutional neural network (3DCNN), behavior recognition can be performed more effectively. 3DCNN has three dimensions: image width, graphic height, and image channel. The convolutional kernel can move in three directions, and the input of one video outputs another video, which retains the input temporal information to better capture the temporal and spatial information in the video (Asadi-Aghbolaghi et al., 2017; Roig Ripoll, 2017; Urabe et al., 2018; Wu et al., 2019; Rastgoo et al., 2020).

In this study, three typical networks were used for the experiments: 3D Convolutional Networks (C3D) (Tran et al., 2015), inflated 3D ConvNet (I3D) (Carreira and Zisserman, 2017) and Expanding Architectures for Efficient Video Recognition (X3D) (Feichtenhofer, 2020). C3D can be regarded as a breakthrough because it is a relatively early proposal to apply the 3DCNN method to behavior recognition. The study proposes the application of 3D convolutional operations to extract spatial and temporal features from video data for behavior recognition. These 3D feature extractors operate in both the spatial and temporal dimensions, thus capturing motion information in the video stream. This structure can generate information channels from adjacent video frames and perform convolution and subsampling in each channel separately to combine the information from all channels to obtain the final features. Compared with 2DCNN, C3D networks are more suitable for learning spatio-temporal features, which can model temporal information by 3D convolution and 3D pooling, whereas 2D convolution can only learn features spatially. The I3D network transforms 2D into 3D, not only to process time repeatedly, which can be obtained by the temporal inflation of all filters and pooling kernels. The main advantage of this method is that the model parameters can be extended to 3D with pretrained 2D images, which solves the problem of not having 3D pretrained parameters. X3D is a relatively new network model with improvements based on previous networks. The previous 3D network mainly expanded the 2D convolutional neural network in the time dimension. However, expanding the time scale is not necessarily the best choice. It is worth expanding on other scales, such as the total frame length of input data, frame rate of input data, size of input frames, and network width and depth. The network eventually outperformed all previous networks in terms of accuracy while requiring only one-fifth of the previous computations and parameters, and it was found that the network could maintain a low number of channels while maintaining high input pixels.

The above networks are for human action behavior datasets in recognition, such as kinetics, UCF101, HMDB-51 and other datasets. For example, kinetics has 400 classes of datasets, each of which comes from a different YouTube video, and the corresponding human action is extracted from the video in a video segment of approximately 10 s.

The main idea of this study was to expand human behavior recognition to insect behavior recognition. Because insect behaviors are much smaller than human behaviors, we are not sure if the network model can have a good extraction of insect fine action features when detecting their behaviors. Therefore, we confirmed this problem experimentally. We labeled the prepared video clips with data, labeled each small video as an action, and then placed them into C3D, I3D, and X3D networks for training (Figure 2). For the recognition of insect spitting behavior, the assigned training sets were placed into C3D, I3D, and X3D networks for iterative training,

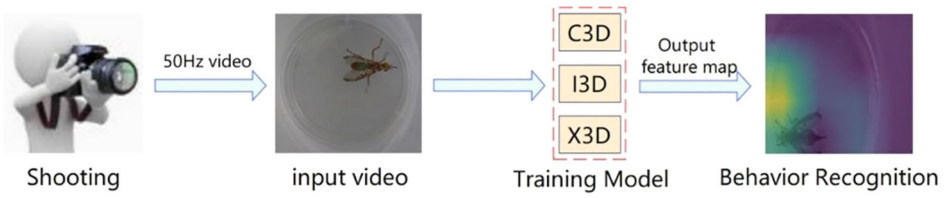



FIGURE 2
The trained model identifies the insect behavior by the output feature map.

and the training effects were compared. The model structure of the I3D for water-splitting behavior recognition is shown in Figure 3.

## 2.6 Regurgitation liquid extraction experiment

In addition to fruit flies, many insects exhibit regurgitation behavior, which has profound implications for insect ecology and plant interactions. Upon detecting the regurgitation phenomenon in fruit flies during the experiments, we performed semantic segmentation on their regurgitant. Subsequently, threshold segmentation was applied to calculate the area of the regurgitant. Through these research steps, we could accurately measure the area of the droplet, providing crucial data for the quantitative study of fruit fly regurgitation behavior.

In segmenting the regurgitated liquid, the Unet network was firstly used. Why was the Unet network selected? We were inspired by medical image segmentation. Medical semantics are simpler and more fixed in structure. The organ itself is fixed in structure and is not particularly rich in semantic information; therefore, high-level semantic information and low-level features are important. The skip connection and U-shaped structure of Unet combine high-level semantic information and low-level features, making it more suitable for medical semantic segmentation. The fruit fly regurgitated image features are similar to medical images. In other words, the regurgitated liquid resembled a group of ellipse-shaped cells. Its structure is more fixed and the semantic structure is relatively simple; therefore, accurate segmentation is required (Fayemiwo et al., 2021; Zunair and Ben Hamza, 2021; Wang et al., 2022)

In this study, we evaluated three distinct neural network models, namely C3D, I3D, and X3D, to ascertain their performance and applicability in our domain. As detailed in Table 1, each model was configured with identical hyperparameters to ensure a fair and consistent comparison. Specifically, we employed a batch size of 16, a momentum value of 0.9, and utilized the SGD optimizer for training. The initial learning rate was set to 1 X 10-4 across all models. Moreover, each model underwent training for a total of 50 epochs. This uniform setup across the different models allowed us to directly compare their performance and isolate the effects of their unique architectural differences on the task at hand.

To obtain better segmentation results, we attempted to modify the backbone network of Unet using Vgg16 and ResNet50, and then added the CBAM attention mechanism to it, which further improved makes the segmentation effect. The model structure is shown in Figure 4. To make the experiments more scientific, ablation experiments are also done in this paper. The semantic segmentation network deeplabv3+ was used, and Xception and MobileNetv2 were also used as the backbone network of DeeplabV3+(Nagrath et al., 2021; Campos et al., 2022; Sutaji and Yildiz, 2022; Xi et al., 2022). The training hyperparameter settings and results are presented in Table 2.

According to Equation 1 and Equation 2. The mIoU in Table 2 serves as a key performance indicator for each model, reflecting the

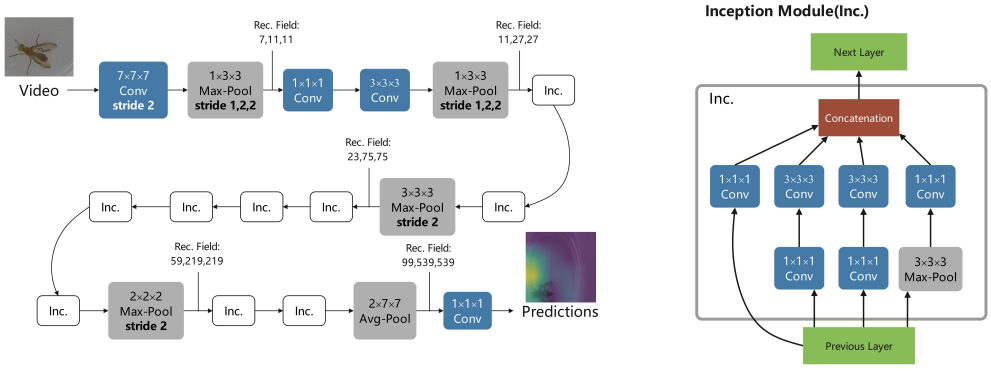



FIGURE 3
I3D behavior detection model architecture.

TABLE 1 Values of the hyperparameters for the three different network models evaluated in the study.

| Model | Batch | Momentum | Optimizer | Initial learning rate | Training epochs |
|---|---|---|---|---|---|
| C3D | 16 | 0.9 | SGD | 1e−4 | 50 |
| I3D | 16 | 0.9 | SGD | 1e−4 | 50 |
| X3D | 16 | 0.9 | SGD | 1e−4 | 50 |

SGD, stochastic gradient descent.

average accuracy of the model in segmenting different classes within images. Higher mIoU values signify better segmentation performance, with the Unet + CBAM (Vgg16) model showing the highest mIoU of 90.96%, indicating superior accuracy in class segmentation compared to other models like Unet with ResNet50 or deeplabV3+ with Xception and MobileNetv2. As shown in Table 2, Unet's accuracy is higher when using vgg16 as the backbone, and combining it with the CBAM attention mechanism. Therefore, the training weights of this model were chosen to segment the randomly selected fruit fly regurgitated images. Before segmentation, a square millimeter piece of labeled paper was placed in a petri dish as a "scale" and photographed together with the fruit fly so that the bottom area of the regurgitated liquid could be derived from the pixels of the regurgitated liquid through the area and pixels of the paper.

After the segmentation, the extracted regurgitated liquid can be clearly seen, but the segmented image contains impurities, such as the fruit fly themselves and the tiny impurities on the petri dish, which not only bring visual disturbance, but also affect the next step of calculating the regurgitated liquid area. Therefore, the threshold segmentation method can be used to remove impurities and background. Since only regurgitated liquid and marker paper pieces need to be retained, we chose binarization the simplest threshold segmentation, to assign black values to all the impurities and background, and to keep and deepen the color of regurgitated water droplets and marker paper pieces, to obtain a completely extracted regurgitated liquid picture. The extraction process is illustrated in Figure 5.

The number of closed shapes in the image and the pixels of each closed shape were calculated using OpenCV, and the area of each liquid was obtained by marking the pixels and area of the paper sheet.

## 2.7 Trajectory tracking experiment

The Yolov5 target detection algorithm combined with the DeepSort algorithm, which has a good tracking performance at present, is used in this study to track the trajectory of fruit flies in the process of regurgitation, and it can realize the counting of fruit flies. We also used the DeepSort network, the most important feature of which is the use of the Kalman filtering algorithm and Hungarian algorithm, both of which can significantly improve the accuracy and speed of multi-object tracking. The Kalman filtering algorithm is divided into two processes: prediction and updating. Prediction: When the target is moved, the target frame position, speed, and other parameters of the current frame are predicted using the target and speed parameters of the previous frame. Update: The two positively distributed states of the predicted and observed values were linearly weighted to obtain the transition state predicted by the current state. In other words, the Kalman filter can predict the position of the current moment based on the position of the target at the previous moment, and can estimate the position of the target more accurately than the sensor. The Hungarian algorithm mainly calculates the similarity to obtain the similarity matrix of the two frames before and after, to determine whether the target in the current frame is the same as the target in the previous frame (Gao and Lee, 2021; Kim and Lee, 2021; Sujin et al., 2021; Ren et al., 2022; Song et al., 2022).

Although the DeepSort network has high accuracy and speed in multi-object tracking, it is mostly used for pedestrian and vehicle tracking and counting. It can achieve good results in tracking objects with relatively large targets and obvious features, but it is seldom used for insect trajectory tracking and counting (Chen et al., 2022; Zhang, 2022). This is because insects are small in size, relatively inconspicuous in features, their trajectories are much smaller than those of straight-line vehicles and pedestrians, and

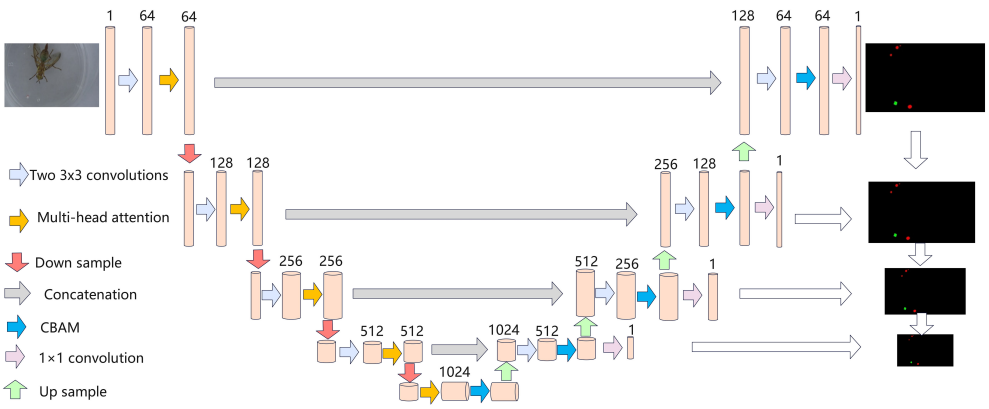



FIGURE 4
Structure diagram of Unet and CBAM attention mechanism.

TABLE 2 Model performance metrics at different training hyperparameter settings for the two convolutional neural networks evaluated in the study.

| Model | Backbone | Optimizer | Initial learning rate | Miou | Loss |
|---|---|---|---|---|---|
| Unet | Vgg16 | adam | 1e−4 | 89.5 | 0.056 |
| Unet + CBAM | Vgg16 | adam | 1e−4 | 90.96 | 0.055 |
| Unet | ResNet50 | adam | 1e−4 | 85.20 | 0.096 |
| Unet + CBAM | ResNet50 | adam | 1e−4 | 85.95 | 0.077 |
| deeplabV3+ | Xecption | SGD | 7e−3 | 80.69 | 0.3144 |
| deeplabV3+ | MobileNetv2 | SGD | 7e−3 | 80.66 | 0.3211 |

there is no obvious motion pattern. No clear pattern of movement was observed. In this study, we used the DeepSort network to track insects and explored whether there is a network model that can meet the requirements of tracking insects.

Therefore, 270 images of fruit flies were used to train the Yolov5 network, and 30 images were used to verify its effect. As shown in Figure 6, we utilize precision and recall, as defined in Equations 3 and 4, to evaluate the training and testing performance of the YoloV5 model. These metrics effectively reflect the model's accuracy and reliability in object detection tasks. after 50 iterations of training, the accuracy of the network reached 99.8%. The best weights of Yolov5 training were used as the weights of DeepSort object tracking for the tracking experiments. Two South Asian solid flies were detected and tracked using a video.

# 3 Results and discussion

## 3.1 Regurgitation behavior recognition

The same video data and labels were input into the model for training, and the experimental results with the same configuration parameters are listed in Table 3. Although the X3D network is more advanced and has better performance in human behavior recognition, it was found through this experiment that after applying it to insect regurgitation behavior recognition, the I3D network performs better with higher accuracy and less training loss, as shown in Figure 7. However, from the perspective of training time, the I3D model was still slightly inadequate, and the training time for each item was longer than that of the other models. The training time does not directly affect the detection effect of the behavior recognition experiments; however, if the model is promoted, it will affect the efficiency of the experiment when training many different datasets. Therefore, improving the network model will also become the main focus of later experiments. We planned to replace the network backbone with a lightweight network such as MobileNet, which can improve the training speed while ensuring accuracy (Table 3).

From the results, we find that the idea of transferring human behavior recognition to insect behavior recognition is feasible. All three networks can effectively recognize the regurgitation behavior of fruit flies, and we chose the I3D model as the best performer. This also suggests that we can extend this experiment to behavioral recognition beyond fruit fly regurgitation to recognize all behaviors of fruit flies and similar insects, including forefoot, hindfoot, and head grooming behaviors.

## 3.2 Regurgitation liquid extraction

The regurgitated liquid picture was segmented twice and the liquid area calculated by OpenCV is shown in Table 4.

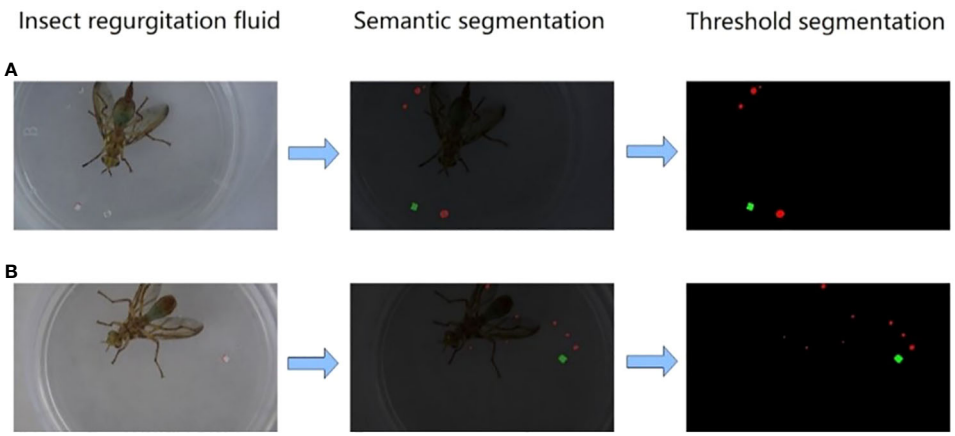



FIGURE 5
**(A, B)** represent two different sets of samples, and after semantic segmentation and threshold segmentation, a clearer position and shape of the regurgitated liquid can be extracted.

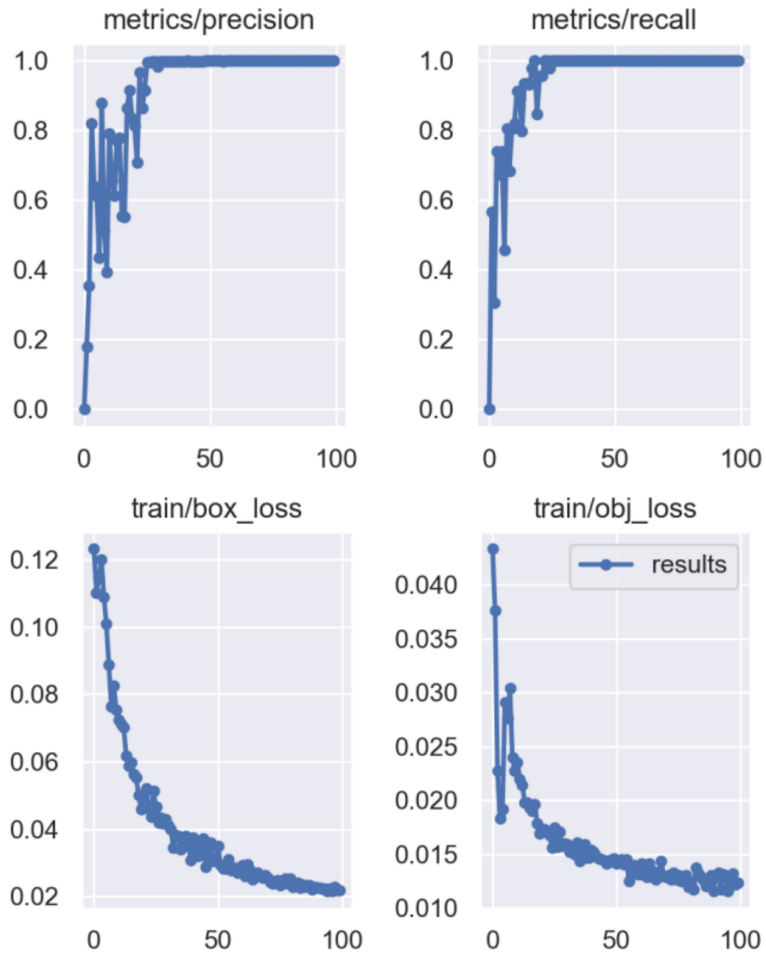



FIGURE 6
Training accuracy, recall rate and training loss graph of Yolov5 network.

This method allows for a general assessment of the amount of regurgitation. Although there are no precise results, this is still beneficial to the analysis of regurgitation research. The experimental results were not completely accurate because the liquid beads were too small, the resolution of the pictures was not high enough, and the liquid beads and background were blurred when the pictures were enlarged and labeled, resulting in less accurate labeling. In addition, the colors of the background and the liquid beads are similar, which makes it difficult to distinguish the liquid beads from the background.

When studying insect regurgitation in a real environment, such as a fruit fly, it is difficult to observe it with the naked eye. The regurgitation behavior occurs for a short period of time and at a high frequency, so it is difficult to know the number of regurgitations even if it is observed with the naked eye. In contrast, the method used in our experiment can clearly mark the number of regurgitated water droplets and extract the approximate area. As mentioned in the introduction, house fly regurgitation spots bring *E. coli* O157:H7 to the vegetables. This problem can be solved by extracting regurgitated liquid. As the study of insects goes deeper, the extraction of insect footprints and the segmentation of individuals are also important, as they can extract different insects from the complex environment and realize the study of insect numbers and individual behavior in many aspects, which is also a prerequisite step for the intelligent diagnosis of pests.

TABLE 3 Model performance metrics for the three different network models evaluated in the study.

| Network Model | Top-1 Accuracy | Loss | Training times/item |
|---|---|---|---|
| C3D | 0.95 | 0.170 | 4.67 |
| I3D | 0.963 | 0.101 | 5.01 |
| X3D | 0.925 | 0.215 | 4.77 |

## 3.3 Trajectory tracking

To observe the trajectory of the fruit flies more clearly, two recording methods were used. The observer can clearly see the trajectory of the fruit fly by combining the two videos as shown in Figure 8.

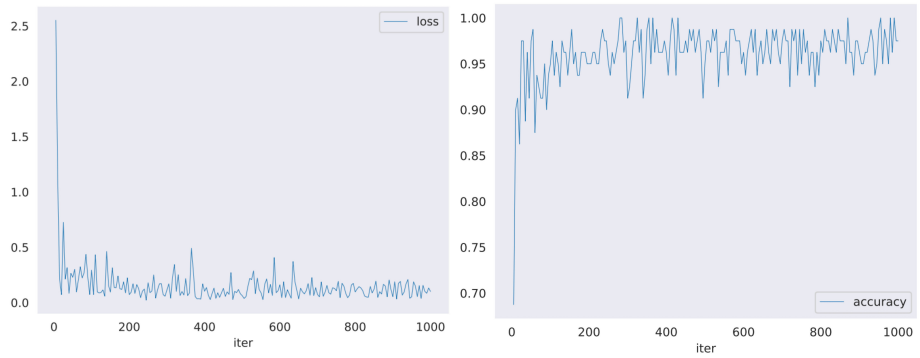



FIGURE 7
Training accuracy and loss diagram of I3D network.

**TABLE 4** Number and area of regurgitated liquid beads extracted in Figures 3A, B.

| Insect regurgitation liquid | Item | Pixel (pt) | Area ($mm^2$) | Totalarea ($mm^2$) |
|---|---|---|---|---|
| a | piece of labeled paper | 2,455.5 | 1 | 1 |
| | Liquid bead No.1 | 894.5 | 0.36 | 1.27 |
| | Liquid bead No.2 | 165.5 | 0.06 | |
| | Liquid bead No.3 | 845.0 | 0.34 | |
| | Liquid bead No.4 | 242.5 | 0.10 | |
| | Liquid bead No.5 | 62.5 | 0.03 | |
| | Liquid bead No.6 | 331.0 | 0.13 | |
| | Liquid bead No.7 | 325.0 | 0.13 | |
| | Liquid bead No.8 | 293.0 | 0.12 | |
| b | piece of labeled paper | 2,059.5 | 1 | 1 |
| | Liquid bead No.1 | 235.5 | 0.11 | 2.64 |
| | Liquid bead No.2 | 2,759.5 | 1.34 | |
| | Liquid bead No.3 | 760.5 | 0.37 | |
| | Liquid bead No.4 | 1,698.5 | 0.82 | |

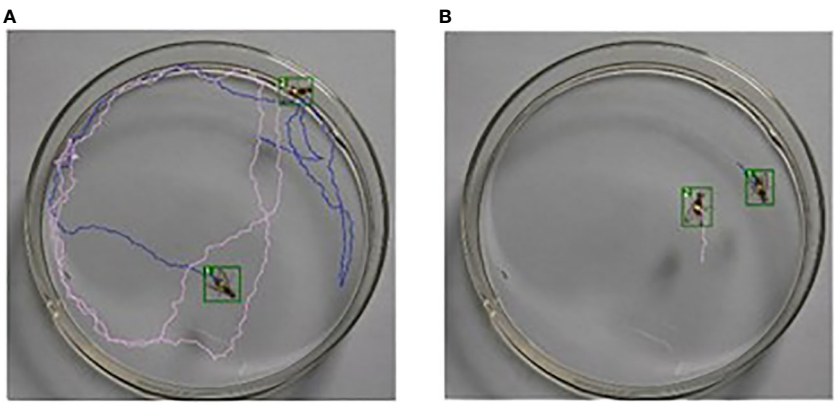



FIGURE 8
**(A)** the petri dish recording all the trajectories and **(B)** the petri dish recording the first 30 frames of the fruit fly trajectory. The latter avoided the messiness of the trajectory.

Tracking the trajectory of fruit flies during their regurgitation can complete the study of fruit fly regurgitation. Additionally, we can apply this model to track and count the trajectories of other insects, which can help analyze insect movement patterns, male/female relationships, and predatory behavior.

## 4 Conclusions

This work demonstrates the feasibility of using deep learning and computer vision techniques to study regurgitation in fruit flies. The proposed method was divided into three main parts. The first is to apply the I3D network to fruit fly regurgitation behavior recognition, the accuracy of which can reach 96.3%. The second step is to segment the extracted regurgitation liquid. The proposed Unet combined with the CBAM attention mechanism model achieves an MIOU of 90.96%, which is 1.46 percentage points higher than the original Unet network and 5–10 percentage points higher than the other network models. Then, threshold segmentation was conducted to obtain the regurgitated liquid quantity and area. The third was to track the trajectory of the fruit fly during its regurgitation by Yolov5+DeepSort. The accuracy of Yolov5 detection of fruit flies was stabilized at 99.8%, and the final tracking effect was satisfactory. The method can be used in fruit fly and other insect regurgitation studies; more importantly, it can be extended to take advantage of deep learning to solve manual observation problems and apply to more insect research tasks according to different needs, which can realize non-destructive research and real-time monitoring of insects.

The current study is only part of fruit fly regurgitation research, and the following three parts will be implemented based on this study. 1. Identification of changes in the mouthparts of fruit flies during regurgitation. When ruminating, the mouthparts of the fruit fly perform specific movements. Although this study was able to detect regurgitation behavior through this feature, there was no detection or visual analysis of the specific movement changes and movement occurrence pattern of the mouthparts. 2. Estimation of the volume of liquid beads regurgitated by fruit fly. In the current research, we extracted the regurgitated liquid using filter paper, which is a traditional method. However, there are errors in this method, and the experimental process is more complicated and time-consuming. An artificial intelligence method to automatically measure the regurgitated liquid volume will be of great help in fruit fly regurgitation research. 3. Realize counting of regurgitated liquid per unit time. This study can realize the counting of regurgitated liquid beads of fruit fly, but our limitation is that it cannot be set flexibly to count the number of regurgitated liquid beads per unit time, which will also be the next focus of work. It is also particularly important to count regurgitated fluid beads in different fruit flies at different times depending on the different experimental studies.

## Data availability statement

The original contributions presented in the study are included in the article/supplementary files. Further inquiries can be directed to the corresponding author.

## Ethics statement

The manuscript presents research on animals that do not require ethical approval for their study.

## Author contributions

TZ: Conceptualization, Data curation, Formal analysis, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. WZ: Funding acquisition, Project administration, Resources, Supervision, Validation, Writing – review & editing. MX: Data curation, Formal analysis, Software, Visualization, Writing – original draft, Writing – review & editing.

## Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This research was funded by China University Industry-University-Research Innovation Fund “New Generation Information Technology Innovation Project” 2020ITA03012.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

# References

Ant, T., Koukidou, M., Rempoulakis, P., Gong, H.-F., Economopoulos, A., Vontas, J., et al. (2012). Control of the olive fruit fly using genetics-enhanced sterile insect technique. *BMC Biol.* 10. doi: 10.1186/1741-7007-10-51

Asadi-Aghbolaghi, M., Bertiche, H., Roig, V., Kasaei, S., and Escalera, S. (2017). "Action recognition from rgb-D data: comparison and fusion of spatio-temporal handcrafted features and deep strategies," in *Proceedings Of The Ieee International Conference On Computer Vision Workshops*. 3179–3188.

Badii, K., Billah, M., Afreh-Nuamah, K., Obeng-Ofori, D., and Nyarko, G. (2015). Review of the pest status, economic impact and management of fruit-infesting flies (Diptera: tephritidae) in Africa. *Afr. J. Agric. Res.* 10 (12), 1488–1498. doi: 10.5897/AJAR2014.9278

Cáceres, C., Tsiamis, G., Yuval, B., Jurkevitch, E., and Bourtzis, K. J. B. M. (2019). Joint FAO/IAEA coordinated research project on "use of symbiotic bacteria to reduce mass-rearing costs and increase mating success in selected fruit pests in support of SIT application". *BMC Microbiol.* 19, 1–2.

Campos, J., Yee, A., and Vega, I. F. (2022). Simplifying vgg-16 for plant species identification. *IEEE Latin America Trans.* 20, 2330–2338. doi: 10.1109/TLA.2022.9904757

Carreira, J., and Zisserman, A. (2017). "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proceedings Of The Ieee Conference On Computer Vision And Pattern Recognition*. 6299–6308.

Chen, X., Jia, Y., Tong, X., and Li, Z. (2022). Research on pedestrian detection and deepsort tracking in front of intelligent vehicle based on deep learning. *Sustainability* 14. doi: 10.3390/su14159281

Dias, N. P., Zotti, M. J., Montoya, P., Carvalho, I. R., and Nava, D. E. (2018). Fruit fly management research: A systematic review of monitoring and control tactics in the world. *Crop Prot.* 112, 187–200. doi: 10.1016/j.cropro.2018.05.019

Díaz-Fleischer, F., Arredondo, J., Lasa, R., Bonilla, C., Debernardi, D., Pérez-Staples, D., et al. (2019). Sickly sweet: insecticidal polyols induce lethal regurgitation in dipteran pests. *Insects* 10 (2), 53. 10, 53. doi: 10.3390/insects10020053

Fayemiwo, M. A., Olowookere, T. A., Arekete, S. A., Ogunde, A. O., Odim, M. O., Oguntunde, B. O., et al. (2021). Modeling A deep transfer learning framework for the classification of covid-19 radiology dataset. *Peerj Comput. Sci.* 7. doi: 10.7717/peerj-cs.614

Feichtenhofer, C. (2020). "X3d: expanding architectures for efficient video recognition," in *Proceedings Of The Ieee/Cvf Conference On Computer Vision And Pattern Recognition*. 203–213.

Gao, g., and Lee, S. (2021). Design and implementation of fire detection system using new model mixing yolov5 and deepsort. *Int. J. Of Advanced Culture Technol.* 9, 260–267.

Guillén, L., Pascacio-Villafán, C., Stoffolano, J. G. Jr., López-Sánchez, L., Velázquez, O., Rosas-Saito, G., et al. (2019). Structural differences in the digestive tract between females and males could modulate regurgitation behavior in anastrepha ludens (Diptera: tephritidae). *J. Insect Sci.* 19 (4), 7.

Hong, S., Zhan, W., Dong, T., She, J., Min, C., Huang, H., et al. (2022). A recognition method of bactrocera minax (Diptera: tephritidae) grooming behavior via A multi-object tracking and spatio-temporal feature detection model. *J. Of Insect Behav.* 35, 67–81. doi: 10.1007/s10905-022-09802-7

Kim, H.-T., and Lee, S.-H. (2021). A study on object distance measurement using opencv-based yolov5. *Int. J. Of Advanced Culture Technol.* 9, 298–304.

Lasa, R., Ortega, R., and Rull, J. (2013). Towards development of A mass trapping device for mexican fruit fly anastrepha ludens (Diptera: tephritidae) control. *Florida Entomologist* 96, 1135–1142. doi: 10.1653/024.096.0354

Nagrath, P., Jain, R., Madan, A., Arora, R., Kataria, P., and Hemanth, J. (2021). Ssdmnv2: A real time dnn-based face mask detection system using single shot multibox detector and mobilenetv2. *Sustain. Cities And Soc.* 66, 102692. doi: 10.1016/j.scs.2020.102692

Navarro-Llopis, V., Vacas, S. J. T., The Detection, C., and Flies, R. O. T. F. (2014). "Mass trapping for fruit fly control," in Trapping and the detection, control, and regulation of tephritid fruit flies: lures, area-wide programs, and trade implications., 513–555. doi: 10.1007/978-94-017-9193-9_15

Rastgoo, R., Kiani, K., and Escalera, S. J. E. S. W. A. (2020). Hand sign language recognition using multi-view hand skeleton. *Expert Syst. Appl.* 150, 113336. doi: 10.1016/j.eswa.2020.113336

Ren, J., Wang, Z., Zhang, Y., and Liao, L. (2022). Yolov5-R: lightweight real-time detection based on improved yolov5. *J. Of Electronic Imaging* 31. doi: 10.1117/1.JEI.31.3.033033

Roig Ripoll, V. (2017). Multimodal 2DCNN action recognition from RGB-D data with video summarization. MS thesis. Universitat PolitÃ© cnica de Catalunya.

She, J., Zhan, W., Hong, S., Min, C., Dong, T., Huang, H., et al. (2022). A method for automatic real-time detection and counting of fruit fly pests in orchards by trap bottles via convolutional neural network with attention mechanism added. *Ecol. Inf.* 70, 101690. doi: 10.1016/j.ecoinf.2022.101690

Song, H., Zhang, X., Song, J., and Zhao, J. (2022). Detection and tracking of safety helmet based on deepsort and yolov5. *Multimed. Tools Appl.*. 82 (7), 10781–10794.

Sujin, Y., Jung, I., Kangdonghwa,, and Baek, H. (2021). Real-time multi-object tracking using mixture of sort and deepsort. *J. Of Korean Institute Of Inf. Technol.* 19, 1–9.

Sutaji, D., and Yildiz, O. (2022). Lemoxinet: lite ensemble mobilenetv2 and xception models to predict plant disease. *Ecol. Inf.* 70. doi: 10.1016/j.ecoinf.2022.101698

Sword, G. A. (2001). Tasty on the outside, but toxic in the middle: grasshopper regurgitation and host plant-mediated toxicity to A vertebrate predator. *Oecologia* 128, 416–421. doi: 10.1007/s004420100666

Timilsena, B. P., and Mikó, I. J. R. I. (2017). Know your insect: the structural backgrounds of regurgitation, A case study on manduca sexta and heliothis virescens (Lepidoptera: sphingidae, noctuidae). *Res. Ideas Outcomes* 3, e11997. doi: 10.3897/rio.3.e11997

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings Of The Ieee International Conference On Computer Vision*. 4489–4497.

Urabe, S., Inoue, K., and Yoshioka, M. (2018). "Cooking activities recognition in egocentric videos using combining 2dcnn and 3dcnn," in *Proceedings Of The Joint Workshop On Multimedia For Cooking And Eating Activities And Multimedia Assisted Dietary Management*. 1–8.

Vayssieres, J.-F., Korie, S., Coulibaly, O., Van Melle, C., Temple, L., and Arinloye, D. (2009). The mango tree in central and Northern Benin: damage caused by fruit flies (Diptera tephritidae) and computation of economic injury level. *Fruits* 64, 207–220. doi: 10.1051/fruits/2009016

Wang, J., Yu, Z., Luan, Z., Ren, J., Zhao, Y., and Yu, G. (2022). Rdau-net: based on A residual convolutional neural network with dfp and cbam for brain tumor segmentation. *Front. In Oncol.* 12. doi: 10.3389/fonc.2022.805263

Wasala, L., Talley, J., Fletcher, J., and Wayadande, A. (2011). House fly regurgitation spots may be A source of E-coli O157:H7 contamination of leafy greens. *Phytopathology* 101, S188–S188.

Wu, D., Chen, J., Sharma, N., Pan, S., Long, G., and Blumenstein, M. (2019). "Adversarial action data augmentation for similar gesture action recognition," in *2019 International Joint Conference On Neural Networks (Ijcnn)*. 1–8.

Xi, D., Qin, Y., and Wang, Z. (2022). Attention deeplabv3 model and its application into gear pitting measurement. *J. Of Intelligent Fuzzy Syst.* 42, 3107–3120. doi: 10.3233/JIFS-210810

Zhang, Q. (2022). Multi-object trajectory extraction based on yolov3-deepsort for pedestrian-vehicle interaction behavior analysis at non-signalized intersections. *Multimed. Tools Appl.* 82 (10), 15223–15245.

Zunair, H., and Ben Hamza, A. (2021). Sharp U-net: depthwise convolutional network for biomedical image segmentation. *Comput. In Biol. And Med.* 136. doi: 10.1016/j.compbiomed.2021.104699

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 15 January 2024
DOI 10.3389/fpls.2023.1323301

Check for updates

OPEN ACCESS

EDITED BY
Lei Shu,
Nanjing Agricultural University, China

REVIEWED BY
Marcin Wozniak,
Silesian University of Technology, Poland
Jakub Nalepa,
Silesian University of Technology, Poland

*CORRESPONDENCE
Wen-Hao Su
wenhao.su@cau.edu.cn

RECEIVED 17 October 2023
ACCEPTED 26 December 2023
PUBLISHED 15 January 2024

CITATION
Lv M and Su W-H (2024) YOLOV5-CBAM-C3TR: an optimized model based on transformer module and attention mechanism for apple leaf disease detection. *Front. Plant Sci.* 14:1323301. doi: 10.3389/fpls.2023.1323301

COPYRIGHT
© 2024 Lv and Su. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# YOLOV5-CBAM-C3TR: an optimized model based on transformer module and attention mechanism for apple leaf disease detection

Meng Lv and Wen-Hao Su*

College of Engineering, China Agricultural University, Beijing, China

Apple trees face various challenges during cultivation. Apple leaves, as the key part of the apple tree for photosynthesis, occupy most of the area of the tree. Diseases of the leaves can hinder the healthy growth of trees and cause huge economic losses to fruit growers. The prerequisite for precise control of apple leaf diseases is the timely and accurate detection of different diseases on apple leaves. Traditional methods relying on manual detection have problems such as limited accuracy and slow speed. In this study, both the attention mechanism and the module containing the transformer encoder were innovatively introduced into YOLOV5, resulting in YOLOV5-CBAM-C3TR for apple leaf disease detection. The datasets used in this experiment were uniformly RGB images. To better evaluate the effectiveness of YOLOV5-CBAM-C3TR, the model was compared with different target detection models such as SSD, YOLOV3, YOLOV4, and YOLOV5. The results showed that YOLOV5-CBAM-C3TR achieved mAP@0.5, precision, and recall of 73.4%, 70.9%, and 69.5% for three apple leaf diseases including Alternaria blotch, Grey spot, and Rust. Compared with the original model YOLOV5, the mAP 0.5increased by 8.25% with a small change in the number of parameters. In addition, YOLOV5-CBAM-C3TR can achieve an average accuracy of 92.4% in detecting 208 randomly selected apple leaf disease samples. Notably, YOLOV5-CBAM-C3TR achieved 93.1% and 89.6% accuracy in detecting two very similar diseases including Alternaria Blotch and Grey Spot, respectively. The YOLOV5-CBAM-C3TR model proposed in this paper has been applied to the detection of apple leaf diseases for the first time, and also showed strong recognition ability in identifying similar diseases, which is expected to promote the further development of disease detection technology.

KEYWORDS

**deep learning, apple leaf, disease detection, YOLOv5, attention mechanism, transformer encoder**

# 1 Introduction

Apples are highly prized for their nutritional richness and rank among the world's most economically significant fruits (Shu et al., 2019). However, due to environmental, bacterial, and insect pests, the growth of apple fruits can be attacked by a variety of diseases, which can lead to a decrease in fruit yield and quality, resulting in huge economic losses. Timely detection and accurate classification of the type of disease is the first step to early control of the disease. The leaves of apple trees occupy most of the area of the tree and are the easiest part to observe. Most apple diseases can be identified by observing diseased leaves (Wang et al., 2009; Vishnu and Rajanith, 2015). Therefore, the research in this study focuses on diseases of apple leaf parts.

Traditionally, the identification of apple leaf disease mostly relied on experienced farmers to identify the disease. However, due to the similarity of diseases or the complexity of symptoms, relying on human eye detection can easily lead to misjudgment of diseases, which can not only solve the problem of diseases but also cause environmental pollution (Liu et al., 2022). The combination of machine learning and image processing replaced human eye detection and provided a new direction for disease detection. For example, Dubey and Jalal (2012) used K-means clustering for the segmentation of apple fruit diseases, then global color histogram, color coherence vector, local binary pattern, and complete local binary pattern were used for feature extraction, the support vector machine (SVM) (Hearst et al., 1998) was used for disease classification, which can achieve an accuracy of 93%. Chuanlei et al. (2017) introduced a method for apple leaf disease detection. To improve the detection accuracy, a region-growing algorithm is used to segment the disease image, a genetic algorithm combined with correlation feature selection is used to select the important features, and finally SVM classifier is used to identify the disease, which was tested on a dataset containing 90 images on a dataset with an accuracy of 90%. Shi et al. (2017) proposed an apple disease recognition method based on two-dimensional subspace learning dimensionality reduction, with recognition accuracy above 90% on the apple leaf disease dataset. Gargade and Khandekar (2021) used K-NN and SVM algorithms to classify apple leaf defects with 99.5% accuracy. Jan and Ahmad (2020) used 11 apple leaf image features and a multilayer perceptron (MLP) pattern classifier to detect apple Alternaria leaf blotch with 99.1% accuracy. However, segmentation based on image processing and feature extraction based on traditional machine learning are extremely complex, leading to inefficient disease diagnosis.

In recent years, convolutional neural network (CNN)-based model avoids complex preprocessing work on images and automatically extracts features through an end-to-end approach (Abade et al., 2021; Dhaka et al., 2021), which is more suitable for solving problems in the field of computer vision. Apple leaf disease detection tasks can be classified into three main categories according to the type of computer vision tasks: 1) image classification, which classifies the detected images into various types of diseases, 2) target detection, which detects and locates the diseases in the images, and 3) image segmentation, which segments the images into semantic disease maps. In general, image classification studies using CNN models are the most common. Based on Densenet-121, Zhong and Zhao (2020) proposed regression, multi-label classification and focal loss function recognition methods for three apple leaf diseases with accuracies of 93.51%, 93.31% and 93.71%, respectively. Yu and Son (2019) used the ROI-aware DCNN model to classify Marssonia blotch and Alternaria leaf spots, which was shown to outperform traditional methods. Singh et al. (2021) improved the classical CNN model to implement Marssonia Coronaria, Rust, and Scab for accurate classification with up to the accuracy of 99.2%. Babu and Ram (2022) proposed a deep residual convolutional neural network (DRCNN) with contrast limited adaptive histogram equalization for weed and soybean crop classification with an accuracy of 97.25%. Kundu et al. (2021) proposed the use of deep learning in conjunction with IoT for automatic detection of pearl millet diseases, and the accuracy of the proposed custom network model is comparable to that of the current state-of-the-art image classification model, with an accuracy of up to 98.78%, while greatly reducing the training time. Image classification methods are excellent at accurately classifying diseases, but their utility is limited by failing to provide information about the location of the disease. In contrast, target detection methods can locate the target object in real-time and obtain more detailed information, which is more conducive to practical application. Currently, target detection methods can be classified into single-stage and two-stage algorithms. Two-stage algorithms such as Faster-RCNN (Ren et al., 2015) and Mask-RCNN (Kaiming et al., 2017) have higher detection accuracy but lose detection speed. In comparison, single-stage algorithms are characterized by a small number of model parameters and fast inference speed, which better meet the needs of practical production environments. The single-stage algorithms are best known as you only look once (YOLO) (Redmon et al., 2016), which turns the detection task into a simple regression problem and has a simple network model that is easy for researchers to learn and train. Although there have been many iterations of YOLO, YOLOV5 remains the most widely used version across all domains (Lang et al., 2022). For example, Chen et al. (2022) added the SE module to YOLOV5 and replaced the original loss function GIOU with EIOU to automatically identify diseases on rubber trees, finally the average accuracy was improved by 5.4% compared to the original YOLOV5. With the aim of improving the accuracy of vegetable disease detection in natural environments, Li (2022) improved the CSP, FPN, and NMS modules in YOLOV5s, and finally achieved a mAP of up to 93.1% on a dataset containing a total of 1,000 images of five diseases. In order to accurately identify and locate tomatoes, Li et al. (2023) optimized YOLOV5 by adding the CARAFE module to obtain a larger sensory field while maintaining lightness, introducing EIOU and quality focal loss to solve the problem of uneven samples, and finally proposing YOLOv5s-CQE. The mAP 0.5of YOLOv5s-CQE on the dataset containing 3,820 tomato images finally reaches 98.68%. Therefore, YOLOV5 shows excellent detection accuracy and fast processing speed in a series of target detection tasks and shows great potential in the automatic identification and classification of apple leaf diseases.

In this study, aiming to achieve accurate detection of three common apple leaf diseases in the natural environment, YOLOV5

was selected as the baseline model, and YOLOV5-CBAM-C3TR was finally proposed by adding different attention mechanisms and C3TR modules and transformer encoders individually or jointly. The specific objectives of this study are as follows: (1) A proposal for an improved YOLOV5 method based on CBAM and C3TR modules for the identification of three apple leaf diseases including Alternaria blotch, Grey spot, and Rust; (2) Comparison of the performance of YOLOV5-CBAM-C3TR, SSD, YOLOV3, YOLOV4, YOLOV5 and other different target detection models on the same dataset containing three diseases; (3) Comparison of the performance improvement of YOLOV5 with the addition of CBAM, SE, ECA and C3TR modules, individually or in combination; (4) development of a model for the effective classification of similar apple leaf diseases. As far as we know, this is the first time that the YOLOV5-CBAM-C3TR model has been used for the identification and localization of apple leaf diseases.

# 2 Materials and methods

## 2.1 Datasets

In this study, the images were collected from the publicly available apple leaf pathology image dataset (https://aistudio.baidu.com/datasetdetail/11591). Disease images in natural environments in the dataset were obtained from a real apple orchard in Yantai, Shandong Province, China. A total of 390 high-quality images of three common apple leaf diseases were selected for study in this dataset. However, the original images cannot be trained, validated, and tested directly. Images used for target detection need to determine the location of the target in the dataset image, which requires the researcher to label the observed targets before starting training, validation, and testing (Wang and Zhao, 2022). The dataset used in this study was in YOLO format and manually labeled for apple leaf diseases using image annotation software. In order to facilitate model training, the labeled images were divided into training, validation, and test sets in a ratio of 8:1:1. In addition, to better adapt the model to different environments and to reduce the negative effects of photometric distortion during training (Zhu et al., 2021), data enhancements such as random cropping, panning, changing luminance, adding noise, rotating, and mirroring were chosen to extend the dataset. Finally, the dataset required for the experiment consisted of 3900 images of apple leaves containing the three diseases, 1680 from the laboratory background and 2220 from the orchard background, as shown in Table 1.

## 2.2 Methods

### 2.2.1 YOLOV5

The aim of this study is to achieve real-time detection and accurate identification of apple leaf diseases. Considering the type of disease detected including early-stage disease, the shape of the infestation is small. Therefore, target detection methods are chosen for identification. Classical single-stage target detection algorithms such as SSD (Liu et al., 2016), YOLOV3 (Redmon and Farhadi, 2018), YOLOV4 (Bochkovskiy et al., 2020), YOLOV5 (Jocher et al., 2021), RetinaNet (Lin et al., 2017) can obtain the positional information of the target object for identification and localization. In this study, YOLOV5 was selected for the detection of apple leaf diseases. As shown in Figure 1, the YOLOV5 model consists of four parts: Input, Backbone, Neck, and Head. The main work of each part is as follows:

1. Input. The Input part of YOLOV5s is preprocessed by adding mosaic data enhancement, adaptive anchor frames, and adaptive image scaling. The model can extract the features better during training and thus shows better results on the dataset.
2. Backbone. The Backbone part mainly relies on the Focus, C3, and SPP modules to extract features from the input images. The Focus module performs slicing operations on the image before it enters the backbone, thus reducing the feature dimensionality. The C3 module, which consists of three convolutional modules and a bottleneck structure, brings the dual advantages of increased computational speed and reduced parameter complexity. The SPP module is a pooling module that passes the input features in parallel to the Maxpool pooling layer to obtain a set of feature maps of different sizes, and finally joins these feature maps together so that feature information at different scales can be captured. The backbone is responsible for passing the extracted position and category information to the Neck layer.
3. Neck. The Neck part of the YOLOV5s combines up-sampling and down-sampling to generate a feature pyramid that improves the detection accuracy of the target object, which on one hand needs to reprocess the extracted features in the backbone network and on the other hand plays an important role in the subsequent detection.
4. Head. The Head part is to classify and predict the results of the neck layer by using a 1 ×1 convolutional layer to

TABLE 1 Apple leaf disease dataset.

| Diseases | Training Set | Validation Set | Test Set | Total |
|---|---|---|---|---|
| Alternaria blotch | 1040 | 130 | 130 | 1300 |
| Grey spot | 1040 | 130 | 130 | 1300 |
| Rust | 1040 | 130 | 130 | 1300 |
| Total | 3120 | 390 | 390 | 3900 |

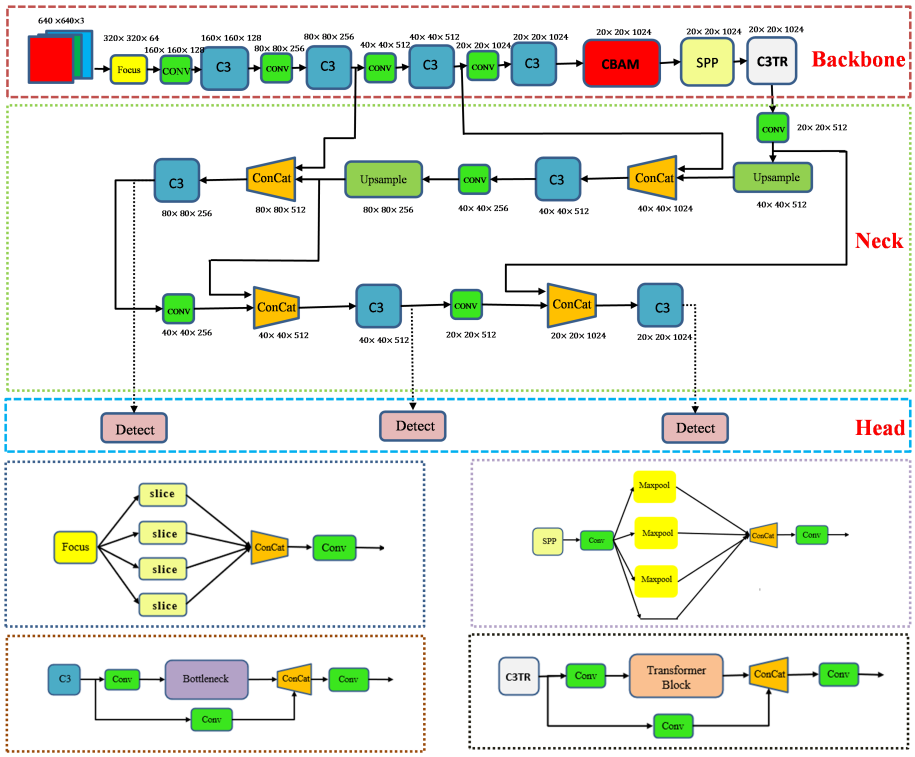



FIGURE 1
Overall structure of YOLOV5-CBAM-C3TR.

generate batch size different three results for final target detection.

## 2.2.2 CBAM module

When detecting apple leaf diseases, intricate background environments can cause interference, which can affect the accuracy of disease recognition. To address this challenge, integrating the attention mechanism becomes a promising solution that enhances the model's ability to selectively focus on relevant features while filtering out irrelevant information. As shown in Figure 2, the convolutional block attention mechanism (CBAM) (Woo et al., 2018) consists of two key components: the channel attention module (CAM) and the spatial attention module (SAM). The CAM emphasizes the key features, while the SAM emphasizes the spatial localization of these key features. The operation of the CAM consists of extracting features through average pooling and maximum pooling respectively. These features are then processed separately through a MLP network, and finally summed and output the feature vector. The mathematical formulation of the Channel Attention Module was shown in Equation 1.

$$M_c(F) = \sigma(\mathrm{MLP}(\mathrm{AvgPool}(F)) + \mathrm{MLP}(\mathrm{MaxPool}(F))) \quad (1)$$

where σ is a nonlinear sigmoid function used to map inputs to continuous outputs between 0 and 1. F is the input feature map, the MLP consists of two linear layers and a ReLU activation function.

SAM generates the spatial attention map by splicing the features that are average pooled and maximum pooled in the channel dimension. The formulation of the Spatial Attention Module was

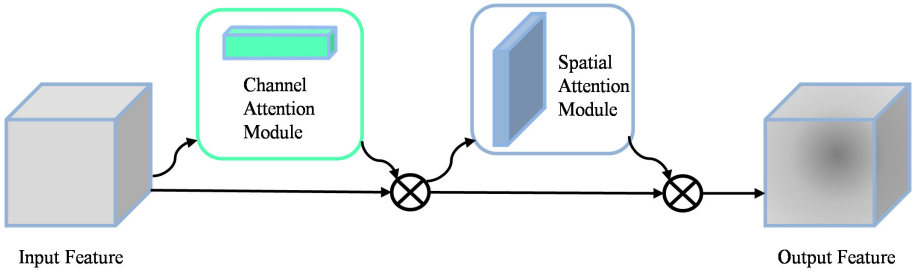



FIGURE 2
Specific structure of the CBAM module.

shown in Equation 2.

$$Ms(F) = \sigma\left(f^{7\times7}(\text{Concat}(AvgPool(F), Maxpool(F)))\right) \quad (2)$$

Where σ denotes the sigmoid function, $f^{7\times7}$ represents a convolutional kernel size of 7 × 7, and Concat denotes the connection operation.

### 2.2.3 C3TR module

In recent years, the transformer (Vaswani et al., 2017) architecture has been widely used in the field of natural language processing (NLP) with resounding success. As with NLP, where large amounts of textual data are key to training, the field of computer vision also relies on large image libraries for effective model learning. The transformer module can acquire complex relationships between different locations in the image. The multiple attention mechanism in the transformer module helps to extract multi-scale information, which can focus on both location and feature information and has great research potential. In order to realize the application of transformer in the field of computer vision, researchers endeavor to replace certain convolutional structures with transformer. For example, in target detection involving a drone capture scene, Zhu et al. (2021) innovatively integrated the transformer block into the C3 module of the YOLOV5 architecture, resulting in the C3TR module. As shown in Figure 1, richer image information extraction is achieved by replacing the bottleneck module in the C3 module.

The transformer block serves as the fundamental constituent within the C3TR framework, adopting the classical transformer encoder architecture. Illustrated in Figure 3, this block is comprised of three primary layers: Flatten, Multi-head attention, and feedforward neural network (FFN).

1. Flatten

The Flatten operation is to flatten the two-dimensional feature vectors obtained by the model based on the image into one-dimensional vectors, which can preserve the positional information of the image. If an input feature map $X \in R^{H\times W\times C}$ is given, it will become $X_1 \in R^{H\times C}$ after the spreading operation, where H×W=H.

2. Multi-head attention

The multiple attention operation is responsible for different linear mappings through the Flatten and LayerNorm, allowing simultaneous attention to feature information at different scales. After converting the feature maps into $Q, K, V \in R^{N\times C}$ as inputs for multi-head attention, each single head performs one feature mapping for Q, k, V. The output formula after the completion of the single-head attention operation was shown in Equation 3 and Equation 4.

$$Output_i = S_i V_i \quad (3)$$

$$S_i = \text{softmax}(Q_i K_i^T) \quad (4)$$

where $Q_i, K_i, V_i$ denote the multiplication of Q, K, V with the weight matrix of the single-head attention mechanism, $S_i$ represents the single-head attention matrix, and $Output_i$ refers to the integration of global information. The $Output_i$ generated after feature mapping for each single-head attention will eventually be unified through the connectivity layer to produce the final output with expression was shown in Equation 5.

$$Output = \text{Conca t}(Output_1, Output_2, \ldots, Output_n) \quad (5)$$

where n represents the number of multi-head attention.

3. FFN

The FFN layer is a feed forward neural network, which is composed of two fully connected layers, one of which contains the Relu activation function and the Dropout function between the two layers. The expression of FFN was shown in Equation 6.

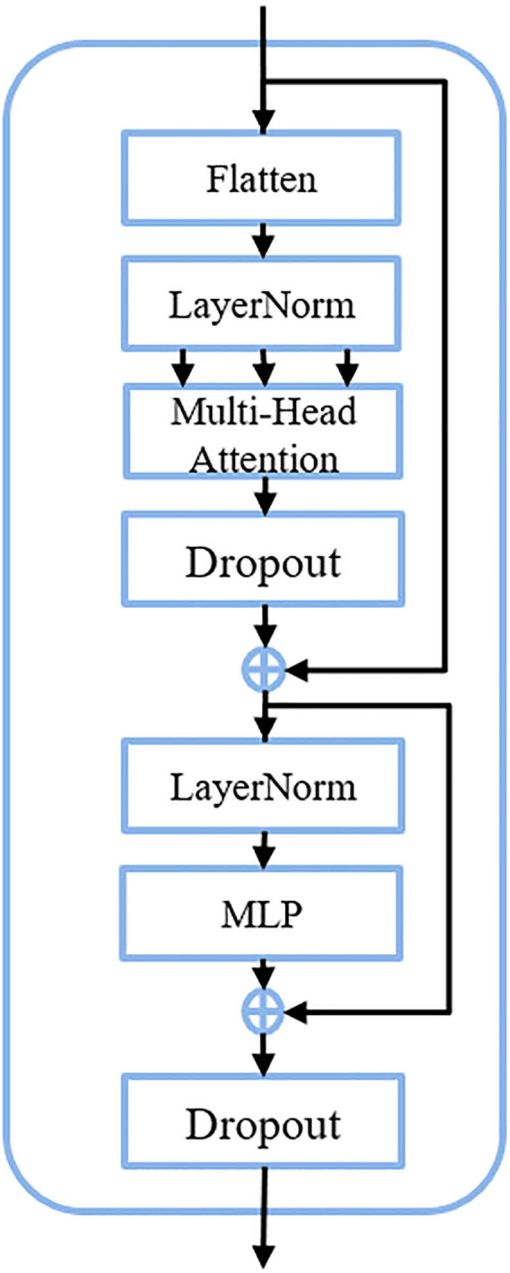



FIGURE 3
Detailed architecture of the transformer block.

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \quad (6)$$

where $x$ represents the feature sequence of the input FFN layer, $W_1$ and $b_1$ represent the weights and offsets of the first fully connected layer, and $W_2$ and $b_2$ represent the weights and offsets of the second fully connected layer, respectively.

#### 2.2.4 Proposed model

Orchard environments are extremely complex. Common problems in target object detection such as similar texture between target object and background, target occlusion, and similarity between target object types. The focus on improving detection accuracy led us to optimize the YOLOV5 framework. This was done by trying to add CBAM, SE (Hu et al., 2018), ECA (Wang et al., 2020), and C3TR modules to improve the performance of the model. Finally, by adding CBAM module before SPP module and C3TR module at the last layer of backbone network, the optimized YOLOV5-CBAM-C3TR model was proposed. Figure 1 shows the overall structure of the optimized model YOLOV5-CBAM-C3TR. Before starting the training, the optimal runtime environment was created, the input images were resized to 640 × 640. After training, the final three different dimensions of the detection header effectively outputted important information related to the type and location of the apple leaf disease.

### 2.3 Model training environment parameter configuration

In this study, the model training environment was built using Pytorch and GPUs with the parameters shown in Table 2. The adaptive moment estimation (Adam) (Kingma and Ba, 2014) was used as the optimizer in the experiments. The input image input size was set to 640×640, obtained by filling the original image. After repeated experiments, the final hyperparameters were set as follows: the initial learning rate was set to 0.0005, the epoch number was set to 100, and the batch size was 8. To ensure the fairness of model comparison, the parameters used in this study were consistent.

### 2.4 Model evaluation

In order to comprehensively assess the performance of the model in apple leaf disease detection, a set of evaluation metrics including precision, recall, mAP@0.5, mAP@[0.5:0.95], F1 Score, and parameters were chosen. Among them, mAP is the mean average precision, which is the evaluation metric of the main model in target detection. mAP 0.5and mAP@[0.5:0.95] are distinguished by the difference in the size of the intersection over union (IOU), which determines that mAP@[0.5:0.95] is more stringent. In addition, the loss value is used to assess the error between the predicted and the ground truth. The training loss reflects the model's ability to fit on that dataset, and the validation loss reflects the model's ability to generalize. The loss value contains three parameters: obj_loss (object loss), cls_loss (classification loss), box_loss (bounding loss). The above metrics were calculated in Equation 7, Equation 8, Equation 9, Equation 10, and Equation 11.

TABLE 2 Software and hardware environment resource configuration.

| Configuration | Parameter |
|---|---|
| Operating system | Ubuntu 20.04 |
| GPU | NVIDIA GeForce RTX 3090 |
| CPU | Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz |
| Memory | 100 GB |
| Language | Python 3.8 |
| Framework | Pytorch 1.10.0 |

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (9)$$

$$mAP = \frac{\sum_{i=1}^{C} AP_i}{C} \quad (10)$$

$$loss = box_loss + obj_loss + cls_loss \quad (11)$$

where $TP$, $FP$ , $FN$ , and $TN$ stand for true positive, false positive, false negative, and true negative, respectively. The $AP_i$ is the average precision value at the i-th species. $C$ is the total number of species.

## 3 Results

### 3.1 Model optimization

To further improve the detection precision of YOLOV5 for apple leaf diseases, different modules including SE, CBAM, ECA, and C3TR were added to improve the detection capability of YOLOV5. As can be seen from Table 3, compared with the original YOLOV5 model, the improved YOLOV5-CBAM-C3TR achieved 73.4%, 40.9%, 70.9%, and 69.5% of mAP@0.5, mAP@[0.5:0.95], precision and recall, which was a significant improvement in detection performance. In addition, the experimental results also showed that the improved YOLOV5-CBAM-C3TR is more suitable for the detection of apple leaf diseases in real and complex environments.

#### 3.1.1 Model performance optimization by adding an individual module

As can be seen from Table 4, the low accuracy of YOLOV5 in detecting apple leaf diseases may be caused by the fact that the two diseases including Alternaria blotch and Grey spot, which were too similar. Therefore, to make the model more focused on extracting the characteristics of apple leaf disease, different attention

TABLE 3 Comparison with different target detection models.

| Methods | mAP@0.5 (%) | mAP@ [0.5:0.95] (%) | Precision (%) | Recall (%) |
|---|---|---|---|---|
| SSD | 66.2 | **46.9** | **96.8** | 36.9 |
| YOLOV3 | 65.1 | 33.3 | 64.3 | 63.4 |
| YOLOV4 | 52.4 | 18.6 | 87.9 | 25.1 |
| YOLOV5 | 67.8 | 34.9 | 67.3 | 66.1 |
| MGA-YOLOV5 | 69.0 | 34.2 | 74.4 | 64.3 |
| BTC-YOLOV5 | 72.0 | 39.8 | 70.7 | 67.9 |
| **YOLOV5-CBAM-C3TR** | **73.4** | 40.9 | 70.9 | **69.5** |

The performance of the indicators is best shown in bold.

mechanisms were tried to be added to the backbone network of YOLOV5 for experiments. As shown in Table 4, the addition of SE, ECA, CBAM and C3TR all improved the accuracy of the YOLOV5 model for detecting apple leaf diseases while keeping the number of model parameters relatively constant. The SE module allows the model to better focus on feature channels that are effective for apple leaf disease identification. Compared to YOLOV5, the addition of the SE model resulted in an improvement of 3.39%, 5.73%, -4.75%, -5.3%, and -5.1% in mAP@0.5, mAP@[0.5:0.95], precision, recall, and F1 score, respectively. The ECA module calculates the correlation of the feature channels so that the model focuses more on the desired feature channels. Compared to YOLOV5, the addition of the ECA model resulted in an improvement of 3.69%, 6.88%, 5.65%, -2.12%, and 1.5% in mAP@0.5, mAP@[0.5:0.95], precision, recall, and F1 score, respectively. Unlike the SE and ECA modules, The CBAM module extracts features by focusing on the channel and spatial information of the image. Compared to YOLOV5, the addition of the CBAM model resulted in an improvement of 7.08%, 12.03%, -7.43%, -7.72%, and -7.65% in mAP@0.5, mAP@[0.5:0.95], precision, recall, and F1 score, respectively. The transformer module in the C3TR module captures global contextual information, which improved the mAP@0.5, mAP@[0.5:0.95], precision, recall, and F1 score of the C3TR module by 6.34%, 15.2%, 8.9%, 2.7%, and 5.7% over the YOLOV5 model, respectively. Adding modules can improve the accuracy of the model in detecting target objects, but it also increases the number of parameters of the model, which is not conducive to the lightweight deployment of the model. By observing the change in the number of model parameters when each module acts alone. It was found that the addition of the SE module severely increased the number of model parameters, while the addition of the other four modules had little effect on the number of model parameters.

### 3.1.2 Model performance optimization by adding multiple modules

As can be seen from Table 5, adding the modules individually all improved the detection accuracy of the YOLOV5 model. In order to further improve the feature extraction ability of the model, the attention mechanism was combined with the C3TR module. The combination experiments of CBAM+C3TR, ECA+C3TR, and SE+C3TR were conducted respectively. Table 3 shows that combining two modules improved the detection accuracy of the YOLOV5 model better than adding a single module. Compared with the addition of SE and C3TR alone, the mAP@0.5, mAP@[0.5:0.95], precision, recall, and F1 score of the YOLOV5-SE -C3TR were improved by 3.7% and 0.8%, 10.8% and 1.7%, 6.2% and -7.1%, and 8.6% and 0.1%, 7.4% and -3.5%, respectively. Compared with the addition of the ECA module and C3TR module alone, the mAP@0.5, mAP@[0.5:0.95], precision, recall, and F1 score of YOLOV5-ECA -C3TR were improved by 3.0% and 0.4%, 7.8%, and 0.0%, -1.1% and -4.1%, 5.4% and 0.4%, and 2.2% and -1.8%, respectively. Compared with the addition of the CBAM module and C3TR module alone, the mAP@0.5, mAP@[0.5:0.95], precision, recall, and F1 score of the YOLOV5-CBAM-C3TR were improved by 1.1% and 1.8%, 4.6% and 1.7%, 13.8% and -3.2%, 13.9%, and

TABLE 4 Comparison of model performance improvement by adding a single module.

| Methods | mAP@0.5 (%) | mAP@ [0.5:0.95] (%) | Precision (%) | Recall (%) | F1 Score | Parameters |
|---|---|---|---|---|---|---|
| YOLOV5 | 67.8 | 34.9 | 67.3 | 66.1 | 66.7 | **7018216** |
| YOLOV5-SE | 70.1 | 36.9 | 64.1 | 62.6 | 63.3 | 7542504 |
| YOLOV5-ECA | 70.3 | 37.3 | 71.1 | 64.7 | 67.7 | 7018217 |
| YOLOV5-CBAM | **72.6** | 39.1 | 62.3 | 61.0 | 61.6 | 7051627 |
| YOLOV5-C3TR | 72.1 | **40.2** | **73.3** | **67.9** | **70.5** | 7060072 |

The performance of the indicators is best shown in bold.

TABLE 5 Comparison of model performance improvement by adding combinations of models.

| Methods | mAP@0.5 (%) | mAP@ [0.5:0.95] (%) | Precision (%) | Recall (%) | F1 Score | Parameters |
|---|---|---|---|---|---|---|
| YOLOV5-SE -C3TR | 72.7 | **40.9** | 68.1 | 68.0 | 68.0 | 7092840 |
| YOLOV5-ECA -C3TR | 72.4 | 40.2 | 70.3 | 68.2 | 69.2 | **7060073** |
| **YOLOV5-CBAM-C3TR** | **73.4** | **40.9** | **70.9** | **69.5** | **70.2** | 7093483 |

The performance of the indicators is best shown in bold.

2.4%, and 14% and -0.4%, respectively. Overall, combining SE, ECA, and CBAM with C3TR all further improved mAP@0.5, mAP@[0.5:0.95], and recall with little parameter change compared to adding each module individually. Although the addition of multiple modules resulted in a decrease in accuracy and F1 score metrics compared to the addition of the C3TR module alone, YOLOV5-CBAM-C3TR had the smallest decrease and the largest increase, achieving almost positive growth and being the best performing model. The added SE or ECA modules need to capture channel information, while the transformer module in C3TR needs to capture context information. The reason for the accuracy degradation may be the mutual interference between multiple modules leading to inadequate feature extraction. On the other hand, the CBAM module, which focuses on both channel and spatial dimension information, interoperates with the C3TR module to better ensure that sufficient feature information is provided to the model.

## 3.2 Model training

Seven target detection models including YOLOV3, YOLOV4, YOLOV5, SSD, MGA-YOLOV5, BTC-YOLOV5, and optimized YOLOV5-CBAM-C3TR were established based on labeled apple leaf disease images. Figure 4A shows the graph of training loss values for each model with increasing epoch values in apple leaf disease detection. In general, the loss functions of each model decreased with increasing epochs and eventually stabilized. The SSD model had the fastest convergence of the training loss curve, but also had the largest loss value after stabilization, which reached full convergence after 10 epochs. The loss functions of the other six models gradually stabilized after 60 epochs of training. Among them, MGA-YOLOV5 had the second-highest training loss value after the training loss function gradually stabilized. The loss functions of YOLOV5-CBAM-C3TR, BTC-YOLOV5, YOLOV3, and YOLOV4 were very similar, with slightly higher stabilized loss values than those of YOLOV5. YOLOV5 has the lowest training loss value of all the models.

Figure 4B shows a plot of the validation loss function with increasing epoch values. As with the training loss function curve, the SSD model still had the fastest convergence rate and stabilized after 10 epochs, while the validation loss value was the highest. The other six target detection models all stabilized around the 30th epoch. Specifically, YOLOV5 and MGA-YOLOV5 had the similar loss function curves after stabilization, with the second highest loss function value. The YOLOV3, YOLOV4, BTC-YOLOV5, and YOLOV5-CBAM-C3TR had also the similar loss curves after stabilization, with slightly higher stabilized loss values than that of YOLOV5, which had the lowest training loss value among all models.

## 3.3 Comparative analysis with different detection models

The objective of this study is to propose a target detection model capable of accurately identifying and locating apple leaf diseases, which can assist the disease precision spraying device for automatic spraying. To verify the effectiveness of YOLOV5-CBAM-C3TR in detecting apple leaf diseases, it was compared with SSD, YOLOV3, YOLOV4, YOLOV5, MGA-YOLOV5 and BTC-YOLOV5 models on the same dataset. The results in Table 3 showed that the mAP 0.5and mAP@[0.5:0.95] of YOLOV4 were the lowest with 52.4% and 18.6% respectively. While the mAP 0.5and mAP@[0.5:0.95] of YOLOV5-CBAM-C3TR were the highest with 73.4% and 40.9% respectively. The precision of SSD was up to 96.8% and the recall was only 36.9%, indicating that SSD was accurate in detecting apple leaf diseases, but there were omissions in disease identification. Similarly, YOLOV4 had large variations in precision and recall, resulting in a poor mAP 0.5 In contrast, YOLOV3, YOLOV5, MGA-YOLOV5, and BTC-YOLOV5 can balance the precision and recall metrics better, with mAP 0.5of 65.1%, 67.8%, 69%, and 72%, respectively. Overall, compared with YOLOV5, the optimized YOLOV5-CBAM-C3TR showed a significant improvement in detection precision, with an 8.25% improvement in mAP@0.5 and a 17.2% improvement in mAP@[0.5:0.95]. In addition, the experimental results also confirms that the optimized YOLOV5-CBAM-C3TR has a high detection accuracy, which is sufficient for practical needs.

## 3.4 Performance of the improved model in apple leaf disease detection

To further validate the effectiveness of the improved model, the original YOLOV5 model and the optimized YOLOV5-CBAM-C3TR model were selected for the comparison of detection results in real environments. A total of 208 sample images with natural environment backgrounds were selected in the test set to examine the detection effect of YOLOV5-CBAM-C3TR in real scenes. Table 6 shows that YOLOV5-CBAM-C3TR improves the correct recognition rate of the three apple leaf diseases compared to

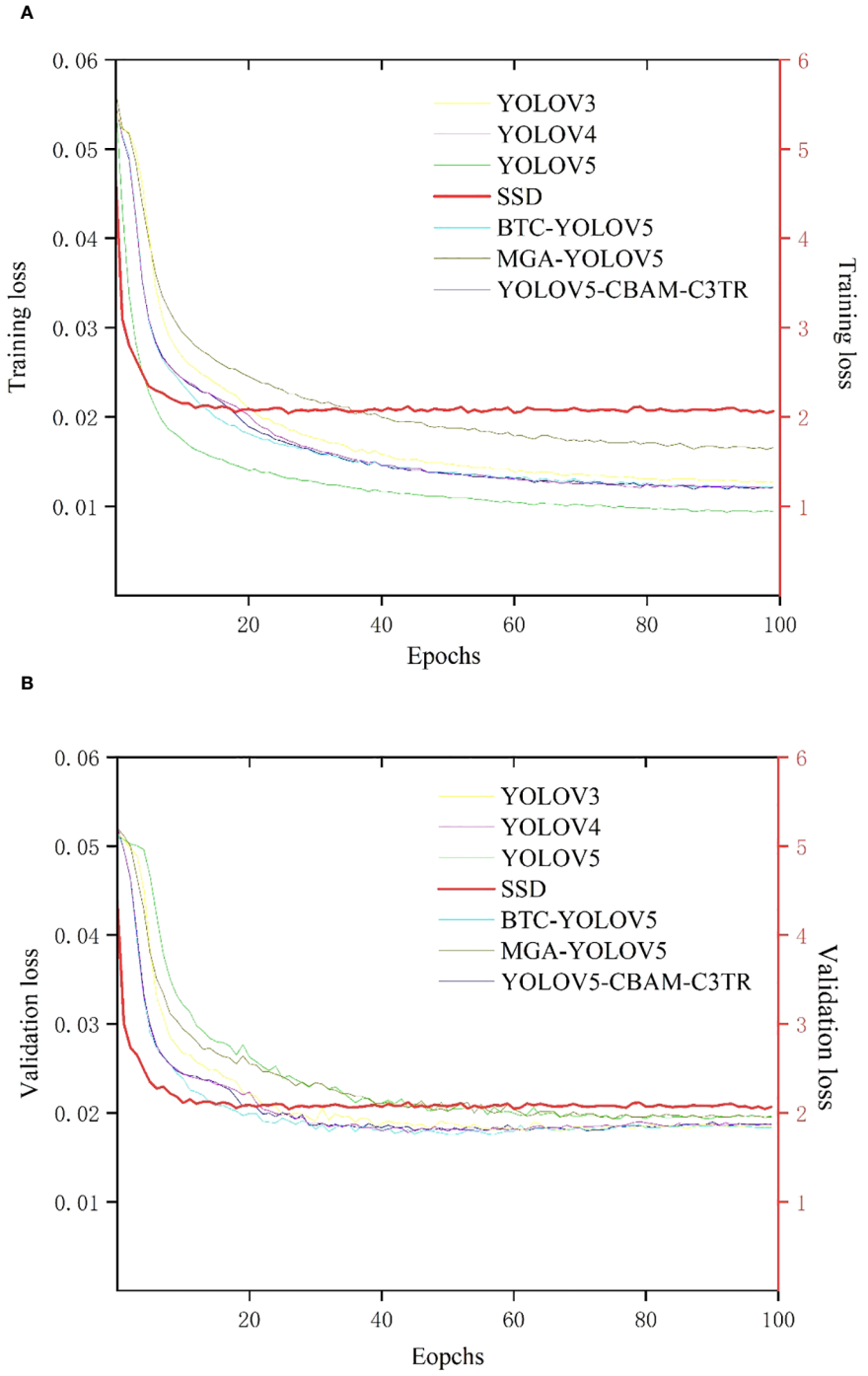



FIGURE 4
**(A)** Curve of training loss values with epoch values, **(B)** Curve of validation loss values with epoch values. Due to the different scales of change in loss function values for SSD and other models, a double Y-axis is used to represent the change in loss function for each model. The left axis represents the scale of variation of the loss function values for the YOLOV3, YOLOV4, YOLOV5, BTC-YOLOV5, MGA-YOLOV5, and YOLOV5-CBAM-C3TR models, and the right axis represents the scale of variation of the loss function values for the SSD model. the loss function curves for the SSD model are shown in bold red.

YOLOV5, with a significant increase of 18.9% in the average accuracy. Figure 5 shows a comparison of typical detection results for the three apple leaf diseases. The results show that the YOLOV5 algorithm has errors in detecting the three apple leaf diseases in a natural environment with a complex background, and the main reason for the unsatisfactory detection results is its inaccurate feature extraction of the diseases. As can be seen in Figure 6, YOLOV5-CBAM-C3TR is able to extract the features of various diseases better, but there is still a risk of misjudging Alternaria blotch and Grey spot, which are two similar diseases. The possible reason for this is that these two diseases are very similar after data enhancement in the simulated natural environment. In this study,

TABLE 6 Test results of improved models in detecting apple leaf diseases.

| Model | Number of Alternaria Blotch Samples | Number of Correctly Detected | Number of Grey Spot Samples | Number of Correctly Detected | Number of Rust Samples | Number of Correctly Detected | Average Accuracy (%) |
|---|---|---|---|---|---|---|---|
| YOLOV5 | 87 | 52 | 67 | 54 | 54 | 50 | 77.7 |
| YOLOV5-CBAM-C3TR | 87 | 81 | 67 | 60 | 54 | 51 | 92.4 |

the CBAM module and the C3TR module were added to YOLOV5, and the two modules work together to enable YOLOV5 to better extract disease features.

## 4 Discussion

In this study, attention mechanism and module with the transformer encoder were added to optimize YOLOV5, and finally proposed YOLOV5-CBAM-C3TR to accurately classify three common diseases of apple leaves. Comparing with the target detection algorithms such as SSD, YOLOv3, YOLOv4, and YOLOv5, YOLOV5-CBAM-C3TR had the highest mAP@0.5 and mAP@[0.5:0.95], which reached 73.4% and 40.9%, respectively. An average accuracy of 92.4% was achieved on a randomly selected sample of 208 images containing the three apple leaf diseases. Empirical results showed that adding CBAM, SE, ECA, and C3TR individually or in combination can significantly improve the detection accuracy of YOLOV5. In contrast, combining each attention mechanism with the C3TR module has a higher




FIGURE 5
**(A)** Original image, **(B)** YOLOV5 detection results, **(C)** YOLOV5-CBAM-C3TR detection results. Different colored bounding boxes are used in the images to distinguish the types of apple leaf diseases, with Alternaria blotch in red, Grey spot in pink and Rust in orange. The names of apple leaf diseases from the first to the third column in the image are: Alternaria blotch, Grey spot, and Rust.

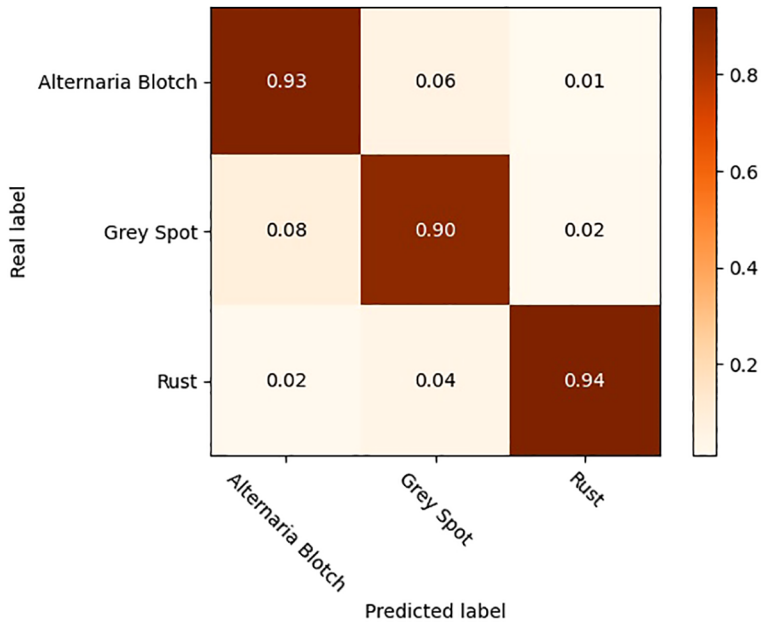



FIGURE 6
Confusion matrix for the detection of three apple leaf diseases.

detection accuracy than adding each module separately. Among them, the combination of CBAM and C3TR provided the most significant performance enhancement for YOLOv5. Different from SE or ECA modules, CBAM module pays attention to both channel information and spatial information, and can better cooperate with C3TR for global information extraction. Certainly, Attention mechanisms have been shown to be effective in many tasks (Xue et al., 2021; Wang et al., 2022; Zhao et al., 2022). However, the task requirements in different scenarios should be carefully considered when choosing the appropriate attention module, which suggests that the selection of modules requires extensive experimentation. Although adding modules can improve the detection accuracy of target objects, it also increases the number of parameters of the model, which is not conducive to the actual deployment of the model. The addition of CBAM, ECA, SE, and C3TR in this study increased the number of parameters in YOLOV5. Future research will consider methods to reduce the number of parameters while maintaining model detection accuracy, such as pruning (Liang et al., 2022) and distillation, in order to achieve a good balance between model detection accuracy and the number of parameters.

Timely detection and control of apple leaf diseases is extremely important. Since different apple leaf diseases may have similar characteristics, even the human eye cannot distinguish them accurately after exposure and other treatments that simulate the natural environment. The experimental data in Table 6 showed that YOLOV5-CBAM-C3TR improved the two types of diseases including Alternaria blotch and Grey spot, by 33.33% and 8.95%, respectively. The average accuracy achieved 92.1% for the three types of diseases. The experimental data affirmed the ability of the optimized model to accurately identify similar diseases. However, factors such as the number of disease types in the data set, the complexity of the environment of the objects to be detected in the image, and the difference in categories of the objects to be detected will affect the accuracy of the network model detection to some extent. Therefore, the selection of detection accuracy to evaluate the model performance should be combined with specific application scenarios. For example, Khan et al. (2022) developed an automated apple leaf disease detection system based on deep learning. The experimental results show that on a dataset containing more than 9000 images, Faster-RCNN can reach 42.01% mAP at 6FPS, showing a good detection accuracy for 9 common apple leaf diseases. The model is tested on data set images that are less disturbed by the real background environment, and its robustness is low in the real environment. The experimental results also indicate that the model is not effective in detecting diseased leaves. When using a target detection model to detect apple leaf diseases, Zhang et al. (2023) found that the MFaster R-CNN model could achieve 97.23% mAP for eight kinds of corn leaf diseases, but only 80.69% mAP on a self-built data set of apple leaf diseases. The above examples show that the accuracy evaluation of the model should be combined with the specific application scenarios of the model. In different tasks and different data sets, the model will show different detection performance. Only 3 kinds of apple leaf diseases were considered in this experiment, while there are more than 200 kinds of apple leaf diseases. Therefore, although YOLOV5-CBAM-C3TR can accurately identify apple leaf diseases similar to those in the classification data set, it may not be universally applicable to other similar diseases. It is necessary to expand the data set of apple leaf disease and collect more comprehensive types of leaf disease for research. In addition, the model proposed in this study needs to be compared with more advanced object detection algorithms such as YOLOV7 and YOLOV8. These questions will be further explored in

the future to improve the accuracy of the model's detection of different apple leaf diseases so that each class of similar diseases can be accurately classified.

## 5 Conclusions

YOLOV5-CBAM-C3TR algorithm was proposed to improve the accuracy of detection of three apple leaf diseases including Alternaria blotch, Grey spot, and Rust. The model was obtained by optimizing YOLOV5 with the addition of an attention mechanism and a module with a transformer encoder. Compared with different target detection models, the optimized YOLOV5-CBAM-C3TR algorithm achieved the highest detection accuracy than other models, with mAP@0.5, mAP@[0.5:0.95], precision, recall of 73.4%, 40.9%, 70.9%, 69.5%, respectively. In randomly selected apple leaf disease samples, the average accuracy based on the YOLOV5-CBAM-C3TR model can reach 92.4%, which was 18.9% higher than that of the original YOLOV5. Moreover, the YOLOV5-CBAM-C3TR model also showed a strong ability to identify similar diseases, and could accurately identify Alternaria blotch and grey spot, which are almost indistinguishable from the naked eye. In the future, YOLOV5-CBAM-C3TR can also be extended to detect similar diseases in other crops.

## Data availability statement

The original contributions presented in the study are included in the article/supplementary material. Further inquiries can be directed to the corresponding author.

## Author contributions

ML: Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft. W-HS: Conceptualization, Funding acquisition, Methodology, Project administration, Resources, Supervision, Writing – review & editing.

## Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This study was supported by the National Natural Science Foundation of China (Grant No. 32101610; 32371991).

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

Abade, A., Ferreira, P. A., and de Barros Vidal, F. (2021). Plant diseases recognition on images using convolutional neural networks: A systematic review. *Comput. Electron. Agric.* 185, 106125. doi: 10.1016/j.compag.2021.106125

Babu, V. S., and Ram, N. V. (2022). Deep residual CNN with contrast limited adaptive histogram equalization for weed detection in soybean crops. *Traitement du Signal* 39, 717. doi: 10.18280/ts.390236

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv* 2004, 10934. doi: 10.48550/arXiv.2004.10934

Chen, Z., Wu, R., Lin, Y., Li, C., Chen, S., Yuan, Z., et al. (2022). Plant disease recognition model based on improved YOLOv5. *Agronomy* 12, 365. doi: 10.3390/agronomy12020365

Chuanlei, Z., Shanwen, Z., Jucheng, Y., Yancui, S., and Jia, C. (2017). Apple leaf disease identification using genetic algorithm and correlation based feature selection method. *Int. J. Agric. Biol. Eng.* 10, 74–83. doi: 10.25165/ijabe.v10i2.2166

Dhaka, V. S., Meena, S. V., Rani, G., Sinwar, D., Kavita,, Ijaz, M. F., et al. (2021). A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. *Sensors* 21, 4749. doi: 10.3390/s21144749

Dubey, S. R., and Jalal, A. S. (2012). "Detection and classification of apple fruit diseases using complete local binary patterns, in: 2012 third international conference on computer and communication technology," in *Presented at the 2012 3rd International Conference on Computer and Communication Technology (ICCCT 2012)*, (Allahabad, India: IEEE). 346–351.

Gargade, A., and Khandekar, S. (2021). "Custard apple leaf parameter analysis, leaf diseases, and nutritional deficiencies detection using machine learning," in *Advances in Signal and Data Processing, Lecture Notes in Electrical Engineering*, vol. pp . Eds. S. N. Merchant, K. Warhade and D. Adhikari (Singapore: Springer Singapore), 57–74.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., and Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Syst. their Appl.* 13, 18–28. doi: 10.1109/5254.708428

Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. *Proc. IEEE Conf. Comput. Vision Pattern Recognition* 1709, 7132–7141. doi: 10.1109/CVPR.2018.00745

Jan, M., and Ahmad, H. (2020). Image features based intelligent apple disease prediction system: machine learning based apple disease prediction system. *IJAEIS* 11, 31–47. doi: 10.4018/IJAEIS.2020070103

Jocher, G., Stoken, A., Borovec, J., Chaurasia, A., Changyu, L., Hogan, A., et al. (2021). ultralytics/yolov5: v5. 0-YOLOv5-P6 1280 models, AWS, Supervise. ly and YouTube integrations. *Zenodo*.

Kaiming, H., Georgia, G., Piotr, D., and Ross, G. (2017). "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (Hawaii, US: Computer Vision Foundation), 2961–2969.

Khan, A. I., Quadri, S. M. K., Banday, S., and Latief Shah, J. (2022). Deep diagnosis: A real-time apple leaf disease detection system based on deep learning. *Comput. Electron. Agric.* 198, 107093. doi: 10.1016/j.compag.2022.107093

Kingma, D. P., and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv* 1412, 6980. doi: 10.48550/arXiv.1412.6980

Kundu, N., Rani, G., Dhaka, V. S., Gupta, K., Nayak, S. C., Verma, S., et al. (2021). IoT and interpretable machine learning based framework for disease prediction in pearl millet. *Sensors* 21, 5386. doi: 10.3390/s21165386

Lang, X., Ren, Z., Wan, D., Zhang, Y., and Shu, S. (2022). MR-YOLO: an improved YOLOv5 network for detecting magnetic ring surface defects. *Sensors* 22, 9897. doi: 10.3390/s22249897

Li, J. (2022). An improved YOLOv5-based vegetable disease detection method. *Comput. Electron. Agric* 202. doi: 10.1016/j.compag.2022.107345

Li, T., Sun, M., He, Q., Zhang, G., Shi, G., Ding, X., et al. (2023). Tomato recognition and location algorithm based on improved YOLOv5. *Comput. Electron. Agric.* 208, 107759. doi: 10.1016/j.compag.2023.107759

Liang, X., Jia, X., Huang, W., He, X., Li, L., Fan, S., et al. (2022). Real-time grading of defect apples using semantic segmentation combination with a pruned YOLO V4 network. *Foods* 11, 3150. doi: 10.3390/foods11193150

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision* (Hawaii, US: Computer Vision Foundation), 2980–2988.

Liu, S., Qiao, Y., Li, J., Zhang, H., Zhang, M., and Wang, M. (2022). An improved lightweight network for real-time detection of apple leaf diseases in natural scenes. *Agronomy* 12, 2363. doi: 10.3390/agronomy12102363

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. (2016). Ssd: Single shot multibox detector, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. *Springer* 9905, 21–37. doi: 10.1007/978-3-319-46448-0_2

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: unified, real-time object detection, in: 2016 IEEE conference on computer vision and pattern recognition (CVPR)," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Computer Vision Foundation), Vol. pp. 779–788(Las Vegas, NV, USA).

Redmon, J., and Farhadi, A. (2018). *YOLOv3: an incremental improvement* (arxiv:1804.02767). doi: 10.48550/arXiv.1804.02767

Ren, S., He, K., Girshick, R., and Sun, J. (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems* (NIPS 2015): Curran Associates, Inc, 28-2015.

Shi, Y., Huang, W., and Zhang, S. (2017). Apple disease recognition based on two-dimensionality subspace learning. *Comput. Eng. Appl.* 53 (22), 180–184.).

Shu, C., Zhao, H., Jiao, W., Liu, B., Cao, J., and Jiang, W. (2019). Antifungal efficacy of ursolic acid in control of Alternaria alternata causing black spot rot on apple fruit and possible mechanisms involved. *Scientia Hortic.* 256, 108636. doi: 10.1016/j.scienta.2019.108636

Singh, S., Gupta, I., Gupta, S., Koundal, D., Mahajan, S., and Pandit, A. (2021). Deep learning based automated detection of diseases from apple leaf images. *Computers Materials Continua* 71, 1849–1866. doi: 10.32604/cmc.2022.021875

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). "Attention is all you Need," in *Advances in Neural Information Processing Systems* (Curran Associates, Inc).

Vishnu, S., and Ranjith, R. A. (2015). Plant disease detection using leaf pattern: A review. *Int. J. Innovative Science Eng. Technol.* 2, 774–780.

Wang, S., He, D., Li, W., and Wang, Y. (2009). Plant leaf disease recognition based on kernel K-means clustering algorithm. *Nongye Jixie Xuebao= Trans. Chin. Soc. Agric. Machinery* 40, 152–155.

Wang, H., Li, Y., Dang, L. M., and Moon, H. (2022). An efficient attention module for instance segmentation network in pest monitoring. *Comput. Electron. Agric.* 195, 106853. doi: 10.1016/j.compag.2022.106853

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., and Hu, Q. (2020). "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, USA: Computer Vision Foundation). 11534–11542.

Wang, Y., and Zhao, J. (2022). MGA-YOLO: A lightweight one-stage network for apple leaf disease detection. *Front. Plant Sci.* (Technical Advances in Plant Science), 13-022. doi: 10.3389/fpls.2022.927424

Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S. (2018). "CBAM: Convolutional Block Attention Module," in *Computer Vision – ECCV 2018, Lecture Notes in Computer Science*. Eds. V. Ferrari, M. Hebert, C. Sminchisescu and Y. Weiss (Cham: Springer International Publishing), 3–19.

Xue, M., Chen, M., Peng, D., Guo, Y., and Chen, H. (2021). One spatio-temporal sharpening attention mechanism for light-weight YOLO models based on sharpening spatial attention. *Sensors* 21, 7949. doi: 10.3390/s21237949

Yu, H.-J., and Son, C.-H. (2019). *Apple Leaf Disease Identification through Region-of-Interest-Aware Deep Convolutional Neural Network* (arxiv) 64 (2), pp. 20507-1-20507-10.

Zhang, Y., Zhou, G., Chen, A., He, M., Li, J., and Hu, Y. (2023). A precise apple leaf diseases detection using BCTNet under unconstrained environments. *Comput. Electron. Agric.* 212, 108132. doi: 10.1016/j.compag.2023.108132

Zhao, S., Liu, J., and Wu, S. (2022). Multiple disease detection method for greenhouse-cultivated strawberry based on multiscale feature fusion Faster R_CNN. *Comput. Electron. Agric.* 199, 107176. doi: 10.1016/j.compag.2022.107176

Zhong, Y., and Zhao, M. (2020). Research on deep learning in apple leaf disease recognition. *Comput. Electron. Agric.* 168, 105146. doi: 10.1016/j.compag.2019.105146

Zhu, X., Lyu, S., Wang, X., and Zhao, Q. (2021). "TPH-YOLOv5: improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios, in: 2021 IEEE/CVF international conference on computer vision workshops (ICCVW)," in *Presented at the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), IEEE*, Montreal, BC, Canada (Southern California, USA: Computer Vision Foundation). 2778–2788.

frontiers | Frontiers in Plant Science

TYPE Methods
PUBLISHED 25 January 2024
DOI 10.3389/fpls.2023.1273029

Check for updates

OPEN ACCESS

EDITED BY
Yalin Wu,
Lushan Botanical Garden (CAS), China

REVIEWED BY
Jana Shafi,
Prince Sattam Bin Abdulaziz University, Saudi Arabia
Parvathaneni Naga Srinivasu,
Prasad V. Potluri Siddhartha Institute of Technology, India

*CORRESPONDENCE
Shansong Wang
sswang@sdust.edu.cn

RECEIVED 05 August 2023
ACCEPTED 29 December 2023
PUBLISHED 25 January 2024

CITATION
Zeng Q, Sun J and Wang S (2024) DIC-Transformer: interpretation of plant disease classification results using image caption generation technology.
*Front. Plant Sci.* 14:1273029.
doi: 10.3389/fpls.2023.1273029

COPYRIGHT
© 2024 Zeng, Sun and Wang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# DIC-Transformer: interpretation of plant disease classification results using image caption generation technology

Qingtian Zeng, Jian Sun and Shansong Wang*

College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China

Disease image classification systems play a crucial role in identifying disease categories in the field of agricultural diseases. However, current plant disease image classification methods can only predict the disease category and do not offer explanations for the characteristics of the predicted disease images. Due to the current situation, this paper employed image description generation technology to produce distinct descriptions for different plant disease categories. A two-stage model called DIC-Transformer, which encompasses three tasks (detection, interpretation, and classification), was proposed. In the first stage, Faster R-CNN was utilized to detect the diseased area and generate the feature vector of the diseased image, with the Swin Transformer as the backbone. In the second stage, the model utilized the Transformer to generate image captions. It then generated the image feature vector, which is weighted by text features, to improve the performance of image classification in the subsequent classification decoder. Additionally, a dataset containing text and visualizations for agricultural diseases (ADCG-18) was compiled. The dataset contains images of 18 diseases and descriptive information about their characteristics. Then, using the ADCG-18, the DIC-Transformer was compared to 11 existing classical caption generation methods and 10 image classification models. The evaluation indicators for captions include Bleu1–4, CiderD, and Rouge. The values of BLEU-1, CIDEr-D, and ROUGE were 0.756, 450.51, and 0.721. The results of DIC-Transformer were 0.01, 29.55, and 0.014 higher than those of the highest-performing comparison model, Fc. The classification evaluation metrics include accuracy, recall, and F1 score, with accuracy at 0.854, recall at 0.854, and F1 score at 0.853. The results of DIC-Transformer were 0.024, 0.078, and 0.075 higher than those of the highest-performing comparison model, MobileNetV2. The results indicate that the DIC-Transformer outperforms other comparison models in classification and caption generation.

KEYWORDS

transformer, image caption, image classification, plant diseases, Faster R-CNN

# 1 Introduction

Rapid and accurate detection of plant diseases is crucial for increasing agricultural productivity. Traditionally, agriculture professionals rely on manual diagnosis to identify plant abnormalities caused by disease (Al-Hiary et al., 2011). However, this approach requires significant human and material resources and is not realistic (Ngugi et al., 2021). In response to these challenges, the use of image processing technology for automated diagnosis of plant diseases has garnered increased attention (Boulent et al., 2019).

In recent years, there has been remarkable progress in image classification due to the emergence of deep learning and neural networks. Among them, convolutional neural networks (CNNs) have shown good performance in image classification (such as Sun et al., 2022; Singh et al., 2023). However, a good CNN requires a large amount of training data (Keshari et al., 2018). Unfortunately, in the field of agricultural plant disease identification, the available labeled data are of poor quality and limited in quantity (Singh et al., 2020). Therefore, the first challenge in the task of classifying agricultural plant diseases through image classification is introduced: how to enhance the model's classification performance with a relatively small number of images.

Among the existing agricultural diseases, some diseases have very similar pathogenic characteristics. For instance, "apple anthracnose" and "pear anthracnose" in Figure 1 are challenging to differentiate based on visual features alone. As a result, the accuracy of CNN-based models in identifying similar disease classes is significantly reduced (Rzanny et al., 2022). However, this rarely occurs when agricultural professionals observe and confirm the disease. This presents a second challenge in identifying images of agricultural diseases: how to develop a model that can accurately identify specific plant diseases by replicating the process of manual disease diagnosis.

To address both of these challenges, this article introduces image caption generation techniques for the following reasons:

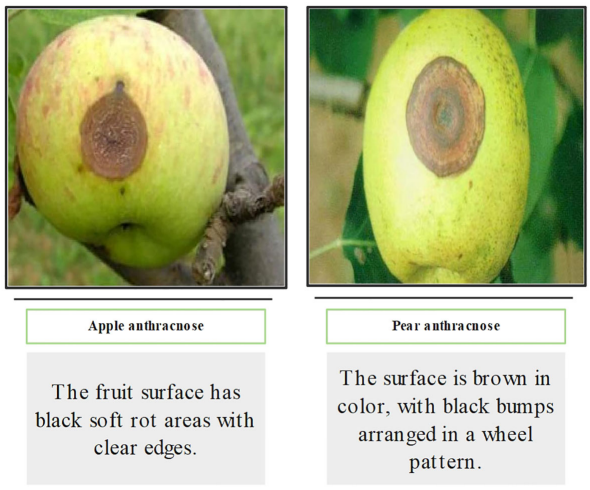



FIGURE 1
Examples of two diseases with similar pathogenic characteristics.

1. Image caption generation technology creates visual semantics based on features in disease images and then generates textual descriptions of affected areas using these visual semantics (Yang et al., 2022). This process closely resembles the behavior of agricultural experts manually identifying disease categories. This can address the second challenge presented.
2. The image classification task is single-task learning. When there are limited training data in the dataset, CNN-based image classification models may struggle to learn sufficient discriminant features, leading to slower performance in recognizing image classes (Luo et al., 2018). When there are limited image data samples, leveraging the benefits of multi-task learning can effectively mitigate the performance decline caused by the small sample size. Multi-task learning aims to extract useful information from other tasks and apply it to the current task, leading to an improved model performance model (Ruder, 2017). Image caption generation technology can extract rich semantic features from images (Xian et al., 2022). These visually and semantically rich features are shared during training, typically serving as discriminant features for recognizing objects. Therefore, by utilizing image caption generation technology, the model will continue to have a positive impact on image recognition even when there are relatively few image samples. Therefore, the combination of image captioning technology and image classification tasks addresses the first challenge. CNN-based image classification models may struggle to learn sufficient discriminant features, leading to slower performance in recognizing image classes (Luo et al., 2018). When there are limited image data samples, leveraging the benefits of multi-task learning can effectively mitigate the performance decline caused by the small sample size. Multi-task learning aims to extract useful information from other tasks and apply it to the current task, leading to an improved model performance model (Ruder, 2017). Image caption generation technology can extract rich semantic features from images (Xian et al., 2022). These visually and semantically rich features are shared during training, typically serving as discriminant features for recognizing objects. Therefore, by utilizing image caption generation technology, the model will continue to have a positive impact on image recognition even when there are relatively few image samples. Therefore, the combination of image captioning technology and image classification tasks addresses the first challenge.
3. By utilizing image caption generation technology, it is possible to combine images and text to create a multi-modal presentation. This tool is valuable for agricultural managers who need to analyze both visual and textual information simultaneously, allowing them to interpret agricultural plant disease classification results more comprehensively.

From the early methods to the recent advancements in deep learning, the accuracy of image caption generation has continuously improved. This has led to increased attention to the diversity of captions based on accuracy, which can generate more stylized image captions (Ghandi et al., 2023). In the field of agricultural diseases, the use of image caption generation technology is limited. While existing image recognition and classification technology for agricultural diseases is advancing, most models only provide the names of disease classes without clear explanations of the classification results (Kumar and Kumar, 2023). This lack of easy-to-understand explanations hinders farmers without specialized knowledge from correctly interpreting the recognition results, which does not align with real-world application needs. Given the current situation, this paper utilizes image caption generation technology in conjunction with image classification technology to produce descriptive information about disease characteristics based on the results of plant disease identification.

Based on image caption generation technology, a method called DIC-Transformer is proposed for agricultural plant disease image classification. This method can generate descriptive interpretations of disease areas in images.

The fourth section breaks down the method into four modules to accomplish three tasks: detection, interpretation, and classification. The four modules are the region detection module, sequence encoding module, caption generation module, and classification module. The disease region detection module completes the task of detecting the disease region. The second and third modules complete the interpretation task. The purpose of the interpretation task is to generate a textual description of the disease image features. The purpose of the classification task is to train a classifier that can identify categories of plant diseases. Finally, the name of the disease corresponding to the image is displayed. In the fifth section, the experimental part, we evaluate the DIC-Transformer on our self-constructed dataset (ADCG-18). The dataset contains 3,971 images, 9,040 instances of disease areas, and 3,971 textual descriptions. Experiments demonstrate that the DIC-Transformer method applied to the ADCG-18 dataset can address the proposed research topic. The article's contributions are divided into the following three parts:

- A method for plant disease image classification is proposed. This method can output the name of the disease class and additionally generate an explanatory description of the characteristics of this disease class.
- An agricultural disease caption generation dataset is collected, named ADCG-18, which contains images of 18 diseases and textual descriptions of the corresponding images.
- Extensive experiments prove that the performance of the DIC-Transformer on the dataset ADCG-18 is the best.

This paper is divided into six chapters: Abstract, Introduction, Related Work, Dataset Construction, Methods, Experiments, and Conclusion. The Abstract introduces the limitations of existing methods, the structure of the DIC-Transformer, the dataset, and the experimental results. The Introduction outlines the article's structure and the primary contributions. The Methods section outlines the principles and core concepts of DIC-Transformer. The Experimental section introduces the planned experiments, including the hardware configuration, software version, framework information, various parameters used, evaluation indicators, and the quantitative and qualitative analyses of the results. The Conclusion briefly outlines the issues addressed by our proposed model, its primary contributions, its limitations, and the future directions for improving the model.

# 2 Related work

## 2.1 Object detection

The continuous advancement of artificial intelligence, big data, and other technologies has ushered in new opportunities and challenges for image processing technology. The application of image processing technology in various fields is also more and more extensive and deep (Li et al., 2023). The rapid evolution of deep learning has also revolutionized the field of object detection. A large number of models of object detection use CNN models related to deep learning, such as R-CNN (Girshick et al., 2014), Fast R-CNN (Girshick, 2015), and Faster R-CNN (Ren et al., 2015). These methods improve detection performance by transforming object detection problems into classification and bounding box regression problems for candidate regions. In addition, these methods are essentially two-stage structures. After further development, single-stage detectors YOLO7 (You Only Look Once) (Wang et al., 2023) and SSD (Single Shot MultiBox Detector) (Liu et al., 2016) appeared. These methods reduce the complex candidate region generation process and improve the detection speed by predicting the class and location of the target directly on the image, but the detection accuracy in some scenarios is slightly inferior to that of the two-stage model. As object detection continues to evolve, researchers are constantly improving algorithms to improve performance. For example, attention mechanism, multi-scale feature fusion, and target shape information are introduced to enhance detection performance. Table 1 summarizes the techniques related to objective detection, showing the name of the method, the time it was proposed, and the advantages and disadvantages of the method. Since the method proposed in this paper mainly requires the accuracy of the object detection module, the two-stage object detection model Faster R-CNN based on deep learning was selected as the disease area detection module of DIC-Transformer.

## 2.2 Image caption generation

Image caption generation is a task that involves generating human-like descriptions or captions for given images. By generating textual descriptions related to the image, additional semantic information can be provided, enabling the computer to understand the image content more fully and deeply. Image captioning models generally come in

TABLE 1 Summary of work related to object detection.

| Object Detection | Time | Advantages | Disadvantages |
|---|---|---|---|
| R-CNN | 2014 | SS is used to generate regions, and CNN is used to extract features | The training is multi-stage, the space consumption is large, and the detection is slow; duplicate calculations are performed at the time of feature extraction |
| Fast R-CNN | 2015 | The speed is higher than that of R-CNN; end-to-end training is achieved | An external SS module is required to generate the candidate region, which can only run on the CPU and is slow |
| SSD | 2016 | High accuracy, fast detection, small targets, and ease of use | Sensitive to target intensity; sensitive to dataset quality; performance degradation in complex scenarios |
| Faster R-CNN (Swin Transformer) | 2022 | High detection accuracy and great speed improvement, truly realize the end-to-end object detection framework | It takes a relatively long time |
| YOLO7 | 2022 | Faster speeds, sufficient accuracy, and lower computing resource requirements | The accuracy is inflated; inaccurate detection of small targets; not versatile enough |

three classes: template-based (TB), retrieval-based, and artificial neural network (ANN)-based (Bai and An, 2018).

The template-based (TB) model is a traditional image caption model, which requires the appropriate title structure to be defined in advance. Kulkarni et al. (2013) proposed a method called Babytalk, which combines computer vision and natural language processing techniques. First, they used computer vision algorithms to extract key visual features from images, such as objects, people, and scenes. They then used these features as input to generate simple descriptions related to the image through natural language processing models. However, image captioning technology based on TB mode can only detect image content visually, which generally causes problems such as complexity, creativity, and extracted sentence coverage. In addition, unlike the manually written image title information, if the main structure of the caption is a constrained template, it will make the generated caption a bit unnatural (Deorukhkar and Ket, 2022).

The retrieval-based image caption generation technology retrieves a sentence or set of sentences from a pre-defined text description dataset based on a given query image to generate the title that best matches the image features of the target image. The generated caption can be either a defined statement or a statement composed of multiple retrieved statements (Bai and An, 2018). Hodosh et al. (2013) treated caption generation as a ranking task. Captions and images are then mapped into the latent space based on canonical relevance. Also, the top-ranking caption for the target image is selected by calculating the cosine similarity between the caption and the image. In addition, two developments in caption retrieval were proposed by Ordonez et al. (2016) to score the relationship between captions and images. The two developments are as follows: one is the retrieval of the entire available image, and the other is the retrieval of captions based on the geometric distance of scenes and objects. In Jeong et al. (2023), a novel search-type radiology report generation module called X-REM is proposed to improve clinical accuracy. Compared to the benchmark retrieval method, X-REM increases the number of zero error reports and reduces the average error severity.

ANN-based models use an encoder–decoder architecture when generating image captions. Images are first encoded to generate a corresponding high-level representation and then decoded using language modeling algorithms. There are two types of ANN-based models: 1) recursion-based models and 2) transformer-based models (Parvin et al., 2023).

The recursive encoder–decoder structure is widely used in multiple tasks, such as machine translation, language generation, and code generation. Among them, long short-term memory network (LSTM) (Naga Srinivasu et al., 2023) and gated recurrent units (GRUs) are neural network units that are often used to construct recursive structures. For example, Ye et al. (2018) utilized attention mechanisms and linear transformations to improve image caption generation. Through steps such as calculating similarity, normalizing processing, and weighted summation, dynamically focus on image areas to generate more accurate and coherent captions. Yang et al. (2018) introduced a shared backbone network that is used to extract image features. Then, on top of the backbone network, multiple domain-specific task networks are built to process image caption generation tasks in different fields. Each task network has its own independent decoder for generating captions for the corresponding domain.

However, the recursion-based encoder–decoder architecture needs to be generated word by word in the process of generating the caption sequence, so the parallel computation is not possible, resulting in a slower speed (Khan et al., 2022). Due to the nature of the self-attention mechanism in the Transformer model, the representation of each position can be computed simultaneously with other locations without the need for sequential loop structures. This makes the Transformer model highly parallel computing power (Khan et al., 2022). At the same time, the transformer model consists of multiple stacked encoder and decoder layers, each with multiple self-attention sublayers and feedforward neural network sublayers. This multi-layered structure enables the gradual extraction of higher-level abstract features and more accurate predictions. These features make the Transformer-based caption generation model have advantages in handling long-distance dependency, parallel computing, and abstract feature extraction. They enable the model to better understand the image content and produce accurate, smooth captions. In the study of Wang et al. (2020), a geometry perception converter is constructed to obtain the geometric representation capabilities of encoders and decoders. Liu et al. (2021a) proposed a full-network structure CPtr based on Transformer for image caption generation tasks. By combining image feature representation with position encoding through an

encoder, the decoder uses multiple Transformer layers to produce accurate captions. The attention mechanism is used to interact with the image with the text to improve the modeling ability. CPtr has shown excellent performance in image understanding and caption generation. In the study of Parvin et al. (2023), a transformer-based image description generative model is proposed that does not rely on recurrent or convolutional neural networks and is able to capture the interrelationships between objects. Experiments on COCO and Flickr datasets prove that the proposed method outperforms various state-of-the-art models in various evaluation indicators. In the study of Fang et al. (2022), ViTCAP, an image captioning model based on a pure visual transformer, is proposed, in which a grid representation is used without extracting regional features. In the study of Fei (2022), an attention-aligned converter for image captions is proposed, called $A^2$, which is a perturbation-based, self-supervised way to guide attention learning without any annotation overhead. In the study of Liu et al. (2022), a new model based on the encoder–decoder framework is proposed. In the encoder, the features of different layers in the ResNet-50 are fused to extract multi-scale information. In the decoder, a multi-layer aggregation converter (MLAT) is proposed to utilize the extracted information to fully generate sentences. In the study of Jing and Jinguang (2023), a hybrid structure image caption generation model based on a convolutional neural network and Transformer was proposed. It mainly fuses lightweight high-precision attention with convolutional networks to form attention residual blocks, which are used to extract visual features from input images. The features are then entered into the sequence model transformer. Table 2 is a summary of the related technologies for image caption generation and lists the proposed time, advantages, and disadvantages of different methods. Then, the appropriate type of method is chosen according to the actual needs. Transformer architecture is first used in natural language processing and later widely used in the field of computer vision. Many transformer-based image captioning models were introduced in the study of Ondeng et al. (2023), including those pre-trained using visual language, which has produced several state-of-the-art models, etc. Recent models show the Transformer's advantages in image caption generation. Therefore, the image caption generation model in this article used a transformer-based structure.

# 3 Dataset construction

## 3.1 Image collection and preprocessing

To establish a plant disease image dataset, it is first necessary to determine which plants and diseases are selected. Then, high-quality images are collected, and finally, the images are preprocessed according to the requirements. The specific process is as follows:

1. Determination of plant species and diseases: Before collecting images, it is first necessary to determine the plant species and diseases. The difficulty of image collection is taken into account when determining plant species and diseases. If a plant disease can be searched for more images and related image characterization information through common search engines, then the disease is listed as a candidate. Finally, 18 diseases belonging to 10 plant species are selected as research objects to construct the dataset, for example, common plants such as apples, pears, and tomatoes.
2. Image collection: Images are collected in a variety of ways, such as search engine downloads, web crawling, and manual collection. The image collection is mainly based on web crawling and search engine downloads, supplemented by manual collection. In the end, more than 50,000 candidate images are collected. At the same time, the collection of images complies with relevant regulations such as copyright and privacy.
3. Image preprocessing: More than 50,000 images need to be processed to ensure that they are fit for use. Disease-affected areas in plant disease images need to have clear outlines, obvious symptoms, appropriate exposure, and other features. Low-quality, blurry, or unclear images need to be removed. The image filtering process is as follows:
    - Auto-filtering: For images that are not related to agricultural plant diseases, such as people, watermarks, and text, we use deep learning models to identify and remove them.

TABLE 2 Summary of work related to image caption generation.

| Image caption | Method | Time | Advantages | Disadvantages |
|---|---|---|---|---|
| Template-based | Babytalk | 2013 | Descriptions can be accurately generated. | The content is single and relatively fixed, and the degree of human involvement is high. |
| Retrieval-based | X-REM | 2023 | Improved accuracy; simple to implement. | Performance depends on dataset size and retrieval algorithms; The generated descriptions are also relatively limited. |
| ANN-based | ALT<br>MLADIC<br>CPtr<br>DPP<br>ViTCAP<br>$A^2$<br>MLAT | 2018<br>2018<br>2021<br>2022<br>2022<br>2022<br>2022 | Automatically learn features in images; high flexibility and generalization ability; strong contextual comprehension. | High data demand; long training time; it is difficult to explain the specific decision-making process of the model for the generation of image captions. |

- Manual filtering: After automatic filtering, because the content of the image is generally very complex, some images still do not meet the requirements, and relevant agricultural professionals need manual screening to ensure that the dataset images are basic and suitable for use.

The filtered image is more suitable for use in terms of content and resolution. In addition, most of the images are based on different backgrounds, closer to real agricultural scenes. Finally, a dataset containing 3,971 high-quality images for 18 diseases is constructed.

In summary, the ADCG-18 contains two modes, text and image, which have the characteristics of data diversity. Many different forms of data can enrich feature representation: images and text are two different forms of data that can provide complementary information to describe an object or scene. By using both image and text data, richer and more comprehensive feature representations can be obtained, which improves the performance and generalization ability of the model.

Data augmentation and transfer learning are also possible: fusing image and text data can augment the size and diversity of datasets to achieve data augmentation and improve the robustness and generalization of models. In addition, transfer learning between image and text data can help learning in one domain and improve the effectiveness of the model by helping another with what is learned in another. Therefore, it would be beneficial to consider both image and text data when building datasets. However, while data diversity is beneficial, there may be some potential biases, such as modal bias and data association bias. Modal bias is due to semantic differences between images and text, and models can produce bias when processing data with different modalities. For example, in image recognition tasks, the model may be more inclined to classify by image features and ignore the information described by text. Data correlation bias is the possibility of correlation bias when image and text data are combined through association rules or manual matching. Even if the association rule or matching process is deterministic, it is inevitable that there will be some errors or biases, resulting in the inaccurate correlation of images and text in the dataset.

These biases can affect the performance and generalization ability of the model, allowing the model to perform poorly against real-world samples. Therefore, when using datasets containing image and text information, it is necessary to pay attention to and minimize these potential biases and make corresponding preprocessing and adjustments to improve the robustness and accuracy of the model.

## 3.2 Data segmentation

The ADCG-18 contains 18 types of diseases due to the presence of data-enhanced similar images in the dataset, so it cannot be randomly divided according to the proportion when dividing the dataset. It is necessary to divide the images of each disease category according to the proportion and finally combine each divided disease data into the final dataset. Otherwise, the data images in the training set and the test set will be duplicated, resulting in higher experimental results than the real results. The data in this article are divided according to a 7:3 ratio. Table 3 shows the results of the dataset division. The method of image enhancement is shown in Figure 2.

## 3.3 Dataset details

The ADCG-18 contains two parts: images and textual descriptions. In addition, the images and textual descriptions in the dataset should correspond to each other. In order to achieve this, it is necessary to determine which descriptive keywords are unique to a certain disease class and ensure that the characteristic keywords in the textual description of each disease are clearly distinguished from the others.

Collect the description of diseases in the dataset from Internet resources such as relevant agricultural websites and then artificially extract sentences that meet the requirements for use. Finally, the textual description of each image consists of approximately six to 14 words. Note that each image corresponds to one textual description.

Image labeling uses relevant tools to mark the disease area in the image, including the label of each disease category and the disease area's true bounding box location. We use the image annotation tool LabelMe (Torralba et al., 2010) to manually mark the boundaries of the disease area in the image and save the boundary information in a JSON file. The version of LabelMe is 4.5.13. Figure 3 shows the result of a hand-annotated image.

Since at least one disease region exists per image, there are more disease instances (NOAD) than images (NOAI) for each class. Since the symptoms of each disease are different, the number of bounding boxes for each disease varies greatly. For example, in Table 4, Infection of Peanut_leaf_spot and Pear_Rust are characterized by the presence of more small-area spots on the foliage, and the number of instances is 1,478 and 951, respectively, which is much higher than the average number of instances. Figure 4 shows some image samples and corresponding textual descriptions. Figure 5 shows the characteristic images of 18 diseases.

In addition, as depicted in Figure 3, we primarily describe the quantity, color, size, sharpness of boundary lines, brightness, darkness, and the degree of variation in characteristics when constructing the dataset. These characteristics encompass all the pathogenesis features of the disease and can fully describe the occurrence of the disease. The text is stored in a JSON file as a dictionary. Each entry begins with a disease name followed by a description of its features. It is important to note that image name

TABLE 3 Dataset segmentation results.

| | Image_num | Instance_num |
|---|---|---|
| Test set | 1,180 | 2,842 |
| Train set | 2,791 | 6,198 |
| Total | 3,971 | 9,040 |

Image_num is the number of disease images. Instance_num is the number of disease instances.

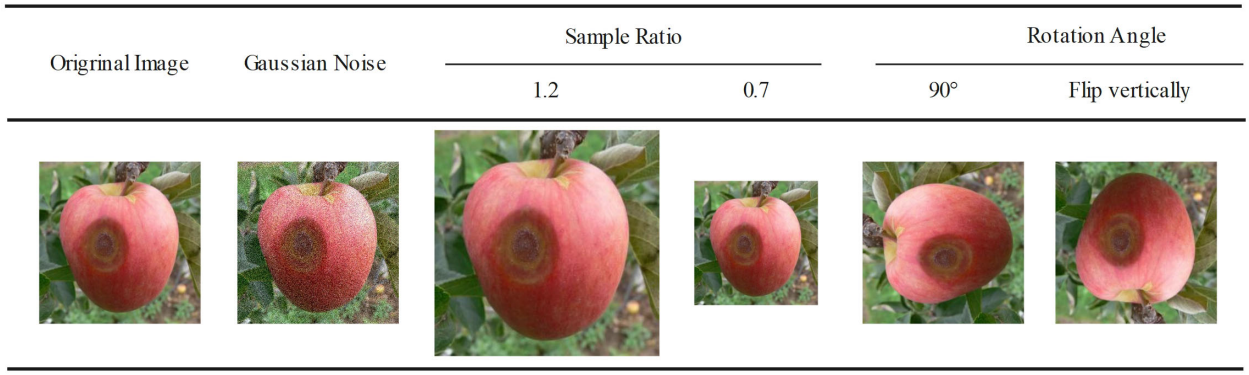



FIGURE 2
Examples of plant disease images of 18 types in the dataset.

labels have been included in the text dataset to link each statement to an image, which aids in the model's training process.

# 4 Methods

## 4.1 Overview

Figure 6 shows the overall structure diagram of the model DIC-Transformer proposed in this paper, with the DIC-Transformer divided into four modules. ① is a disease region detection module, which is used to obtain relevant information about the disease area in the image. Relevant information is divided into two categories: location vectors and visual feature vectors of disease areas. ② and ③ are sequence encoding module and caption generation module, respectively, which are used for image caption generation tasks. ④ is a classification module designed to classify images. We divide these four modules into two stages, where the first stage model includes module ① and the second stage model includes modules ②, ③, and ④.

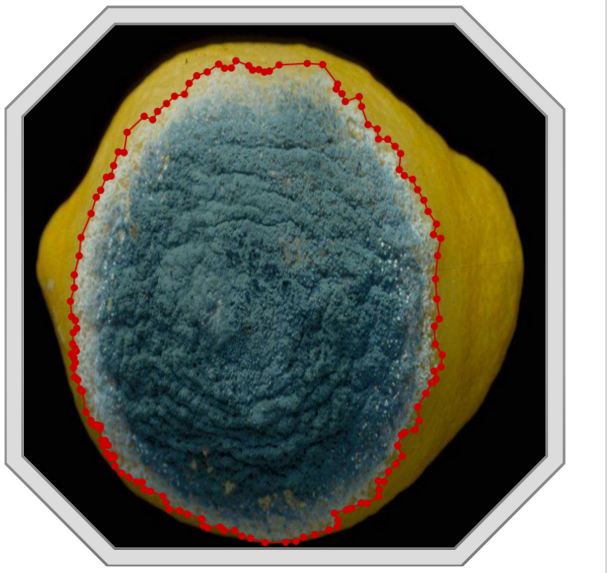


FIGURE 3
An example image processed by LabelMe.

The first stage model, the region detection module, can acquire the label of the disease image and the vector representation of the disease area feature in the image. These feature vectors are integrated into the sequence as inputs to the sequence coding module. The average value of the integrated visual feature vector sequence is used to represent the entire image. The output of the sequence encoding module, in turn, serves as keywords and values for attention blocks in each text generation module. The input to the text generation module is a sequence of word vectors, and the first position of the sequence is the symbol [Start], which marks the beginning of generating the word vector sequence. The output of the text generation module is a sequence of word vectors similar to the input. The text generation module is like RNN one word input, the Q is calculated by the word that has appeared, the K and V are calculated by the sequence coding module, and the result is obtained after all the text generation modules and then FC+Softmax. After that, the result is used as input to the text generation module, and the whole process is repeated until the symbol END is output. The output of the END symbol at this point indicates that a text sequence has been generated. The output of each layer of the text generation module is a visual vector weighted by text features. Now, it is used as the keyword and value of the classification module, and the average value of the visual feature vector sequence is used as the query of the classification module. Finally, the output of the disease classification module is the probability for each category. In order to clearly describe the implementation details of each module, a symbology is established.

## 4.2 Region detection module

The region detection module is used to obtain the feature vectors and bounding box coordinates for each disease area in the image. Let $I = \{I_1, \ldots, I_s, \ldots, I_S\}$ denote the set of images, where

10.3389/fpls.2023.1273029

TABLE 4 Statistical analysis of datasets.

| SN | Category name | NOAI | NOAD | NOTRI | NOTRD | NOTI | NOTD |
|---|---|---|---|---|---|---|---|
| 1 | Anthrax | 165 | 344 | 116 | 219 | 49 | 125 |
| 2 | Apple_bitter_rot | 252 | 408 | 176 | 270 | 76 | 138 |
| 3 | apple_cracking | 156 | 248 | 110 | 175 | 46 | 73 |
| 4 | Apple_mould_heart_disease | 195 | 277 | 135 | 177 | 60 | 100 |
| 5 | Apple_water_heart_disease | 230 | 898 | 162 | 684 | 68 | 214 |
| 6 | Brown_rot | 261 | 289 | 173 | 184 | 88 | 105 |
| 7 | Citrus_schizoderma | 381 | 557 | 291 | 324 | 90 | 233 |
| 8 | cracking | 371 | 673 | 263 | 388 | 108 | 285 |
| 9 | Jujube_Anthrax | 219 | 649 | 150 | 337 | 69 | 312 |
| 10 | Mango_Anthracnose | 217 | 343 | 152 | 258 | 65 | 85 |
| 11 | Olive_anthracnose | 115 | 234 | 77 | 137 | 38 | 97 |
| 12 | Papaya_anthracnose | 212 | 673 | 149 | 499 | 63 | 174 |
| 13 | Peanut_leaf_spot | 235 | 1,478 | 164 | 1,140 | 71 | 338 |
| 14 | Pear_Rust | 268 | 951 | 190 | 680 | 78 | 271 |
| 15 | Penicillium | 165 | 187 | 115 | 136 | 50 | 51 |
| 16 | strawberry_anthracnose | 174 | 204 | 121 | 139 | 53 | 65 |
| 17 | Tomato_anthracnose | 188 | 348 | 130 | 251 | 58 | 97 |
| 18 | Tomato_cotton blight | 167 | 279 | 117 | 200 | 50 | 79 |

SN, the category index; NOAI, the total number of images; NOAD, the total number of disease areas; NOTRI, the number of images in the training set; NOTRD, the number of disease areas in the training set; NOTI, the number of images in the test set; NOTD, the number of disease areas in the test set.

$S$ denotes the total number of images. Let $D^s = \{d_1^s, d_2^s \ldots, d_t^s\}$ denote disease areas that appear in each image $I_s$. Each disease area contains two types of information: a) $d_{t,1}^s \in \mathbb{R}^{1\times T}$ represents the label of the diseased area, where $T$ represents the total number of categories for disease areas, and b) $d_{t,2}^s \in \mathbb{R}^{1\times 4}$ represents the bounding box coordinates of each diseased area. Specifically, the region detection module learns two mapping functions: $\mu_a$ and $\mu_b$. $\mu_a$ is used to obtain the category labels of each disease area, while $\mu_b$ is used to obtain the bounding box coordinates for each disease area. The details of the two functions are shown in Equations 1, 2.

$$\hat{d}_{*,2}^s = \mu_b(I_s; \alpha_b), d_{*,2}^s \in \mathbb{R}^{t\times 4} \tag{1}$$

$$\hat{d}_{*,1}^s = \mu_a(I_s; \alpha_a), d_{*,1}^s \in \mathbb{R}^{t\times T} \tag{2}$$

where $\alpha_a$ represents the weight of the function $\mu_a$ and $\alpha_b$ represents the weight of the function $\mu_b$. Area ① in Figure 6 represents the region detection module, where $F^s = \{f_1^s, f_2^s, \ldots, f_t^s$ represents the features of diseased areas in the image. These features are obtained from the feature map generated by the last convolutional layer of the backbone network in the object detection process. Specifically, $f_t^s \in \mathbb{R}^{1\times C}$ represents the feature vector of a specific disease area in the image. The default value for $C$ is 1,024.

## 4.3 Sequence encoding module

The structure of the sequence encoding module is shown in area ② in Figure 6. The intermediate variable of the function $\mu_a$ in the region detection module is used as input to the sequence encoding module, where the intermediate variable is essentially the visual vector generated by the integration of all disease regions in the image. This collection of visual vectors is referred to as the visual vector sequence, denoted as $F^s = \{f_1^s, f_2^s, \ldots, f_t^s\}$, where $F^s \in \mathbb{R}^{t\times C}$. To encode this sequence of visual vectors, an input embedding layer is utilized, which consists of two fully connected layers with an output dimension of $C'$, having a default value of 1,024. This process is represented by Equation 3.

$$F_s^\star = MLP_{emb}^1(F^s; \alpha_{emb}^1), F_s^\star \in \mathbb{R}^{t\times C'} \tag{3}$$

where $\alpha_{emb}^1$ is the weight of $MLP_{emb}^1$.

The self-attention mechanism can be regarded as an improvement mechanism for the attention mechanism in a certain application scenario. It becomes less dependent on external information and has a superior performance in capturing internal correlations in data or features. The calculation process of the self-attention mechanism is as follows: first, the input data is converted into an embedding vector. According to the embedding

10.3389/fpls.2023.1273029




FIGURE 4
Dataset examples.

vector, the three vectors of $Query(Q = \{q_1, \ldots, q_s\})$, $Key$ $(K = \{k_1, \ldots, k_t\})$, and $Value(V = \{v_1, \ldots, v_t\})$ are obtained. Calculate a score for each vector. To ensure gradient stability, use score normalization by dividing by., where $\sqrt{Y}$ is a scaling factor to prevent the input values for the softmax function from becoming too large. Apply a softmax activation function to score. The specific process is represented by Equation 4.

$$Self\ Atention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{Y}})V \tag{4}$$

where $Q \in \mathbb{R}^{s\times a}, K \in \mathbb{R}^{t\times a}, V \in \mathbb{R}^{t\times a}$.

The attention mechanism is position-insensitive, and even swapping the position of two elements in the sequence has no effect on the encoded result. Therefore, a positional vector notation is proposed in the Transformer to add a fixed positional vector to each position of the sequence. However, the visual vector sequences in this paper only have a spatial position relationship and no context relationship. Therefore, for the use of positional encoding, we use the four coordinates of the visual bounding box as positional encoding, discarding the original sine cosine function. In addition, the multi-head self-attention mechanism is used to study the relationships within the visual vector sequence. The self-attention mechanism is repeated $h$ times in the long self-attention mechanism. The specific process is represented by Equation 5.

$$\begin{aligned} &MultiHead(Q, K, V) = Concat(H_1, \ldots, H_h)W^O, \\ &H_s = Self\ Atention(QW_s^Q, KW_s^K, VW_s^V) \end{aligned} \tag{5}$$

where $W^O \in \mathbb{R}^{a\times a}$ and $W_s^Q, W_s^K, W_s^V \in \mathbb{R}^{a\times \frac{a}{h}}$.

The sequence coding module consists of two sublayers: the multi-head self-attention layer and the feedforward neural network layer. Each sublayer is followed by the AddNorm function. The AddNorm function is a common regularization operation commonly used in neural networks. It combines residual connectivity and layer normalization to enhance the representation and training effect of the network. The calculation process of the sequence encoding module is shown in Equations 6, 7.

$$A_s^1 = AddNorm(MultiHead([F_s^*; D_2^s], [F_s^*; D_2^s], [F_s^*; D_2^s])) \tag{6}$$

$$\begin{aligned} Sout_s^1 &= AdNorm(FFN(A_s^1)) \\ &= AddNorm(max(0, A_s^1 W_1 + b_1)W_2 + b_2) \end{aligned} \tag{7}$$

where $A_s^1 \in \mathbb{R}^{t\times C'}, W_1, W_2 \in \mathbb{R}^{C'\times C'}, b_1, b_2 \in \mathbb{R}^{C'}, Sout_s^1 \in \mathbb{R}^{t\times C'}$. The output of each layer of the encoder in the sequence encoding block is $Sout_s^N \in \mathbb{R}^{t\times C'}$, N represents the number of encoders in the sequence encoding module.

## 4.4 Caption generation module

Area ③ in Figure 6 shows the structure of the caption generation module. This module predicts the next word based on the input word and ensures that the resulting sentence visually corresponds to the disease area in the image. Prediction begins upon recognizing the marker [*Start*]} and concludes when the marker [*End*]} is generated. Both [*Start*]} and [*End*]} are zero vectors. The input to the caption generation module is denoted as $P_s^{input} = \{[Start], p_1, \ldots, p_l, \ldots, p_L,\}$, where $P_s^{input} \in \mathbb{R}^{(L+1)\times L'}$ represents the description statement corresponding to the visual part $D_s$. The output of the caption generation module is represented as $P_s^{output} = p_1, \ldots, p_l, \ldots, p_L, [End]$, where $P_s^{output} \in \mathbb{R}^{(L+1)\times L'}$. Here, $L$ is the length of the input statement and $L'$ is the number of words in the database.

The input statement of the module needs to be encoded by the word embedding layer, which consists of two fully connected layers with an output dimension of $C'$. The embedding layer process is represented by Equation 8.

$$P_s^{input,*} = MLP_{emb}^2(P_s^{input}; \alpha_{emb}^2), P_s^{input} \in \mathbb{R}^{(L+1)\times C'} \tag{8}$$

where $\alpha_{emb}^2$ is the weight of $MLP_{emb}^2$.

The decoder model consists of N identical decoder blocks stacked, each consisting of three different sublayers. There are residual connections and layer normalization between each sublayer. Moreover, the masked multi-head self-attention sub-layer uses a mask to prevent information from future output words from being used when training a given output word. The form of MASK is shown in Equation 9.

$$MASK = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & 1 \end{bmatrix}, MASK \in \mathbb{R}^{(L+1)\times(L+1)} \tag{9}$$




FIGURE 5
Examples of plant disease images of 18 types in the dataset.

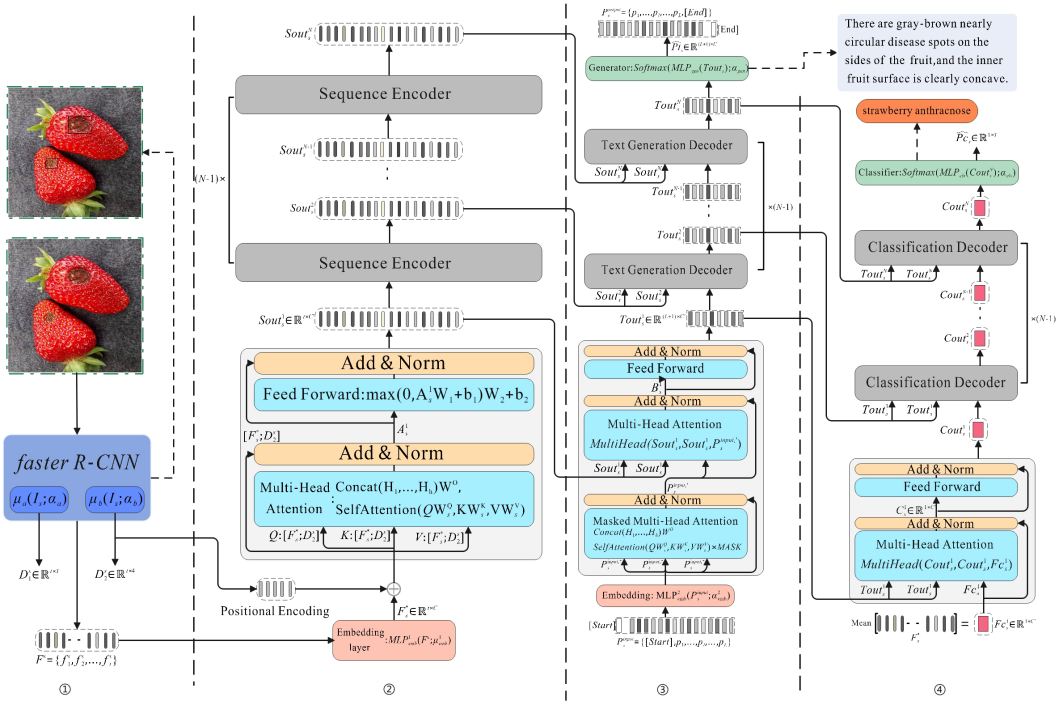



FIGURE 6
Model overview diagram. The numbers at the bottom of the figure represent four modules: ① represents the region detection module, ② represents the sequence encoding module, ③ represents the caption generation module, and ④ represents the classification module.

Thus, the process of self-attention with masking can be expressed as Equation 10.

$$\begin{aligned} &MaskMultiHead(Q,K,V) = Concat(H_1,\ldots,H_h)W^O \\ &H_s = Self\ \ Attention(QW_s^Q, KW_s^K, VW_s^V) \times MASK \end{aligned} \tag{10}$$

where $W^O \in \mathbb{R}^{a\times a}$ and $W_s^Q, W_s^K, W_s^V \in \mathbb{R}^{a\times \frac{a}{h}}$.

The first layer decoder in the caption generation module is taken as an example, and formulas are used to describe the working process of the decoder. The input encoded by the embedding layer is first copied three times and then input into a masked self-attention block to obtain an output, which can be represented by Equation 11.

$$P_s^{input,'} = AddNorm(maskMultiHead(P_s^{input,*}, P_s^{input,*}, P_s^{input,*})) \tag{11}$$

where $P_s^{input,'} \in \mathbb{R}^{(l+1)\times C'}$. The output $Sout_s^1$ of the first-layer encoder in the sequence encoding module is $K$, $V$, and $P_s^{input,'}$ is $Q$. Then, $Q$, $K$, and $V$ are used to calculate multi-head self-attention, the purpose of which is to explore the implicit relationship between visual features and semantic features, essentially using visual vectors weighted by text features to generate descriptive statements. The specific process can be represented by Equations 12, 13.

$$B_s^1 = AddNorm(MultiHead(Sout_s^1, Sout_s^1, P_s^{input,'})) \tag{12}$$

$$Tout_s^1 = AddNorm(FFN(B_s^1)) \tag{13}$$

where $B_s^1 \in \mathbb{R}^{(L+1)\times C'}$. $Tout_1^s \in \mathbb{R}^{(L+1)\times C'}$ represents the output of the first layer decoder of the caption generation module.

The caption generation module contains a total of N decoders, and the output of the last layer decoder is $Tout_s^N$. Then, two fully connected layers are used to convert the output $Tout_s^N$ into the distribution probability of each word in the database. This probability can be expressed using Equation 14.

$$\widehat{Pt_s} = Softmax(MLP_{gen}(Tout_s^N); \alpha_{gen}) \tag{14}$$

where $\hat{P}t_s \in \mathbb{R}^{(L+1)\times L'}$ is the weight of $MLP_gen$.

## 4.5 Classification module

The structure of the classification module is shown in area ④ in Figure 6. The purpose of the classification module is to predict disease image categories based on the output of the caption generation module. The output of the caption generation module serves as $K$ and $V$ for attention blocks in the classification decoder. Its output is essentially a visual vector weighted by text features. In addition, the classification module and the caption generation module have the same number of decoders. Let $Fc_s^1 = Mean(F_s^*)$, $Fc_s^1 \in \mathbb{R}^{1\times C'}$ denote a complete image of an agricultural disease, which is taken as the $Q$ in the attention block. Taking the first attention block in the classification module as an example, the operation process is represented by Equations 15, 16.

$$C_s^1 = AddNorm(MultiHead(Tout_s^1, Tout_s^1, Fc_s^1)) \tag{15}$$

$$Cout_s^1 = AddNorm(F\,F\,N(C_s^1)) \tag{16}$$

where $C_s^1 \in \mathbb{R}^{1\times C'}$. $Cout_s^1 \in \mathbb{R}^{1\times C'}$ is the output of the first decoder in the classification module. Let $Cout_s^N \in \mathbb{R}^{1\times C'}$ denote the output of the last decoder. The output $Cout_s^N$ of the last decoder is then converted into a probability distribution of the agricultural disease image category through two fully connected layers. The probability equation is shown in Equation 17.

$$\widehat{P_{c_s}} = Softmax(MLP_{cls}(Cout_s);\ \alpha_{cls}) \tag{17}$$

where $\hat{P}c_s \in \mathbb{R}^{1\times T}$, $T$ is the number of disease categories, and $\alpha_{cls}$ is the weight of $MLP_{cls}$.

# 5 Experimental

## 5.1 Experimental content

The DIC-Transformer mainly includes three modules: disease region detection module, image caption generation module, and classification module. Then, the results of Faster R-CNN experiments in 16 different backbones are analyzed, and an optimal object detection model is selected to process the input images. Next, the second task, the image caption generation task, needs to be tested and analyzed, and its main job is to compare DIC-Transformer with some existing caption generation models. Finally, we need to evaluate the performance of the classifier. Since DIC-Transformer is a two-stage method, the experiment mainly consists of two tasks:

1. Object detection backbone comparison experiment based on Faster R-CNN.
2. Quantitative and qualitative analyses of the DIC-Transformer. This task is divided into the following four experiments:
   a. Comparative experiment of image caption generation model.
   b. Comparative experiments of classification models.
   c. Ablation experiments of DIC-Transformer.
   d. Qualitative analysis of DIC-Transformer and classic image caption generation models.

## 5.2 Experimental details

Object detection backbone comparison experiments based on Faster R-CNN are implemented in two open-source frameworks, Detectron2 (Wu et al., 2019) and MMDetection (Chen et al., 2019). Detectron2 is a robust object detection platform developed by FAIR (Facebook AI Research) in 2019. Several state-of-the-art detection and segmentation algorithms are already integrated, eliminating the need to develop these networks from the ground up. There are two types of object detectors: one-stage and two-stage detectors. Detectron2 is a two-stage system, and the detection task is carried

out in two steps. The first step is to extract the region of interest (RoI). The second step involves target classification and positioning. The nature of these two-stage detectors makes them slower than one-stage detectors such as YOLO and SSD, but they can produce more accurate results. Under Detectron2, we utilize the Faster R-CNN + FPN algorithm and employ a pre-trained model. In Detectron2, there are no epochs, only iterations. However, the maximum number of iterations can be artificially set based on the size of the dataset.

MMDetection is an open-source project initiated by SenseTime and the Chinese University of Hong Kong for object detection tasks. It implements a wide range of object detection algorithms based on PyTorch and encapsulates the processes of dataset construction, model building, training strategies, and other tasks into modules. When building a new algorithm with MMDetection, the process typically involves the following steps: registering the dataset, registering a model, building a configuration file, and conducting training and validation.

The second stage of the DIC-Transformer is trained and tested on an NVIDIA P100-16G with CUDA 11.3 using Python 3.8 and PyTorch 1.10 on Ubuntu 18.04. Additionally, the version of Detectron2 is v0.6, and the version of MMDetection is 2.25.1.

The parameters used in the experimental process are detailed in Table 5. Both frameworks employ the same parameter settings, where unexposed parameters utilize the default values within the framework.

The region detection module is used to extract the feature and location of the disease area as the input of the caption generation module. The parameters for all comparison and ablation experiments are shown in Table 6.

The second stage of the DIC-Transformer is trained and tested on NVIDIA P100-16G with CUDA 11.3 using Python 3.8 and PyTorch 1.10 on Ubuntu 18.04. Additionally, the Detectron2 version used is v0.6, and the MMDetection version is 2.25.1.

## 5.3 Evaluation metrics

We mainly evaluate three tasks: object detection task, image caption generation task, and image classification task. Object detection task uses $mAP/mAP_{50}/mAP_{75}$ as quantitative indicators. The image caption generation task uses BLEU, Cider-D, and Rouge as quantitative indicators. Image classification task uses Acc, Recall, and F1 as quantitative indicators. Some basic concepts of evaluation indicators are as follows.

### 5.3.1 IoU

This represents an intersection over the previous union, essentially the overlap between the predicted range and the true range divided by the sum of the predicted range and the true range. Equation 18 is the calculation process of the IoU.

$$IoU = \frac{P \cap G}{P \cup G} \tag{18}$$

where $P$ is the predicted bounding box and $G$ is the ground-truth bounding box.

### 5.3.2 Confusion matrix

This is a summary of the prediction results, where TP represents the number of predictions that will be positive to positive classes. FN differs from TP in that it is the number of positive classes predicted as negative classes. FP is the exact opposite of FN, and it is the number of negative classes predicted as positive classes. The final TN is the exact opposite of TP, which represents the number of predicted negative classes as negative classes. In the taxonomic issue, for a disease category like strawberry anthracnose, the sample labeled "strawberry anthracnose" is considered a positive sample, while all other samples are considered negative. Therefore, in the classification problem, TP represents the image predicted by the model as strawberry anthracnose, and the true label is also strawberry anthracnose, i.e., the number of samples correctly predicted as positive. FN represents the image that the model predicts to be non-strawberry anthracnose, while the true label is strawberry anthracnose, i.e., the number of samples that are falsely predicted as positive. TN represents the image that the model predicts to be non-strawberry anthracnose, while the true label is also not strawberry anthracnose, i.e., the number of samples correctly predicted to be negative. FP represents the image that the model predicts to be strawberry anthracnose, but the true label is not strawberry anthracnose, i.e., the number of samples that are incorrectly predicted as positive.

### 5.3.3 Precision

It indicates the proportion of samples that are actually positive in a sample that is predicted to be positive. The calculation process is shown in Equation 19.

$$precision = \frac{TP}{TP + FP} \tag{19}$$

TABLE 5 Faster R-CNN benchmark experiment parameter setting.

| Parameter | Pre-trained model | Optimizer | Warm-up strategy | Warm-up iters | Warm-up ratio |
|---|---|---|---|---|---|
| Value | ImageNet | SGD | Liner | 1,000 | 0.001 |
| Parameter | BatchSize | Learning rate | Momentum | Weight decay | Iteration rounds |
| Value | 16 | 0.02 | 0.9 | 0.0001 | 10,000 |

TABLE 6 Comparative experiment parameter settings for DIC-Transformer.

| Parameter | Noamopt Warmup | Optim Alpha | Optim Beta | Optim Epsilon | Noamopt | Optimizer |
|---|---|---|---|---|---|---|
| Value | Adam | 0.9 | 0.999 | 1e−4 | True | 2,000 |
| Parameter | Noamopt Factor | Grad Clip | BatchSize | Learning Rate | T-Depth | Epoch |
| Value | 1 | 0.1 | 16 | 2e−4 | 6 | 30 |

### 5.3.4 mAP

It measures the ability of the trained model to detect all classes. $mAP/mAP_{50}/mAP_{75}$ is $mAP^{IoU=0.5:0.05:0.95}/mAP^{IoU=0.5}/mAP^{IoU=0.75}$. $IoU$ = 0.5:0.05:0.95 means that the intersection over union ratio is calculated for values ranging from 0.5 to 0.95, with an interval of 0.05. Equation 20 represents the calculation process of $mAP$.

$$mAP = \frac{\sum_{i=1}^{K} AP_i}{K} \tag{20}$$

where $AP$ is defined as the area between the interpolated precision-recall curve and the X-axis, $K$ represents the number of categories, and $AP_i$ represents the $AP$ value for a category.

### 5.3.5 BLEU

What the BLEU algorithm is actually doing: judging how similar two sentences are. Equation 21 represents the implementation of BLEU.

$$bleu_n = \frac{\sum_{c\in candidates} \sum_{n-gram\in e} Count_{clip}(n-gram)}{\sum_{c'\in candidates} \sum_{n-gram'\in e'} Count(n-gram')} \tag{21}$$

where the purpose of the first summation symbol in the formula is to calculate all the candidates because there may be several sentences in the calculation. Then, the purpose of the second summation symbol is to count all n-gram in the candidate. The number of n-gram in the reference statement is denoted by $Count_{clip}(n-gram)$. Thus, the numerator is the representation of how many n-gram appears in a given candidate reference statement. The number of n-gram′ in the candidate is represented by $Count(n-gram')$. Therefore, the denominator is the number of n-gram among all candidates.

### 5.3.6 Cider-D

The purpose of A is to prevent gaming problems with evaluation indicators. The problem with gaming is to prevent optimizing the algorithm for a certain evaluation indicator so that when the human gives a low score, the evaluation index gives a high score. Equations 22, 23 describe the calculation process for this evaluation metric.

$$Clder - D_n(x_i, y_i) = \frac{100}{m}\sum_j e^{\frac{-(l(x_i)-l(y_{i,j}))^2}{2\delta^2}} * \frac{\min\left(g^n(x_i), g^n(y_{i,j})\right)\cdot g^n(y_{i,j})}{||g^n \le ft(x_i)||||g^n(y_{i,j})||} \tag{22}$$

$$Cider\ D\ (x_i, y_i) = \sum_{N}^{n=1} w_n CiderD_n(x_i, y_i) \tag{23}$$

where $l(x_i)$ is the length of the text generated by the model and $l(y_{ij})$ is the length of the real text. Multiplying by 100 makes the value of Cider-D similar in size to the value of other evaluation indicators. $g^n$ consists of $g_1, g_{2\ldots}\ g_n$. $g_k$ is used to calculate the TF-IDF value for each N-gram. We set $\delta$ = 6 and N = 4.

### 5.3.7 Rouge

Rouge metrics are very similar to BLEU metrics. The main difference is that ROUGE is based on recall, while BLEU focuses more on precision. The calculation process is shown in Equation 24.

$$Rouge_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \tag{24}$$

where $\beta$ = 1.2. The calculation process for $R_{lcs}$ and $P_{lcs}$ is shown in Equations 25, 26, respectively.

$$R_{lcs} = \frac{LCS(X,Y)}{m} \tag{25}$$

$$P_{lcs} = \frac{LCS\ \ (X,Y)}{n} \tag{26}$$

where $X$ represents the text generated by the model, and the length is $m$. $Y$ represents the real text of the image, and the length is $n$. $LCS$ is the longest common subsequence.

### 5.3.8 Acc

The full name of ACC is Accuracy, which stands for accuracy. The accuracy can be expressed by Equation 27.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{27}$$

### 5.3.9 Recall

It indicates the proportion of correctly predicted true values among all positive cases, which can be understood as how many correct targets are recalled. The calculation process is shown in Equation 28.

$$Recall = \frac{TP}{TP + FN} \tag{28}$$

### 5.3.10 F1

The core idea of F1 is to improve Precision and Recall as much as possible and also to make the difference between the two as small as possible. Equation 29 represents the calculation process for evaluating criterion F1.

$$F1 = 2 \times \frac{P \times Recall}{P + Recall} \tag{29}$$

where $P$ is precision.

## 5.4 Experimental results

### 5.4.1 Object detection benchmark experiment based on Faster R-CNN

The primary objective of this experiment is to assess the impact of different feature extraction networks on object detection. In the end, 16 backbones are selected to evaluate the overall performance of the Faster R-CNN. The backbones fall into 10 main types: ResNet (He et al., 2016), Res2Net (Gao et al., 2019), ResNeSt (Zhang et al., 2022), RegNet (Radosavovic et al., 2020), HrNet (Sun et al., 2019), HarDNet (Chao et al., 2019), EfficientNet (Tan and Le, 2019), MobileNetV2 (Sandler et al., 2018), VoVNet (Lee et al., 2019), and Swin Transformer (Liu et al., 2021b). While the experiments are conducted using two different frameworks, Detectron2 and MMDetection, the parameters remain consistent. Table 5 shows the parameter settings used in the current experiment, and undisclosed parameters use the default settings of the respective frameworks.

In the first stage of object detection, the size of the input image is not fixed, but we normalize the size of the image in the dataset. This means that the processed image has a moderate aspect ratio and sharp pixels. Before extracting the image features in the initial stage, we standardize the image size to 448 × 448 and extract 1 × 1,024 feature tensors using the feature extractor. These are then saved in an npy file for use in subsequent image caption generation tasks. It should be noted that there will be multiple diseased regions in an image, and each diseased region will generate a 1 × 1,024 tensor. Therefore, there will be several tensors in the.npy file corresponding to an image. The Faster R-CNN model used to extract image features utilizes the Swin Transformer as its backbone. In the Swin Transformer, the default stride of the convolution is set to 2, and the size of the convolution kernel is 3 × 3. This is because in the Swin Transformer Block, the convolutional kernel size of each 2D convolutional layer is set to 3 × 3 for local feature extraction. It is important to note that the size of the convolutional kernel can be adjusted based on the specific task and dataset to achieve optimal performance. In the Swin Transformer, using a 3 × 3 size as the default is a common choice, but it can be adjusted as necessary. Specifically, in a Swin Transformer Block, a 2D convolutional layer typically employs a convolution operation with a stride of 2. Using a two-step convolution can effectively decrease the size of the feature map and reduce the computational workload. This configuration is used in the Swin Transformer to achieve a balance between chunk processing and attention mechanisms, leading to improved performance and results. The activation function used in the encoder–decoder is ReLU, which enhances the expressive capability of the features through non-linear transformation. It strengthens the part with larger values and suppresses the part with smaller values, resulting in better expression of the relevant features.

Tables 7, 8 show the experimental results under the Detectron2 and MMDetection frameworks, respectively. Finally, the Swin Transformer is selected for feature extraction. Swin Transformer adopts a hierarchical structure, creating layered representations by initially using small-sized patches and gradually merging adjacent patches into deeper layers of the Transformer. When $IoU$ = 0.5:0.05:0.95, $mAP$ is 0.674. When $IoU$ = 0.5, $mAP_{50}$ is 0.862. When $IoU$ = 0.75, $mAP_{75}$ is 0.793. The results indicate that, among the tested backbones, the Swin Transformer performs the best. Moreover, the experimental results show that there is no absolute linear upward relationship between the backbone performance and parameter scale, and the appropriate parameter scale should be analyzed according to the specific use scenario. For example, ResNet-101 and ResNet-50 have $mAP_{50}$ values of 0.771 and 0.788, respectively, indicating a proportional relationship between backbone performance and parameter scale. However, HarDNet-68 and HarDNet-39 achieve $mAP$ values of 0.679 and 0.729, respectively, suggesting an inversely proportional relationship between backbone performance and parameter scale.

### 5.4.2 Quantitative and qualitative analyses of DIC-Transformer

We show the changes in various information during the model training process in the form of a line chart. Figure 7 shows the change process of learning rate during model training. Figure 8 shows the changes in the three losses in the training process, which are the loss changes in the image caption generation module, the loss changes in the image classification module, and the total loss changes in the model. Figure 3 shows the variation curves of various evaluation indicators of image classification and image subtitle generation results.

#### 5.4.2.1 DIC-Transformer image caption generation performance test

This is followed by a simple description of the eight image caption generation models to be compared. AoANet (Huang et al., 2019) introduces a new multi-level attention mechanism to enhance image caption generation by incorporating attention weighting in an attention weight. UpDown (Anderson et al., 2018) introduces underlying and top-level attention mechanisms, proposes trainable underlying features, and combines the attention mechanism with the language model. Adaatt (Lu et al., 2017) introduces visual flags to guide the allocation of attention, proposes an adaptive attention mechanism, and builds an end-to-end generative model. ShowTell (Vinyals et al., 2015) utilizes attention mechanisms and employs Multilayer Perceptron (MLP) as generators, combining CNN and LSTM. FC (Rennie et al., 2017) uses a CNN-encoded image as input to predict the second word. Instead of using a static spatial pool representation of the image, the attention model in Att2in (Dosovitskiy et al., 2020) dynamically reweights the input space (CNN) feature to focus on a specific area of the image at each time

TABLE 7 Faster R-CNN benchmark experiment (Detectron2 framework).

| Framework | Backbone | *mAP* | $mAP_{50}$ | $mAP_{75}$ |
|---|---|---|---|---|
| Faster R-CNN | HarDNet-68 | 0.472 | 0.679 | 0.514 |
| | HarDNet-39 | 0.469 | 0.729 | 0.506 |
| | EfficientNet-B4 | 0.544 | 0.781 | 0.640 |
| | EfficientNet-B0 | 0.532 | 0.740 | 0.618 |
| | MobileNetV2 | 0.511 | 0.734 | 0.600 |
| | VoVNet-19 | 0.625 | 0.792 | 0.733 |
| | VoVNet-99 | 0653 | 0.812 | 0.763 |
| | Swin Transformer | 0.674 | 0.862 | 0.793 |

step. $M^2$ Transformer (Cornia et al., 2020) integrates prior knowledge of learning, learns multi-level representations of relationships between image regions, and uses mesh connections to leverage low-level and high-level features during the decoding stage. DLCT (Luo et al., 2021) enhances contextual information and fine-grained details through a new bidirectional self-attention (DWSA) and a locally constrained cross-attention module. ExpansionNet v2 (Hu et al., 2022) explores the potential performance bottlenecks in input length in deep learning methods.

LATGeO (Dubey et al., 2023a) proposes a novel attention mechanism that is capable of efficiently processing geometrically related objects in images when generating image descriptions. This method not only considers global information but also pays attention to the spatial relationship between different objects in the image through a fine-grained local attention mechanism. The proposed framework, label-attention transformer with geometrically coherent objects (LATGeO (Dubey et al., 2023b)) acquires proposals of geometrically coherent objects using a deep neural network (DNN) and generates captions by investigating their relationships using label attention module (LAM). Stack-Captioning (Gu et al., 2018), a coarse-to-fine multi-stage prediction framework for image captioning, is composed of multiple decoders, each of which operates on the output of the previous stage, producing increasingly refined image descriptions. A new diffusion model-based paradigm tailored for image captioning is proposed, namely, Semantic-Conditional Diffusion Networks (SCD-Net; Luo et al., 2023), which breaks the deeply rooted conventions in learning Transformer-based encoder–decoder.

TABLE 8 Faster R-CNN benchmark experiment (MMDetection framework).

| Framework | Backbone | *mAP* | $mAP_{50}$ | $mAP_{75}$ |
|---|---|---|---|---|
| Faster R-CNN | ResNet-101 | 0.580 | 0.771 | 0.671 |
| | ResNet-50 | 0.585 | 0.788 | 0.694 |
| | Res2Net-101 | 0.582 | 0.772 | 0.682 |
| | RegNetX-3.2GF | 0.580 | 0.798 | 0.683 |
| | ResNeSt-50 | 0.525 | 0.746 | 0.597 |
| | ResNeSt-101 | 0.609 | 0.821 | 0.705 |
| | HrNet w-18 | 0.584 | 0.782 | 0.649 |
| | HrNet w-32 | 0.599 | 0.782 | 0.691 |

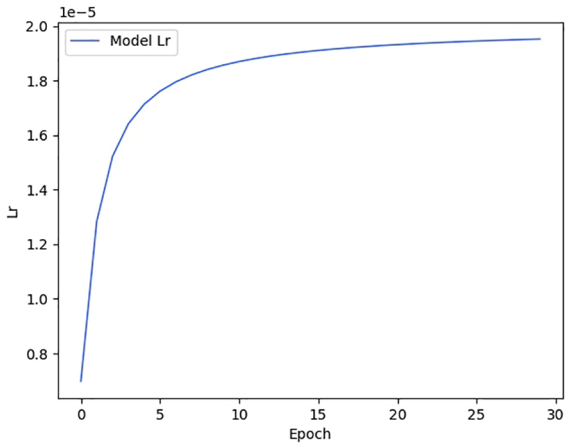



FIGURE 7
Model learning rate change curve.

Next, we test the performance of the DIC-Transformer and the above model on the image classification task based on ADCG-18. The criteria used to evaluate the model are Bleu1, Bleu2, Bleu3, Bleu4, CiderD, and Rough, which are common in the field of caption generation. Finally, Table 9 shows the test results of seven classic caption generation models and the DIC-Transformer model.

In this paper, there are seven comparison methods used for image caption generation tasks, which are roughly divided into two categories: one is the top-down method, and the other is a bottom-up approach. Experiments show no significant difference between top-down and bottom-up approaches. For example, the ShowTell method exhibits the worst results in the top-down method, whereas the Fc method performs the best effect, showing little difference from the optimal method in the bottom-up method. The model with the highest CiderD score is AoANet, with a value of 431.09. However, the model presented in this paper achieves a CiderD score

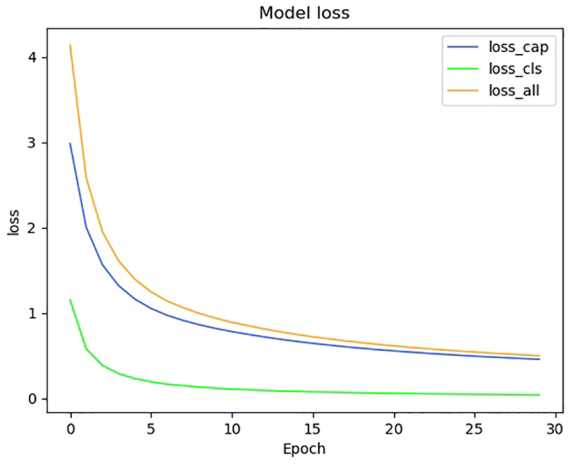



FIGURE 8
Model loss change curve.

TABLE 9 Comparison results of DIC-Transformer and classic image caption generation models.

| Method | Bleu1 | Bleu2 | Bleu3 | Bleu4 | CiderD | Rough |
|---|---|---|---|---|---|---|
| Fc | 0.746 | 0.535 | 0.364 | 0.249 | 420.96 | 0.707 |
| Att2in | 0.712 | 0.466 | 0.242 | 0.159 | 342.95 | 0.637 |
| DLCT | 0.721 | 0.502 | 0.348 | 0.293 | 434.54 | 0.664 |
| LATGeO | 0.682 | 0.472 | 0.305 | 0.214 | 335.26 | 0.636 |
| Stack-Captioning | 0.706 | 0.495 | 0.335 | 0.257 | 362.24 | 0.669 |
| UpDown | 0.679 | 0.489 | 0.316 | 0.228 | 383.40 | 0.657 |
| SCD-Net | 0.714 | 0.451 | 0.253 | 0.161 | 351.24 | 0.640 |
| AoANet | 0.727 | 0.539 | 0.353 | 0.280 | 431.09 | 0.677 |
| $M^2$ Transformer | 0.738 | 0.543 | 0.378 | 0.271 | 433.71 | 0.685 |
| ExpansionNet v2 | 0.741 | 0.549 | 0.386 | 0.287 | 437.09 | 0.708 |
| LATGeO | 0.739 | 0.531 | 0.363 | 0.278 | 432.12 | 0.701 |
| Ours | 0.756 | 0.561 | 0.404 | 0.294 | 450.51 | 0.721 |

of 450.51, which is 19.42 points higher than the CiderD score of AoANet.

#### 5.4.2.2 DIC-Transformer classification performance test

The purpose of this experiment is to demonstrate that DIC-Transformer is superior to other classification models in terms of classification performance. To compare with classical classification methods, DIC-Transformer is evaluated on ADCG-18, and the results are shown in Table 10. Under the context of error propagation (where the region detection module is trained solely on the training set), the classical CNN model with the best classification performance is MobilenetV2, achieving an Acc value of 0.830. In contrast, DIC-Transformer achieves an Acc value of 0.854, which is 0.024 higher than the Acc of MobilenetV2. In addition, in the absence of error propagation (where the region detection module is trained using both the training and testing sets), the DIC-Transformer's ACC value increases by an additional 0.031, resulting in an Acc value of 0.885. The results show that in the classification performance comparison experiment using the ADCG-18, DIC-Transformer outperforms other classical classification models, showing a better classification effect.

TABLE 10 Comparison results of DIC-Transformer and classical CNN models.

| Method | Acc | Recall | F1 |
|---|---|---|---|
| AlexNet, Krizhevsky et al. (2012) | 0.743 | 0.641 | 0.652 |
| ResNet-101 | 0.827 | 0.779 | 0.786 |
| EfficientNet | 0.814 | 0.775 | 0.778 |
| VggNet, Simonyan and Zisserman (2014) | 0.772 | 0.698 | 0.705 |
| GoogleNet, Szegedy et al. (2015) | 0.754 | 0.683 | 0.695 |
| InceptionV3, Szegedy et al. (2016) | 0.717 | 0.640 | 0.653 |
| MobileNetV2 | 0.830 | 0.776 | 0.778 |
| RepVGG, Ding et al. (2021) | 0.793 | 0.705 | 0.712 |
| GhostNet, Han et al. (2020) | 0.752 | 0.685 | 0.689 |
| SqueezeNet, Koonce and Koonce (2021) | 0.784 | 0.725 | 0.732 |
| Ours w/EP | 0.854 | 0.854 | 0.853 |
| Ours w/o EP | 0.885 | 0.885 | 0.886 |

#### 5.4.2.3 Ablation experiment of DIC-Transformer

In this set of ablation experiments, the main purpose is to verify the contribution of each module in the model. The modules involved in the experiment include the patch embedding module, the region detection module, the caption generation module, and the position encoding module. The results are shown in Table 11. In Experiments I and II, the region detection module is not used, and the image is divided into 16 * 16 and 32 * 32 grid input to the sequence encoding module. This method of dividing images comes from VIT (Dosovitskiy et al., 2020), called the PetchEmbedding method. In Experiment III, the visual feature vector outputted by the region detection module is used as the input to the sequence encoding module. The output of the sequence encoding module is then fed into the caption generation module to obtain the visual vector weighted by text features. Finally, this vector is used as the input of the classification module to obtain the category label of the disease. In addition, the positional coding module is used in the sequence encoding module. Experiment IV is based on Experiment III but without a position coding module. Experiment V removes the caption generation module from the setup in Experiment IV, and the visual feature vectors generated by the sequence encoding module are directly inputted to the classification module to classify the image. Experiments VI and VII remove the caption generation module from the setup in Experiments I and II, respectively. They directly input the visual feature vectors obtained by the segmented image processed by the sequence encoding module into the classification module to obtain the category label of the image.

TABLE 11 In the presence of error propagation, the effect of a single module on model performance results.

| Ablation | | | | | Image caption | | | | | | Classification | | | Index |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P16 | P32 | Detection | Describe | Position | Bleu1 | Bleu2 | Bleu3 | Bleu4 | CiderD | Rouge | Acc | Recall | F1 | |
| ✓ | | | ✓ | | 0.612 | 0.443 | 0.285 | 0.204 | 404.67 | 0.537 | 0.714 | 0.714 | 0.711 | I |
| | ✓ | | ✓ | | 0.632 | 0.487 | 0.313 | 0.224 | 413.25 | 0.583 | 0.786 | 0.786 | 0.782 | II |
| | | ✓ | ✓ | | 0.722 | 0.539 | 0.389 | 0.283 | 430.71 | 0.678 | 0.837 | 0.837 | 0.834 | III |
| | | ✓ | ✓ | ✓ | 0.756 | 0.561 | 0.404 | 0.295 | 450.51 | 0.721 | 0.854 | 0.854 | 0.853 | IV |
| | | ✓ | | ✓ | – | – | – | – | – | – | 0.846 | 0.846 | 0.844 | V |
| ✓ | | | | | – | – | – | – | – | – | 0.478 | 0.478 | 0.467 | VI |
| | ✓ | | | | – | – | – | – | – | – | 0.425 | 0.425 | 0.416 | VII |

The ✓ in the table means that the model in the experiment contains the current module, and the model in the experiment does not contain the current module without the ✓ symbol.

Experiments I, II, and III are conducted to verify the influence of the region detection module on the overall model performance, as shown in Table 11. Taking Bleu as an example, the average values of Experiments I and II are 0.612 and 0.632, respectively, while the average value of Experiment III is 0.722. The latter is 0.11 and 0.09 higher than the average values of the previous two experiments. This difference is due to the image meshing method used in Experiments I and II, which results in the divided images containing much background information. Additionally, Experiments I and II demonstrate that meshing the image into 32 * 32 works better than dividing it into 16 * 16 grids. Notably, location information is utilized in both experiments. Experiments III and IV evaluate the effect of the position coding module in the sequence encoding module on the overall model performance. In Experiment IV, the bounding box coordinates of the disease area in the image are used as location encoding. The mean value of Bleu1 in Experiment IV using position coding is 0.756, whereas the mean value of Bleu1 in Experiment III without position coding is 0.722, resulting in a difference of 0.034. This indicates that using bounding box coordinates as position coding has a positive impact on the model's performance. Experiments IV and V evaluate the influence of the caption generation module on classification performance. Taking Acc as an example, the classification result of the model containing the caption generation module is 0.854, while the classification result of the model without the caption generation module is 0.846. The results indicate that the introduction of the caption generation module to weigh visual feature vectors with text features can improve classification performance. In addition, in both experiments, location information is used. Experiments VI and VII remove the caption generation module based on Experiments I and II, respectively. The results show that the classification performance is significantly reduced, with the classification accuracy being 0.478 and 0.425, respectively.

Finally, the influence of error propagation on model performance in DIC-Transformer is quantitatively analyzed. In effect, all the images in the dataset are used as the training set for the first stage disease region detection module. This eliminates errors caused by the region detection module. Next, the trained disease area detector is used to obtain the feature information and disease area location details of the image. The ablation experiment is then repeated using the obtained data. The results of this experiment are summarized in Table 12. The mean accuracy in the final results of Experiments X, XI, and XII is 0.875, while Experiments VIII, IX, XIII, and XIV are consistent with the results presented in Table 11, as they do not involve the region detection module.

#### 5.4.2.4 Case study of the DIC-Transformer

To demonstrate the superiority of the proposed DIC-Transformer in the field of image caption generation, we compare it with the AoANet, AdaAtt, UpDown, and ShowTell models. The caption generation results of each model are shown in Figure 9. It

TABLE 12 In the absence of error propagation, the effect of a single module on model performance results.

| Ablation | | | | | Image caption | | | | | | Classification | | | Index |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P16 | P32 | Detection | Describe | Position | Bleu1 | Bleu2 | Bleu3 | Bleu4 | CiderD | Rouge | Acc | Recall | F1 | |
| ✓ | | | ✓ | | 0.612 | 0.443 | 0.285 | 0.204 | 404.67 | 0.537 | 0.714 | 0.714 | 0.711 | VIII |
| | ✓ | | ✓ | | 0.632 | 0.487 | 0.313 | 0.224 | 413.25 | 0.583 | 0.786 | 0.786 | 0.782 | IX |
| | | ✓ | ✓ | | 0.764 | 0.583 | 0.411 | 0.325 | 478.56 | 0.729 | 0.862 | 0.862 | 0.8 | X |
| | | ✓ | ✓ | ✓ | 0.7778 | 0.599 | 0.416 | 0.346 | 490.22 | 0.74 | 0.885 | 0.885 | 0.886 | XI |
| | | ✓ | | ✓ | – | – | – | – | – | – | 0.877 | 0.877 | 0.877 | XII |
| ✓ | | | | | – | – | – | – | – | – | 0.478 | 0.478 | 0.467 | XIII |
| | ✓ | | | | – | – | – | – | – | – | 0.425 | 0.425 | 0.416 | XIV |

The ✓ in the table means that the model in the experiment contains the current module, and the model in the experiment does not contain the current module without the ✓ symbol.




FIGURE 9
Comparison of caption generation results with real caption.

should be noted that the image caption dataset used in the experiment is Chinese, and the statements in the case study are translated into English for ease of understanding. The red part of the figure represents the wrong word, the blue part represents the missing word or the wrong combination of words and other inaccurate results, and the green part represents the correct disease keyword. Based on the four examples in Figure 9, we perform a meticulous analysis of the captions generated by each model, and the results are as follows:

1. As illustrated in Figure 9, the captions generated by the other four models lack relevant disease feature keywords. Specifically, the models AoANet and UpDown do not generate the color of the rings, the model AdaAttNet does not include any ring-related statements, and the model ShowTell omits the color of the disease area.
2. In the case of apple mold heart disease shown in Figure 9, the image caption generated by the model AoANet repeats the phrase "gray-green mold" twice, and the model AdaAttNet also repeats the word "mold" twice, both of which result in semantic confusion. Although both UpDown and ShowTell did not have duplicate issues, they both lacked the keyword "mold".
3. In the last image, the caption generated by models AOANet and UpDown lacks color information. The model AdaAttNet generates words with repetitive meanings in captions. The model ShowTell generates the wrong word "with".

However, the DIC-Transformer model we build generates comprehensive keywords and fluent sentences. The above experimental results can fully prove the superiority of the DIC-

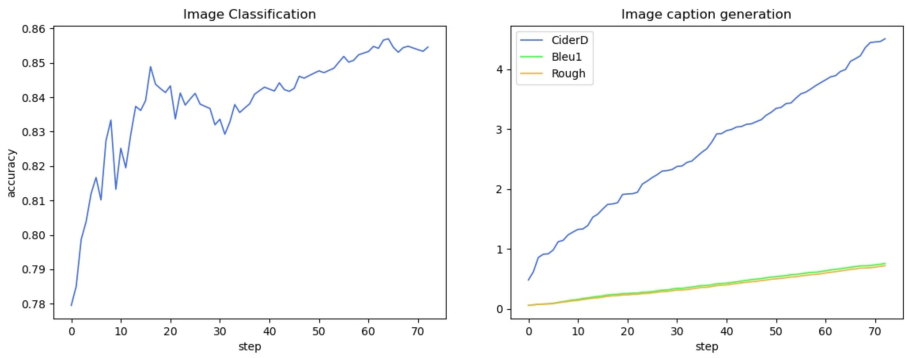



FIGURE 10
The change curve of evaluation indicators related to image classification and image caption generation.

Transformer in image caption generation. In order to more fully evaluate the quality of the generated statements, we invite experts or volunteers to perform a manual evaluation of the generated statements. This can be carried out through subjective scoring or preference sorting. By gathering opinions and feedback from evaluators, we can determine the advantages of the utterances generated by our model over other methods in terms of readability, fluency, and accuracy. In addition to expert assessments, we also conduct user surveys to gain a broader perspective. By inviting real users to rate and rank the generated statements, we can obtain more feedback that verifies the superiority of the statements generated by our model in terms of user experience. Through the above statistical analysis methods, we can objectively prove the superiority of the statements generated by our model over other methods.

#### 5.4.2.5 Summary of the comparison results between DIC-Transformer and other methods

In this paper, we conduct a total of three experiments, and the specific results of the experiments are as follows.

The first experiment is the Faster R-CNN benchmark experiment, which aims to select an optimal backbone for Faster R-CNN. Specifically, we select 16 different backbones based on the dataset to test the performance of Faster R-CNN in detecting disease regions, and the experimental results show Swin Transformer works best as backbones.

In the second experiment, the DIC-Transformer is compared with seven existing image caption generation models. The experimental results in Table 7 show that the model proposed by us based on ADCG-18 performs well in the field of image caption generation, and the performance is due to the other seven existing models. First, we consider the caption generation accuracy of the model. Compared to other models, DIC-Transformer can more accurately generate caption descriptions that match the content of the image. Second, we evaluate the language fluency and sentence quality of the model. The results show that the captions generated by DIC-Transformer are more natural and smooth, and the sentence structure is more reasonable and coherent. In contrast, other models may have some imperfections or non-conform to grammatical rules in terms of language expression.

The third experiment aims to test the classification performance of the DIC-Transformer based on ADCG-18. We compare the DIC-Transformer with seven classical CNN models, and the results show that the classification performance of the DIC-Transformer is better than that of other existing models. Figure 10 shows the variation curves of the evaluation indicators related to image classification and image caption.

In summary, our experimental results show that our proposed method has better performance than other methods in different tasks. This provides broad prospects for the application of our method in areas such as caption generation and image classification.

To evaluate the results of image caption generation, it is necessary to comprehensively consider the indicators of accuracy, completeness, semantic consistency, naturalness, and information richness and combine the experimental results for objective evaluation. The quantitative experimental results prove the accuracy of the model, and the indicators such as completeness and information richness of the generated sentences are also superior in the qualitative analysis section.

## 6 Conclusion

In this paper, we propose a model called DIC-Transformer, which is capable of generating explanatory textual descriptions corresponding to the classification results of agricultural plant diseases. Then, we collect a dataset called ADCG-18, which includes images of 18 diseases and their corresponding textual descriptions. We conduct numerous experiments on the proposed method from multiple angles on this constructed dataset to demonstrate the method's superiority. For the two challenges presented in the Introduction section, the experimental results indicate that the identification accuracy of DIC transformers reaches 85.4% in a relatively small sample, which is 2.4% higher than that of the model with the best performance based on CNN. This demonstrates that the DIC-Transformer effectively addresses the first challenge of enhancing image classification performance in the context of limited data volume. In response to the second challenge, we introduced technology for generating image captions. The technology can generate descriptive information based on images, similar to how agricultural experts describe plant diseases. The results of ablation experiments show that combining image caption generation and image classification technology can improve the accuracy of image classification. Based on ADCG-18, the results of the six evaluation indexes Bleu1, Bleu2, Bleu3, Bleu4, Cider-D, and Rouge of the proposed DIC-Transformer model are 0.756, 0.561, 0.404, 0.294, 450.51, and 0.721, respectively. These results indicate that the model outperforms other models.

Image caption generation technology can accurately describe the diseases affecting crops, enhance the precision of disease identification, aid in early detection and diagnosis of plant diseases, and minimize crop losses. It can also be integrated with other agricultural intelligence technologies, such as drones and sensors, to achieve automatic monitoring and management of farmland. This integration can enable farmers and agricultural practitioners to accurately identify diseases through image capture, even without expertise in disease identification. This will help reduce the barrier to identifying diseases and promote the adoption and application of agricultural technology.

However, the model we propose still has certain limitations. The DIC-Transformer is a two-stage model that suffers from the issue of error propagation, as depicted in Figure 10. For instance, a distinctive characteristic of peanut leaves is their green color. If the first-stage region detection module fails to recognize this aspect, the second-stage caption generation module will not generate a corresponding

caption. In our upcoming work, we intend to integrate the initial stage of the feature extraction module with the subsequent caption generation and classification modules to create a comprehensive model. This will help prevent some of the issues that arise with two-stage models and improve their performance.

## Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

## Author contributions

QZ: Funding acquisition, Writing – original draft, Writing – review & editing. JS: Methodology, Validation, Writing – original draft. SW: Investigation, Software, Writing – review & editing.

## Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work is supported by National Key R&D Program of China (2022ZD0119500 and 2022ZD0119501); NSFC (U1931207 and 61702306); Sci. & Tech. Development Fund of Shandong Province of China (ZR2022MF288); the Taishan Scholar Program of Shandong Province (ts20190936), and SDUST Research Fund (2015TDJH102 and 2019KJN024); Shandong Chongqing Science and technology cooperation project (cstc2020jscx-lyjsAX0008).

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

Al-Hiary, H., Bani-Ahmad, S., Reyalat, M., Braik, M., and Alrahamneh, Z. (2011). Fast and accurate detection and classification of plant diseases. *Int. J. Comput. Appl.* 17, 31–38. doi: 10.5120/2183-2754

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., et al. (2018). "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. Salt Lake City, UT, USA. 6077–6086. doi: 10.1109/CVPR.2018.00636

Bai, S., and An, S. (2018). A survey on automatic image caption generation. *Neurocomputing* 311, 291–304. doi: 10.1016/j.neucom.2018.05.080

Boulent, J., Foucher, S., Théau, J., and St-Charles, P.-L. (2019). Convolutional neural networks for the automatic identification of plant diseases. *Front. Plant Sci.* 10, 941. doi: 10.3389/fpls.2019.00941

Chao, P., Kao, C.-Y., Ruan, Y.-S., Huang, C.-H., and Lin, Y.-L. (2019). "HarDNet: A Low Memory Traffic Network," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 3552–3561. doi: 10.1109/ICCV.2019.00365

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., et al. (2019). MMDetection: Open mmlab detection toolbox and benchmark. *arXiv* e-prints, 2019. doi: 10.48550/arXiv.1906.07155

Cornia, M., Stefanini, M., Baraldi, L., and Cucchiara, R. (2020). "Meshed-memory transformer for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition ( CVPR)*, Seattle, WA, USA, 10578–10587. doi: 10.1109/CVPR42600.2020.01059

Deorukhkar, K., and Ket, S. (2022). A detailed review of prevailing image captioning methods using deep learning techniques. *Multimedia Tools Appl.* 81, 1313–1336. doi: 10.1007/s11042-021-11293-1

Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., and Sun, J. (2021). "Repvgg: Making vgg-style convnets great again," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 13733–13742. doi: 10.1109/CVPR46437.2021.01352

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv e-prints*, 2020. doi: 10.48550/arXiv.2010.11929

Dubey, S., Olimov, F., Rafique, M. A., Kim, J., and Jeon, M. (2023a). Label-attention transformer with geometrically coherent objects for image captioning. *Inf. Sci.* 623, 812–831. doi: 10.1016/j.ins.2022.12.018

Dubey, S., Olimov, F., Rafique, M. A., Kim, J., and Jeon, M. (2023b). Label-attention transformer with geometrically coherent objects for image captioning. *Inf. Sci.* 623, 812–831. doi: 10.1016/j.ins.2022.12.018

Fang, Z., Wang, J., Hu, X., Liang, L., Gan, Z., Wang, L., et al. (2022). "Injecting semantic concepts into end-to-end image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 18009–18019. doi: 10.1109/CVPR52688.2022.01748

Fei, Z. (2022). "Attention-aligned transformer for image captioning," *AAAI*, Vol. 36 (1), 607–615.

Gao, S.-H., Cheng, M.-M., Zhao, K., Zhang, X.-Y., Yang, M.-H., and Torr, P. (2019). Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 652–662. doi: 10.1109/TPAMI.2019.2938758

Ghandi, T., Pourreza, H., and Mahyar, H. (2023). Deep learning approaches on image captioning: A review. *ACM Computing Surveys* 56, 1–39. doi: 10.1145/3617592

Girshick, R. (2015). "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 1440–1448. doi: 10.1109/ICCV.2015.169

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 580–587. doi: 10.1109/CVPR.2014.81

Gu, J., Cai, J., Wang, G., and Chen, T. (2018). "Stack-Captioning: Coarse-to-Fine Learning for Image Captioning," in *Proceedings of the AAAI conference on artificial intelligence*, Vol. 1. 26.

Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., and Xu, C. (2020). "GhostNet: More features from cheap operations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 1580–1589. doi: 10.1109/CVPR42600.2020.00165

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 770–778. doi: 10.1109/CVPR.2016.90

Hodosh, M., Young, P., and Hockenmaier, J. (2013). Framing image description as a ranking task: Data, models and evaluation metrics. *J. Artif. Intell. Res.* 47, 853–899. doi: 10.1613/jair.3994

Hu, J. C., Cavicchioli, R., and Capotondi, A. (2022). ExpansionNet v2: Block static expansion in fast end to end training for image captioning. *arXiv*. doi: 10.48550/arXiv.2208.06551

Huang, L., Wang, W., Chen, J., and Wei, X.-Y. (2019). "Attention on attention for image captioning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 4634–4643. doi: 10.1109/ICCV.2019.00473

Jeong, J., Tian, K., Li, A., Hartung, S., Behzadi, F., Calle, J., et al. (2023). Multimodal image-text matching improves retrieval-based chest x-ray report generation. *arXiv* e-prints, 2023. doi: 10.48550/arXiv.2303.17579

Jing, L., and Jin-guang, C. (2023). Image caption generation method based on channel attention and transformer. *Comput. Modernization* 0(05), 8–12.

Keshari, R., Vatsa, M., Singh, R., and Noore, A. (2018). "Learning structure and strength of cnn filters for small sample size training," in *proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 9349–9358. doi: 10.1109/CVPR.2018.00974

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., and Shah, M. (2022). Transformers in vision: A survey. *ACM computing surveys (CSUR)* 54, 1–41. doi: 10.1145/3505244

Koonce, B., and Koonce, B. (2021). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. *arXiv* e-prints. doi: 10.48550/arXiv.1602.07360

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60 (6), 84–90.

Kulkarni, G., Premraj, V., Ordonez, V., Dhar, S., Li, S., Choi, Y., et al. (2013). Babytalk: Understanding and generating simple image descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 2891–2903. doi: 10.1109/TPAMI.2012.162

Kumar, C., and Kumar, V. (2023). "Vegetable plant leaf image classification using machine learning models," in *Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems. Lecture Notes in Networks and Systems*, vol 612. Springer, Singapore.

Lee, Y., Hwang, J.-W., Lee, S., Bae, Y., and Park, J. (2019). "An energy and gpu-computation efficient backbone network for real-time object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Long Beach, CA, USA, 752–760. doi: 10.1109/CVPRW.2019.00103

Li, D., Sun, X., Jia, Y., Yao, Z., Lin, P., Chen, Y., et al. (2023). A longan yield estimation approach based on uav images and deep learning. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1132909

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. (2016). "SSD: Single Shot MultiBox Detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. 21–37 (Springer).

Liu, W., Chen, S., Guo, L., Zhu, X., and Liu, J. (2021a). Cptr: Full transformer network for image captioning. *arXiv* e-prints, 2021. doi: 10.48550/arXiv.2101.10804

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021b). "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 10012–10022. doi: 10.1109/ICCV48922.2021.00986

Liu, C., Zhao, R., and Shi, Z. (2022). Remote-sensing image captioning based on multilayer aggregated transformer. *IEEE Geosci. Remote Sens. Lett.* 19, 1–5. doi: 10.1109/LGRS.2022.3150957

Lu, J., Xiong, C., Parikh, D., and Socher, R. (2017). "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 375–383. doi: 10.1109/CVPR.2017.345

Luo, Y., Ji, J., Sun, X., Cao, L., Wu, Y., Huang, F., et al. (2021). "Dual-level collaborative transformer for image captioning," in *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35, 2286–2293. doi: 10.48550/arXiv.2101.06462

Luo, J., Li, Y., Pan, Y., Yao, T., Feng, J., Chao, H., et al. (2023). "Semantic-conditional diffusion networks for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*., 23359–23368. doi: 10.48550/arXiv.2212.03099

Luo, C., Li, X., Wang, L., He, J., Li, D., and Zhou, J. (2018). "How does the data set affect cnn-based image classification performance?," in *2018 5th international conference on systems and informatics (ICSAI)*, Nanjing, China, 361–366 (IEEE). doi: 10.1109/ICSAI.2018.8599448

Naga Srinivasu, P., Krishna, T. B., Ahmed, S., Almusallam, N., Khaled Alarfaj, F., Allheeib, N., et al. (2023). Variational autoencoders-basedself-learning model for tumor identification and impact analysis from 2-d mri images. *J. Healthcare Eng.* 2023, 1566123.

Ngugi, L. C., Abelwahab, M., and Abo-Zahhad, M. (2021). Recent advances in image processing techniques for automated leaf pest and disease recognition–a review. *Inf. Process. Agric.* 8, 27–51. doi: 10.1016/j.inpa.2020.04.004

Ondeng, O., Ouma, H., and Akuon, P. (2023). A review of transformer-based approaches for image captioning. *Appl. Sci.* 13, 11103. doi: 10.3390/app131911103

Ordonez, V., Han, X., Kuznetsova, P., Kulkarni, G., Mitchell, M., Yamaguchi, K., et al. (2016). Large scale retrieval and generation of image descriptions. *Int. J. Comput. Vision* 119, 46–59. doi: 10.1007/s11263-015-0840-y

Parvin, H., Naghsh-Nilchi, A. R., and Mohammadi, H. M. (2023). Transformer-based local-global guidance for image captioning. *Expert Syst. Appl.* 223, 119774. doi: 10.1016/j.eswa.2023.119774

Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., and Dollár, P. (2020). "Designing network design spaces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 10425–10433. doi: 10.1109/ CVPR42600.2020.01044

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (6), 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. (2017). "Self-critical sequence training for image captioning," arXiv e-prints, 2016. doi: 10.48550/arXiv.1612.00563

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv* e-prints, 2017. doi: 10.48550/arXiv.1706.05098

Rzanny, M., Wittich, H. C., Mäder, P., Deggelmann, A., Boho, D., and Wäldchen, J. (2022). Image-based automated recognition of 31 poaceae species: The most relevant perspectives. *Front. Plant Sci.* 12, 3315. doi: 10.3389/fpls.2021.804140

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). "Mobilenetv2: Inverted residuals and linear bottlenecks," *arXiv* e-prints, 2018. doi: 10.48550/arXiv.1801.04381

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv* e-prints, 2014. doi: 10.48550/arXiv.1409.1556

Singh, V., Chug, A., and Singh, A. P. (2023). Classification of beans leaf diseases using fine tuned cnn model. *Proc. Comput. Sci.* 218, 348–356. doi: 10.1016/j.procs.2023.01.017

Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., and Batra, N. (2020). "PlantDoc: A dataset for visual plant disease detection," arXiv e-prints, 2019. doi: 10.48550/arXiv.1911.10317

Sun, X., Li, G., Qu, P., Xie, X., Pan, X., and Zhang, W. (2022). Research on plant disease identification based on cnn. *Cogn. Robotics* 2, 155–163. doi: 10.1016/j.cogr.2022.07.001

Sun, K., Xiao, B., Liu, D., and Wang, J. (2019). "Deep high-resolution representation learning for human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 5686–5696. doi: 10.1109/CVPR.2019.00584

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). "Going deeper with convolutions," arXiv e-prints, 2014. doi: 10.48550/arXiv.1409.4842

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Tan, M., and Le, Q. (2019). "EfficientNet: Rethinking Model Scaling for convolutional neural networks," in *International conference on machine learning*. 6105–6114 (PMLR). doi: 10.48550/arXiv.1905.11946

Torralba, A., Russell, B. C., and Yuen, J. (2010). Labelme: Online image annotation and applications. *Proc. IEEE* 98, 1467–1484. doi: 10.1109/JPROC.2010.2050290

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 3156–3164. doi: 10.1109/CVPR.2015.7298935

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2023). "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, BC, Canada, 7464–7475. doi: 10.1109/CVPR52729.2023.00721

Wang, Q., Wan, J., and Chan, A. B. (2020). On diversity in image captioning: Metrics and methods. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 1035–1049. doi: 10.1109/TPAMI.2020.3013834

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. (2019) *Detectron2*. Available at: https://github.com/facebookresearch/detectron2.

Xian, T., Li, Z., Tang, Z., and Ma, H. (2022). Adaptive path selection for dynamic image captioning. *IEEE Trans. Circuits Syst. Video Technol.* 32, 5762–5775. doi: 10.1109/TCSVT.2022.3155795

Yang, X., Liu, Y., and Wang, X. (2022). "ReFormer: The relational transformer for image captioning," arXiv e-prints, 2021. doi: 10.48550/arXiv.2107.14178

Yang, M., Zhao, W., Xu, W., Feng, Y., Zhao, Z., Chen, X., et al. (2018). Multitask learning for cross-domain image captioning. *IEEE Trans. Multimedia* 21, 1047–1061. doi: 10.1109/TMM.2018.2869276

Ye, S., Han, J., and Liu, N. (2018). Attentive linear transformation for image captioning. *IEEE Trans. Image Process.* 27, 5514–5524. doi: 10.1109/TIP.2018.2855406

Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., et al. (2022). "ResNeSt: split-attention networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, New Orleans, LA, USA, 2736–2746. doi: 10.1109/CVPRW56347.2022.00309

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 23 February 2024
DOI 10.3389/fpls.2024.1269423

Check for updates

OPEN ACCESS

EDITED BY
Po Yang,
The University of Sheffield, United Kingdom

REVIEWED BY
Jun Liu,
Weifang University of Science and Technology, China
Guoxiong Zhou,
Central South University Forestry and Technology, China
Jakub Nalepa,
Silesian University of Technology, Poland

*CORRESPONDENCE
Qiudong Yu
624432360@qq.com

RECEIVED 30 July 2023
ACCEPTED 07 February 2024
PUBLISHED 23 February 2024

CITATION
Liu Y, Yu Q and Geng S (2024) Real-time and lightweight detection of grape diseases based on Fusion Transformer YOLO.
*Front. Plant Sci.* 15:1269423.
doi: 10.3389/fpls.2024.1269423

COPYRIGHT
© 2024 Liu, Yu and Geng. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# Real-time and lightweight detection of grape diseases based on Fusion Transformer YOLO

Yifan Liu, Qiudong Yu* and Shuze Geng

College of Information Technology Engineering, Tianjin University of Technology and Education, Tianjin, China

**Introduction:** Grapes are prone to various diseases throughout their growth cycle, and the failure to promptly control these diseases can result in reduced production and even complete crop failure. Therefore, effective disease control is essential for maximizing grape yield. Accurate disease identification plays a crucial role in this process. In this paper, we proposed a real-time and lightweight detection model called Fusion Transformer YOLO for 4 grape diseases detection. The primary source of the dataset comprises RGB images acquired from plantations situated in North China.

**Methods:** Firstly, we introduce a lightweight high-performance VoVNet, which utilizes ghost convolutions and learnable downsampling layer. This backbone is further improved by integrating effective squeeze and excitation blocks and residual connections to the OSA module. These enhancements contribute to improved detection accuracy while maintaining a lightweight network. Secondly, an improved dual-flow PAN+FPN structure with Real-time Transformer is adopted in the neck component, by incorporating 2D position embedding and a single-scale Transformer Encoder into the last feature map. This modification enables real-time performance and improved accuracy in detecting small targets. Finally, we adopt the Decoupled Head based on the improved Task Aligned Predictor in the head component, which balances accuracy and speed.

**Results:** Experimental results demonstrate that FTR-YOLO achieves the high performance across various evaluation metrics, with a mean Average Precision (mAP) of 90.67%, a Frames Per Second (FPS) of 44, and a parameter size of 24.5M.

**Conclusion:** The FTR-YOLO presented in this paper provides a real-time and lightweight solution for the detection of grape diseases. This model effectively assists farmers in detecting grape diseases.

KEYWORDS

grape diseases detection, YOLO, transformer, lightweight, real-time

# 1 Introduction

China's extensive agricultural heritage, spanning over 2000 years, encompasses grape cultivation. Not only it is a significant grape-producing nation but it also stands as the largest exporter of grapes worldwide. Grapes are not only consumed directly but are also processed into various products such as grape juice, raisins, wine, and other valuable commodities, thus holding substantial commercial value (El-Saadony et al., 2022). However, during the grape growth process, susceptibility to diseases can lead to reduced grape yield and significant economic losses (Elnahal et al., 2022). Hence, the timely and effective detection of grape diseases is crucial for ensuring healthy grape growth. Conventionally, the diagnosis of grape diseases predominantly relies on field inspections by agricultural experts (Liu et al., 2022; Ahmad et al., 2023). This approach incurs high costs, has a lengthy cycle, and lacks operational efficiency.

The development of computer vision and machine learning technology provides a new solution for real-time automatic detection of crop diseases (Fuentes et al., 2018, 2019). Traditional machine learning methods in crop diseases identification and positioning have made some valuable experience, such as image segmentation [such as K-means clustering (Trivedi et al., 2022) and threshold method (Singh and Misra, 2017)], feature detection [such as SURF (Hameed and Üstündağ, 2020), KAZE (Rathor, 2021), and MSER blob (Lee et al., 2023)], and pattern recognition [such as KNN (Balakrishna and Rao, 2019), SVM, and bp neural network (Hatuwal et al., 2021; Kaur and Singh, 2021)]. Due to the complexity of image preprocessing and feature extraction, these methods are still ineffective in detection.

Deep learning can automatically learn the hierarchical features of different disease regions without manual design of feature extraction and classifier, with excellent generalization ability and robustness. The detection of crop diseases through CNN has become a new hotspot in intelligent agriculture research. Jiang et al. (2019) proposed a novel network architecture invar-SSD based on VGG-Net and inception to the detection of apple leaf diseases, mAP reached 78.8%. Yang et al. (2023) proposed a SE-VGG16 model uses VGG16 as the basis and adds the SE attention, which classified corn weeds with an average accuracy of 99.67%. Guan et al. (2023) proposed a dise efficient based on the EfficientNetV2 model, achieved an accuracy of 99.80% on the plant disease and pest dataset. The above three methods are merely applicable for simple classification tasks. However, when it comes to detection tasks, the prevailing approach currently in use is YOLO. Liu and Wang (2020) proposed an improved YOLOv3 algorithm to detect tomato diseases and insect pests. Results show that the detection accuracy is 92.39%, and the detection time is 20.39 ms. Wang et al. (2022) proposed a lightweight model based on the improved YOLOv4 to detect dense plums in orchards. Compared with YOLOv4 model, the model size is compressed by 77.85%, the parameters are only 17.92%, and the speed is accelerated by 112%. Kuznetsova et al. (2020) designed harvesting robots based on a YOLOv3 algorithm, apple detection time averaged 19 ms with 90.8% recall, and 7.8% False Positive Rate (FPR). Qi et al. (2021) proposed a highly fused, lightweight detection model named the Fusion-YOLO model to detect the early flowering stage of tea chrysanthemum. Huang et al. (2021) used the YOLOv5 algorithm to detect the citrus collected by UAV, the detection accuracy rate was 93.32%. Qiu et al. (2022) used YOLOv5 for detecting citrus greening disease. The F1 scores for recognizing five symptoms achieved 85.19%. Zhou et al. (2022) proposed an improved YOLOX-s algorithm. Compared with the original YOLOX-s, the model improved the detection Average precision (AP) of kiwifruit by 6.52%, reduced the number of parameters by 44.8% and upgraded the model detection speed by 63.9%. Soeb et al. (2023) used YOLOv7 for five tea leaf diseases in natural scene, which validated by detection accuracy 97.3%, precision 96.7%, recall 96.4%, mAP 98.2%, and F1-score 0.965, respectively.

The application of machine learning and deep learning in crop disease detection in recent years is summarized. Deep learning, especially CNN, has also made some contributions to grape disease detection. Ji et al. (2020) designed the United Model and selected 1,619 images of healthy and three kinds of diseased grape leaves in Plant village, with detection accuracy up to 98.57%. However, it should be noted that all the data were obtained from laboratory samples, and no comparative experiments were conducted in a natural environment. Sanath Rao et al. (2021) used a pre-trained AlexNet to classify grapes and mango leaf diseases, achieved accuracy of 99% and 89% for grape leaves and mango leaves, respectively. Ji et al. (2020) proposed a united CNN architecture based on InceptionV3 and ResNet50 and can be used to classify grape images into four classes, achieved average validation accuracy of 99.17% and test accuracy of 98.57%. Adeel et al. (2022) proposed a entropy-controlled CNN to identify grape leaf diseases at the early stages, achieved an accuracy of 99%. Lu et al. (2022) proposed a Ghost-conv. and Transformer networks for diagnosing 11 classes grape leaf and pest, reached 180 frames per second (FPS), 1.16 M weights and 98.14% accuracy. After adding Transformer and Ghost-conv., the performance is improved significantly, but only the identification work is done. Xie et al. (2020) presented a Faster DR-IACNN model with higher feature extraction capability, achieved a precision of 81.1% mAP, and the detection speed reaches 15.01 FPS. The above two methods only detect grape leaf diseases. Sozzi et al. (2022) evaluated six versions of the YOLO (YOLOv3, YOLOv3-tiny, YOLOv4, YOLOv4-tiny, YOLOv5x, and YOLOv5s) for real-time bunch detection and counting in grapes. Pinheiro et al. (2023) presented three pre-trained YOLO models (YOLOv5x6, YOLOv7-E6E, and YOLOR-CSP-X) to detect and classify grape bunches as healthy or damaged by the number of berries with biophysical lesions, highlighting YOLOv7 with 77% of mAP and 94% of the F1-score. Both of the aforementioned methods solely utilized YOLO for grape bunch detection and did not involve disease detection. Zhu et al. (2021) proposed YOLOv3-SPP network for detection of black rot on grape leaves, applied in field environment with 86.69% precision and 82.27% recall. Zhang Z. et al. (2022) proposed a YOLOv5-CA, which highlights the downy mildew disease–related visual features to achieve an mAP of 89.55%. Both methods employed YOLO for the detection of a single disease in grapes. We have listed the advantages and disadvantages of different methods for plant disease detection in Table 1.

TABLE 1 Comparison of the advantages and disadvantages of different methods.

| Method | Advantage | Disadvantage |
|---|---|---|
| Machine learning | - Less data and computing resources.<br>- High interpretability. | - Difficult to handle complex problems.<br>- Poor detection accuracy. |
| Deep learning (classification) | - The model can automatically learn image feature representations.<br>- High detection accuracy. | - More data and computing resources.<br>- The task is relatively simple; the practical value is limited. |
| Deep learning (detection) | - Higher accuracy and generalization hold significant practical value.<br>- End-to-end, one-stage models (YOLO) are easy to implement. | - Additional data, annotations, and computing resources are necessary.<br>- Balancing detection accuracy and speed is a challenging task. |

There are also several challenges in grape disease detection: (1) grape fruits and inflorescence are small and dense, making it difficult to detect the incidence area, which can be very small. (2) Photos taken in natural scenes are susceptible to external interference. (3) The model needs to balance detection accuracy with lightweight requirements for deployment and real-time performance. To address these challenges, this paper proposes a real-time detection model based on Fusion Transformer YOLO (FTR-YOLO) for grape diseases. The main contributions of this paper are summarized as follows:

- Regarding the issue of limited detection of disease types in other models and the detection under non-natural environments, we have collected four grape diseases (anthracnose, grapevine white rot, gray mold, and powdery mildew) datasets in natural environments, covered different parts such as leaves, fruits, and flower inflorescence. The primary source of the dataset comprises RGB images acquired from plantations situated in North China.
- In backbone, we integrate learnable downsampling layer (LDS), effective squeeze and excitation (eSE) blocks, and residual connections based on VoVnet, effectively improving the ability of network to extract feature information. In neck component, an improved real-time Transformer with two-dimensional (2D) position embedding and single-scale Transformer encoder (SSTE) are incorporated to the last feature map to accurate detection of small targets. In head component, the Decoupled Head based on the improved Task-Aligned Predictor (ITAP) is adopted to optimize detection accuracy.
- To address the challenges with deploying application using models that have a large capacity and slow inference speed, we replace the convolution with ghost module in the model, abandon Transformer decoder, and adopt more efficient SSTE with VoVnet-39 of fewer layers to ensure the lightweight and detection speed.

The rest of the article is organized as follows: **Section 2** explicates the datasets and experimental settings and the network architecture and improvement of FTR-YOLO. **Section 3** presents the evaluation of the experimental performance and analyses. Discussions of the performance are presented in **Section 4**. Last, **Section 5** offers conclusions and suggestions for future work.

# 2 Materials and methods

## 2.1 Experimental dataset building

In the process of building grape diseases detection dataset, smartphone is used to collect photos in the local orchard. The photos are taken in different time periods, weather conditions, and scenes. The labeling tool is used to mark the images, the region of interest by manually marking the rectangle, and then generated the configuration file automatically.

Data augmentation is employed to expand the number of images within the training dataset. The methods include random flipping, Gaussian blur, affine transformation, image interception, filling, and so forth. The network model is designed to enhance randomly selected images by one or several operations.

The number of samples for each category is shown in Table 2. Through data enhancement, the dataset is expanded to 4,800 images. The ratio of training set and test set is 8:2.

The overall structure of FTR-YOLO is shown in Figure 1. The primary innovations of the model are represented by streamlined modules. For comprehensive details, please consult the detailed illustrations provided in **Sections 2.2–2.4**.

## 2.2 Backbone of FTR-YOLO

In backbone component, a lightweight high-performance VoVnet (LH-VoVNet) (Zhao et al., 2022) network is used. The proposed net adds the LDS Layer, eSE attention module (Long et al., 2020) and residual connection on the basis of One-Shot Aggregation (OSA) module. Also, the Conv. layer is replaced with Ghost Module (Zhang B. et al., 2022) to further lightweight the network. The LH-VoVNet has shorter computation time and higher detection accuracy compared with other common backbone, which is more suitable for grape disease detection tasks.

TABLE 2 The number of samples for each disease type.

| Disease | Sample size | Number of labeled samples (bounding box) | Percent of bounding box samples |
|---|---|---|---|
| Anthracnose | 1200 | 4587 | 20.53% |
| White rot | 1200 | 6025 | 26.97% |
| Gray moid | 1200 | 5160 | 23.09% |
| Powdery mildew | 1200 | 6571 | 29.41% |
| Total | 4800 | 22343 | 100% |

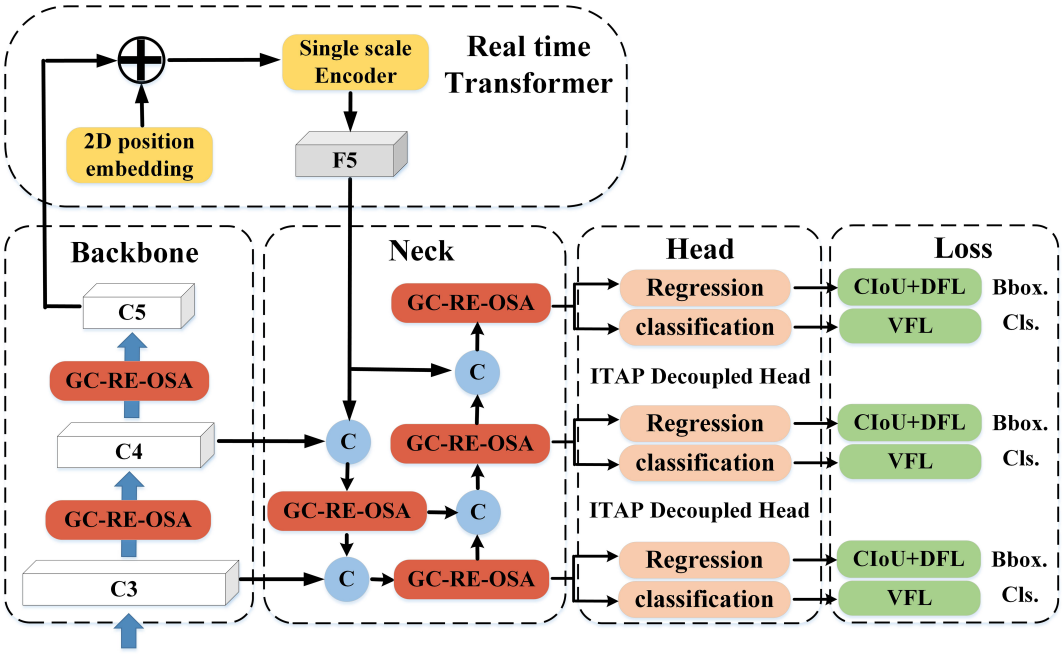



FIGURE 1
The architecture of FTR-YOLO.

### 2.2.1 VoVNet

One of the challenges with DenseNet (Jianming et al., 2019) is that the dense connections can become overly cumbersome. Each layer aggregates the features from the preceding layers, leading to feature redundancy. Furthermore, based on the L1 norm of the model weights, it is evident that the middle layer has minimal impact on the final classification layer, as shown in Figure 2A. Instead, this information redundancy is a direction that can be optimized, so the OSA module is adopted, as shown in Figure 2B. Simply put, the OSA aggregates all the layers up to the final one, effectively addressing the prior issue encountered with DenseNet. Since the number of input channels per layer is fixed, the number of output channels can be consistent with the input to achieve the minimum MAC, and the 1 × 1 Conv. layer is no longer required to compress features, the OSA module is computationally efficient.

### 2.2.2 LDS layer

At present, in common networks, the steps of downsampling feature maps are usually completed at the first Conv. of each stage. Figure 3A shows the general Residual block. In Path A, once the input data are received, it undergoes a 1 × 1 Conv. with a stride of 2. This operation leads to a loss of 3/4 of the information in the input feature maps.

To solve this problem, the LDS layer is adopted. The downsampling is moved to the following 3 × 3 Conv. in Path A, and the identity part (Path B) downsampling is done by the added

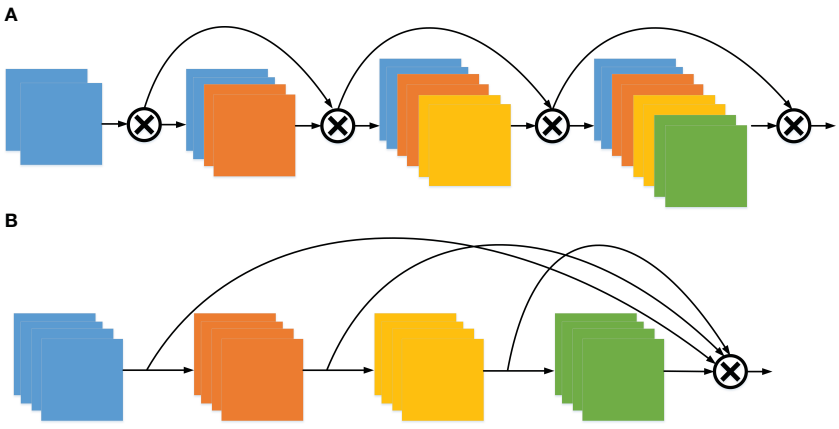



FIGURE 2
The architecture of DenseNet and VoVNet. **(A)** Dense aggregation (DenseNet) and **(B)** One-shot aggregation (VoVNet).

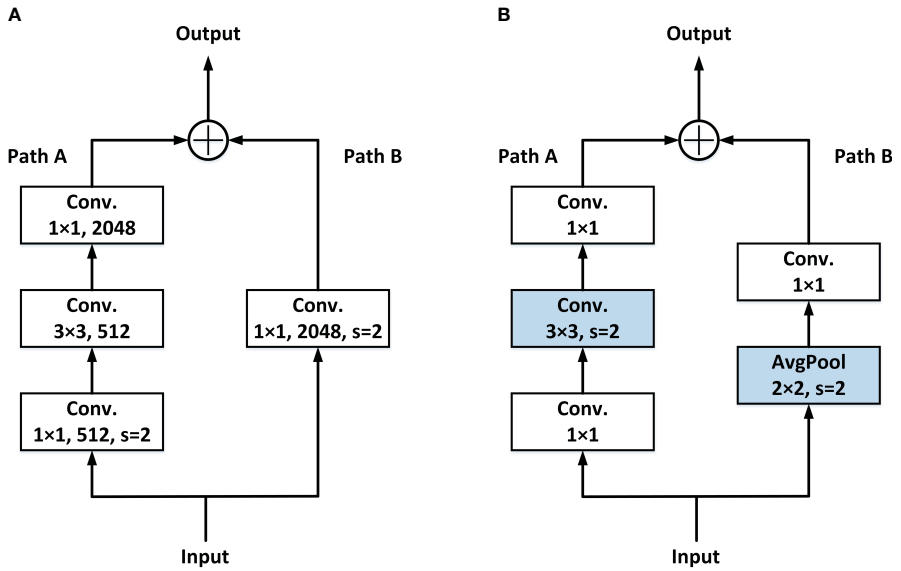



FIGURE 3
Two different methods of downsampling. **(A)** Conv. downsampling and **(B)** LDS downsampling.

avg-pool, so as to avoid the loss of information caused by the simultaneous appearance of 1 × 1 Conv. and stride. Details are shown in Figure 3B.

### 2.2.3 RE-OSA module

The pivotal element of the VoVnet lies in the OSA module as described in **Section 2.2.1**. While the performance of the OSA module is not enhanced, it offers lower MAC and improved computational efficiency. Therefore, this paper adds eSE block and residual connection in OSA module to further enhance features and improve detection accuracy, called RE-OSA module.

The core idea of SE Block is to learn the feature weight according to loss through the network (Hu et al., 2018), so that the effective feature map has a larger weight and the rest of the feature map has a smaller weight to train the model to achieve better results. The SE module squeezes the entire spatial features on a channel into a global feature by global average pooling, then two fully connected (FC) layers are used to concat the feature map information of each channel. Assume that the input feature map $X_i \in R^{C\times W\times H}$, the channel attention map $A_{ch}(X_i) \in R^{C\times 1\times 1}$ is computed in Equations 1, 2.

$$A_{ch}(X_i) = \sigma(W_c(\delta(W_{c/r}(\mathcal{F}_{gap}(X_i))))) \tag{1}$$

$$\mathcal{F}_{gap}(X) = \frac{1}{WH}\sum_{i,j=1}^{W,H} X_{i,j} \tag{2}$$

Where $\mathcal{F}_{gap}$ is channel-wise global average pooling, $W_{c/r}$, $W_c \in R^{C\times 1\times 1}$ are weights of two FC layers, σ denotes ReLU activation function, δ denotes sigmoid activation function.

In SE block, to avoid the computational burden of such a large model, reduction ratio $r$ is used in the first FC layer to reduce the input feature channels from $c$ to $c/r$. The second FC layer needs to expand the reduced number of channels to the original channel $c$. In this process, the reduction of channel dimensions leads to the loss of channel information.

Therefore, we adopt eSE that uses only one FC layer with $c$ channels instead of two FC layers without channel dimension reduction, which rather maintains channel information and in turn improves performance. In this paper, the ReLU/sigmoid activation function in the module is replaced by the SiLU function with better performance in YOLOv7 (Wang et al., 2023). The eSE is computed in Equations 3, 4:

$$A_{eSE} = \vartheta(W_c(\mathcal{F}_{gap}(X_i))) \tag{3}$$

$$X_{refine} = A_{eSE} \otimes X_i \tag{4}$$

where ϑ denotes SiLU activation function. As a channel attentive feature descriptor, the $A_{eSE} \in R^{C\times 1\times 1}$ is applied to the diversified feature map $X_i$ to make the diversified feature more informative. Finally, the refined feature map $X_{refine}$ is obtained by channel-wise multiplication $A_{eSE}$ and $X_i$.

### 2.2.4 Lightweight with ghost convolution

It can be seen from **Section 2.2.3** that Conv. layer appears most frequently in VoVNet. As a result, the whole network has a large amount of computation and parameter volume, which is not conducive to lightweight deployment.

To solve this problem, this paper adopts a structure—Ghost Module, which can generate a large number of feature graphs with cheap operations. This method can reduce the amount of computation and parameter volume on the basis of ensuring the performance ability of the algorithm.

In the feature map extracted by the mainstream deep neural networks, the rich and even redundant information usually ensures a comprehensive understanding of the input data. These redundancies are called ghost maps.

The ghost module consists of two parts. One part is the feature map generated by the ordinary Conv. The other part is the ghost maps generated by simple linear operation Φ. It is assumed that the input feature map of size $h \times w \times c$ is convolved with n sets of kernels of size $k \times k$, and the output feature map of size $h' \times w' \times n$. In the ghost model, m groups of $k \times k$ kernels are convolved with input to generate the identity maps of size $h' \times w' \times m$, after which the identity maps are linearly transformed by depth-wise convolution ($k$=5) to produce ghost maps. Finally, identity maps are concated with ghost maps to generate ghost convolution. The ghost convolution acceleration ratio $r_s$ and compression ratio $r_c$ are calculated compared with ordinary convolution, as shown in Equations 5, 6.

$$r_s = \frac{n \cdot h\prime \cdot w\prime \cdot c \cdot k \cdot k}{\frac{n}{s} \cdot h\prime \cdot w\prime \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot h\prime \cdot w\prime \cdot c \cdot d \cdot d} \approx s \quad (5)$$

$$r_c = \frac{n \cdot c \cdot k \cdot k}{\frac{n}{s} \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot c \cdot d \cdot d} \approx s \quad (6)$$

where the numerator is the complexity of ordinary convolution. The denominator is the complexity of ghost module. $s$ is the total mapping generated by each channel (one identity map and s-1 ghost maps), $c$ is the number of input feature maps, generally $s \ll c$; $n/s$ refers to the identity map output by general convolution; $d \times d$ is the average kernel size of depth-wise Conv. and has a similar size to $k \times k$.

Equations 5, 6 show that, compared with ordinary Conv., Ghost-conv. greatly reduces the amount of computation and the number of parameters.

Finally, GC-RE-OSA module replaced 3 × 3 Conv. in RE-OSA module (**Section 2.2.3**) with Ghost-conv. The structure of GC-RE-OSA is shown in Figure 4.

The specific structure of LH-VoVnet can be found in Table 3. LH-VoVNet comprises a stem block that consists of three 3 × 3 Conv. layers, followed by GC-RE-OSA modules implemented in four stages. At the start of each stage, an LDS with a stride of 2 is utilized (**Section 2.2.2**). The model achieves a final output stride of 32. For more details, please refer to **Sections 2.2.3** and **2.2.4**.

## 2.3 Neck of FTR-YOLO

Indeed, the Transformer model relies on a global attention mechanism that requires substantial computational resources for optimal performance (Carion et al., 2020). Consequently, it becomes crucial to address this issue effectively. To mitigate this concern, we eschew the initial image or multi-layer feature maps as input and instead incorporate only the final feature map obtained from the backbone. This is then directly connected to the neck. Additionally, we select only two improved modules of Position Embedding and Encoder.

Within the neck component, we utilize the current optimal dual-flow PAN + FPN structure and enhance it through integration with the GC-RE-OSA module introduced in this paper.

### 2.3.1 Real-time transformer

To enhance the detection accuracy, an enhanced global attention mechanism based on the Vision Transformer (ViT) is introduced. This modification takes into consideration that some grape diseases may share similarities, while others have limited occurrence areas. By incorporating this improved global attention mechanism, the detection accuracy can be further improved in detecting different grape diseases.

The current common detection transformer (DETR) algorithms extract the last three layers of feature maps (C3, C4, and C5) from the backbone network as the input. However, this approach usually has two problems:

1. Previous DETRs, such as deformable DETR (Zhu et al., 2020), flatten multi-scale features, and concatenate them into a single long-sequence vector. This approach not only enables effective interaction between the different scale

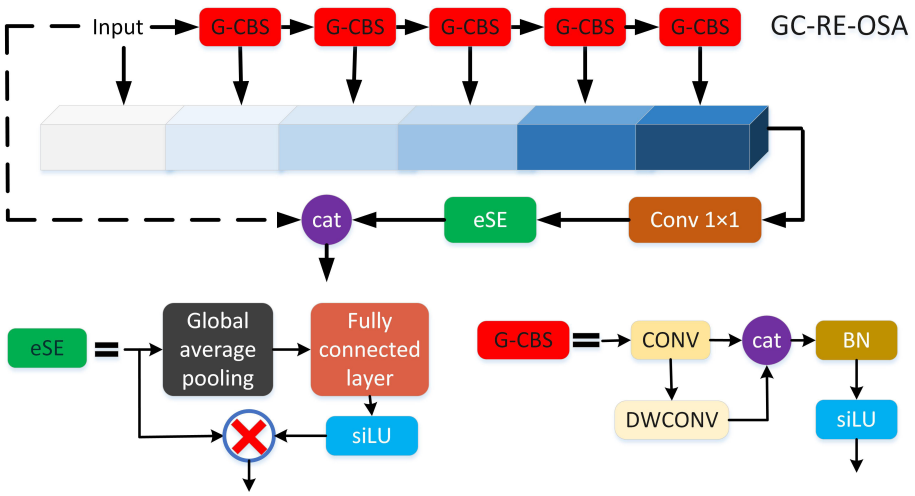



FIGURE 4
The structure of GC-RE-OSA module.

TABLE 3 The specific structure of LH-VoVnet.

| Type | Output stride | Stage | Output channel |
|---|---|---|---|
| Stem | 2<br>2<br>2 | 3×3 Ghost-conv., 64, Stride = 2<br>3×3 Ghost-conv., 64, Stride = 1<br>3×3 Ghost-conv., 128, Stride = 1 | 64 |
| Stage 1 | 4 | LDS Layer ×1, GC-RE-OSA×1 | 128 |
| Stage 2 | 8 | LDS Layer ×1, GC-RE-OSA×1 | 256 |
| Stage 3 | 12 | LDS Layer ×1, GC-RE-OSA×2 | 512 |
| Stage 4 | 32 | LDS Layer ×1, GC-RE-OSA×2 | 1024 |

features but it also introduces significant computational complexity and increases the time required for processing.

2. Compared to the shallower C3 and C4 features, the deepest layer C5 feature has deeper, higher level, and richer semantic features. These semantic features are more useful for distinguishing different objects and are more desirable for Transformer. Shallow features do not play much of a role due to the lack of better semantic features.

To address these issues, we only select the C5 feature map output by the backbone network as the input for the Transformer. To retain key feature information as much as possible, we replaced the simple flattening of feature maps into a vector with a 2D encoding in the Position Embedding module (Wu et al., 2021). Additionally, a lightweight single-scale Transformer encoder is adopted.

The Multi-Head Self-Attention (MHSA) aggregation in Transformer combines input elements without differentiating their positions; thus, Transformer possess permutation invariance. To alleviate this issue, we need to embed spatial information into the feature map, which requires adding 2D position encoding to the final layer feature map. Specifically, the original sine and cosine positional encodings in Position Embedding are respectively extended to column and row positional encodings, and concatenated with them finally.

After the feature map is processed by 2D position embedding, we use a single-scale Transformer Encoder, which only contains one Encoder layer (MHSA + Feed Forward network) to process the output of Q, K, and V at three scales. Note that the three scales share one SSTE and, through this shared operation, the information of the three scales can interact to some extent. Finally, the processing results are concatenated together to form a vector, which is then adjusted back to a 2D feature map, denoted as F5. In the neck part, C3, C4, and F5 are sent to dual-flow PAN + FPN for multi-scale feature fusion. See Figure 1 for details.

### 2.3.2 Dual-flow PAN + FPN

In order to achieve better information fusion of the three-layer feature maps (C3, C4, and F5), our enhanced neck implements a dual-stream PAN + FPN architecture, which is featured in the latest YOLO series. In addition to this, we have introduced GC-RE-OSA module to ensure faster detection speed while preserving accuracy. A comparison between YOLOv5 (Jocher et al., 2021) (Figure 5A) and our enhanced neck structure (Figure 5B) is provided. Our improved architecture substitutes the C3 module with the GC-RE-OSA module and eliminates the Conv. prior to upsampling. This enables direct utilization of features output from diverse stages of the backbone.

## 2.4 Head of FTR-YOLO

For the Head component, we have employed Decoupled Head to perform separate classification and regression tasks via two

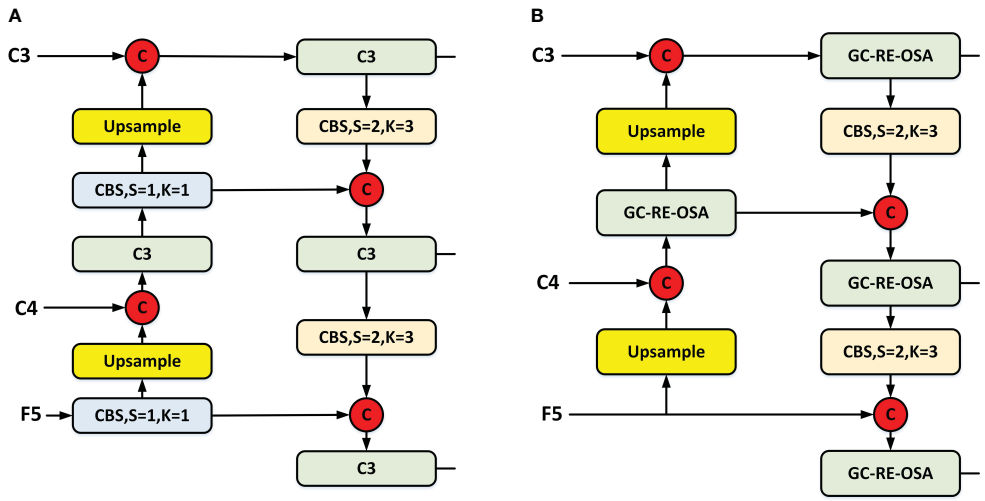



FIGURE 5
Two different neck structures. **(A)** YOLOv5 neck and **(B)** ours.

distinct convolutional channels. Furthermore, our architecture includes the ITAP within each branch, which enhances the interaction between the two tasks.

Object detection commonly faces a task conflict between classification and localization. While decoupled head is successfully applied to SOTA YOLO model in YOLOX (Ge et al., 2021), v6 (Li et al., 2023), v7 (Wang et al., 2023), and v8 (Terven and Cordova-Esparza, 2023), drawing lessons from most of the one-stage and two-stage detectors, single-stage detectors perform classification and localization tasks in parallel using two independently functioning branches. However, this dual-branch approach may lack interaction, resulting in inconsistent predictions during execution.

To address this issue, we drew inspiration from the TAP in TOOD (Feng et al., 2021) and made some improvements to maintain accuracy while improving speed. As shown in Figure 6, the ITAP uses eSE to replace the layer attention in TOOD. To further enhance efficiency, we incorporated a more efficient Convolution+BN layer+Silu (CBS) module before the shortcut. Moreover, during the training phase, we utilized different loss for the two branches.

## 2.5 Label assignment and loss

The loss calculation in our study employed the label assignment strategy. SimOTA is employed in YOLOX, v6 and v7 to enhance their performance. Task alignment learning (TAL) proposed in TOOD is used in YOLOv8. This strategy entails selecting positive samples based on the weighted scores of the classification and regression branches within the loss function. For the classification branch, we utilize the varifocal loss (VFL) (Zhang et al., 2021), while for the regression branch, the distribution focal loss (DFL) (Li et al., 2020) is employed. Furthermore, we incorporate the Complete-IoU (CIoU) Loss. The combination of these three losses is achieved through weighted proportions.

VFL utilizes the target score to assign weight to the loss of positive samples. This implementation significantly amplifies the impact of positive samples with high IoU on the loss function. Consequently, the model prioritizes high-quality samples during the training phase while de-emphasizing the low-quality ones. Similarly, both approaches utilize IoU-aware classification score (IACS) as the target for prediction. This enables effective learning of a combined representation that includes both classification score and localization quality estimation. By employing DFL to tackle the uncertainty associated with bounding boxes, the network gains the ability to swiftly concentrate on the distribution of neighboring regions surrounding the target location. See Equation 7 for details.

$$Loss = \frac{\alpha \cdot loss_{VFL} + \beta \cdot loss_{CIoU} + \gamma loss_{DFL}}{\sum_{i}^{N_{pos}} \hat{t}} \tag{7}$$

where $\hat{t}$ denotes the normalized score used in TOOD, α, β, and γ represent different weights.

# 3 Experimental results

The experimental hardware environment is configured with INTEL I7-13700 CPU, 32GB RAM, and GEFORCE RTX3090 graphics. The operating system is Windows10 professional edition, the programming language is Python 3.8, and the acceleration environment is CUDA 11.1 and CUDNN 8.2.0. The training parameters of the training process used in the experiment are shown in Table 4.

## 3.1 Ablation study on backbone

The improved network is composed of backbone, neck, and head, so the influence of the improvement of each part on the model performance should be verified respectively.

In this paper, the LH-VoVNet is verified through experiment. The improvements include (1) the LDS layer is used for downsampling. (2) By adding eSE block and RE-OSA module. (3) The Conv. is replaced with Ghost Module to further lightweight the network. The results of the ablation study are shown in Table 5.

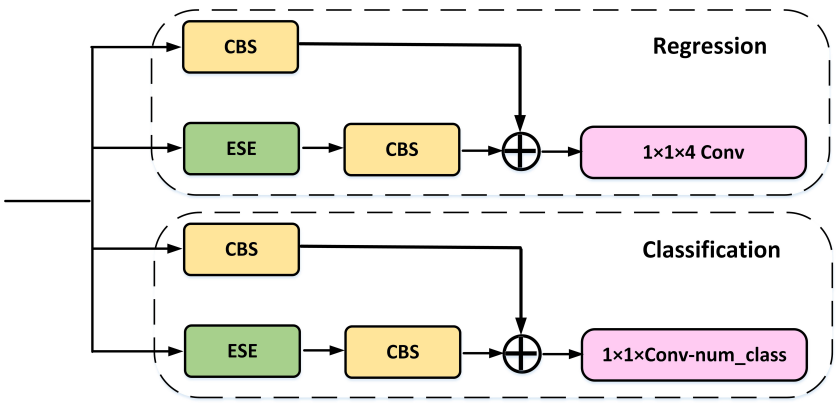



FIGURE 6
ITAP decoupled head structures.

TABLE 4 The implementation details of training parameters.

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| Optimizer | AdamW | Weight decay | 0.0005 |
| Learning rate | 0.001 | Momentum | 0.937 |
| Batch size | 8 | warmup steps | 300 |
| Image size | 640*640 | Epochs | 200 |
| NMS threshold | 0.7 | EMA decay | 0.9998 |

On the basis of VoVnet, compared by adding LDS layer/RE-OSA module improves accuracy by 1.06%/1.42% mAP. By replacing Ghost-conv., the number of parameters in the network is greatly reduced (−62.7%), the FPS is significantly improved (+78.9%), and the detection performance is also slightly improved (+0.31%). Finally, the integration of these three components shows that mAP 86.79% (+2.17%) is optimal, Params 24.7MB (−50.1%) and FPS 56 (+47.4%), achieve lightweight and real-time in backbone.

## 3.2 Ablation study on neck

To verify the effectiveness of the proposed neck, we evaluate the indicators of the set of variants designed in **Section 2.3**, including mAP, number of parameters, latency and FPS. The backbone used in the ablation experiment is LH-VoVNet. The improvements include the following: (1) Only the C5 feature map output by the backbone as the input is selected for the Transformer. (2) The real-time Transformer only includes 2D position embedding and SSTE to further lightweight the network. (3) The C3 module is replaced with GC-RE-OSA module. The parameters for the Transformer Encoder are as follows: num of head = 8, num of encoder layers = 1, hidden dim = 256, dropout = 0.1, activation = relu.

The experimental results are shown in Table 6. On the basis of YOLOv5 neck, by adding real-time Transformer delivers 1.41% AP improvement, while increasing the number of parameters by 4.5%, the latency by 47.2%, decreasing the FPS by 17.9%. This demonstrates the effective enhancement of detection accuracy by Transformer while maintaining a high degree of lightweight and real-time performance. By adding GC-RE-OSA module delivers 0.45% AP improvement, the number of parameters experienced a slight increase of 4.5%, the latency decreases by 25.0%, and the FPS increase by 8.9%. This shows that the module not only enables lightweight networking but also enhances performance. Finally, the integration of these two components shows that mAP 88.85% (+2.06%) is optimal, Params 22.5MB (−8.9%), Latency 56.3ms (+7.2%), and FPS 49 (−12.5%). The improved neck further enhances network detection performance and lightweight, albeit with a slight fluctuation in FPS and Latency that has negligible impact on real-time detection.

TABLE 5 The results of the ablation study of backbone components.

| Methods | mAP@0.5 | Params(M) | FPS |
|---|---|---|---|
| VoVnet | 84.62 | 49.0 | 38 |
| +LDS layer | 85.68 | 49.4 | 37 |
| +RE-OSA module | 86.04 | 53.5 | 24 |
| +Ghost-conv. | 84.93 | **18.3** | **68** |
| LH-VoVNet | **86.79** | 24.7 | 56 |

Bold values represents the optimal values.

TABLE 6 The results of the ablation study of neck components.

| Methods | mAP@0.5 | Params (M) | Latency (ms) | FPS |
|---|---|---|---|---|
| YOLOv5 neck | 86.79 | 24.7 | 52.5 | 56 |
| +Real time Transformer | 88.20 | 25.8 | 77.3 | 46 |
| +GC-RE-OSA module | 87.22 | **20.3** | **39.4** | **61** |
| Ours neck | **88.85** | 22.5 | 56.3 | 49 |

Bold values represents the optimal values.

## 3.3 Ablation study on head and loss

To verify the effectiveness of the proposed head, we evaluate the indicators of the set of variants designed in **Sections 2.4 and 2.5**, including mAP, number of parameters, latency, and FPS. We conduct this experiment on above-modified model, which uses LH-VoVNet, improved neck, and YOLOv5 head as the baseline. The parameters for the TAL are as follows: topk = 13, alpha = 1.0, and beta = 6.0. Similarly, for the SimOTA Assigner, the parameters are center_radius = 2.5 and topk = 10. In Equation 7, the weights assigned to the three losses are as follows: VFL ($\alpha$ = 1.0), CIoU ($\beta$ = 2.5), and DFL ($\gamma$ = 0.5). The experimental results are shown in Table 7.

On the basis of YOLOv5 head, by adding ITAP Decoupled Head delivers 0.61% AP improvement, while increasing the number of parameters by 6.2%, the latency by 6.6%, decreasing the FPS by 8.2%. This indicates that the improved head has minimal impact on parameter and computational speed, while simultaneously enhancing detection accuracy. By adding SimOTA delivers 0.27% AP improvement, the number of parameters/Latency/FPS experience a slight fluctuation by +2.2%/+1.4%/−2.0%. By adding TAL delivers 1.06% AP improvement, the number of parameters/Latency/FPS experience a slight fluctuation by +2.7%/+1.8%/−2.0%.

TABLE 7 The results of the ablation study of head & loss components.

| Methods | mAP@0.5 | Params (M) | Latency (ms) | FPS |
|---|---|---|---|---|
| YOLOv5 head | 88.85 | **22.5** | **56.3** | **49** |
| +ITAP decoupled head | 89.46 | 23.9 | 60.0 | 45 |
| +SimOTA | 89.12 | 23.0 | 57.1 | 48 |
| +TAL | 89.91 | 23.1 | 57.3 | 48 |
| Ours head | **90.67** | 24.5 | 61.5 | 44 |

Bold values represents the optimal values.

After comparing the label assignments of SimOTA and TAL, it was found that TAL exhibited superior performance, thus making it the preferred choice for our paper. Finally, we adopted a hybrid methodology comprising ITAP Decoupled Head+TAL, resulting in an optimized mAP of 90.67% (+1.82%). Additionally, there was an augmentation in the model's parameters and Latency to 24.5MB (+8.9%) and 61.5 (+9.2%), respectively, while the FPS decreased to 44 (−10.2%).

## 3.4 Comparison with other detectors

Table 8 compares FTR-YOLO with other real-time detectors (YOLOv5, YOLOv6, YOLOv7, YOLOv8, and PP-YOLOE) and Vision Transformer detector (DINO-DETR).

Compared to real-time detectors YOLOv5/YOLOv6/YOLOv7/YOLOv8/PP-YOLOE, FTR-YOLO significantly improves accuracy by 6.44%/3.13%/1.81%/1.75%/4.92% mAP, increases FPS by 10.0%/18.9%/7.3%/0.0%/7.3%, and reduces the number of parameters by 47.1%/58.5%/33.1%/43.4%/53.1%. Even among the AP metrics for the four categories, the FTR-YOLO algorithm consistently demonstrates the best performance. Additionally, the differences in AP values among the four disease categories are relatively small, indicating that the FTR-YOLO algorithm exhibits good robustness. This demonstrates the superior performance of FTR-YOLO compared to the state-of-the-art YOLO detectors in terms of accuracy, speed, and lightweight.

In order to determine the statistical significance of the differences between various algorithms, we performed four independent repeated experiments for each algorithm. A *t* test was employed, and the *p*-values for mAP among different algorithms were computed. Due to substantial variations in parameters, including image input size and training epochs, between the DINO-DETR algorithm and other detection algorithms, it was excluded from the statistical analysis. The experimental results reveal that the *p*-values comparing different algorithms are considerably small, all well below 0.01, signifying noteworthy variances between the algorithms.

Compared to DINO-DETR, the number of parameters/mAP/FPS experience a fluctuation by −48.3%/−0.45%/+2100.0%. This observation highlights that, while DINO achieves a slightly higher mAP of 0.45% compared to FTR-YOLO, it fails to meet real-time requirements due to its significantly lower FPS (2). Furthermore, there is no discernible advantage in terms of model lightweight.

## 3.5 Object size sensitivity analysis

Different disease types, periods, and locations result in different characteristics and sizes. The improved network proposed in this paper effectively enhances the detection accuracy in small object scenario. In order to verify the detection effect of the small object detection performance, the test dataset is divided into five groups based on the size of the disease area. The, 0%–10%, 10%–20%, 20%–40%, 40%–60%, and 60%–90%, five groups are named with different labels: XS, S, M, L, and XL, which represent the size of different objects. The comparison of the detection accuracy of six common algorithms with FTR-YOLO for five different sizes.

As shown in Figure 7, The YOLOv5 and PP-YOLOE perform well in large target region (XL and L), but the detection effect of small target decreases sharply (XS and S). YOLOv6, v7, and v8 have shown slight improvements in detection accuracy compared to YOLOv5. Among them, v8 performs better on smaller scales (XS and S) while demonstrating similar detection effectiveness on M, L, and XL scales. The DINO-DETR is optimal in the detection accuracy on smaller scales (XS and S). FTR-YOLO demonstrates superior performance on M (90.43%), L (95.30%), and XL (98.73%) scales. The mAP values show a significant improvement when compared to the other five YOLO algorithms on both XS and S scales. Specifically, it shows improvements of 7.81%/6.71%/4.69%/3.31%/6.58% on XS scale, and improvements of 10.65%/8.01%/5.67%/4.82%/7.24% on S scale. These improvements highlight the effectiveness of the system in achieving higher mAP values compared to its counterparts. While it may have slightly lower performance than DINO-DETR, FTR-YOLO is still the optimal choice due to its lightweight and real-time capabilities.

## 3.6 Image size sensitivity analysis

The Batch Random Resize is applied to a batch of images, which helps increase the diversity and randomness of the data. By introducing such variations during training, the model becomes

TABLE 8 The comparison results of different methods.

| Method | Size | Params(M) | AP for each category* | | | | mAP@0.5 | FPS | *p*-value |
|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | | | |
| Yolo V5 | 640*640 | 46.3 | 87.42 | 76.03 | 85.29 | 88.18 | 84.23 | 40 | < 0.01 |
| Yolo V6 | 640*640 | 59.0 | 90.70 | 88.59 | 80.37 | 90.14 | 87.54 | 37 | < 0.01 |
| Yolo V7 | 640*640 | 36.6 | 89.51 | 90.44 | 84.23 | 91.26 | 88.86 | 41 | < 0.01 |
| Yolo V8 | 640*640 | 43.3 | 89.83 | 88.47 | 85.30 | 92.08 | 88.92 | **44** | < 0.01 |
| PP-YOLOE | 640*640 | 52.2 | 88.15 | 78.59 | 84.42 | 91.84 | 85.75 | 41 | < 0.01 |
| DINO-DETR | 800*1333 | 47.4 | **91.79** | 90.58 | **88.76** | **93.35** | **91.12** | 2 | —— |
| FTR-YOLO | 640*640 | **24.5** | 90.73 | **90.67** | 88.54 | 92.74 | 90.67 | **44** | < 0.01 |

*In Table 8, 1, 2, 3, and 4 represent the four types of detected diseases: 1, anthracnose; 2, grapevine white rot; 3, gray mold; 4, powdery mildew.
Bold values represents the optimal values.

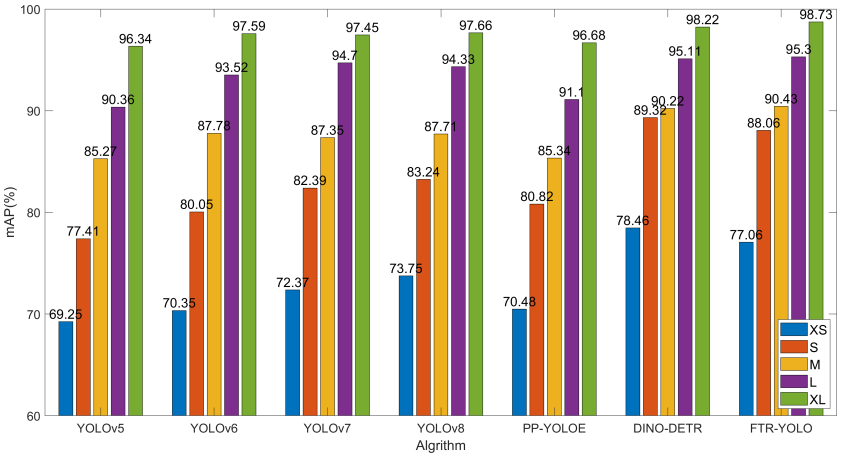



FIGURE 7
Object size sensitivity analysis.

more robust and better able to generalize to unseen examples. This technique can contribute to improving the overall performance and generalization ability of the model in tasks such as object detection or image classification. In our experiment, the data were randomly resized into the following 10 different sizes: [320, 384, 448, 480, 512, 544, 576, 640, 672, 704, 736, and 768].

To further validate the detection performance on images of varying sizes, we categorized the dataset into three groups based on different sizes: (1) small size, less than or equal to 480; (2) medium size, ranging from 480 to 768; (3) large size, greater than 768. Figure 8 shows the detection performance of seven different algorithms.

The detection accuracy among samples of different sizes does not show significant variation, as illustrated in Figure 8. However, it should be noted that the detection accuracy is affected by the distortion introduced when resizing small-sized images to 640. Among the various algorithms, the DINO-DETR algorithm is particularly sensitive to this impact. On the other hand, FTR-YOLO demonstrates superior performance on small-sized images (87.34%) and medium-sized images (90.80%). Additionally, FTR-YOLO significantly improves mAP values compared to the other five YOLO algorithms on small-sized images by 4.73%, 3.31%, 2.02%, 1.49%, and 3.85%. It also improves mAP values on medium-sized images by 5.07%, 3.55%, 2.20%, 2.24%, and 4.78%. Furthermore, it improves mAP values on large-sized images by 5.41%, 2.92%, 1.46%, 1.34%, and 4.85%. Although FTR-YOLO may have slightly lower performance than DINO-DETR on large-sized images, it is still considered the optimal choice due to its lightweight design and real-time capabilities.

Based on the comparative evaluation in **Sections 3.4–3.6**, LH-VoVnet-39 outperforms YOLO's backbone CSPDarknet-53 or CSPResnet-50, which replaced the convolution downsampling operation with LDS, enabling the model to better preserve important features. Additionally, the GC-RE-OSA module, along with residual connections and eSE attention mechanism, further improves feature extraction. Furthermore, we have made improvements to the TAP and loss selection based on YOLOv7

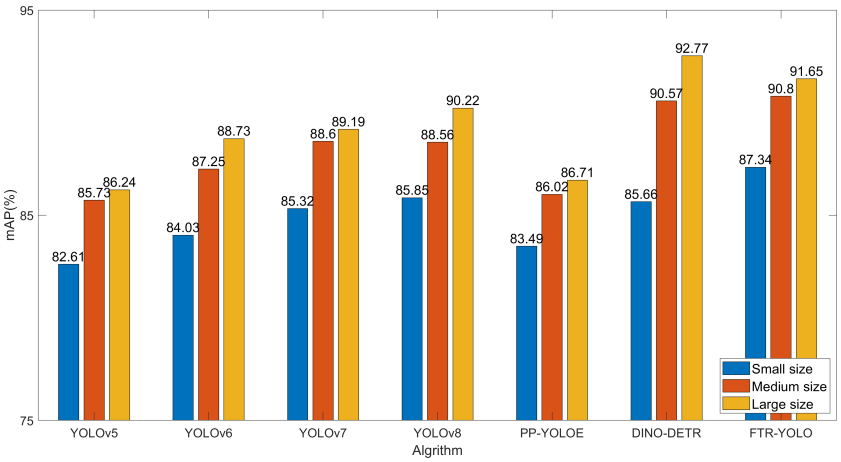



FIGURE 8
Image size sensitivity analysis.

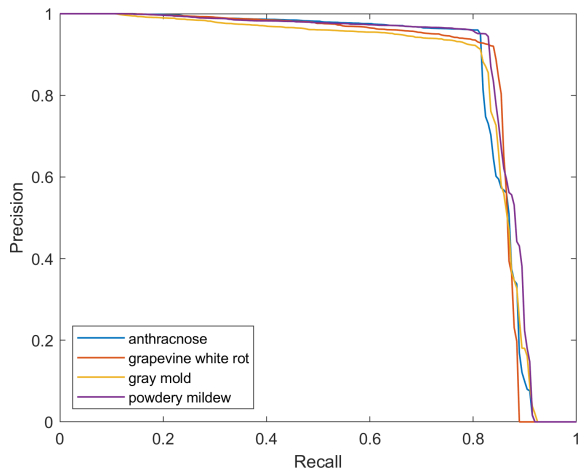



FIGURE 9
The p–r curve of FTR-YOLO.

and v8 decoupled heads. As a result, FTR-YOLO demonstrates superior performance in terms of mAP and AP values for each category, with minimal numerical differences and strong generalization capabilities (Table 8).

Due to VoVnet-39 having fewer layers and the utilization of lightweight ghost modules instead of convolutions, in addition to a real-time transformer that consists of 2D position embedding and a single-scale Transformer encoder, but does not include decoder, FTR-YOLO achieves comparable FPS performance to YOLOv8 while delivering optimal results (Table 8).

On the other hand, DINO-DETR, with its multi-scale Transformer encoder and decoder, possesses more input feature maps and layers, resulting in better performance for object detection. It outperforms FTR-YOLO in specific metrics such as mAP in Table 8, mAP for XS and S object scales in Figure 7, and mAP for large-sized inputs in Figure 8. However, this improvement comes at the cost of significantly increased computational complexity, leading to an FPS of only 2, which limits its practical applications.

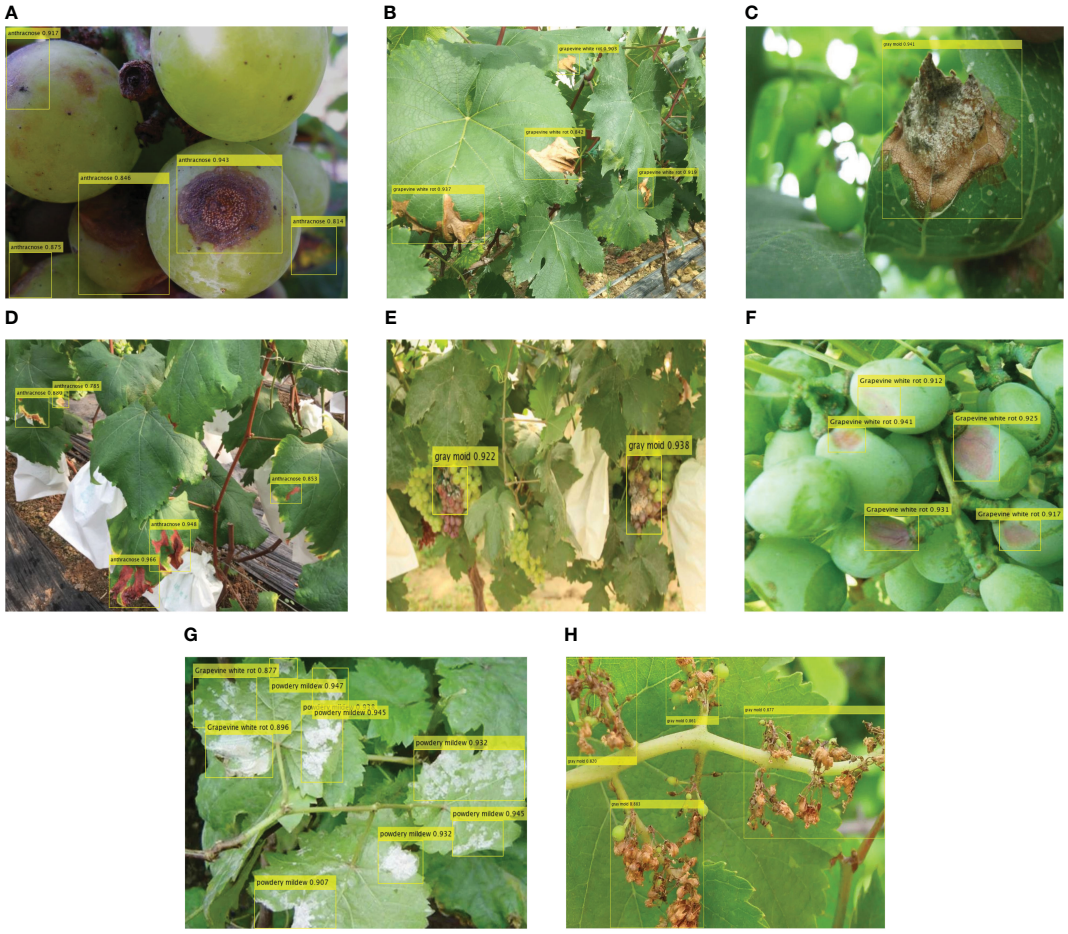



FIGURE 10
The detection results of FTR-YOLO. **(A)** diseased leaves of anthracnose, **(B)** diseased leaves of grapevine white rot, **(C)** diseased leaves of gray mold, **(D)** diseased leaves of powdery mildew, **(E)** diseased fruits of gray mold, **(F)** diseased fruits of grapevine white rot, **(G)** diseased fruits of anthracnose, and **(H)** diseased inflorescence of gray mold.

### 3.7 Performance visualization on FTR-YOLO

The precision–recall curves of each disease are provided in Figure 9, which intuitively shows the detailed relationship between precision and recall. It has been observed that as recall increases, the rate of change in precision also increases. When the graph's curve is closer to the upper right corner, it indicates that the drop in precision as recall increases is less noticeable, indicating improved overall performance of FTR-YOLO.

The detection results of four diseases of grape are shown in Figure 10. Figures 10A–D show the detection results of diseased leaves of anthracnose, grapevine white rot, gray mold, powdery mildew, respectively, while Figures 10E–G show the detection results of diseased fruits of gray mold, grapevine white rot and anthracnose respectively. Figure 10H shows the diseased inflorescence of gray mold. The results indicate that the FTR-YOLO model exhibits precise detection of diverse symptoms in different parts of the vine within natural scenes. This underscores the model's remarkable generalization and robustness. It is evident that the majority of detection boxes have scores exceeding 0.8. Additionally, a substantial portion of the diseased areas have been accurately detected, highlighting the exceptional precision and precise localization capabilities of the proposed model. We also compared the detection performance of different algorithms. For details, please see Figure 11.

The experimental results in Figure 11 show that YOLOv5 missed some small objects, while the PPYOLOE and DINO-DETR algorithms detected additional object areas. There are slight differences in the detected bounding boxes and confidence levels among the different algorithms, which overall align with the experimental results obtained in the paper. The proposed FTR-YOLO (Figure 10A) performs well in terms of detection accuracy and confidence levels.

## 4 Discussions

Based on the information provided, the FTR-YOLO model is proposed in this paper to achieve accurate, real-time, and lightweight intelligent detection of four common grape diseases in natural environments. The model incorporates several improvements in its components. In backbone, the LH-VoVNet is introduced, which includes LDS layer and Ghost-conv. Additionally, eSE blocks and residual connections are added to the OSA module (GC-RE-OSA module). Experimental results presented in Table 5 demonstrate that the LH-VoVNet achieves optimal performance in terms of detection (mAP 86.79%), lightweight design (Params 24.7MB), and real-time capabilities (FPS 56). The neck component also undergoes improvements. Only the C5 feature map output by the backbone is selected as the input for the real-time Transformer, includes 2D position embedding and SSTE. Additionally, the C3 module is replaced with the GC-RE-OSA module in PAN + FPN. Experimental results presented in Table 6 show that the improved neck further enhances performance in detection (mAP 88.85%) and lightweight design (Params 22.5MB). In the head and loss component, the ITAP is proposed, and TAL is used with VFL and DFL. Experimental results presented in Table 7 demonstrate that the ITAP Decoupled Head + TAL achieves an optimized mAP of 90.67%. Moreover, Table 8; Figures 7, 8 show the superior performance of FTR-YOLO compared to state-of-the-art YOLO detectors in terms of accuracy (mAP 90.67%), speed (FPS 44), and lightweight design (Params 24.5MB), particularly improved accuracy on smaller scales (XS and S) and different sample sizes.

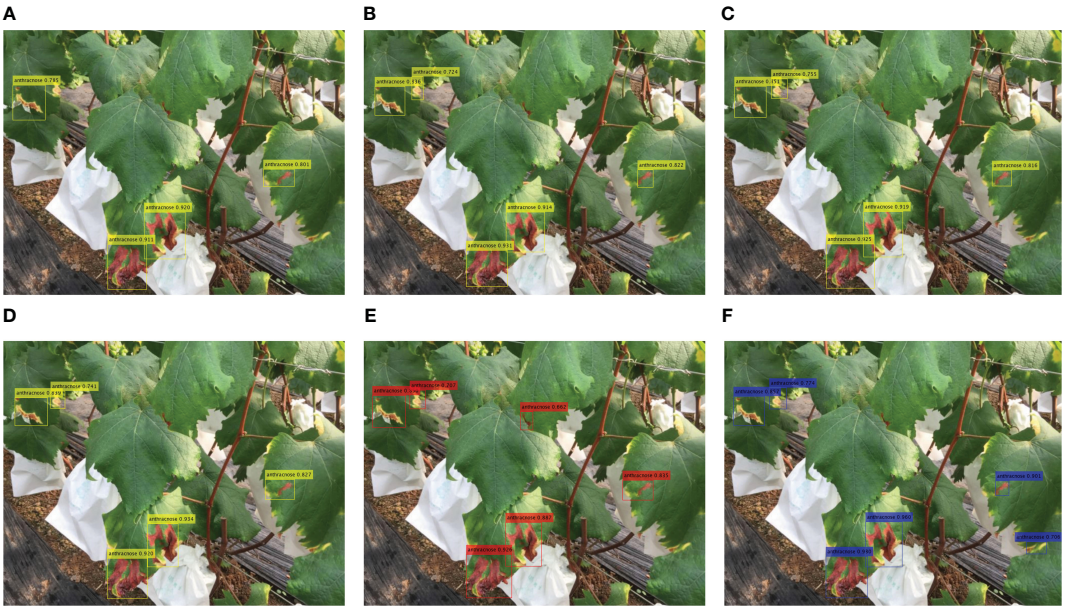



FIGURE 11
The detection results of different methods. **(A)** YOLOv5, **(B)** YOLOv6, **(C)** YOLOv7, **(D)** YOLOv8, **(E)** PPYOLO-E, and **(F)** DINO-DETR.

## 5 Conclusion and future works

In this paper, we propose a real-time and lightweight detection model, called Fusion Transformer YOLO, for grape disease detection. In backbone, we integrate GC-RE-OSA module based on VoVnet, effectively improving the ability of network to extract feature information while keeping the network lightweight. In neck component, an improved Real-Time Transformer with 2D position embedding and SSTE are incorporated to the last feature map to accurate detection of small targets in natural environments. In head component, the Decoupled Head based on the ITAP is adopted to optimize detection strategy. Our proposed FTR-YOLO achieved 24.5MB Params, 90.67% mAP@0.5 with 44 FPS, which outperformed YOLOv5-v8 and PP-YOLOE. Although FTR-YOLO uses a real-time Transformer to improve model performance, it still falls behind DETR in terms of performance due to DETR's multi-scale and multi-layer global transformer architecture.

Future studies plan to explore the fusion of CNN and transformer models, as well as the integration of multimodal features, to further enhance the model's performance. Additionally, this paper focuses on disease detection in grapes, theoretically, the FTR-YOLO algorithm has the potential to achieve good performance when retrained on other datasets. It can be applied to tasks such as the detection of plant traits and pest diseases in other plants.

## Data availability statement

The original contributions presented in the study are included in the article/supplementary material. Further inquiries can be directed to the corresponding author.

## Author contributions

YL: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. QY: Data curation, Funding acquisition, Writing – review & editing. SG: Supervision, Validation, Visualization, Writing – review & editing.

## Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This research is supported by the National Natural Science Foundation of China under Grant No. ZZG0011806; Scientific research Project of Tianjin Science and Technology Commission under Grant No. 2022KJ108 and 2022KJ110; Tianjin University of Technology and Education Key Talent Project under Grant No. KYQD202104 and KYQD202106.

## Acknowledgments

We are grateful for the reviewers' hard work and constructive comments, which allowed us to improve the quality of this manuscript.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

Adeel, A., Khan, M. A., Akram, T., Sharif, A., Yasmin, M., Saba, T., et al. (2022). Entropy-controlled deep features selection framework for grape leaf diseases recognition. *Expert Syst.* 39, e12569. doi: 10.1111/exsy.12569

Ahmad, A., Saraswat, D., and El Gamal, A. (2023). A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools. *Smart Agric. Technol.* 3, 100083. doi: 10.1016/j.atech.2022.100083

Balakrishna, K., and Rao, M. (2019). Tomato plant leaves disease classification using KNN and PNN. *Int. J. Comput. Vision Image Process. (IJCVIP)* 9, 51–63. doi: 10.4018/IJCVIP.2019010104

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). *End-to-end object detection with transformers. European conference on computer vision* (Cham. Springer International Publishing), 213–229.

Elnahal, A., El-Saadony, M., Saad, A., Desoky, E.-S., Eltahan, A., Rady, M., et al. (2022). Correction: The use of microbial inoculants for biological control, plant growth promotion, and sustainable agriculture: A review. *Eur. J. Plant Pathol.* 162, 1–1. doi: 10.1007/s10658-022-02472-3

El-Saadony, M. T., Saad, A. M., Soliman, S. M., Salem, H. M., Ahmed, A. I., Mahmood, M., et al. (2022). Plant growth-promoting microorganisms as biocontrol agents of plant diseases: Mechanisms, challenges and future perspectives. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.923880

Feng, C., Zhong, Y., Gao, Y., Scott, M. R., and Huang, W. (2021). *Tood: Task-aligned one-stage object detection. 2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (IEEE Computer Society, Montreal, QC, Canada), 3490–3499. doi: 10.1109/ICCV48922.2021.00349

Fuentes, A. F., Yoon, S., Lee, J., and Park, D. S. (2018). High-performance deep neural network-based tomato plant diseases and pests diagnosis system with refinement filter bank. *Front. Plant Sci.* 9. doi: 10.3389/fpls.2018.01162.

Fuentes, A., Yoon, S., and Park, D. S. (2019). Deep learning-based phenotyping system with glocal description of plant anomalies and symptoms. *Front. Plant Sci.* 10. doi: 10.3389/fpls.2019.01321

Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J. (2021). Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*. doi: 10.48550/arXiv.2107.08430

Guan, H., Fu, C., Zhang, G., Li, K., Wang, P., and Zhu, Z. (2023). A lightweight model for efficient identification of plant diseases and pests based on deep learning. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1227011

Hameed, J., and Üstündağ, B. (2020). Evolutionary feature optimization for plant leaf disease detection by deep neural networks. *Int. J. Comput. Intell. Syst.* 13, 12. doi: 10.2991/ijcis.d.200108.001

Hatuwal, B., Shakya, A., and Joshi, B. (2021). Plant leaf disease recognition using random forest, KNN, SVM and CNN. *Polibits* 62, 13–19. doi: 10.17562/PB-62-2

Hu, J., Shen, L., and Sun, G. (2018). "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. Salt Lake City, UT, USA 7132–7141. doi: 10.1109/CVPR.2018.00745

Huang, H., Huang, T., Li, Z., Lyu, S., and Hong, T. (2021). Design of citrus fruit detection system based on mobile platform and edge computer device. *Sensors* 22, 59. doi: 10.3390/s22010059

Ji, M., Zhang, L., and Wu, Q. (2020). Automatic grape leaf diseases identification via UnitedModel based on multiple convolutional neural networks. *Inf. Process. Agric.* 7, 418–426. doi: 10.1016/j.inpa.2019.10.003

Jiang, P., Chen, Y., Liu, B., He, D., and Liang, C. (2019). Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. *IEEE Access* 7, 59069–59080. doi: 10.1109/Access.6287639.

Jianming, Z., Chaoquan, L., Xudong, L., Hye-Jin, K., and Jin, W. (2019). A full convolutional network based on DenseNet for remote sensing scene classification. *Math. Biosci. Eng.* 16, 3345–3367. doi: 10.3934/mbe.2019167

Jocher, G., Stoken, A., Borovec, J., Chaurasia, A., Changyu, L., Hogan, A., et al. (2021). *ultralytics/yolov5: v5. 0-YOLOv5-P6 1280 models, AWS, Supervise. ly and YouTube integrations* (Zenodo).

Kaur, P. P., and Singh, S. (2021). *Classification of Herbal Plant and Comparative Analysis of SVM and KNN Classifier Models on the Leaf Features Using Machine Learning* (Singapore: Springer Singapore), 227–239.

Kuznetsova, A., Maleva, T., and Soloviev, V. (2020). *Detecting apples in orchards using YOLOv3 and YOLOv5 in general and close-up images. Advances in Neural Networks–ISNN 2020: 17th International Symposium on Neural Networks, ISNN 2020, Cairo, Egypt, December 4–6, 2020, Proceedings 17, 2020* (Springer, Cham), 233–243. doi: 10.1007/978-3-030-64221-1_20

Lee, C. P., Lim, K. M., Song, Y. X., and Alqahtani, A. (2023). Plant-CNN-ViT: plant classification with ensemble of convolutional neural networks and vision transformer. *Plants* 12, 2642. doi: 10.3390/plants12142642

Li, C., Li, L., Geng, Y., Jiang, H., Cheng, M., Zhang, B., et al. (2023). Yolov6 v3. 0: A full-scale reloading. *arXiv preprint arXiv:2301.05586*. doi: 10.48550/arXiv.2301.05586

Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., et al. (2020). Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Adv. Neural Inf. Process. Syst.* 33, 21002–21012. doi: 10.1109/CVPR46437.2021.01146

Liu, H., Jiao, L., Wang, R., Xie, C., Du, J., Chen, H., et al. (2022). WSRD-Net: A convolutional neural network-based arbitrary-oriented wheat stripe rust detection method. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.876069

Liu, J., and Wang, X. (2020). Tomato diseases and pests detection based on improved Yolo V3 convolutional neural network. *Front. Plant Sci.* 11. doi: 10.3389/fpls.2020.00898

Long, X., Deng, K., Wang, G., Zhang, Y., Dang, Q., Gao, Y., et al. (2020). PP-YOLO: An effective and efficient implementation of object detector. *arXiv preprint arXiv:2007.12099*. doi: 10.48550/arXiv.2007.12099

Lu, X., Yang, R., Zhou, J., Jiao, J., Liu, F., Liu, Y., et al. (2022). A hybrid model of ghost-convolution enlightened transformer for effective diagnosis of grape leaf disease and pest. *J. King Saud University-Computer Inf. Sci.* 34, 1755–1767. doi: 10.1016/j.jksuci.2022.03.006

Pinheiro, I., Moreira, G., Queirós da Silva, D., Magalhães, S., Valente, A., Moura Oliveira, P., et al. (2023). Deep learning YOLO-based solution for grape bunch detection and assessment of biophysical lesions. *Agronomy* 13, 1120. doi: 10.3390/agronomy13041120

Qi, C., Nyalala, I., and Chen, K. (2021). Detecting the early flowering stage of tea chrysanthemum using the F-YOLO model. *Agronomy* 11, 834. doi: 10.3390/agronomy11050834

Qiu, R.-Z., Chen, S.-P., Chi, M.-X., Wang, R.-B., Huang, T., Fan, G.-C., et al. (2022). An automatic identification system for citrus greening disease (Huanglongbing) using a YOLO convolutional neural network. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.1002606

Rathor, S. (2021). "Ensemble based plant species recognition system using fusion of hog and kaze approach," in *Futuristic Trends in Network and Communication Technologies*. Eds. P. K. Singh, G. Veselov, V. Vyatkin, A. Pljonkin, J. M. Dodero and Y. Kumar (Springer Singapore, Singapore), 536–545.

Sanath Rao, U., Swathi, R., Sanjana, V., Arpitha, L., Chandrasekhar, K., and Naik, P. K. (2021). Deep learning precision farming: grapes and mango leaf disease detection by transfer learning. *Global Transitions Proc.* 2, 535–544. doi: 10.1016/j.gltp.2021.08.002

Singh, V., and Misra, A. K. (2017). Detection of plant leaf diseases using image segmentation and soft computing techniques. *Inf. Process. Agric.* 4, 41–49. doi: 10.1016/j.inpa.2016.10.005

Soeb, M. J. A., Jubayer, M. F., Tarin, T. A., Al Mamun, M. R., Ruhad, F. M., Parven, A., et al. (2023). Tea leaf disease detection and identification based on YOLOv7 (YOLO-T). *Sci. Rep.* 13, 6078. doi: 10.1038/s41598-023-33270-4

Sozzi, M., Cantalamessa, S., Cogato, A., Kayad, A., and Marinello, F. (2022). Automatic bunch detection in white grape varieties using YOLOv3, YOLOv4, and YOLOv5 deep learning algorithms. *Agronomy* 12, 319. doi: 10.3390/agronomy12020319

Terven, J., and Cordova-Esparza, D. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* 5, 1680–1716. doi: 10.3390/make5040083

Trivedi, V. K., Shukla, P. K., and Pandey, A. (2022). Automatic segmentation of plant leaves disease using min-max hue histogram and k-mean clustering. *Multimedia Tools Appl.* 81, 20201–20228. doi: 10.1007/s11042-022-12518-7

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2023). "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (CVPR), Vancouver, BC, Canada 7464–7475. doi: 10.1109/CVPR52729.2023.00721

Wang, L., Zhao, Y., Liu, S., Li, Y., Chen, S., and Lan, Y. (2022). Precision detection of dense plums in orchards using the improved YOLOv4 model. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.839269

Wu, K., Peng, H., Chen, M., Fu, J., and Chao, H. (2021). "Rethinking and improving relative position encoding for vision transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. (ICCV): Montreal, QC, Canada 10033–10041. doi: 10.1109/ICCV48922.2021.00988

Xie, X., Ma, Y., Liu, B., He, J., Li, S., and Wang, H. (2020). A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks. *Front. Plant Sci.* 11. doi: 10.3389/fpls.2020.00751

Yang, L., Xu, S., Yu, X., Long, H., Zhang, H., and Zhu, Y. (2023). A new model based on improved VGG16 for corn weed identification. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1205151

Zhang, Z., Qiao, Y., Guo, Y., and He, D. (2022). Deep learning based automatic grape downy mildew detection. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.872107

Zhang, H., Wang, Y., Dayoub, F., and Sunderhauf, N. (2021). "Varifocalnet: An iou-aware dense object detector," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. (CVPR): Nashville, TN, USA 8514–8523. doi: 10.1109/CVPR46437.2021.00841

Zhang, B., Wang, R., Zhang, H., Yin, C., Xia, Y., Fu, M., et al. (2022). Dragon fruit detection in natural orchard environment by integrating lightweight network and attention mechanism. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.1040923.

Zhao, L., Niu, R., Li, B., Chen, T., and Wang, Y. (2022). Application of improved instance segmentation algorithm based on VoVNet-v2 in open-pit mines remote sensing pre-survey. *Remote Sens.* 14, 2626. doi: 10.3390/rs14112626

Zhou, J., Hu, W., Zou, A., Zhai, S., Liu, T., Yang, W., et al. (2022). Lightweight detection algorithm of kiwifruit based on improved YOLOX-S. *Agriculture* 12, 993. doi: 10.3390/agriculture12070993

Zhu, J., Cheng, M., Wang, Q., Yuan, H., and Cai, Z. (2021). Grape leaf black rot detection based on super-resolution image enhancement and deep learning. *Front. Plant Sci.* 12. doi: 10.3389/fpls.2021.695749

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2020). Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*. doi: 10.48550/arXiv.2010.04159

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 11 March 2024
DOI 10.3389/fpls.2024.1305358

Check for updates

OPEN ACCESS

EDITED BY
Sook Yoon,
Mokpo National University, Republic of Korea

REVIEWED BY
Nebojsa Bacanin,
Singidunum University, Serbia
Yunchao Tang,
Guangxi University, China
Aibin Chen,
Central South University Forestry and Technology, China

*CORRESPONDENCE
Jianlong Wang
wangjianlong24@hpu.edu.cn

RECEIVED 01 October 2023
ACCEPTED 15 February 2024
PUBLISHED 11 March 2024

CITATION
Wang J, Jia J, Zhang Y, Wang H and Zhu S (2024) RAAWC-UNet: an apple leaf and disease segmentation method based on residual attention and atrous spatial pyramid pooling improved UNet with weight compression loss.
*Front. Plant Sci.* 15:1305358.
doi: 10.3389/fpls.2024.1305358

COPYRIGHT
© 2024 Wang, Jia, Zhang, Wang and Zhu. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# RAAWC-UNet: an apple leaf and disease segmentation method based on residual attention and atrous spatial pyramid pooling improved UNet with weight compression loss

Jianlong Wang[1]*, Junhao Jia[1], Yake Zhang[2], Haotian Wang[1] and Shisong Zhu[1]

[1]School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, China, [2]School of Computer and Information Engineering, Henan Normal University, Xinxiang, China

**Introduction:** Early detection of leaf diseases is necessary to control the spread of plant diseases, and one of the important steps is the segmentation of leaf and disease images. The uneven light and leaf overlap in complex situations make segmentation of leaves and diseases quite difficult. Moreover, the significant differences in ratios of leaf and disease pixels results in a challenge in identifying diseases.

**Methods:** To solve the above issues, the residual attention mechanism combined with atrous spatial pyramid pooling and weight compression loss of UNet is proposed, which is named RAAWC-UNet. Firstly, weights compression loss is a method that introduces a modulation factor in front of the cross-entropy loss, aiming at solving the problem of the imbalance between foreground and background pixels. Secondly, the residual network and the convolutional block attention module are combined to form Res_CBAM. It can accurately localize pixels at the edge of the disease and alleviate the vanishing of gradient and semantic information from downsampling. Finally, in the last layer of downsampling, the atrous spatial pyramid pooling is used instead of two convolutions to solve the problem of insufficient spatial context information.

**Results:** The experimental results show that the proposed RAAWC-UNet increases the intersection over union in leaf and disease segmentation by 1.91% and 5.61%, and the pixel accuracy of disease by 4.65% compared with UNet.

**Discussion:** The effectiveness of the proposed method was further verified by the better results in comparison with deep learning methods with similar network architectures.

KEYWORDS

apple leaf and disease, CBAM, Resnet, ASPP, weight compress

# 1 Introduction

The apple leaf is an important organ for the growth and development of apples. Apple is also the most grown fruit in northern China because of its high nutritional value, containing high levels of calcium, iron, zinc and other trace elements (Su et al., 2022). According to Shenzhen Daochuang Intelligence, China's annual apple bagging output will reach 36.8 million tons in 2023. A key factor affecting apple yield is leaf photosynthetic area. However, its cultivation often breeds different diseases. Apples grow many diseases during cultivation and leaf diseases are a common plant disease (van Bruggen et al., 2016). The diseases are caused by fungi, bacteria, or viruses and can affect leaf respiration, which in turn affects apple growth and yield (Lee et al., 2020). Therefore, the quick detection of apple leaf diseases and precise spraying of pesticides according to the size of the leaf disease area are essential to guarantee the healthy growth of apples (Sun et al., 2021). Precision disease control techniques also play a decisive role in securing apple yields (Tang et al., 2023).

Traditional apple leaf and disease detection methods usually rely on manual visual identification or capture of pests to determine the likelihood of disease occurrence (Liu and Wang, 2021). The method involves high labor costs, lengthy time consumption, and is easily influenced by subjective factors (Li et al., 2021). With the development of image segmentation technology, traditional segmentation includes the Canny Edge Detection Algorithm (Xizhen et al., 2021), the Region Growing Algorithm (Jin et al., 2018), the Watershed Algorithm (Jin et al., 2018), and so on. These methods primarily concentrate on the local pixel relationships and can easily lead to the misconception of local optimization. Traditional segmentation techniques tend to produce discontinuous or incomplete segmentation results when applied to images with complex textures or shapes (Lu et al., 2023b). Moreover, it requires manual adjustment of certain parameters, which can be challenging for non-professionals. Lychee picking robots using artificial intelligence algorithms to proactively remove obstacles have been proposed, which provide an intelligent technology that reduces labor costs (Wang et al., 2023).

The combination of metaheuristics and machine learning methods is also an important research direction. The enhanced version of the firefly algorithm (FA) makes a great contribution to the prevention of overfitting in network training (Bacanin et al., 2021). A hierarchical feature selection method based on genetic algorithm for handwritten word recognition is proposed, which uses a hierarchical feature selection model to optimize the handwritten word images and extract the local and global features (Malakar et al., 2020). The genetically guided best artificial flora algorithm is proposed, which is used to solve the problems of artificial neural network training and feature selection (Bacanin et al., 2022). To predict the number of COVID-19 cases, a novel technique that combines machine learning and beetle antennae search methods is proposed, providing an effective technical support for controlling the outbreak of the epidemic (Zivkovic et al., 2021). A new deep neural network with transfer learning in remote object detection from drone is proposed, where the use of migration learning accelerates the training process and improves the generalization ability of the model (Woźniak et al., 2022). Although these articles may require further personalization to suit the specific research needs, they provide us with useful experiences and methods that can be fully utilized in further studies. The fusion of deep learning and image processing algorithms to detect and count banana strings method is proposed which combines the advantages of deep learning and image processing to improve the efficiency of counting (Wu et al., 2023). Automatic and intelligent data collectors and classifiers have been proposed. It is used for data collection, detection and classification of pearl millet rust and rice blast disease (Kundu et al., 2021). The articles provide a variety of techniques available for leaf and disease segmentation.

In recent years, convolutional neural networks have made significant advancements in leaf and disease detection, which is an end-to-end learning approach (Shi et al., 2023). It can automatically extract advanced image features and reduce the need for human intervention (Lu et al., 2023a).Convolutional neural networks demonstrate strong generalization capabilities, thus holding great potential for applications in disease detection (Liu et al., 2017). Full convolutional neural networks (Long et al., 2015) achieved pixel-level classification for the first time. Furthermore, the adaptability and transferability of various versions of DeepLab (Chen et al., 2017) and UNet (Ronneberger et al., 2015) have attracted a large number of researchers. A Survey of Deep Convolutional Neural Networks Applied for Prediction of Plant Leaf Diseases was raised. The article details summarize the advantages and disadvantages of different deep learning techniques for the agricultural sector (Dhaka et al., 2021). It is able to cope with the task of segmenting different diseases and is highly effective in dealing with simple environmental segmentation. In indoor environments, the segmentation of apple leaves and diseases performs well. However, in outdoor environments, the segmentation result on leaves and diseases is not satisfactory, due to the interference of light and overlapping.

With the application of relevant deep learning methods, more and more scholars have conducted extensive research on their application in apple leaf and disease segmentation (Wani et al., 2022). A fully automatic segmentation method for plant leaf images in complex environments was presented (Gao and Lin, 2019). A simple and effective semantic segmentation architecture based on a composite backbone, where OTSU was used to obtain a binary image (Yan et al., 2023b). CoAtNet integrates the attention mechanism of transformers into convolution operations for segmenting cotton leaves. Detection and classification of citrus leaf diseases based on MobileNet and self-structuring was introduced (Barman et al., 2020). The method incorporates channel attention (CA) mechanism into the ShuffleNet architecture and uses squeeze-and-excitation blocks as the CA mechanism to enhance the performance of ShuffleNet in grape leaf segmentation (Tang et al., 2020). The research of the mentioned has achieved good performance in leaf segmentation. However, in outdoor environments, diseases on the leaves cannot be accurately identified. Wang Y et al. put forward a lightweight single-stage network, which named as MGA-YOLO (Wang et al., 2022b). Based on the AlexNet model, Fu uses dilated convolution to extract coarse-grained features of diseases in the model, and extracts

apple leaf diseases at multiple scales (Fu et al., 2022). A two-stage DeepLabv3+ with adaptive loss is introduced, which incorporates receptive field block and reverse attention modules, and adjusts the speed of dilated convolutions in atrous spatial pyramid pooling (ASPP) for the segmentation of apple leaf images in complex scenes (Zhu et al., 2023a).The ALDD-YOLO lightweight apple leaf disease detection model has been raised, which introduces Mobilenet-v3s to compress model size (Xu and Wang, 2023). The EADD-YOLO by improving lightweight YOLOv5 was presented. It reconstructs the backbone network with lightweight inverted residual modules and introduces them into the network to reduce feature extraction and fusion, thereby improving the efficiency of segmenting leaves (Zhu et al., 2023b). The above papers all focus on the segmentation of apple leaves and diseases from the lightweighting, and they perform well in real-time detection on mobile devices, but there may be shortcomings in disease segmentation. A lightweight dense scale network (LDSNet) for corn leaf disease classification and recognition was proposed by Zeng Y et al (Zeng et al., 2022), using different expansion rate convolutions and attention fusion methods to improve the recognition of leaves and diseases. The apple leaf and disease segmentation recognition model based on a hybrid loss function and the Convolutional Block Attention Module (CBAM) was proposed (Zhang et al., 2023). The Swin Transformer is a network model for enhancing data and identifying cucumber leaf disease (Wang et al., 2022a). An enhanced TransUNet deep learning network was posed for recognizing rice leaves (Yan et al., 2023a). An improved DeepLabv3+ deep learning network structure for segmenting grape leaf black rot has been proposed (Yuan et al., 2022). The above method is suitable for single background and high-resolution apple leaf disease images, but it does not perform well in mixed environments.

Based on the above discussion, The advances of this paper are that it proposes an improved UNet that has residual attention and an atrous spatial convolutional pooling pyramid with weight compression loss. The primary task of the proposed network is to address the pixel-scale imbalance problem that exists in mixed scenes, especially when the network captures apple leaf and disease images. Accurate segmentation of leaves and diseases provides reliable technical means for precise analysis of apple health and helps to improve the intelligence and efficiency of orchard management.

The main novelties of this work are as follows:

- To overcome the imbalance in pixel representation between leaves and diseases in mixed environments, the weighted compression loss function includes a variable modulation factor before CE, enhancing the network's sensitivity to diseases during the training process.
- Res_CBAM is formed by integrating the residual structure with CBAM. The proposed Res_CBAM allows the network to capture multi-layered disease features and pay more attention to disease edge pixels.
- The improved ASPP structure allows the model to capture contextual information through multiscale receptive fields and different sampling rates, thereby enhancing its performance in the segmentation of diseased pixels.

The rest of the paper is structured as follows: In Section 2, the related materials and methods are presented. The material includes the obtained datasets and how to deal with them, while the methodology is a description of the details of the proposed RAAWC-UNet. The experimental results are analyzed, and the impact of network modules is discussed in Section 3. Section 4 summarizes the whole paper and makes recommendations for future research.

# 2 Materials and methods

## 2.1 Data processing

### 2.1.1 Image datasets

Images used in the study were collected from the Northwest Agriculture and Forestry University (Northwest A&F University) Baishui Apple Experiment Station (Baishui County, Weinan City, Shaanxi Province), Luochuan Apple Experiment Station (Luochuan County, Yan'an City, Shaanxi Province), and Qingcheng Apple Experiment Station (Qingyang County, Qingyang City, Gansu Province). The apple leaf and disease datasets produced by Northwest A&F University. Most of the image were taken on a sunny day with good light, and a few pictures were collected on a rainy day. It was taken at a distance of 10-15 cm using an ABM-50OGE/BB-500GE color digital camera and an Honor V10 mobile phone. The environmental conditions include sunny, cloudy, and rainy days, and different collection environments can further enhance the diversity of the datasets.

In complex field environments, the influence of leaf characterization factors, environmental factors, and leaf disease types makes the precise delineation of leaf diseases quite difficult. (1) The influence of leaf characteristic factors: the shape, color, texture and other characteristics of leaves will affect the division of leaf diseases. For example, the disease may change the color of the leaf or cause spots, and image processing algorithms can use these features to detect and segment diseased areas. (2) Influence of environmental factors: Environmental factors such as light conditions, humidity, and temperature can also affect the classification of leaf diseases. Poor lighting conditions or shadows can affect the image quality and thus the segmentation of the disease. Changes in humidity and temperature may also cause water droplets to appear on the surface of the leaf, thus affecting disease segmentation. (3) Types of leaf diseases: Different types of leaf diseases may be affected by different characteristics and environmental factors. For example, some diseases may cause an overall discoloration of the leaf, while others may cause spots only on specific areas of the leaf.

Figure 1 illustrates the influence of outdoor environments on leaves. Figure 1A demonstrates the effect of light and shadow on apple leaves, Figure 1B displays folding leaves at the edges of the leaf, Figure 1C reveals wrinkled edges of the leaf, and Figure 1D shows water droplets on the leaf. Apple leaf and diseases include blotch, brown spot, grey spot, rust, and mosaic disease. The total number of original images of the four apple leaf diseases is 1866, and the resolution of the images are all 512×512 pixels. Table 1

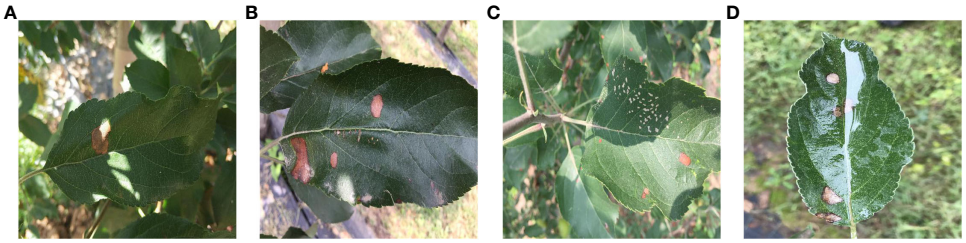



FIGURE 1
Apple leaves and diseases in outdoor environments. **(A)** Light effects. **(B)** Leaf edge folding. **(C)** wrinkled edges of the leaf. **(D)** Water droplet effects.

shows the indoor and outdoor distribution of four apple leaf disease pictures.

Figure 2 presents representative images of four apple leaf diseases, emphasizing the different types of diseases and distinctions. Each type of disease is selected from indoor and outdoor respectively to show the original image and label. Brown spot disease only has indoor images. The disease in Figure 2A is apple alternaria blotch spot, which mainly affects apple leaves, petioles, branches and fruits, producing very small brown spots on the young leaves of new shoots. The spots are often surrounded by a purple halo with clear margins. As the temperature rises, the spots can expand to 5-6 millimeters and become dark brown. The cause of alternaria blotch spot is mainly a strong virus strain of Streptomyces apples, which affects the normal growth of leaves, often resulting in twisted and wrinkled leaves, scorched parts of the disease. The disease in Figure 2B is brown spot, with a diameter of 3-5 millimeters. There are larger brown-green spots around the diseases, which are irregularly shaped, hence it is called green-brown disease. The pathogen of brown spot disease is caused by bivalve infestation, which leads to early defoliation of apple trees, reduces photosynthesis of apple leaves, causes malnutrition, and reduces the economic benefits of fruit growers. The disease in Figure 2C is a grey spot, usually 2-6 mm in diameter, with clear, reddish-brown edges that turn grey later, with small black dots scattered in the center, and is mainly caused by the pear leaf spot fungus. Leaves caused by this disease usually don't turn yellow and fall off, but severely affected leaves will scorch, which in turn affects apple yield. The disease shown in Figure 2D is rust, which is initially orange-red in color and consists of small dots about 1 to 2 mm in diameter. If not controlled, the spots will grow larger and darker in the middle until they become black dots. At this time, the outermost ring of the spot is relatively light in color, and the pathogen severely damages the young fruits, resulting in the development of bad fruits.

Figure 3 shows the box plots of each category of the ALDD datasets, respectively analyzing the characteristics and differences of the four types of disease datasets under indoor and outdoor conditions. From Figures 3A, B, it can be seen that the RGB pixel values of the same type of disease are similar, the distribution range shows a decreasing trend, and the outliers of the disease are few. Due to the large difference in the pixels of the background, the RGB channel pixel values of the indoor are higher than those of the outdoor. Figure 3C is Brown spot, which only has indoor disease, reflecting that the pixels of the leaves and disease of brown spot are relatively complex, while the pixels of the background are relatively concentrated. Figures 3D, E are grey spots, and Figures 3C, D have the same mean value of the background pixels, and by looking at the original datasets, it can be noticed that the indoor background of the brown spot and the grey spot are close to each other. Figures 3F, G are Rust, the background category pixel range in Figure 3F is smaller than that in Figure 3G and relatively more concentrated.

### 2.1.2 Image preprocessing

Firstly, preprocessing operations were performed on the images. The data on brown spots, grey spots, and rust spots in the outdoor environment were selected from the datasets. The leaves and diseases were labeled using Photoshop and Labelme, respectively, under the guidance of an apple leaf disease recognition expert. Photoshop was used to quickly mark leaves and background on apple leaves using the magic wand tool. The marking of diseases using Labelme allows for precise labeling of diseases on leaves. The final labels are saved in PNG format, which makes the labeling more precise and efficient. The labeling style is shown in Figure 4. Secondly, in order to avoid the overfitting problem in the later network training, improve the generalization ability of the model, and enhance diversity, the image was enhanced to simulate the complexity of the outdoor environment. Such as light intensity, light color temperature, and shadow effect. As shown in Figure 5, there are two main methods for image enhancement: (1) A geometric transformation of the image, which randomly flips and crops the original image horizontally and vertically. (2) Pixel

TABLE 1 Amount of data on apple leaf disease species.

| | Alternaria blotch | Brown spot | Grey spot | Rust | Total |
|---|---|---|---|---|---|
| indoor | 125 | 277 | 177 | 424 | 1003 |
| outdoor | 270 | – | 167 | 51 | 488 |
| total | 395 | 277 | 344 | 475 | 1491 |

10.3389/fpls.2024.1305358

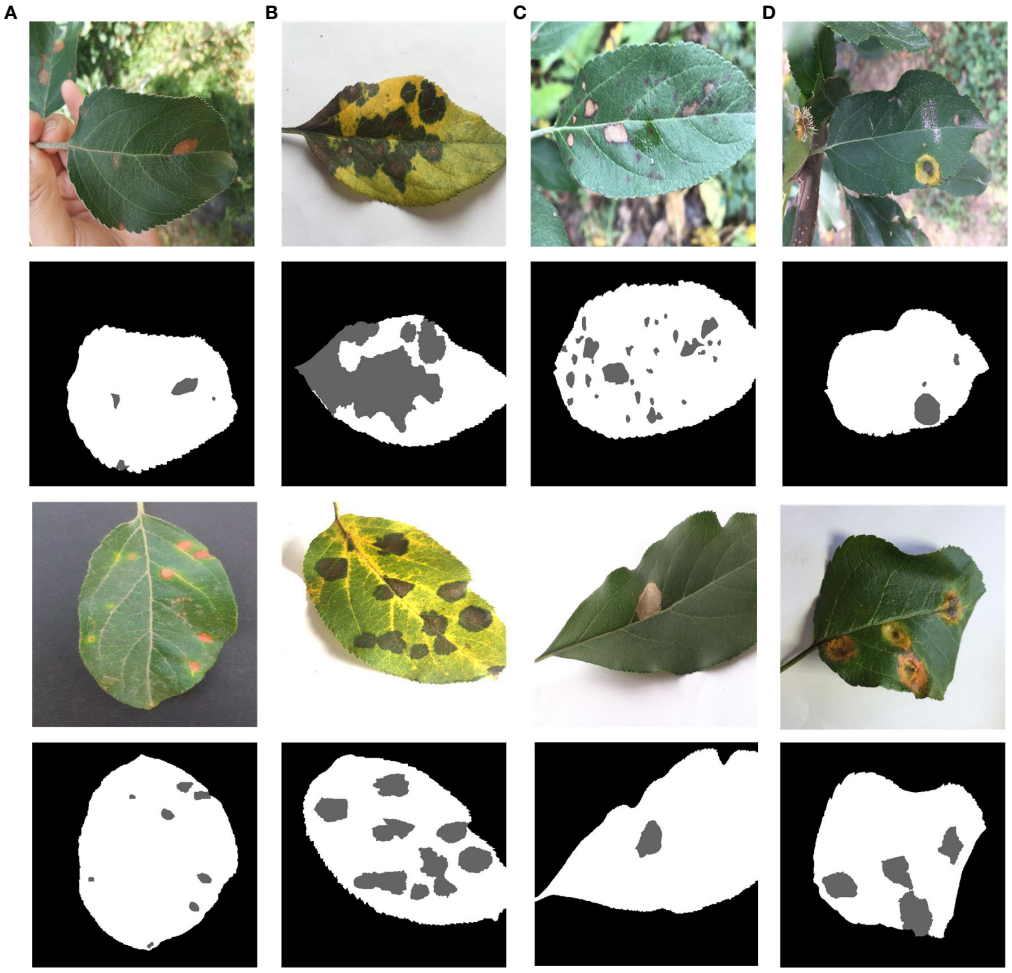



FIGURE 2
Four types of apple leaves and diseases. **(A)** Alternaria Blotch. **(B)** Brown Spot. **(C)** Grey Spot. **(D)** Rust.

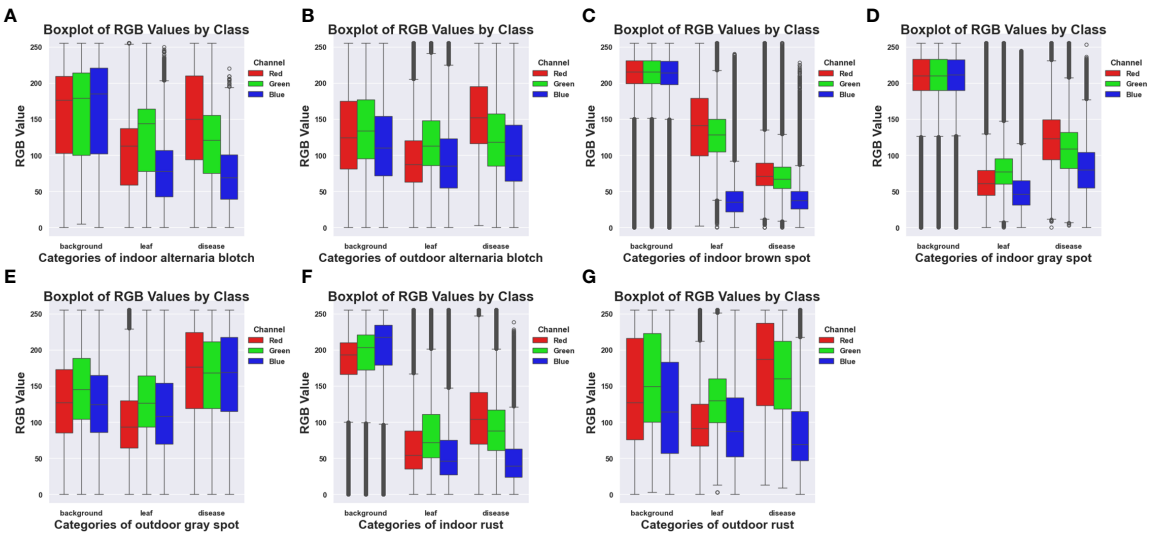



FIGURE 3
Box plots of RGB pixels of different diseases for the ALDD datasets. **(A)** Indoor Alternaria Blotch. **(B)** Outdoor Alternaria Blotch. **(C)** Indoor Brown Spot. **(D)** Indoor Grey Spot. **(E)** Outdoor Grey Spot. **(F)** Indoor Rust. **(G)** Outdoor Rust.

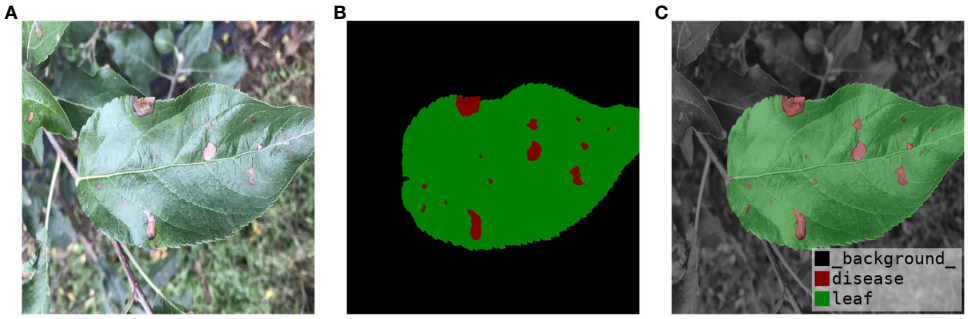



FIGURE 4
Image annotation. **(A)** Original Image. **(B)** Label Image. **(C)** Label Visualization.

transformation of the image, where the image brightness is randomly varied by 0.5–1.2 times, the contrast is randomly varied by 0.5–2.5 times, and the chromaticity is randomly varied by 0.5–2.5 times.

Then, color augmentation were applied to the indoor apple leaf datasets, increasing it from 1,003 images to 2,006 images. The outdoor datasets was expanded by randomly changing brightness, contrast, and color. The outdoor apple datasets was expanded from 488 to 1,952 sheets. This ensured that the indoor-to-outdoor ratio was as close to 1:1 as possible, thus reducing experimental error. The major apple leaf diseases included in the datasets were alternaria, brown spot, grey spot, and rust. Figure 5 illustrates the five enhancement methods for apple brown spots as an example.

## 2.2 The proposed method

In this section, we present the proposed framework called RAAWC-UNet and the components of each module. Residual convolutional block attention module and atrous spatial pyramid pooling improved UNet with weighted compression loss is simplified as RAAWC-UNet. The proposed network mainly consists of two improved modules and the proposed loss function. Residual and CBAM are combined into Res_CBAM, and ASPP is improved into a module suitable for leaf datasets segmentation with the proposed weight compression loss.

### 2.2.1 RAAWC-UNet framework

The outdoor image is affected by light, water droplets, bad weather, and the overlap of some outdoor leaves and target leaves,

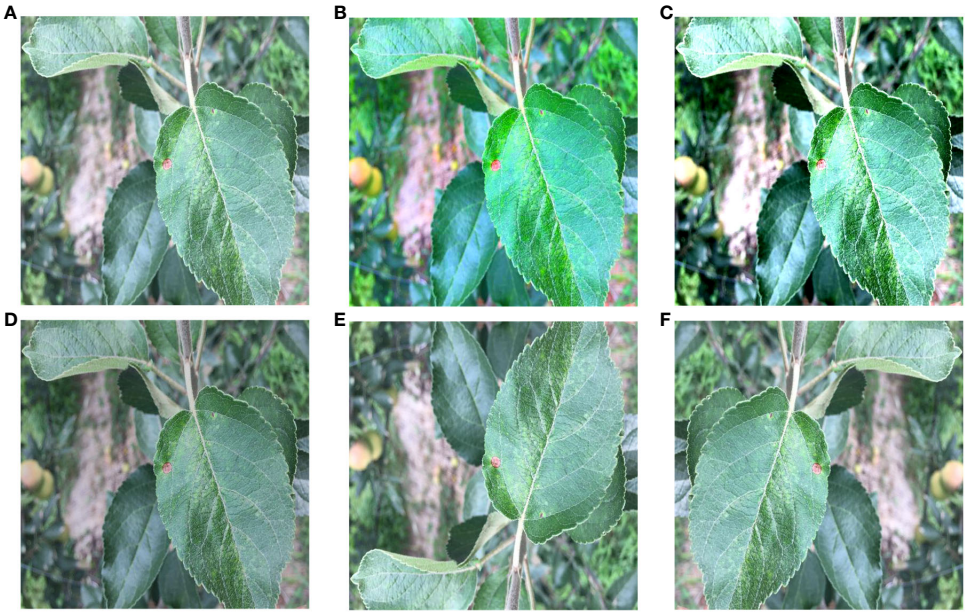



FIGURE 5
Image Enhancement. **(A)** Original image. **(B)** Color change. **(C)** Contrast change. **(D)** Brightness change. **(E)** Vertical flip. **(F)** Horizontal flip.

which leads to the difficulty of segmenting outdoor leaves and diseases. UNet performs well when dealing with single background leaf and disease segments. However, its performance is slightly lacking when coping with small target segments, such as diseases on apple leaves. On the one hand, the reduction of feature maps in UNet's downsampling process can decrease the amount of information for small targets. On the other hand, during the decoding process of UNet, information is recovered from lower-level feature maps. However, this process may lose some global information, leading to the loss of context information in different regions. Therefore, to overcome this difficulty, more powerful feature extraction and detail preservation mechanisms need to be introduced into UNet to better handle these challenging small target segmentation tasks.

In order to improve the accuracy and robustness of apple leaf and disease segmentation, this study introduces an improved model on UNet, namely RAAWC-UNet. In this model, we use UNet as the backbone network and make a series of improvements based on it to better adapt to the challenging requirements of small target segmentation.

In the RAAWC-UNet model, the main improvement is in the convolutional blocks in the encoder part. The Residual and CBAM modules were introduced and integrated to form the Res_CBAM module. The Residual structure not only maintains the stable transmission of features but also helps to retain the detailed information in the network. It enables the network to be better adapted to the segmentation of small target areas, which are diseases. The CBAM makes the network more attentive to the pixel regions in the image that are decisive for classification while ignoring insignificant regions in both the channel and spatial dimensions. The merged Res_CBAM blocks help to integrate high-level and low-level semantic features, thus effectively combining detailed information with the global context. The model can better understand the features of the diseased region and improve the accuracy and robustness of segmentation.

In addition, we replace the last downsampling layer with ASPP. The ASPP module can cover image features of different sizes within the perceptual region by utilizing different hollow rates of multiscale convolutional operations. It captures fine-grained information in leaf and disease images while also capturing a wider range of contextual relationships. Applying the modified ASPP structure to the last layer of downsampling not only improves ability to segment but also enhances generalization to a wide range of complex field scenarios. Overall, the improvement greatly enriches the model's ability to understand the disease, thus enabling it to exhibit higher accuracy and robustness in pixel-level image segmentation. The network architecture of the proposed RAAWC-UNet is shown in Figure 6. The specific parameters of the proposed network structure are shown in Table 2. The proposed model can be reconstructed based on the structure of the proposed network diagram and the detailed operational parameters in the table.

### 2.2.2 Res_CBAM module

Residual module is a type of building block commonly used in deep convolutional neural networks, originally proposed by residual network in 2015 (Yang et al., 2023). As shown in Figure 7, the module aims to solve the problem of vanishing and exploding gradients during training, and the skipping mechanism of the residual module allows UNet to capture feature information at different levels. For apple leaf and disease segmentation, different levels of features can provide rich contextual information to accurately distinguish between diseased and normal regions. UNet can combine low-level detailed features with high-level semantic features to obtain better segmentation results. In

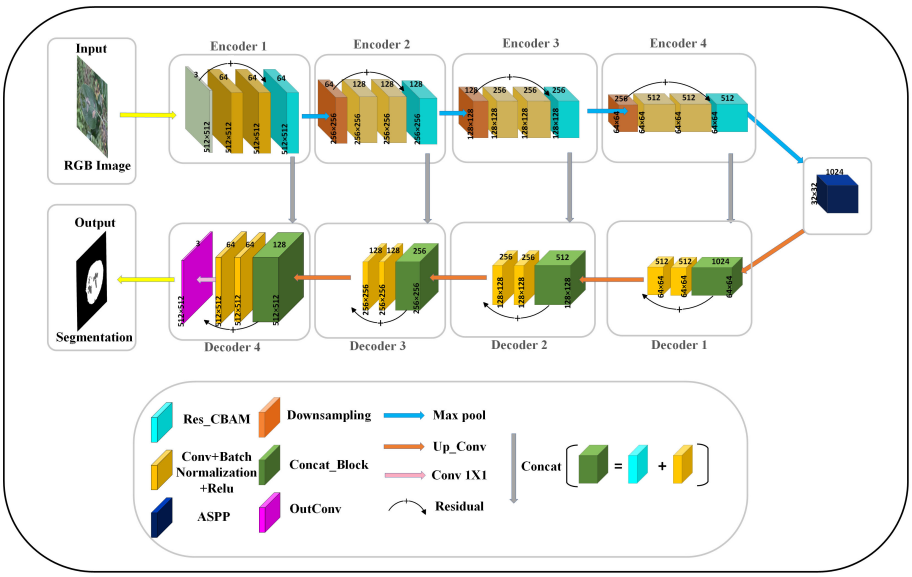



FIGURE 6
The general architecture of RAAWC-UNet with 4 encoders and 4 decoders, encoding and decoding are connected by a modified ASPP. The downsampling block, the convolutional block and Res_CBAM form an encoder, and the decoder consists of a convolutional block connected by Concat and residuals.

TABLE 2 Parameters of each module in the proposed model.

| Layer | Operation | Size | Output |
|---|---|---|---|
| Encoder1 | Input<br>Conv×2<br>Residual<br>Res_CBAM | 3×512×512<br>3×3, 3×3<br>3×3, 1×1<br>1×1, 7×7 | -<br>64×512×512<br>64×512×512<br>64×512×512 |
| Encoder2 | Downsampling<br>Conv×2<br>Residual<br>Res_CBAM | 2×2, stride=2<br>3×3, 3×3<br>3×3, 1×1<br>1×1, 7×7 | 64×256×256<br>128×256×256<br>128×256×256<br>128×256×256 |
| Encoder3 | Downsampling<br>Conv×2<br>Residual<br>Res_CBAM | 2×2, stride=2<br>3×3, 3×3<br>3×3, 1×1<br>1×1, 7×7 | 128×128×128<br>256×128×128<br>256×128×128<br>256×128×128 |
| Encoder4 | Downsampling<br>Conv×2<br>Residual<br>Res_CBAM | 2×2, stride=2<br>3×3, 3×3<br>3×3, 1×1<br>1×1, 7×7 | 256×64×64<br>512×64×64<br>512×64×64<br>512×64×64 |
| ASPP | Dilation_Conv<br>Downsampling | Rate: 6, 12, 18<br>2×2, stride=2 | 1024×64×64<br>1024×32×32 |
| Decoder1 | Up_Conv<br>Concat<br>Conv×2<br>Residual | Scale_factor=2<br>-<br>3×3, 3×3<br>3×3, 1×1 | 512×64×64<br>1024×64×64<br>512×64×64<br>512×64×64 |
| Decoder2 | Up_Conv<br>Concat<br>Conv×2<br>Residual | Scale_factor=2<br>-<br>3×3, 3×3<br>3×3, 1×1 | 256×128×128<br>512×128×128<br>256×128×128<br>256×128×128 |
| Decoder3 | Up_Conv<br>Concat<br>Conv×2<br>Residual | Scale_factor=2<br>-<br>3×3, 3×3<br>3×3, 1×1 | 128×256×256<br>256×256×256<br>128×256×256<br>128×256×256 |
| Decoder4 | Up_Conv<br>Concat<br>Conv×2<br>Residual<br>Output | Scale_factor=2<br>-<br>3×3, 3×3<br>3×3, 1×1<br>1×1 | 64×512×512<br>128×512×512<br>64×512×512<br>64×512×512<br>3×512×512 |

addition, since leaf and disease segmentation tasks are very sensitive to edge information, the residual can provide better edge retention.

In this study, the UNet encoder adds the residual structure. The residual network was proposed by (He et al., 2016). To ensure that the size of the feature map obtained after two convolution operations is the same as the size of the skip connection output, we use a 1 × 1 convolution kernel to tune the channels. Thus, if the input is represented by $x$, the modified Res output is represented by a function: $f(x) + g(x)$.

The following are typical representatives of channel and spatial attention:

(1) The Squeeze-and-Excitation (SE) block (Deng et al., 2023) is a network module proposed in 2018. The SE module first averages the input feature maps globally and secondly compresses the information from each channel into a scalar value. Finally, it is processed using the nonlinear activation function ReLU. The SE fundamentally removes the weights of each channel.

(2) The Efficient Channel Attention (ECA) block aims to improve the computational efficiency of channel attention (Yuan et al., 2022). The core of ECA is to adjust the importance of channel features by introducing an adaptive weight on each channel. A different convolutional kernel is applied on each channel to adaptively compute the attention weights for each channel, and these weights need to be normalized. To create a weighted feature map, the original feature map is dot-multiplied with the normalized weights. The weighting enhances the attention to the important channels and thus improves the discriminative properties of the features. The ECA is not designed for global average pooling, so it is more computationally efficient than SE.

(3) The CBAM is the attention module proposed by (Woo et al., 2018). There are two main directions for modules that improve network performance: channels and spatial attention. The channel module is similar to the channel of SE. The spatial module aims to weight different spatial locations to capture key parts of the image. All of this information is then fused through a convolutional layer to generate spatial attention weights.

In order to make the network more focused on the disease and ignore irrelevant background information. The CBAM is added to the downsampling of the network. The CBAM consists of channel and spatial attention modules. The channel attention module reinforces the channel features of the disease and adaptively selects the channel features related to the disease to better capture the boundary of the disease. The spatial attention module focuses on the spatial dimension and extracts key features. Feature maps are

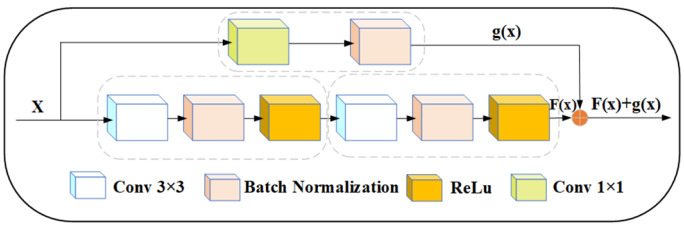



FIGURE 7
Modified residual structure, x represents the input, and x goes into two convolution blocks and a 1×1 convolution block, respectively. The output F(x) and g(x) tensors have the same dimension, and then F(x) and g(x) are summed up.

generated through pooling operations to highlight the spatial locations of leaves and diseases. The spatial attention module helps the model better adapt to different lighting and shading situations. Figure 8 illustrates the channel and spatial attention module. The CBAM module combines channel and spatial information to enhance the expressive and sensory capabilities of the model. It improves the performance, generalization, and interpretability of the model. This provides important applications for apple leaf and disease segmentation.

Given a feature map $T \in \mathbb{R}^{C\times1\times1}$ as input, the 1D channel feature map $P_c \in \mathbb{R}^{C\times1\times1}$ and the 2D spatial feature map $P_s \in \mathbb{R}^{C\times1\times1}$, are derived sequentially (Equations 1–3):

$$T' = P_c(T) \otimes T, \tag{1}$$

$$T'' = P_s(T') \otimes T', \tag{2}$$

$$O = T + T' + T'' \tag{3}$$

$\otimes$ represents the multiplication between elements. The essence is to perform a broadcasting mechanism. $T''$ is the output after passing through channel attention and spatial attention. $O$ is the output after residual attention. The Res_CBAM involves connecting the features before attention, the features enhanced by channel attention, and the features with spatial attention on top of channel attention through residual connections. The Res_CBAM example diagram is shown in Figure 9.

The feature map fused different levels of feature information through the residual structure. The feature map improves the accuracy of apple leaf disease segmentation through channel attention and spatial attention, emphasizes the key features, and enhances the segmentation results.

### 2.2.3 Modified ASPP module

ASPP is commonly used in image semantic segmentation tasks (Wang et al., 2021). The ASPP module is designed to help the network perceive information from different sensory fields. It operates with multiple parallel convolutional branches with different sampling rates to extend the receptive field and capture information at different scales without introducing additional parameters. The common sampling rates include 3, 6, 12, 18, etc., with larger sampling rates providing a wider range of contextual information and smaller sampling rates retaining more details. The output of the ASPP module is usually a splice or overlay of the outputs of the branches. The model can then acquire contextual information at different scales to better understand the image content and perform accurate segmentation. The modified ASPP structure removes the Conv1 × 1 and pooling operations and only keeps three Conv3 × 3 with different expansion rates. Therefore, it is possible for the modified ASPP to significantly reduce the number of parameters and the amount of float computation while still maintaining the ability to efficiently extract leaf and disease features. The modified ASPP structure is shown in Figure 10.

### 2.2.4 Weight compression loss

Cross entropy (CE) is a commonly used loss function in segmentation networks to measure the difference between model predictions and true segmentation labels. However, CE loss has limitations when dealing with apple leaf and disease segmentation problems. On one hand, CE loss cannot effectively deal with the category imbalance problem, which may cause the model to be biased towards more healthy pixels. On the other hand, CE loss cannot capture spatial continuity, and diseased regions may be disconnected or blurred.

CE loss is the classical loss function in semantic segmentation (Jadon, 2020), which is defined in Equation 4:

$$L_{ce}(y,p) = \begin{cases} -log\ (p) & if\ \ y = 1 \\ -log(1-p) & if\ \ y = 0 \end{cases} \tag{4}$$

The $p$ represents probability between the predicted value and the true value. To address the drawback of CE loss, the proposed loss function primarily deals with the issue of foreground and background pixel imbalance. We have named it WC_Loss. The

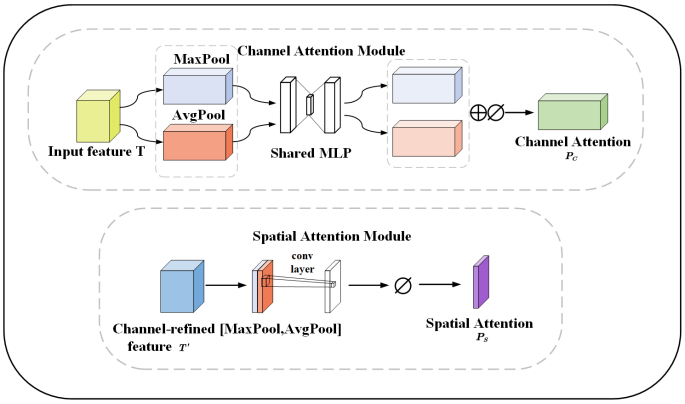



FIGURE 8
The channel and spatial attention module, MaxPool and AvgPool represent global maximum pooling and average pooling respectively. Shared MLP represents shared fully connected layers. ⊕ represents addition, ⊘ stands for sigmoid operation. [MaxPool, AvgPool] represents concatenation of the global maximum pooling and average pooling.

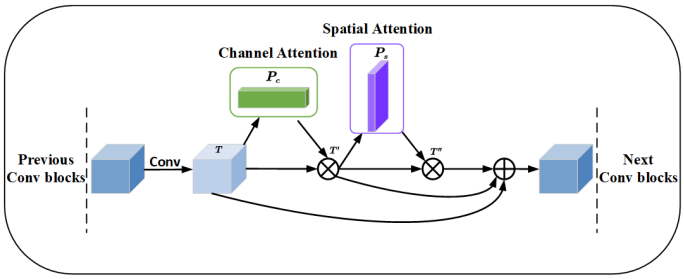



FIGURE 9
Residual convolutional block attention module, where ⊗ denotes the corresponding element multiplication and ⊕ represents the element addition, before the multiplication operation, the channel attention and spatial attention need to be broadcast according to the spatial dimension and channel dimension respectively.

Loss function is defined in Equation 5:

$$WCL_{ce}(y,p) = \begin{cases} -(1-arctan(p))^{\gamma} log(p) \; if \; y=1 \\ -(arctan(p))^{\gamma} log(1-p) \; if \; y=0 \end{cases} \quad (5)$$

The $arctan(p)$ maps the input probability values to $(0,\pi/4)$, while $[1 - arctan(p)]$ is mapping the probabilities to $(1 - \pi/4,1)$. In contrast to the CE loss function, the weights in front of the CE loss are always 1 and are not elastic. $[1 - \arctan(p)]^2$ has the advantage of shortening the mapping range and adaptively adjusting the size of the front weights. The adjusted values of the weight compression loss function for mixed environments, leaf and disease pixel regions will be more continuous, helping the model to adapt more smoothly to difficult and easy samples during training.

For the apple leaf disease segmentation task, the proposed loss function WC_Loss has the following advantages: Firstly, in the apple leaf disease segmentation task, the pixel ratios of healthy and diseased leaves are unbalanced. The proposed loss function reduces the weight of easy-to-categorize samples and pays more attention to difficult-to-categorize samples, which effectively handles the imbalance between leaf and disease categories. Secondly, by introducing an adjustable hyperparameter $\gamma$, WC_Loss is able to focus on the difficult samples and improve the learning ability for apple leaf and disease samples. When $\gamma = 0$, WC_Loss becomes CE_Loss. Thirdly, the value of the modulation factor $[1-arctan(p)]$ is determined by the probability $p$, which decreases as the probability $p$ increases. When $p$ increases, the pixels representing the leaf and the background are easy to classify. $\gamma$ increases, the modulation factor $[1 - arctan(p)]$ compresses. It is not the larger $\gamma$ the better the accuracy of segmentation, but $\gamma$ has to find a suitable value. Finally, the contribution to the overall loss reduces the effect of apple leaf samples and improves the accuracy of disease category classification. In conclusion, WC_Loss is a loss function that can effectively solve the problem of a serious imbalance in the proportion of leaf and disease pixels and improve the generalization ability of the model.

As shown in Figure 11, five different hyperparameters $\gamma$ are set. As the value of $\gamma$ increases, the possibility of fixing the probability of true labeling, the strength of the modulation factor compression increases.

The curve of the loss function shows that as the confidence level $p$ increases, the loss value becomes smaller. According to the experimental results in Subsection 3.5, it is found that the best performance is achieved with $\gamma = 2$, combining leaf segmentation and disease segmentation.

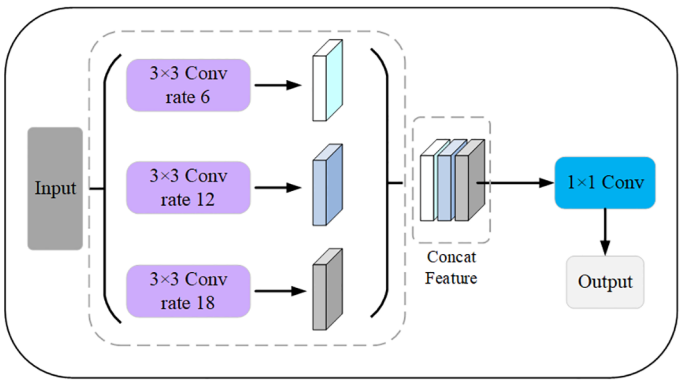



FIGURE 10
Modified ASPP module. The expansion factors of the 3 pooled pyramids are 6, 12, and 18, respectively, which are concatenated after passing through the feature layer of the pooled pyramid, and then the number of channels is varied using a 1×1 convolution.

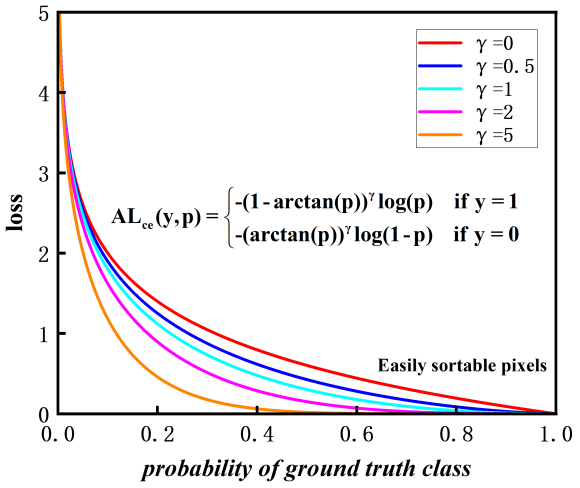



FIGURE 11
Comparison of weight compression loss and hyperparameter γ.

# 3 Results and discussion

In this section, the experimental settings, evaluation metrics, and experimental results will be described in detail. The chapter includes the tuning of the loss functions of the participating designs, baseline comparisons, and ablation experiments containing individual modules. Then, the advantages and limitations of the proposed model are discussed.

## 3.1 Experimental setup

The experimental software environment is PyTorch 2.0.0 and Python 3.9. The server configuration is an Intel(R) Core (TM) i9-10900K CPU @ 3.70 GHz, 128.0 GB RAM, NVIDIA Quadro RTX 8000, CUDA 11.7. The server system is Ubuntu 18.04, and the hyperparameters are determined as follows: The initial learning rate is set to 1e-4, the number of epochs is 200, the batchsize is 4, and the model is optimized by the SGD optimizer. At the same time, the learning rate decay strategy is used to make the late gradient descent smooth and easy to converge, and the algorithm is easy to approximate the optimal solution. The experiment is done three times, and the distance of each evaluation measure is controlled at 0.0001–0.0005, which is regarded as the model being close to the optimum. The hyperparameters are set as shown in Table 3.

In the experiments of this study, the initialization images were used with a size of 512×512×3. For effective training and evaluation. The datasets are divided into training sets, validation sets, and test sets, which are in the ratio of 6: 2: 2 and it is ensured that the datasets cover four different disease scenarios. The training sets contains 2374 images, and the validation and test sets contain 792 images each. With this division, each type of datasets contains a variety of diseases to ensure the comprehensiveness and robustness of the model.

This normalization operation not only helps to speed up the convergence of the neural network and improve generalization ability, but also effectively deals with the problem of vanishing gradients. This stage is crucial to the entire research process since it helps the model comprehend the characteristics of the images and produce predictions that are more precise.

TABLE 3 Hyperparameter settings.

| Hyperparameter | Values |
|---|---|
| Num_classes | 3 |
| Batch Size | 4 |
| Epochs | 200 |
| Optimizer | SGD |
| Learning_rate | 1e-4 |
| Momentum | 0.9 |

Figure 12 illustrates the training indicators for the RAAWC-UNet model. The loss curves for the training and validation sets are shown in Figure 12A. Figure 12A shows the loss curves for the training and validation sets. It is clear from Figure 12A that the loss curves for both training and validation sets show a satisfactory trend. This indicates that the model performs well during the training process. The accuracy curves for the three categories of background, leaf, and disease are shown in Figure 12B. As seen in Figure 12B, the model performs admirably in terms of handling the three categories of complex background, leaves, and diseases. Figure 12C demonstrates the variation of mean Intersection over Union (mIoU). The model also produces good results in terms of mIoU, as can be seen in Figure 12C.

In summary, Figure 12 visualizes the performance of the RAAWC-UNet model during the training process. The good performance of the loss curves, accuracy and mIoU further validates the correctness of the model for apple leaf and disease segmentation tasks.

## 3.2 Evaluation indicators

To evaluate the performance of apple leaf and disease segmentation network models in complex environments. Three types of evaluation metrics are used, which are overall performance metrics, pixel-level metrics and additional metrics for leaves and diseases.

Firstly, the overall performance evaluation metrics for the leaf and disease segmentation task were used as Accuracy (Equation 6). Accuracy is the ratio between the number of samples correctly predicted by the model and the total number of samples and is used to measure the overall predictive accuracy of the model over the entire datasets.

$$Acc = \frac{TP + TN}{TP + FN + FP + TN} \quad (6)$$

Secondly, the pixel-level metrics include Precision (Equation 7), Recall (Equation 8), F1 Score (Equation 9), mPA (Equation 10) and mIoU (Equation 11). Precision is a measure of the proportion of pixels predicted to be in the positive category by the model over all the pixels labelled as positive. Recall indicates the proportion of

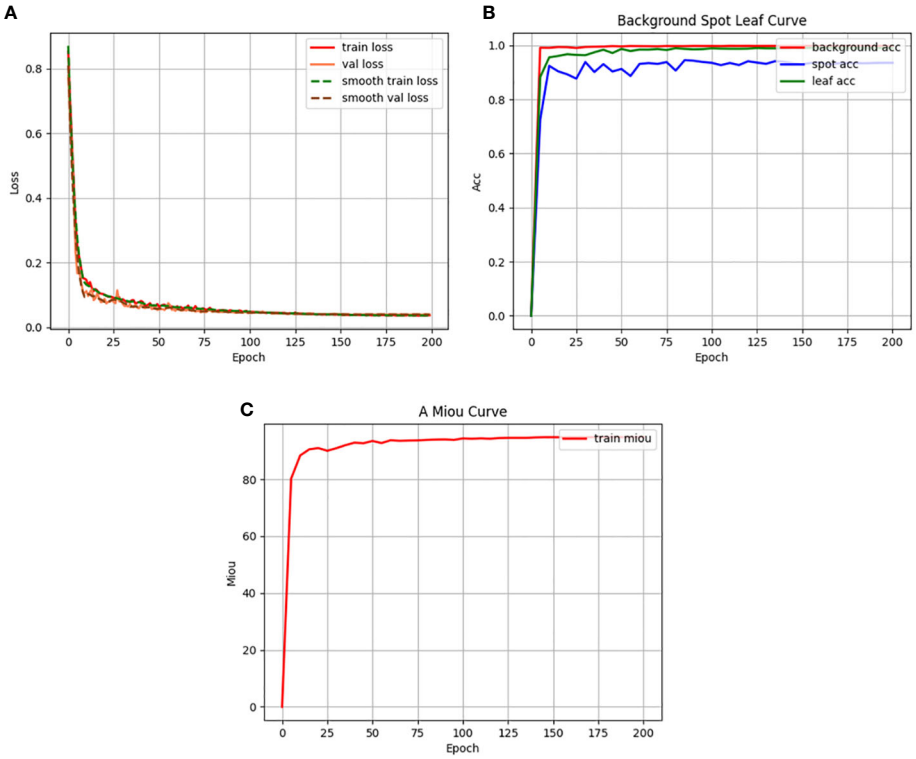



FIGURE 12
Training indicators. **(A)** Loss function curve. **(B)** Accuracy curve. **(C)** Mean intersection over union curve.

pixels correctly marked as positive by the model over all the pixels in the actual positive category. F1 Score is a performance metric that combines Precision and Recall. It aims to balance the accuracy and the percentage coverage of the model.

$$P = \frac{TP}{TP + FP} \tag{7}$$

$$R = \frac{TP}{TP + FN} \tag{8}$$

$$F_1 = \frac{2*P*R}{P + R} \tag{9}$$

TP in the formula indicates the number of samples correctly identified as positive category, TN is the number of samples correctly identified as negative category, FP is the number of samples incorrectly identified as positive category, and FN indicates the number of samples incorrectly identified as negative category.

The mPA of accuracy category is calculated for each category at the pixel level and then averaged across all categories of accuracy. The mIoU is an important measure of the model's segmentation performance, and its size directly reflects the model's performance for pixel-level segmentation tasks. Where $k$ denotes the total number of categories, *Pii* denotes the number of pixels predicted to be category $i$ and truly belonging to category $i$, *Pij* represents the number of pixels predicted to be category $i$ but really correspond to category $j$, and *Pij* indicates the number of pixels predicted to be category j but actually refer to category i. The mIoU is the pixel accuracy predicted by the model on each category and averages the results across all categories to assess the metrics. $\sum_{j=0}^{k} P_{ij}$ means the number of pixels of category $_i$ that are partitioned into a total of $k$ categories, and $\sum_{j=0}^{k} P_{ji}$ means the number of pixels of category $j$ that are redundantly partitioned into the i-th category.

$$mPA = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}P_{ji}}, i = 0, 1, 2, \ldots, k \tag{10}$$

$$mIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}P_{ij} + \sum_{j=0}^{k}P_{ji} - P_{ii}}, i = 0, 1, 2, \ldots, k \tag{11}$$

Finally, the additional metric uses Cohen's kappa coefficient (Equation 12). It is used in statistics to evaluate the consistency of a multicategorical accuracy model for background, leaf, and disease. The *po* is the empirical probability of a label assigned to any sample, and *pe* is the expected consistency between two annotators when they randomly assign labels.

$$\kappa = (p_o - p_e)/(1 - p_e) \tag{12}$$

## 3.3 Discussion of different attention modules

In complex outdoor environments, leaf images are easily affected by environmental factors. In order to accurately capture the texture boundary information of leaf diseases. The spatial

attention module of CBAM helps to extract information at different scales in the image, better capturing the boundaries and spatial distribution of the diseases. SE and ECA segment the leaves and diseases only from the channel. CBAM has strong performance in two dimensions, it performs better in segmenting apple leaves and diseases.

The comparison results of objective indicators using different attentional mechanisms are shown in Supplementary Table 1. It can be seen that the addition of the SE, ECA and CBAM modules increases the overall accuracy of the network on the validation sets by 0.36%, 0.31%, and 0.39%, and the IoUs of the apples leaf pixel segmentation on the test sets by 0.83%, 0.84%, and 1.21%, and that of the disease pixel segmentation by 1.22%, 1.57%, and 3.9%. The mPA and mean precision on the test sets increased by 1.44%, 1.53%, 1.7% and 0.83%, 0.81%, 0.96%. Supplementary Table 1 reveal that the encoder needs to consider not only channel attention but also spatial attention during feature extraction. It shows that the attention mechanism of the model plays a key role in both channel and spatial aspects in a particular task.

Supplementary Figure 1 shows the results of image segmentation using different attention modules. By comparing Supplementary Figures 1C–F, it can be observed that the ability of the model to extract small lesions is enhanced with the addition of the attention module. The comparison of Supplementary Figures 1D–F shows that SE and ECA attention modules only have channel attention to extract a small number of minor illnesses pixels. CBAM has both channel and spatial attention and will be more comprehensive in extracting small diseases. According to the experimental results, CBAM performs better in apple leaf and disease segmentation compared to the other two attention modules. The objective results presented in Supplementary Table 1, as well as the multiple subjective validations in Supplementary Figure 1, indicate the relative superior performance of CBAM in the segmentation tasks of leaves and diseases. The effectiveness of CBAM in improving the model's ability to focus on image details and specific regions is emphasized.

## 3.4 Discussion on hyperparametric optimizers and learning rate

Through the comparative analysis of three sets of experiments, the best choices of learning rate and optimizer were determined. The results of the experiments are shown in Supplementary Table 2. In the experiments, 1e-4 and 5e-4 were tried as two different learning rates, while two optimizers, SGD and Adam, were used. The experimental results show that leaf and disease segmentation perform better when the learning rate is set to 1e-4. Although the Adam optimizer can achieve faster model convergence, the stability of the results is poor, so SGD is finally chosen as the optimizer.

## 3.5 Discussion on the hyperparameter γ

To verify the effectiveness of the weight compression loss, we performed integration using Res_CBAM and a modified ASPP to validate the equation of the modulation factors. The experiment was set up for five groups and the hyperparameter $\gamma$ was changed regularly to find the most appropriate hyperparameter configuration.

Theoretically, a larger value of $\gamma$ indicates that the model pays more attention to the difficult-to-categorize disease pixels and ignores the samples of relatively easy-to-categorize leaves during the training process. But in practical applications, as shown in Supplementary Table 3, when the $\gamma$ value is too large, it often leads to the occurrence of overfitting phenomenon, which negatively affects the model's performance. Therefore, when weighing the importance of difficult samples and the generalization ability of the model, it is crucial to choose the $\gamma$ value reasonably. From Supplementary Table 3, it can be seen that when the $\gamma$ value is 2, the model performs optimally.

## 3.6 Ablation experiments

In this subsection, five sets of experiments are designed to validate the high accuracy of the proposed network. The control variable method is adopted to test the effectiveness of each module in the network to extract leaf and disease.

As shown in Supplementary Table 4, the baseline for Test 1 is UNet. For Test 2, the CE loss used in Test 1 is replaced with the proposed weight compression loss. Tests 3 and 4, based on Test 2, add the Res_CBAM module and the modified ASPP module, respectively. Test 5 integrates Res_CBAM and revised ASPP, and the loss function uses our proposed weight compression loss.

Comparing test 1 and test 2 it can be observed that the IoU of leaves and diseases increased by 0.54% and 3.96% respectively, on the test sets after using the weighted compression loss function. The overall mPA and mean precision increased by 1.51% and 0.65%. It can be seen that the imbalance in the proportion of leaf and disease pixels can be effectively dealt with after using the weighted compression loss. Comparison between Test 2 and Test 3, the IoU of leaf and disease increased by 0.84% and 0.94%, and the overall mPA and mean precision increased by 0.26% and 0.31%, which shows that Res_CBAM improves the model's ability to express and perceive leaf and disease. Comparing Test 4 and Test 2, the last layer of the model is replaced with a modified ASPP, the IoU for leaves and diseases increased by 1.31% and 1.38%, and the overall mPA and mean precision increased by 0.38% and 0.54%. It enables the model to integrate different scales of semantic information and improves the adaptive ability of the network. Test 5 was compared with Test 4 and Test 2. After adding Res_CBAM and modified ASPP, the model's IoU values for leaves and diseases were higher than when only Res_CBAM or ASPP were added.

Supplementary Figure 2 show the original and ground truth images. The weight compression loss can be observed by comparing Supplementary Figures 2C, D, which improves the extraction of disease information. Based on the segmentation results of Supplementary Figures 2E–G, the following conclusion can be drawn: The model with the blend Res_CBAM and modified ASPP module performs better in leaf and disease segmentation extracted from the outdoor environment.

To better understand the performance of the proposed model, the model was validated on the test sets and the confusion matrix was plotted, which can intuitively show the prediction results of the classification model on each category. Supplementary Figure 3 presents the confusion matrix for the three types of pixel value percentages, where it can be observed that the percentage of pixels for the disease category is lower than that of the background and leaves. In the confusion matrix, the disease category Ground Truth was incorrectly predicted as leaves with 7.72% overall. It indicates that there is still some difficulty in segmenting the small size of the disease.

Supplementary Table 5 objectively shows the accuracy of the proposed network in each category of pixels on the four disease test sets. From Supplementary Table 5, it can be observed that the Brown Spot segmentation metrics are higher than Alternaria Blotch, Grey Spot, and Rust in all categories of pixels because Brown Spot exists only in indoor environments without the disturbance of outdoor environments. In contrast, Alternaria Blotch and Grey Spot segmentation metrics are relatively low, not only with the interference of background factors, but also with smaller and more diseased pixels.

## 3.7 Comparison of results for different segmentation networks

To validate the performance of the proposed network for apple leaf and disease segmentation in mixed environments, the results of the proposed network were compared with those of other different networks on the same datasets. FCN model (Gao and Lin, 2019), SegNet mode (Badrinarayanan et al., 2017), PSPNet model (Zhu et al., 2021), DeepLabv3+ model (Yuan et al., 2022), SwinUnet model (Wang et al., 2022a), UTNet model (Gao et al., 2021), DFL-UNet +CBAM model (Zhang et al., 2023), and TransUNet model (Yan et al., 2023a) are selected as the comparison methods. The above network architecture and proposed method are used to compare the effects of leaf and diseases segmentation on the same datasets. In the following experiment, we used five objective evaluation indicators, including mIoU, mPA, mPrecision, mRecall, and accuracy, to compare the performance of segmentation of each method. As can be seen from Table 4, the proposed network architecture has 87.15% IoU and 92.34% Recall on disease segmentation. Table 4 demonstrate that the proposed network, RAAWC-UNet, outperforms some mainstream

TABLE 4 Results of different segmentation networks on the ALDD test sets.

| Model | Categories | IoU/(%) | R/(%) | P/(%) | F1/(%) | κ |
|---|---|---|---|---|---|---|
| FCN | Background | 98.98 | 99.45 | 99.52 | 99.48 | |
| | Leaf | 96.33 | 98.37 | 97.89 | 98.12 | 0.9410 |
| | Disease | 74.97 | 83.36 | 88.17 | 85.69 | |
| SegNet | Background | 99.43 | 99.64 | 99.78 | 99.70 | |
| | Leaf | 97.66 | 99.22 | 98.42 | 98.81 | 0.9238 |
| | Disease | 81.26 | 85.96 | 93.70 | 89.66 | |
| PSPNet | Background | 99.36 | 99.67 | 99.69 | 99.67 | |
| | Leaf | 97.55 | 98.90 | 98.62 | 98.75 | 0.9643 |
| | Disease | 82.00 | 88.34 | 91.95 | 90.10 | |
| ENet | Background | 99.14 | 99.52 | 99.61 | 99.56 | |
| | Leaf | 96.87 | 98.66 | 98.16 | 98.40 | 0.9218 |
| | Disease | 78.22 | 85.43 | 90.27 | 87.78 | |
| Deeplabv3+ | Background | 99.44 | 99.70 | 99.74 | 99.71 | |
| | Leaf | 97.72 | 98.99 | 98.71 | 98.84 | 0.9635 |
| | Disease | 82.25 | 88.73 | 91.85 | 90.26 | |
| Swin-UNet | Background | 97.77 | 98.58 | 99.16 | 98.86 | |
| | Leaf | 94.06 | 97.64 | 96.25 | 96.94 | 0.9182 |
| | Disease | 79.15 | 87.26 | 89.48 | 88.35 | |
| UTNet | Background | 99.21 | 98.42 | 99.73 | 99.07 | |
| | Leaf | 94.59 | 99.09 | 98.15 | 98.61 | 0.9304 |
| | Disease | 82.56 | 87.11 | 93.09 | 90.00 | |

*(Continued)*

TABLE 4 Continued

| Model | Categories | IoU/(%) | R/(%) | P/(%) | F1/(%) | κ |
|---|---|---|---|---|---|---|
| DFL-UNet +CBAM | Background | 99.15 | 99.62 | 99.32 | 99.46 | |
| | Leaf | 98.14 | 99.14 | 99.01 | 99.07 | 0.9647 |
| | Disease | 86.32 | 92.06 | 93.67 | 92.85 | |
| TransUNet | Background | 96.07 | 97.01 | 99.00 | 97.99 | |
| | Leaf | 90.67 | 97.42 | 92.90 | 95.10 | 0.9020 |
| | Disease | 79.19 | 85.82 | 91.11 | 88.38 | |
| RAAWC-UNet | Background | **99.59** | **99.77** | **99.82** | **99.79** | |
| | Leaf | **98.35** | **99.29** | **99.04** | **99.16** | **0.9788** |
| | Disease | **87.15** | **92.10** | **94.20** | **93.13** | |

The bold values are the maximum values of the comparison algorithm in the three categories.

segmentation networks including FCN, SegNet, PSPNet, ENet, Deeplabv3+, Swin-UNet, UTNet, DFL-UNet +CBAM, and TransUNet. It increased the IoU for leaf segmentation by 2.02%, 0.69%, 0.8%, 1.48%, 0.63%, 4.29%, 3.76%, 0.21%, and 7.68%. Additionally, the IoU for disease segmentation increased by 12.18%, 5.89%, 5.15%, 8.93%, 4.9%, 8%, 4.59%, 0.83%, and 7.96%. The precision of leaves and diseases increased by 1.15%, 0.62%, 0.42%, 0.88%, 0.33%, 2.79%, 0.89%, 0.03%, 6.14% and 6.03%, 0.5%, 2.25%, 3.93%, 2.35%, 4.72%, 1.11%, 0.53%, and 3.09% respectively. The recall of leaves and diseases increased by 0.92%, 0.07%, 0.39%, 0.63%, 0.3%, 1.65%, 0.2%, 0.15%, 1.87% and 8.71%, 6.14%, 3.76%, 6.67%, 3.37%, 4.84%, 4.99%, 0.04%, and 6.28%, respectively. The RAAWC-UNet is higher than the above-mentioned other network structures.

Figure 13 shows the impact of different segmentation models on the segmentation of dense leaves and diseases in outdoor environments. Figure 13A is a typical representation of leaf and dense disease segmentation under complex backgrounds. Figure 13B shows leaf and disease segmentation ground truth. As shown in Figure 13C, FCN has the worst segmentation effect, only large diseases can be segmented, and the edges of the leaves cannot be segmented with the effect of the saw tooth of the original leaves. Compared to Figures 13D, E, G, B, SegNet, PSPNet, and Deeplab v3 + are unable to segment small diseases in dense areas. The

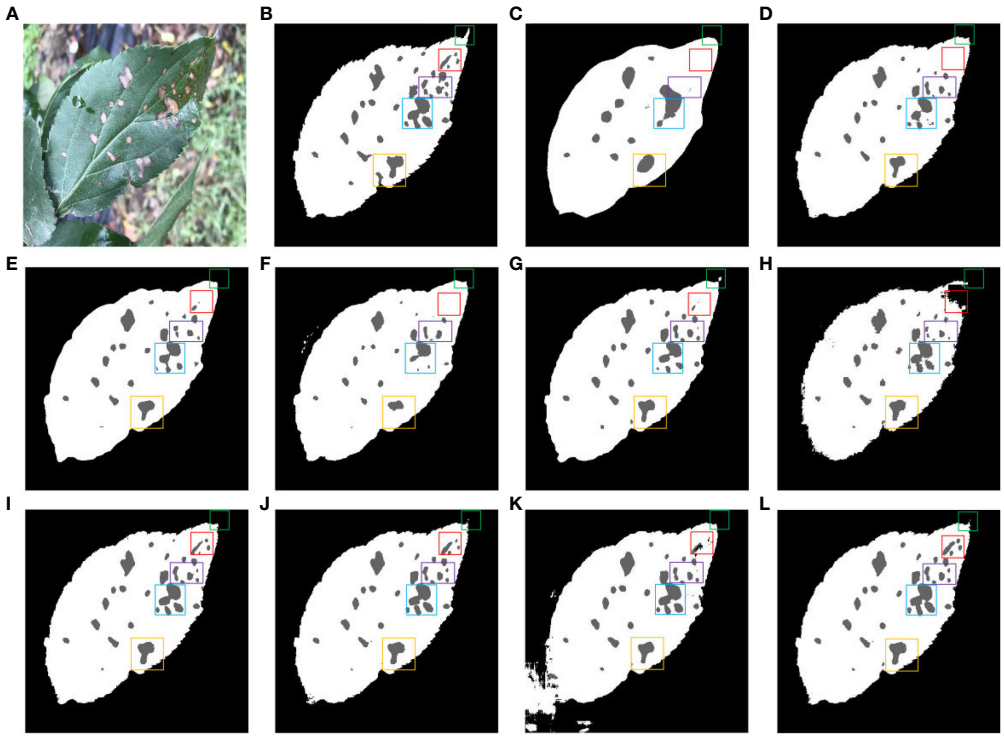



FIGURE 13
Comparison of different methods for leaf and disease segmentation. **(A)** Original images. **(B)** Ground truth. **(C)** FCN. **(D)** SegNet. **(E)** PSPNet. **(F)** ENet. **(G)** Deeplab v3+. **(H)** Swin-UNet. **(I)** UTNet. **(J)** DFL-UNet +CBAM. **(K)** TransUNet. **(L)** Proposed method.

segmentation results of Figures 13F, K indicate that Enet and TransUNet incorrectly segmented overlapping leaves in an outdoor environment. As displayed in Figures 13H–J, Swin-UNet, UTNet, and DFL-UNet +CBAM cannot effectively segment wrinkles, wavy leaves, and diseases in dense areas. From Figure 13L, it can be concluded that the proposed method is better than other models for both leaf and disease segmentation in outdoor environments.

Figure 14 shows a comparison of multiple segmentation networks for leaves and diseases affected by water droplets. Figure 14A is a typical representation of leaf and disease segmentation under the influence of water droplets. Figure 14B shows ground truth for leaf and disease segmentation. Comparison of Figures 14C, D, F, J shows the poor effectiveness of FCN, SegNet, ENet and DFL-UNet +CBAM segmentation of long strips of disease. Figures 14H, I compared to Figure 14B show that SwinUNet and UTNet have jagged tooth-like segmentation of disease pixels. The comparison of disease segmentation between Figures 14E, G, K with Figure 14B reveals that PSPNet, Deeplab v3+, and TransUNet are not precise in segmenting water droplet-affected diseases. From Figure 14L, it is clear that the proposed method performs better than the other models in disease segmentation for long strips and affected by water droplets.

Figure 15 shows the different segmentation models for leaf and disease segmentation affected by light. Figure 15A presents a typical schematic of leaf and disease segmentation under the influence of light. Figure 15B demonstrates the ground truth for leaf and disease segmentation. Figure 15C FCN has the worst results, the leaves do not have the megadentate shape of the ground truth, and Figures 15D, H illustrate the poor segmentation of large disease pixels in SegNet and SwinUNet under the influence of light. Figures 15E, F, I, J exhibit the inability of PSPNet, ENet, UTNet and DFL-UNet +CBAM to accurately segment small diseases in dense areas affected by light. Figures 15G, K show that Deeplab v3+ and TransUNet incorrectly segmented the leaf pixels, and Figure 15L displays that the proposed network segmented the leaf pixels relatively well compared to the other networks under the influence of light.

The objective results of different segmentation networks on the ALDD test sets show that the proposed network has better segmentation ability for leaves and diseases than other models in various complex environments.

## 3.8 Evaluation metrics for different segmentation networks within multiple scenarios

To further validate the performance of the proposed model and other comparative algorithms for segmentation of background, leaf and disease in different scenarios, the experiments are conducted with FCN, SegNet, PSPNet, ENet, Deeplabv3+, Swin-UNet, UTNet, DFL-UNet+CBAM, TransUNet, and the proposed RAAWC-UNet on both indoor and outdoor scenarios with Alternaria blotch, Brown spot, Grey spot and Rust.

Supplementary Table 6 objectively shows the mIoU and mPA of the above 10 segmentation algorithms on the test sets with 7 scenarios. As a whole, the two objective indicators of our

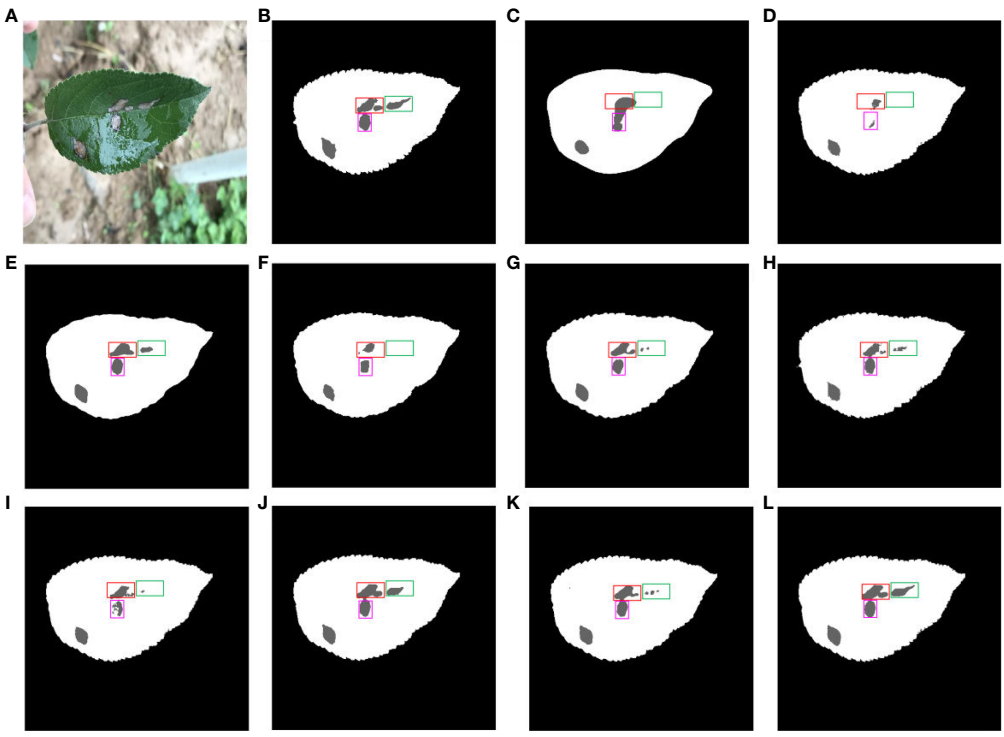



FIGURE 14
Comparison of multiple segmentation networks for leaves and diseases affected by water droplets. **(A)** Original images. **(B)** Ground truth. **(C)** FCN. **(D)** SegNet. **(E)** PSPNet. **(F)** ENet. **(G)** Deeplab v3+. **(H)** Swin-UNet. **(I)** UTNet. **(J)** DFL-UNet +CBAM. **(K)** TransUNet. **(L)** Proposed method.

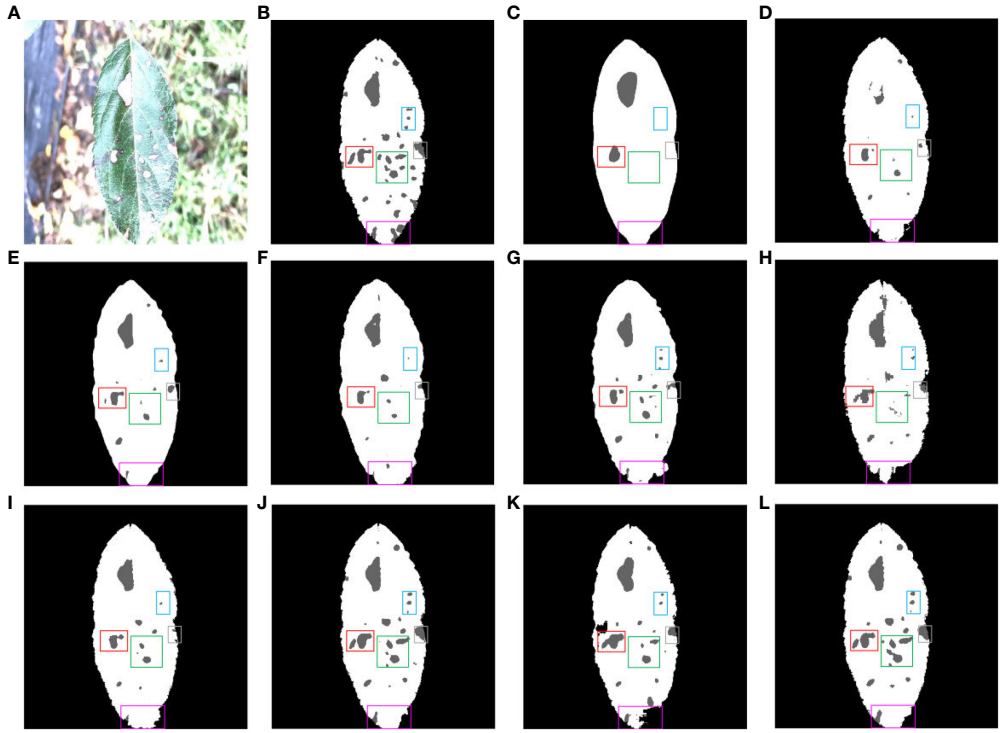



FIGURE 15
Comparison of multiple segmentation networks for leaves and diseases affected by light. **(A)** Original images. **(B)** Ground truth. **(C)** FCN. **(D)** SegNet. **(E)** PSPNet. **(F)** ENet. **(G)** Deeplab v3+. **(H)** Swin-UNet. **(I)** UTNet. **(J)** DFL-UNet +CBAM. **(K)** TransUNet. **(L)** Proposed method.

proposed model surpass those of other models, which demonstrates the superiority of our model. In addition, it is worth noting that due to the dense and widespread occurrence of Grey spot, the segmentation performance of the proposed RAAWC-UNet is comparatively the poorest among the various types of diseases. However, it is still higher than that of the other algorithms mentioned above in Grey spot segmentation.

## 4 Conclusions

In this study, the proposed RAAWC-UNet was developed for segmenting apple and leaf diseases in mixed environments by incorporating ASPP, fusing residual and CBAM modules into UNet with weight compression loss. RAAWC-UNet performs outstandingly in disease segmentation compared to other leaf and disease segmentation networks. The Res_CBAM module effectively captures features at different levels while integrating channel and spatial information. It not only enriches contextual information but also enhances the model's perceptual capabilities, addressing the issue of foreground and background pixel imbalance. The ASPP module adapts by utilizing different dilation rates, flexibly adjusting the convolution kernel's perceptual field to accommodate various leaf and disease pixel sizes. The weight compression loss helps with fast convergence early in the model training. The proposed method is better than most of the segmentation algorithms, and the model presents superior performance especially when dealing with small size and diverse disease segmentation tasks.

Compared with some commonly used networks, the model has lower computational complexity and fewer parameters. While, its computational complexity and parameter quantity are slightly higher in comparison of some lightweight networks. However, the model was better at the task of segmenting leaves and diseases when dealing with factors such as light and water droplets in outdoor environments.

In addition, the following research direction will focus on expanding the scope of data collection and considering various harsh environmental conditions, such as fog, rain and frost, to further study the impact of these environments on the performance of the model. Eventually, we are committed to research on model optimization to meet the needs of resource constrained environments. This effort will further promote the research and development of smart agriculture.

## Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: https://drive.google.com/file/d/1qV3zZCNh8FhrMwQwZds9rRkm9SUQXV5P/view?usp=sharing.

## Author contributions

JW: Conceptualization, Funding acquisition, Validation, Writing – review & editing. JJ: Conceptualization, Methodology,

Software, Validation, Writing – original draft, Writing – review & editing. YZ: Data curation, Supervision, Writing – review & editing. HW: Visualization, Writing – review & editing. SZ: Project administration, Supervision, Writing – review & editing.

## Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This research was funded partly by the Doctoral Foundation of Henan Polytechnic University under Grant B2022-15.

## Acknowledgments

The authors would like to thank the reviewers for their constructive comments and suggestions which strengthened a lot this paper.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2024.1305358/full#supplementary-material

## References

Bacanin, N., Bezdan, T., Al-Turjman, F., and Rashid, T. A. (2022). Artificial flora optimization algorithm with genetically guided operators for feature selection and neural network training. *Int. J. Fuzzy Syst.* 24, 2538–2559. doi: 10.1007/s40815-021-01191-x

Bacanin, N., Stoean, R., Zivkovic, M., Petrovic, A., Rashid, T. A., and Bezdan, T. (2021). Performance of a novel chaotic firefly algorithm with enhanced exploration for tackling global optimization problems: Application for dropout regularization. *Mathematics* 9, 01–33. doi: 10.3390/math9212705

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoderdecoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 2481–2495. doi: 10.1109/TPAMI.34

Barman, U., Choudhury, R. D., Sahu, D., and Barman, G. G. (2020). Comparison of convolution neural networks for smartphone image based real time classification of citrus leaf disease. *Comput. Electron. Agric.* 177, 01–09. doi: 10.1016/j.compag.2020.105661

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 01–14. doi: 10.1109/TPAMI.2017.2699184

Deng, Y., Xi, H., Zhou, G., Chen, A., Wang, Y., Li, L., et al. (2023). An effective image-based tomato leaf disease segmentation method using mc-unet. *Plant Phenomics* 5, 01–17. doi: 10.34133/plantphenomics.0049

Dhaka, V. S., Meena, S. V., Rani, G., Sinwar, D., Ijaz, M. F., and Woźniak, M. (2021). A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. *Sensors* 21, 01–34. doi: 10.3390/s21144749

Fu, L., Li, S., Sun, Y., Mu, Y., Hu, T., and Gong, H. (2022). Lightweight-convolutional neural network for apple leaf disease identification. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.831219

Gao, L., and Lin, X. (2019). Fully automatic segmentation method for medicinal plant leaf images in complex background. *Comput. Electron. Agric.* 164, 01–14. doi: 10.1016/j.compag.2019.104924

Gao, Y., Zhou, M., and Metaxas, D. N. (2021). "Utnet: a hybrid transformer architecture for medical image segmentation," in *Medical Image Computing and Computer Assisted Intervention– MICCAI 2021: 24th International Conference*, Strasbourg, France, September 27–October 1, 2021. 61–71, Proceedings, Part III 24 (New Jersey, USA: Springer).

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. (Las Vegas, NV, USA: IEEE). 770–778. doi: 10.1109/CVPR.2016.90

Jadon, S. (2020). "A survey of loss functions for semantic segmentation," in *2020 IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB)*. (Boston, MA, USA: IEEE). 01–07. doi: 10.1109/CIBCB48159.2020.9277638

Jin, S., Su, Y., Gao, S., Wu, F., Hu, T., Liu, J., et al. (2018). Deep learning: individual maize segmentation from terrestrial lidar data using faster r-cnn and regional growth algorithms. *Front. Plant Sci.* 9. doi: 10.3389/fpls.2018.00866

Kundu, N., Rani, G., Dhaka, V. S., Gupta, K., Nayak, S. C., Verma, S., et al. (2021). Iot and interpretable machine learning based framework for disease prediction in pearl millet. *Sensors* 21, 01–23. doi: 10.3390/s21165386

Lee, S. H., Goëau, H., Bonnet, P., and Joly, A. (2020). New perspectives on plant disease characterization based on deep learning. *Comput. Electron. Agric.* 170, 01–12. doi: 10.1016/j.compag.2020.105220

Li, L., Zhang, S., and Wang, B. (2021). Plant disease detection and classification by deep learning— a review. *IEEE Access* 9, 56683–56698. doi: 10.1109/ACCESS.2021.3069646

Liu, B., Zhang, Y., He, D., and Li, Y. (2017). Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry* 10, 01–11. doi: 10.3390/sym10010011

Liu, J., and Wang, X. (2021). Plant diseases and pests detection based on deep learning: a review. *Plant Methods* 17, 1–18. doi: 10.1186/s13007-021-00722-9

Long, J., Shelhamer, E., and Darrell, T. (2015). "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3431–3440.

Lu, Y., Ji, Z., Yang, L., and Jia, W. (2023b). Mask positioner: An effective segmentation algorithm for green fruit in complex environment. *J. King Saud University-Computer Inf. Sci.* 35, 01–10. doi: 10.1016/j.jksuci.2023.101598

Lu, B., Lu, J., Xu, X., and Jin, Y. (2023a). Mixseg: a lightweight and accurate mix structure network for semantic segmentation of apple leaf disease in complex environments. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1233241

Malakar, S., Ghosh, M., Bhowmik, S., Sarkar, R., and Nasipuri, M. (2020). A ga based hierarchical feature selection approach for handwritten word recognition. *Neural Computing Appl.* 32, 2533–2552. doi: 10.1007/s00521-018-3937-8

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention– MICCAI 2015: 18th International Conference*, Munich, Germany, October 5-9, 2015. 234–241, Proceedings, Part III 18 (Cham: Springer International Publishing).

Shi, T., Liu, Y., Zheng, X., Hu, K., Huang, H., Liu, H., et al. (2023). Recent advances in plant disease severity assessment using convolutional neural networks. *Sci. Rep.* 13, 01–15. doi: 10.1038/s41598-023-29230-7

Su, J., Zhang, M., and Yu, W. (2022). "An identification method of apple leaf disease based on transfer learning," in *2022 7th International Conference on Cloud Computing*

*and Big Data Analytics (ICCCBDA)*. (Chengdu, China: IEEE). 478–482. doi: 10.1109/ICCCBDA55098.2022.9778876

Sun, H., Xu, H., Liu, B., He, D., He, J., Zhang, H., et al. (2021). Mean-ssd: A novel realtime detector for apple leaf diseases using improved light-weight convolutional neural networks. *Comput. Electron. Agric.* 189, 01–13. doi: 10.1016/j.compag.2021.106379

Tang, Y., Chen, C., Leite, A. C., and Xiong, Y. (2023). Precision control technology and application in agricultural pest and disease control. *Front. Plant Sci.* 14, 1163839. doi: 10.3389/fpls.2023.1163839

Tang, Z., Yang, J., Li, Z., and Qi, F. (2020). Grape disease image classification based on lightweight convolution neural networks and channelwise attention. *Comput. Electron. Agric.* 178, 01–09. doi: 10.1016/j.compag.2020.105735

van Bruggen, A. H., Gamliel, A., and Finckh, M. R. (2016). Plant disease management in organic farming systems. *Pest Manage. Sci.* 72, 30–44. doi: 10.1002/ps.4145

Wang, C., Du, P., Wu, H., Li, J., Zhao, C., and Zhu, H. (2021). A cucumber leaf disease severity classification method based on the fusion of deeplabv3+ and u-net. *Comput. Electron. Agric.* 189, 01–13. doi: 10.1016/j.compag.2021.106373

Wang, C., Li, C., Han, Q., Wu, F., and Zou, X. (2023). A performance analysis of a litchi picking robot system for actively removing obstructions, using an artificial intelligence algorithm. *Agronomy* 13, 2795. doi: 10.3390/agronomy13112795

Wang, F., Rao, Y., Luo, Q., Jin, X., Jiang, Z., Zhang, W., et al. (2022a). Practical cucumber leaf disease recognition using improved swin transformer and small sample size. *Comput. Electron. Agric.* 199, 01–10. doi: 10.1016/j.compag.2022.107163

Wang, Y., Wang, Y., and Zhao, J. (2022b). Mga-yolo: A lightweight one-stage network for apple leaf disease detection. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.927424

Wani, J. A., Sharma, S., Muzamil, M., Ahmed, S., Sharma, S., and Singh, S. (2022). Machine learning and deep learning based computational techniques in automatic agricultural diseases detection: Methodologies, applications, and challenges. *Arch. Comput. Methods Eng.* 29, 641–677. doi: 10.1007/s11831-021-09588-5

Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S. (2018). "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, (Munich, Germany: Springer International Publishing). Vol. 11211. 3–19.

Woźniak, M., Wieczorek, M., and Siłka, J. (2022). "Deep neural network with transfer learning in remote object detection from drone," in *Proceedings of the 5th international ACM mobicom workshop on drone assisted wireless communications for 5G and beyond*. (Sydney NSW Australia: ACM). 121–126. doi: 10.1145/3555661.3560875

Wu, F., Yang, Z., Mo, X., Wu, Z., Tang, W., Duan, J., et al. (2023). Detection and counting of banana bunches by integrating deep learning and classic image-processing algorithms. *Comput. Electron. Agric.* 209, 107827. doi: 10.1016/j.compag.2023.107827

Xizhen, S., Wei, Z., Yiling, G., and Shengyang, Y. (2021). "Edge detection algorithm of plant leaf image based on improved canny," in *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*. (Xi'an, China: IEEE). 342–345. doi: 10.1109/ICSP51882.2021.9408929

Xu, W., and Wang, R. (2023). Alad-yolo: an lightweight and accurate detector for apple leaf diseases. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1204569

Yan, C., Li, Z., Zhang, Z., Sun, Y., Wang, Y., and Xin, Q. (2023a). High-resolution mapping of paddy rice fields from unmanned airborne vehicle images using enhanced-transunet. *Comput. Electron. Agric.* 210, 01–12. doi: 10.1016/j.compag.2023.107867

Yan, J., Yan, T., Ye, W., Lv, X., Gao, P., and Xu, W. (2023b). Cotton leaf segmentation with composite backbone architecture combining convolution and attention. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1111175

Yang, L., Yu, X., Zhang, S., Long, H., Zhang, H., Xu, S., et al. (2023). Googlenet based on residual network and attention mechanism identification of rice leaf diseases. *Comput. Electron. Agric.* 204, 01–11. doi: 10.1016/j.compag.2022.107543

Yuan, H., Zhu, J., Wang, Q., Cheng, M., and Cai, Z. (2022). An improved deeplab v3+ deep learning network applied to the segmentation of grape leaf black rot spots. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.795410

Zeng, W., Li, H., Hu, G., and Liang, D. (2022). Lightweight dense-scale network (ldsnet) for corn leaf disease identification. *Comput. Electron. Agric.* 197, 01–15. doi: 10.1016/j.compag.2022.106943

Zhang, X., Li, D., Liu, X., Sun, T., Lin, X., and Ren, Z. (2023). Research of segmentation recognition of small disease spots on apple leaves based on hybrid loss function and cbam. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1175027

Zhu, X., Cheng, Z., Wang, S., Chen, X., and Lu, G. (2021). Coronary angiography image segmentation based on pspnet. *Comput. Methods Programs Biomedicine* 200, 01–14. doi: 10.1016/j.cmpb.2020.105897

Zhu, S., Ma, W., Lu, J., Ren, B., Wang, C., and Wang, J. (2023a). A novel approach for apple leaf disease image segmentation in complex scenes based on two-stage deeplabv3+ with adaptive loss. *Comput. Electron. Agric.* 204, 01–19. doi: 10.1016/j.compag.2022.107539

Zhu, S., Ma, W., Wang, J., Yang, M., Wang, Y., and Wang, C. (2023b). Eadd-yolo: An efficient and accurate disease detector for apple leaf using improved lightweight yolov5. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1120724

Zivkovic, M., Bacanin, N., Venkatachalam, K., Nayyar, A., Djordjevic, A., Strumberger, I., et al. (2021). Covid-19 cases prediction by using hybrid machine learning and beetle antennae search approach. *Sustain. cities Soc.* 66, 01–22. doi: 10.1016/j.scs.2020.102669

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 27 March 2024
DOI 10.3389/fpls.2024.1340584

Check for updates

OPEN ACCESS

EDITED BY
Md. Motaher Hossain,
Bangabandhu Sheikh Mujibur Rahman Agricultural University, Bangladesh

REVIEWED BY
Shaikhul Islam,
Bangladesh Wheat and Maize Research Institute (BWMRI), Bangladesh
Hasan M. Abdullah,
Bangabandhu Sheikh Mujibur Rahman Agricultural University, Bangladesh

*CORRESPONDENCE
Qinjian Zhang
zhangqinjian@bistu.edu.cn

†These authors have contributed equally to this work

RECEIVED 18 November 2023
ACCEPTED 11 March 2024
PUBLISHED 27 March 2024

CITATION
Wu Y, Xi Z, Liu F, Hu W, Feng H and Zhang Q (2024) A deep semantic network-based image segmentation of soybean rust pathogens. *Front. Plant Sci.* 15:1340584. doi: 10.3389/fpls.2024.1340584

COPYRIGHT
© 2024 Wu, Xi, Liu, Hu, Feng and Zhang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# A deep semantic network-based image segmentation of soybean rust pathogens

Yalin Wu[1†], Zhuobin Xi[2†], Fen Liu[1†], Weiming Hu[1], Hongjuan Feng[3] and Qinjian Zhang[2*]

[1]Lushan Botanical Garden, Jiangxi Province and Chinese Academy of Sciences, Jiujiang, China, [2]Mechanical Electrical Engineering School, Beijing Information Science & Technology University, Beijing, China, [3]Zhongzhen Kejian (ShenZhen) Holdings Co., Ltd, Shenzhen, China

**Introduction:** Asian soybean rust is a highly aggressive leaf-based disease triggered by the obligate biotrophic fungus *Phakopsora pachyrhizi* which can cause up to 80% yield loss in soybean. The precise image segmentation of fungus can characterize fungal phenotype transitions during growth and help to discover new medicines and agricultural biocides using large-scale phenotypic screens.

**Methods:** The improved Mask R-CNN method is proposed to accomplish the segmentation of densely distributed, overlapping and intersecting microimages. First, Res2net is utilized to layer the residual connections in a single residual block to replace the backbone of the original Mask R-CNN, which is then combined with FPG to enhance the feature extraction capability of the network model. Secondly, the loss function is optimized and the CIoU loss function is adopted as the loss function for boundary box regression prediction, which accelerates the convergence speed of the model and meets the accurate classification of high-density spore images.

**Results:** The experimental results show that the mAP for detection and segmentation, accuracy of the improved algorithm is improved by 6.4%, 12.3% and 2.2% respectively over the original Mask R-CNN algorithm.

**Discussion:** This method is more suitable for the segmentation of fungi images and provide an effective tool for large-scale phenotypic screens of plant fungal pathogens.

KEYWORDS

**Asian soybean rust, *Phakopsora pachyrhizi*, deep learning, instance segmentation, mask R-CNN**

# 1 Introduction

Soybean (*Glycine max*) is one of the most economically efficient crops since it is an important source of food, protein, and vegetable oil. Asian Soybean Rust (ASR) is a globally aggressive foliar disease of soybean plants that can cause up to 80% losses and have a significant impact on production costs in various geographical areas invaded by the pathogen (Lorrain et al., 2019). Fungus *Phakopsora pachyrhizi* is the causal agent of ASR. Infection begins with the deposition of uredospores on soybean leaves, where the rust fungus invades the epidermal cells of the host through the appressorium formed during spore germination and extracts nutrients from the host body (Goellner et al., 2010; Loehrer and Schaffrath, 2011). This fungus can defoliate soybean fields and accelerate maturation with a reduction of seed size and weight and may lead to complete crop failure within a few days (Hartman et al., 2015). Currently, timely fungicide application is the only means of controlling ASR (Saito et al., 2021).

It is important to analyze the characterization of fungi germinating *in vitro* for ASR disease control and research. Researchers have found that automated microscopy-based phenotyping is typically used under genetically or environmentally sensitive conditions to probe the relationship between cell structure and function by unbiased quantification of phenotypic changes in response to perturbations of interest (Liberali et al., 2015; Usaj et al., 2016). Some studies have applied fungal images to the field of drug discovery and development (Calderone et al., 2014; Carolus et al., 2020). By analyzing the morphology and characteristics of fungi, researchers are able to better understand the structure and function of fungi. Statistical information on fungal spores can reveal the degree of resistance and activity of spores to discover new drug candidates and therapeutic options. Large-scale phenotypic screening of multiple compounds acting on fungal spores can identify suitable fungicides and drugs for ASR. However, these drugs usually have to be screened from hundreds of compounds by expert labor, which requires huge processing time. These image-based methods have found their greatest application in the pharmaceutical industry, where they have been used to primary screening stages of drug discovery, drug target validation, early evaluation of toxicity properties and complex multivariate drug profiling (Zanella et al., 2010; Reisen et al., 2015). For example, haploid yeast was treated with drugs that perturb cell wall and the dose-dependent changes in morphology were analyzed to identify drugs that interfere with cell wall synthesis (Okada et al., 2014).

As show in Figure 1, Segmentation is one of the significant steps of phenotypic screening (Cabre et al., 2021). Accurate image segmentation of fungal spores can characterize phenotypic changes during fungal growth, and accurate segmentation significantly determines the efficiency and effectiveness of drug screening, contributing to the discovery of drugs and agricultural fungicides using large-scale phenotypic screening, as well as to the development of strategies for the control of ASR using biotechnological approaches. Manual segmentation of images is cost-ineffective and time-consuming for expert annotation, and thus is impractical for large data segmentation. More importantly, due to the variability of individuals, manual segmentation can introduce large segmentation errors and biases, so there is a need to find an accurate and efficient automatic segmentation method. Due to the different degree of response of the fungi to different drugs, it appears that the spore morphology of fungi in the process of reproduction appears to have a large morphological variability. Moreover, the interaction between fungi, many fungi overlap, distort and adhere to each other, which can make accurate segmentation difficult. Finally, the collected microscope images have low contrast, and the fungal edges are very blurred and difficult to identify accurately, while some of the images have impurities.

Some microscopy applications use machine learning algorithms, such as those for range thresholding, simple filters, and edge detection based on intensity changes are now widely used (Melo et al., 2019). Traditional image segmentation methods

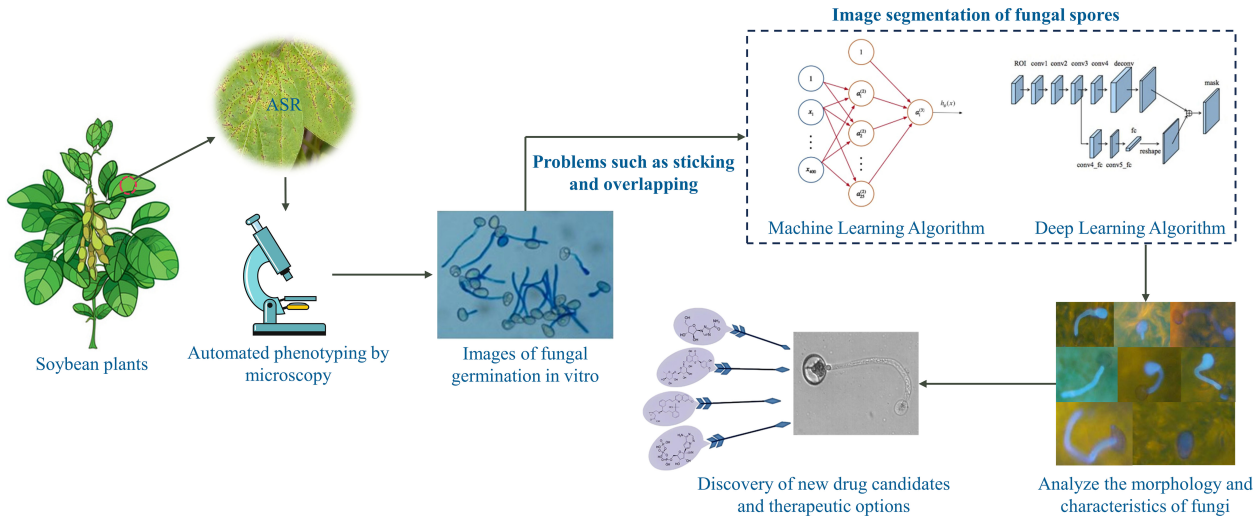



FIGURE 1
Large-scale phenotypic screening of multiple drugs based on automated fungal segmentation.

include Otsu's thresholding (Otsu, 1979), watershed algorithm (Beucher, 1992) and clustering (Coleman and Andrews, 1979). For instance, a Gaussian Separate Degree is used for Otsu method, called as G-Otsu, is proposed to segment anthrax spore images (Zhao et al., 2020). Korsnes et al. (2016) used methods such as mean gradient and morphological processing to detect spore boundaries for spore segmentation, followed by egg shape-fitting techniques to fit spore perimeter. Using K-means method to segment spores of *Puccinia striiformis* f. sp. *tritici* (*Pst*) by clustering pixel values, and isolate touching spores based on the shape and area factors (Lei et al., 2018). However, these classical machine vision methods are sensitive to noise and lack robustness, and usually cannot realize the segmentation of complex shapes.

Deep learning can learn how to extract the features from a large number of samples. Zhao et al. took anthrax spores as the research objects and applied CFL (Constrained Focal Loss) Loss function to DeeplabV3+. Experimentally, this proposed CFLNet* can achieve better performance than original DeepLabv3+ (Zhao et al., 2019). Yang et al. (2020) proposed a Nuclear Segmentation Tool (NuSeT), which assimilates the advantages of semantic segmentation (U-Net) and instance segmentation (Mask R-CNN) and can work with both fluorescent and histopathology image samples. Xie et al. (2021) used Mask Scoring R-CNN network to detect mango disease spores to control and prevent mango disease. Li et al. (2023) proposed an MG-YOLO detection algorithm that introduces Multi-head self-attention in the YOLO backbone and optimizes the network neck and pyramid structure for fast and accurate gray mold spores detection, with a detection accuracy of 0.983 for the improved model and a time spent of 0.009 seconds per image. Zhang et al. (2023) introduced the attention mechanism module (ECA-Net) and adaptive feature fusion mechanism (ASFF) into the feature pyramid structure of YOLO to detect *Fusarium germinate* spores of small targets, and the average recognition accuracy of this model was 98.57%.

Image segmentation includes semantic and instance segmentation. The task of semantic segmentation is to classify each pixel in the image without separating the objects (Long et al., 2015), but this does not apply to our fungal segmentation task because there are a large number of *Phakopsora pachyrhizi* adhering or overlapping in the image, which can cause the touching fungus to not be segmented from each other and cause under-segmentation problems. Instance segmentation is a combination of the object detection and the semantic segmentation, where the object is detected in the image and then each pixel is labeled. Identifying *Phakopsora pachyrhizi* in an image is best viewed as an instance segmentation task (Hariharan et al., 2014). In this paper, we propose an improved Mask R-CNN spore segmentation method to solve these problems and improve the accuracy of spore segmentation. The main objectives include:

Optimization of backbone using Res2net block. by hierarchizing the residual connections in a single residual block, it is possible to achieve a multi-scale characterization of the fine-grained layers and, at the same time, increase the size of the sensory field at each level of the network. The use of Feature Pyramid Grids (FPGs) highlights the importance of deep pyramid representations by improving single-path feature pyramid networks by significantly improving their performance at a similar computational cost. Use CIOU as a bounding box regression loss function to reduce the error.

A new method for fungal spore segmentation is proposed; extensive experiments show that this method achieves better segmentation performance under high density and overlapping conditions.

## 2 Materials and methods

In this section, we first summarize the problems and challenges faced in segmenting target images. Then, the segmentation model is designed and optimized, including the optimization of backbone and the optimization of mask branch.

### 2.1 Data source

In this paper, a spore image dataset was established to characterize the phenotypic transformation of fungi during *in vitro* growth in the presence of different fungicides. We used PerkinElmer's Opera QEHS high content rotary confocal system to observe fungal spores at different stages of growth and extract high quality images. The rotary disk confocal microscope can scan multi-channel fluorescence signals in a short time, and reduces the influence of detection environment on cells through extremely sensitive confocal imaging and synchronous acquisition.

Fresh leaves with rust organelles that had broken through the leaf epidermis and yellow rust spores were collected from the experimental field and brought back to the laboratory. Firstly, the surface of the diseased leaves was rinsed with running water, and then the leaf surface around the rust organelles was wiped with 75% ethanol, and then the diseased leaves were put into a petri dish with a wet filter paper at the bottom to keep humidity, and then fresh spores scattered around the rust organelles were collected after 1 d. Fresh spores collected were put into a 2 mL EP tube with appropriate amount of sterile water containing 0.3% Tween 80, and shaken well to make a spore suspension. The collected fresh rust spores were put into 2 mL EP tubes, and the spore suspension was made by adding appropriate amount of sterile water containing 0.3% Tween 80 and shaking well. 100μl of the sample was inoculated into the wells of a 96-well plate, and the spores from the 96-well plate were mixed in batches with six solvents: Carbendazim (1 ppm), DMSO (0.1%), PIK-75 (3.3 ppm), Solatenol (0.041 ppm), Solatenol (10 ppm) and TOU-951 (1.1 ppm). After 90 minutes, spores were stained using Calcofluor White solution with KOH and imaging of spores was recorded every 15 minutes one the Opera QEHS at a magnification of 10x to track spore growth status, with a total of 9 time-state data recorded. Hundreds of images of fungal spores at different stages of growth were collected under each time period for each chemical treatment.

Each image has a size of 685 × 503 pixels and a pixel range of 8bit, saved in TIF format. We randomly selected 300 images as our dataset which fully contains the various morphology of fungal spores under the action of fungicides, as is show in Figure 2. (Due to the different efficacy of drugs acting on fungi, fungal spores vary greatly in phenotype, such as size, length, and number.) The

10.3389/fpls.2024.1340584

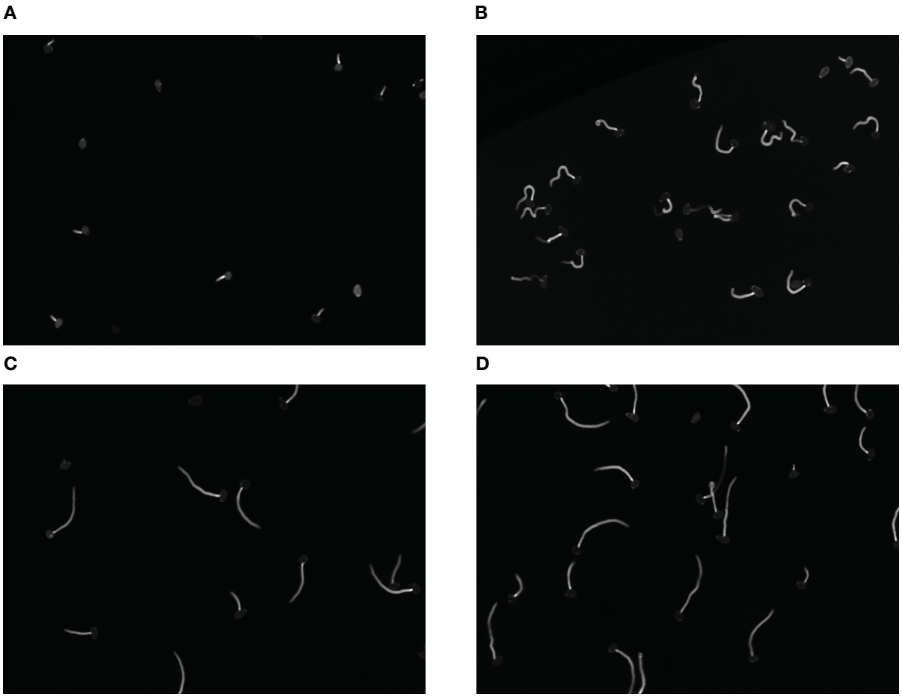



FIGURE 2
**(A–D)** Respectively represent spores in different growth states.

diversity of fungi images collected improves the generalization ability and robustness of algorithm.

Each image in the dataset was manually annotated for network training by the open-source software Labelme (Russell et al., 2008) which labels the pixels of each class. Specifically, to instance segmentation, each single fungus segmentation served as an instance. As is shown in Figure 3.

The *Phakopsora pachyrhizi* dataset was divided into training and test sets in a ratio of 80:20, where 240 images were used as the training set (containing 5526 spores) and 60 images were used as the training set (containing 1621 spores). Thereafter, the labeled image instance information is stored in test set and training set json files respectively. The dataset is shown in Table 1.

## 2.2 Instance segmentation methods for spores

Microscopic image segmentation is an intricate task, with the target spores in *Phakopsora pachyrhizi* images encompass fungal spores with variable shapes and the same image mixed with multiple different growth states. An integral fungus consists of 2 parts, the germ tube and the spore. Due to the defects of light and the different sensitivity of different parts of the spore to light, the image of the stained spore has a low contrast, and it is difficult for the naked eye to detect its edges, which improves the difficulty of accurate segmentation. Additionally, growth phenotypes, such as fungal germination count, germ tube length, and growth direction,

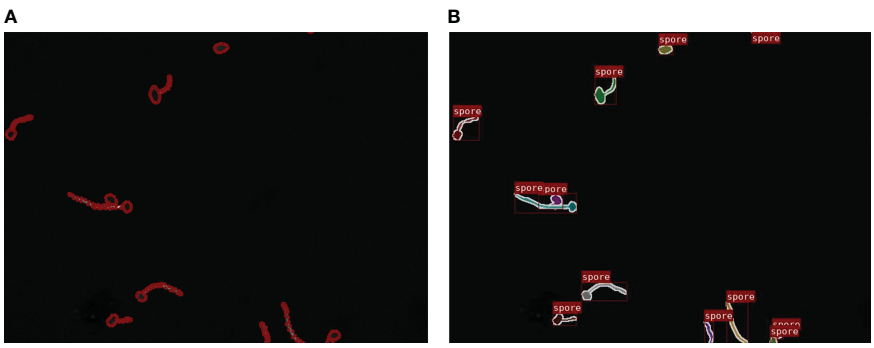



FIGURE 3
Marking process: **(A)** Data annotation. **(B)** Visualization of the mask image.

exhibited significant variations under distinct medicinal treatments. Furthermore, fungi within the same image often appear densely populated, particularly during the later growth stages when exposed to certain solutions. Germ tubes tend to spread across a wide area, resulting in increased instances of crossing, overlapping, and clumping. These challenges have posed difficulties in achieving precise segmentation of *Phakopsora pachyrhizi*.

Image segmentation entails the meticulous classification of pixels into specific categories within an image. In contrast to semantic segmentation, instance segmentation not only segregates diverse objects within an image but also goes a step further by assigning a distinct classification to each individual pixel within the identified instances. This approach facilitates precise localization and differentiation of individual entities within the input image, leading to heightened accuracy in the process.

In the context of microscopic image spore segmentation, the application of instance segmentation techniques offers multifaceted benefits. Beyond effectively addressing challenges arising from spore intersections, overlaps, and adhesions, instance segmentation ensures the isolation of each spore as an autonomous entity, thereby averting any potential information ambiguities. Moreover, this approach excels at precisely determining the spatial coordinates of each spore, encompassing crucial details such as spore boundaries and internal structures. Such precision assumes paramount significance in comprehending the spatial arrangement and density distribution of spores within the image. Through instance segmentation, a nuanced understanding of the spore layout and distribution emerges, thereby enabling more informed analyses and interpretations. Instance segmentation offers the capability to establish distinct units of analysis for each spore, enabling segmentation at an individual level. This approach facilitates the quantification of various attributes like size, shape, color, and additional characteristics inherent to each spore. Consequently, this yields a more comprehensive dataset, enabling a deeper exploration into the intricate nuances of spore variation, interactions, and other pertinent traits. The detailed data acquired through instance segmentation serves as a foundation for conducting exhaustive investigations into the diverse aspects of spore behavior, facilitating enhanced insights and understanding.

Mask R-CNN is a classical top-down two-stage instance segmentation network, which can be considered as the extension of the Faster R-CNN architecture. This network builds upon the original network structure, incorporating additional branches to facilitate the prediction of segmentation masks for each ROI, all the while concurrently performing classification and bounding box regression. The process begins with the input image being fed through the backbone network, resulting in a feature map. This map is then utilized in the Region Proposal Network (RPN) to generate the corresponding anchor boxes. Subsequently, the feature maps linked to each anchor box are homogenized to a consistent size through RoIAlign, ensuring compatibility for further processing. Eventually, this standardized feature map is introduced to a fully connected layer, followed by anchor position refinement executed through regression layers, and class probabilities estimation performed through classification layers. The combination of these processes yields accurate instance segmentation results, with the model generating both precise boundaries and segmentation masks for identified objects.

Researchers are constantly exploring new ways to combine Mask R-CNN with other techniques to improve the performance of segmentation tasks. For example, Seki and Toda (2022) utilized Mask R-CNN to segment lettuce seeds and extract their morphological parameters. Jia et al. (2022) Optimizing Mask R-CNN using the lightweight backbone network MobileNetv3 speeds up the model and meets the storage resource requirements of mobile robots. Chen et al. (2023) incorporated the attention mechanism into the backbone network of Mask R-CNN, which can better detect and segment the tapping area of natural rubber trees under different shooting conditions. Although Mask R-CNN has demonstrated excellent performance in the field of instance segmentation, it still faces great challenges when dealing with data such as spore images, which are characterized by a high degree of overlap and adhesion. In view of this, it is particularly crucial to develop an efficient segmentation strategy for spore image characteristics.

In order to enhancing the precision of spore segmentation, this research introduces an enhanced methodology for spore segmentation using Mask R-CNN. This approach integrates a variant of the Mask R-CNN architecture by incorporating Feature Pyramid Grids (FPG). The single-path feature pyramid network is improved by using FPG, where the feature scale space is represented as a regular lattice of parallel pathways and the pathways are fused together through multidirectional transversal links, which significantly improves the performance of the network with similar computational cost. The backbone network was optimized using an improved backbone, and using Res2Net module by layering residual connections in a single residual block allows for multiscale features at fine-grained layers while increasing the size of the perceptual field at each level of the network.

### 2.2.1 Feature pyramid grids

Feature Pyramid Grid (FPG) is an FPN-derived deep multi pathway network as shown in Figure 4. The feature scale space of this deep multi pathway feature pyramid network is a fusion of multidirectional lateral connections between parallel paths for information exchange at all levels to build a robust network with high discriminatory power and fine resolution across spatial dimensions. The single pyramid path back-propagates semantic information into the network by successively up-sampling the feature maps. FPG is a parallel extension of the single pyramid, which enriches the multidirectional (semantic) information in the scale space through the lateral connections between feature maps, allowing complex hierarchical features to be learned across scales. Lateral connection has 4 categories. Among them, AcrossSame is the fusion of features of the same level with those in the neighboring paths after using 1*1 convolution. AcossUp uses a convolution of 3*3 stride of 2 to fuse the low-level features of the previous pathway with the high-level features of the next neighboring pathway. AcrossDown fuses the high-level features of the previous pathway to the low-level features of the next neighboring pathway by nearest interpolation convolution with a scaling factor of 2. AcrossSkip uses 1*1 convolutional skip connections between same-level features.

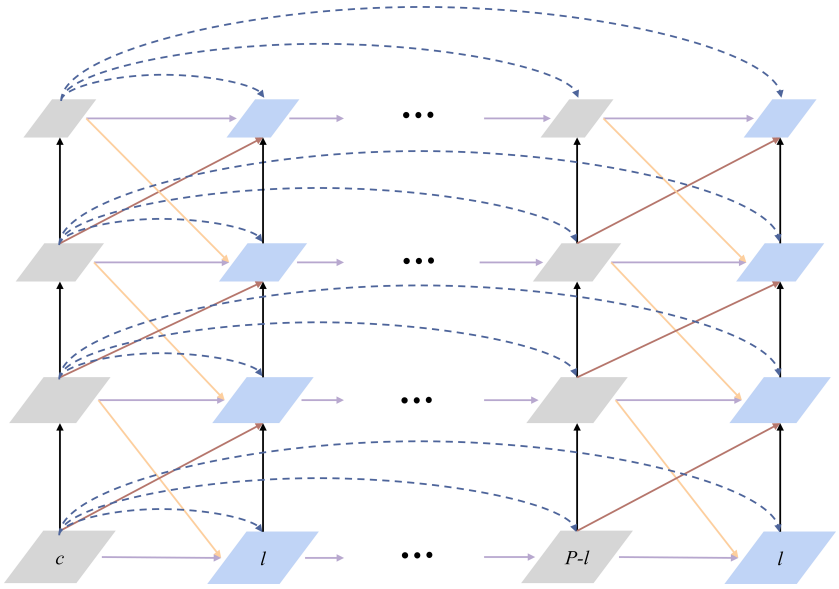



FIGURE 4
A Feature Pyramid Grid (FPG).

Each convolutional block consists of a ReLU, a convolutional layer and a BatchNorm layer, and the fusion function uses element-wise Sum.

### 2.2.2 Composite backbone

The backbone network in instance segmentation is used to extract features from the input image and determines the feature representation capability of the model. For densely distributed fungi in the image, the large appearance of spores especially at the late stage of chemical treatments processing elevates the segmentation difficulty. In order to obtain better segmentation performance under high density and overlapping conditions, we optimize the backbone network based on FPN using Res2Net (Gao et al., 2019) fusion FPG as a composite backbone. Compared with ResNet, Res2Net adds small residual blocks to extract features with different receptive fields and multiple scales, so that the network can learn multiple features with different scales, in order to promote the communication of multi-scale features.

The Res2Net structure is shown in Figure 5. In the Res2net module, the input features are categorized into s subset, denoted as xi, i∈{1,2,…,s}; the number of channels of the feature map in each group is 1/s of the number of channels of the input feature map. Then, each set of feature maps undergoes a 3×3 convolution (denoted as $K_i$), except for x1. Starting from x3, the feature map xi of the ith group is first summed with the $K_{i-1}$ output of the previous group, and the result of the sum is subjected to the $K_i$ operation. The whole process is represented in Formula (1).

$$y_i = \begin{cases} x_i & i = 1 \\ \boldsymbol{K_i}(x_i) & i = 2 \\ \boldsymbol{K_i}(x_i + y_{i-1}) & 2 < i \le s \end{cases} \quad (1)$$

where $y_i$ is the output of the module that is fed into the next convolutional layer. s is *scal*, which serves as the number of parameters controlling the dimensionality of the dimensions, and the larger s is, the better the multidimensional characterization. In

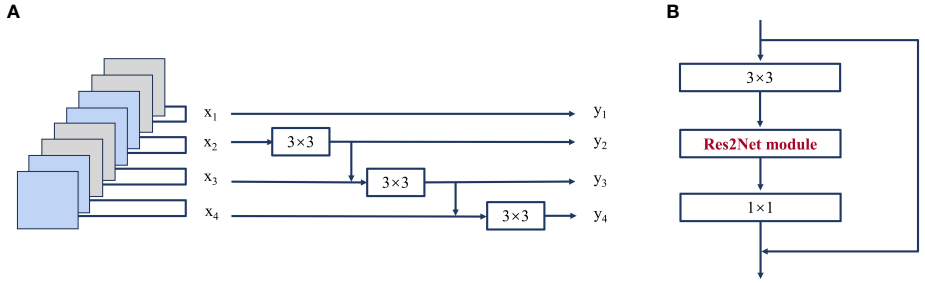



FIGURE 5
**(A)** Res2Net module (scale=4); **(B)** Res2Net block.

this study, the s value size of 26 is used as the Res2Net block, which is used to modify the ResNet structure in the FPN.

Meanwhile, drawing on the idea of FPG, the improved backbone is combined with FPG to represent the feature scale space as a regular lattice of parallel pathways, and the pathways are fused together by multidirectional transverse links, improving the single-path feature pyramid network, which significantly improves the performance of the network with similar computational cost.

The design of Backbone is shown in Figure 6. The bottom-up C2, C3, C4 and C5 are Res2Net module layers, and the stride of each layer is 2. The number of channels of the structural layers output from C2 to C4 are adjusted using 1*1 convolution, which produces P21, P31, P41 and P51, respectively, with P61 being derived from P51. To enhance computational efficiency, some simplifications were made in the lateral connections of FPG. Specifically, the AcrossSkip connection was removed, and a subset of the AcrossSkip, SameUp, and AcrossDown connections were omitted. We retained half of the triangular structure in the lateral connections, and in our experiments, we opted for P=9 paths to enrich the network's capabilities.

### 2.2.3 Loss function

In the whole improved network structure, the corresponding loss function consists of five parts, which are: the RPN's classification result prediction $L_{R\text{-}cls}$, its bounding box regression prediction $L_{R\text{-}box}$, alongside the final classification result prediction $L_{cls}$, final bounding box regression prediction $L_{box}$, and the final mask image prediction $L_{mask}$. Loss function is calculated by Formula (2).

$$L = L_{R-cls} + L_{R-box} + L_{cls} + L_{box} + L_{mask} \tag{2}$$

Classification loss $L_{cls}$ computes the loss of class probability using Cross Entropy.

$L_{mask}$ uses the Binary crossentropy loss function, calculated by the Formula (3).

$$L_{mask} = -\sum_y y \log(1-\hat{y}) + (1-y)\log(1-\hat{y}) \tag{3}$$

where $y$ denotes the binarized ground truth, $\hat{y}$ denotes the predicted segmentation result after binarization.

Edge information is very important for instance segmentation, and they can characterize the instance well. Mask R-CNN begins by utilizing the $\text{smooth}_{L1}$ function for calculating edge loss in target detection. Within this approach, losses for the four coordinate points are computed separately and aggregated to derive the ultimate edge loss. Despite assuming independence among the four points, there exists a certain degree of correlation among them in reality. The process of assessing box detection involves employing Intersection over Union (IoU), which differs from the regression coordinate box derived from the four points. Multiple detection boxes might yield identical $\text{smooth}_{L1}$ Loss values despite differing IoU values. To address this disparity, IoU Loss was introduced as a solution.

However, researchers and scholars soon found that IoU Loss has a drawback: when the prediction box does not intersect the target box, the loss function is not derivable. This problem makes the boundary information ignored in the prediction, and inaccurate edge detection occurs in the experiment, which affects the accuracy of segmentation. In order to meet the accurate segmentation of high-density spore images and improve the sensitivity of boundary segmentation, this paper optimizes the loss function and adopts the $L_{CIoU}$ loss function as the loss function of the bounding box regression prediction, which accelerates the convergence speed of the model and makes the results of boundary segmentation more accurate.

The $L_{CIoU}$ calculates the discrepancy between the predicted bounding box and the ground truth. Its definition is outlined in Formula (4).

$$\begin{cases} L_{CIoU} = 1 - IoU + \frac{\rho^2(b,b^{gt})}{c^2} + \frac{v^2}{(1-IoU)+v} \\ IoU = \frac{|b \cap b^{gt}|}{|b \cup b^{gt}|} \end{cases} \tag{4}$$

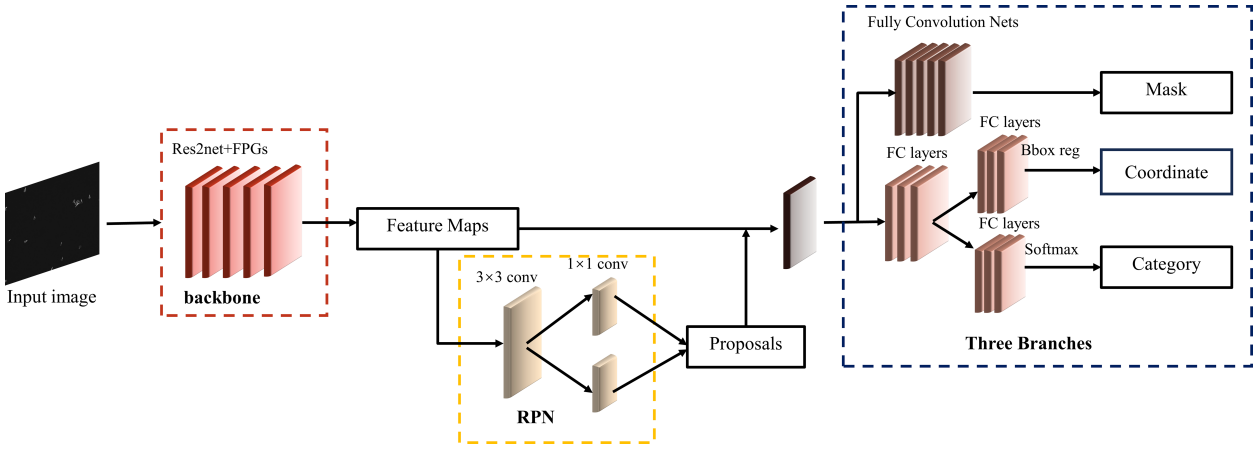



FIGURE 6
The overall framework of the improved Mask R-CNN.

Here, v signifies the alignment of the two frame aspect ratios, while b and $b^{gt}$ denote the center coordinates of the prediction and actual boxes respectively. ρ represents the Euclidean distance between their center points, indicative of the diagonal span of the smallest enclosed area containing both boxes. IoU stands for the Intersection over Union, representing the ratio between the shared area and the combined area of the predicted and actual bounding boxes.

In summary, the loss function used in this paper takes into account the error factor between the predicted value and the true value, which improves the convergence rate of the model, and the optimized network is more accurate in terms of error, and more flexible and feasible.

# 3 Results and discussion

## 3.1 Evaluation metrics

In this section, we present the key metrics used to measure the performance of spore instance segmentation. By leveraging these metrics, we can objectively analyze and showcase the strengths of our method, while also enabling a comprehensive comparison with existing instance segmentation techniques.

Average precision (AP) and average recall (AR) are the main evaluation metrics currently used in the field of object detection and instance segmentation. These metrics depend on two different segmentation masks: a ground truth segmentation mask labelled by experts and an output segmentation mask predicted by the network. Calculating AP and AR requires first calculating Precision, Recall, and IoU (Intersection over Union), as shown in Formulas (5–7).

$$Precision = \frac{TP}{TP + FP} \times 100\,\% \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \times 100 \tag{6}$$

TP is true positive which means the number of correctly detect fungal areas, FP as false positive means the number of incorrectly detect fungi areas and FN is false negative which means the number of fugal areas incorrectly detected as background. Precision represents the proportion of TP in the predicted fugal areas and Recall means the proportion of TP in the true fungal areas.

$$IOU = \frac{\text{target} \cap \text{prediction}}{\text{target} \cup \text{prediction}} \times 100\,\% \tag{7}$$

IoU is the metrics to evaluate segmentation accuracy in one category, which calculate the intersection over union between predicted object and ground truth object.

Since the spore image is a small object, we select four metrics, AP, AP50, AP75, and AR, for network performance evaluation. Where AP is io ranging from 0.5 to 0.95 with a step rate of 0.05, AP50 is an iou threshold of 0.50, and AP75 is an iou threshold of 0.75. The higher these values are, the more desirable the instance segmentation model is (Tong et al., 2020).

AR calculates the average recall at different thresholds, how many real objects are correctly detected by the model. AR is the maximum recall of a given fixed number of detections per image, averaged over all IoU and all categories. Since there is only one type of spore, the category is 1. In this study, AR was calculated and averaged over 10 IoU thresholds between 0.5 and 0.95, as shown in Formula (8).

$$p_{\text{interp}}(r) = \max_{\tilde{r}:\tilde{r}\geq r} p(\tilde{r}) \tag{8}$$

where $p(\tilde{r})$ is the measured precision at recall $\tilde{r}$.

AP is calculated by the Formula (9).

$$AP = \frac{1}{11}\sum_{r\in\{0,0.1,\ldots 1\}} p_{\text{interp}}(r) \tag{9}$$

## 3.2 Implementation details

The computational environment for this study utilizes Python 3.7.3 and Ubuntu 18.04 LTS, employing Jupyter Notebook as the editor. The integrated model outlined above was constructed on an Intel(R) Core (TM) i7-12700H (20 CPU) with a 2.30GHz processor, 16 GB of DDR4 RAM, and three graphics cards: two discrete graphics cards (NVIDIA GeForce RTX 3060 laptop GPU with 6023 MB, NVIDIA GeForce RTX 3080 Ti with 12108 MB) along with one integrated graphics card (Intel(R) Iris(R) Xe graphics card with 128 MB), which were utilized for training and testing.

The instance segmentation model was trained with stochastic gradient descent (SGD) method, batch size was set to 4, momentum factor was 0. 9, the initial learning rate was 0.08, and for each epoch, the learning rate changed to 0.9 times of the previous one. The total number of epochs for model training is 100, and when training for the first 60 epochs, the pre-feature extraction network is frozen, and only the neck network and the detection head network are trained in order to improve the training speed of the network model.

TABLE 1 Number of images and spores in the training and test sets in the dataset under different solution treatments, respectively.

| Dataset | | | Carbendazim | DMSO | Solatenol (0.041 ppm) | Solatenol (10 ppm) | TOU-951 |
|---|---|---|---|---|---|---|---|
| Training set | Images | 240 | 48 | 48 | 48 | 48 | 48 |
| | Spore instance | 5526 | 1023 | 1130 | 970 | 1112 | 1291 |
| Test set | Images | 60 | 12 | 12 | 12 | 12 | 12 |
| | Spore instance | 1621 | 306 | 312 | 289 | 302 | 412 |

10.3389/fpls.2024.1340584

## 3.3 Performance comparison with state-of-the art methods and visualization analysis

In order to verify the effectiveness and accuracy of the model in spore instance segmentation, we use the same spore dataset under the same training environment and experimental configuration, and analyze them in comparison with Mask R-CNN, Mask Scoring R-CNN (Huang et al., 2019), YOLACT (Bolya et al., 2019) and Cascade Mask R-CNN (Cai and Vasconcelos, 2019) models. The parameters introduced in 3.2 are used as evaluation indexes to compare the performance with several other methods, and the experimental results are shown in Table 2. All algorithms are trained for 100 epochs, and after each epoch is completed, the mAP values for mask segmentation and box detection are calculated, as shown in Figures 7 and 8.

In this application, detection accuracy refers to the detection of spore individuals from complex environments, detection accuracy refers to the model correctly identifying and localizing spore instances in complex environments, and segmentation accuracy is concerned with the model correctly segmenting each spore at the pixel level, and segmentation accuracy is as important as detection accuracy. As can be seen from the table, our improved algorithm outperforms these classical instance segmentation algorithms in both segmentation accuracy and detection accuracy. From the parameter comparison, the detection accuracy of the model is 0.712 and the segmentation accuracy is 0.618, which are 3.5% and 5% better than the existing optimal methods, respectively. As can be seen in Figures 7 and 8, when the training epoch is less than 20, the advantage of the method is not obvious. However, as the training epoch increases, the detection mAP of our proposed method clearly outperforms these state-of-the-art methods. Our model is more powerful because we not only optimize the backbone network to efficiently extract global and local features; we also introduce a deep multipath feature pyramid network to construct fine-resolution features with strong semantic information; all of these improvements greatly improve the robustness of the CNN to geometric transformations of the target. Our model is able to explore the complex nuances of spore variants, interactions and other related features in greater depth, understand the spatial arrangement and density distribution of spores in an image, effectively resolve spore crossings, overlaps and adhesions, and ensure that each spore is separated as an independent entity.

In order to highlight the superiority of the proposed architecture more intuitively, a visual comparative analysis between the current networks and ours is carried out. As shown in Figures 9A–C, in the figure, are three different growth patterns of spores, where I is the original spore images, and II to VI are Cascade Mask R-CNN, Mask R-CNN, Mask Scoring R-CNN, YOLACT, and ours, respectively.

As can be seen from the confidence level of the anchor box and the box in the figure, the example segmentation results of these advanced networks for a single scattered distribution of spore

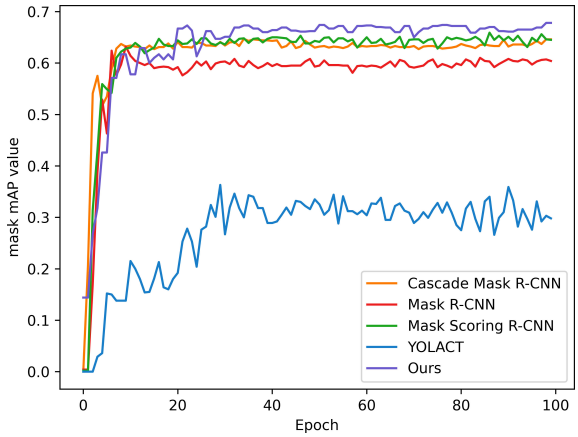



FIGURE 7
Mask segmentation mAP of the model.

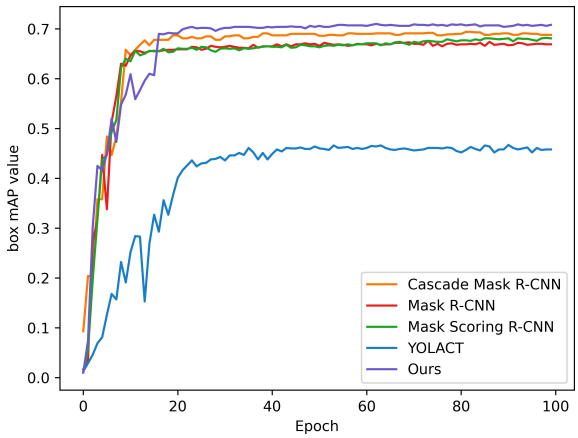



FIGURE 8
Box detection mAP of the model.

TABLE 2 Evaluation results of serious models.

| Models | box_mAP | box_mAP_50 | box_mAP_75 | mask_mAP | mask_mAP_50 | mask_mAP_75 |
|---|---|---|---|---|---|---|
| Cascade Mask R-CNN(ResNet-101) | 0.688 | 0.911 | 0.802 | 0.646 | 0.796 | 0.696 |
| Mask R-CNN(ResNet-101) | 0.669 | 0.938 | 0.754 | 0.604 | 0.757 | 0.619 |
| Mask Scoring R-CNN(ResNet-101) | 0.681 | 0.929 | 0.787 | 0.645 | 0.785 | 0.695 |
| YOLACT(ResNet-101) | 0.458 | 0.836 | 0.49 | 0.298 | 0.442 | 0.036 |
| Ours | 0.712 | 0.936 | 0.809 | 0.678 | 0.828 | 0.724 |

10.3389/fpls.2024.1340584

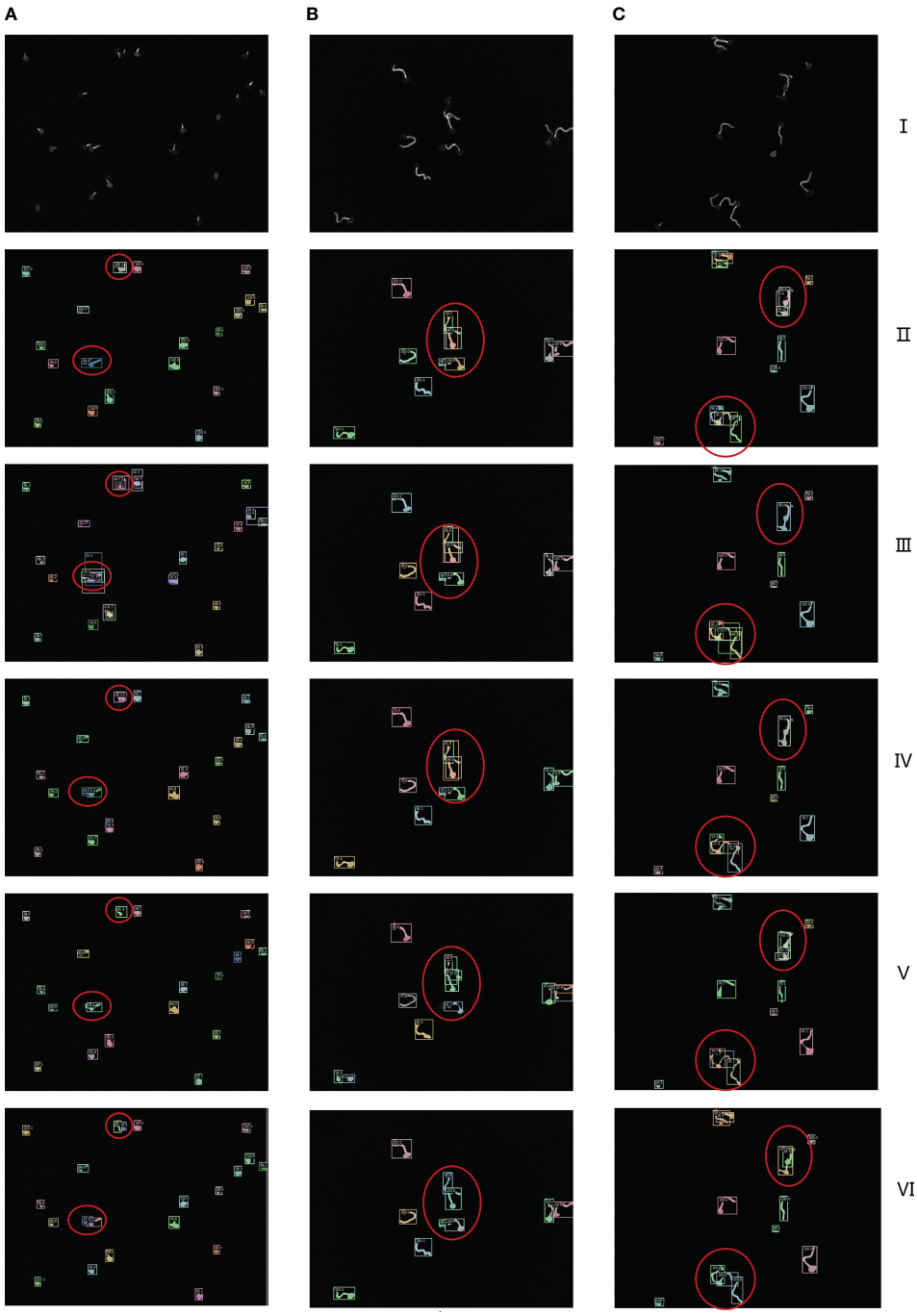



FIGURE 9
I is the original image of spores, II to VI are Cascade Mask R-CNN, Mask R-CNN, Mask Scoring R-CNN, YOLACT, ours network model visualization and analysis images respectively. **(A–C)** are the segmentation results of three randomly selected spore images under different networks.

images are more general. As can be seen from the red circles in the figure, under-segmentation and over-segmentation occur for spore cross, adhesion and overlapping part segmentation with low confidence and lack of refinement and edge processing. In sharp contrast, our network model generated finer detection segmentation images. To further demonstrate the visual analysis results of this network, we performed a zoom-in comparison, as shown in Figure 10.

We chose two images with sparse and tight spore distribution, and both images contain cross overlapping spore features. As can be seen from the figure, the model uses CIoU loss optimization to obtain the optimal prediction box, so that the box detection part can quickly and accurately find the differences between spore individuals with high confidence. At the same time, the model effectively solves the problem of poor robustness of fuzzy pixel segmentation, and the mask segmentation part not only refines the

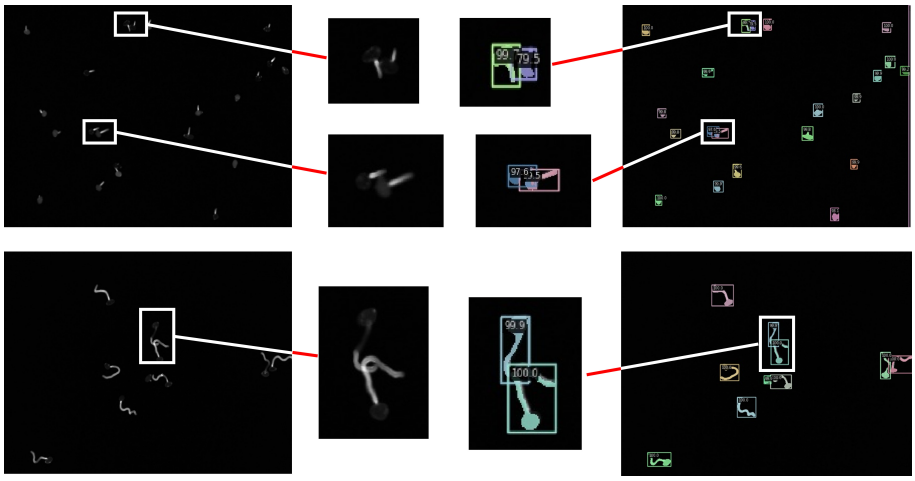


FIGURE 10
Visualization results of the target network.

overall segmentation, but also greatly improves the edge segmentation accuracy, so that the cross-overlapping spore images are separated as separate individuals.

## 3.4 Ablation study of improved models

Based on the improved Mask R-CNN model proposed in this paper, ablation experiments were conducted to compare the experimental results of the original model with the improved ResNet-101, ResNet-101+FPG, Res2Net-101+FPG, and Res2Net +FPGs, as shown in Table 3. Table 3 shows the experimental results of different backbone networks: after adding FPG to the original network, the detection accuracy of the model decreases significantly and the instance segmentation accuracy decreases, but the segmentation accuracy is greatly improved; after replacing ResNet-101, the detection accuracy, segmentation accuracy and accuracy are significantly improved and exceed the original model, which indicates that each module of the improvement has a positive effect on instance segmentation. Finally, the best performing model was found to come from the combined effect of the two improved modules, which improved the detection accuracy, segmentation accuracy, and instance segmentation accuracy by 6.4%, 12.3%, and 2.2%, respectively, compared to the Mask R-CNN model, proving that these improvement strategies of the model are effective.

# 4 Conclusion

In this study, an improved Mask R-CNN method is proposed to accomplish the task of segmentation of densely distributed and overlapping crossed *Phakopsora pachyrhizi* micro-images. The method is optimized and improved from the original MaskR-CNN. Firstly, the res2net was used to replace the backbone network of the original Mask R-CNN by layering the residual connections in a single residual block and then combining it with FPG in order to improve the fine resolution and high-resolution capability, strengthen the feature extraction capability of the network model, and enhance the detection accuracy. Secondly, for the problem of inaccurate edge detection of the original model, the loss function is optimized, and the CIoU loss function is adopted as the loss function of the boundary box regression prediction, which accelerates the convergence speed of the model, meets the accurate segmentation of high-density spore images, and improves the sensitivity of boundary segmentation. Compared with the original model, it is more robust and further improves the accuracy of instance segmentation. In summary, the proposed model can better detect and segment spores under various conditions.

However, this study suffers from an insufficient number of samples in the dataset, and the accuracy of detection and segmentation in the case of spore stacking with a large number of anomalies needs to be further improved. In the follow-up work,

TABLE 3 Evaluation results of ablation experiments.

| Models | box_mAP | mask_mAP | acc | #param. | GFLOPs |
|---|---|---|---|---|---|
| Original Mask R-CNN(ResNet-101) | 0.669 | 0.604 | 95.99 | 63.388M | 302 |
| Mask R-CNN(ResNet-101+FPG) | 0.621 | 0.67 | 94.72 | 63.274M | 147 |
| Mask R-CNN(Res2Net-101+FPG) | 0.69 | 0.678 | 96.77 | 63.951M | 152 |
| Ours | 0.712 | 0.678 | 98.144 | 63.951M | 154 |

collecting and labeling more spore clusters with complex shapes should be considered to expand the spore dataset under different overlap types. Meanwhile, in order to improve the performance of the network, the effect of spore morphology such as length, width and area on segmentation can be deeply investigated, and its features can be fused with image information and input into the segmentation network. Finally, in future research, the scheme proposed in this paper needs to be installed and applied in real scenarios to validate the performance of the model and algorithm. The technique can be applied to perform automatic segmentation of images on microscopes to facilitate the discovery of new drug candidates and the discovery of therapeutic options. Simultaneously, it offers valuable insights to the fields of agriculture, ecology, and medicine, enhancing our understanding and management of fungal-related issues, including disease transmission and ecological balance.

## Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

## Author contributions

YW: Validation, Writing – review & editing. ZX: Writing – original draft. FL: Data curation, Writing – review & editing. WH: Investigation, Writing – review & editing. HF: Writing – review & editing. QZ: Supervision, Writing – review & editing.

## Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work was supported by the National Natural Science Foundation of China (Grant No. 82272534), Shenzhen Key Laboratory of Bone Tissue Repair and Translational Research (NO. ZDSYS20230626091402006).

## Acknowledgments

This work was technically supervised by Professor Fu-Xin Wei of the Seventh Affiliated Hospital of Sun Yat-sen University in the field of cells.

## Conflict of interest

Author HF was employed by the company Zhongzhen Kejian ShenZhen Holdings Co., Ltd.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

Beucher, S. (1992). The watershed transformation applied to image segmentation. *Scanning microscopy* 1992, 28.

Bolya, D., Zhou, C., Xiao, F., and Lee, Y. J. (2019). "Yolact: Real-time instance segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*. 9157–9166. doi: 10.1109/ICCV.2019.00925

Cabre, L., Peyrard, S., Sirven, C., Gilles, L., Pelissier, B., Ducerf, S., et al. (2021). Identification and characterization of a new soybean promoter induced by Phakopsora pachyrhizi, the causal agent of Asian soybean rust. *BMC Biotechnol.* 21, pp.1–pp13. doi: 10.1186/s12896-021-00684-9

Cai, Z., and Vasconcelos, N. (2019). Cascade R-CNN: High quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, pp.1483–1498. doi: 10.1109/TPAMI.2019.2956516

Calderone, R., Sun, N., Gay-Andrieu, F., Groutas, W., Weerawarna, P., Prasad, S., et al. (2014). Antifungal drug discovery: the process and outcomes. *Future Microbiol.* 9, 791–805. doi: 10.2217/fmb.14.32

Carolus, H., Pierson, S., Lagrou, K., and Van Dijck, P. (2020). Amphotericin B and other polyenes—Discovery, clinical use, mode of action and drug resistance. *J. Fungi* 6, 321. doi: 10.3390/jof6040321

Chen, Y., Zhang, H., Liu, J., Zhang, Z., and Zhang, X. (2023). Tapped area detection and new tapping line location for natural rubber trees based on improved mask region convolutional neural network. *Front. Plant Sci.* 13, 1038000. doi: 10.3389/fpls.2022.1038000

Coleman, G. B., and Andrews, H. C. (1979). Image segmentation by clustering. *Proc. IEEE* 67, 773–785. doi: 10.1109/PROC.1979.11327

Gao, S. H., Cheng, M. M., Zhao, K., Zhang, X. Y., Yang, M. H., and Torr, P. (2019). Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 652–662. doi: 10.1109/TPAMI.34

Goellner, K., Loehrer, M., Langenbach, C., Conrath, U. W. E., Koch, E., and Schaffrath, U. (2010). Phakopsora pachyrhizi, the causal agent of Asian soybean rust. *Mol. Plant Pathol.* 11, 169–177. doi: 10.1111/j.1364-3703.2009.00589.x

Hariharan, B., Arbeláez, P., Girshick, R., and Malik, J. (2014). "Simultaneous detection and segmentation," in *Computer Vision–ECCV 2014* (Cham: Springer International Publishing), 297–312. Available at: http://dx.doi.org/10.1007/978-3-319-10584-0_20

Hartman, G. L., Chang, H. X., and Leandro, L. F. (2015). Research advances and management of soybean sudden death syndrome. *Crop Prot.* 73, 60–66. doi: 10.1016/j.cropro.2015.01.017

Huang, Z., Huang, L., Gong, Y., Huang, C., and Wang, X. (2019). "Mask scoring r-cnn" in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6409–6418. doi: 10.1109/CVPR.2019.00657

Jia, W., Wei, J., Zhang, Q., Pan, N., Niu, Y., Yin, X., et al. (2022). Accurate segmentation of green fruit based on optimized mask RCNN application in complex orchard. *Front. Plant Sci.* 13, 955256. doi: 10.3389/fpls.2022.955256

Korsnes, R., Westrum, K., Fløistad, E., and Klingen, I. (2016). Computer-assisted image processing to detect spores from the fungus Pandora neoaphidis. *MethodsX* 3, 231–241. doi: 10.1016/j.mex.2016.03.011

Lei, Y., Yao, Z., and He, D. (2018). Automatic detection and counting of urediniospores of Puccinia striiformis f. sp. tritici using spore traps and image processing. *Sci. Rep.* 8, 13647. doi: 10.1038/s41598-018-31899-0

Li, K., Zhu, X., Qiao, C., Zhang, L., Gao, W., and Wang, Y. (2023). The gray mold spore detection of cucumber based on microscopic image and deep learning. *Plant Phenomics* 5, 0011. doi: 10.34133/plantphenomics.0011

Liberali, P., Snijder, B., and Pelkmans, L. (2015). Single-cell and multivariate approaches in genetic perturbation screens. *Nat. Rev. Genet.* 16, 18–32. doi: 10.1038/nrg3768

Loehrer, M., and Schaffrath, U. (2011). Asian soybean rust–meet a prominent challenge in soybean cultivation. *Soybean–biochemistry Chem. Physiol. Rijeka Croatia: InTech*, 83–100. doi: 10.5772/15651

Long, J., Shelhamer, E., and Darrell, T. (2015). "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3431–3440. doi: 10.1109/cvpr.2015.7298965

Lorrain, Cécile, Gonçalves dos Santos, K. C., Germain, H., Hecker, A., and Duplessis, Sébastien (2019). Advances in understanding obligate biotrophy in rust fungi. *New Phytol.* 222, 1190–1206. doi: 10.1111/nph.15641

Melo, C. A. D., Lopes, J. G., Andrade, A. O., Trindade, R. M., and Magalhaes, R. S. (2019). Semi-automated counting model for arbuscular mycorrhizal fungi spores using the Circle Hough Transform and an artificial neural network. *Anais da Academia Bras. Ciências* 91, e20180165. doi: 10.1590/0001-3765201920180165

Okada, H., Ohnuki, S., Roncero, C., Konopka, J. B., and Ohya, Y. (2014). Distinct roles of cell wall biogenesis in yeast morphogenesis as revealed by multivariate analysis of high-dimensional morphometric data. *Mol. Biol. Cell* 25, 222–233. doi: 10.1091/mbc.e13-07-0396

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Trans. systems man cybernetics* 9, 62–66. doi: 10.1109/TSMC.1979.4310076

Reisen, F., De Chalon, A. S., Pfeifer, M., Zhang, X., Gabriel, D., and Selzer, P. (2015). Linking phenotypes and modes of action through high-content screen fingerprints. *Assay Drug Dev. Technol.* 13, 415–427. doi: 10.1089/adt.2015.656

Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2008). LabelMe: a database and web-based tool for image annotation. *Int. J. Comput. Vision* 77, 157–173. doi: 10.1007/s11263-007-0090-8

Saito, H., Yamashita, Y., Sakata, N., Ishiga, T., Shiraishi, N., Usuki, G., et al. (2021). Covering soybean leaves with cellulose nanofiber changes leaf surface hydrophobicity and confers resistance against Phakopsora pachyrhizi. *Front. Plant Sci.* 111, 146–147. doi: 10.3389/fpls.2021.726565

Seki, K., and Toda, Y. (2022). QTL mapping for seed morphology using the instance segmentation neural network in Lactuca spp. *Front. Plant Sci.* 13, 949470. doi: 10.3389/fpls.2022.949470

Tong, K., Wu, Y., and Zhou, F. (2020). Recent advances in small object detection based on deep learning: A review. *Image Vision Computing* 97, 103910. doi: 10.1016/j.imavis.2020.103910

Usaj, M. M., Styles, E. B., Verster, A. J., Friesen, H., Boone, C., and Andrews, B. J. (2016). High-content screening for quantitative cell biology. *Trends Cell Biol.* 26, 598–611. doi: 10.1016/j.tcb.2016.03.008

Xie, X., Wang, J., Hu, Z., and Zhao, Y. (2021). "Intelligent detection of mango disease spores based on mask scoring R-CNN," in *2021 5th Asian Conference on Artificial Intelligence Technology (ACAIT)*. 768–774 (IEEE), October. Available at: http://dx.doi.org/10.1109/acait53529.2021.9731325

Yang, L., Ghosh, R. P., Franklin, J. M., Chen, S., You, C., Narayan, R. R., et al. (2020). NuSeT: A deep learning tool for reliably separating and analyzing crowded cells. *PloS Comput. Biol.* 16, e1008193. doi: 10.1371/journal.pcbi.1008193

Zanella, F., Lorens, J. B., and Link, W. (2010). High content screening: seeing is believing. *Trends Biotechnol.* 28, 237–245. doi: 10.1016/j.tibtech.2010.02.005

Zhang, D. Y., Zhang, W., Cheng, T., Zhou, X. G., Yan, Z., Wu, Y., et al. (2023). Detection of wheat scab fungus spores utilizing the Yolov5-ECA-ASFF network structure. *Comput. Electron. Agric.* 210, 107953. doi: 10.1016/j.compag.2023.107953

Zhao, Y., Lin, F., Liu, S., Hu, Z., Li, H., and Bai, Y. (2019). Constrained-focal-loss based deep learning for segmentation of spores. *IEEE Access* 7, 165029–165038. doi: 10.1109/Access.6287639

Zhao, Y., Liu, S., Hu, Z., Bai, Y., Shen, C., and Shi, X. (2020). Separate degree based Otsu and signed similarity driven level set for segmenting and counting anthrax spores. *Comput. Electron. Agric.* 169, 105230. doi: 10.1016/j.compag.2020.105230

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 28 March 2024
DOI 10.3389/fpls.2024.1340884

Check for updates

OPEN ACCESS

EDITED BY
Tonghai Liu,
Tianjin Agricultural University, China

REVIEWED BY
Guoxiong Zhou,
Central South University Forestry and Technology, China
Xiujun Zhang,
Chinese Academy of Sciences (CAS), China

*CORRESPONDENCE
Hongchun Qu
hcchyu@gmail.com

RECEIVED 19 November 2023
ACCEPTED 26 February 2024
PUBLISHED 28 March 2024

CITATION
Qu H, Zheng C, Ji H, Huang R, Wei D, Annis S and Drummond F (2024) A deep multi-task learning approach to identifying mummy berry infection sites, the disease stage, and severity.
*Front. Plant Sci.* 15:1340884.
doi: 10.3389/fpls.2024.1340884

COPYRIGHT
© 2024 Qu, Zheng, Ji, Huang, Wei, Annis and Drummond. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# A deep multi-task learning approach to identifying mummy berry infection sites, the disease stage, and severity

Hongchun Qu [1,2]*, Chaofang Zheng [1], Hao Ji [1,2], Rui Huang [2], Dianwen Wei [3], Seanna Annis [4,5] and Francis Drummond [4,5]

[1]Institute of Ecological Safety and College of Automation, Chongqing University of Posts and Telecommunications, Chongqing, China, [2]College of Computer Science, Chongqing University of Posts and Telecommunications, Chongqing, China, [3]Institute of Natural Resources and Ecology, Heilongjiang Academy of Sciences, Harbin, China, [4]School of Biology and Ecology, University of Maine, Orono, ME, United States, [5]Cooperative Extension, University of Maine, Orono, ME, United States

**Introduction:** Mummy berry is a serious disease that may result in up to 70 percent of yield loss for lowbush blueberries. Practical mummy berry disease detection, stage classification and severity estimation remain great challenges for computer vision-based approaches because images taken in lowbush blueberry fields are usually a mixture of different plant parts (leaves, bud, flowers and fruits) with a very complex background. Specifically, typical problems hindering this effort included data scarcity due to high manual labelling cost, tiny and low contrast disease features interfered and occluded by healthy plant parts, and over-complicated deep neural networks which made deployment of a predictive system difficult.

**Methods:** Using real and raw blueberry field images, this research proposed a deep multi-task learning (MTL) approach to simultaneously accomplish three disease detection tasks: identification of infection sites, classification of disease stage, and severity estimation. By further incorporating novel superimposed attention mechanism modules and grouped convolutions to the deep neural network, enabled disease feature extraction from both channel and spatial perspectives, achieving better detection performance in open and complex environments, while having lower computational cost and faster convergence rate.

**Results:** Experimental results demonstrated that our approach achieved higher detection efficiency compared with the state-of-the-art deep learning models in terms of detection accuracy, while having three main advantages: 1) field images mixed with various types of lowbush blueberry plant organs under a complex background can be used for disease detection; 2) parameter sharing among different tasks greatly reduced the size of training samples and saved 60% training time than when the three tasks (data preparation, model development and exploration) were trained separately; and 3) only one-sixth of the network parameter size (23.98M vs. 138.36M) and one-fifteenth of the computational cost (1.13G vs. 15.48G FLOPs) were used when compared with the most popular Convolutional Neural Network VGG16.

**Discussion:** These features make our solution very promising for future mobile deployment such as a drone carried task unit for real-time field surveillance. As an automatic approach to fast disease diagnosis, it can be a useful technical tool to provide growers real time disease information that can prevent further disease transmission and more severe effects on yield due to fruit mummification.

KEYWORDS

**mummy berry disease, multi-task learning, transfer learning, convolutional neural network, blueberry, *Monilinia vaccinii-corymbosi*, *Vaccinium angustifolium***

## 1 Introduction

Lowbush blueberry or wild blueberry (*Vaccinium angustifolium* Aiton), is a North American native berry crop which is both economically and culturally important in Maine and other Northern New England states in the US, the Canadian Maritime provinces and the province of Quebec in Canada. The lowbush blueberry production system can be a major source of income for growers in this region (Hanes et al., 2015). The state of Maine is the world's largest producer of wild blueberry (Obsie et al., 2020). Nearly 67.7 million pounds were produced in 2017 (*nass.usda.gov*). However, lowbush blueberry is highly susceptible to mummy berry disease caused by the fungus *Monilinia vaccinii-corymbosi* (Read) (MVC) (Penman and Annis, 2005). Mummy berry is a serious disease for many *Vaccinium* species and can cause up to 70% of yield loss, especially in the moist oceanic climate where it grows in Maine and Maritime Canada, posing an economic challenge to growers and affecting local economies (Hanes et al., 2015).

Correctly identifying mummy berry infection in its early stages and providing a reasonable estimation of disease severity could play a vital role in efficient disease control and prevention in the future. Because management procedures such as determining the timing and dose of fungicide sprays, as well as the introduction of commercially available honey bees (Qu and Drummond, 2018) relies upon sufficient information regarding where, when and how blueberry plants are infected (Penman and Annis, 2005). Specific information on the identification of infection sites on leaves, flowers or fruits; classification of infection stages; and estimation of disease severity will be useful in managing mummy berry disease. Considering the range in size of lowbush blueberry fields, which can be up to hundreds of hectares, automatic mummy berry infection identification techniques such as Computer Vision and Image Processing (CVIP) (Vishnoi et al., 2021) are advantageous, not only because of their extensive labor-saving potential, but also because of the potential comprehensive disease relevant measurements that can be used in directing chemical treatments across large crop land areas over time (Bock et al., 2010). In the near future, a drone carried disease detection task unit based on CVIP would be a very efficient tool for field disease surveillance (Liu and Wang, 2021).

Computer Vision and Image Processing methods have previously been used in a wide variety of diagnosis applications in precision agriculture such as plant species classification, leaf disease recognition, and plant disease severity estimation (Liang et al., 2019). In the last decade, many traditional machine learning models were proposed for the detection and classification of plant diseases. Rumpf et al. (2010) studied the early diagnosis and classification of diseases infecting sugar beet based on spectral plant indexes using Support Vector Machine (SVM). Ramesh et al. (2018) proposed a disease classification method using Random Forest algorithms to identify healthy and diseased images. Waghmare et al. (2016) proposed a Multiclass Support Vector Machine as a classification model for grape leaves and they identified diseases like Black rot with a reported accuracy of 96.6%.

Nevertheless, these traditional CVIP methods are heavily dependent on experience such as manual selection of disease spot features plus artificial classifiers, which inevitably lead to objective disease feature extraction (Sastry and Zitter, 2014). In the real environment, challenges, such as low contrast, high noise of lesions with respect to the background, large variations in size and scale of the target area, unstable illumination conditions, and image capture shooting angle, can compromise the practicality of these conventional CVIP methods (Alruily, 2021). In contrast, deep learning approaches, due to their excellent automatic feature engineering and self-learning capabilities, have resulted in state-of-the-art performance when compared to traditional CVIP approaches in different domains (Alom et al., 2019). Among deep learning methods, Convolutional Neural Networks (CNN) have shown extraordinary performance in image recognition tasks (Fuentes et al., 2017). Geetharamani and Arun Pandian (2019) trained a 9-layer CNN architecture on the PlantVillage dataset with different epoch, batch size and dropout rate. In the performance comparison with popular transfer learning approaches, the proposed model achieved 96.5% classification accuracy on the test dataset. Lu et al. (2017) proposed a technique to enhance the identification ability of CNNs to effectively classify 10 rice diseases through deep convolution neural networks. A similar approach was also developed by Ferentinos (2018) to recognize and diagnose plant diseases based on simple leaf images of healthy

and diseased crops. In addition, Fuentes et al. (2017) developed a practical solution based on a robust CNNs-based detector for real-time recognition of diseases and pests of tomato plants.

Although the existing deep learning techniques have achieved significant success in plant disease detection and diagnosis, they usually train a single model or a set of separate models to solve a specific task. For example, when addressing the two tasks, such as plant disease identification and severity estimation, it is common to train two separate models to optimize their respective performance metrics. This idea of problem solving is called single-task learning (STL). Single-task learning ignores over-lapping information between related tasks that might be very helpful for enhancing model generality and promoting model performance. Alternatively, multi-task learning (MTL) can train a system capable of solving multiple tasks simultaneously by sharing representations between them (Ruder, 2017). By doing so, MTL can not only achieve higher detection accuracy than STL by learning joint generalized representations (Zhang and Yang, 2017), but can also sufficiently decrease overfitting risk by using the domain information contained in the training signals of related tasks as an inductive bias. Furthermore, the joint training of MTL can also shrink the size of training data (equivalent to data augmentation), which is critical in dealing with data deficient problems (Caruana, 1997).

To simultaneously identify mummy berry infection sites, classify disease stages and estimate the severity level for real farming conditions is difficult. Unlike controllable conditions in the laboratory, field-taken images of lowbush blueberry usually have mixed plant parts (leaves, buds, flowers and fruits) embedded in a very complex background with low contrast posing great challenges for current deep learning models. The first one is data scarcity due to the extremely high manual labelling cost. This is the major obstacle for training deep learning models for different tasks. The second is the tiny, low contrast disease features that interfered with and occluded by a complex environment. This requires precisely capturing these features across various spatial scales. The third is the larger parameter size and difficulty in training and using it for real time application, which requires a smaller deep model favoring both training convergence and responding speed.

Therefore, the main goal of this research was to propose a novel deep learning model that integrates the techniques, such as residual learning, coordinate attention mechanism, and group convolution, into a deep multi-task learning (MTL) approach to simultaneously accomplish three disease detection tasks of identifying the infection site, classifying the infection stage, and estimating disease severity. This deep MTL model is expected to achieve higher detection efficiency compared with eight state-of-the-art deep learning models in terms of detection accuracy and parameter size. The key contributions of this research are as follows:

1) A deep multi-task learning (MTL) approach was developed to simultaneously accomplish three mummy berry disease detection tasks with limited data;
2) Novel superimposed attention mechanism modules applied to deep learning was found to enhance disease feature extraction from both channel and spatial perspectives, enabling better performance in an open and complex environment compared to other CNNs;
3) Integrating grouped convolution to MTL enabled it to learn a varied set of low-level and high-level disease features in a more parallel manner, resulting in a significant reduction in computational complexity and faster convergence.

# 2 Materials and methods

## 2.1 Overview

This research investigated a deep learning based MTL framework to simultaneously identify infection sites (where diseased tissues are located), classify the infection stage (primary or secondary stage) and estimate the infection severity from raw images of lowbush blueberry stems. The task of building a deep MTL framework contains three main procedures represented as blocks, which are data preparation, model development and model exploration (Figure 1). In the data preparation block, healthy and diseased blueberry images were collected from fields and online resources and were manually labeled with the infection status. These labeled images were varied for the purpose of augmentation and were randomly distributed into three training datasets and three testing datasets. In the model development block, EfficientNet (Tan and Le, 2019) was used as the protype to establish the deep MTL framework having one parameter sharing module and three task-specific modules. Once the deep MTL framework (model) was formed, four instances of it with variations in parameters and configurations were generated, trained and tested. In the model exploration block, the four model instances were first compared with the state-of-the-art deep solutions in terms of accuracy and F1-score (see section 3.2 Evaluation metrics), then the advantages of using multi-task and transfer learning were determined by two sets of ablation experiments. Finally, several applications of mummy berry disease detection and feature maps of the deep MTL structure were visualized.

## 2.2. Preliminary

The life cycle of mummy berry disease contains two distinct stages. In the primary infection stage, successfully overwintered pseudosclerotia beneath fallen blueberry leaves start developing apothecia which discharge sexual spores (ascospores) in the early spring to infect leaves and flowers. Infected leaves often turn from greenish red or greenish pink color to a rosy brown and sometimes form a shepherd's crook or curl. The infected emerging floral buds usually have a brown discoloration or blighted appearance (McGovern et al., 2012). Following primary infection, secondary asexual spores (conidia), appearing as a white to grey powder, are produced on the infected, blighted leaves and flowers. The production of asexual spores (conidia) is called sporulation and is regarded as the phase immediately preceding secondary infection,

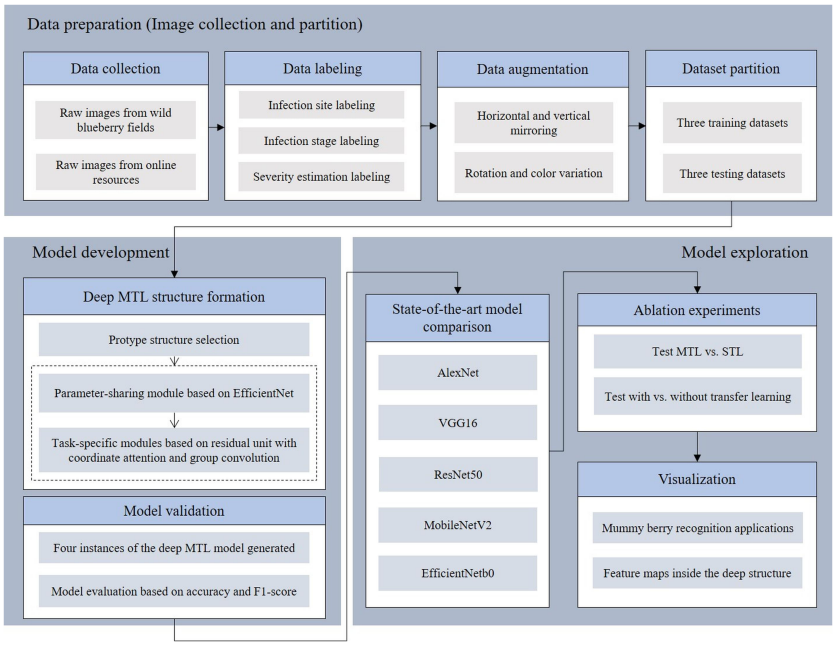



FIGURE 1
Overview of the three tasks of the mummy berry disease detection research: data preparation, model development and exploration.

although for our modeling we included sporulation as the initial stage of secondary infection because it is the conidia which are vectored to the flower resulting in flower infection and subsequent fruit infection that result in secondary infection. Conidia are vectored by insect pollinators to open flowers, the conidia then germinate and grow down the style to reach and infect the ovules. Infected immature blueberry fruit initially appears waxy green, but begins to discolor as the disease develops. Finally, the infected mature berries become gray and shriveled while healthy mature berries are a waxy blue to purplish color. Secondary infection subsequently results in a mummified berry (Batra, 1983; Batra and Batra, 1985).

In reviewing the literature on the pathological characteristics of mummy berry disease, we emphasize that the symptoms occur only on leaves (primary infection), flowers (primary infection) and fruits (secondary infection) of lowbush blueberry plants. The color and shape of the infected plant organs are identifiable features that indicate the infection site. Since primary infection symptoms are only present in leaves and flowers, these can be distinguished by features such as white, powdery conidia on the sporulating organs just prior to secondary infection, and the mummified berries can act as a secondary infection symptom, so that the two infection stages can be classified from images. Also, the number of infected tissues and their occupied area on the image can provide useful information for a disease severity estimation. Therefore, our study site identification refers to the classification between healthy and diseased leaves, flowers and fruits (Figure 2). Stage classification is to distinguish infection phases by examining the symptoms that are exclusively featured by the primary or secondary infection stages (Figure 3). Severity estimation was performed by calculating the percentage of area occupied by the diseased tissues in relation to the whole image (Figure 4).

## 2.3 Dataset

### 2.3.1 Data collection

The dataset created for this study contains raw images of healthy and diseased blueberry flowers, leaves and fruits. Our primary image source was research scientists at the University of Maine, Orono, Maine USA. Over the past several years, hundreds of lowbush blueberry images were acquired from the University of Maine's lowbush blueberry experimental fields. These images were taken in a wild blueberry growing environment with a complex background. However, the number of raw blueberry images were still far less than the requirement of deep neural network training and validation. Therefore, we also used python Scrapy (www.google.com) to search and extract online mummy berry images, such as Google and the National Ecological Observatory Network (Bugwood.org) to expand our dataset. The extra images collected online not only helped to alleviate a data deficiency problem, but were useful for generalizing the training features of mummy berry disease, which is an effective way of reducing the risk of overfitting (Perez and Wang, 2017).

### 2.3.2 Data labeling

A total of 927 raw blueberry images were collected which includes the categories of healthy and diseased flowers, leaves and fruits. The labeling process aimed at manually classifying each image to the corresponding categories according to its feature.




FIGURE 2
Examples of mummy berry infection site: healthy flowers **(A)**, healthy leaves **(B)**, healthy fruits **(C)**, infected flowers **(D)**, infected leaves **(E)** and infected (mummified) fruits **(F)**.

For example, an image having the content of mummified berries was labeled as infected fruit and secondary infection stage. The labeling process was assisted by domain experts with their pathological expertise in mummy berry disease. Based on the obtained images, three datasets were generated to solve the three tasks of disease recognition. The three datasets were named as infection site identification dataset (Figure 2), infection stage classification dataset (Figure 3) and severity estimation dataset (Figure 4).

The infection site identification dataset consists of the images featuring healthy and diseased flowers, leaves and fruits, which was divided into 6 categories (Figure 2, Table 1); the infection stage classification dataset contains three categories of symptom images. The first is the healthy category that consists of healthy flowers, leaves and fruits. The second category is the primary infection stage involving infected pinkish leaves and emerging floral buds. The third category is the secondary infection stage featuring sporulating leaves and flowers, as well as mummified fruits (Figure 3, Table 2): the severity estimation dataset was created by calculating the percentage of area occupied by infected tissues in the whole image. This calculation was done by using an image segmentation method (Haralick and Shapiro, 1985) with manual corrections where necessary. For certain severity ranges, labels were assigned as follows: healthy (< 0.5%), very low (0.5% ~ 5%), low (5% ~ 10%), high (10% ~ 15%) and very high (>15%) (Figure 4, Table 3).

### 2.3.3 Data augmentation

Complex models particularly deep learning ones tend to suffer overfitting when trained with small narrowly constituted datasets (Shorten and Khoshgoftaar, 2019). To address this issue, data augmentation techniques were used to generate synthetic samples of the raw data in order to increase the generalization ability of the deep model (Perez and Wang, 2017). In this work, two distinct techniques of data augmentation were employed. The first data augmentation technique utilizes horizontal and vertical mirroring (Figures 5B, C) of the original images (Figure 5A). While the second data augmentation technique adjusts brightness, contrast, and saturation of the original images (Taylor and Nitschke, 2018) (Figures 5D–F). By doing this, the number of images in the original three-category dataset was increased. Considering the balance of data between categories, the total images in each of the three datasets were expanded to approximately 1200 images.

## 2.4 The model

### 2.4.1 Theory of deep multi-task learning

One challenge of mummy berry disease detection is that the images of blueberry plants are usually a mixture of different plant parts with various visual focal distances and complex backgrounds. This challenge greatly restricts the conventional image segmentation methods for feature extraction and consequently causes a large uncertainty in recognition and classification. Deep learning architecture can take advantage of automated feature engineering to avoid the dependence on prior domain knowledge and human interventions in feature extraction. Convolutional neural networks (CNNs) are typical representations of deep learning architecture for image-based classification. The

10.3389/fpls.2024.1340884




FIGURE 3
Examples of infection stage: healthy (row **(A)**, primary infection (row **(B)**, and sporulation on leaves initiating secondary infection, and secondary infection of mummified fruit (row **(C)**.

procedure of any CNN-based deep learning solution can be divided into two separated working phases, which are feature extraction and feature classification. The first phase takes raw images and outputs a feature vector to the second phase. Then the later processes the feature vector and outputs the result according to the specific classification requirements.

Another challenge using MTL is that compared with single task solutions, the accomplishment of multiple tasks (i.e., infection site identification, stage classification and severity estimation), may require larger training sample sizes, take more time to train, and experience higher risks of overfitting data. However, this problem can be solved if we sufficiently utilize the common knowledge among tasks instead of processing them separately, which leads to the multi-task leaning(MTL) concept in a deep learning architecture (Zhang and Yang, 2017). Multi-task Learning aims to take advantage of useful information obtained from multiple related tasks and by doing so, helps improve the generalization performance of all the tasks (Caruana, 1997). In order to fully characterize MTL, we provide a commonly accepted definition of MTL (Ruder, 2017).

### 2.4.2 MTL Definition

Given $m$ learning tasks $\{T_i\}_{i=1}^{m}$ where all the tasks or a subset of them are related, multi-task learning aims to help improve the learning ability of a model for $T_i$ by using the knowledge contained in all or in part of the $m$ tasks. Based on the definition of MTL, we focus on supervised learning tasks since images in this study were already precisely labeled. In the setting of supervised learning tasks,




FIGURE 4
Examples of primary infection severity estimation: healthy **(A)**, very low **(B)**, low **(C)**, high **(D)**, very high **(E)**.

usually a task $T_i$ is accompanied by a training dataset $Di$ consisting of $n_i$ training samples, i.e., $D_i = \left\{X_j^i, y_j^i\right\}_{j=1}^{n_i}$, where $X_j^i \in R^{d_i}$ is the $j^{th}$ training instance in $T_i$ and $y_j^i$ is its label. We denote $\boldsymbol{X}_i$ as the training data matrix for $T_i$, i.e., $X_i = (X_1^i, \ldots, X_{n_i}^i)$. Here we consider a general setting for MTL that all the $\boldsymbol{X}_i$ are different from each other. Therefore, our theoretical solution was that within a CNN architecture, firstly we employed a hard parameter-sharing technique (Caruana, 1993) to extract feature vectors for the three tasks because we hypothesized that many features are likely to be shared between images in the infection site, stage and severity datasets. For example, features indicating infection stage can be used to infer the site of infection, e.g., pixels of mummies can also be used to identify the secondary infection stage. Once the feature vectors were produced, we utilized the three task-specific deep structures to simultaneously perform site identification, stage classification and severity estimation. The theoretical solution is summarized in Figure 6 (the technical details are given in the following implementation section).

### 2.4.3 Deep MTL model implementation

The Deep MTL model for mummy berry disease recognition is composed of four modules: one Parameter-sharing module and three Task-specific modules Figure 6. The Parameter-sharing module is the most front basic module and takes raw pixels of plant images as input and automatically extracts features feeding the Task-specific modules. The Parameter-sharing module was constructed based on the basal deep structure of EfficientNet, which is a family of CNNs released in 2019 by Google AI (Tan and Le, 2019). We chose EfficientNet as the protype to implement the Parameter-sharing module of the Deep MTL because it offers excellent scaling ability in several dimensions such as network width, depth, and image resolution in either a simple or compound manner. This scaling ability allows EfficientNet to achieve ideal balance between accuracy and the size of network parameters.

Specifically, the Parameter-sharing module was made up of nine phase operations Table 4. We intentionally removed the final phase operation of the EfficientNet to compress the output channel. Therefore, the output channel of the Parameter-sharing module was compressed to 320 channels.

The Task-specific module was constructed based on the residual unit (He et al., 2016) and featuring was performed with the attention mechanism (Hou et al., 2021) and the group convolution operation (Su et al., 2020) because successfully identifying fine features such as conidia on sporulating tissues is critical for infection stage classification. The Residual unit was employed to address the degradation problem and difficulties in learning identity maps for multiple non-linear layers, which have been proven effective in many visual tasks. Attention mechanisms (AM), used to "tell" a model 'what' and 'where' to attend, have been extensively studied (Mnih et al., 2014; Xu et al., 2015) and widely deployed for boosting the performance of modern deep neural networks. There are three typical AMs, i.e., The Squeeze-and-Excitation Networks (SE) (Hu et al., 2018) focuses on the attention information of the feature channels, the Convolutional Block Attention Module (CBAM) (Woo et al., 2018) uses both channel and spatial AM in a serial manner, and the Collaborative attention mechanism (CA) (Hou et al., 2021) employes both channel and spatial AM, but in a parallel manner.

In general, a visual attention mechanism can enhance information extraction from a channel or spatial perspective, or both. Channel-based AM boosts specific feature layers (i.e., channels) possessing more interesting information and lessens others in the feature map, while the spatial-based AMs can focus on specific interesting region of the feature space and ignore the background. The different characteristics of AM varieties and their successful applications indicate that the combination of these AMs in deep neural networks could be effective to solve this problem.

TABLE 1 The number of images in each of the six categories of the infection site identification dataset after data augmentation.

| Infection site identification dataset | | | |
|---|---|---|---|
| Healthy flower | 222 | Infected flower | 144 |
| Healthy leaf | 104 | Infected leaf | 90 |
| Healthy fruit | 172 | Infected fruit | 370 |
| | | Total | 1102 |

TABLE 2 The number of images in each of the three categories of infection stage classification dataset after data augmentation.

| Infection stage classification dataset | | | |
|---|---|---|---|
| Healthy | Primary infection | Secondary infection | Total |
| 505 | 384 | 336 | 1225 |

TABLE 3 The number of images in each of the five categories of severity estimation dataset after data augmentation.

| Severity estimation dataset | | | | | |
|---|---|---|---|---|---|
| Healthy | Very low | Low | High | Very high | Total |
| 531 | 251 | 112 | 130 | 204 | 1228 |

Because focusing on features of interest across various spatial scale and abstract levels could be vital for extracting disease features from complex backgrounds with likely occlusions.

We proposed a superposed structure combining the three attention mechanisms (SE, the CBAM, CA) in the CNN. After the convolutional layer, the features are learned in different channels by the SE attention mechanism, then by CBAM attention mechanism, learning through both the channel and space. Next, they are passed to the CA attention module to extract feature information as well as feature location information on the feature channels, and finally passed to the pooling layer and the fully connected layer. Figure 7 delineates the structure of the Task-specific module.

The grouped convolution was also applied to use different sets of convolution filter groups on the same input image. It allows one to create two or more deep learning models that can be trained and backpropagated in a parallel manner (Su et al., 2020). In other words, this approach creates a deep network with a limited number of layers, so that they are replicated to form multiple pathways for convolutions on a single image.

The Parameter-sharing module and the three Task-specific modules formed the main structure of the Deep MTL model, which was named in this study as the MummyBerryNet. The workflow of MummyBerryNet is as follows. First, the raw images of the three tasks are alternately input into the Parameter-sharing module to obtain three feature vectors. Then, each feature vector is input into the corresponding Task-specific module, and finally the classification result is obtained through the fully connected layer. The architecture of MummyBerryNet is described in Table 5.

### 2.4.4 Four instances of the deep MTL model

The baseline network, EfficientNet, is a family of eight versions of CNNs with different tradeoffs between performance and network size ranging from 5.3M to 66M parameters (Tan and Le, 2019). It provides a variety of options to achieve an expected accuracy with an affordable computational cost. Considering the uncertainty and complexity of mummy berry disease recognition, the exploration of the fitness of these models to the real and complex environment is worthwhile. Therefore, our solution was to choose four models to generate four instances of the Parameter-sharing module. The selection criterion




FIGURE 5
Example of data augmentation: original image **(A)**. Images **(B, C)** represent variants by flipping the original image horizontally and vertically, whereas images **(D-F)** represent variants by corrupting the original image in brightness, contrast and saturation, respectively.

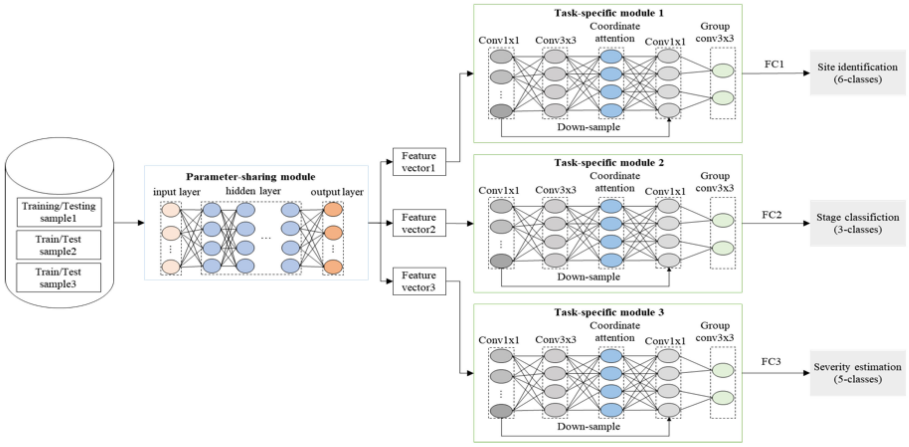



FIGURE 6
Architecture of our Deep MTL model, which is composed of a parameter-sharing module and three task-specific modules for simultaneously solving the three tasks (infection site identification, stage classification and severity estimation). FC stands for Full Connected Layer.

used is as follows: we estimated the accuracy-cost ratio of eight versions and selected the top four as the candidate baseline networks (Figure 1. Model Size vs. ImageNet Accuracy of Tan and Le, 2019), which were EfficientNet-B0, B1, B2 and B3. Their parameter sizes were therefore compressed into the range between 5.3M and 12M. The four instances of the Deep MTL model, which was previously named as MummyBerryNet, were: MummyBerryNet-B0, MummyBerryNet-B1, MummyBerryNet-B2 and MummyBerryNet-B3. The four instances of MummyBerryNet were different in the Parameter-sharing module but were almost the same in the Task-specific modules, as shown in Table 6, Tables A1-A4 (see Appendix). After model implementation, we then trained for and conducted experiments on each of the four instances independently.

### 2.4.5 Training the deep MTL model

In the training process, we converted the input images (i.e., the training samples obtained from offline data augmentation, see subsection 2.3.3.) with the fixed size of 224 × 224 × 3 to meet the input size requirements of MummyBerryNet. The number of images in the training and testing datasets were kept at a 70/30 ratio. To train MummyBerryNet for the three mummy berry recognition tasks, several batches of computation were conducted. MummyBerryNet received new image samples from each new training batch. Each sample consists of images from all three tasks. The network weights were adjusted repeatedly until MummyBerryNet learned the most relevant discriminative features for a given task, i.e., the cross-entropy loss of the deep neural network converged. The training was performed by adapting pre-trained networks on the ImageNet dataset, which was achieved by means of transfer learning (details below). The four instances of MummyBerryNet were trained end-to-end, without freezing the training of any other layers. The stochastic gradient descent (SGD) algorithm was employed to improve the performance for all experiments. The learning rates were dynamically decreased by 1/10 at every 15 epochs during training, with the initial learning rate set at 0.005 for the first step. The weight decay of 0.0001 and batch size of 16 were used in the training process. The specifications of the optimizer and parameters for training MummyBerryNet are listed

TABLE 4 The structure of the Parameter-sharing module based on the EfficientNet baseline network – Each row describes a phase $i$ with $L_i$ layers, with input resolution $<H_i, W_i>$ and output channels $C_i$ (Tan and Le, 2019).

| Phase $i$ | Operator $F_i$ | Resolution $H_i \times W_i$ | Channels $C_i$ | Layers $L_i$ |
|---|---|---|---|---|
| 1 | Conv3×3 | 224×224 | 32 | 1 |
| 2 | MBConv1, K3×3 | 112×112 | 16 | 1 |
| 3 | MBConv6, K3×3 | 112×112 | 24 | 2 |
| 4 | MBConv6, K5×5 | 56×56 | 40 | 2 |
| 5 | MBConv6, K3×3 | 28×28 | 80 | 3 |
| 6 | MBConv6, K5×5 | 14×14 | 112 | 4 |
| 7 | MBConv6, K5×5 | 14×14 | 192 | 5 |
| 8 | MBConv6, K3×3 | 7×7 | 320 | 1 |
| 9 | Conv1×1 & Pooling & FC | 7×7 | 1280 | 1 |

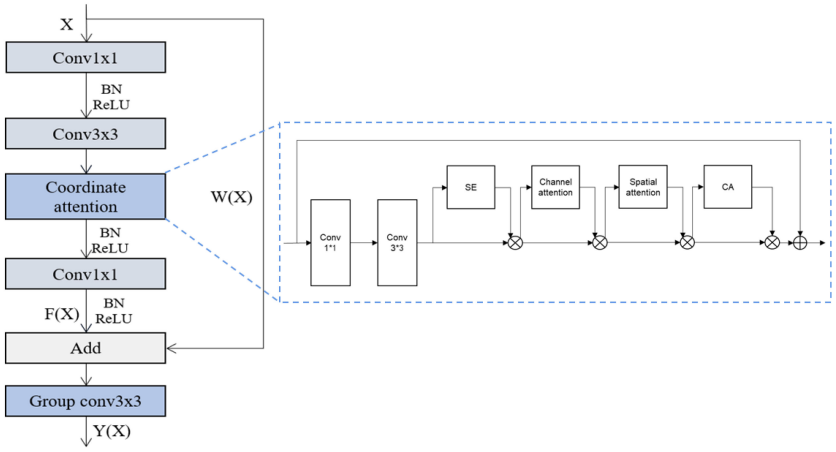



FIGURE 7
Structure of the Task-specific modules. Input comes from the feature vectors. The magnified part at the right detailed the structure of the attention mechanism module imbedded in the left structure. SE stands for the Squeeze-and-Excitation Networks, CBAM stands for the Convolutional Block Attention Module, and CA stands for the Collaborative attention mechanism.

in Table 7. To implement this in Keras, we defined a step decay function and used *LearningRateScheduler* callback to take the step decay function as the argument and return the updated learning rates for use in the SGD optimizer.

To make the training more efficient and achieve better performance of the CNN-based MummyBerryNet in the context of data limitation, the transfer learning technique was applied. Training was performed by adapting pre-trained networks on the ImageNet dataset. In this study, we only loaded the pre-trained weights of EfficientNet into the Parameter-sharing module of MummyBerryNet, while the weights of the Task-specific modules were initialized randomly. In particular, in order to apply the pre-trained weights to the Parameter-sharing module, we eliminated the weights of the ninth phase of the convolution and the fully connected layer.

TABLE 5 The architecture of the Deep MTL model: MummyBerryNet.

| Layer name | Output tensor | Configurations* |
|---|---|---|
| Input | 3×224×224 | Augmented images |
| Parameter-sharing module | 320×7×7 | EfficientNet |
| Task-specific module | 2048×5×5 | $\begin{bmatrix} 1\times1,\ 320 \\ 3\times3,\ 640 \\ 1\times1,1280 \\ 3\times3,2048,C=32 \end{bmatrix}\times3$ |
| FC1 | 6-classes | |
| FC2 | 3-classes | |
| FC3 | 5-classes | |

*'1×1'and '3×3'indicate the convolution operation with kernel equal 1 or 3; 'C' represents channel of convolution. 'FC' stands for Fully Connected Layer.

# 3 Experiments

## 3.1 Experimental setup

In order to evaluate the efficacy of our model in solving the three tasks of mummy berry disease detection, the performance of training and testing the CNN-based MummyBerryNet was first evaluated by widely accepted metrics (see below), then three model exploration experiments were conducted. They were: 1) the performance comparisons between the state-of-the-art deep learning models; 2) the ablation experiments for testing the effects of MTL scheme and transfer learning; and 3) the visualization for several mummy berry disease detection applications. Ablation experiments are used to study the performance of an AI (artificial intelligence) system by removing certain components, to understand the contribution of the component to the overall system.

The deep learning frameworks used for performance comparisons were AlexNet, VGG16, ResNet50, MobileNetV2, and EfficientNet. The selection of these six deep learning models was motivated by the fact that, except for EfficientNet, the other eight models (AlexNet, VGG16, ResNet50, MobileNetV2, EfficientNet-B0, EfficientNet-B1, EfficientNet-B2 and EfficientNet-B3) have established themselves as the most renowned and widely used CNNs for image classification tasks. These CNNs are widely used as benchmarks for evaluating deep learning models (Chen et al., 2023). MummyBerryNet was compared with the six state-of-the-art deep learning frameworks under the same experimental configuration conditions. In the experimental process, we found that the initial learning rate had a strong influence on the performance of all models. By conducting multiple experiments on a training set, we determined the optimal learning rates for six models ranged from 0.001 to 0.005. In contrast with the transfer learning process of MummyBerryNet, which was given previously, we first loaded all the pre-training weights for the six state-of-the-

TABLE 6 Configurations of the four instances of MummyBerryNet.

| Input layer / Model instances | Input | Parameter-sharing module | Task-specific Module[1] | FC1[2] | FC2 | FC3 |
|---|---|---|---|---|---|---|
| MummyBerryNet-B0 | 3×224×224 | EfficientNet-B0, Output: 320×7×7 (Table A1) | $\begin{bmatrix} 1\times1,\ 320 \\ 3\times3,\ 640 \\ 1\times1,1280 \\ 3\times3,2048, \\ C=32 \end{bmatrix}\times3$ Output: 2048×5×5 | 6 classes | 3 classes | 5 classes |
| MummyBerryNet-B1 | 3×240×240 | EfficientNet-B1, Output: 320×8×8 (Table A2) | $\begin{bmatrix} 1\times1,\ 320 \\ 3\times3,\ 640 \\ 1\times1,1280 \\ 3\times3,2048, \\ C=32 \end{bmatrix}\times3$ Output: 2048×6×6 | | | |
| MummyBerryNet-B2 | 3×260×260 | EfficientNet-B2, Output: 352×9×9. (Table A3) | $\begin{bmatrix} 1\times1,\ 320 \\ 3\times3,\ 640 \\ 1\times1,1280 \\ 3\times3,2048, \\ C=32 \end{bmatrix}\times3$ Output: 2048×7×7 | | | |
| MummyBerryNet-B3 | 3×300×300 | EfficientNet-B3, Output:384×10×10 (Table A4) | $\begin{bmatrix} 1\times1,\ 320 \\ 3\times3,\ 640 \\ 1\times1,1280 \\ 3\times3,2048, \\ C=32 \end{bmatrix}\times3$ Output: 2048×8×8 | | | |

[1]'1×1' and '3×3' indicate the convolution operation with kernel equal 1 or 3 and 'C' represent channel of convolution. 2: [2] 'FC' stands for Fully Connected Layer.

art models, and then modified the number of output features of the fully connected layer to train them.

The ablation experiment had two purposes: 1) evaluate the impact of transfer learning on model performance in contrast with the scenario where no transfer learning was applied; and 2) test the advantages of knowledge sharing for multiple-task learning. Therefore, we disintegrated the Deep MTL model into several STL modes, trained them and then compared them with the original MTL model. The experimental configuration of the two ablation experiments was basically the same as that in the comparison experiment.

All experiments were conducted based on the publicly available code of PyTorch (Machine learning open-source library) framework and a CPU/GPU platform which was built with a Xeon(R) 2.20 GHz (E5-2650 v4) CPU, 128 GB of memory and one Tesla P100-PCIE-12GB Graphics board.

## 3.2 Evaluation metrics

To estimate the effectiveness of MummyBerryNet in solving the MTL problem of mummy berry disease detection, the metric of accuracy was used. Specifically, the metric of accuracy is defined as the proportion of true results (including both the True Positives and the True Negatives) among the total number of samples examined (Equation 1):

$$\textbf{Accuracy} = \frac{\textbf{TP} + \textbf{TN}}{\textbf{TP} + \textbf{TN} + \textbf{FP} + \textbf{FN}} \quad (1)$$

where TP=True Positive, FP=False Positive, TN =True Negative, and FN=False Negative rates.

However, accuracy carries more weight on the True Positives and True Negatives than the False Positives and Negatives. This may bias perception of the disease detection results. Furthermore, accuracy has been found to be sensitive to imbalanced samples.

TABLE 7 Specification of the optimizer and parameters for training MummyBerryNet.

| Parameter | Setting |
|---|---|
| Optimizer | Stochastic Gradient Descent (SGD) |
| Loss function | Cross-Entropy |
| Learning rate* | 0.005 |
| Epochs | 50 |
| Batch size | 16 |
| Weight decay | 0.0001 |
| Momentum | 0.9 |

∗ Decreased by a factor of 1/10 at every 15 epochs.

To overcome this problem, we added the F1-score to balance the evaluation metrics, which gives more weight to False Negatives and False Positives, and also performs better when the sample classes are imbalanced (Qu and Liu, 2020). The F1-score is defined as (Equations 2–4):

$$\mathbf{F1-score} = \frac{2 * \mathbf{Precision} * \mathbf{Recall}}{\mathbf{Precision} + \mathbf{Recall}} \quad (2)$$

where,

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}} \quad (3)$$

$$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}} \quad (4)$$

The third metric used was certainty, which was used for mummy berry disease detection applications. For each run of the MummyBerryNet model, the output is a vector containing the likelihood that the detection result should be classified to a category. The category with the maximum likelihood or certainty value is then regarded as the prediction result of the model. We applied the Softmax function (Liu et al., 2016) to an $n$-dimensional vector of the model output and rescaled them so that the elements of the $n$-dimension were in the range [0,1] and summed to 1. The output of the Softmax function is defined as the detection certainty of an input image (Equation 5):

$$Softmax(X_i) = \frac{\mathbf{exp}(X_i)}{\sum_j \mathbf{exp}(X_j)} \quad (5)$$

where $X$ represents the n-dimensional vector outputted by MummyBerryNet, $X_i$ represents a component of $X$, $\sum_j \exp(X_j)$ represents the sum of $X$.

# 4 Results

## 4.1 Model training and validation

After inspection of the dynamics of cross entropy loss and accuracy in both the training and validation process of the four instances of MummyBerryNet, we found that the well trained MummyBerryNet is a good-fit model for mummy berry disease detection. Overall, the training loss decreased very quickly before epoch 10, then continuously but slowly went down and started to converge after epoch 30, suggesting an effective training process Figure 8. The differences of cross entropy loss between training and validation were less than 5%, which demonstrated that MummyBerryNet can learn the fundamental patterns of mummy berry disease in the training samples and had excellent generalizability across different datasets without overlapping. Our results also showed that MummyBerryNet's risk of overfitting is fairly low. As for accuracy, MummyBerryNet also quickly reached the best performance zone, higher than 95%, after epoch 15. The overall speed of convergence of the four instances of MummyBerryNet was better than our expectation, which can be explained by the contribution of transfer learning and weight sharing among tasks. The total number of parameters (Parameter-sharing plus Task-specific modules) of the four instances of MummyBerryNet ranged from 21.4M to 28.4M (see Table A5 in Appendix A). The four instances were only 30% of the size of the biggest NefficientNet version (B7).

All four instances of the MummyBerryNet model achieved high accuracy and F1-scores Table 8 in mummy berry disease detection. In the infection site identification task, the first instance, i.e., MummyBerryNet-B0 had the highest accuracy (96.81%) and F1-score (97.03%) and the lowest record was obtained by MummyBerryNet-B2 with an accuracy of 95.63% and F1-score of 95.50%. In the infection stage classification task, MummyBerryNet-B1 achieved the highest accuracy of 97.13% and the highest F1-score of 97.68%, while the lowest accuracy and F1-score were recorded by MummyBerryNet-B2. The second instance MummyBerryNet-B1 also achieved the best accuracy of 96.51% and best F1-score of 92.04% in the severity estimation task, which can be regarded as the best candidate of the four instances of our Deep MTL model, if we also take into consideration that it had the second smallest parameter size and the second lowest computational cost (see Appendix A, Table A5).

A Confusion matrix is an important statistical tool used for machine learning model analysis to evaluate the performance of a classification task. It represents the relationship between the predicted results and the true labels generated by the model. This matrix is a tabular representation that displays the count of accurate and inaccurate predictions made by the model through a comparison of predicted values with the actual values Figure 9. The elements on the diagonal represent the count of samples correctly predicted by the model, whereas the off-diagonal elements represent misclassifications. In multi-class classification tasks, the accuracy of the model for each category can be accurately calculated by analyzing the diagonal elements.

## 4.2 Comparisons

The systematic comparisons between the four instances of MummyBerryNet model and the eight state-of-the-art CNNs demonstrated that the MummyBerryNet model outperformed the eight CNNs (Figure 10; data can be found in the Appendix B, Table B2). In the site identification task, MummyBerryNet-B0, B1 and B3 achieved the highest accuracy and F1-score, while the AlexNet and MobileNetV2 were the lowest detectors, which were around 14% lower

10.3389/fpls.2024.1340884

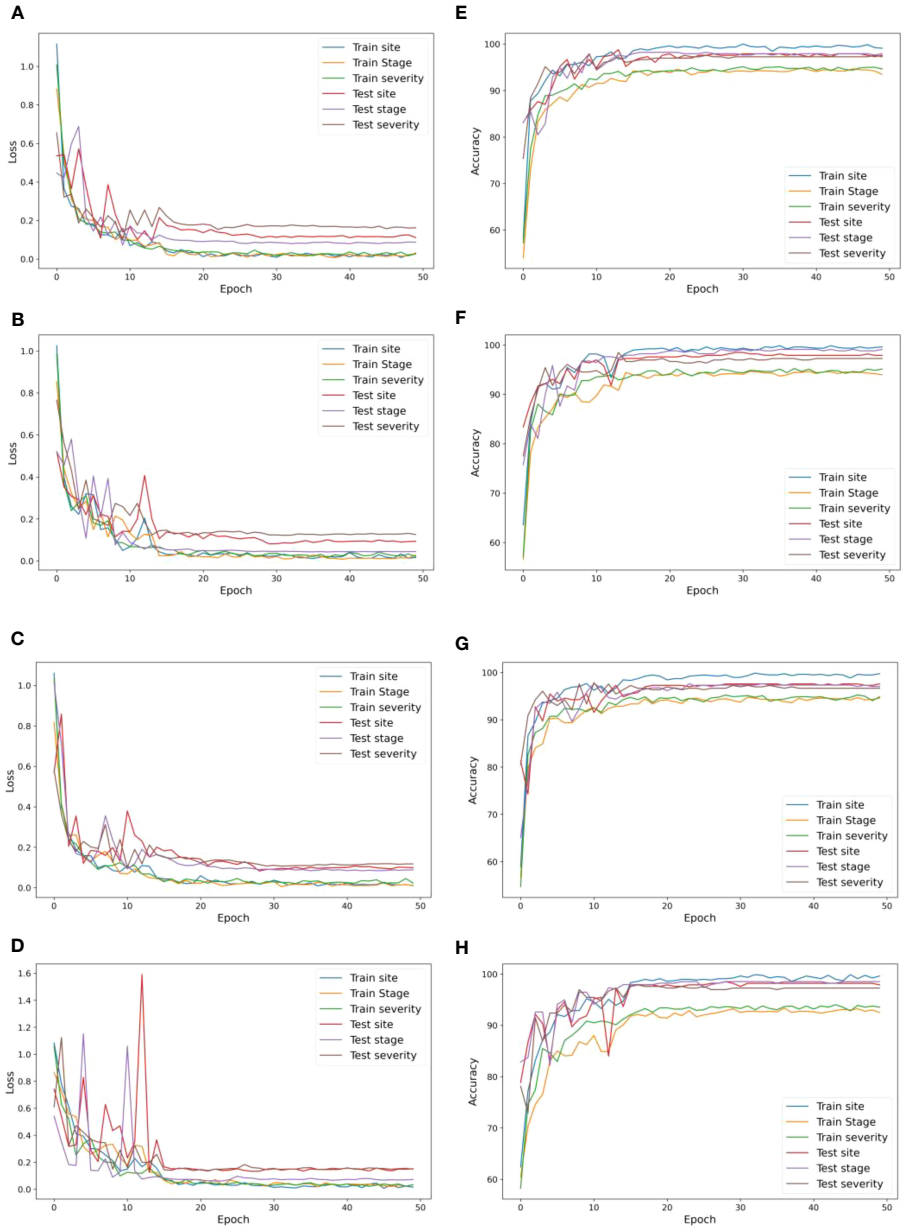



FIGURE 8
The classification cross entropy loss (left) and detection accuracy (right, in percent) during model training and validation. Panels **(A–D)** were: the model loss changing with training and testing iterations of the four instances MummyBerryNet-B0 to B3, respectively; and panels **(E–H)** were the detection accuracy changing with training and testing iterations of the four instances: MummyBerryNet-B0 to B3.

TABLE 8 Performance[1] of the four instances of MummyBerryNet on the validation dataset.

| Instance of the Model | Site identification | | Stage classification | | Severity estimation | |
|---|---|---|---|---|---|---|
| | Accuracy | F1-score | Accuracy | F1-score | Accuracy | F1-score |
| MummyBerryNet-B0 | **96.81** | **97.03** | 96.26 | 96.83 | 95.91 | 92.31 |
| MummyBerryNet-B1 | 96.52 | 96.56 | **97.13** | **97.68** | **96.51** | **92.04** |
| MummyBerryNet-B2 | 95.63 | 95.50 | 95.68 | 96.25 | 95.91 | 92.14 |
| MummyBerryNet-B3 | 96.52 | 96.72 | 96.55 | 97.04 | 95.91 | 91.12 |

[1]The best model performance is shown in bold.

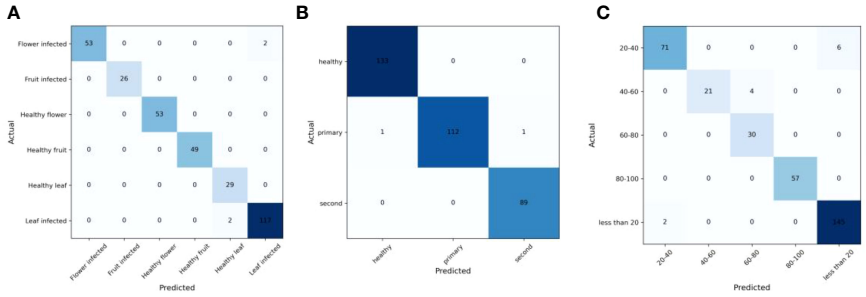



FIGURE 9
Confusion matrix for MummyBerryNet model for three tasks: the infection site identification task **(A)**; the infection stage classification task **(B)**; and the severity estimation task **(C)**.

than our model. Even the worst instance of our model (MummyBerryNet-B2) outperforms the other comparison models in terms of both accuracy and F1 scores. In the stage classification task, all four instances of MummyBerryNet were listed as the top-4 best competitors on both accuracy and F1-score. MummyBerryNet-B1, as the best detector, was 4.5% higher in accuracy and 5% higher in F1-score than EfficientNet-B2. In the severity estimation task MummyBerryNet-B1 overwhelmingly outperformed the other CNNs. Overall, MummyBerryNet had obvious advantages in site identification and stage classification compared to the eight CNNs, but the advantages declined around 1.5% in the severity estimation task. Due to the complex background and high mixture of plant parts, as well as noise and distortion, all methods faced a great challenge in estimating the proportion of diseased plant tissues perfectly.

## 4.3 Ablation experiments

The results of the ablation experiments demonstrated the effectiveness of applying MTL and transfer learning. In the first ablation experiment, two scenarios were compared to determine differences between MTL and STL learning in mummy berry disease detection (Figure 11A, data in Appendix B, Table B3). On average, MTL can increase detection accuracy by 3% compared to the STL learning scheme. However, the advantage gained by MTL was dominated by stage classification, which was 6% higher than the STL scheme and was twice as high as that in severity estimation. This result revealed that the advantage of using parameter or knowledge sharing between different tasks might be negatively correlated to the amount of information in the training samples because the disease severity needs much more information per unit to be accurately estimated than the other two categories, even when we had balanced the samples for the three tasks for training.

In the second ablation experiment, two scenarios in which model trainings with or without adopting transfer learning were compared. As shown in Figure 11B (data in Appendix B, Table B4), transfer learning can increase disease detection accuracy up to 30%, a large increase that once again demonstrated the effectiveness of transfer learning. Almost all of the four instances of MummyBerryNet gained the same level of enhancement by using transfer learning.

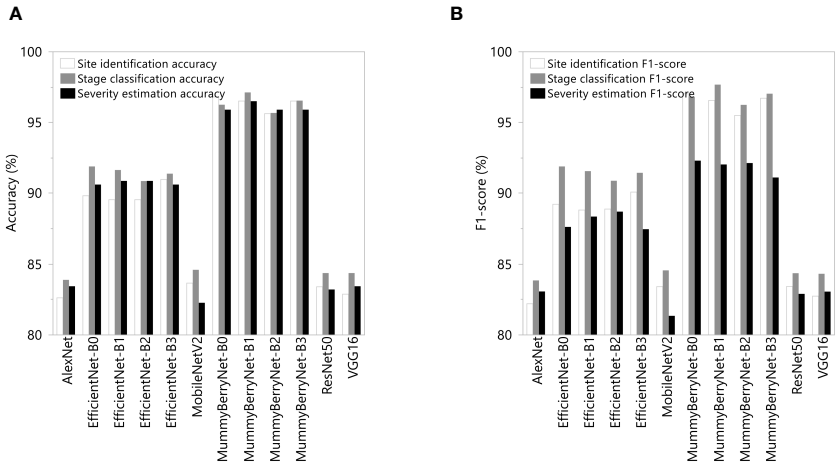



FIGURE 10
Comparations of detection accuracy **(A)** and F1-score **(B)** between the four instances of MummyBerryNet and the eight state-of-the-art CNNs in the three detection tasks: the infection site identification; the stage classification; and the severity estimation.

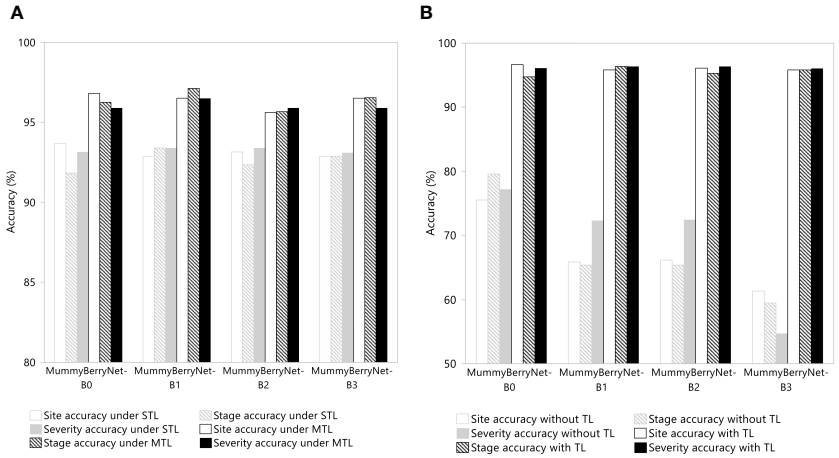



FIGURE 11
Efficacy of MTL **(A)** and transfer learning (TL) **(B)** in network performance enhancement in the three detection tasks: the infection site identification; the stage classification; and the severity estimation of the four instances of MummyBerryNet.

## 4.4 Visualization

### 4.4.1 Applications

MummyBerryNet-B1, having the optimal tradeoff between accuracy and computational cost among the four instances of MummyBerryNet, was selected as the best detector and applied to three mummy berry disease detection tasks. Figure 12 visualized the detection results for infection site identification (a), stage classification (b) and severity estimation (c), respectively. In each task, a randomly selected image was input into MummyBerryNet-B1. The detection tasks were then calculated, and the detection result was listed in tables. Each table had three columns indicating the ranking of detection probability of each predicted class and certainty. In the middle panel of Figure 12A, for example, both the likelihood of diseased leaf and flower in the image were detected with 95.55% and 4.50% certainty for leaf and flower, respectively. The detected classes were listed and sorted on descending certainty, while the classes with zero likelihood were removed from the table.

### 4.4.2 Feature map inside the deep structure

Deep feature visualization of CNNs can help us to understand the process of feature extraction from the millions of image patches. This is also useful to adjust the optimal hyperparameters during training. The visualization results of deep learning shallow convolutional neural network conform to the image information understandable by humans using visual perceptions. This helps us to intuitively observe and understand the focus area of model feature extraction in specific disease detection tasks. A feature visualization was constructed to observe the correspondence behavior of the first convolution layer of MummyBerryNet in Figure 13. The visualization demonstrated that feature maps obtained from the first convolution layer primarily focused on the color and contour extraction of blueberry flowers, leaves and fruits, which can clearly show that the lesion areas of mummy berry diseases received special attention and better explained what our model had actually learned. In order to present the transformation of feature extraction in MummyBerryNet, Figure 14 illustrates the comparisons between different feature maps obtained by the layers in the Parameter-sharing and Task-specific module. In the feature maps, each point was a rectified activation, meaning that the brighter it was, the greater activation value it represented. The difference between modules and layers clearly demonstrated that MummyBerryNet was characterized by an excellent learning process for fine grain aspects of blueberry plant tissue with disease.

# 5 Discussion

## 5.1 Detection of mummy berry disease under a complex background

This study proposed a deep learning-based computer vision solution for automatic mummy berry disease detection. To the best of our knowledge, this is the first method in the lowbush blueberry research community that can automatically and simultaneously identify infection site, classify infection stage, and estimate severity from field acquired lowbush blueberry plant images. The primary advantage of this machine learning approach is that a Convolutional Neural Network (CNN) was employed to detect various diseased plant parts interacting with a complex background, but no explicit feature engineering was involved (Liu et al., 2020). Although automatic plant disease detection using deep neural networks is no longer a “cutting-edge” technique, detecting small lesions mixed with different types of noise and coexisting plant parts varying in size, shape, angle, focal distance, and contrast is still a big challenge (Liu and Wang, 2021), it can be solved by the integrated attention mechanism. In general, a visual attention mechanism can enhance information extraction from a channel or spatial perspective, or both. A channel-based attention mechanism boosts specific feature layers (i.e., channels) possessing more interesting information and lessens others in the feature map, while the spatial-based attention mechanisms can focus on a

10.3389/fpls.2024.1340884

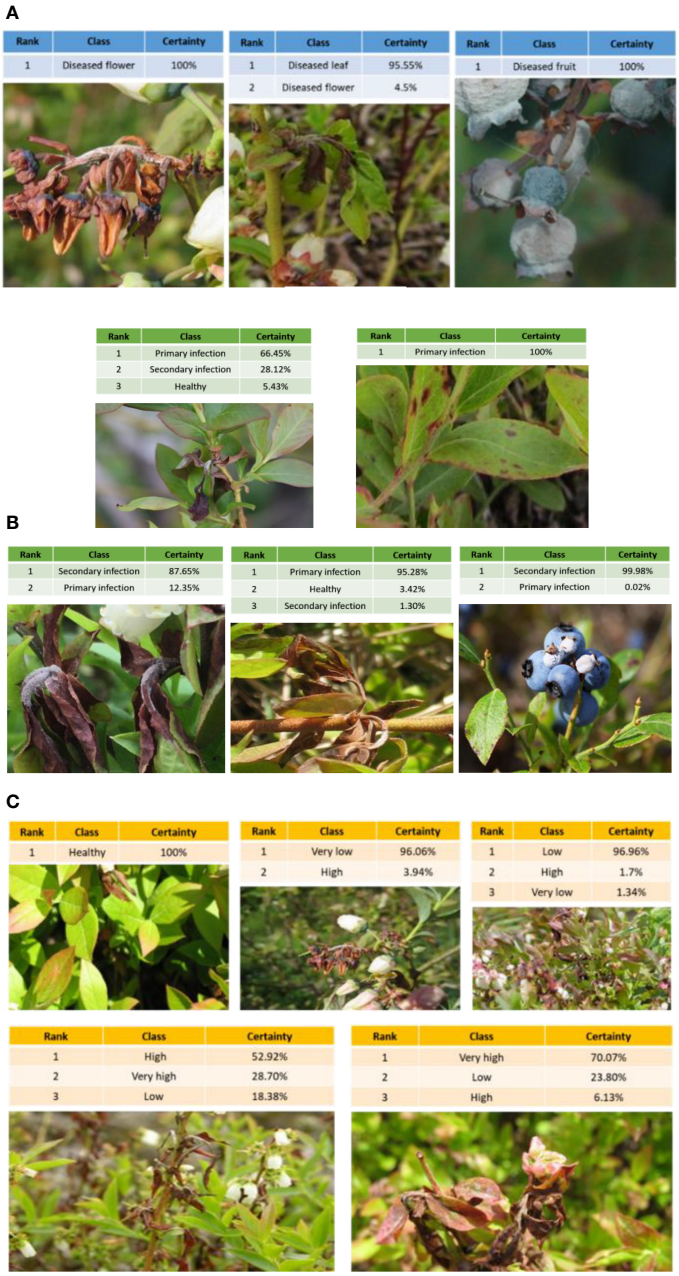



FIGURE 12
Applications of infection site identification **(A)**, stage classification **(B)** and severity estimation **(C)** using MummyBerryNet-B1, the best detector among the four instances of MummyBerryNet model. In A, the right picture in the second row is an example of a false positive identifying a picture of another leaf spot as primary infection by mummy berry.

specific region of the feature space and ignore the background. The different characteristics of attention mechanism varieties and their successful applications indicated that the combination of attention mechanisms in deep neural networks is effective to solve this problem. Focusing on features of interest across various spatial scale and abstract levels is vital for extracting disease features from a complex background with likely occlusions, which has been proven by our experimental results.

In addition, we used a unique deep structure design with the help of visualization to understand how the focus area of model feature extraction operates for specific disease detection tasks (Figures C1, C2). This approach is particularly effective when dealing with small and low contrast targets. This is because visualization can help to identify different traits such as color, shape and contour in different layers (Nagasubramanian et al., 2019) (Figure 14). Furthermore, our four-instance model exploration method also greatly helped to refine model structure for mummy berry disease detection (Picon et al., 2019). In comparison with a similar technical approach in which a depth-separable CNN was employed instead of a standard convolution to detect grape leaf disease (Liu et al., 2020), our method performed better in detection accuracy even if our counterpart already outperformed the standard ResNet and GoogleNet structure (Thangaraj et al., 2021).

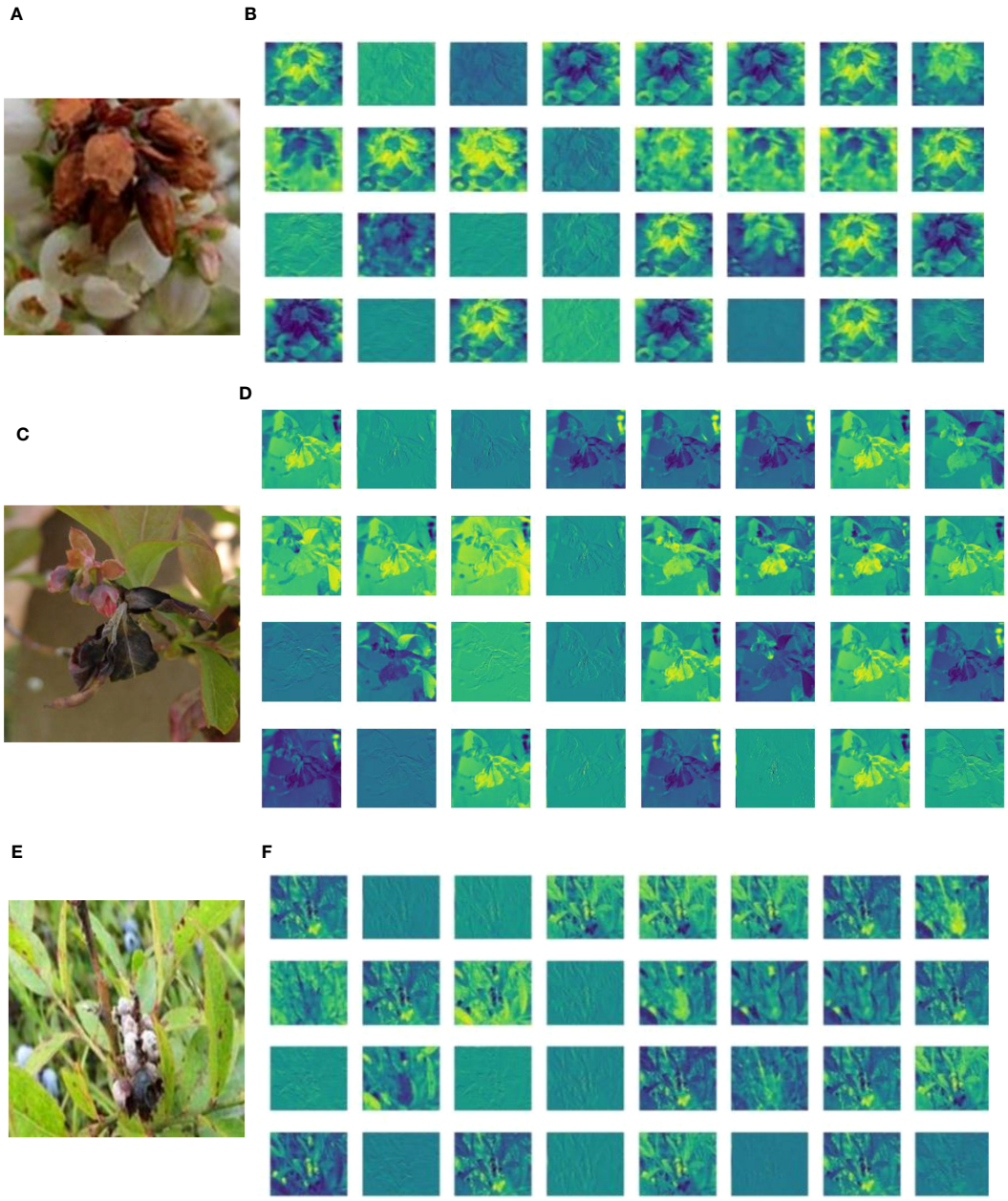



FIGURE 13
Visualization of CNNs (MummyBerryNet) in the initial layers, which consist of a 3x3 convolution and batch normalization. Panels **(A, C, E)** represent the original images (size 224× 224) of blueberry flowers, leaves, and fruits, respectively; panels **(B, D, F)** were the feature maps of flowers, leaves, and fruits with training (size 112 × 112).

When it comes to plant disease detection in a complex background, our method has achieved superior detection accuracy in contrast with the state-of-the-art CNN-based counterparts (Xie et al., 2020; Chen et al., 2022; Liu and Zhang, 2022). The most important explanation is the application of grouped convolutions. Taking one modular block of a filter group and replicating it allowed us to build wider networks so that the learned features can be largely diversified. With the help of grouped convolution, our method is able to learn a varied set of low-level and high-level features, which is vital to more accurately detect disease in real farming conditions. In addition, the data augmentation employed in this work is able to capture and add some extra hidden features in the original data set, which is particularly useful in the context of a complex background where multi-level features are usually needed (Barbedo, 2018). Our method has shown better practicability when compared with recent advances in deep learning-based plant disease recognition solutions, such as the plant disease classification systems developed by Ferentinos (2018) and Mohanty et al. (2016) where lesions were finely focused and the target plant organs were imaged in a laboratory model with high resolution, simple uniform background and no overlap among plant organs and other tissues. In the real farming environment, lighting issues and occlusion problems are common

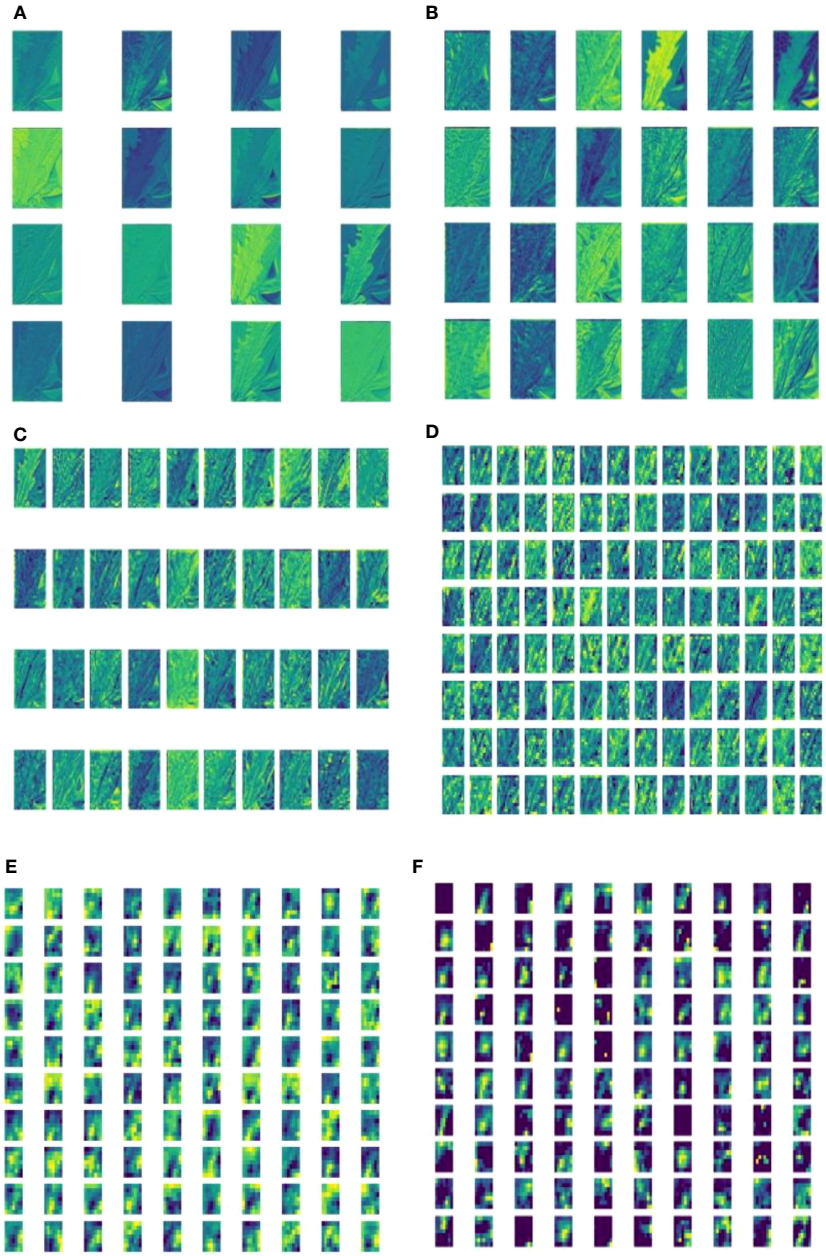



FIGURE 14
Visualization of feature maps inside MummyBerryNet. Panel images **(A-E)** were obtained by the Parameter-sharing module (layer), Panel **(F)** images were obtained by the Task-specific module (layer).

challenges that have to be considered. In comparison with current advancements (Martinelli et al., 2015; Liu and Wang, 2020a; b), our deep MTL method with shared feature capturing has shown an alternative way to overcome complex and chaotic background issues.

## 5.2 Multi-task learning with lower computational cost and higher training efficiency

The second advantage or enhancement in our model is that the MTL scheme combined with transfer learning has shown a promising solution for image-based agriculture disease detection with lower computational cost and higher training efficiency. This advantage came from the parameter sharing mechanism. Our unique model structure, one Parameter-sharing module plus three Task-specific modules in MummyBerryNet, not only can take advantage of correlated knowledge representations among tasks to effectively decrease the risk of overfitting (Caruana, 1997), but also can train multiple tasks all together to implicitly decrease the required training samples as well as training time. The hypothesis that many features are shared between images in the infection site, stage and severity datasets therefore has successfully been tested. One example is that the

pixels of mummies can be used to identify both the infection site and the secondary infection stage.

The most successful instance of the MummyBerryNet model, i.e. MummyBerryNet-B1, achieved the highest accuracy and the highest F1-score in both stage classification and severity estimation, but only had one-sixth of the network parameter size (23.98M vs. 138.36M) and one-fifteenth of the computational cost (1.13G vs. 15.48G of FLOPs, see Appendix A, Table A5) compared with the most popular CNN: VGG16. This advantage can be even larger if transfer learning is applied. One surprising finding in our ablation experiments was that transfer learning can significantly boost the accuracy of severity estimates (up to 30 percent on average). Since the estimation of disease severity requires much more information than classifying diseased plant parts and stages, such as differences in size, density, color and scale to make a reasonable estimation, this feature also made our solution very promising for future mobile deployment such as a drone carried task unit for real-time field surveillance.

The grouped convolutions imbedded into the MTL paradigm is also an important reason for performance superiority. In this approach, each filter convolves only on some of the feature maps obtained from kernel filters in its filter group, resulting in less redundant convolutions. It allows us to drastically lower the computations to get output feature maps. It also enables efficient data and model parallelism, which obviously benefits faster convergence, compared to the methods proposed by Chen et al. (2022) and Liu and Zhang (2022).

## 5.3 Limitations and future research direction

One of the limitations of MummyBerryNet is that we found all of its four instances had difficulty in detection of sporulation on shoots. One example was that in the mummy berry disease detection application (the first image in the B section of Figure 12), sporulating tissues had a higher probability of misclassification than other categories, e.g., the certainty of secondary infection was only 87.65%. This was due to the very limited samples of sporulation, which cannot train the deep structure well enough for the generality necessary for the task of accurate detection. This made the detection probability even lower when facing the small size and low contrast feature of sporulating tissues. We may need more relevant training data and particular attention to the mechanism (Fukui et al., 2019) of deep learning in the future version of MummyBerryNet to improve the detection accuracy. The second limitation was difficulty in the estimation of infection severity. We found that the accuracy of severity estimation was always the lowest among the three tasks (Figures 10, 11). It is no doubt that sophisticated image segmentation with a complex background decreased the accuracy of severity estimation. However, we still believe that the primary reason was the manual labeling of severity levels as the function of infected area. Manually labeling severity levels doesn't provide enough information connecting depth and scale of overlapped organs to a severity level. In future research, multi-view approaches (Zhou et al., 2021) integrating scale-aware information (Zhang et al., 2017) could be a feasible way to tackle this issue.

Aside from the above objective limitations, the technical limitation is about how multiple attention mechanisms could be effectively applied and collaborated with CNN. Although our experimental results suggested that the use of channel and spatial AMs are effective there is a need for further tuning to better facilitate real disease detection applications. In our current model, superimposing three different attention mechanisms in the residual blocks, may not be the best solution in terms of accuracy and performance. The superimposed attention mechanisms may be redundant in the feature extraction process by repeatedly extracting image features on different channels and spaces, leading to unexplainable results. However, due to the limitation of interpretability of deep learning, it is still unclear how these channel or spatial AMs should be combined and how the combination of AMs should be incorporated into the deep neural networks to effectively detect sporulation on shoots. The type of AMs, the number of AMs and their positions in the deep neural network need to be systematically investigated in future research. Due to the limited resources, this research was not able to conduct this investigation. But it is important to examine this uncertainty of AM corporation in tiny feature detection in an open and complex environment.

# 6 Conclusion

To solve the practical problem of field disease surveillance, this research proposed an innovative multi-task learning mode for mummy berry disease diagnosis. The model integrated the techniques, such as residual learning, coordinate attention mechanism, and group convolution, into a deep multi-task learning (MTL) approach to simultaneously accomplish three disease detection tasks of identifying infection sites, classifying infection stages, and estimating disease severity. This deep MTL model achieved higher detection efficiency compared with the state-of-the-art deep learning models in terms of detection accuracy and performance, due to the three key contributions: 1) the MTL approach can simultaneously accomplish three mummy berry disease detection tasks with limited data; 2) A novel superimposed attention mechanism modules applied to deep learning can enhance disease feature extraction from both channel and spatial perspective, enabling better performance in open and complex environment compared to other CNNs; and 3) Integrating grouped convolution to MTL enables to learn a varied set of low-level and high-level disease features in a more parallelism manner, resulting a significant lower computational complexity and faster convergence. These features make our solution very promising for future mobile deployment such as a drone carried task unit for real-time field surveillance. As an automatic approach to fast disease diagnosis, it can be a useful technical tool to provide growers real time disease information that can prevent further disease transmission and more severe effects on yield due to fruit mummification.

## Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

## Author contributions

HQ: Conceptualization, Funding acquisition, Investigation, Methodology, Project administration, Writing – original draft. CZ: Data curation, Investigation, Software, Validation, Writing – original draft. HJ: Data curation, Investigation, Software, Validation, Visualization, Writing – original draft. RH: Data curation, Formal Analysis, Investigation, Methodology, Visualization, Writing – original draft. DW: Conceptualization, Funding acquisition, Project administration, Resources, Writing – review & editing. SA: Conceptualization, Data curation, Resources, Supervision, Validation, Writing – review & editing. FD: Formal Analysis, Investigation, Resources, Supervision, Validation, Writing – review & editing.

## Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. Funding was received from the National Key Research and Development Program of China (2022YFF1300503-03) and the National Natural Science Foundation of China (61871061), which is gracefully acknowledged.

## Acknowledgments

We are grateful to several graduate students in the School of Biology and Ecology of the University of Maine for their efforts of taking field images. We are also appreciative for the photographers Caleb Slemmons, Gerald Holmes, etc., who shared their blueberry images on the National Ecological Observatory Network (Bugwood.org).

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2024.1340884/full#supplementary-material

## References

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., et al. (2019). A state-of-the-art survey on deep learning theory and architectures. *Electronics* 8, 292. doi: 10.3390/electronics8030292

Alruily, M. (2021). Classification of arabic tweets: A review. *Electronics* 10, 1143. doi: 10.3390/electronics10101143

Barbedo, J. G. A. (2018). Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Comput. Electron. Agric.* 153, 46–53. doi: 10.1016/j.compag.2018.08.013

Batra, L. R. (1983). Monilinia vaccinii-corymbosi (Sclerotiniaceae): Its biology on blueberry and comparison with related species. *Mycologia* 75, 131–152. doi: 10.1080/00275514.1983.12021642

Batra, L. R., and Batra, S. W. T. (1985). Floral mimicry induced by mummy-berry fungus exploits host's pollinators as vectors. *Science* 228, 1011–1013. doi: 10.1126/science.228.4702.1011

Bock, C. H., Poole, G. H., Parker, P. E., and Gottwald, T. R. (2010). Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging. *Crit. Rev. Plant Sci.* 29, 59–107. doi: 10.1080/07352681003617285

Caruana, R. (1993). Multitask learning: A knowledge-based source of inductive bias1. In: *Proceedings of the Tenth International Conference on Machine Learning*; 1993 Jun 27-29; Citeseer, Amherst, MA, USA, p. 41–48. doi: 10.1016/b978-1-55860-307-3.50012-5

Caruana, R. (1997). Multitask learning. *Mach. Learn.* 28, 41–75. doi: 10.1023/A:1007379606734

Chen, R., Qi, H., Liang, Y., and Yang, M. (2022). Identification of plant leaf diseases by deep learning based on channel attention and channel pruning. *Front. Plant Sci.* 13, 1023515. doi: 10.3389/fpls.2022.1023515

Chen, Y., Huang, Y., Zhang, Z., Wang, Z., Liu, B., Liu, C., et al. (2023). Plant image recognition with deep learning: A review. *Comput. Electron. Agric.* 212, 108072. doi: 10.1016/j.compag.2023.108072

Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* 145, 311–318. doi: 10.1016/j.compag.2018.01.009

Fuentes, A., Yoon, S., Kim, S. C., and Park, D. S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* 17, 2022. doi: 10.3390/s17092022

Fukui, H., Hirakawa, T., Yamashita, T., and Fujiyoshi, H. (2019). Attention branch network: Learning of attention mechanism for visual explanation. In *VPR 2019: Proceedings of the IEEECVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*; 2019 Jun 15-20; Long Beach, California, USA, p. 10705–10714. doi: 10.1109/cvpr.2019.01096

Geetharamani, G., and Arun Pandian, J. (2019). Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Comput. Electr. Eng.* 76, 323–338. doi: 10.1016/j.compeleceng.2019.04.011

Hanes, S. P., Collum, K. K., Hoshide, A. K., and Asare, E. (2015). Grower perceptions of native pollinators and pollination strategies in the lowbush blueberry industry. *Renew. Agric. Food Syst.* 30, 124–131. doi: 10.1017/S1742170513000331

Haralick, R. M., and Shapiro, L. G. (1985). Image segmentation techniques. *Comput. Vis. Graph. Image. Process.* 29, 100–132. doi: 10.1016/S0734-189X(85)90153-7

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *CVPR 2016: Proceedings of IEEE Conf. Comput. Vis. Pattern Recognit.*; 2016 Jun26 -Jul01; Las Vegas, Nevada, USA. p. 770–778. doi: 10.1109/cvpr.2016.90

Hou, Q., Zhou, D., and Feng, J. (2021). Coordinate attention for efficient mobile network design. In *CVPR 2021: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021 Jun 20-25; Nashville, TN, USA; p. 13713–13722. doi: 10.48550/arXiv.2103.02907

Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. In *CVPR 2018: Proceedings of IEEECVF Conf. Comput. Vis. Pattern Recognit.*. 2018 Jun 18-22; Salt Lake City, USA; p. 7132–7141. doi: 10.1109/cvpr.2018.00745

Liang, Q., Xiang, S., Hu, Y., Coppola, G., Zhang, D., and Sun, W. (2019). PD2SE-Net: Computer-assisted plant disease diagnosis and severity estimation network. *Comput. Electron. Agric.* 157, 518–529. doi: 10.1016/j.compag.2019.01.034

Liu, B., Ding, Z., Tian, L., He, D., Li, S., and Wang, H. (2020). Grape leaf disease identification using improved deep convolutional neural networks. *Front. Plant Sci.* 11. doi: 10.3389/fpls.2020.01082

Liu, J., and Wang, X. (2020a). Early recognition of tomato gray leaf spot disease based on MobileNetv2-YOLOv3 model. *Plant Methods* 16, 83. doi: 10.1186/s13007-020-00624-2

Liu, J., and Wang, X. (2020b). Tomato diseases and pests detection based on improved yolo V3 convolutional neural network. *Front. Plant Sci.* 11. doi: 10.3389/fpls.2020.00898

Liu, J., and Wang, X. (2021). Plant diseases and pests detection based on deep learning: a review. *Plant Methods* 17, 22. doi: 10.1186/s13007-021-00722-9

Liu, K., and Zhang, X. (2022). PiTLiD: identification of plant disease from leaf images based on convolutional neural network. *IEEE/ACM Trans. Comput. Biol. Bioinf.* 20, 1278–1288.

Liu, W., Wen, Y., Yu, Z., and Yang, M. (2016). Large-margin softmax loss for convolutional neural networks. In *ICML 2016: Proceedings of international Conference on Machine Learning*; 2016 Jun 19-24; New York USA; arXiv:1612.02295. doi: 10.48550/arXiv.1612.02295

Lu, Y., Yi, S., Zeng, N., Liu, Y., and Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. *Neurocomputing* 267, 378–384. doi: 10.1016/j.neucom.2017.06.023

Martinelli, F., Scalenghe, R., Davino, S., Panno, S., Scuderi, G., Ruisi, P., et al. (2015). Advanced methods of plant disease detection. A review. *Agron. Sustain. Dev.* 35, 1–25. doi: 10.1007/s13593-014-0246-1

McGovern, K. B., Annis, S. L., and Yarborough, D. E. (2012). Efficacy of organically acceptable materials for control of mummy berry disease on lowbush blueberries in maine. *Int. J. Fruit Sci.* 12, 188–204. doi: 10.1080/15538362.2011.619350

Mnih, V., Heess, N., and Graves, A. (2014). Recurrent models of visual attention. In *NIPS 2014: Proceedings of The Twenty-eighth Annual Conference on Advances in Neural Information Processing Systems 27*; 2014 Dec 8-13; Montreal, Quebec, Canada; p.2204-2212. doi: 10.48550/arXiv.1406.6247

Mohanty, S. P., Hughes, D. P., and Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Front. Plant Sci.* 7. doi: 10.3389/fpls.2016.01419

Nagasubramanian, K., Jones, S., Singh, A. K., Sarkar, S., Singh, A., and Ganapathysubramanian, B. (2019). Plant disease identification using explainable 3D deep learning on hyperspectral images. *Plant Methods* 15, 98. doi: 10.1186/s13007-019-0479-8

nass.usda.gov. Available online at: https://www.nass.usda.gov/ (Accessed 15 November 2023).

Obsie, E. Y., Qu, H., and Drummond, F. (2020). Wild blueberry yield prediction using a combination of computer simulation and machine learning algorithms. *Comput. Electron. Agric.* 178, 105778. doi: 10.1016/j.compag.2020.105778

Penman, L. N., and Annis, S. L. (2005). Leaf and flower blight caused by monilinia vaccinii-corymbosi on lowbush blueberry: effects on yield and relationship to bud phenology. *Phytopathology®* 95, 1174–1182. doi: 10.1094/PHYTO-95-1174

Perez, L., and Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv*. doi: 10.48550/arxiv.1712.04621

Picon, A., Seitz, M., Alvarez-Gila, A., Mohnke, P., Ortiz-Barredo, A., and Echazarra, J. (2019). Crop conditional Convolutional Neural Networks for massive multi-crop plant disease classification over cell phone acquired images taken on real field conditions. *Comput. Electron. Agric.* 167, 105093. doi: 10.1016/j.compag.2019.105093

Qu, H., and Drummond, F. (2018). Simulation-based modeling of wild blueberry pollination. *Comput. Electron. Agric.* 144, 94–101. doi: 10.1016/j.compag.2017.11.003

Qu, H., and Liu, G. (2020). Threshold optimized strategy based on improved flower pollination algorithm for unbalanced data. *2020. IEEE 10th. Int. Conf. Intell. Syst.* 00, 551–556. doi: 10.1109/is48319.2020.9199930

Ramesh, S., Hebbar, R., Niveditha., M., Pooja., R., Prasad Bhat., N., Shashank, N., et al. (2018). Plant disease detection using machine learning. In *ICDI3C 2018: Proceedings of Int. Conf. Des. Innov. 3Cs Comput. Commun. Control*; 2018 Apr 25-28; Bangalore, India p. 41–45. doi: 10.1109/ICDI3C45473.2018

Ruder, S. (2017). An Overview of Multi-Task Learning in Deep Neural Networks. *arXiv*. doi: 10.48550/arxiv.1706.05098

Rumpf, T., Mahlein, A.-K., Steiner, U., Oerke, E.-C., Dehne, H.-W., and Plümer, L. (2010). Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance. *Comput. Electron. Agric.* 74, 91–99. doi: 10.1016/j.compag.2010.06.009

Sastry, K. S., and Zitter, T. A. (2014). Plant virus and viroid diseases in the tropics. *Epidemiol. Management.* 2, 149–480. doi: 10.1007/978-94-007-7820-7_2

Shorten, C., and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *J. big. Data* 6, 1–48. doi: 10.1186/s40537-019-0197-0

Su, Z., Fang, L., Kang, W., Hu, D., Pietikäinen, M., and Liu, L. (2020) Computer vision-ECCV 2020 in *ECCV 2020: Proceedings of 16th European Conference on Computer vision*, Glasgow, UK, August 23–28, 2020. 138–155, Proceedings, Part VI. doi: 10.1007/978-3-030-58539-6_9

Tan, M., and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*. 2019 Jun 9-15; Long Beach USA, 6105–6114. doi: 10.48550/arxiv.1905.11946

Taylor, L., and Nitschke, G. (2018). Improving deep learning with generic data augmentation. In *2018 IEEE symposium series on computational intelligence (SSCI)*. 1542–1547 (IEEE).

Thangaraj, R., Anandamurugan, S., and Kaliappan, V. K. (2021). Automated tomato leaf disease classification using transfer learning-based deep convolution neural network. *J. Plant Dis. Prot.* 128, 73–86. doi: 10.1007/s41348-020-00403-0

Vishnoi, V. K., Kumar, K., and Kumar, B. (2021). Plant disease detection using computational intelligence and image processing. *J. Plant Dis. Prot.* 128, 19–53. doi: 10.1007/s41348-020-00368-0

Waghmare, H., Kokare, R., and Dandawate, Y. (2016). Detection and classification of diseases of grape plant using opposite color local binary pattern feature and machine learning for automated decision support system. In *SPIN 2016: Proceedings of the 3rd International Conference on Signal Processing and Integrated Networks*. 2016 Feb 11-12; Noida, India. 513–518. doi: 10.1109/spin.2016.7566749

Woo, S., Park, J., Lee, J. Y., and Kweon, I. S. (2018). Cbam: Convolutional block attention module. In. In *ECCV 2018: Proceedings of the 15th European conference on computer vision*. 2018 Sep 8-14; Munich, Germany; p. 3–19. doi: 10.1007/978-3-030-01234-2_1

Xie, X., Ma, Y., Liu, B., He, J., Li, S., and Wang, H. (2020). A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks. *Front. Plant Sci.* 11. doi: 10.3389/fpls.2020.00751

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., et al. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *ICML 2015: Proceedings of the 32nd International Conference on Machine Learning*. 2015 Jul 6-11; Lille, France; p. 2048–2057. doi: 10.48550/arxiv.1502.03044

Zhang, Y., and Yang, Q. (2017). A survey on multi-task learning. *IEEE Trans. Knowl. Data Eng.* 34, 5586-5609. doi: 10.1109/TKDE.2021.3070203

Zhang, M., Yang, Y., Shen, F., Zhang, H., and Wang, Y. (2017). Multi-view feature selection and classification for Alzheimer's Disease diagnosis. *Multimedia. Tools Appl.* 76, 10761–10775. doi: 10.1007/s11042-015-3173-5

Zhou, J., Zhang, Q., and Zhang, B. (2021). An automatic multi-view disease detection system *via* Collective Deep Region-based Feature Representation. *Futur. Gener. Comput. Syst.* 115, 59–75. doi: 10.1016/j.future.2020.08.038

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 28 March 2024
DOI 10.3389/fpls.2024.1333236

Check for updates

OPEN ACCESS

EDITED BY
Tonghai Liu,
Tianjin Agricultural University, China

REVIEWED BY
Ayodeji Olalekan Salau,
Afe Babalola University, Nigeria
Parvathaneni Naga Srinivasu,
Prasad V. Potluri Siddhartha Institute of Technology, India

*CORRESPONDENCE
Zhenping Qiang
qzp@swfu.edu.cn
Giovanni Pau
giovanni.pau@unibo.it

RECEIVED 04 November 2023
ACCEPTED 26 February 2024
PUBLISHED 28 March 2024

CITATION
Lin H, Qiang Z, Tse R, Tang S-K and Pau G (2024) A few-shot learning method for tobacco abnormality identification. *Front. Plant Sci.* 15:1333236. doi: 10.3389/fpls.2024.1333236

COPYRIGHT
© 2024 Lin, Qiang, Tse, Tang and Pau. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# A few-shot learning method for tobacco abnormality identification

Hong Lin[1,2], Zhenping Qiang[1]*, Rita Tse[2], Su-Kit Tang[2] and Giovanni Pau[3,4]*

[1]College of Big Data and Intelligent Engineering, Southwest Forestry University, Kunming, China, [2]Faculty of Applied Sciences, Macao Polytechnic University, Macao, Macao SAR, China, [3]Department of Computer Science and Engineering, University of Bologna, Bologna, Italy, [4]Samueli Computer Science Department, University of California, Los Angeles, Los Angeles, CA, United States

Tobacco is a valuable crop, but its disease identification is rarely involved in existing works. In this work, we use few-shot learning (FSL) to identify abnormalities in tobacco. FSL is a solution for the data deficiency that has been an obstacle to using deep learning. However, weak feature representation caused by limited data is still a challenging issue in FSL. The weak feature representation leads to weak generalization and troubles in cross-domain. In this work, we propose a feature representation enhancement network (FREN) that enhances the feature representation through instance embedding and task adaptation. For instance embedding, global max pooling, and global average pooling are used together for adding more features, and Gaussian-like calibration is used for normalizing the feature distribution. For task adaptation, self-attention is adopted for task contextualization. Given the absence of publicly available data on tobacco, we created a tobacco leaf abnormality dataset (TLA), which includes 16 categories, two settings, and 1,430 images in total. In experiments, we use PlantVillage, which is the benchmark dataset for plant disease identification, to validate the superiority of FREN first. Subsequently, we use the proposed method and TLA to analyze and discuss the abnormality identification of tobacco. For the multi-symptom diseases that always have low accuracy, we propose a solution by dividing the samples into subcategories created by symptom. For the 10 categories of tomato in PlantVillage, the accuracy achieves 66.04% in 5-way, 1-shot tasks. For the two settings of the tobacco leaf abnormality dataset, the accuracies were achieved at 45.5% and 56.5%. By using the multisymptom solution, the best accuracy can be lifted to 60.7% in 16-way, 1-shot tasks and achieved at 81.8% in 16-way, 10-shot tasks. The results show that our method improves the performance greatly by enhancing feature representation, especially for tasks that contain categories with high similarity. The desensitization of data when crossing domains also validates that the FREN has a strong generalization ability.

KEYWORDS

**tobacco disease identification, few-shot learning, feature representation, instance-embedding, task-adaptation, cross-domain**

# 1 Introduction

Tobacco is a valuable crop that has a significant economic impact in many countries, such as China, India, and the USA, where it serves as an important tax resource for government revenue. Diseases and pests always lead to the degradation of the quality and yield (Strange and Scott, 2005). For tobacco plants, diseases or pests always cause serious damage to tobacco leaves, which are the main harvest of tobacco plants. Even if the leaves are not destroyed, the quality will be greatly reduced by the infection. Due to the high incidence of diseases, disease control in tobacco cultivation is heavily dependent on pesticides that threaten the safety of humans, animals, soil, and the environment (World Health Organization, 2017; Kahl et al., 2018). The traditional diagnosing methods rely on biochemical experiments or experts that are expensive and untimely. Farmer experiences sometimes are inaccurate. Therefore, an autodiagnosing system that can provide fast and easily accessible services for farmers is required in the agriculture industry (Kamilaris and Prenafeta-Boldú, 2018).

With the boom in deep learning, image-based recognition methods have been greatly improved. Computer vision-based deep learning methods rely on large-scale data to achieve good performance. For plant diseases, the collection of data is not only time-consuming but also requires the involvement of experts. Date deficiency has been the barrier to taking advantage of deep learning methods for plant disease identification (Kamilaris and Prenafeta-Boldú, 2018). Few-shot learning (FSL) has been proposed in recent years to target the problem of data shortages. By mimicking human perception, FSL methods do not need large amounts of data to learn new concepts.

The metric-based methods, as a mainstream method of FSL, are widely studied due to their intuitive underlying theory and good performance (Li et al., 2021). The query sample should be classified into the category that is the nearest one in support categories (Wang et al., 2020). Many classical methods in this branch have been proposed from different perspectives: feature extraction (Salau and Jain, 2019), distance metrics, etc., such as the siamese network (Koch et al., 2015), matching network (Vinyals et al., 2016), prototypical network (Snell et al., 2017), relation network (Sung et al., 2018), CoveMNet (Li et al., 2019), meta-baseline method (Chen et al., 2021), etc. Due to the fact that many scenarios cannot meet the high requirement of data, many related types of research, such as meta-transfer learning (Sun et al., 2019), cross-modal zero-shot hashing (Song et al., 2022), etc., were conducted actively to improve the identification performance.

Because FSL meets the small data scenarios, it raises high concerns in studies of plant disease identification (Yang et al., 2022). The siamese network, triplet network, baseline, baseline++, DAML, matching network, FEAT (Ye et al., 2020), etc., were used for plant disease identification (Argüeso et al., 2020; Jadon, 2020; Zhong et al., 2020; Afifi et al., 2021; Li and Chao, 2021; Li and Yang, 2021; Nuthalapati and Tunga, 2021). The most commonly used dataset is PlantVillage (PV). Moreover, citrus, bananas, coffee, rice, cucumber, etc., were studied. These methods were tried from various perspectives and made important progress.

Although FSL has the advantage in scenarios of data deficiency, it still has some challenging issues that need to be addressed. We argue that the weak feature representation is the most fundamental issue that needs to be addressed urgently. As is well known, traditional deep learning methods rely on large-scale data in training to obtain rich feature representation. While in FSL, the network has never been trained by the target categories. Hence, feature extraction ability is weak. The weak feature representation directly leads to low accuracy. In FSL, the basic idea is that the model can generalize previous knowledge to new concepts. Hence, the weak feature representation naturally causes troubles in the generalization of the model. Generalization refers to the ability of a trained model to perform well on unseen or previously unseen data in deep learning. It is a crucial purpose of training deep learning models as it determines if they can make accurate predictions beyond the training data. In the problem definition of FSL, generalization has a higher requirement since it requires generalizing to unseen categories in training. When the unseen categories are from another domain, it is called cross-domain, which demands stronger generalization capabilities of the model. Cross-domain generalization is a more challenging issue in FSL, while it is common in applications. Taking tobacco disease identification as an example, suppose that all categories of tobacco diseases are required to be identified and only a few images are available for each category, which means that these data cannot be used in training according to the definition of FSL. Here, cross-domain, which uses data from domain A for training and data from domain B for testing, is inevitable.

For most of the current metric-based FSL methods, the framework generally includes an embedding and a distance measurement module. Enhancing the feature representation means getting a better embedding. In the pipeline of most networks, the CNN backbone is used for embedding an image into feature space. Generally, at the end of the CNN, a global average pooling (GAP) is used to vectorize the feature maps. Under the data-limited condition, we try to mine more features from embedding. For this reason, we propose to use global max pooling (GMP) and GAP together to enhance the feature representation. In addition, we found that instead of a Gaussian distribution, the feature vectors have a right-skewed distribution in our previous work (Lin et al., 2022a). Gaussian-like calibration (GC) is used to make the distribution of features close to the normal distribution. The data distribution affects the performance of the distance measurement. Power transform (PT) is one of the methods adopted in this work for calibrating the skewness. Hence, in our design, we also adopt PT to calibrate features. The GAP, GMP, and PT are used for the instance embedding.

In addition to instance embedding, task adaptation is a popular method to improve the feature representation. In the FSL paradigm, because the classification relies on the given support set, the context of the support set significantly affects the identification result. The difficulty of the classification greatly depends on the identification range in which the object is

**Abbreviations:** FSL, few-shot learning; TLA, tobacco leaf abnormalities dataset; PV, PlantVillage; GAP, global avg-pooling; GMP, global max-pooling; GC, Gaussian-like calibration; PT, power transform; TA, task adaptation.

classified. Specifically, it is a challenging issue for tasks with high similarity categories. For example, given two tasks as shown in Figure 1, task 1 contains three different diseases of grape, and task 2 contains three different diseases belonging to grape, apple, and peach. Now the query sample is a kind of grape disease. The stars are the embeddings after going through the encoder independently. Obviously, the identification of the query sample is quite difficult in task 1 but much easier in task 2 due to the context. However, cases like task 1 occur more frequently because users are concerned with the identification of diseases belonging to the same species than those belonging to different species. The images of the same species always share many common features, which makes the classification difficult. That is the reason that in (Lin et al., 2022a, Lin et al., 2022b), the identification of the 10 categories of tomatoes is the most difficult task. For cases like task 1, an independent and static embedding without contextualization is not enough. If the embeddings can be adapted according to their current context, such as these circles, the features will be more discriminative. The dynamic contextual adaptation and the classification will be easier and more flexible.

Yoon et al. (2019) proposed TapNet by using feature linear projection for task adaptation. Baik et al. (2021) proposed a task-adaptive loss function. Zhao et al. (2021) proposed a cross-nonlocal neural network for capturing the long-range dependency of the samples and the current task. Lichtenstein et al. (2020) proposed TAFSSL in transductive and semisupervised cases when some additional unlabeled data accompanies them. In Huang et al. (2022), a task-adaptive transformer module was proposed to automatically establish links between support and query images. Some methods concerning the relationship of the context, such as transformer (Vaswani et al., 2017), LSTM (Shi et al., 2015), deep set (Zaheer et al., 2017), graph convolutional network (Zhang et al., 2019), can also be used. For example, the FEAT proposed in Ye et al. (2020) uses a transformer to adapt the support set.

Based on the above analysis, we propose a feature representation enhancement network (FREN) that intends to improve the feature representation not only in instance embedding but also in task adaptation. We use the meta-baseline (Chen et al., 2021) as the baseline network, which is a metric-based FSL network. For instance, in embedding, we use GAP, GMP, and PT to enhance the feature representation. For task adaptation, we get inspiration from FEAT and adopt self-attention for contextualization. Different from their works, our method adapts the support feature vectors and the query feature vectors both to keep them in a consistent feature space. Another difference is that they adapt the centroids, but we adapt the support feature vector first before calculating centroids to preserve more features.

In brief, the main contributions of this work are summarized in three ways:

1. We propose the network FREN, which integrates double pooling, self-attention, and Gaussian-like calibration for enhancing feature representation.
2. We create the dataset TLA, which fills the gap in tobacco disease data.
3. We demonstrate that FREN outperforms the other related works and has good generalization. The identification of tobacco abnormalities is discussed, and some solutions for the application are proposed. These solutions also can be applied to other plants.

The rest of this paper is organized as follows: Section 2 is the details of our method. Section 3 introduces the materials used in this work, including hardware and data. Three datasets are Mini-Imagenet, PV, and TLA. The settings for data are also introduced in this section. Section 4 is about the experiments and results. Identification of 10 categories of tomato of PV and 16 categories of tobacco abnormalities are illustrated in this section. Section 5 is the discussion, including the motivation, contributions findings, and limitations of this work. Section 6 is the conclusion.

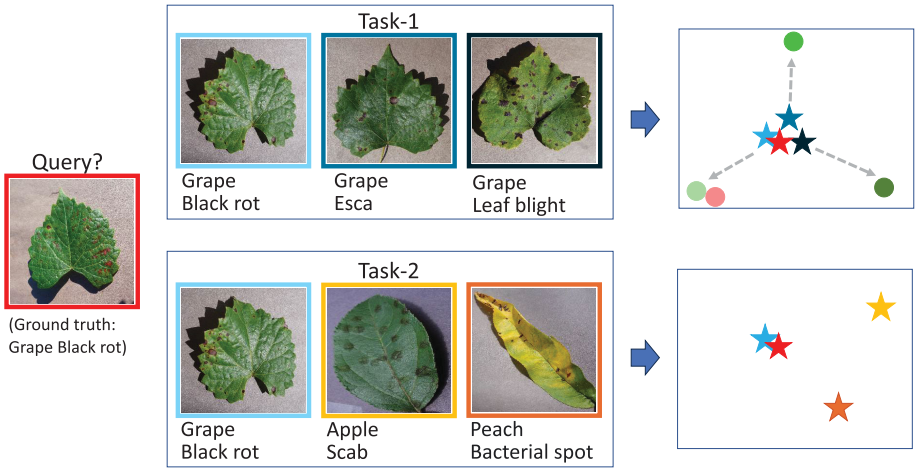



FIGURE 1
An example indicating the significance of task adaptation.

# 2 Method

## 2.1 Problem definition

FSL is the method by which the categories appearing in the test are never seen in training. For the identification of the unseen categories, only a few samples are given as supporting materials. The data are organized as tasks denoted as $T$, defined in Equation 2. Each task $T_i$ includes a support set and a query set, which are denoted as $S$ and $Q$. The $S$ contains $n$ categories, and each category contains $k$ samples, which is denoted as *n-way, k-shot*. The categories of $Q$ should be covered in the range of $S$, and the number of samples $w$ in $Q$ is not limited. The objective is to classify the $w$ samples into the $n$ categories. It is a supervised learning method, which means the samples used in training are given labels. It is denoted by $(x,y)$, which is a (*image*, *label*) pair. The problem can be formulized as follows:

$$C_{train} \cap C_{test} = \varnothing \tag{1}$$

$$T = (S(n-way, k-shot), Q(n-way, w-samples)) \tag{2}$$

where $C_{train}$ is the category in training, $C_{test}$ is the category in testing; Equation 1 means there is no intersection between them. The pair of (*n-way*, *k-shot*) indicates the difficulty of the task. The increase of $n$ indicates the increase in the complexity of the task, and the increase of $k$ indicates that the task gets more support.

## 2.2 Framework

The training contains two steps, which are pretraining and meta-learning. In pretraining, a linear layer is used as the classifier, and data are used image-wise. In meta-learning, the network is initialized by the trained model from pretraining. The linear layer is replaced by a distance measurement module. The data are used task-wise. The purpose of pretraining is to provide a pretrained encoder for meta-learning. This stage mimics the human cognitive mechanism, in which humans already have prior knowledge before doing a specific task. While the goal of meta-learning is to learn to learn, a linear layer maps features to a specific set of categories, while a distance measurement module is used to distinguish between the different categories, whatever they may be. A well-trained, pretrained encoder can be seen as a solid foundation that facilitates subsequent meta-learning.

The architecture is still the classical format: embedding + distance measurement module, as shown in Figure 2. In this work, the embedding includes two parts, which are instance embedding and task adaptation (TA). Each task goes through the encoder to be a set of feature vectors, which includes $V^S$ and $V^Q$. Each feature representation is concatenated by the results of GMP and GAP. GC is a component used to calibrate the skewed distribution of the vectors from GMP and GAP, respectively. After GC, the outputs of instance embedding are $V^{S'}$ and $V^{Q'}$. Subsequently, the two sets are contextualized by self-attention,

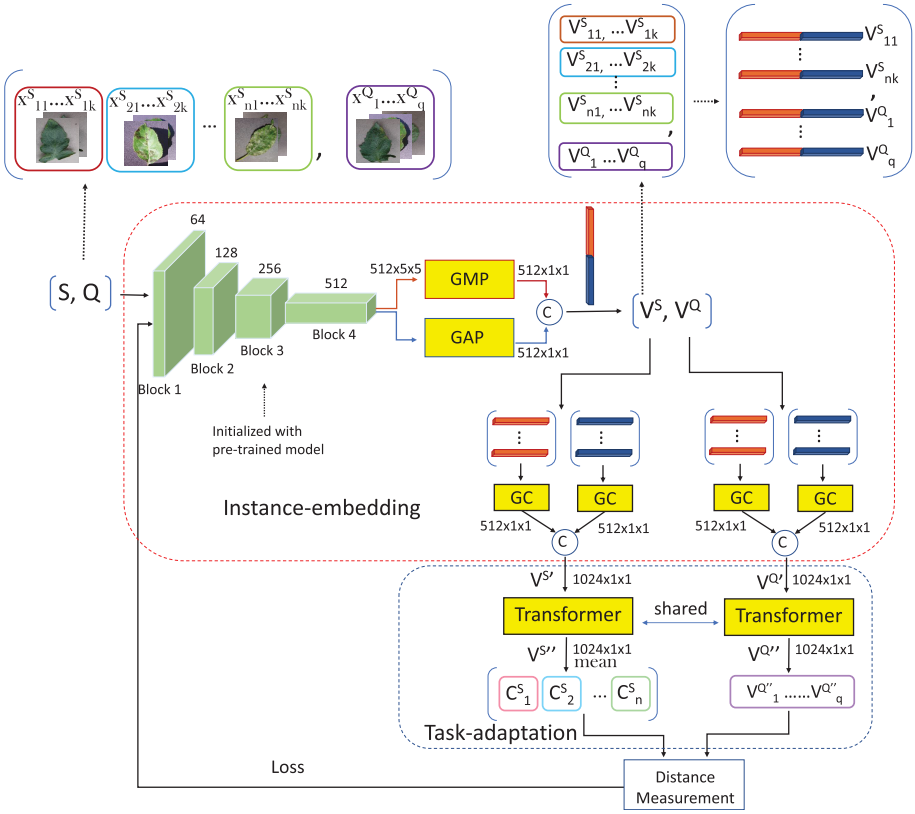



FIGURE 2
The architecture of FREN.

respectively, to be $V^{S''}$ and $V^{Q''}$. The mean vectors of $V^{S''}$ by categories are calculated as centroids, which are defined as Equation 3:

$$C_i^S = \frac{1}{k} \sum_{j \in k} f_\theta(x_{ij}) \quad (i \in n,\ x \in S) \tag{3}$$

where $C_i^S$ denotes the centroid of the $i$th category in $S$, $f_\theta$ includes instance embedding and task adaptation, as shown in Figure 2, the $x_{ij}$ denotes the $j$th sample of the $i$th category in $S$, the $n$ is the number of ways, and the $k$ is the number of shots in the task.

At the end, the distances of each vector $V^{Q''}$ from the centroids are calculated to determine the classification. Cosine similarity is used to calculate the distances to determine the classification (Han et al., 2012) defined in Equation 4:

$$< A,\ B > \ = \ \frac{A \cdot B}{||A||\ ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4}$$

The probability that sample $x^Q$ belongs to category $C_i$ is calculated with Equation 5:

$$p(y = C_i | x^Q,\ S) = \frac{exp\ (< f_\theta(x^Q),\ \ C_i >)}{\sum_{i \in n} exp(< f_\theta(x^Q),\ C_i >)} \tag{5}$$

where $p(y = C_i|x^Q,S)$ is the softmax possibility of the sample $x^Q$ belonging to the category $C_i$ by given $S$, and<.,.> denotes the distance of two vectors. Due to the fact that it is a classification task, we use cross-entropy loss defined in Equation 6:

$$L(y,\ C_i) = -\frac{1}{q} \sum_{i=1}^{q} C_i\ log\ (P((y = C_i)|x^Q,\ S)) \tag{6}$$

where $q$ is the number of query samples of a task. The objective is defined in Equation 7:

$$\theta \leftarrow \underset{\theta}{\operatorname{argmin}} -\frac{1}{w} \sum_{x^Q \in Q} \sum_{c \in C} \lambda log\ (P(y = c|x^Q,\ S)) \tag{7}$$

where $\theta$ is trainable parameters; $w$ is the number of samples in $Q$, $C$ is the category set of $S$; $\lambda$ is a sign value, it is 1 when $y = c$, it is 0 when $y \neq c$; $y$ is the prediction category; and $P(y = c|x^Q,S)$ is the possibility of query sample $x^Q$ supporting $S$ belongs to class $c$.

### 2.2.1 Instance embedding

In this work, we adopt Resnet12 as the CNN encoder, which includes four residual blocks. The output from the last residual block is a set of feature maps with the shape of $c \times h \times w$ ($c$ is the channels, $h$ and $w$ are the height and weight of each feature map). The shape of a tensor from the last residual block of Resnet12 is 512 × 5 × 5. Global pooling is adopted to reduce the dimension of each feature map to a value. By using global pooling, the shape of the tensor becomes 512 × 1 × 1, which can be seen as a vector. A vector is much easier for subsequent computing. The GAP is the most common operation for downsampling. Each feature map is seen as a 5 × 5 matrix. For a matrix, besides the mean value, the max value, the min value, and the standard deviation are also feature values. These feature values have different purposes. Generally, the average value is commonly used to represent a set. The maximum value and the minimum value, to some extent, represent significant features. Only using GAP is crude, as many useful features are lost. More features should be mined from these feature maps, especially under the few-shot condition. Therefore, besides GAP, GMP is used to vectorize the feature maps. The outputs of GAP and GMP are concatenated into one feature vector.

For the vectors from double pooling, GC is executed to make these vectors close to a normal distribution and it does not change the dimension of the feature vector. The subvectors of GAP and GMP should be calibrated, respectively, and then concatenated together. The dimension of the feature vector output from the instance embedding is 1,024 × 1 × 1. The GC is arranged before the self-attention because both the two modules change the vectors, and the effectiveness of task adaptation could be weakened or destroyed by calibration.

PT is a family of functions applied to create a monotonic transformation of data using power functions, which makes the data a more Gaussian-like distribution. As a method of Gaussian calibration, PT can adjust the distribution shape of data to some extent, making it more consistent with modeling assumptions and improving modeling accuracy. It is described by Tukey in Tukey et al. (1977) as Equation 8:

$$v' = \begin{cases} (v + 1e-6)^\beta & if \quad \beta > 0 \\ log(v + 1e-6) & if \quad \beta = 0 \\ -((v + 1e-6)^\beta) & if \quad \beta < 0 \end{cases} \tag{8}$$

where $v = [v_1,\ldots, v_i,\ldots,v_d] \in \mathbb{R}^d$ is a $d$ dimension vector, $1 \le i \le d$, $v_i$ denotes its value in the $i$th position, $\epsilon = 1e{-}6$ is used to guarantee that $f_\theta(x) + \epsilon$ is strictly positive in every position, and $\beta$ is a hyperparameter to determine the skewing degree. Note that $\beta = 1$ leads to no effect, and decreasing $\beta$ can phase out the right-skewed distribution. It has been demonstrated that when $\beta$ is 0.5, the feature distribution gets closest to the Gaussian distribution, and the results are the best (Lin et al., 2022a). We also use $\beta = 0.5$ in our experiments. We designed a GC module that consists of four steps:

- Nonnegative processing: the raw feature vector contains negative values. A nonnegative processing is needed before calculating the square root ($\beta = 0.5$). We lift all values of a vector until no negative values exist by changing the data distribution.
- PT.
- Euclidean normalization: it is used to scale the features to avoid the large variance feature vectors that predominate the others (Hu et al., 2021).
- Centralization: it makes all values symmetrical on the $y$-axis.

### 2.2.2 Task adaptation

In task adaptation, the desired outputs from self-attention are two sets of contextualized feature vectors of $V^{S'}$ and $V^{Q'}$, denoted as $V^{S''}$ and $V^{Q''}$. The self-attention does not change the dimension of each input element. Hence, the dimension of each feature vector $V^{S''}$ remains at 1,024 × 1 × 1. In FSL, the context mainly means the support set.

However, if only adapting $V^{S'}$ without $V^{Q'}$, the adaptation may cause a gap between $V^{S''}$ and $V^{Q''}$. Hence, $V^{S'}$ and $V^{S'}$ are both adapted. After adaptation, $V^{S''}$ is used to calculate centroids.

The self-attention, also known as an intra-attention mechanism, is designed to capture dependencies and relationships between different elements within a sequence of data. Self-attention has several advantages, including the ability to capture long-range dependencies, handle variable-length inputs, and model relationships between distant elements in the sequence. Also, it has the permutation-invariant property to keep the output sequence in the same order as the input. This is an important attribute to guarantee that a certain output vector corresponds absolutely to the input vector. Because the vectors belonging to the same category are used to calculate the mean vector after adaptation, it is important to ensure that these vectors remain in the same order without causing any confusion. These characteristics are very suitable for us to use self-attention for task adaptation.

In the implementation of the self-attention, for the input set $X$, three vectors, $Q$, $K$, and $V$ are generated by linear projection with three learnable matrixes, $W_Q$, $W_K$, and $W_V$. The $Q$ is seen as a query, the $K$ is the key, and the $V$ is the value of each element in the set. The relevance of a query element to the other elements is calculated by a scaled dot product of the $Q$ with each $K$. The output is a set of values called attention score, which will be used as the weight to extract features from each $V$. A higher score means stronger relevance of the two elements. The transformed vector is not itself but the summation of the attention score weighted values, which means the output has already been contextualized. The matrix representation can be described in Equations 9, 10:

$$Attention(Q,\ K,\ V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{9}$$

$$Q = X \times W_Q; K = X \times W_K; V = X \times W_V \tag{10}$$

# 3 Materials

## 3.1 Hardware

The configuration of hardware used in this work is: graphics: Tesla V100-DGXS-32 GB; video memory: 32 G; processor: Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20 GHz; and operating system: Ubuntu 18.04.6 LTS. The deep learning framework is PyTorch.

## 3.2 Data

In this work, we use three datasets in experiments, which are Mini-Imagenet, PV, and TLA. The settings and purposes of use are described below.

### 3.2.1 Mini-Imagenet

As a subset of Imagenet, Mini-Imagenet includes 100 categories and 600 images per category. This dataset is a general dataset that includes categories in a wide range. In this work, it is used as pretraining material and also as testing material for cross-domain.

### 3.2.2 PlantVillage

PV (Hughes and Salathé, 2015) is a dataset of plant diseases. It was released in 2015 by Pennsylvania State University. It is the most frequently used dataset in academic research up to now for plant disease recognition. It includes 50,403 images crossing over 14 crop species and covering 38 categories, as shown in Figure 3. This dataset is used for two purposes: (1) to verify the superiority of our method by comparing it with related works; and (2) to be the material in cross-domain testing. We use the data after augmentation and select 1,000 images per category.

### 3.2.3 Tobacco leaf abnormality dataset

All the images of TLA were collected in July and August, which is the mature period of tobacco. These images were taken in the field at the location (N25.75, W100.13). The temperature was in the range of 16°C–28°C. It is the rainy season in this location. The photographic equipment is a Canon digital camera (Canon D200). The original resolution of the image is 5,184 × 3,456. To maintain the practical significance of this dataset, we try to include all abnormalities found during the fieldwork. The tobacco agronomists undertake the labeling work to guarantee the accuracy of labels. This dataset contains the most common infections of tobacco and can be extensively used for identification in other tobacco cultivation areas (ITAS, 2020). Finally, we classified 16 categories, which include 10 infection diseases from bacteria, fungi, and viruses, three nonparasitic diseases, two pest-trace left, and a healthy category. Two settings are included in TLA: raw setting and processed setting.

In this work, only single-disease identification is involved. Multidisease identification is discussed in Section 5. We try to select the images where only one disease occurs in each image to raw setting. The raw data show the long-tail distribution, obviously. Some diseases are very common in this field, such as wildfire, frog eye, tobacco mosaic virus (TMV), weather fleck, etc., and some diseases are rarely found, such as anthracnose, tomato spotted wilt virus (TSWV), etc. The raw setting meets the requirement of up to seven-shot testing because the least number of samples is eight. The information on raw setting is listed in Table 1, and the samples are shown in Figure 4. Meanwhile, a processed setting is provided in TLA. We preprocess the images by slicing them so that only one disease is included in each image. There are still 16 categories in the processed setting. The number of samples is listed in Table 1, and the samples are shown in Figure 5.

There are three reasons for the segmentation of the raw images. First, the segmentation can help reduce the effects of surroundings, such as other leaves, soils, etc. Second, segmentation can help highlight the spots. Because some disease spots are very small, such as frog eye, resizing and downsampling cause significant loss to these small spots. Third, the segmentation can help increase the number of samples of these uncommon diseases. After segmentation, the number of these categories is increased, such as TSWV, anthracnose, black shank, and genetic abnormality. The

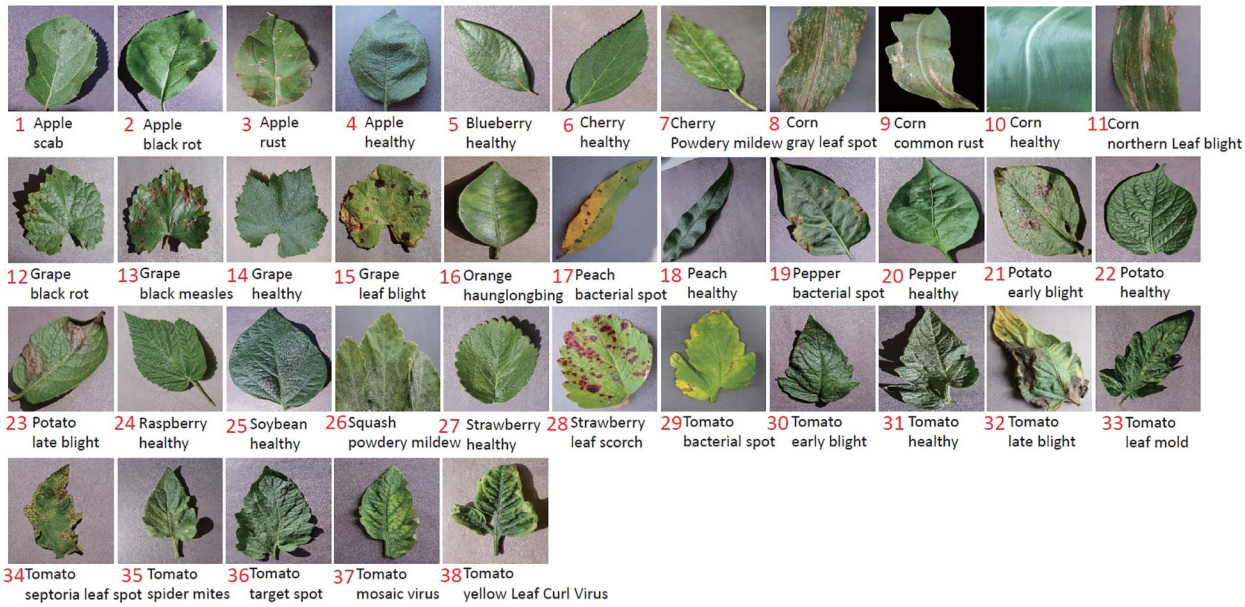



FIGURE 3
The samples of 38 categories in PV.

images of TSWV and anthracnose are increased more than black shank and genetic abnormality because the lesions of the first two diseases are smaller than the last two; the segmentation operation does not destroy the intact lesion features. Again, the segmentation can guarantee that each image includes a single disease.

By using computer vision for disease identification, the key is to identify the discriminative features without any biochemical testing. Hence, the visible symptoms of different diseases are very critical. In fact, many diseases show very similar symptoms initially and then show some different characteristics as they progress. The typical symptoms are summarized below.

1. Wildfire. The spots are initially very small and greasy and later on become necrotic and quickly turn brown. Lesions may conflate on the lamina, and altered tissues may rot and fall. With tabtoxine-producing strains, the spots are surrounded by a more or less marked yellow halo.
2. Brown spot. The spots first are small wet spots and become brown and circular rapidly, often containing discrete concentric rings and surrounded by a halo of chlorotic tissue. In humid conditions, it has a black, velvet-like conidial layer on the surface of spots.
3. Frog eye. The lesions are small, circular, light beige, and parchment-like. Some spots are covered by tiny black dots composed of clusters of conidiophores and conidia.
4. Anthracnose. This lesion is initially dark and oily, then becomes greyish, parchment-like, and surrounded by a brown border. The acervulus can be observed in the center of the enlarged spots.
5. Target spot. The first symptoms of the target spot are small white or tan-colored primary lesions. Next, a series of necrotic rings around the primary lesion are created as the spot spreads outward. It has a yellow-colored halo of chlorotic tissue bordering the outermost necrotic layer. The necrotic tissue in the center will split or fall out when the spot grows large enough.
6. Black shank. The typical features of this infection are on the roots and stems. Foliar infections occur during rainy periods. They are large brown or black spots on the lower leaves.
7. TMV. Mottling and, more or less importantly, “vein banding” are observed. The lamina is sometimes heavily deformed by the presence of blisters. Leaves can curl up and become filiform.
8. Cucumber mosaic virus (CMV). More or less severe mosaic patterns, vein banding, or interveinal yellowing. It causes various anomalies of the lamina, such as blisters, filiform shape, or curling. Following the veins, localized necrotic lesions consisting of small beige to brown etches are observed. The chlorotic or necrotic lines give lamina an “oak leaf” appearance sometimes.
9. Potato virus Y (PVY). Mottling, vein yellowing (vein clearing), or greener lamina areas along the veins (vein banding) can be observed. Browning of the midrib or secondary veins shows on the lamina. When the infections are very severe, necrosis tissues with a beige to brown color may appear close to the veins.
10. TSWV. Zonate necrotic spots and concentric necrotic rings on the leaves can be observed. They are yellow at first but quickly turn to a reddish-brown color. The apex leaves are also distorted. Brown to black elongated lesions are observed on the veins, petioles, and stems. The top of the plant sometimes bends toward the ground.

TABLE 1 The categories and number of samples in TLA.

| Super category | ID | Category | Raw setting | Processed setting |
|---|---|---|---|---|
| Bacteria | | | | |
| | 1 | *Pseudomonas syringae* pv. *tabaci* (wildfire) | 89 | 89 |
| Airborne fungi | | | | |
| | 2 | Alternaria leaf spot (brown spot) | 50 | 50 |
| | 3 | *Cercospora nicotianae* (frog eye) | 50 | 50 |
| | 4 | *Colletotrichum tabacum* (anthracnose) | 8 | 27 |
| Soil-borne fungi | | | | |
| | 5 | Target spot | 50 | 50 |
| | 6 | *Phytophthora nicotianae* (black shank) | 11 | 12 |
| Virus | | | | |
| | 7 | Tobacco mosaic virus (TMV) | 66 | 66 |
| | 8 | Cucumber mosaic virus (CMV) | 51 | 51 |
| | 9 | Potato virus Y (PVY) | 35 | 35 |
| | 10 | Tomato spotted wilt virus (TSWV) | 8 | 24 |
| Nonparasitic | | | | |
| | 11 | Pollution spots (weather fleck) | 59 | 59 |
| | 12 | Sunburn/sunscald | 68 | 68 |
| | 13 | Genetic abnormality | 9 | 13 |
| Pest-trace left | | | | |
| | 14 | *Phthorimaea operculella* | 50 | 50 |
| | 15 | Nematodes | 41 | 41 |
| Others | | | | |
| | 16 | Healthy | 50 | 50 |

11. Weather fleck. It is induced by ozone ($O_3$), which is an air pollutant. After exposure to high levels of atmospheric pollutants, leaves may develop dark green water-soaked spots. Within hours, the spots turn dark brown, sometimes remain brown, but often turn white within 48 h, and the spots may coalesce.
12. Sunscald. Wide parts of tissues turning brown and gradually drying up are particularly observed on the leaves exposed to sunlight during periods of extreme heat.
13. Genetic abnormality. The plant modified the habit, color, and shape of some of its organs.
14. *Phthorimaea operculella*. The affected tobacco leaves were caved into wide submerged channels, leaving only the upper and lower transparent epidermis, which later turned into irregular yellow-brown or russet patches.
15. Nematodes. The distinctive features of root-knot nematodes are found mainly in the roots, and damage to the leaves is mainly wilt along the leaf margins and leaf tips.

# 4 Experiment and result

In total, we conducted 17 experiments to verify the proposed methods and the identification results of the TLA. These experiments were conducted with different data settings in pretraining, meta-learning, and testing. For different experiments, the results with different *n-way* and *k-shot* are reported. The configurations are summarized in Table 2 for better understanding.

## 4.1 Implementation details

In our experiments, Resnet12 is adopted as the backbone network. The Resnet12 includes four residual blocks. The channels of kernels in the four blocks are [64, 128, 256, 512]. Each block contains three $3 \times 3$ convolutional layers with one stride, a ReLU activation function, and a max pooling for downsampling. All images are resized as $3 \times 80 \times 80$. After going through the four

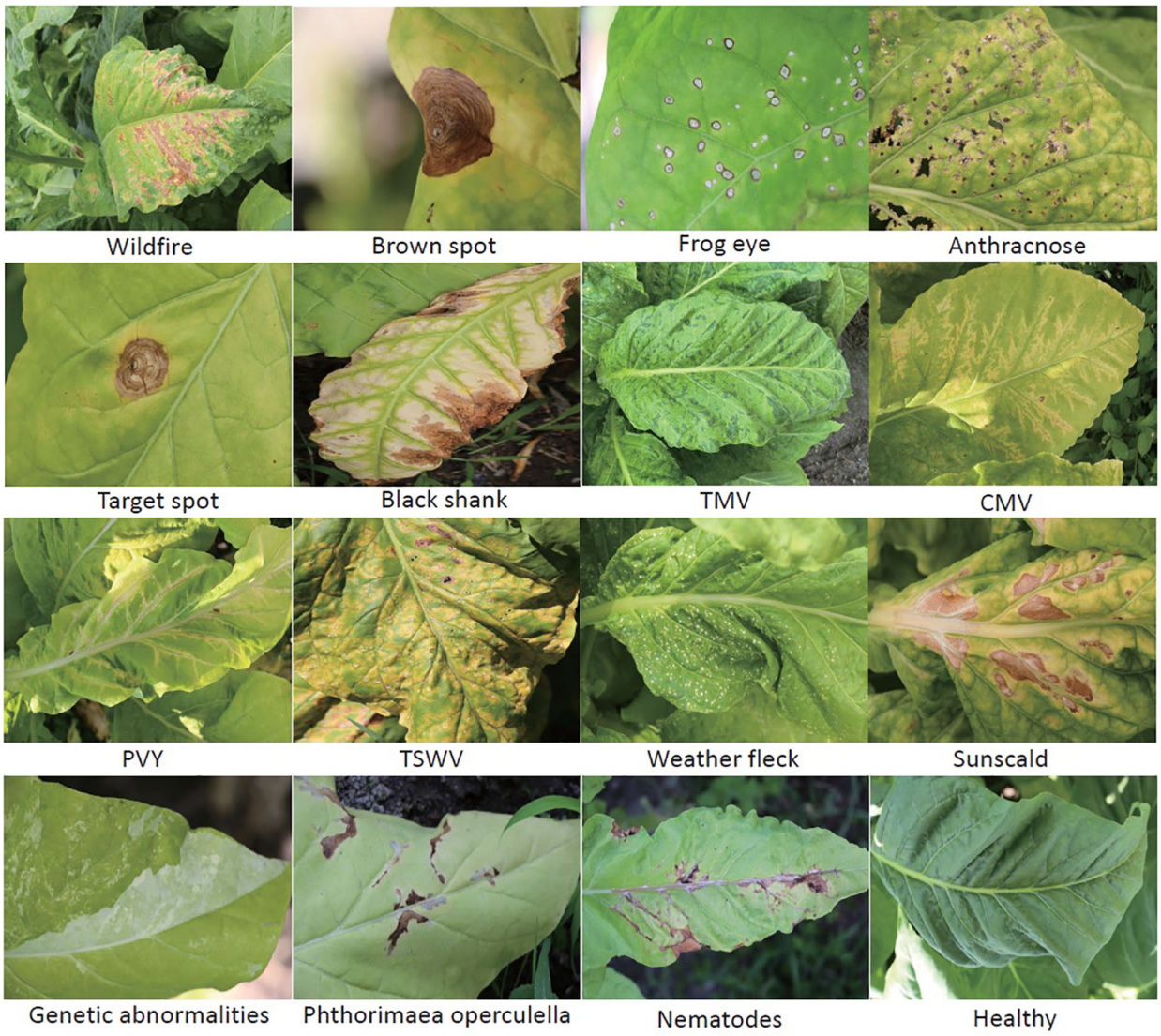



FIGURE 4
The 15 abnormalities and a healthy category in TLA.

residual blocks, each image is parsed into feature maps of 512 × 5 × 5. In pretraining, the batch size is 128, training epochs are 100, the optimizer is SGD, the learning rate is 0.1, weight decay is 5e-4, and milestones are 90. In meta-learning, the batch size is 200, each contains four tasks, training epochs are 20, the optimizer is SGD, the learning rate is 0.001, and weight decay is 5e-4.

## 4.2 Experiments on PV

All the identification accuracies are average accuracy values (ACC (%)) of 10 epochs. First, we conduct a group of experiments to show the improvement of our method by comparing it with the baseline and FEAT. In e1 to e7, the most difficult also the most meaningful setting of PV is used. Ten categories (ID: 29-30) belonging to tomato are used for testing, and the remaining 28 categories (ID: 1-28) covering 13 species are used for training (category ID refer to Figure 3).

As shown in Table 3, we conducted experiments on baseline, FEAT, and FREN, and a group of ablation experiments on FREN. The e1 is the baseline in this work. In baseline, only a GAP is used to vectorize the feature maps. The e1, e3, e4, e6, and e7 are the ablation experiments. On the baseline, we add the TA module in the e3. Compared with the e1, the accuracy of the 1-shot setting has been improved from 57.46% to 62.50%. It indicates the effectiveness of the TA module. In the e4, we add the GMP on the baseline, and the accuracy of the 1-shot setting has been improved from 57.46% to 62.62%. It validates that the GMP also enhances feature representation. In e6, we add GMP and TA to the baseline. The accuracy of e6 is higher than e4 and e4, which means that the two components both make contributions to the improvements. It also indicates that self-attention works well with the double-pooling

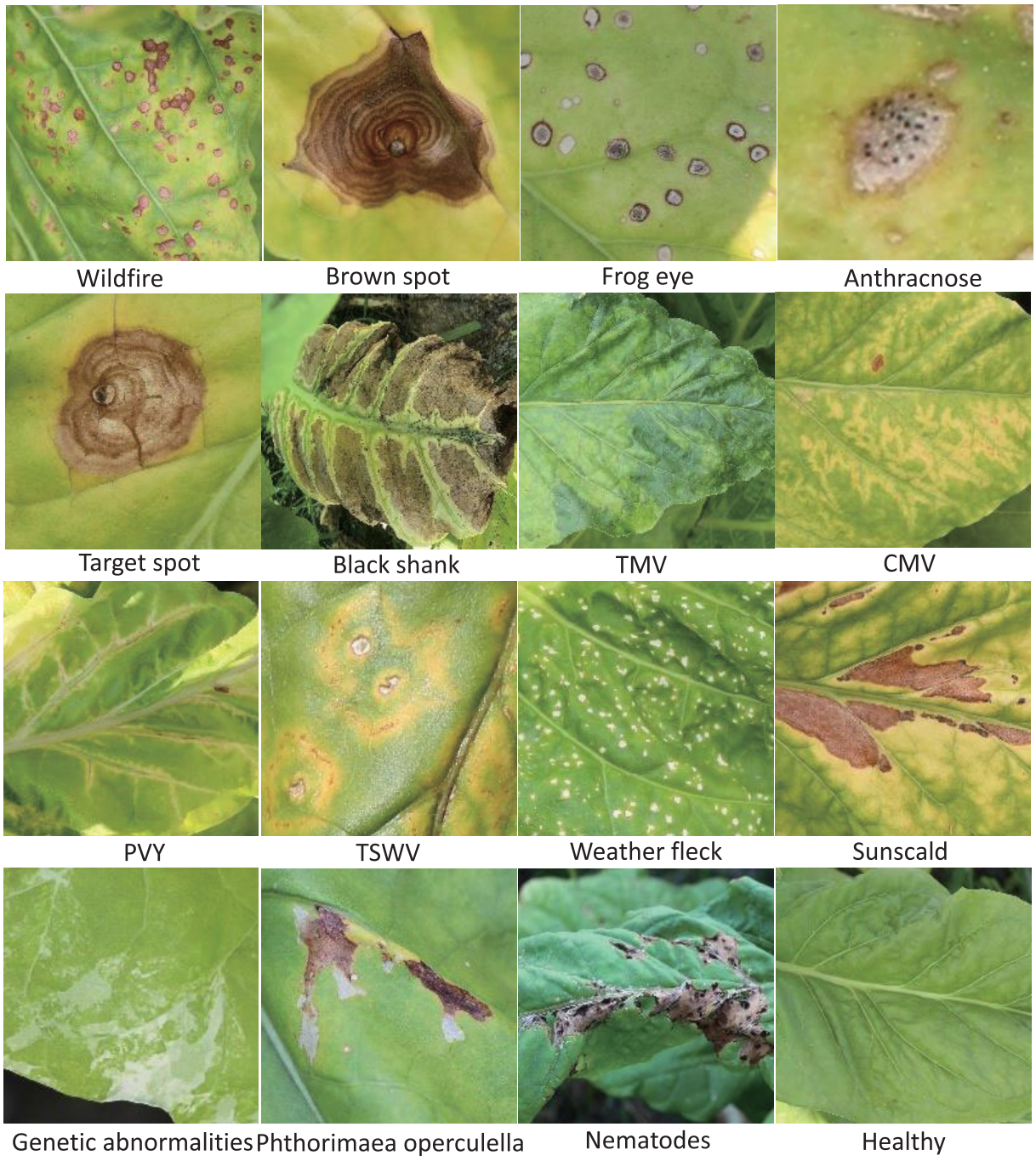



FIGURE 5
The images of the processed setting in TLA.

vectors. The e2 and e5 are two experiments to compare the FEAT with the proposed FREN. In FEAT, they also use a task-adaptation module. We got inspiration from their work, but our design is different from theirs. Compared with e2 and e3, the results of e3 are better than e2, which indicates that our task adaptation is more effective than theirs. Comparing e2 with e1, the FEAT is more powerful than e1, which indicates that task adaptation is critical for the FSL method. Compared with the e2 and e5, the improvement of GMP is again validated. Compared with e4 and e5, the two groups of results are very close, which indicates that the contribution of the task adaptation is weak. While comparing the e5 to the e6, the superiority of the TA proposed in our work has again been verified. In the e7, the FREN achieves the best performance by using the GAP, GMP, and GC together to be the independent embedding and using the TA for task adaptation. It improves the accuracy by about 10% on the 1-shot and 5-shot tasks compared with the baseline. The

TABLE 2 The configuration of our experiments.

| ID | Pretraining | Meta-learning | Testing | *n*-way | *k*-shot |
|---|---|---|---|---|---|
| e1, e2, e3, e4, e5, e6, e7 | PV-28 | PV-28 | PV-10-T | 5 | 1, 5, 10, 20, 30, 40, 50 |
| e8, e9 | PV-28 | PV-28 | PV-10-T | 5 | 1, 5, 10 |
| e10, e11, e12, e13, e14, e15 | Mini-100 | PV-38 | TLA-raw | 16 | 1, 5 |
| e16 | Mini-100 | PV-38 | TLA-pro | 16 | 1, 5, 10 |
| e17-A, e17-B | Mini-100 | PV-38 | TLA-pro | 20 | 1, 5, 10 |

The PV is separated into two parts: PV-28 and PV-10-T.
PV-10-T, the 10 categories of tomatoes in PV; PV-28, the remaining 28 categories of PV; PV-38, the full 38 categories of PV; Mini-100, the full 100 categories of Mini-Imagenet; TLA-raw, the raw setting of TLA; TLA-pro, the processed setting of TLA.

training loss, training accuracy, validation loss, and validation accuracy of pretraining and meta-learning are shown in Figure 6.

We use t-SNE (Van der Maaten and Hinton, 2008) to visualize the results of baseline and FREN, as shown in Figure 7. It is obvious that the feature vectors parsing by the baseline are interlaced without a clear classification border. By using FREN, the feature vectors are clustered more tightly in each category, and the distance between categories is expanded.

Multi-head self-attention is an extension of the self-attention mechanism in the transformer architecture. By employing multiple attention heads, the model can capture different types of information and learn more complex relationships within the input sequence. The $Q$, $K$, and $V$ are linearly projected for $h$ times. The independent attention outputs are then concatenated and once again projected to obtain the final values. We conducted a group of comparative experiments to show the effect of $h$ heads. The results are listed in Table 4. It shows that the performance does not improve with the increase in heads. For a lightweight CNN (e.g., Resnet12), the self-attention module accounts for a large proportion of the overall model. When increasing the number of heads, the size of self-attention also increases sharply. The lightweight deep networks perform better than larger or deeper networks in FSL (Lin et al., 2022b). This finding is demonstrated in this work again. In the rest of the experiments, we use one-head self-attention.

## 4.3 Compared with related works

In order to show the superiority of our method, we compare it with some recent related research. As mentioned, various data settings for PV have been used in existing works. In them, the setting of 10 categories of tomatoes is the most difficult setting for testing. The reason is that when the categories are very similar to each other, the classification becomes challenging. However, this setting is more concerning for farmers because they are concerned with identifying the different diseases of the same plant. So, here, we compare with recent work that also conducted experiments with the same data setting and used the FSL methods, which are close to our research. These works are semisupervised method (Li and Chao, 2021), transformer contextualization + Mahalanobis distance (Nuthalapati and Tunga, 2021), multiscale feature fusion + channel attention (Lin et al., 2022b), frequency feature representation + Gaussian calibration (Lin et al., 2022a). The semisupervised method by Li and Chao (2021) is an early work to utilize a semisupervised learning strategy in FSL. It is a new attempt that follows the saturation of traditional FSL methods for plant disease identification. In the work of Nuthalapati and Tunga (2021), the authors also use self-attention to contextualize the support set and use squared Mahalanobis distance to calculate the distance of the query samples to the support samples. This is a typical example of task adaptation in this research field. The works on multiscale feature fusion

TABLE 3 The ACC (%) of baseline, FEAT, and the ablation experiments of FREN.

| ID | Method | Module | | | | | *k*-shot | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | GAP | GMP | FEAT | TA | GC | 1 | 5 | 10 | 20 | 30 | 40 | 50 |
| el | Baseline | ✓ | | | | | 57.46 | 75.12 | 79.32 | 81.41 | 82.48 | 83.32 | 83.50 |
| e2 | FEAT | ✓ | | ✓ | | | 61.57 | 79.67 | 83.57 | 85.51 | 86.43 | 86.93 | 87.01 |
| e3 | | ✓ | | | ✓ | | 62.50 | 81.28 | 85.29 | 87.21 | 88.24 | 88.57 | 88.75 |
| e4 | | ✓ | ✓ | | | | 62.21 | 79.52 | 83.00 | 84.74 | 85.62 | 86.29 | 86.57 |
| e5 | | ✓ | ✓ | ✓ | | | 62.17 | 80.09 | 83.77 | 85.66 | 86.71 | 87.14 | 87.45 |
| e6 | | ✓ | ✓ | | ✓ | | 64.27 | 83.04 | 86.62 | 88.54 | 89.33 | 89.78 | 89.75 |
| e7 | FREN | ✓ | ✓ | | ✓ | ✓ | **66.06** | **84.22** | **87.52** | **89.48** | **90.35** | **90.48** | **90.82** |

Experiment configuration: data (10 categories of tomato of PV for testing, the remaining 28 categories of PV for training), Resnet12, 5-way, one-head, lr (pretraining: 0.1, meta-learning: 0.001), epoch (pretraining: 100, meta-learning: 50).
Bold font indicates the highest results.




FIGURE 6
**(A)** The loss and accuracy of training and validation in pretraining. **(B)** The loss and accuracy of training and validation in meta-learning.

+ channel attention (Lin et al., 2022b) and frequency feature representation + Gaussian calibration (Lin et al., 2022a) do not use task adaptation but feature enhancement. In the work of Lin et al. (2022a), they use a Gaussian calibration that has positive effects on the performance. The results are shown in Table 5. We can see that the proposed FREN outperforms these works on 1-*shot* to 20-*shot* settings. The results demonstrate that the enhancement of instance embedding and task adaptation are both significant, and the Gaussian calibration is not only effective for the frequency feature representation but also useful for spatial feature representation.

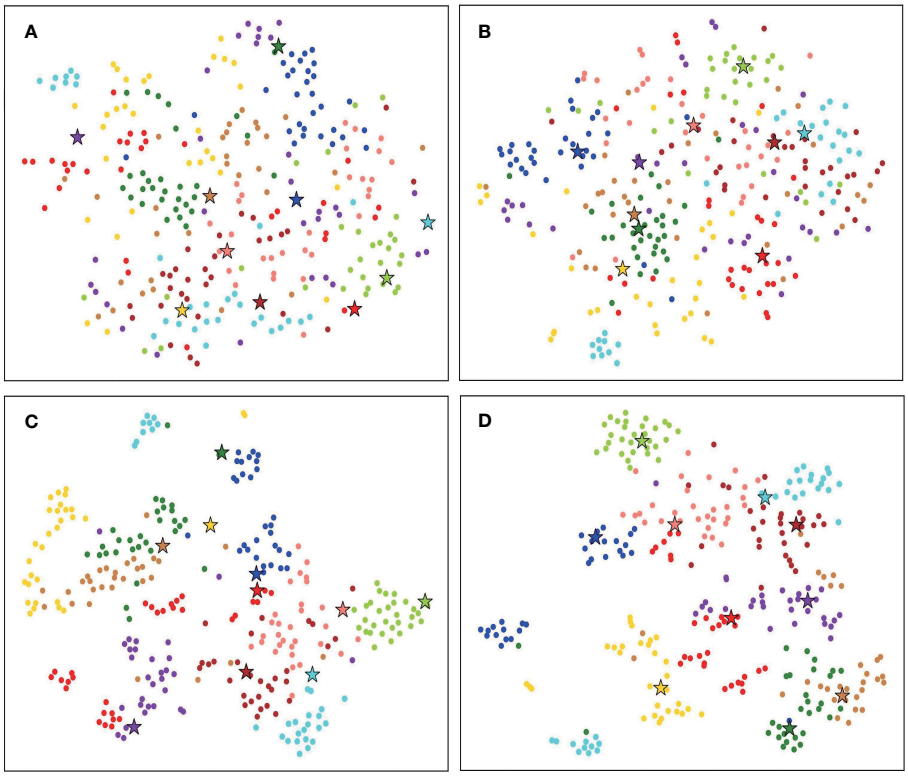



FIGURE 7
The t-SNE visualization of 10 tomato disease identification results. **(A)** The result of baseline 10-way, 1-shot task. **(B)** The result of baseline 10-way, 1-shot task. **(C)** The result of FREN 10-way, 1-shot task. **(D)** The result of FREN 10-way, 1-shot task.

TABLE 4 The ACC (%) of different heads of the self-attention.

| ID | Head | Parameter | Time | 1-shot | 5-shot | 10-shot |
|---|---|---|---|---|---|---|
| e7 | 1 | 12.2 M | 27.1 m | **66.06** | **84.22** | **87.52** |
| e8 | 2 | 16.4 M | 31.4 m | 65.78 | 84.06 | 87.43 |
| e9 | 3 | 20.6 M | 32.6 m | 66.02 | 84.16 | 87.47 |

Experiment configuration: data (10 categories of tomato of PV for testing, the remaining 28 categories of PV for training), Resnet12, 5-way, one head, lr (pretraining: 0.1, meta-learning: 0.001), epoch (pretraining: 100, metalearning: 50).
Bold font indicates the highest results.

## 4.4 Experiments on TLA

### 4.4.1 Group 1

In this section, we use FREN and TLA to discuss the identification of tobacco abnormalities. In this group, baseline, FEAT, and FREN are compared, Mini-Imagenet and PV are used as training data, and the raw setting of TLA is used as testing data. The goal is to identify the 16 categories in TLA. In pretraining, Mini-Imagenet is used. In meta-learning, Mini-Imagenet and PV are used to evaluate the generalization of FREN. In our previous work (Lin et al., 2022b), we found that a similar dataset used in meta-learning is more beneficial for cross-domain identification. In this work, we still use Mini-Imagenet as the general dataset and PV as a similar dataset of TLA.

The 5-way is the standard configuration in current research on FSL, which means that just five categories are selected in each task. In application, 5-way is not reasonable because it is hard to predict the category of the test data sample in the current five ways. Hence, for application purposes, the *n*-way is set as the full category (16 categories) in our experiments. The results of two configurations, 16-way, 1-shot, and 16-way, 5-shot, are reported, as shown in Table 6. The results of 1-shot are also shown in the confusion matrix in Figure 8. Each category is queried 1,000 times.

In meta-learning, e10, e12, and e14 use 100 categories of Mini-Imagenet, and e11, e13, and e15 use 38 categories of PV. All experiments use TLA for testing. In this group of experiments, we found that:

1. FREN outperforms the baseline and FEAT significantly.
2. In all three methods, the performance of using PV is better than using Mini-Imagenet in meta-learning. It verifies that using a similar dataset in meta-learning is a better choice for cross-domain.
3. Although using similar data is better than using general data in meta-learning, we found that the tendency is obvious in baseline and FEAT, but the performance gets very close in FREN. It means that FREN has better generalization in more complex cross-domain situations, such as from general datasets to TLA.
4. In the confusion matrix in Figure 8, it clearly shows the interplay between the categories. In the confusion matrix, the TP and TN indicate a right identification, which is located at the diagonal. The FP and FN indicate false identifications.

For wildfire, many categories, such as anthracnose, TSWV, sunscald brown spot, etc., have close features with it, as shown in Figure 9A. The reason for this is that wildfire is an infection that exhibits very different symptoms in its evolutionary phases. The diverse symptoms lead to confusion in identification.

Frog eye, anthracnose, and weather fleck are similar to each other, as shown in Figure 9B (1). The common pattern the three categories is the many small, independent white spots. The difference exists in details, as shown in Figure 9B (2); frog eye always has a dark brown or black border, anthracnose has the acervulus in the center of the spot, and weather fleck does not have an obvious border or acervulus.

Brown spots, target spots, and sunscalds have a similar appearance and are brown-colored lesions. The same as the target spot, the concentric ring is the most typical feature of the brown spot, as shown in Figure 9C (1). The difference between the two diseases is that the brown spot has a velvet-like conidial layer on the surface of the spot, but the target spot does not have it. Sunscald does not have an infection trace around the border and in the center of the lesion. The burned area is smooth and with a clear edge. The differences are as shown in Figure 9C (2).

TABLE 5 The ACC (%) of FREN compared with the related works.

| ID | Method | 1-shot | 5-shot | 10-shot | 20-shot |
|---|---|---|---|---|---|
| | Iterative SS (Li and Chao, 2021) | 34.0 | 53.1 | 68.8 | 75.6 |
| | Transformer + Mahalanobis (Nuthalapati and Tunga, 2021) | 46.6 | 63.5 | –[a] | –[a] |
| | CMSFF + CA (Lin et al., 2022b) | 60.7 | 78.1 | 82.2 | 84.5 |
| | Frequency + GC (Lin et al., 2022a) | 64.5 | 80.9 | 84.1 | 85.9 |
| e7 | FREN | **66.1** | **84.2** | **87.5** | **89.5** |

Testing data: 10 tomato categories in PV.
[a]Not applicable, bold font indicates the highest results.
Experiment configuration: data (training: ID 1-28 of PV10, test: ID 2938 of PV), Resnet12, 5-way, one-head, lr (pretraining: 0.1, meta-learning: 0.001), epoch (pretraining: 100, meta-learning: 50).

10.3389/fpls.2024.1333236

TABLE 6 The ACC (%) of raw setting of TLA using FREN.

| ID | | e10 | e11 | e12 | e13 | e14 | e15 |
|---|---|---|---|---|---|---|---|
| | Method | Baseline | Baseline | FEAT | FEAT | FREN | FREN |
| | Meta-learning data | Mini-100 | PV-38 | Mini-100 | PV-38 | Mini-100 | PV-38 |
| 16-Way, 1-shot ACC (%) | | | | | | | |
| | Avg-acc | 34.5 | 36.1 | 25.5 | 36.1 | 45.0 | 45.5 |
| 1 | Wildfire | 17.2 | 15.1 | 13.3 | 18.9 | 25.5 | 27.5 |
| 2 | Brown spot | 31.2 | 26.5 | 3.9 | 27.6 | 42.9 | 32.5 |
| 3 | Frog eye | 38.8 | 36.1 | 26.0 | 25.7 | 66.2 | 47.3 |
| 4 | Anthracnose | 25.0 | 28.7 | 20.3 | 26.5 | 29.3 | 26.2 |
| 5 | Target spot | 25.0 | 31.2 | 7.5 | 32.3 | 42.2 | 38.7 |
| 6 | Black shank | 63.0 | 62.5 | 20.9 | 50.9 | 48.5 | 55.3 |
| 7 | TMV | 51.5 | 56.6 | 51.2 | 78.5 | 65.4 | 79.7 |
| 8 | CMV | 29.1 | 32.5 | 21.8 | 33.0 | 30.9 | 36.1 |
| 9 | PVY | 22.7 | 23.0 | 30.9 | 23.7 | 23.7 | 25.9 |
| 10 | TSWV | 33.1 | 36.8 | 29.6 | 24.3 | 33.2 | 56.7 |
| 11 | Weather fleck | 57.3 | 61.4 | 59.3 | 81.7 | 64.0 | 65.1 |
| 12 | Sunscald | 24.5 | 28.0 | 20.1 | 33.2 | 48.7 | 41.3 |
| 13 | Genetic abnormality | 27.2 | 25.1 | 21.0 | 46.0 | 42.7 | 54.5 |
| 14 | *Phthorimaea operculella* | 19.4 | 22.5 | 17.5 | 21.6 | 56.4 | 43.5 |
| 15 | Nematodes | 34.3 | 36.6 | 27.3 | 21.8 | 43.8 | 47.0 |
| 16 | Healthy | 53.4 | 54.2 | 37.6 | 32.3 | 56.8 | 50.1 |
| 16-way, 5-shot ACC (%) | | | | | | | |
| | Avg-acc | 51.3 | 53.8 | 37.7 | 49.9 | 66.7 | 64.4 |
| 1 | Wildfire | 25.9 | 20.6 | 20.1 | 37.5 | 47.4 | 44.1 |
| 2 | Brown spot | 42.5 | 40.5 | 4.3 | 42.1 | 63.5 | 50.7 |
| 3 | Frog eye | 54.0 | 52.9 | 28.9 | 32.3 | 82.1 | 59.0 |
| 4 | Anthracnose | 35.1 | 38.3 | 26.3 | 31.7 | 34.4 | 25.5 |
| 5 | Target spot | 41.7 | 48.4 | 8.1 | 45.9 | 63.9 | 57.0 |
| 6 | Black shank | 88.9 | 94.3 | 29.6 | 78.7 | 79.4 | 86.4 |
| 7 | TMV | 77.3 | 80.4 | 77.0 | 95.1 | 88.8 | 93.6 |
| 8 | CMV | 44.1 | 47.6 | 24.4 | 46.9 | 45.8 | 52.1 |
| 9 | PVY | 28.6 | 30.1 | 53.0 | 32.6 | 34.0 | 35.7 |
| 10 | TSWV | 51.8 | 59.3 | 61.9 | 31.0 | 63.6 | 87.2 |
| 11 | Weather fleck | 76.1 | 79.1 | 84.4 | 94.2 | 83.7 | 87.9 |
| 12 | Sunscald | 37.5 | 46.0 | 35.6 | 47.2 | 70.8 | 58.7 |
| 13 | Genetic abnormality | 63.4 | 52.8 | 27.2 | 77.6 | 81.2 | 82.5 |
| 14 | *Phthorimaea operculella* | 32.4 | 38.0 | 24.2 | 28.0 | 80.9 | 69.4 |
| 15 | Nematodes | 51.7 | 57.5 | 44.4 | 26.1 | 64.5 | 63.4 |
| 16 | Healthy | 70.3 | 75.0 | 53.9 | 50.9 | 82.9 | 77.8 |

Mini-100, 100 categories of Mini-Imagenet; PV-38, 38 categories of PV.
Experiment configuration: data (pretraining: 100 categories of Mini-Imagenet, meta-learning: 38 categories of PV, test: raw setting of TLA), Resnet12, 16-way, one-head, lr (pretraining: 0.1, meta-learning: 0.001), epoch (pre-training: 100, meta-learning: 50).

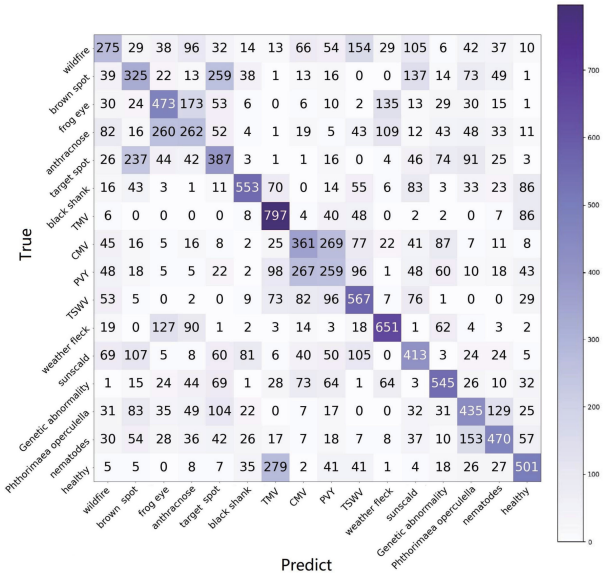



FIGURE 8
The confusion matrix of e15, using the raw setting of TLA and 16-way, 1-shot task.

The confusion matrix shows that TMV, CMV, PVY, and TSWV have a high degree of similarity, which coincides with the property that they all belong to virus infections. Specifically, CMV and PVY both have lightning-like patterns and look similar to each other, as shown in Figure 9D (1). The difference is that PVY has vein necrosis, but CMV does not have this symptom, as shown in Figure 9D (2).

The two pest-trace left symptoms are similar to each other and have irregular brown patterns, as shown in Figure 9E (1). The typical difference is that the trace of *Phthorimaea operculella* has a translucent film, as shown in Figure 9E (2).

Another unexpected result is that the healthy category is highly confused with TMV. TMV is a systemic disease with green islands and chlorotic symptoms that spread throughout the leaves and even the entire plant. Healthy leaves are not completely flat and smooth. In high humidity conditions, such as during the rainy season, the mesophyll grows faster than the veins, resulting in wrinkles. The shadow of wrinkles leads to the mottled pattern, which is similar to the mottled pattern of the green island of TMV.

### 4.4.2 Group 2

In the second group of experiments, Mini-Imagenet is still used in pretraining, PV is used in meta-learning, and processed setting of TLA is used for testing.

In Table 7, e16 is conducted with the processed setting. The results of 16-way, 1-shot, 5-shot, and 10-shot tasks are reported. The accuracy is improved by about 10% in 1-shot and 5-shot tasks, respectively, from 45.5% to 56.5% and from 64.4% to 73.7%. The accuracy of frog eye is improved from 38.8% to 68.3%, the accuracy of anthracnose is improved from 25.0% to 71.5%, and the accuracy of weather fleck is improved from 57.3% to 98.3%, which shows that the effectiveness of segmentation to those diseases with small spots are prominent.

However, even using a processed setting, the accuracy of wildfire is still low at 30.4%. In Figure 10, it is easily found that many diseases affect the identification of wildfires. The reason is that wildfire has many widely varying symptoms, which means that the centroid loses its representativeness. For 1-shot, if the query sample and the support sample happen to be not the same symptom, the support sample loses the meaning of support. For *n*-shot, varying symptoms lead to the meaninglessness of the centroid. In our opinion, the support samples should be highly representative of a certain symptom. Based on this point of view, we classify the samples of wildfires into five categories, as shown in Figure 11. Hence, the 16 categories are enlarged to 20 categories, named as processed setting 20. The results of this setting are listed in Table 7 (e17-A).

As the way increases from 16 to 20, the avg-acc drops to 53.9%. As shown in Figure 12, wildfire 1 has a high ratio to be classified into the rest symptoms of wildfire, which is the same as the rest of the symptoms. Because they are all wildfires, they all have more or less the same characteristics. If a case of wildfire 1 is predicted to be wildfire 2, it is seen as a correct identification because wildfire 1 and wildfire 2 are both wildfires. Therefore, from the perspective of an application, all the cases of wildfire 1 classified from wildfire 2 to wildfire 5 can be counted into wildfire 1 (263 + 187 + 61 + 132 + 99 = 742). The accuracy of wildfire 1 is counted as 74.2%. All the predictions belonging to the subsymptoms can be counted into the super category. Even if the increase in ways will cause a decrease avg-acc, it is worth using this solution from an application perspective.

All five wildfire symptoms can be used to calculate the accuracy of wildfires. The sum of the pink area divided by 5,000 is seen as the accuracy of wildfire. By using this solution, the accuracy of wildfire has been improved from 30.4% to 73.0%, and the avg-acc of 16 categories is improved from 56.5% to 60.7%. The recalculated accuracies are italic numbers listed in the column of e17-B in Table 7. The avg-acc of 20-way, 10-shot achieves 81.8%, which is an acceptable result.

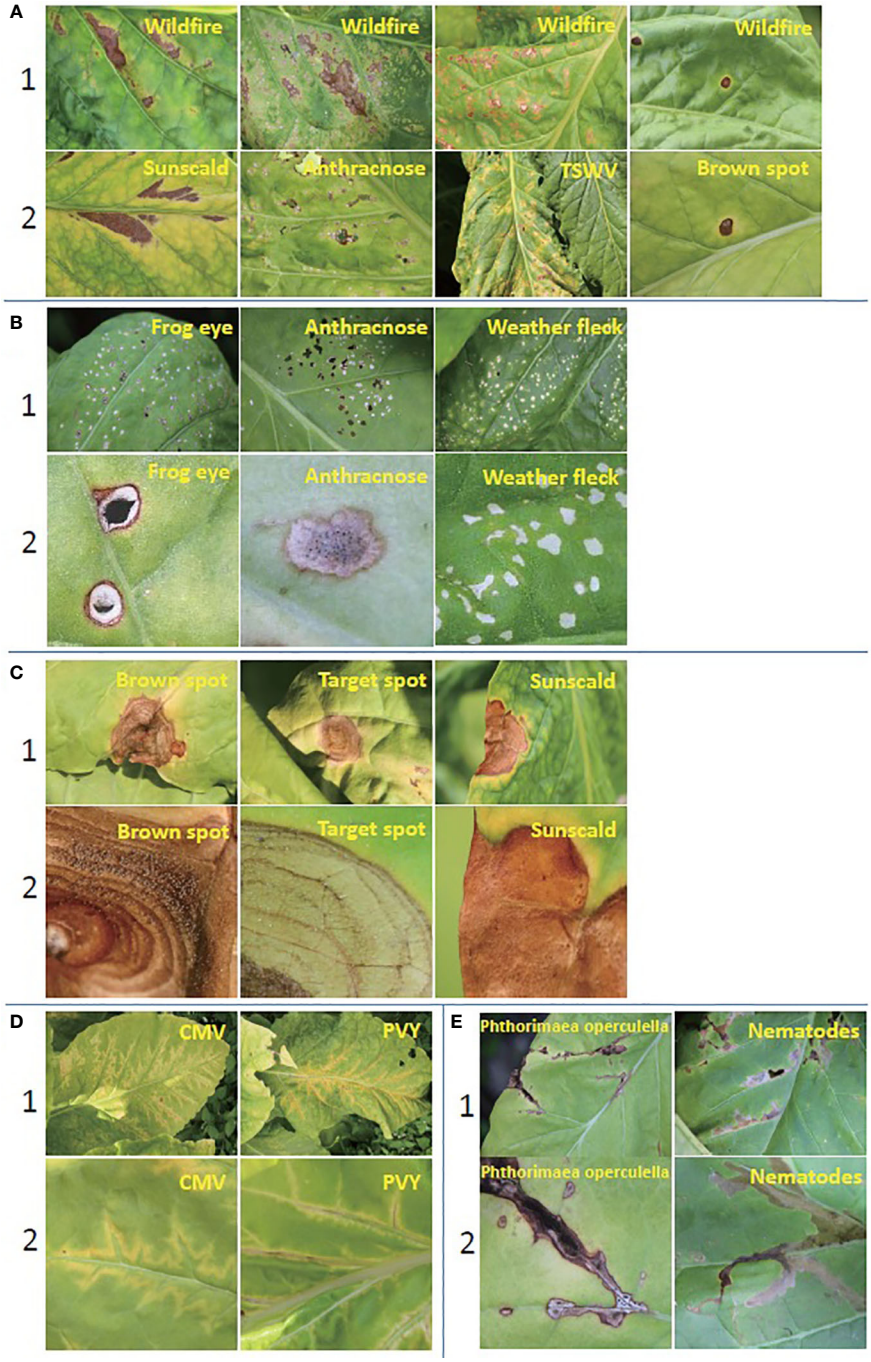



FIGURE 9
The similar symptoms of TLA. **(A)** Four diseases similar to wildfire. **(B)** The group of frog eye, anthracnose, and weather fleck, as well as the details of the lesion. **(C)** Two pest-trace left of similar symptoms. **(D)** The group of brown spots, target spots, and sunscalds, as well as the details of the lesion. **(E)** The group of CMV and PVY, and the details of the lesion.

# 5 Discussion

## 5.1 Motivation

Although FSL has raised great concern in plant disease identification, weak feature representation and demanding requirement of cross-domain are two pressing issues that hinder its application. These issues motivate us to make some efforts toward the method. Meanwhile, given that tobacco is a valuable crop and related studies are rare, it is worth studying and filling up this gap.

## 5.2 Our work and contribution

From the perspective of the method, we proposed FREN. In this network, the feature representation is enhanced by using double-pooling for vectorization, removing the skewness of distribution by

TABLE 7 The ACC (%) of processed setting of TLA using FREN.

| ID | Category | e16 (16-way) | | | e17-A (20-way) | | | e17-B (20-way) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 1-shot | 5-shot | 10-shot | 1-shot | 5-shot | 10-shot | 1-shot | 5-shot | 10-shot |
| | Avg-acc | 56.5 | 73.7 | 79.0 | 53.9 | 73.3 | 77.6 | *60.7* | *78.2* | *81.8* |
| | Wildfire 1 | –[a] | –[a] | –[a] | 26.3 | 41.1 | 51.0 | *74.2* | *82.8* | *85.5* |
| | Wildfire 2 | –[a] | –[a] | –[a] | 33.9 | 54.2 | 63.0 | *74.2* | *91.6* | *92.3* |
| | Wildfire 3 | –[a] | –[a] | –[a] | 44.2 | 68.7 | 74.3 | *52.9* | *74.4* | *81.6* |
| | Wildfire 4 | –[a] | –[a] | –[a] | 75.1 | 93.2 | 93.1 | *90.7* | *98.2* | *98.9* |
| | Wildfire 5 | –[a] | –[a] | –[a] | 48.4 | 81.4 | 87.4 | *73.1* | *89.8* | *94.8* |
| 1 | Wildfire | 30.4 | 48.8 | 58.7 | –[a] | –[a] | –[a] | *73.0* | *87.4* | *90.6* |
| 2 | Brown spot | 47.3 | 63.5 | 71.3 | 47.9 | 66.9 | 71.9 | 47.9 | 66.9 | 71.9 |
| 3 | Frog eye | 68.3 | 84.4 | 89.4 | 63.4 | 79.2 | 80.7 | 63.4 | 79.2 | 80.7 |
| 4 | Anthracnose | 71.5 | 94.2 | 97.8 | 69.2 | 94.8 | 95.6 | 69.2 | 94.8 | 95.6 |
| 5 | Target spot | 45.6 | 64.8 | 69.6 | 44.2 | 60.9 | 69.9 | 44.2 | 60.9 | 69.9 |
| 6 | Black shank | 42.4 | 64.9 | 79.1 | 48.6 | 81.3 | 88.4 | 48.6 | 81.3 | 88.4 |
| 7 | TMV | 90.5 | 96.5 | 97.3 | 91.1 | 96.2 | 96.1 | 91.1 | 96.2 | 96.1 |
| 8 | CMV | 42.7 | 58.9 | 62.4 | 38.4 | 54.7 | 63.6 | 38.4 | 54.7 | 63.6 |
| 9 | PVY | 46.7 | 60.4 | 66.3 | 46.9 | 61.3 | 66.6 | 46.9 | 61.3 | 66.6 |
| 10 | TSWV | 40.8 | 54.0 | 63.3 | 39.7 | 48.4 | 48.6 | 39.7 | 48.4 | 48.6 |
| 11 | Weather fleck | 98.3 | 97.2 | 99.1 | 84.6 | 97.2 | 97.4 | 84.6 | 97.2 | 97.4 |
| 12 | Sunscald | 39.2 | 60.0 | 66.4 | 34.2 | 53.9 | 57.3 | 34.2 | 53.9 | 57.3 |
| 13 | Genetic abnormality | 74.2 | 93.6 | 94.6 | 75.0 | 94.8 | 95.6 | 75.0 | 94.8 | 95.6 |
| 14 | *Phthorimaea operculella* | 53.7 | 81.2 | 86.1 | 55.4 | 81.2 | 86.4 | 55.4 | 81.2 | 86.4 |
| 15 | Nematodes | 55.0 | 73.5 | 77.7 | 51.8 | 72.8 | 79.3 | 51.8 | 72.8 | 79.3 |
| 16 | Healthy | 57.9 | 82.8 | 85.6 | 59.1 | 83.4 | 86.4 | 59.1 | 83.4 | 86.4 |

[a]Not applicable, italic font indicates recalculated results.
e16 uses processed setting. e17-A uses processed setting 20. e17-B is the same experiment as e17-A but recalculated.
Experiment configuration: data (pretraining: 100 categories of Mini-Imagenet; meta-learning: 38 categories of PV; test: processed setting of TLA), Resnet12, 16-way (e16), 20-way (e17), one-head, lr (pretraining: 0.1; meta-learning: 0.001), epoch (pretraining: 100, meta-learning: 50).

PT, and task-adapting by self-attention. In experiments, we compared FREN with related works on the public dataset PV to show its superiority.

In addition, we created the dataset TLA, which includes 16 categories, two settings, and 1,430 images in total. In experiments, we analyzed the characteristics of different diseases and the factors affecting their performance. From the perspective of application, we proposed some solutions to promote FSL in practical scenarios. In brief, our contributions can be refined as follows:

1. The network FREN for FSL is proposed.
2. The dataset TLA has been created. It is published for researchers who are interested in it.
3. The superiority of FREN is verified, the identification of the 16 categories of TLA is conducted and discussed, and some solutions that can be used for other plants or for any other classification task are proposed.

## 5.3 Findings

### 5.3.1 About method

1. GAP is the most frequently used method to downsample a set of feature maps to a feature vector. Actually, except for the average value, a matrix has many feature values, such as the max value, min value, standard deviation value, etc. In our experiments, we tried many combinations, such as tmax only, max+avg+min, max+min, avg+(max-min), avg+std, and avg+std (after GC to each feature map), etc. Finally, we have found that using only GMP produces the closest results to GAP, while using both GAP and GMP achieves the best results. This means that the mean and max values are the key features of the feature map. In those special conditions that suffer from limited feature representation,

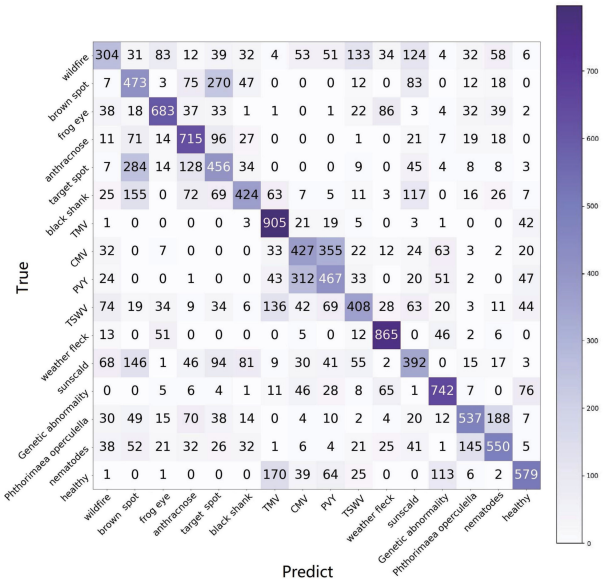



FIGURE 10
The confusion matrix of e16, processed setting of TLA and 16-way, 1-shot task.

such as the few-shot condition, this is an easy and worthwhile approach to enhance feature representation.

2. Our design is called task adaptation because FREN implements adaptation to the entire task, including the support set and query set, in training and testing and also keeps the independence of context of the support set. We argue that the use of self-attention in Ye et al. (2020) is not reasonable. In their work, the support feature vector of a category is concatenated with the group of query feature vectors of the same category and then fed into self-attention to calculate a part of the loss in training. This means that the categories of the query samples have been known and leaked even if it is in training.
3. In our previous work (Lin et al., 2022a), we proposed a Gaussian-like calibration module to remove the skewness of distribution of frequency feature representation. In this work, this module is verified to be still efficient in the spatial domain.
4. No matter whether in application or in academic research, cross-domain means the domain in training is open for a certain target domain. In this work, we found that the baseline and FEAT still follow this rule but FREN shows different performance. In 1-shot tasks, the performance of using a general dataset and a similar dataset gets very close, and even in 5-shot tasks, using a general dataset is better than using a similar dataset. This indicates that FREN desensitizes from data used in training. In other words, FREN has stronger generalization crossing between different domains, and the choice of data used in training can be more free.

The opposite side of generalization is overfitting, where the model excessively learns the training data and fails to generalize well to unseen examples. Some regularization techniques, such as L1 or L2 regularization, batch normalization, etc., are employed to help prevent overfitting by adding constraints or introducing noise during training. For example, L1 and L2 regularization, also known as weight decay, penalizes large weights by adding the absolute values of weights or the squared values of weights to the loss. The smaller and more evenly distributed weights are encouraged to be prioritized. The batch normalization makes the model less sensitive to the scale and distribution of the inputs.

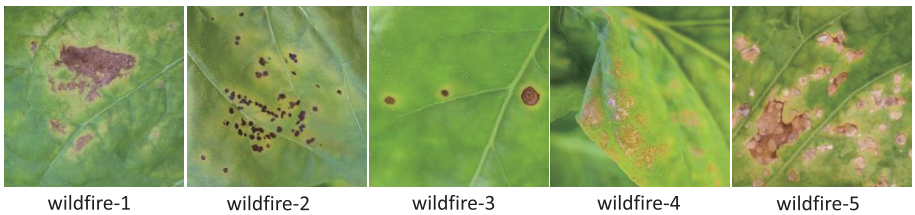



FIGURE 11
The five different symptoms of wildfire.

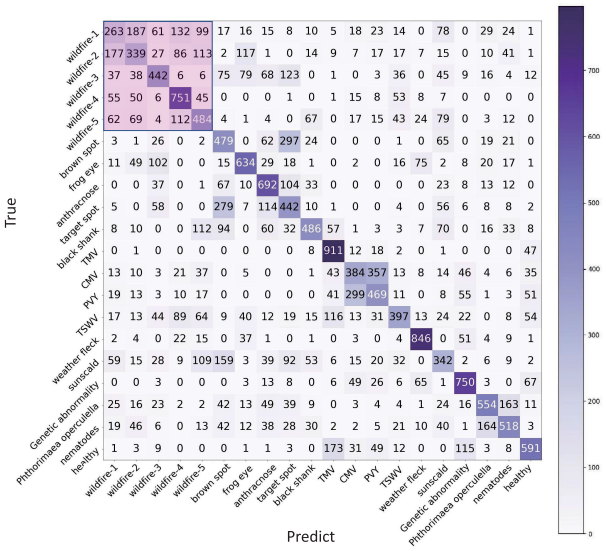



FIGURE 12
The confusion matrix of e17-A, processed setting 20 of TLA and 20-way, 1-shot task.

The GC module can also be seen as a kind of regularization. Different distributions are harmful for the deep learning network to learn patterns. By using the GC module, firstly, the data are uniformed into the same distribution. For a metric-based method, it is better and more balanced to compare the distance of vectors that follow the same distribution. The principle of PT is the same as batch normalization which can reduce the sensitivity of the model to different distributions. Also, for the data coming from different domains, it can shrink the big gap caused by the differences between different domains. In the GC module, the Euclidean normalization and the centralization can be seen as a kind of regularization. They uniformize the values into the same range, forbidding the influence of large values dominating the others. The GC module obviously greatly contributes to the improvement of generalization.

The added GMP is used for enriching the feature representation of samples. It can be seen as adding a dimension to the feature. More diverse features can reduce the mapping dependency of the features only to the results of GAP. Hence, the GMP also contributes to improving generalization.

The task-adaptation module is used to discuss the relationship between the categories. It is more inclined to extract the relationships of the different categories instead of extracting the solid features of a specific category. Therefore, it is more flexible to generalize to the categories and domains that are different from training. The underlying task adaptation makes the method more of a higher-level abstraction than a specific classification task since it can better cope with the problem of generalization.

### 5.3.2 About application

Some existing works studied the attention of the CNN and used visualization methods to show the attention areas. Even for images of conditions, the area of attention is sometimes not focused on the lesion, let alone on the images taken in a field with complicated surroundings. Multidisease also leads to the confusion in the attention area. A deeper network is required if one expects to automatically focus on the lesion through the network. However, the paradox is that smaller-size networks (e.g., Resnet12) perform better than deeper networks in FSL, which has been verified by Lin et al. (2022b). Therefore, finding the lesion automatically is not realistic right now. That is the reason we conduct segmentation of the raw images. This may be questioned as it is a manual operation. In our point of view, the user, as part of the interactive application, should clearly indicate the areas of interest. The involvement of users can greatly reduce the complexity of the system and increase its accuracy. Most cell phones have a basic function to edit images nowadays, and the segmentation is easy to accomplish.

When creating TLA, we found that symptoms are complex and variable. Many factors, such as different phases in the life cycle, species (the tobacco plant has many varieties), weather conditions (sunlight, moisture, air, etc.), location (longitude, latitude), nutritional conditions (nitrogen, potash, etc.), etc., can lead to different symptoms of the same disease. By using traditional deep learning methods, this issue can be fixed by using large-scale data in training. However, this problem always leads to low accuracy in FSL. For these cases, 2. we propose creating categories not only by disease but also by symptom. Users do not care about the avg-acc of all subcategories. The accuracy of super-categories is improved and provided to users by the user interface. In this work, we use wildfire as an example. Not just wildfire, this solution can be used for other diseases having this problem and can also apply to any other plants. Especially for the concerned problem of identifying the phases of disease, this solution is worth trying.

### 5.4 Limitations

1. Although plant experts are involved in the labeling work of TLA, incorrect labels may exist because only images of the leaves were provided to the experts. However, sometimes the other information, such as the images of the back side of the leaf, stem, and root, or the location of the leaf (top leaf or the bottom leaf), etc., is the basis for judgment, which is not provided to experts.
2. All the images were taken during the mature period of tobacco; the other phases of the tobacco life cycle are not discussed in this work. The phases of the disease are also not classified in TLA. In fact, many diseases show the same symptom in the initial infection and then gradually become different. The identification of the phases of disease is still a challenging problem.
3. Multidisease classification is not involved in this work. When creating a dataset, we found that more than half of the images are multidisease. The plant is easily infected by other diseases after the first infection because the immune system is attacked and becomes weak. Hence, multidisease commonly occurs, especially in the mature period. Using classification methods for multidisease is not reasonable for two reasons: (1) feeding an image containing several diseases may make the network confused and obtain meaningless results; and (2) multiple diseases generate so many combinations that it is hard to collect samples. For example, three diseases can generate seven combinations. Hence, we think it is not reasonable to solve the problem by using classification methods.
4. When taking photos in the field, we found that the sunlight affected the quality greatly. Many details disappear in the strong sunlight. Hence, in sunshine weather, we use umbrellas to block out the strong light without preprocessing the light.

## 6 Conclusion and recommendation

In this work, we create a dataset TLA and propose FREN for image-based tobacco abnormality identification. For the data, TLA includes 16 categories and 1,430 images, covers 10 infection diseases, three noninfection diseases, two pest-trace left, and a healthy class. For the method, we argue that a good embedding not only depends on the instance embedding of the sample itself but also relies on the internal relationship of the support set in FSL. Therefore, we proposed the FREN, which improves the embedding by integrating the GMP in instance embedding and self-attention in task adaptation. By using PV, we demonstrated the superiority of our FREN. The accuracy is achieved at 66.04%, which has been lifted 8.6% from the baseline method and 4.49% from FEAT in a 5-way, 1-shot task of tomato identification. On the TLA, the FREN achieves 45.5% and 56.5% for the two settings, respectively. Meanwhile, we proposed a multisymptom solution from the perspective of application. By using this solution, the best accuracy can be lifted to 60.7% in 16-way, 1-shot tasks and achieved at 81.8% in 16-way, 10-shot tasks. The solution can be used for other plants. The results of our method give us the confidence to advance few-shot learning into applications, although it still has a lot of room for improvement in the future.

For the research direction in the future, we think the identification can be executed in a tobacco field environment. Different from the identification in the lab, the raw image needs various preprocessing, such as background processing, light processing, etc. Users are more concerned with disease identification in the early stages. So, the phase identification of diseases is an important research direction. For multidisease identification, the classification method is not an optimal choice. Semantic segmentation is a good solution and worthy of study.

## Data availability statement

The original contributions presented in the study are publicly available. This data can be found here: https://drive.google.com/drive/folders/1Qn5UjATDaDpRoF1dCTdp62tlnAXJv0MF?usp=sharing.

## Author contributions

HL: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. ZQ: Conceptualization, Funding acquisition, Methodology, Writing – review & editing. RT: Conceptualization, Writing – review & editing. ST: Conceptualization, Writing – review & editing. GP: Conceptualization, Writing – review & editing, Methodology.

## Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work was funded by the Natural Science Foundation of China (grant number 12163004), the Yunnan Fundamental Research Projects (CN) (grant number 202101BD070001-053, 202401AS070009), the Yunnan Fundamental Research Program of Agricultural Special Project (CN) (grant number 202301BD070001-008), and the Fundamental Research Projects of Yunnan Provincial Department of Education (CN) (grant number 2022J0496).

## Acknowledgments

This work was supported in part by the Macao Polytechnic University-Edge Sensing and Computing: Enabling Human-centric (Sustainable) Smart Cities (RP/ESCA-01/2020). The labeling work

was supported by Dr. Lin Liu from the College of Tobacco Science, Yunnan Agricultural University, China.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

Afifi, A., Alhumam, A., and Abdelwahab, A. (2021). Convolutional neural network for automatic identification of plant diseases with limited data. *Plants* 10, 28. doi: 10.3390/plants10010028

Argüeso, D., Picon, A., Irusta, U., Medela, A., San-Emeterio, M. G., Bereciartua, A., et al. (2020). Few-shot learning approach for plant disease classification using images taken in the field. *Comput. Electron. Agric.* 175, 105542. doi: 10.1016/j.compag.2020.105542

Baik, S., Choi, J., Kim, H., Cho, D., Min, J., and Lee, K. M. (2021). "Meta-learning with task-adaptive loss function for few-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. (Los Angeles, CA, U.S.: Computer Vision Foundation), 9465–9474.

Chen, Y., Liu, Z., Xu, H., Darrell, T., and Wang, X. (2020). "Meta-baseline: Exploring simple meta-learning for few-shot learning," in *Proceedings of the IEEE/CVF international conference on computer vision*, 9062–9071. doi: 10.48550/arXiv.2003.04390

Han, J., Kamber, M., and Pei, J. (2012). "2 - getting to know your data," in *Data mining (Third edition), 3rd ed*. Eds. J. Han, M. Kamber and J. Pei (Morgan Kaufmann, Boston), 39–82. doi: 10.1016/B978-0-12-381479-1.00002-2

Hu, Y., Gripon, V., and Pateux, S. (2021). "Leveraging the feature distribution in transfer-based few-shot learning," in *International Conference on Artificial Neural Networks*. (New York City, U.S.: Springer), 487–499.

Huang, Y., Chang, F., Tao, Y., Zhao, Y., Ma, L., and Su, H. (2022). Few-shot learning based on attn-cutmix and task-adaptive transformer for the recognition of cotton growth state. *Comput. Electron. Agric.* 202, 107406. doi: 10.1016/j.compag.2022.107406

Hughes, D., and Salathé, M. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060*. doi: 10.48550/arXiv.1511.08060

ITAS, Y. (2020). *Yunnan Institute of Tobacco Agricultural Science, Field planting techniques for highquality plue-cured tobacco production in Yunnan* (Beijing: China science publishing & media Ltd).

Jadon, S. (2020). "Ssm-net for plants disease identification in low data regime," in *2020 IEEE/ITU International Conference on Artificial Intelligence for Good (AI4G) (IEEE)*. (Piscataway, NJ, U.S.: IEEE), 158–163. doi: 10.1109/AI4G50087.2020.9311073

Kahl, V. F. S., Dhillon, V., Fenech, M., de Souza, M. R., da Silva, F. N., Marroni, N. A. P., et al. (2018). Occupational exposure to pesticides in tobacco fields: the integrated evaluation of nutritional intake and susceptibility on genomic and epigenetic instability. *Oxid. Med. Cell. Longevity* 2018. doi: 10.1155/2018/7017423

Kamilaris, A., and Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Comput. Electron. Agric.* 147, 70–90. doi: 10.1016/j.compag.2018.02.016

Koch, G., Zemel, R., and Salakhutdinov, R. (2015). "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, Lille, Vol. 2.

Li, Y., and Chao, X. (2021). Semi-supervised few-shot learning approach for plant diseases recognition. *Plant Methods* 17, 1–10. doi: 10.1186/s13007-021-00770-1

Li, W., Xu, J., Huo, J., Wang, L., Gao, Y., and Luo, J. (2019). "Distribution consistency based covariance metric networks for few-shot learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, (AAAI Press). Vol. 33. 8642–8649. doi: 10.1609/aaai.v33i01.33018642

Li, Y., and Yang, J. (2021). Meta-learning baselines and database for few-shot classification in agriculture. *Comput. Electron. Agric.* 182, 106055. doi: 10.1016/j.compag.2021.106055

Li, X., Yang, X., Ma, Z., and Xue, J.-H. (2021). Deep metric learning for few-shot image classification: A selective review. *arXiv preprint arXiv:2105.08149*. doi: 10.48550/arXiv.2105.08149

Lichtenstein, M., Sattigeri, P., Feris, R., Giryes, R., and Karlinsky, L. (2020). "Tafssl: Task-adaptive feature sub-space learning for few-shot classification," in *European Conference on Computer Vision*, (Heidelberg, Germany: Springer), 522–539.

Lin, H., Tse, R., Tang, S.-K., Qiang, Z., and Pau, G. (2022a). Few-shot learning for plant-disease recognition in the frequency domain. *Plants* 11, 2814. doi: 10.3390/plants11212814

Lin, H., Tse, R., Tang, S.-K., Qiang, Z.-p., and Pau, G. (2022b). Few-shot learning approach with multi-scale feature fusion and attention for plant disease recognition. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.907916

Nuthalapati, S. V., and Tunga, A. (2021). "Multi-domain few-shot learning and dataset for agricultural applications," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. (Los Angeles, CA, U.S.: Computer Vision Foundation), 1399–1408. doi: 10.48550/arXiv.2109.09952

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Adv. Neural Inf. Process. Syst.* 28.

Snell, J., Swersky, K., and Zemel, R. S. (2017). Prototypical networks for few-shot learning. *Adv. Neural Inf. Process Syst.* 30. doi: 10.48550/arXiv.1703.05175

Song, L., Shang, X., Yang, C., and Sun, M. (2022). Attribute-guided multiple instance hashing network for cross-modal zero-shot hashing. *IEEE Trans. Multimedia* 25, 5305-5318. doi: 10.1109/TMM.2022.3190222

Strange, R. N., and Scott, P. R. (2005). Plant disease: a threat to global food security. *Annu. Rev. Phytopathol.* 43, 83–116. doi: 10.1146/annurev.phyto.43.113004.133839

Sun, Q., Liu, Y., Chua, T.-S., and Schiele, B. (2019). "Meta-transfer learning for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (Los Angeles, CA, U.S.: Computer Vision Foundation). doi: 10.1109/CVPR41558.2019

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. (2018). "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. (Los Angeles, CA, U.S.: Computer Vision Foundation), 1199–1208. doi: 10.48550/arXiv.1711.06025

Tukey, J. W. (1977). *Exploratory data analysis* Vol. 2 (New York: Reading, MA).

Van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-sne. *J. Mach. Learn. Res.* 9, 2579-2605. Available at: http://jmlr.org/papers/v9/vandermaaten08a.html.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Adv. Neural Inf. Process. Syst.* 30. Available at: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., and Wierstra, D. (2016). Matching networks for one shot learning. *Adv. Neural Inf. Process. Syst.* 29, 3630–3638. doi: 10.48550/arXiv.1606.04080

Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)* 53, 1–34. doi: 10.1145/3386252

World Health Organization. (2017). *Tobacco and its environmental impact: an overview* (Switzerland: World Health Organization).

Yang, J., Guo, X., Li, Y., Marinello, F., Ercisli, S., and Zhang, Z. (2022). A survey of few-shot learning in smart agriculture: developments, applications, and challenges. *Plant Methods* 18, 1–12. doi: 10.1186/s13007-022-00866-2

Ye, H.-J., Hu, H., Zhan, D.-C., and Sha, F. (2020). "Few-shot learning via embedding adaptation with set-to-set functions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (Los Angeles, CA, U.S.: Computer Vision Foundation), 8808–8817.

Yoon, S. W., Seo, J., and Moon, J. (2019). "Tapnet: Neural network augmented with task-adaptive projection for few-shot learning," in *International Conference on Machine Learning (PMLR)*. (Proceedings of Machine Learning Research), 7115–7123.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. (2017). Deep sets. *Adv. Neural Inf. Process. Syst.* 30. Available at: https://proceedings.neurips.cc/paper_files/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf.

Zhang, S., Tong, H., Xu, J., and Maciejewski, R. (2019). Graph convolutional networks: a comprehensive review. *Comput. Soc. Networks* 6, 1–23. doi: 10.1186/s40649-019-0069-y

Zhao, J., Yang, Y., Lin, X., Yang, J., and He, L. (2021). "Looking wider for better adaptive representation in few-shot learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, (AAAI Press) Vol. 35. 10981–10989. doi: 10.1609/aaai.v35i12.17311

Zhong, F., Chen, Z., Zhang, Y., and Xia, F. (2020). Zero-and few-shot learning for diseases recognition of citrus aurantium l. using conditional adversarial autoencoders. *Comput. Electron. Agric.* 179, 105828. doi: 10.1016/j.compag.2020.105828

# Frontiers in
# Plant Science

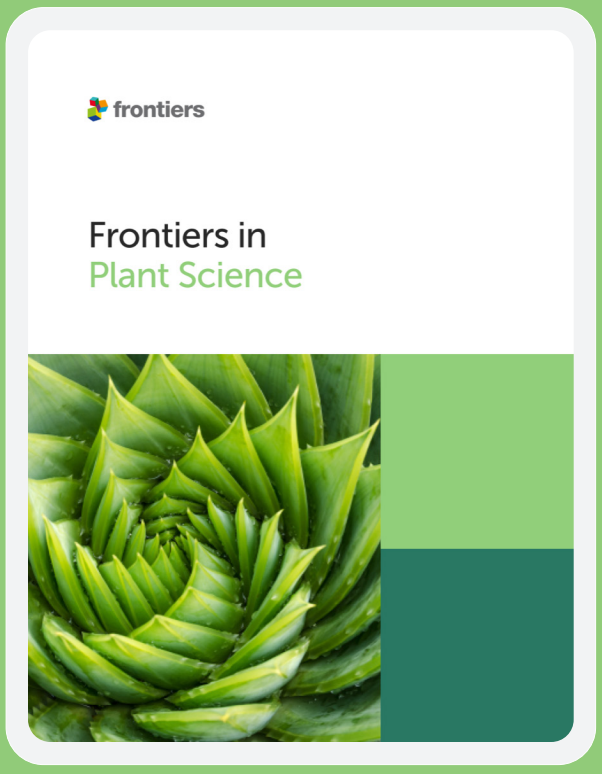
**Cultivates the science of plant biology and its applications**

The most cited plant science journal, which advances our understanding of plant biology for sustainable food security, functional ecosystems and human health.

---

## Discover the latest Research Topics

See more →




**Frontiers**

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

**Contact us**

+41 (0)21 510 17 00
frontiersin.org/about/contact