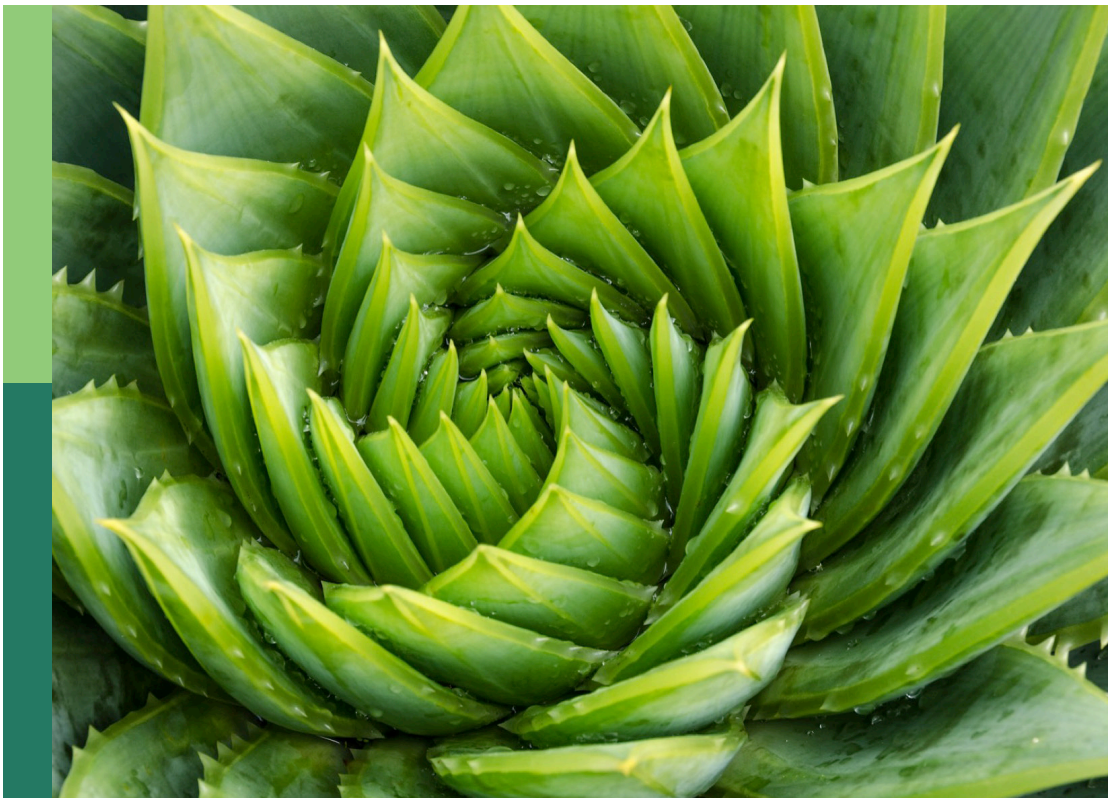
frontiers | Research Topics

Harnessing machine learning to decode plant-microbiome dynamics for sustainable agriculture

Edited by
Eman Mohammad Khalaf, Mohsen Yoosefzadeh Najafabadi, Mohamed Mysara and Ahmed M. El-Baz

Published in
Frontiers in Plant Science
Frontiers in Microbiomes



ISSN 1664-8714
ISBN 978-2-8325-6459-2
DOI 10.3389/978-2-8325-6459-2

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

Harnessing machine learning to decode plant-microbiome dynamics for sustainable agriculture

Topic editors

Eman Mohammad Khalaf — Damanhour University, Egypt

Mohsen Yoosefzadeh Najafabadi — University of Guelph, Canada

Mohamed Mysara — Nile University, Egypt

Ahmed M. El-Baz — Delta University for Science and Technology, Egypt

Citation

Khalaf, E. M., Yoosefzadeh Najafabadi, M., Mysara, M., El-Baz, A. M., eds. (2025). *Harnessing machine learning to decode plant-microbiome dynamics for sustainable agriculture*. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-8325-6459-2

Table of contents

frontiers | Frontiers in Microbiomes

TYPE Editorial
PUBLISHED 03 June 2025
DOI 10.3389/frmbi.2025.1602938

OPEN ACCESS

EDITED AND REVIEWED BY
Robert Beiko,
Dalhousie University, Canada

*CORRESPONDENCE
Mohsen Yoosefzadeh Najafabadi
myoosefz@uoguelph.ca
Eman M. Khalaf
eimanpharmacist@gmail.com

RECEIVED 30 March 2025
ACCEPTED 21 May 2025
PUBLISHED 03 June 2025

CITATION
Yoosefzadeh Najafabadi M, Khalaf EM, Mysara M and El-Baz AM (2025) Editorial: Harnessing machine learning to decode plant-microbiome dynamics for sustainable agriculture.
Front. Microbiomes 4:1602938.
doi: 10.3389/frmbi.2025.1602938

Editorial: Harnessing machine learning to decode plant-microbiome dynamics for sustainable agriculture

Mohsen Yoosefzadeh Najafabadi[1]*, Eman M. Khalaf[1,2]*, Mohamed Mysara[3] and Ahmed M. El-Baz[4]

[1]Department of Plant Agriculture, University of Guelph, Guelph, ON, Canada, [2]Department of Microbiology and Immunology, Faculty of Pharmacy, Damanhour University, Damanhour, Egypt, [3]Bioinformatics Group, Center for Informatics Science, School of Information Technology and Computer Science, Nile University, Giza, Egypt, [4]Department of Microbiology and Immunology, Faculty of Pharmacy, Delta University for Science and Technology, Gamasa, Egypt

KEYWORDS

crop disease detection, crop yield improvement, machine learning algorithms, plant microbiome, phenotypic traits

Editorial on the Research Topic

Harnessing machine learning to decode plant-microbiome dynamics for sustainable agriculture

The world's growing population of nine billion people is facing a severe global food insecurity crisis, especially in low and middle-income countries (Hong et al., 2022). Improving crop yield and productivity through structured breeding programs is a key strategy to address this issue (Yoosefzadeh-Najafabadi et al., 2024). Plants and microbes have evolved intricate relationships over millennia, providing benefits such as enhanced growth, improved nutrient uptake, and increased stress tolerance to plants (Trivedi et al., 2022). In recent years, research has focused on the interplay between the plant microbiome and phenotype to enhance breeding programs (Nerva et al., 2022; Araujo et al., 2024; Batool et al., 2024).

Traditional analysis methods struggle to handle data from high-throughput technologies such as meta-genomics, meta-transcriptomics, and meta-proteomics (Yoosefzadeh Najafabadi and Torkamaneh, 2025), leading to a lack of understanding of how the microbiome influences plant traits (Trivedi et al., 2022). Advanced data analysis techniques have been developed to integrate and analyze data from multiple omics sources effectively (Trivedi et al., 2022). To harness the potential of plant microbiomes, researchers are increasingly turning to machine learning, a subset of artificial intelligence that enables computers to learn from data and make predictions (De Souza et al., 2020). Deep-learning models, a powerful type of machine learning, are particularly effective for analyzing complex biological data. These models are built from layers of interconnected nodes that process input data, such as microbial DNA sequences or plant images, to identify patterns and relationships. Developers must make critical decisions when designing these models, such as choosing the number and type of layers, selecting the data features to focus on (e.g., specific microbial traits), and determining how the model learns from errors (Zhou and Gallins, 2019). These choices depend on the specific problem, such as detecting crop

diseases or predicting yield, and are guided by the need for accuracy, computational efficiency, and applicability to real-world farming conditions (Zhou and Gallins, 2019).

The development of a machine vision-based method using an enhanced YOLOv5s model for grading individual peanut pod rot, which is a major plant disease affecting peanut production were investigated in a recent paper published by (Liu et al.) YOLO is a real-time object detection algorithm known for its speed and efficiency. Unlike traditional methods that repurpose classifiers or localizers to perform detection, YOLO frames object detection as a single regression problem, directly predicting bounding boxes and class probabilities from full images in one evaluation. This model, which relies on deep-learning principles to process images, incorporates a Shuffle Attention module to focus on key visual features and replaces the loss function CIoU with EIoU to improve accuracy in distinguishing non-rotted and rotten peanuts in complex backgrounds. The study also highlighted the potential for future research to enhance prediction performance for different peanut varieties and to consider factors like rotten kernel rate for better yield estimation. In another study by Pandiyaraju et al., the possibility of using a machine vision-based approach for grading individual peanut pod rot using an improved YOLOv5s algorithm were investigated. The study addresses the challenges of visually identifying and classifying peanut pod rot by introducing a Shuffle Attention module to enhance feature representation and accuracy in complex backgrounds. The proposed model demonstrated high recognition rates for non-rotted and rotten peanuts, offering a promising solution for automated grading of peanut pod rot, providing advancements in disease resistance evaluation and germplasm selection in peanut breeding. Another use of YOLO algorithms was reported by Wang et al. where they enhanced the identification of potato seedlings in drone-acquired images by introducing a new lightweight model named VBGS-YOLOv8n. By utilizing a modified version of YOLOv8n with a lighter backbone network and incorporating improvements such as a bidirectional feature pyramid network and GSConv and Slim-neck designs, the model achieves high precision and detection performance.

Precise identification and enumeration of flax plant organs play a vital role in acquiring key phenotypic data necessary for selecting and managing flax varieties. In research conducted by Kai et al., a Flax-YOLOv5 model is presented to extract phenotypic information from flax plants. By extending the YOLOv5x network with the BiFormer module, which integrates bi-directional encoders and converters to focus on essential features adaptively, the model's computational efficiency is enhanced. Zhang et al. introduced a novel method for detecting small target cotton bolls in cotton fields using unmanned aerial vehicle (UAV) imagery. By employing the YOLO SSPD model, which integrates space-to-depth convolution and a small target detector head, the researchers achieved significant improvements in boll detection accuracy on UAV imagery. The model demonstrated high precision and efficiency in detecting cotton bolls, supporting the cotton production process and enhancing reliability in yield estimates. In another research conducted by Tang et al. they tried to overcome the issues related to low detection accuracy and limited applicability across different ripeness levels and varieties of large non-green-ripe citrus fruits in complex environments. The study introduces YOLOC-tiny, a precise and lightweight model based on YOLOv7 that leverages EfficientNet-B0 as the feature extraction backbone. To enhance detection performance, a convolutional block attention module (CBAM) is integrated into the aggregation network, along with an adaptive intersection over union regression loss function tailored to large non-green-ripe citrus characteristics. Furthermore, a layer-based adaptive magnitude pruning technique is utilized to reduce redundancy in model parameters. In practical applications such as fruit-picking robots, YOLOC-tiny achieves a high accuracy of 92.8% at a swift frame rate of 59 frames per second. (Wang et al.) also introduced an improved target detection and pose estimation model called PAE-YOLO for identifying Xiaomila fruits in complex farmland environments. The model combines an EMA attention mechanism and a DCNv3 deformable convolution module to enhance feature extraction capability and reduce computational complexity. Experimental results show that the PAE-YOLO model outperforms other classic detection models in terms of accuracy, model size, and computational efficiency. The model achieved an average mean accuracy of 88.8% and a F1 score of 83.2%, with improved performance in target detection and posture estimation.

Efficiently detecting tomatoes in complex environments is important for automating tomato harvesting. The proposed S-YOLO model by Sun, an enhancement of YOLOv8s, introduces innovations such as a lightweight GSConv_SlimNeck structure, improved α-SimSPPF and β-SIoU algorithms, and an SE attention module to boost detection accuracy and speed (Figure 1). Experimental results show the S-YOLO model achieves 96.60% accuracy and 74.05 FPS, outperforming previous models and making it ideal for use in robotic tomato-picking systems. In a study conducted by Liu et al., the YOLO-SwinTF proposed based on YOLOv7, incorporates Swin Transformer blocks for capturing global visual information and Trident Pyramid Networks for improved feature communication. The model uses Focaler-IoU to adjust focus on sample distribution. Tested on a tomato dataset, it achieved higher recall, precision, F1 score, and AP compared to YOLOv7, showing strong robustness in challenging conditions and improved detection accuracy without compromising speed.

Plant diseases pose a significant threat to global agriculture by negatively impacting crop yield and quality (Yoosefzadeh Najafabadi, 2021). Despite the challenges associated with identifying and classifying these diseases, a new approach leveraging deep learning algorithms and convolutional neural networks (CNNs) has been proposed to accurately detect and categorize leaf diseases in economically important crops such as strawberries, peaches, cherries, and soybeans (Prince et al.). For this aim, a research focuses on categorizing 10 disease classes for these crops, comprising 6 diseased classes and 4 healthy classes, using a CNN-support vector machine (SVM) model (Prince et al.). Various pre-trained models were employed, with the proposed model achieving an average accuracy of 99.09%, outperforming established models like VGG16. The model utilizes Class Activation Maps generated through the Grad-CAM technique to visually illustrate detected diseases and produce heatmaps

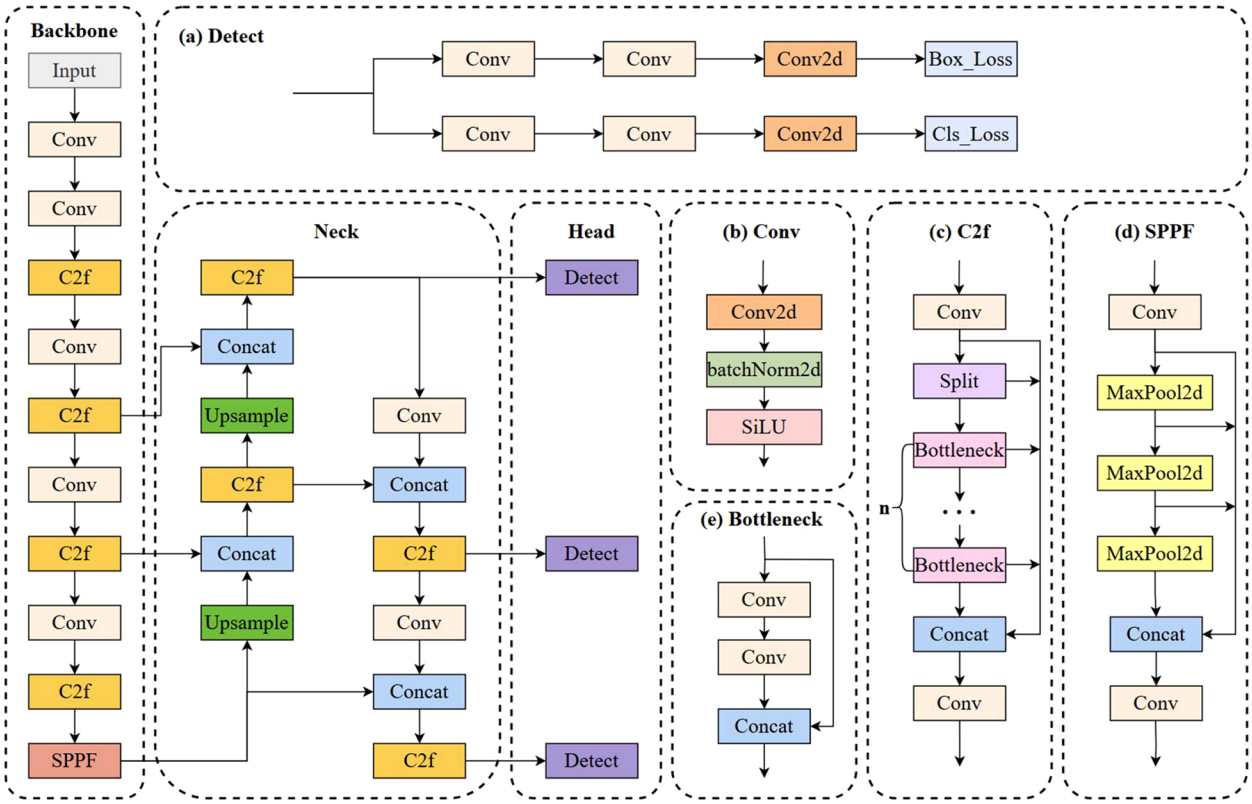


FIGURE 1
YOLOv8 algorithm model. The network consists of three main components: the Backbone, Neck, and Head. It incorporates several functional modules, including the detection head **(a)**, convolutional block **(b)**, cross stage partial with bottlenecks (C2f) **(c)**, spatial pyramid pooling–fast (SPPF) **(d)**, and bottleneck module **(e)**. Source: Reproduced from Sun.

highlighting the areas requiring classification (Prince et al.). The FCHF-DETR model developed by Xin and Li, an enhancement of RT-DETR-R18, addressed the challenges of detecting tomato leaf diseases with FasterNet, Cascaded Group Attention, and HSFPN. Using a dataset of 3147 images, the model achieved high precision and recall while reducing computational demands. In addressing the challenge of identifying tea plant diseases amidst complex backgrounds, the ECA-ResNet50 model improved the ResNet50 architecture by using a multi-layer small convolution kernel strategy and introducing the ECA attention mechanism (Li and Zhao). This enhances feature extraction, achieving a 93.06% accuracy rate, a 3.18% improvement over the original model. The model's strong generalization capabilities indicate its effectiveness in mitigating background interference and precisely recognizing tea disease targets across various plant datasets (Li and Zhao).

Chinese Herbal Medicine (CHM) faces automation challenges in microscopic identification due to traditional method limitations and dataset issues. In a study developed by Zhu et al. introduced a deep learning-based approach, employing segmentation-combination data augmentation and a shallow-deep dual attention module to enhance feature focus. The CHMMI approach achieves high precision and outperforms models such as YOLOv5 and ResNet, offering a robust solution to modernize CHM identification. Jia et al. proposed an enhanced DeepLabv3+ model, named DFMA, incorporating a novel PSPA-ASPP structure for efficient phenotyping analysis. Tested on various datasets, the model achieved high mIoU scores, outperforming existing models. It provides detailed segmentation and precise seedling measurements, offering an automated solution to improve analysis efficiency and overcome traditional method challenges.

Potatoes are known as one of the staple foods globally, and timely detection of foliar diseases is essential for healthy yields. Traditional image classification struggles with inconsistent data, so a new model combines EfficientNet-LITE for feature extraction with KE-SVM Optimization for classification. The method developed by Sangar and Rajasekar refined accuracy by cross-referencing misclassifications, achieving improved accuracy (87.82% for uncontrolled data and 99.54% for controlled data) while maintaining computational efficiency. The model's small size and low floating point operations per second (FLOPs) make it ideal for mobile and edge devices, enhancing its practical use in precision agriculture. Hyperspectral images provide detailed information, important for classifying corn seed varieties with different internal structures. Existing methods struggle with feature extraction from these complex datasets, resulting in low accuracy (Wang et al.). To overcome this, the spectral-spatial attention transformer network (SSATNet) is proposed by Wang et al., which utilizes 3D and 2D convolutions for feature extraction and incorporates a transformer encoder with cross-attention for global perspective refinement. This approach improves classification performance on hyperspectral

corn image datasets, demonstrating its effectiveness over current methods.

Despite the transformative potential of machine learning in analyzing plant-associated microbiomes, several challenges persist. High-quality, standardized datasets are often scarce, particularly for underrepresented crops or regions, limiting model generalizability. Scalability remains a hurdle, as many models require significant computational resources, which may not be accessible to small-scale farmers or researchers in low-resource settings. Additionally, integrating multi-omics data with environmental and phenotypic variables across diverse agricultural systems is complex, often leading to inconsistent predictions. These limitations highlight the need for robust, adaptable frameworks that can accommodate varied data types and practical constraints. Looking forward, promising directions include fostering interdisciplinary collaborations between plant scientists, data scientists, and farmers to develop user-friendly tools that bridge research and application. Advances in computational efficiency, such as lightweight models and edge computing, could democratize access to machine-learning technologies. Furthermore, field-based validations and longitudinal studies are essential to ensure models perform reliably under real-world conditions. By addressing these challenges and leveraging emerging technologies, the scientific community can unlock the full potential of plant microbiomes to enhance crop resilience and global food security.

Author contributions

MN: Conceptualization, Investigation, Project administration, Resources, Writing – original draft, Writing – review & editing. EK: Conceptualization, Investigation, Project administration, Resources, Writing – review & editing. MM: Writing – review & editing. AE: Writing – review & editing.

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.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Publisher's note

References

Araujo, A. S. F., de Araujo Pereira, A. P., da Costa, D. P., de Medeiros, E. V., Araujo, F. F., Sharma, S., et al. (2024). Enhancing plant resilience to pathogens through strategic breeding: Harnessing beneficial bacteria from the rhizosphere for progeny protection. *Rhizosphere* 30, 100890. doi: 10.1016/j.rhisph.2024.100890

Batool, M., Carvalhais, L. C., Fu, B., and Schenk, P. M. (2024). Customized plant microbiome engineering for food security. *Trends Plant Science* 29(4), 482–494. doi: 10.1016/j.tplants.2023.10.012

De Souza, R. S. C., Armanhi, J. S. L., and Arruda, P. (2020). From microbiome to traits: designing synthetic microbial communities for improved crop resiliency. *Front. Plant Sci.* 11, 1179. doi: 10.3389/fpls.2020.01179

Hong, H., Najafabadi, M. Y., Torkamaneh, D., and Rajcan, I. (2022). Identification of quantitative trait loci associated with seed quality traits between Canadian and Ukrainian mega-environments using genome-wide association study. *Theor. Appl. Genet.* 135, 2515–2530. doi: 10.1007/s00122-022-04134-8

Nerva, L., Sandrini, M., Moffa, L., Velasco, R., Balestrini, R., and Chitarra, W. (2022). Breeding toward improved ecological plant–microbiome interactions. *Trends Plant Sci.* 27, 1134–1143. doi: 10.1016/j.tplants.2022.06.004

Trivedi, P., Batista, B. D., Bazany, K. E., and Singh, B. K. (2022). Plant–microbiome interactions under a changing world: responses, consequences and perspectives. *New Phytol.* 234, 1951–1959. doi: 10.1111/nph.v234.6

Yoosefzadeh Najafabadi, M. (2021). *Using advanced proximal sensing and genotyping tools combined with bigdata analysis methods to improve soybean yield* (Canada: University of Guelph).

Yoosefzadeh-Najafabadi, M., Hesami, M., and Eskandari, M. (2024). "Machine learning-enhanced utilization of plant genetic resources," in *Sustainable utilization and conservation of plant genetic diversity* (USA: Springer), 619–639.

Yoosefzadeh Najafabadi, M., and Torkamaneh, D. (2025). Machine learning-enhanced multi-trait genomic prediction for optimizing cannabinoid profiles in cannabis. *Plant J.* 121, e17164. doi: 10.1111/tpj.17164

Zhou, Y.-H., and Gallins, P. (2019). A review and tutorial of machine learning methods for microbiome host trait prediction. *Front. Genet.* 10, 579. doi: 10.3389/fgene.2019.00579

frontiers | Frontiers in Plant Science

TYPE Methods
PUBLISHED 15 April 2024
DOI 10.3389/fpls.2024.1364185

OPEN ACCESS

EDITED BY
Mohsen Yoosefzadeh Najafabadi,
University of Guelph, Canada

REVIEWED BY
Jieli Duan,
South China Agricultural University, China
Jakub Nalepa,
Silesian University of Technology, Poland

*CORRESPONDENCE
Lifeng Liu
liulifeng@hebau.edu.cn
Limin Shao
shaolimin@hebau.edu.cn

†These authors have contributed equally to this work

RECEIVED 01 January 2024
ACCEPTED 29 March 2024
PUBLISHED 15 April 2024

CITATION
Liu Y, Li X, Fan Y, Liu L, Shao L, Yan G, Geng Y and Zhang Y (2024) Classification of peanut pod rot based on improved YOLOv5s. *Front. Plant Sci.* 15:1364185. doi: 10.3389/fpls.2024.1364185

Classification of peanut pod rot based on improved YOLOv5s

Yu Liu[1†], Xiukun Li[1,2†], Yiming Fan[1], Lifeng Liu[1,2*], Limin Shao[1,3*], Geng Yan[1], Yuhong Geng[1] and Yi Zhang[1]

[1]Hebei Agricultural University, Baoding, China, [2]State Key Laboratory of North China Crop Improvement and Regulation, Baoding, China, [3]Technology Innovation Center of Intelligent Agricultural Equipment, Baoding, China

Peanut pod rot is one of the major plant diseases affecting peanut production and quality over China, which causes large productivity losses and is challenging to control. To improve the disease resistance of peanuts, breeding is one significant strategy. Crucial preventative and management measures include grading peanut pod rot and screening high-contributed genes that are highly resistant to pod rot should be carried out. A machine vision-based grading approach for individual cases of peanut pod rot was proposed in this study, which avoids time-consuming, labor-intensive, and inaccurate manual categorization and provides dependable technical assistance for breeding studies and peanut pod rot resistance. The Shuffle Attention module has been added to the YOLOv5s (You Only Look Once version 5 small) feature extraction backbone network to overcome occlusion, overlap, and adhesions in complex backgrounds. Additionally, to reduce missing and false identification of peanut pods, the loss function CIoU (Complete Intersection over Union) was replaced with EIoU (Enhanced Intersection over Union). The recognition results can be further improved by introducing grade classification module, which can read the information from the identified RGB images and output data like numbers of non-rotted and rotten peanut pods, the rotten pod rate, and the pod rot grade. The Precision value of the improved YOLOv5s reached 93.8%, which was 7.8%, 8.4%, and 7.3% higher than YOLOv5s, YOLOv8n, and YOLOv8s, respectively; the mAP (mean Average Precision) value was 92.4%, which increased by 6.7%, 7.7%, and 6.5%, respectively. Improved YOLOv5s has an average improvement of 6.26% over YOLOv5s in terms of recognition accuracy: that was 95.7% for non-rotted peanut pods and 90.8% for rotten peanut pods. This article presented a machine vision- based grade classification method for peanut pod rot, which offered technological guidance for selecting high-quality cultivars with high resistance to pod rot in peanut.

KEYWORDS

peanut pod rot, machine vision, improved YOLOv5s, Shuffle Attention, grading classification

1 Introduction

Peanut pod rot, also known as fruit rot, significantly impacts peanut yield and quality, with occurrences noted in the United States (Wheeler et al., 2016), Egypt (Elsayed Abdalla and Abdel-Fattah, 2000) and various regions of China, including Shandong (Zhang et al., 2016) and Hebei Province (Li et al., 2011). The disease's prevalence and severity are leading to increased losses annually, with affected plots seeing up to a 15% yield reduction and severely infected areas losing up to 50%. In some cases, it can lead to total crop failure (He et al., 2022). So far, *N. vasinfect* (Gai et al., 2011; Sun et al., 2011), *Fusarium* sp (Liu et al., 2020), *N. striata* (Sun et al., 2012), *P. myriotylum* (Yu et al., 2019), and *R. solani* (Chi et al., 2015) have been identified as the pathogenic bacteria of peanut pod rot in China. Peanut pod rot poses a severe danger to the safety of peanut output and quality, and it is critical to strengthen effective prevention and control of it.

The difficulty in preventing and treating peanut pod rot can be attributed to the wide range of pathogen hosts (Abd El-aal et al., 2013) and the current lack of varietal resistance (Walker and Csinos, 1980; Lewis and Filonow, 1990; Besler et al., 2003). Varietal resistance is frequently improved through breeding, which is an efficient method of preventing peanut pod rot (Wynne et al., 1991). By assessing the resistance grade of individual peanut plants to pod rot, superior germplasm can be identified, facilitating the development of new peanut varieties. There is comparatively little research on peanut pod rot in China, with the majority of studies on the pathology of peanuts being on leaf diseases, bacterial wilt, and web blotch. At present, the grade classification of individual peanut pod rot is still usually done manually. Manual categorization is labor-intensive, time-consuming, and prone to errors like misidentification, abandonment, and repeated recognition as work time grows, which is thus not ideal for large-scale grading because of the varied grades of peanut decay. More precise grade classification can be attained by machine vision, which can precisely identify and interpret illness signs in photos, extract important information from them, classify and assess them in accordance with predetermined criteria. Additionally, machine vision technology can expedite breeding operations by increasing the speed and efficiency of grade classification in comparison to manual categorization.

CNN (Convolutional neural network) has recently achieved substantial results in the field of object identification (Zaidi et al., 2022), including Faster R-CNN (Ren et al., 2017), YOLO (Redmon et al., 2016), SSD (Single Shot MultiBox Detector) (Liu et al., 2016), etc. Crop identification based on machine vision is more efficient and less expensive, exhibiting a progressive trend of replacing manual identification. Machine vision models have excelled in crop disease detection. Habib et al. (2020) achieved over 90% accuracy in classifying papaya diseases using K-means clustering for segmentation and support vector machines for identification. Harakannanavar et al. (2022) improved this technique by extracting tomato leaf boundaries with K-means clustering and contour tracing, employing SVM (Support Vector Machine), CNN, and K-NN (K-Nearest Neighbors) algorithms for classification, with CNN attaining an impressive 99.6% accuracy rate. Hua et al. (2022) introduced a PD R-CNN algorithm for crop disease detection that incorporates multi-feature decision fusion, consistently delivering accuracy rates above 85% across various disease types. In citrus orchards, Pydipati et al. (2005) developed an algorithm using the CCM (Color Co-occurrence Method) combined with Mahalanobis distance-based and neural network classifiers, achieving over 95% accuracy in distinguishing between healthy and diseased citrus leaves by leveraging hue and saturation features. To address the challenge of diagnosing visually similar corn diseases in the field, He et al. (2023) enhanced the Faster R-CNN by integrating batch normalization and a central loss function, resulting in a model that surpassed the original Faster R-CNN and SSD in terms of average recall rate, F1 score, and both accuracy and detection speed. While these algorithms excel at identifying and labeling lesions, they do not quantify the number of lesions or provide crop counts. Our study addresses this gap by utilizing the YOLO series algorithm, renowned for its object detection capabilities, to recognize peanut images.

The use of YOLO algorithms in agriculture is now a comparatively developed technique. By introducing light-weighting enhancements to YOLOv3, Shen and Zhao (2021) developed a peanut seed identification model with great accuracy that can operate in real-time on the CPU. By adding DenseNet interlayer density, Gai et al. (2023) enhanced the feature extraction ability of the YOLOv4 backbone network CSPDarknet53. Sozzi et al. (2022) tested six versions of the original YOLO model, and the results demonstrated that YOLOv5s can identify green grapes quickly and accurately. Lawal (2023) upgraded the YOLOv5 backbone and neck networks and changed the loss function to EIoU to improve the robustness in complicated and ever-changing situations. Lawal (2021) improved the YOLOv3 model to solve interference problems such as branch and leaf obstruction, lighting shifts, and fruit overlapping. In the identification application of tomatoes, the improved YOLOv3 model exhibited an average prediction rate of 99.5%. Aran et al. (2016) employed a BPNN (Back-propagation neural network) for the grade classification of cashews, reaching an accuracy of 96.8%.

These methodologies can be well coupled with machine vision in their respective crop fields, providing technological backing for the feasibility of this study. The primary challenge faced in this study was to reduce the model size while maintaining recognition performance, in order to adapt it for embedded systems and enable effective grading of outdoor peanut pod rot. The challenges include the scarcity and diversity of data, which complicate the collection of standardized datasets and model training; the complexity of peanut pod rot features, especially the high variability at different stages, presents significant difficulties for accurate identification and grading; although existing machine vision models perform excellently in several other domains, specific improvements are still required to enhance performance for the characteristics of peanut pod rot.

There is currently no research on grading peanut pod rot using machine vision. This study aims to integrate lightweight object detection models into portable devices to support field applications. Given the high computational resource demands, YOLOv8 is not suitable for mobile or embedded devices with limited computing power. In contrast, the YOLOv5 series of algorithms, with their

smaller size, are more suitable for integration into such embedded systems. Among the various versions of YOLOv5, the YOLOv5s has the smallest model size, with a 35% and 70% reduction in size compared to YOLOv8s and YOLOv8n, respectively, making YOLOv5s an ideal choice for integration into resource-constrained devices. To enhance the data reliability and work efficiency, the future approach to image acquisition will shift from single-plant per image to multiple-plants per image, guiding the detection task towards small object detection. With its multi-scale feature fusion, optimized anchoring mechanism, powerful data augmentation, and highly customizable architecture, YOLOv5s has proven to improve the precision of small object detection while maintaining rapid processing speed. Based on these factors, the model was selected for optimization to meet the needs of practical applications.

To facilitate the screening of peanut germplasm resources resistant to pod rot, this paper proposed a grading algorithm based on Shuffle Attention and prediction box location optimization, targeting interference such as peanut pod adhesion, root stem and leaf occlusion. To begin, using the YOLOv5s identification model, the Shuffle Attention mechanism was used to improve the capability of feature representation, location accuracy of lesion area, and robustness in complex backdrops. Then, the loss function was enhanced to improve the regression accuracy of the prediction box and reduce the likelihood of errors and omissions. Finally, the rotten pod rate was estimated by calculating the quantity of rotten peanut pods according to the projected results. The grade classification was carried out based on the rotten pod rate and the results were further compared with those of YOLOv5s, YOLOv8n, and YOLOv8s models. Based on this, the efficiency of the proposed method in this study can be verified.

The rest of this work is structured as follows: Section 2 discusses the planting environment of peanuts, the establishment procedure of the dataset, and the design and optimization of the pod rot grading model. Section 3 introduces relevant tests and compares the recognition and prediction performance of four models. Section 4 discusses the shortcomings of the proposed method and future research directions for the grade classification of peanut pod rot. Section 5 highlights the experimental results of the proposed model, emphasizing the application value of this study.

2 Materials and methods

2.1 Sample acquisition

The samples were collected from the Experimental Station of Hebei Agricultural University in Qingyuan District, Baoding City, Hebei Province (38°80'N, 115°57'E). A cultivar of peanut, Jinongxian No.1, was taken as the experimental sample in this study, which was planted in spring, 2023, with ridge plastic film and mulching, ridge spacing of 85 cm and two rows per ridge. The average row spacing was 42.5 cm, with a hole spacing of 15.5 cm and two seeds per hole. The planting density was 60750 holes/acre.

Thirty peanuts were taken as samples from the field to the laboratory for washing to remove soil on surfaces. To acquire the dataset, pictures were taken using a SAMSUNG Galaxy S20+ phone with 64 megapixels. The sampling period was set from September 27th to September 29th, 2023, all of which are sunny days. The shooting time was set from 12:00 to 14:00 with sufficient light and 16:00 to 18:00 with dim light. All pictures were taken under natural light, and a total of 2000 peanut images were collected. The shooting angle was set as either top right or side up, while the shooting distance was set as long shot, close shot, and ultra-close shot. The distance from peanuts in the long shot was about 120 cm, the close shot about 40 cm, and the ultra-close shot about 10 cm.

High-yielding peanut plants tend to stack more frequently because of the abundance of pods, which makes automatic identification challenging. It is unavoidable to run into problems like peanut occlusion and adhesion when taking pictures. Individual peanut and pod images were captured independently to better avoid interference in image recognition and enhance the accuracy and robustness of the model. Figure 1 presents the images of typical samples.

2.2 Dataset production and image enhancement

The images obtained by the phone have a pixel size of 4032*1816. Although a large pixel size can improve the training effect, it significantly affects the training speed. As a result, the pixel size of the original image was resized to be 1400*631.

Labeling was used to annotate the gathered peanut images. Mark the non-rotted peanuts (G) and rotten peanuts (R) individually throughout labeling, and save the files on the computer in the "xml" format. Before training the object detection model, five enhancement procedures were randomly combined and applied to each image to increase the sample size and boost the training effect. The enhancement treatment included noise addition, cutout, rotation, cropping, translation, horizontal flip, and vertical flip. Figure 2 depicts the enhanced image. The dataset was finally expanded to 12,000 sheets, which promoted the learning effect of the model on the characteristics of non-rotted and rotten peanuts. There was a total of 83,850 labels in the dataset, including 56,730 non-rotted peanuts and 27,120 rotten peanuts. The dataset was randomly divided into training and testing sets in a 9:1 ratio.

2.3 YOLOv5 model

The YOLOv5 network structure (Qiao et al., 2021) consists of three main components: Backbone, Neck, and Prediction Head, as shown in Figure 3. The Backbone network adopts the CSPDarknet53 architecture, which performs well in feature extraction and was used to extract rich multi-scale features from input images. The feature fusion module was used to fuse feature maps with different scales from the Backbone network. YOLOv5 employed a Feature Pyramid Network (FPN) to fuse features at different levels through upsampling and downsampling, thereby improving the accuracy and robustness of object detection. The

FIGURE 1
Original peanut image samples. **(A)** Individual non-rotted peanut; **(B)** Individual rotten peanut; **(C)** Low-yielding plant without occlusion and adhesion; **(D)** High-yielding plant with severe occlusion and adhesion.

Prediction Head was responsible for generating the bounding box and category prediction of the object. YOLOv5 adopted a decoupled multi-level prediction head structure that can effectively handle objects of different scales, achieving a good balance between the speed and accuracy of identification. The combination of these components gave YOLOv5 excellent performance and efficiency in object detection tasks.

2.4 Improvement of feature extraction backbone network

This study enhances YOLOv5s to classify the grade of each peanut and calculate the rotten pod rate. It is required to output the total number of G and R labels.

Some peanuts grow densely and have problems like adhesion and occlusion, which makes it challenging to effectively identify some peanuts separately. Therefore, a Shuffle Attention (SA) module (Zhang and Yang, 2021) was devised in this study. Shuffle Attention is a method of describing feature dependencies through grouping, parallel processing, and information exchange. According to the schematic diagram shown in Figure 4, SA first divided the channel dimensions into several subfeatures and processed each subfeature in both spatial and channel dimensions using the Shuffle Unit. The channel shuffle operator was then employed to enhance information exchange between distinct subfeatures after all subfeatures had been summarized. After that, Shuffle Attention was placed after each C3 module in the Backbone, which made local features visible to the attention module. The Shuffle Attention was performed on each layer to share learning pressure.

The purpose of adding the SA module is as follows:

1. Boost the capacity for feature representation. Through channel shuffling and self-attention mechanisms, the SA module can improve the network's ability to represent features, including long-distance dependency and contextual information. It can also help extract features related to peanut pod rot from images more effectively, such as fine details of lesion areas and contextual information.
2. Improve the positioning accuracy of lesion areas. The SA module employs a self-attention mechanism to gather association information from various positions of the image. Based on this, the lesion area of peanut pod rot can be located more precisely, thereby improving positioning accuracy and minimizing missing and false identification.
3. Enhance the ability to distinguish between non-rotted and rotten peanuts. Peanuts differ from one another in their

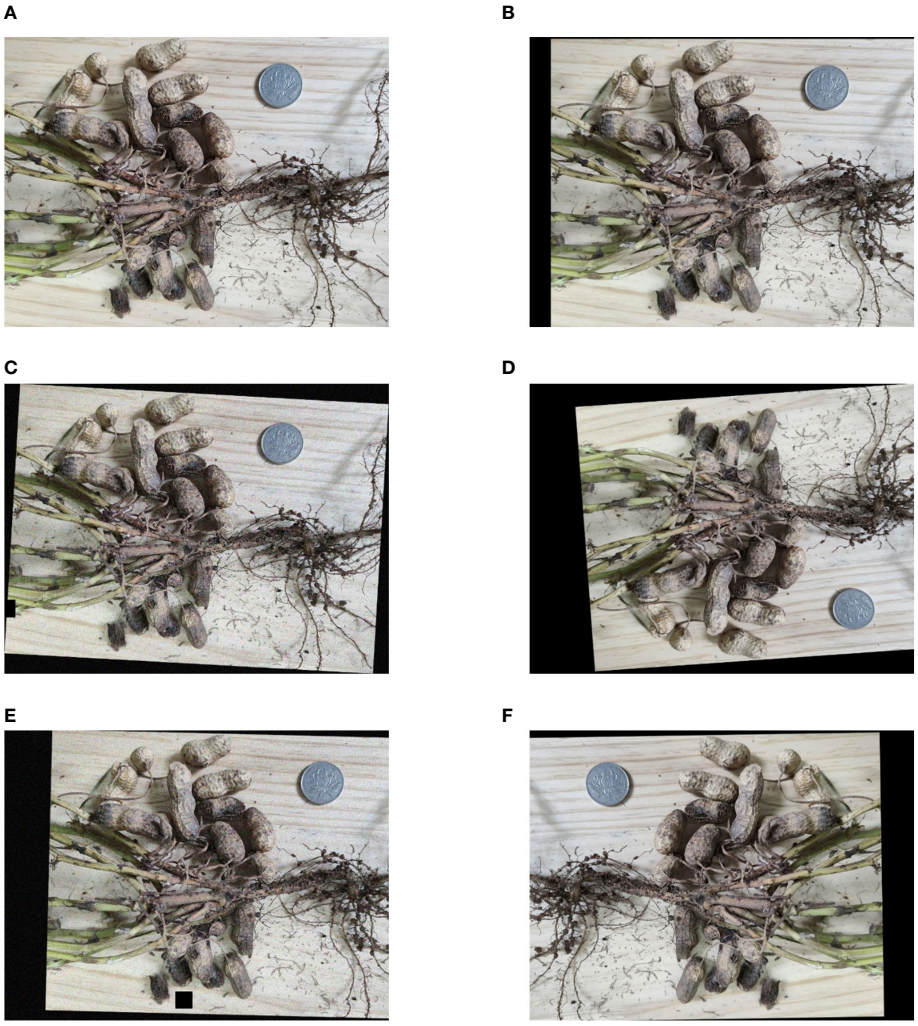


FIGURE 2
Original and enhanced images. **(A)** Original image; **(B)** Translation; **(C)** Rotation+cutout+noise; **(D)** Vertical flip+rotation+translation; **(E)** Rotation +cutout+noise+translation; **(F)** Horizontal flip+rotation+translation.

physical characteristics. The channel shuffling and self-attention mechanism of the SA module can distinguish between rotten and non-rotted peanuts based on minute feature differences. YOLOv5s can learn and discriminate between rotten and non-rotted peanuts, boosting the network's ability to differentiate pod quality.

It can be concluded that the SA module has increased the feature representation ability, the positioning accuracy of the lesion area, and the capacity to discriminate different disease grades. The introduction of the SA module to YOLOv5s has promoted the accuracy and robustness of peanut pod rot identification by improving the effectiveness of grade classification. Figure 5 depicts the overall architecture design of adding a SA module to YOLOv5s.

2.5 Loss function

The loss functions of YOLOv5s include Classification Loss (L_{cla}), Localization Loss (L_{loc}), and Confidence Loss (L_{conf}). The total loss function is the sum of the three, as shown in Equation (1):

$$Loss = L_{cla} + L_{loc} + L_{conf} \quad (1)$$

Currently, the Localization Loss used in the YOLOv5s model is CIoU (Lu et al., 2022). The sample size of non-rotted peanuts in the dataset was much larger than that of rotten peanuts. The significant quantity difference resulted in a problem of imbalanced samples. Therefore, there is a higher requirement for the accuracy of prediction box regression. The calculation formula for CIoU is as shown in Equations (2)–(4):

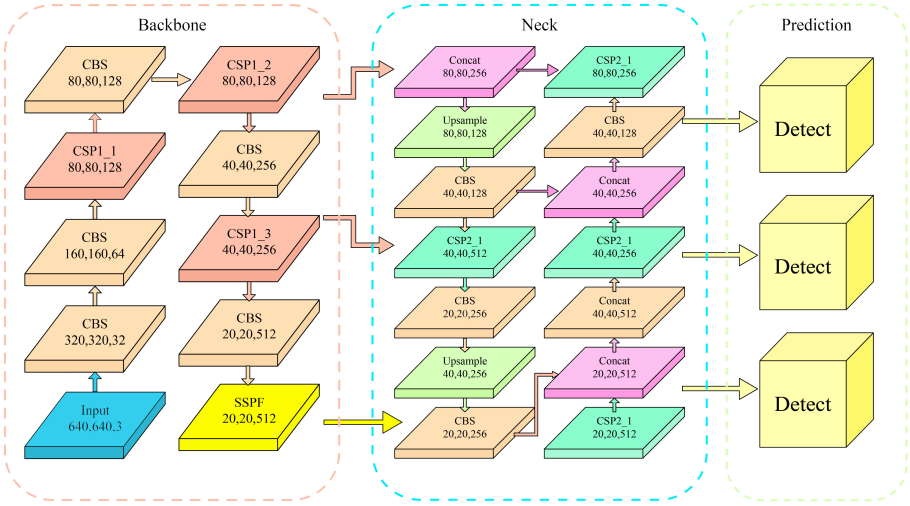


FIGURE 3
Network architecture diagram of YOLOv5.

$$CIoU = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + av \tag{2}$$

$$v = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2 \tag{3}$$

$$\alpha = \begin{cases} 0, & if\ IoU < 0.5 \\ \frac{v}{1-IoU+v}, & if\ IoU \geq 0.5 \end{cases} \tag{4}$$

Where, IoU refers to the intersection over union between the ground truth box and the prediction box. $\rho^2(b, b^{gt})$ refers to the Euclidean distance between the center points of two boxes. c^2 is the squared≥value of the diagonal length of the minimum closure region that can contain two boxes at the same time. The ratio of the two represents the distance between the ground truth box and the prediction box. av is the influencing factor of the length-width ratio between the two boxes. w, h, w^{gt}, and h^{gt} represent the width and height of the prediction box and the ground truth box, respectively.

When there is an inclusion phenomenon between the detection box and the ground truth box, CIoU overcomes the problems of degradation to IoU as well as the slow convergence in the horizontal and vertical dimensions when the two boxes cross. Although CIoU offers certain advantages over IoU, the difference in aspect ratio given by v in the formula is not the real difference between width and height and its confidence, which will impede effective similarity optimization of the model.

EIoU takes into account the real difference in length, width, overlapping area, and center point distance (Zhang et al., 2022). It solves the imprecise definition of aspect ratio based on CIoU by calculating the difference in width and height instead of aspect ratio, thus boosting regression accuracy. The imbalance between non-rotted and rotten peanut samples in BBox regression can be resolved by introducing Focal Loss. Therefore, EIoU was used in place of CIoU in this study, and the calculation formula for EIoU is as shown in Equation (5):

$$EIoU = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \frac{\rho^2(w, w^{gt})}{c_w^2} + \frac{\rho^2(h, h^{gt})}{c_h^2} \tag{5}$$

Where, c_w and c_h are the width and height of the bounding rectangle of the two boxes, respectively. $\frac{\rho^2(w,w^{gt})}{c_w^2}$ and $\frac{\rho^2(h,h^{gt})}{c_h^2}$ reveal the difference in width and height between the prediction box and the ground truth box.

The improved model is named YOLOv5s-ES, which was established based on the YOLOv5s model with an introduction of the SA module and a replacement of CIoU with EIoU.

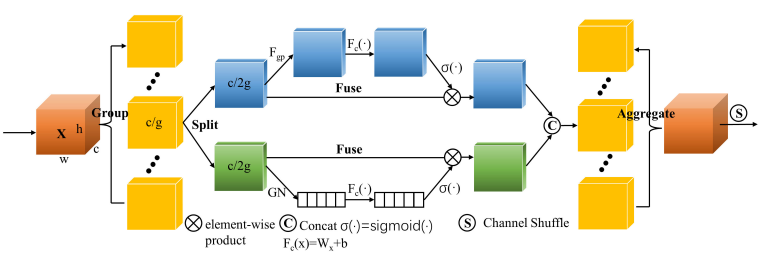


FIGURE 4
Schematic diagram of Shuffle Attention module.

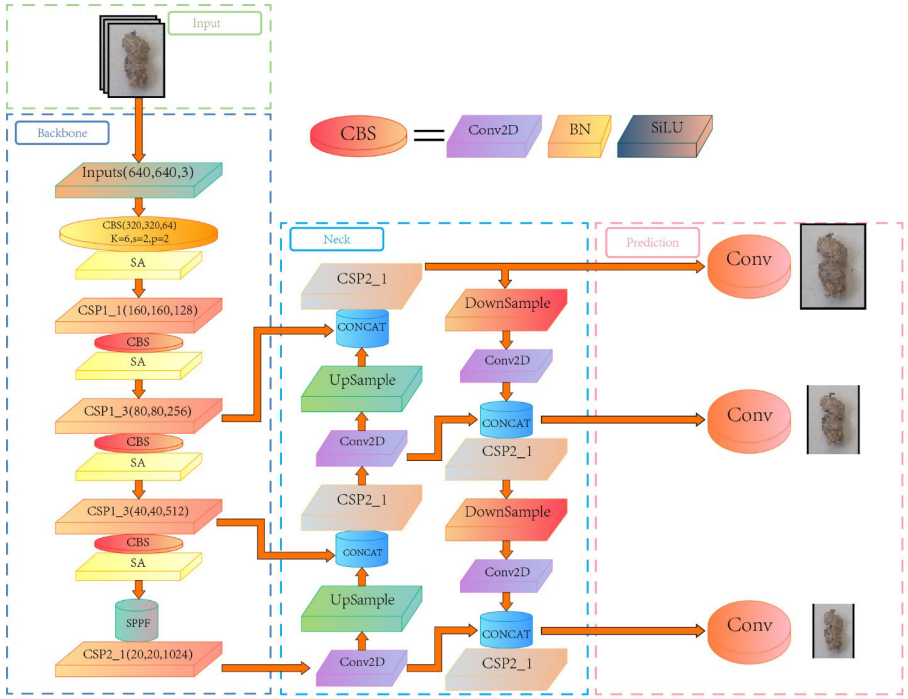


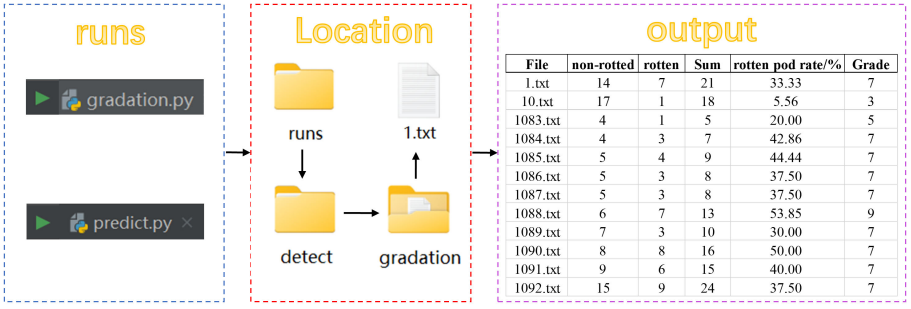
FIGURE 5
YOLOv5s architecture diagram with added Shuffle Attention module.

2.6 Grade classification module

On the one hand, the grade classification of peanut pod rot can be used to determine the severity of diseases. Different stages of the disease may necessitate different prevention and control measures, and the grading aids in the selection of appropriate tactics as well as the improvement of preventative and control effectiveness. On the other hand, the grade classification can offer timely awareness of the disease progression. Taking early response measures is advantageous for sensible resource allocation and cost reductions. The grade classification of peanut pod rot can be claimed to increase targeted and effective prevention and control work, ensure peanut output and quality, and reduce economic losses.

According to the findings of Wheeler et al. (2016), the following are the grading criteria for peanut pod rot: Level 1 for no rotten fruit, with a rotten pod rate of 0; Level 3 for a rotten pod rate between 0 and 10%; Level 5 for a rotten pod rate between 10% and 25%; Level 7 for a rotten pod rate between 25% and 50%; and Level 9 for a rotten pod rate larger than 50%.

As shown in Figure 6, an external grade classification module was put after the Prediction network to perform the grading function. After executing detect.py, the predicted images were generated in the exp folder, along with a graduation folder. This folder includes.txt files with the predicted image information, as well as statistical data on the number of non-rotted and rotten peanuts. Running gradation.py after generating the text file information will generate an.xlsx file in the root directory that contains the amount of non-rotted and rotten peanuts, as well as the overall number, rotten pod rate, and grade classification of rotten peanuts for all predicted images. The numbers of non-rotted and



File	non-rotted	rotten	Sum	rotten pod rate/%	Grade
1.txt	14	7	21	33.33	7
10.txt	17	1	18	5.56	3
1083.txt	4	1	5	20.00	5
1084.txt	4	3	7	42.86	7
1085.txt	5	4	9	44.44	7
1086.txt	5	3	8	37.50	7
1087.txt	5	3	8	37.50	7
1088.txt	6	7	13	53.85	9
1089.txt	7	3	10	30.00	7
1090.txt	8	8	16	50.00	7
1091.txt	9	6	15	40.00	7
1092.txt	15	9	24	37.50	7

FIGURE 6
Implementing the peanut pod rot grading system in PyCharm.

TABLE 1 Parameter configuration of training model.

Parameter	Value
Num class	2
Epoch	200
Batch size	32
Initial learning rate	0.01

rotten peanuts are shown in the second and third columns, respectively. The rotten pod rate is shown in the fifth column. The grade of individual rotten peanuts, as decided by the grading criteria, is shown in the sixth column. The formula for calculating the rotten pod rate is shown in Equation (6):

$$Rotten\quad pod\quad rate = \frac{Number\quad Rotten}{Number\quad Non-rotted + Number\quad Rotten} \tag{6}$$

Where, *Number Rotten* refers to the number of rotten peanuts; *Number Non-rotted* refers to the number of non-rotted peanuts.

3 Results

3.1 Model specification

CUDA 11.3 and cuDNN8.0 were the network training environments used in this study. A 12GB NVIDIA RTX3070Ti was used as the training accelerator. Facebook's open-source deep learning framework Python 1.11.0 was employed as the development environment, and the programming language used was Python 3.9.7. Adaptive Moment Estimation (Adam) was used to automatically modify the learning rate and solve the gradient vanishing problem, which allowed the model to converge faster and perform better. Table 1 displays the parameter configuration of the training model.

3.2 Evaluation indicator

This study utilized two methods, visual evaluation, and quantitative comparison, to evaluate the grading performance. Visual evaluation is a common way to visually compare and evaluate the detection results. In quantitative analysis, the evaluation indicators are Precision (P), Average Precision (AP), mean Average Precision (mAP), and Comparison Precision (CP). The calculations of the three indicators are shown in Equations (7)–(9):

$$P = \frac{TP}{TP + FP} \times 100\,\% \tag{7}$$

$$mAP = \frac{\sum_{n=1}^{2} AP(n)}{2} \times 100\,\% \tag{8}$$

$$CP = \frac{AS}{RS} \times 100\,\% \tag{9}$$

Where, TP is the quantity of label boxes for non-rotted and rotten peanuts that accurately match the prediction boxes. FP is the number of prediction boxes containing inaccurate forecasts. P is the percentage of non-rotted and rotten peanuts that were accurately identified in each prediction box. AP represents the average Precision value of each category. mAP represents the average Precision value of all categories. AS (Automatic Statistics) represents the number of images where the model correctly

FIGURE 7
Comparison of recognition results of four models. **(A)** No adhesion; **(B)** Slight adhesion; **(C)** Severe adhesion.

TABLE 2 Data comparison between the three enhanced models and YOLOv5s.

No.	Added SA Module	EIoU	mAP/%	P-value/%
1	×	×	86.2	/
2	√	×	88.7	0.544
3	×	√	87.5	3.759
4	√	√	92.4	0.002

identifies non-rotted and rotten peanuts in the image. RS (Realistic Statistics) represents the actual number of images of different types. CP represents the comparison precision.

3.3 Experiment result analysis

Plant phenotypic detection makes extensive use of object detection. In order to compare the detection performance of YOLOv5s-ES on peanut images, this study used three YOLO-based object detection models, i.e. YOLOv5s, YOLOv8n, and YOLOv8s. Comparative experiments were carried out under the conditions of no adhesion, slight adhesion, and severe adhesion to validate the improving effect of the model. Comparative experiments aid in understanding the differences in performance between different models and drive future improvements to object detection algorithms. Simultaneously, code availability and repeatability were taken into consideration to assure the dependability and reproducibility of the experiment. Figure 7 depicts the identification results of each model under various adhesion situations.

Figure 7A depicts a peanut image with no adhesion. It can be seen that the four models all had good recognition performance, achieving proper recognition with no omissions or errors. Figure 7B depicts a peanut image with slight adhesion, and the recognition ability of the three unimproved models all dropped. YOLOv8n missed two peanuts, and YOLOv5s missed one. Although YOLOv8s distinguished all the peanuts, the accuracy of the prediction box was low, and a single peanut pod was not marked. Figure 7C depicts a peanut image with severe adhesion. The identification ability of the other three models was considerably diminished, with the exception of the YOLOv5s-ES model. YOLOv8n missed 4 peanuts, with low prediction accuracy. YOLOv5s missed 3, with a relatively high accuracy of the prediction box. Although YOLOv8s recognized all the peanuts, the accuracy of the prediction box was extremely low, with cases of repeated and incorrect recognition. The YOLOv5s-ES model recognized all the peanuts correctly, with only one prediction box being inaccurately labeled. It can be concluded that the improved model YOLOv5s-ES effectively solved the problems that other three algorithms encountered when predicting images, and had the feasibility of grading peanut pod rot in practical applications.

The SA module was introduced to the YOLOv5s-ES model and the loss function CIoU was replaced with EIoU. Ablation experiments were carried out on the YOLOv5s-ES model to confirm the efficacy of the enhanced model. The experimental outcomes are displayed in Table 2, the mAP values represent the average results of five-fold cross-validation.

The mAP of the model increased by 2.5% after the SA module was introduced to the YOLOv5s backbone network, as shown in Table 2. The mAP increased by 1.3% after improving the loss function of the original model. After incorporating both improvements into the model, the value of mAP reached 92.4%,

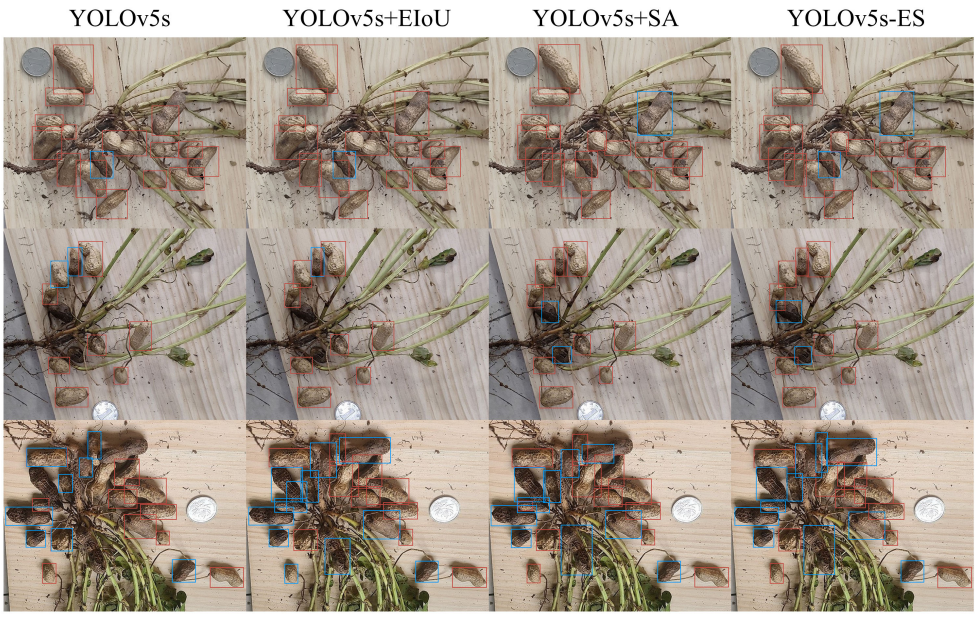


FIGURE 8
Visual comparison between the three enhanced models and YOLOv5s.

TABLE 3 Comparison accuracy value comparison of different algorithms.

	YOLOv5s			YOLOv5s-ES			YOLOv8n			YOLOv8s		
	AS_1	RS	CP_1/%	AS_2	RS	CP_2/%	AS_3	RS	CP_3/%	AS_4	RS	CP_4/%
No	50	50	100	50	50	100	49	50	98	50	50	100
Slight	42	50	84	46	50	92	40	50	80	42	50	84
Severe	38	50	76	46	50	92	36	50	72	39	50	78
Total	130	150	86.67	142	150	94.67	125	150	83.33	131	150	87.33

6.2% higher than that of YOLOv5s. In order to be more convincing, this study verified whether the differences between the algorithm variants were statistically significant and calculated the corresponding P-values. The results showed that the P-values of the three variants of the algorithms were less than 5%, which proved that each improvement was significantly correlated to the improvement of the detection performance. Based on this, the effectiveness of the improved model can be verified.

This study aims to enhance the performance of a peanut image recognition model, particularly under complex background conditions, through two key improvements. To assess the effectiveness of these enhancements, three groups of high-yield peanut images, which demonstrated superior recognition capabilities in preliminary experiments, were selected as cases. These images encompass rich background information and typical challenges such as mutual occlusion and environmental noise.

The comparison of the visualization results of the ablation experiment in Figure 8 reveals the effectiveness of the model improvement. By integrating the SA (Spatial Attention) mechanism, the model focuses more on key areas when processing peanut images in complex backgrounds, significantly reducing the missed detection and false detection rates of the model, especially in cases where peanut leaves and roots are mixed or adhered to each other, improving the accuracy and robustness of recognition. Furthermore, the model adopts the EIoU loss function instead of the traditional IoU loss, which increases the comprehensive consideration of the target shape, size, and center point, improves the accuracy of bounding box positioning, and is crucial for the accurate classification of peanut fruit rot.

3.4 Comparative experiments between multiple algorithms

Based on the RS values and AS values of the four models, the CP values were calculated to validate the identification performance of the enhanced model on a solitary image. The AS value indicates the number of images in which the algorithm properly distinguished non-rotted and rotten peanuts in the image. The RS value indicates the number of images with severe adhesion, slight adhesion, and no adhesion. One hundred and fifty images of peanuts were chosen at random for the validation dataset of the experiment, with 50 images for each adhesion type. Four models - YOLOv5s, YOLOv5s-ES, YOLOv8n, and YOLOv8s - were used to identify the 150 images. The numbers of images for non-rotted and rotten peanuts that can

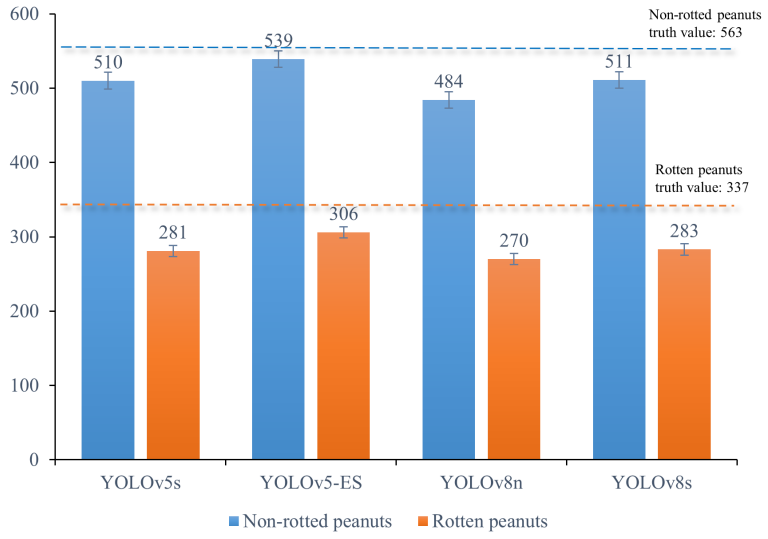


FIGURE 9
Statistical graph of non-rotted and rotten peanuts identified by four different models.

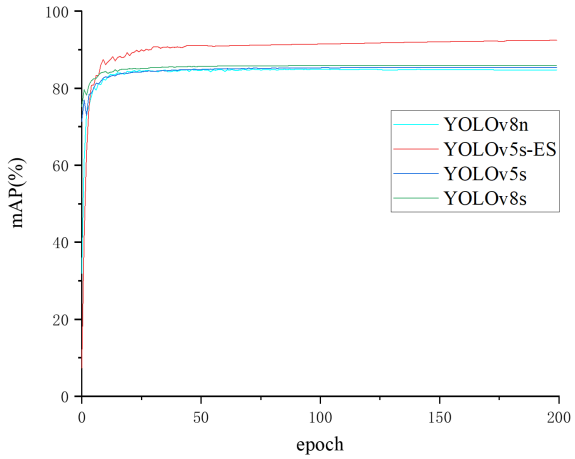


FIGURE 10
Comparison curve of mean Average Precision values of different algorithms.

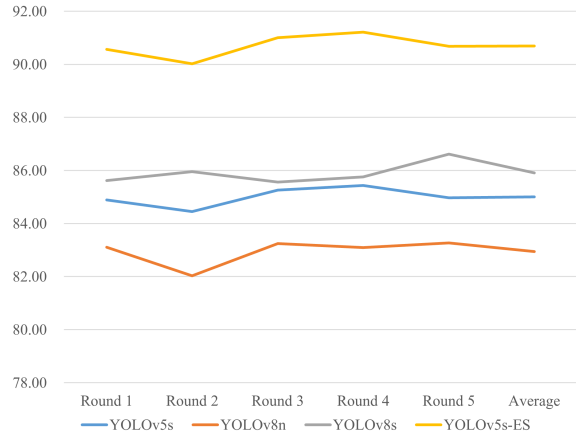


FIGURE 12
Comparison of Recall (%) for five-fold cross-validation of four models.

be successfully identified via the four models were recorded as AS_1, AS_2, AS_3 and AS_4. The corresponding CP_1, CP_2, CP_3 and CP_4 were calculated as well. Table 3 displays the comparison precision values of the four models.

Due to the relatively simple identification of peanut images with no adhesion, more attention was paid to comparing the prediction results of images with slight and severe adhesions. The comparison precision of the four models was 84%, 92%, 80%, and 84%, for the images with slight adhesion and 76%, 92%, 72%, and 78% for the images with severe adhesion, respectively. When it came to prediction performance, YOLOv5s-ES outperformed the three unaltered models with an improvement in the case of slight adhesion and a significant improvement in the case of severe adhesion.

To confirm the enhanced model's capacity to distinguish between non-rotted and rotten peanuts, 100 peanut images containing a higher proportion of rotten peanuts - a total of 563 non-rotted ones and 337 rotten ones - were chosen for identification using the four models. Statistical analysis was performed to determine how many rotten and non-rotted peanuts were identified, and the results were illustrated in Figure 9.

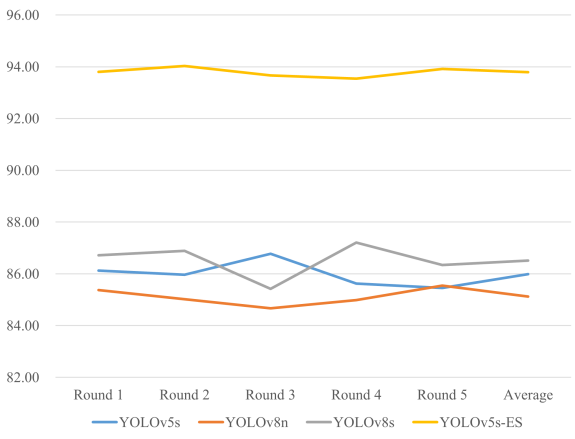


FIGURE 11
Comparison of Precision (%) for five-fold cross-validation of four models.

Figure 9 illustrates that YOLOv8s identified non-rotted peanuts with a high recognition rate of 90.76%, but only 83.98% for rotten peanuts, the recognition rate of YOLOv5s is basically the same as YOLOv8s. This is due, in part, to an imbalance in the sample size between non-rotted and rotten peanuts, which limited the information available for model learning about rotten peanuts. However, some rotten pods shared coloration with rotten stems, roots, and leaves, making identification more challenging. YOLOv8n had a moderate recognition rate and a significantly weaker capacity to distinguish between rotten and non-rotted peanuts, this model had an overall recognition rate of about 83%. The above data is essentially in line with the comparison precision values listed in Table 3. The enhanced YOLOv5s-ES model can identify rotten peanuts with a recognition rate of 90.8% and non-rotted ones of 95.74%. The enhanced model considerably enhanced the capacity to identify rotten peanuts and had a slight improvement in identifying non-rotted ones.

To further illustrate the superiority of the algorithm proposed in this study, four models were compared for mAP change curves on the same dataset. The mAP change curve during training is displayed in Figure 10. It can be seen that YOLOv5s, YOLOv8n, YOLOv8s, and YOLOv5s-ES had mAP values of 85.7%, 84.7%, 85.9%, and 92.4%, respectively. The convergence rates of all four curves were incredibly quick, and the three unimproved models achieved fitting with around 75 epochs. Excessive data fitting may result in unstable model parameters. When there is some randomness or fluctuation in the data, the model may update parameters excessively to accommodate these changes, resulting in inconsistent model performance. Instability may affect the model's reliability and interpretability, resulting in poor performance in practical applications since it cannot catch potential patterns and overall trends in the data. After 100 epochs, the mAP of YOLOv5s-ES hit 91.4% and tended to

TABLE 4 Comparison of precision and recall metrics across four models using five-fold cross-validation.

Model	YOLOv5s	YOLOv5s-ES	YOLOv8n	YOLOv8s
Precision (%)	86.0	93.8	85.1	86.5
Recall (%)	85.0	90.7	83.0	85.9

stabilize, eventually achieving 92.4%. It can be concluded that the enhanced model leveraged the likelihood of capturing real patterns and overall trends in the data, rather than unnecessarily responding to the noise and intricacies of the training data. In this way, the generalization ability of the model can be promoted on unknown data, making it more suitable for practical applications.

To address the potential inaccuracies in assessment results that might arise from a single dataset split, a five-fold cross-validation study was conducted on four different models. Precision and Recall values from five separate trials were collected and averaged. The results of the five-fold cross-validation for both metrics are presented in Figures 11 and 12. The data in the figures reveal only minor fluctuations in the model's recognition capabilities across the five randomly partitioned datasets, confirming the model's robust generalization performance in identifying peanut fruit rot disease. The Precision of the improved model YOLOv5s-ES was 93.8%, 7.8%, 8.7%, and 7.3% higher than YOLOv5s, YOLOv8n, and YOLOv8s, respectively. The Recall value was 90.7%, which increased by 5.7%, 7.7%, and 4.8% than the other three models, respectively. As shown in Table 4.

4 Discussions

Peanut pod rot causes fruit degradation and yield loss, making prevention and management difficult and potentially transmittable. Grade classification of peanut pod rot allows for the evaluation of disease resistance, the selection of outstanding germplasm resources, and the promotion of breeding improvement. This study suggests an object detection approach based on YOLOv5s-

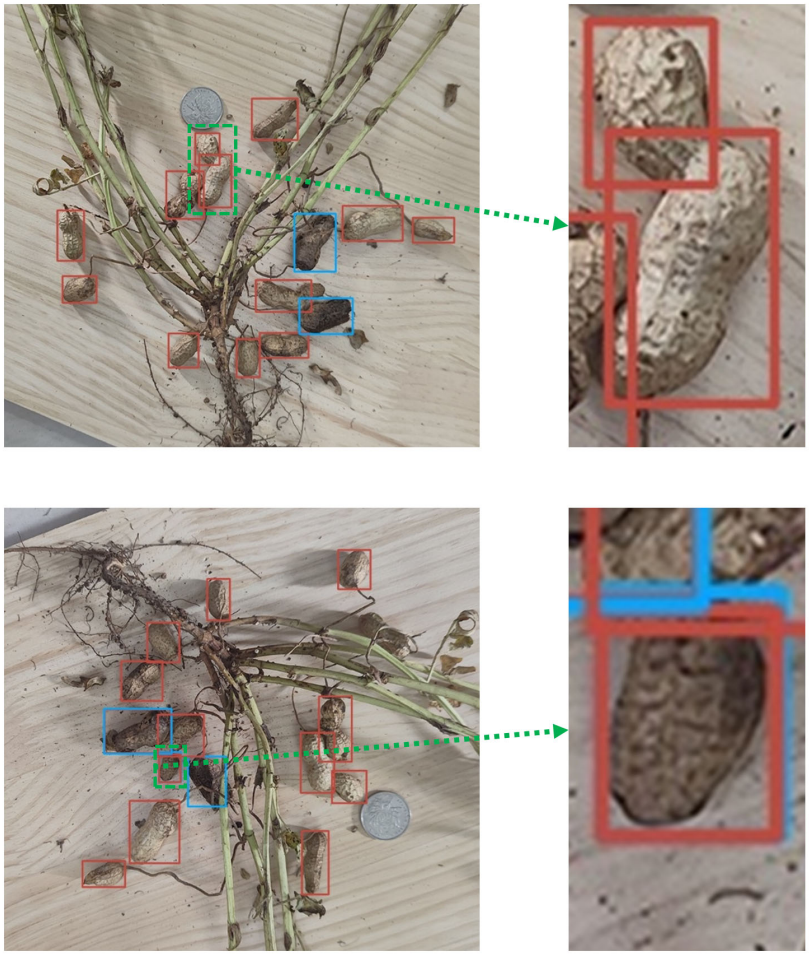


FIGURE 13
Two types of misidentification present in the improved model.

ES in response to the drawbacks of manual classification, which can successfully increase the efficacy and precision of pod rot grading and eventually replace conventional manual classification. Although this study is of great significance in addressing pod rot grading, there are certain concerns that require additional research and analysis.

The improved YOLOv5s-ES model may encounter misidentification during prediction. Two typical examples are shown in Figure 13.

In Figure 13A, the model accurately identified and labeled the rotten peanut, but incorrectly identified the peanut as a non-rotted one and repeated labeling, leaving the model unable to differentiate between the non-rotted and rotten types of the peanut. One reasonable explanation on the one hand is the insufficient debugging of the model parameter threshold, which makes it hard for the model to reliably identify whether this type of peanut belongs to non-rotted or rotten. Based on this, improvement can be achieved through parameter adjustment, threshold modification, etc. On the other hand, some peanut pods have a moderate degree of decay, making it hard to distinguish between the non-rotted and rotten types solely based on phenotypic sampling. In this case, semantic segmentation methods can be introduced. Specifically, the diseased area of each peanut is calculated, the proportion of which can be used to determine whether the pod belongs to a rotten one. In this way, the problem can be solved using the judgment results of semantic segmentation combined with object detection algorithms.

In Figure 13B, a peanut pod was mistakenly identified as two pods, meaning that the model labeled a valencia type peanut as a double-kernal one and a single-kernal one during prediction. This error tends to happen when the sample size is insufficient. During training, the model identified a small number of valencia type peanuts, so that when new valencia type peanuts appeared, the entire pod could not be correctly identified and was misjudged as two or more double-kernal and single-kernal pods. Increasing the sample size, especially the images of valencia type peanuts, is an effective way to solve such recognition errors.

Furthermore, after being infected with peanut pod rot, some peanut pods only form a thin coating of decay on the surface, leaving the kernels unaffected. As a result, the impact on peanut yield includes the rotten kernel rate. The degree of pod rot was used to classify peanut pod rot in this study, and the rotten kernel rate was not considered. As a result, the projected data has a poor practical application value in yield estimation, which is a shortcoming of machine vision-based pod rot grade classification. In order to ensure that the design scheme can be used effectively in more aspects, greater attention may be paid to the grading of peanut pod rot under the dual factors of rotten pod rate and rotten kernel rate.

Moreover, due to the differences in pod rot among various peanut varieties and the lack of relevant samples, this study cannot predict whether the model's recognition capability for images of other peanut varieties will decrease. In order to overcome the aforementioned drawbacks, we will expand the sample size of different kinds of peanuts, conduct transfer learning across different varieties with the model, combine semantic segmentation methods, and enhance the model's performance. First, we will ascertain whether a single peanut has pod rot. Then, the peanut pods in the image will be annotated using object identification methods to improve the accuracy of the results. To further increase prediction accuracy and visibility, it is feasible to introduce an instance segmentation algorithm and confirm its benefits in extreme peanut adhesion scenarios. Additionally, data on peanut pod rot in complex environments should be analyzed concurrently to strengthen the resilience of the model and make it more applicable to peanut plants in various conditions and cultivars.

5 Conclusions

Starting with the relevance of grading individual peanut pod rot, this study employed the Jinongxian No.1 peanut as the experimental object in the field management planting base. To address the inadequacies of the current grade classification for peanut pod rot, a machine vision-based method was proposed using a modified loss function and feature extraction backbone network of the YOLOv5s algorithm.

(1) The SA module was introduced to the YOLOv5s network as the main framework to overcome problems like adhesion and obstruction in the dense development of certain peanut plants, which are vulnerable to interference from roots, stems, and leaves. The feature extraction ability of the network for identifying non-rotted and rotten peanuts was enhanced by substituting the EIoU for the CIoU in the original network in response to the sample imbalance problem caused by the fact that the number of non-rotted pods is much higher than the number of rotten pods in actual situations.

(2) With a Precision value of 93.8%, the improved model YOLOv5s-ES outperformed YOLOv5s, YOLOv8n, and YOLOv8s by 7.8%, 8.4%, and 7.3%, respectively. Its mAP value was 92.4%, outperforming YOLOv5s, YOLOv8n, and YOLOv8s by 6.7%, 7.7%, and 6.5%, respectively. With a non-rotted pods recognition rate of 95.74% and a rotten pods recognition rate of 90.8%, the comparison precision reached 94.67%, satisfying the requirements of exact recognition.

(3) With the addition of a grade classification module after the Prediction network, this study realized the calculation of the number of non-rotted and rotten peanuts as well as the rotten pod rate in the images. The results were then written into a.txt file. The grading of pod rot can be completed by adding the grade classification module to the YOLOv5s-ES model, which allows the database to read text files and record the number of non-rotted and rotten peanuts, the rotten pod rate, and the grading of pod rot.

In conclusion, the improved model proposed in this study will help the automatic grade classification of individual peanut pod rot in practical prediction applications, facilitating in the screening of superior germplasm resources and peanut breeding.

Data availability statement

The dataset supporting this study can directly download using the link below: https://github.com/JiaLBYGG/Peanut-pod-rot-dataset.git.

Author contributions

YL: Writing – original draft, Validation, Methodology, Conceptualization. XL: Writing – review & editing. YF: Writing – original draft, Validation, Project administration, Data curation, Conceptualization. LL: Writing – review & editing, Supervision, Funding acquisition. LS: Writing – review & editing, Visualization, Supervision, Funding acquisition. GY: Writing – original draft, Investigation, Formal analysis, Data curation. YG: Writing – original draft, Investigation, Data curation. YZ: Writing – original draft.

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 Key Research and Development Program of China (Grant No. 2023YFD1202800) and State Key Laboratory of North China Crop Improvement and Regulation (Grant No. NCCIR2022ZZ-15).

Acknowledgments

We would like to thank KetengEdit (www.ketengedit.com) for its linguistic assistance 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

References

Abd El-aal, A. N. A., Khalifa, M. M. A., and Abol-Ela, M. F. (2013). Inheritance of some economic characters, reaction to pod rot diseases and aflatoxen contamination in peanut (arachis hypogaea l.). *J. Plant Product.* 4, 445–470. doi: 10.21608/jpp.2013.72146

Aran, M. O., Nath, A. G., and Shyna, A. (2016). "Automated cashew kernel grading using machine vision," in *2016 International Conference on Next Generation Intelligent Systems (ICNGIS)*, (Kottayam, India: IEEE), 1–5. doi: 10.1109/ICNGIS.2016.7854063

Besler, B. A., Grichar, W. J., Brewer, K. D., and Baring, M. R. (2003). Assessment of six peanut cultivars for control of rhizoctonia pod rot when sprayed with azoxystrobin or tebuconazole1. *Peanut Sci.* 30, 49–52. doi: 10.3146/pnut.30.1.0010

Chi, Y. C., Xu, M. L., Yang, J. G., Wang, F. L., and Wu, J. X. (2015). First report of rhizoctonia solani causing peanut pod rot in China. *Plant Dis.* 100, 1008. doi: 10.1094/PDIS-07-15-0840-PDN

Elsayed Abdalla, M., and Abdel-Fattah, G. M. (2000). Influence of the endomycorrhizal fungus glomus mosseae on the development of peanut pod rot disease in Egypt. *Mycorrhiza* 10, 29–35. doi: 10.1007/s005720050284

Gai, R., Chen, N., and Yuan, H. (2023). A detection algorithm for cherry fruits based on the improved yolo-v4 model. *Neural Comput. Appl.* 35, 13895–13906. doi: 10.1007/s00521-021-06029-z

Gai, Y., Pan, R., Ji, C., Deng, M., and Chen, W. (2011). First report of peanut foot rot caused by neocosmospora vasinfecta var. Africana in jiangxi province, China. *Plant Dis.* 95, 1480. doi: 10.1094/PDIS-06-11-0489

Habib, M. T., Majumder, A., Jakaria, A. Z. M., Akter, M., Uddin, M. S., and Ahmed, F. (2020). Machine vision based papaya disease recognition. *J. King Saud Univ. - Comput. Inf. Sci.* 32, 300–309. doi: 10.1016/j.jksuci.2018.06.006

Harakannanavar, S. S., Rudagi, J. M., Puranikmath, V. I., Siddiqua, A., and Pramodhini, R. (2022). Plant leaf disease detection using computer vision and machine learning algorithms. *Global Trans. Proc.* 3, 305–310. doi: 10.1016/j.gltp.2022.03.016

He, W., Feng, L., Li, Z., Zhang, K., Zhang, Y., Wen, X., et al. (2022). Fusarium neocosmosporiellum causing peanut pod rot and its biological characteristics. *Acta Phytopathol. Sin.* 52, 493–498. doi: 10.13926/j.cnki.apps.000494

He, J., Liu, T., Li, L., Hu, Y., and Zhou, G. (2023). Mfaster r-cnn for maize leaf diseases detection based on machine vision. *Arab. J. Sci. Eng.* 48, 1437–1449. doi: 10.1007/s13369-022-06851-0

Hua, S., Xu, M., Xu, Z., Ye, H., and Zhou, C. (2022). Multi-feature decision fusion algorithm for disease detection on crop surface based on machine vision. *Neural Comput. Appl.* 34, 9471–9484. doi: 10.1007/s00521-021-06388-7

Lawal, M. O. (2021). Tomato detection based on modified yolov3 framework. *Sci. Rep.* 11, 1447. doi: 10.1038/s41598-021-81216-5

Lawal, O. M. (2023). Yolov5-LiNet: A lightweight network for fruits instance segmentation. *PloS One* 18, e282297. doi: 10.1371/journal.pone.0282297

Lewis, P. I., and Filonow, A. B. (1990). Reaction of peanut cultivars to pythium pod rot and their influence on populations of pythium spp. In soil1. *Peanut Sci.* 17, 90–95. doi: 10.3146/i0095-3679-17-2-11

Li, S., Jia, H., Zhao, J., and Chen, D. (2011). Identification and pathogenicity of peanut pod rot in hebei province, China. *J. Hebei Agric. Sci.* 15, 37–39. doi: 10.16318/j.cnki.hbnykx.2011.05.012

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., et al. (2016). SSD: Single shot multibox detector. *Lecture Notes in Computer Science*. 9905, 21–37. doi: 10.1007/978-3-319-46448-0_2

Liu, X., Fu, D., Yu, F., Yang, W., and Yin, X. (2020). Isolation, characterization and biological properties of peanut pod rot bacteria in hainan province. *Jiangsu Agric. Sci.* 48, 104–107. doi: 10.15889/j.issn.1002-1302.2020.06.021

Lu, J., Liang, X., Yu, C., Lan, Y., Qiu, H., Huang, J., et al. (2022). Fast identification of nematodes based on coordinate attention mechanism and efficient bounding box regression loss. *Trans. Chin. Soc. Agric. Eng.* 38, 123–132. doi: 10.11975/j.issn.1002-6819.2022.22.013

Pydipati, R., Burks, T. F., and Lee, W. S. (2005). Statistical and neural network classifiers for citrus disease detection using machine vision. *Trans. ASAE* 48, 2007–2014. doi: 10.13031/2013.19994

Qiao, S., Chen, L. C., and Yuille, A. (2021). "DetectoRS: Detecting objects with recursive feature pyramid and switchable atrous convolution," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Nashville, TN, USA: IEEE), 10208–10219. doi: 10.1109/CVPR46437.2021.01008

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)*, (Las Vegas, NV, USA: IEEE), 779–788. doi: 10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Trans. On Pattern Anal. Mach. Intell.* 39, 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Shen, Z., and Zhao, Z. (2021). "Improved lightweight peanut detection algorithm based on yolo v3," in *2021 International Conference on Artificial Intelligence, Big Data*

and Algorithms (CAIBDA), (Xi'an, China: IEEE), 171–176. doi: 10.1109/CAIBDA53561.2021.00043

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

Sun, W., Feng, L., Guo, W., Liu, D., Li, Y., and Ran, L. (2012). First report of peanut pod rot caused by neocosmospora vasinfecta in northern China. *Plant Dis.* 96, 455. doi: 10.1094/PDIS-03-11-0240

Sun, W., Feng, L., Guo, W., Liu, D., Yang, Z., Liu, L., et al. (2011). First report of neocosmospora striata causing peanut pod rot in China. *Plant Dis.* 96, 146. doi: 10.1094/PDIS-06-11-0461

Walker, M. E., and Csinos, A. S. (1980). Effect of gypsum on yield, grade and incidence of pod rot in five peanut cultivars1. *Peanut Sci.* 7, 109–113. doi: 10.3146/i0095-3679-7-2-13

Wheeler, T. A., Russell, S. A., Anderson, M. G., Woodward, J. E., Serrato-Diaz, L. M., Frenc-Monar, R. D., et al. (2016). Management of peanut pod rot ii: comparison of calendar and threshold-based fungicide timings. *Crop Prot.* 87, 13–18. doi: 10.1016/j.cropro.2016.04.011

Wynne, J. C., Beute, M. K., and Nigam, S. N. (1991). Breeding for disease resistance in peanut (arachis hypogaea l.). *Annu. Rev. Phytopathol.* 29, 279–303. doi: 10.1146/annurev.py.29.090191.001431

Yu, J., Xu, M., Liang, C., Zhang, X., Guo, Z., Wu, J., et al. (2019). First report of pythium myriotylum associated with peanut pod rot in China. *Plant Dis.* 103, 1794. doi: 10.1094/PDIS-02-19-0321-PDN

Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., and Lee, B. (2022). A survey of modern deep learning based object detection models. *Digit. Signal Process.* 126, 103514. doi: 10.1016/j.dsp.2022.103514

Zhang, Y., Ren, W., Zhang, Z., Jia, Z., Wang, L., and Tan, T. (2022). Focal and efficient iou loss for accurate bounding box regression. *Neurocomputing* 506, 146–157. doi: 10.1016/j.neucom.2022.07.042

Zhang, Q., and Yang, B. (2021). SA-net: shuffle attention for deep convolutional neural networks. *Arxiv* [Preprint]. Available at: https://arxiv.org/abs/2102.00240.

Zhang, C., Zhang, T., Wu, C., and Lu, X. (2016). Identification and biological characteristics of fungal isolates causing peanut pod rot pathogen. *Peanut J.* 45, 27–31. doi: 10.14001/j.issn.1002-4093.2016.03.005

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 01 May 2024
DOI 10.3389/fpls.2024.1387350

OPEN ACCESS

EDITED BY
Mohsen Yoosefzadeh Najafabadi,
University of Guelph, Canada

REVIEWED BY
Parvathaneni Naga Srinivasu,
Prasad V. Potluri Siddhartha Institute of Technology, India
Muhammad Azam,
University of Agriculture, Faisalabad, Pakistan

*CORRESPONDENCE
Guanping Wang
wangguanping@gsau.edu.cn

RECEIVED 17 February 2024
ACCEPTED 15 April 2024
PUBLISHED 01 May 2024

CITATION
Wang L, Wang G, Yang S, Liu Y, Yang X, Feng B, Sun W and Li H (2024) Research on improved YOLOv8n based potato seedling detection in UAV remote sensing images. *Front. Plant Sci.* 15:1387350. doi: 10.3389/fpls.2024.1387350

Research on improved YOLOv8n based potato seedling detection in UAV remote sensing images

Lining Wang, Guanping Wang*, Sen Yang, Yan Liu, Xiaoping Yang, Bin Feng, Wei Sun and Hongling Li

Mechanical and Electrical Engineering College, Gansu Agricultural University, Lanzhou, Gansu, China

Introduction: Accurate detection of potato seedlings is crucial for obtaining information on potato seedlings and ultimately increasing potato yield. This study aims to enhance the detection of potato seedlings in drone-captured images through a novel lightweight model.

Methods: We established a dataset of drone-captured images of potato seedlings and proposed the VBGS-YOLOv8n model, an improved version of YOLOv8n. This model employs a lighter VanillaNet as the backbone network instead of the original YOLOv8n model. To address the small target features of potato seedlings, we introduced a weighted bidirectional feature pyramid network to replace the path aggregation network, reducing information loss between network layers, facilitating rapid multi-scale feature fusion, and enhancing detection performance. Additionally, we incorporated GSConv and Slim-neck designs at the Neck section to balance accuracy while reducing model complexity.

Results: The VBGS-YOLOv8n model, with 1,524,943 parameters and 4.2 billion FLOPs, achieves a precision of 97.1%, a mean average precision of 98.4%, and an inference time of 2.0ms. Comparative tests reveal that VBGS-YOLOv8n strikes a balance between detection accuracy, speed, and model efficiency compared to YOLOv8 and other mainstream networks. Specifically, compared to YOLOv8, the model parameters and FLOPs are reduced by 51.7% and 52.8% respectively, while precision and a mean average precision are improved by 1.4% and 0.8% respectively, and the inference time is reduced by 31.0%.

Discussion: Comparative tests with mainstream models, including YOLOv7, YOLOv5, RetinaNet, and QueryDet, demonstrate that VBGS-YOLOv8n outperforms these models in terms of detection accuracy, speed, and efficiency. The research highlights the effectiveness of VBGS-YOLOv8n in the efficient detection of potato seedlings in drone remote sensing images, providing a valuable reference for subsequent identification and deployment on mobile devices.

KEYWORDS

potato seedling detection, UAV remote sensing, YOLOv8n, lightweight, VanillaNet, GSConv, Slim-Neck

1 Introduction

In recent years, the global cultivation area for potatoes has remained stable at approximately 20 million hectares, with China's contribution consistently exceeding 25% (Shi and Xu, 2023). This makes potato cultivation vitally important for food security, economic growth, and poverty alleviation, particularly in densely populated developing countries such as China (Lun et al., 2023). A critical phase in the potato growth cycle is the seedling stage, where accurate detection and counting of seedlings are crucial for predicting yields and achieving high-quality production (Shi et al., 2022). However, traditional manual monitoring methods are costly, inefficient, inaccurate, and often lack representativeness, which impedes the timely and effective implementation of replanting strategies (Lu et al., 2023). The advent of drones, characterized by their agility, compact size, and cost-effectiveness, has increasingly attracted the attention of researchers (Saifizi et al., 2019; Li S. et al., 2023). Utilizing drones in conjunction with deep learning for the automatic detection of crop seedlings presents a simple yet effective method that significantly reduces labor costs and facilitates automation.

Drone platforms, through real-time imagery captured by onboard cameras, have found extensive applications in various fields for target detection (Osco et al., 2020). However, detecting targets from a drone's perspective often involves dealing with complex environmental backgrounds and small, sometimes blurry, targets. Additionally, the hardware limitations of drones can restrict the complexity of deployable models, leading to less than optimal detection outcomes (Wu et al., 2010; Sishodia et al., 2020). Deep learning algorithms for target detection are generally categorized into two main types: single-stage algorithms, such as Centernet, RetinaNet, SSD, and YOLO, which offer good real-time performance but lower accuracy, particularly in detecting small targets; and two-stage algorithms, like R-CNN, Fast R-CNN, and Faster R-CNN, which provide higher accuracy but at the cost of speed, making them unsuitable for rapid crop information acquisition by drones. The YOLO series, known for its superior performance, has been extensively applied in detection tasks across various domains (Liu et al., 2018; Liang et al., 2022). A current research challenge, and the focus of this study, is leveraging YOLO for accurate and efficient crop seedling detection from a drone's perspective while maintaining a manageable model size.

The YOLO series models have been broadly applied to drone image datasets. For instance, research by Jianqing Zhao et al. (Zhao et al., 2021) introduced an enhanced YOLOv5 model with an added micro-scale detection layer for wheat ear detection in drone images, achieving a 94.1% accuracy rate, a 10.8% improvement over the standard YOLOv5. However, this method is complex and time-consuming, and the limited memory and processing power available on drones make efficient crop detection challenging. Wang et al. (Wang F et al., 2023) addressed the characteristics of small targets in drone images by embedding a small target detection structure (STC) in the Neck of YOLOv8, capturing comprehensive global and contextual information and incorporating a global attention module (GAM), which significantly improved performance but also increased the parameter count. Li et al. (Li Y. et al., 2023) introduced the concept of Bi-PAN-FPN in YOLOv8 to enhance feature fusion across different scales and utilized the GhostblockV2 structure, achieving an accuracy improvement but falling short compared to other models. Addressing the challenges of insufficient drone computing power and the issue of small targets in drone imagery, Shijie Li (Li, 2023) proposed modifications to the YOLOv5 model, reducing the model's parameter count from 7.5M to 4.2M, albeit with a 1.7% decrease in detection accuracy. To address the balance between detection accuracy and model size, scholars have conducted relevant research, proposing the use of lightweight convolutional approaches aimed at reducing computational load during the convolution process. For example, Liu et al. (Liu et al., 2022) proposed an improved YOLOv4 model based on MobileNetv2 as the backbone network for orange fruit recognition in orchards, which reduced the model size by 197.5 M and achieved an average recognition accuracy of 97.24%, though the detection time was only reduced by 11.39ms. Rihong Zhang et al (Zhang et al., 2023). introduced a YOLOV4 pineapple seedling heart detection model incorporating a lightweight attention mechanism module CBAM, which reduced the total parameter count by 70% and achieved a recognition accuracy of 95.5%, but the improvement in detection speed was not significant.

While previous methods have shown effectiveness in detecting and counting crops in the field, the unique challenges posed by potato seedlings in UAV imagery—such as their dense distribution, significant overlap, small size, and the complexity of their background, result in a higher likelihood of both false positives and missed detections. These issues compromise the precision of potato seedling detection. Furthermore, the constraints imposed by UAV hardware platforms complicate the task of balancing detection accuracy, speed, and the efficient use of hardware resources. Notably, there is a scarcity of detection methods that are both efficient and specifically tailored to potato seedlings. To address these challenges, this paper introduces a novel lightweight algorithm, VBGS-YOLOv8n. By employing VanillaNet, a network characterized by its simplicity and reduced number of layers, as the backbone network in place of the original YOLOv8n model, we significantly decrease the model's computational complexity. We enhance the model's feature fusion capabilities by substituting the PANet path aggregation network with a bidirectional feature pyramid network (BiFPN). Additionally, integrating GSconv convolution within the YOLOv8n's neck and replacing all C2F networks with the VoV-GCSP module further boosts the model's performance. This innovative approach facilitates the efficient detection of potato seedlings in UAV remote sensing images, representing a significant advancement in the field.

2 Materials and methods

2.1 Potato seedling image acquisition

Potato seedling drone images were collected at Xinghuaping Village, Tonganyi Town, Longxi County, Dingxi City, Gansu Province. The images were captured using a quadcopter drone (DJI Phantom 4 Advanced) and DJI GS Pro. The drone's RGB camera captured images vertically from above with a shutter speed of 2 seconds. To prevent image blurring, a hover-and-capture

method was employed at each waypoint. The front and side overlaps were set at 80% and 70% respectively. The images had a resolution of 4056×3040 pixels and were saved in JPG format. The image collection took place in mid-May and mid-June 2022, between 10:00-12:00. To enhance the model's ability to generalize for potato seedling detection in various environments, images were collected at drone heights of 5 meters and 10 meters. A total of 409 original images were collected, as shown in Figure 1, covering different heights, growth stages and plots.

2.2 Dataset construction

The process of potato seedling RGB image detection using the enhanced VBGS-YOLOv8n model is illustrated in Figure 2. In this study, Pix4Dmapper software was utilized for rapid stitching and inspection of drone images in the experimental area. During the stitching process, location information was obtained using the GPS system of the drone platform at the time of image capture. Pix4Dmapper then matched approximately 30,000 tie points per original image based on the flight's POS (Position and Orientation System) data. Subsequently, automatic aerial triangulation technology was employed to calculate the true position data and stitching numbers of the images, leading to the creation of a point cloud model. Following this, the positions and stitching parameters of the original images were automatically optimized and calibrated to generate a Digital Orthophoto Map (DOM) depicting the entire experimental plot (Figure 2B). The process resulted in orthophoto images at heights of 5 meters and 10 meters (Figure 2C) for two distinct periods. These orthophoto images were then cropped to obtain the dataset images required for model training and prediction (Figure 2D). A total of 3089 cropped images were obtained, each with a pixel size of 800×800. To ensure model detection accuracy, 2195 images were selected after screening out unsuitable ones to form the dataset for this study. Manual annotation of the dataset using the LabelImg annotation tool was performed (Figure 2E). Subsequently, the improved model (Figure 2F) was trained, and the best model after training was used to detect images in the experimental plots (Figure 2G), yielding the detection results (Figure 2H). During annotation, objects were labeled with

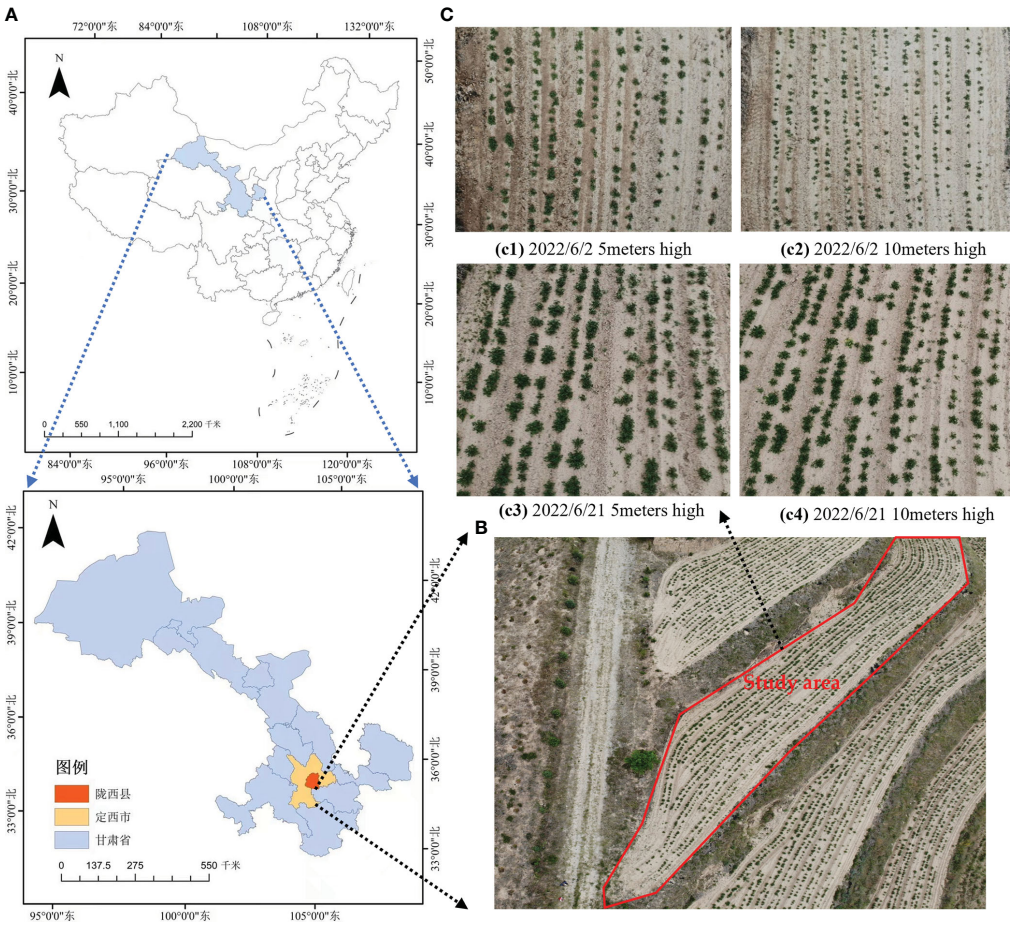


FIGURE 1
Overview of experimental area and captured images. **(A)** The geographical location of Longxi County, Ding xi City; **(B)** Location of the study area; **(C)** Images of potato seedlings at different heights and growth stages of UAVs.

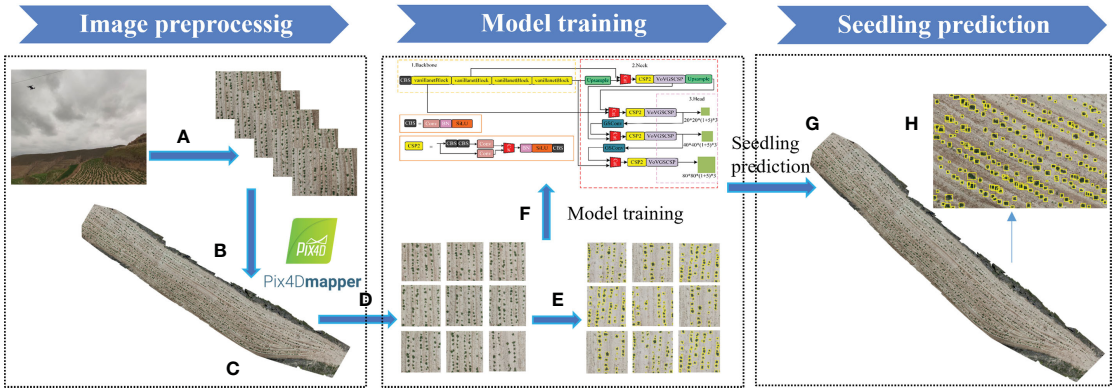


FIGURE 2
Workflow for image preprocessing and model prediction. **(A)** images taken by UAVs; **(B)** Stitching the images taken by the UAV using Pix4d software; **(C)** Orthophoto generated; **(D)** The large image is cropped into a small image (608 × 608 pixels) for model input; **(E)** annotated image; **(F)** model training; **(G)** The result image of the model prediction output; **(H)** A magnified view of the output image.

bounding boxes that best fit them and assigned the label "seedling," resulting in the generation of XML files in VOC format. Refer to Figure 3 for annotated illustrations. Subsequently, the XML files were converted to TXT files required by YOLO using a script. The dataset images and their corresponding TXT files were randomly divided in an 8:1:1 ratio into training set (1754 images), validation set (220 images), and test set (220 images) to adhere to the standard coco format, completing the dataset construction.

2.3 Original YOLOv8n

As a one-stage object detection algorithm, YOLOv8 introduces a more lightweight network structure compared to its predecessors, maintaining high accuracy while achieving faster inference speeds. Moreover, YOLOv8 incorporates advanced training methods and techniques, leading to shorter training times and quicker convergence rates. In this study, to balance high detection accuracy with minimal storage usage and enhanced recognition speed for future deployment on mobile devices, the research opts for the YOLOv8n detection model known for its low complexity and lightweight design.

The YOLOv8n network architecture comprises three main components: the input layer (Input), the backbone network (Backbone), the neck network (Neck), and the detection head (Head). The input layer preprocesses image inputs for the model, while the backbone network, based on CSPDarkNet-53 and utilizing the C2f module, extracts features from input images to generate multi-scale feature maps. The backbone structure is shown in Figures 4A, B is a CBS structure diagram. The C2f module in YOLOv8 provides feature fu-sion functionality, which can enhance the performance of object detection, as illustrated in Figure 4C. The convolution utilizes CBS, comprising three components: a 2D convolution, 2D BatchNorm, and SiLU activation function. The SiLU activation is computed by multiplying its input with the sigmoid function, i.e., xσ (x). In the case of SPPF, a CBS convolutional layer is followed by three consecutive Maxpooling operations. The feature map without Maxpooling and the feature map obtained after each subsequent Maxpooling operation are concatenated to achieve feature fusion. The structure is shown in Figure 4D. The Neck layer adopts the PANet structure, merging feature maps from various scales to capture more global and semantically rich features, thereby enhancing object detection accuracy and recall. The Detect module employs a Decoupled Head, separating regression and prediction branches to predict features across three dimensions, providing class and positional information for the network's predictions.

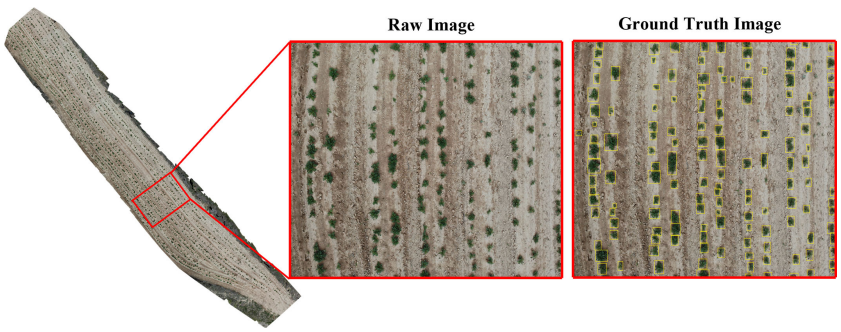


FIGURE 3
An example of a labeled image used for model training.

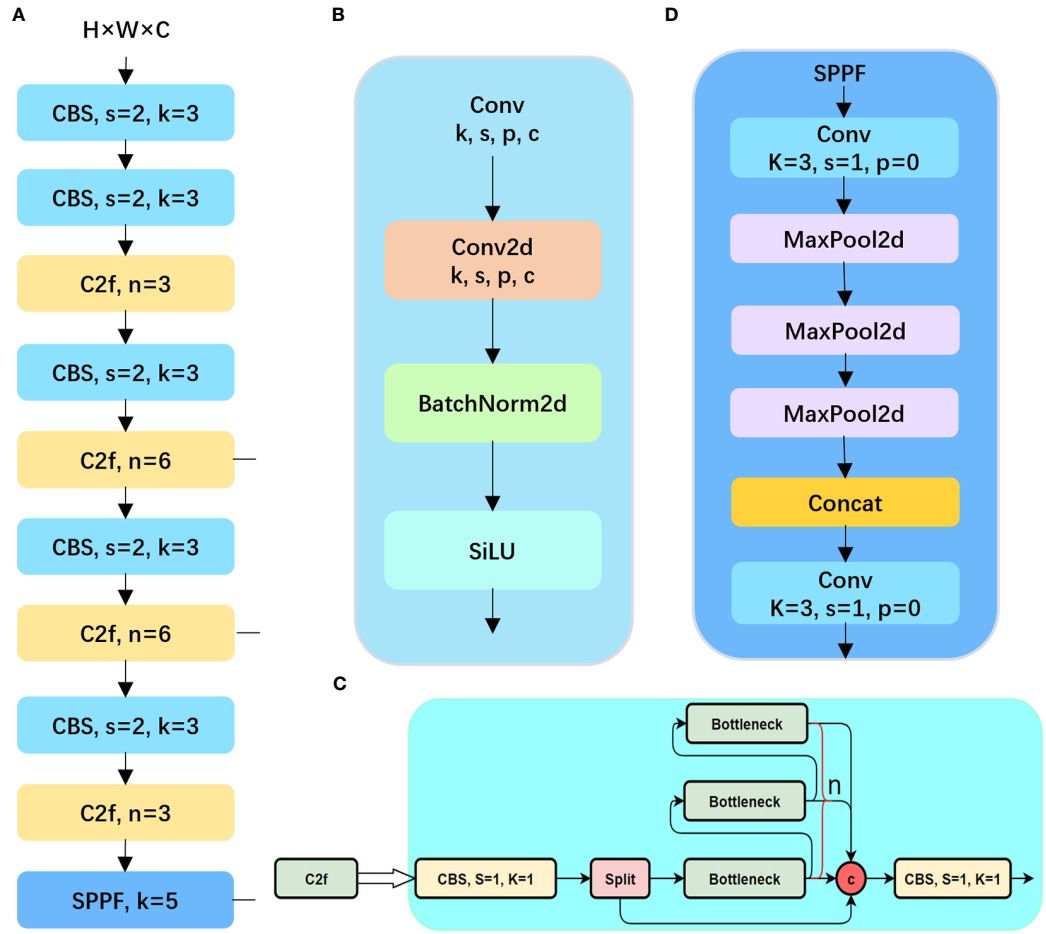


FIGURE 4
The backbone structure of the yolov8 model and the diagram of each module. **(A)** the overall structure of the backbone; **(B)** the structure of the CBS module; **(C)** the C2f module; **(D)** and the SPPF module.

In the YOLOv8 model, the loss function plays a crucial role in training the network to accurately predict object bounding boxes and class probabilities. The loss function used in YOLOv8 is a combination of localization loss, confidence loss, and classification loss. The localization loss in YOLOv8 is typically calculated using metrics like Mean Squared Error (MSE) or Smooth L1 Loss. It penalizes the model for inaccuracies in predicting the bounding box coordinates (center coordinates and width/height) compared to the ground truth bounding box. By minimizing the localization loss, the model learns to accurately predict the spatial location and size of objects in the image, improving the precision of object localization. Next, YOLOv8 utilizes binary cross-entropy loss to compute the target confidence loss, assessing the model's confidence accuracy by comparing predicted target probabilities with ground truth labels. Optimizing the confidence loss enables the model to distinguish objects from the background, enhancing its object detection capabilities. Additionally, the classification loss evaluates the model's category classification accuracy using binary cross-entropy loss. The calculation formula for classification loss is shown in Equation (1). About Regression Loss, YOLOv8 introduces a Distance-based Focal Loss (DFL) to complement Anchor-Free methods, focusing on optimizing probabilities for the nearest left and right positions to the label y, facilitating quicker convergence on target positions and neighboring regions' distributions. DFL is calculated as shown in Equation 2.

$$Loss_{cls} = -\sum_{c=1}^{M} y_{o,c} log(p_o, c) \tag{1}$$

where $y_{o,c}$ is an indicator. 1 if sample o belongs to category c, and 0 vice versa. p_o is the probability that the model predicts that sample o belongs to category c.

$$\mathrm{DFL}(S_i, S_{i+1}) = -((y_{i+1} - y)\log(S_i) + (y - y_i)\log(S_{i+1})) \tag{2}$$

The detailed conversion process of transforming labels into DFL format is as follows: y = distance from the center to a specific edge/current downsampling ratio.

The Bounding Box Loss calculates the sum of squared differences between the predicted and actual coordinates, as depicted in Equation 3.

$$Loss_{bbox} = \sum_{i=1}^{N} (x_i - \hat{x}_i)^2 \tag{3}$$

where x_i represents the coordinates of the true bounding box, and $\hat{x}_i$ represents the coordinates of the predicted bounding box.

The loss function is used as the optimization goal to guide the model to reduce the gap between the prediction box and the real box during the training process.

2.4 Improvement of the YOLOv8n model

2.4.1 VBGS-YOLOv8n model structure

The YOLOv8n object detection model has been widely applied in the agricultural field due to its excellent recognition accuracy and speed (Sapkota et al., 2023; Wang G et al., 2023). However, the detection of potato seedlings poses some challenges as it involves small target detection tasks. For instance, when deploying the detection task to mobile devices, it is necessary to consider the lightweight nature of the network structure and the reduction of device power consumption. Additionally, due to the small size and overlapping nature of potato seedlings captured by UAVs, there is a risk of missed detections and low accuracy in small target detection. Therefore, this paper proposes a VBGS-YOLOv8n deep learning algorithm based on the YOLOv8n, aiming to achieve higher detection accuracy and a more lightweight model design to better recognize potato seedlings. First, lightweight improvements were made to the backbone, followed by the introduction of the weighted bidirectional feature pyramid network (BiFPN) at the Neck layer, along with the GSConv network, replacing the c2f module with VoV-GSCSP.

The structural design of the proposed VBGS-YOLOv8n model, as depicted in Figure 5, involves replacing the CSPDarkNet network of the original YOLOv8 with the lightweight VanillaNet algorithm. The backbone network comprises the initial 4 layers of VBGS-YOLOv8n, starting with a 640*640 RGB image input. With a stride of 4 and double downsampling, spatial feature extraction and data normalization convolution processing are applied, resulting in a halved image resolution. This processed image is then fed into the VanillaNet backbone network. Within the backbone network, stages 1, 2, and 3 utilize max-pooling layers with a stride of 2 to reduce spatial dimensions while retaining crucial feature information, doubling the channel count at each layer. Stage 3, representing the third layer of the network, undergoes an 8x downsampling to yield an image with 512 channels. Stage 4 maintains the channel count without increase, following an average pooling layer. The final layer consists of a fully connected layer for classification output with a stride of 1. Each layer in the VanillaNet backbone network employs 1x1 convolution kernels to preserve feature map details efficiently. The input features are downsampled to appropriate sizes, resulting in image resolutions of 160*160, 160*160, and 80*80 at Layer 1, Layer 2, and Layer 4, respectively.

The 1st, 3rd, and 4th layers serve as inputs to the neck structure. In contrast to the PANet bidirectional pathway network used in the original YOLOv8n network's neck structure, the VBGS-YOLOv8n model integrates a BiFPN with adjustable weights in each concat module of the neck network for feature extraction. The BiFPN facilitates more efficient multi-scale feature fusion. Furthermore, the c2f modules at each layer are replaced with the cross-level subnetwork VoV-GSCSP module. Additionally, GSConv convolution is applied at the 11th and 14th layers of VBGS-YOLOv8n, aiming to reduce computational costs and maintain inter-channel connections effectively. Through a process of layer-wise upsampling and feature concatenation, diverse scale feature information is fused. By the 16th layer of the model, the number of output channels in the image is increased to 1024.Subsequently, the three output branches from the neck are directed to the detection head for loss computation or result inference. YOLOv8 introduces a decoupled head, replacing the coupled head of previous YOLO models. This decoupled head separates the regression and prediction branches, utilizing the integral form proposed in the distribution focal loss strategy for the regression

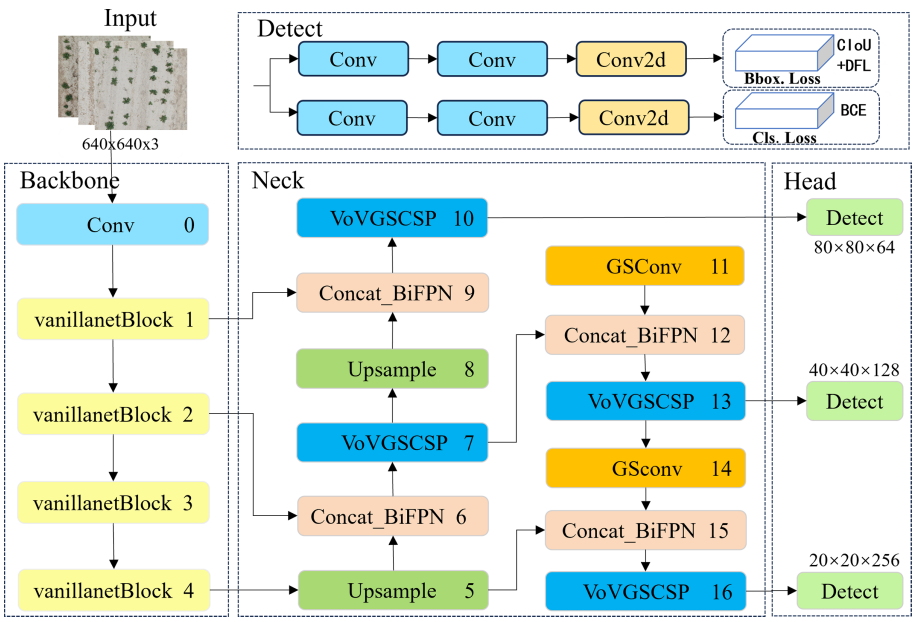


FIGURE 5
The network architecture diagram of the improved VBGS-YOLOv8n.

branch. The decoupled head exhibits faster convergence and improved performance. In VBGS-YOLOv8n, the head network generates images of sizes 80×80, 40×40, and 20×20 for potato seedling detection.

2.4.2 Lightweight backbone network

VanillaNet, a lightweight neural network architecture that emphasizes simplicity, was introduced by Huawei's Noah's Ark Lab (Chen et al., 2023). By avoiding complexities like excessive depth, shortcuts, and self-attention mechanisms, VanillaNet achieves a balance of simplicity and performance. Overcoming the inherent complexity of traditional deep networks, VanillaNet emerges as an optimal choice for environments with limited resources. Its streamlined architecture not only enhances comprehension but also provides an effective solution for efficiently deploying potato seedling detection in drone-based remote sensing applications.

VanillaNet is characterized by the absence of convolution layers and branches in its network structure, as depicted in Figure 6. The network comprises a backbone, main body, fully connected layers, and 5 activation functions. The design principle follows a gradual reduction in resolution and an increase in channel numbers, without incorporating shortcuts, attention mechanisms, or other computations.

For the backbone, a 4×4×3×C convolution layer is utilized with a stride of 4, following common configurations from [18,31,32], to transform 3-channel images into features with C channels. In stages 1, 2, and 3, max-pooling layers with a stride of 2 are used to decrease size and feature maps while doubling the channel count. Stage 4 maintains the channel count unchanged by employing average pooling. The final fully connected layer is dedicated to producing classification outcomes. Each convolution layer employs a 1×1 kernel to retain feature map details while minimizing computational costs. Batch Normalization (BN) is applied after each layer to streamline the training process and enhance the simplicity of the architecture. This approach achieves an optimal trade-off between speed and accuracy, showcasing the excellence of VanillaNet.

While VanillaNet's simple structure is easy to implement, its limited nonlinearity hinders network performance enhancement. To tackle this challenge, the authors introduce a deep training strategy and incorporate a series-inspired activation function to boost the network's nonlinear expressive capacity.

The deep training strategy involves splitting the network into two convolution layers, increasing the network depth only during training, and merging them during inference. This approach reduces network computation and complexity. The split convolution layers will utilize the following Equation 4 activation function:

$$A^{'}(x) = (1-\lambda)A(x) + \lambda x \tag{4}$$

When training converges, the two convolutional layers without non-linear activation are merged into one layer, achieving the effect of deep training and shallow inference.

(1) Activation Function Inspired by Series: Concurrently stacking activation functions can significantly enhance the non-linearity of the activation function. Representing the single activation function of the input in the neural network as $A(x)$ Equation 5:

$$A_s(x) = \sum_{i=1}^{n} a_i A(x + b_i) \tag{5}$$

In the equation, n represents the number of stacked activation functions, while a_i, b_i are the scale and bias of each activation to avoid simple accumulation. To further enrich the sequence, given an input feature $x \in R^{H\times W\times C}$ where H, W and C are its width, height, and number of channels, the activation function is formulated as Equation 6:

$$A_s(x_h, w, c) = \sum_{i,j\in\{-n,n\}} a_{i,j,c} A(x_{i+h,j+w,c} + b_c) \tag{6}$$

From the equation, it can be found that when $n = 0$, the proposed method can be regarded as a general extension of existing activation functions.

The computational complexity expression of the proposed activation function O(CONV)compared to its corresponding convolutional layer is shown in Equation 7).

$$\frac{O(\text{CONV})}{O(\text{SA})} = \frac{H \times W \times C_{\text{in}} \times C_{\text{out}} \times K^2}{H \times W \times C_{\text{in}} \times n^2} = \frac{C_{out} \times k^2}{n^2} \tag{7}$$

In the equation, C_{in} represents the input channels, C_{out} represents the output channels, and k represents the kernel size. Taking the fourth stage of VanillaNet-B as an example, where $C_{out} = 2048$, $k = 1$, $n = 7$, the ratio is only 84, indicating that the computational cost of this activation function is much lower than that of a convolutional layer. Therefore, the use of these two non-linear solutions can significantly improve the detection accuracy of VanillaNet.

2.4.3 BiFPN feature fusion

Feature fusion is a critical aspect in object detection, aiding in the extraction of information from various scales to enhance detection

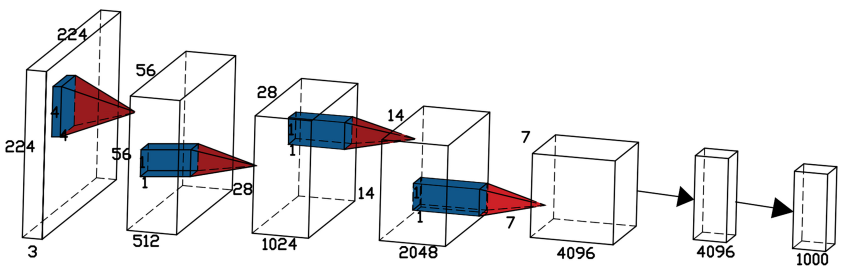


FIGURE 6
The architecture of the VanillaNet-6 consisting of only 6 convolutional layers.

accuracy. The traditional Feature Pyramid Network (FPN) structure serves as a method for feature fusion, integrating a top-down pathway to merge multi-scale features from levels 3 to 7 (P3 to P7), as depicted in Figure 7. Expanding on FPN, the YOLOv8 feature extraction network incorporates PANet (Figure 7B), which introduces an additional bottom-up pathway aggregation network to FPN (Figure 7A). However, these fusion methods can lead to information loss or feature redundancy (Wang Y et al., 2023). This study introduces an efficient BiFPN (Figure 7C) structure that leverages effective bidirectional cross-scale connections and weighted feature fusion. By adjusting feature map scales through upsampling and downsampling operations, different scale features are fused to preserve finer details, thereby improving small object detection accuracy.

BiFPN (Tan et al., 2020) is a network structure that efficiently incorporates repeated bidirectional cross-scale connections and weighted feature fusion. In comparison to PANet, BiFPN eliminates nodes with single input edges that do not merge different features, making it lighter and faster in inference speed with fewer parameters. Additionally, an extra edge is introduced between the original input and output nodes at the same layer to enhance the fusion of additional image features. By leveraging bidirectional repeated connections for information fusion, feature details are preserved, enhancing accuracy in small object detection. BiFPN utilizes a weighted feature fusion mechanism that differentiates and merges various input features through learning, adapting to different resolutions, and addressing feature loss issues caused by simple overlaying of feature maps. It serves as a straightforward and efficient feature fusion approach. BiFPN adopts the Fast Normalized Fusion method, akin to Softmax, mapping each input value to the range [0, 1], thereby improving training speed and efficiency, enhancing data consistency and comparability for better analysis and decision-making, as depicted in Equation (8).

$$O = \sum_i \frac{w_i \star I_i}{\varepsilon + \sum_j w_j} \tag{8}$$

In the equation, I_i represents the input features, w_i and w_j denote the weights obtained during network training, $\varepsilon = 0.0001$.

2.4.4 GSConv network and Slim-Neck design paradigm

In order to achieve real-time object detection on mobile devices, reducing model complexity, enhancing detection speed, and maintaining high accuracy are essential for the task of potato seedling image detection captured by drones. GSConv+Slim-Neck is a lightweight network proposed for a vehicle-mounted edge autonomous driving computing platform (Li et al., 2022). This network design aims to facilitate efficient object detection to meet real-time application requirements. GSConv strikes a balance between model accuracy and speed, enabling model lightweighting while preserving accuracy. Introducing GSConv provides a design paradigm called Slim-Neck, which utilizes a one-time aggregation method to create the cross-level subnetwork (GSCSP) module VoV-GSCSP. This module reduces computational and network structural complexity, thereby enhancing detection accuracy. Hence, this paper adopts this network to reduce model complexity, enhance detection speed, and maintain high accuracy for mobile deployment, offering an effective solution.

On edge devices, achieving real-time lightweight detection with large models poses challenges. Traditional Depthwise Separable Convolution (DSC) models struggle to achieve high accuracy due to the separation of channel information during computation. This separation diminishes the feature extraction and fusion capabilities of DSC, hindering lightweight high-precision detection. Therefore, GSConv is proposed, merging standard convolution with Depthwise Separable Convolution. The principle involves downsampling with a regular convolution, followed by DWConv depthwise convolution to fuse the results of SCconv and DSCconv, and finally introducing shuffle operations to combine corresponding channels. The structure is illustrated in Figure 8.

GSConv has a noticeable impact on lightweight models. Given that the Neck receives feature maps with maximal channel capacity and minimal spatial dimensions, this paper employs GSConv within the

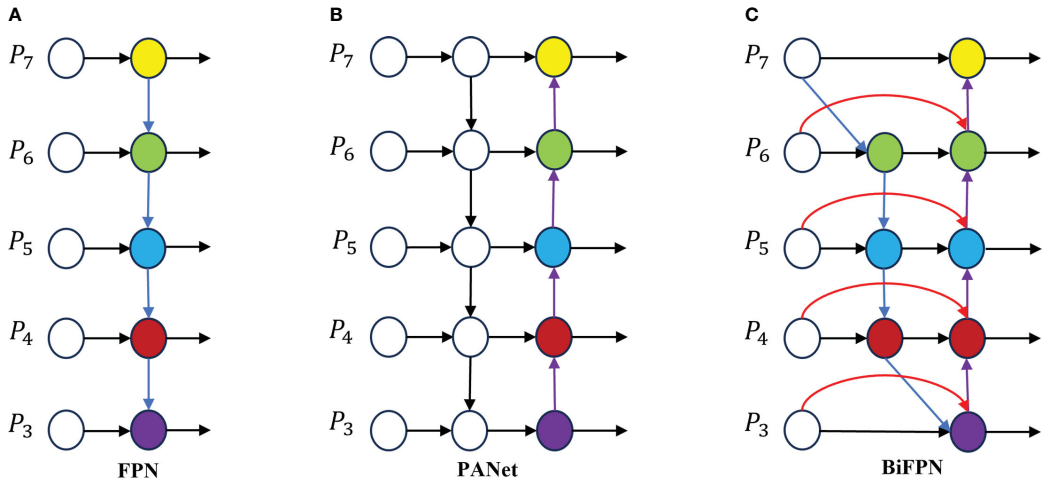


FIGURE 7
Feature network design **(A)** FPN network; **(B)** the principle of PANet; **(C)** is BiFPN schematic.

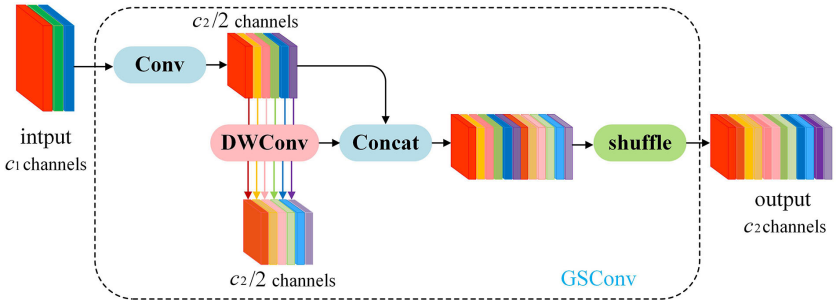


FIGURE 8
The structure of the GSConv module.

Neck. With reduced redundant information in the feature map at this stage, compression is unnecessary, allowing the attention module to operate more effectively, leading to a reduction in model layers and inference time.

Introducing GSConv provides a Slim-Neck design paradigm. Initially, this design replaces SC with the lightweight convolution method GSConv in the Neck. GSConv aims to closely match the convolutional computing capability of SC while reducing computational costs. Subsequently, GSbottleNeck is introduced based on GSConv. Similarly, a one-time aggregation method is utilized to design the cross-level subnetwork (GSCSP) module VoV-GSCSP, which simplifies computational and network structural complexity, enhancing detection accuracy. The structure is depicted in Figure 9. This paper replaces the C2f module in the

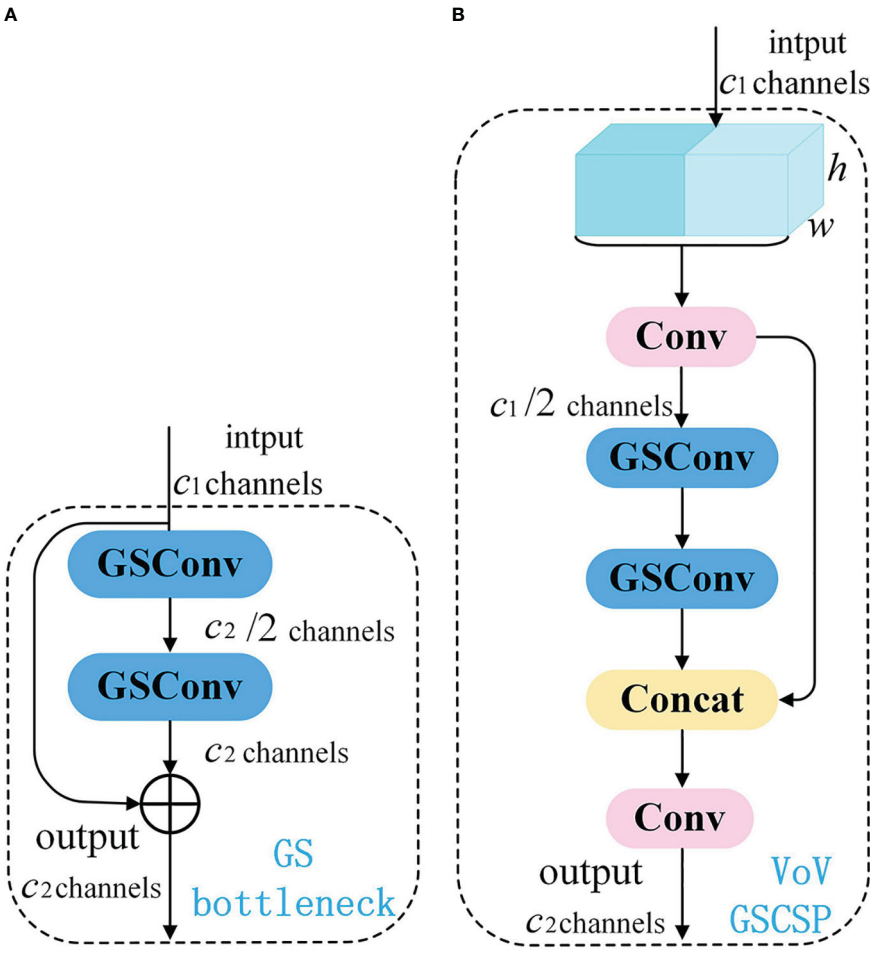


FIGURE 9
Schematic diagram of Slim-neck paradigm design structure. **(A)** The structures of the GS bottleneck module; **(B)** The VoV-GSCSP modules.

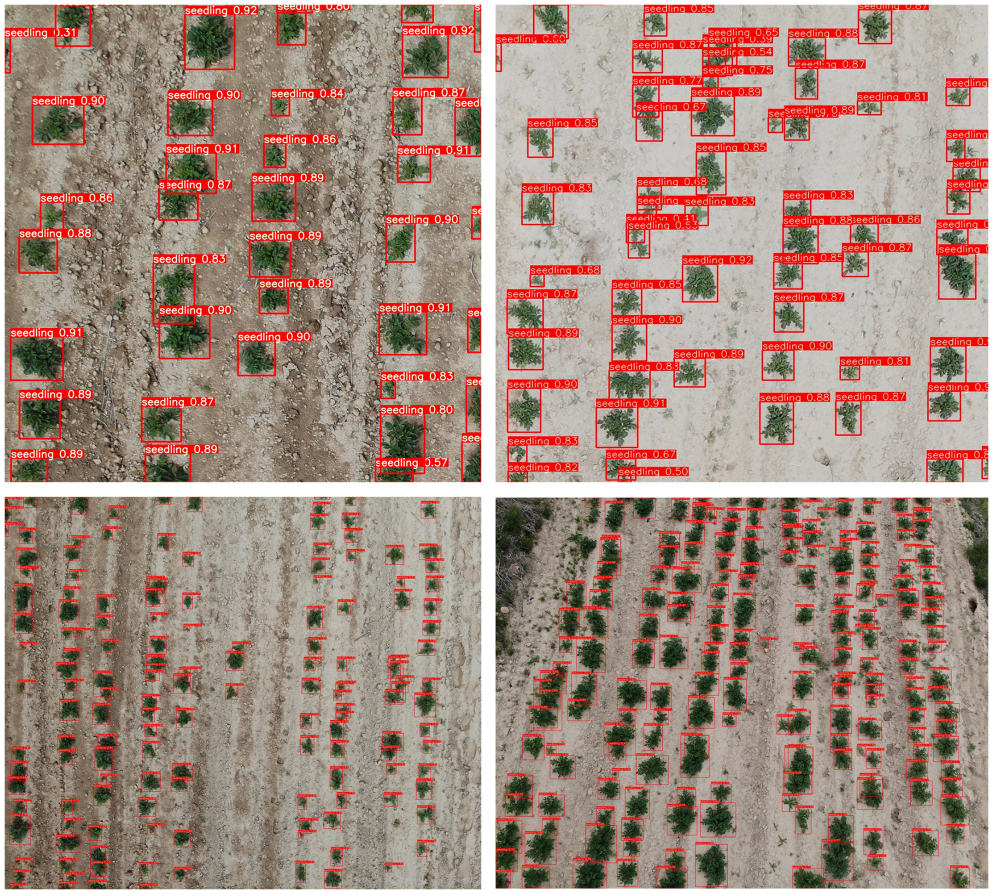


FIGURE 10
Effect of the detection results after the model is introduced into the BiFPN+GSConv+Slim-neck module.

YOLOv8 structure with the VoV-GSCSP module to enhance detection performance. After integrating the BIFPN+GSConv +Slim-Neck module, the detection results are illustrated in Figure 10.

The detection results demonstrate that the model incorporating BIFPN and GSConv+Slim-Neck achieves high confidence scores when detecting images of seedlings in different environments and growth stages. Nearly all seedling targets are successfully identified, highlighting the feasibility and effectiveness of this improvement method.

2.5 Model training and evaluation metrics

2.5.1 Experimental environment

The configuration of the experimental environment and the settings of relevant parameters during the trial process are presented in Table 1.

2.5.2 Evaluation metrics

This study employs Precision (P) in Equation 9, Recall (R) in Equation 10, Mean Average Precision (mAP) as model accuracy evaluation metrics as in Equation 11, and uses parameters, computation, (i.e., the number of floating-point operations), and Detection Time to measure model complexity and speed. The calculation formulas are as follows.

$$P = \frac{TP}{TP + FP} \times 100\,\% \tag{9}$$

$$R = \frac{TP}{TP + FN} \times 100\,\% \tag{10}$$

$$\mathrm{mAP} = \frac{\sum_{i=1}^{N} \int_{0}^{1} P(R)\mathrm{d}R}{N} \times 100\,\% \tag{11}$$

TP represents the number of correctly detected potato sprouts in the image; TN represents the number of instances where the model predicts a negative class and the actual label is also negative. FP stands for the count of false detections as potato sprouts; FN indicates the number of missed targets; AP is the Average Precision, represented by the area enclosed by the P-R ($\varepsilon = 0.0001$) curve and the coordinate axis; N denotes the number of categories. In this study, only potato sprouts are detected, hence $N = 1$.

3 Results and analysis

3.1 VBGS-YOLOv8n ablation experiment

The VBGS-YOLOv8n model proposed in this study adopts a three-step improvement strategy. Firstly, the BiFPN bidirectional

TABLE 1 Experimental environment and related parameter settings.

Training Environment	Details
Programming	Python3.9
Deep learning framework	Pytorch 2.0
GPU	NVIDIA GeForce RTX3060
Operating system	Windows11
img size	640 x 640

feature pyramid network replaces the PANet pathway aggregation network to enhance feature fusion capabilities and improve small object detection accuracy. Secondly, the GSConv+Slim-Neck is integrated into the Neck section to further enhance model performance. Lastly, to achieve model lightweighting, the main network in the Backbone layer is replaced with the VanillaNet network. To validate the effectiveness of the VBGS-YOLOv8n model in potato seedling detection, this study conducted 7 sets of ablation experiments, with results shown in Table 2. Additionally, the training process curve of the model is illustrated in Figure 11.

From the data in Table 2, it is evident that introducing the BiFPN module alone in the original model improves the model's detection accuracy, recall rate, and mAP value by 1.1, 0.5, and 0.8 percentage points, respectively, albeit with a slight increase in model parameters. When adopting the Gsconv+SlimNeck design paradigm alone, compared to the original YOLOv8n, the model with this module shows an increase of 1.4 and 0.6 percentage points in accuracy and mAP value, respectively. Additionally, the model's parameter count decreases by 11.3%, computational load significantly reduces, and inference speed improves by 13.8%, indicating a notable enhancement in detection accuracy and model performance. Furthermore, replacing the Backbone network of the original YOLOv8n model with the lightweight VanillaNet network substantially reduces model parameters and computational load, with a 0.2 percentage point increase in accuracy. However, this change leads to a decrease of 0.3 and 0.1 percentage points in recall rate and mAP, respectively. This is attributed to VanillaNet's lightweight design, which greatly reduces the number of convolutional layer channels and network depth, resulting in decreased performance when handling complex scenes or small targets, thereby impacting recall rate and mean average precision in object detection.

By integrating three improvement strategies, the final outcome of this study is the VBGS-YOLOv8n model. Compared to the original YOLOv8n model, the VBGS-YOLOv8n model shows improvements of 1.4 and 0.8 percentage points in accuracy and mAP, respectively. Additionally, it significantly reduces model parameters and computational load while enhancing inference speed. Specifically, the parameter count is only 48.3% of the original model, the computational load is 47.2% of the original model, and the inference speed increases by 45.0%. However, due to the adoption of the lightweight VanillaNet network, the model's recall rate decreases by 0.6 percentage points. Nevertheless, considering the study's focus on potato seedling monitoring, the slight decrease in recall rate, alongside the improved mAP and reduced model complexity, can be deemed negligible in terms of overall effectiveness.

3.2 Comparison of detection before and after improvement

The original YOLOv8n network and the improved VBGS-YOLOv8n model were compared on a test set of 220 images. One image of potato seedlings was randomly selected from three different scenarios with varying heights and environmental conditions for demonstration of the detection performance, as shown in Figure 12.

The detection results demonstrate the superiority of the VBGS-YOLOv8n model in recognizing various sizes and shapes of potato seedlings, surpassing the original YOLOv8n model significantly. The VBGS-YOLOv8n model can almost entirely identify targets, successfully avoiding instances of missed detections and even detecting overlapping potato seedlings independently. In contrast, the original YOLOv8n model exhibits noticeable issues with missed detections, particularly for smaller potato seedlings in multi-target scenarios, and performs poorly in identifying overlapping potato seedlings.

TABLE 2 Comparison of ablation experiment performance.

Model	BiFPN	Gsconv+ slimNeck	VanillaNet	Precision (%)	Recall (%)	mAP (%)	Parameters (M)	Complexity (GFLOPs)	Inference time (ms)
baseline				95.7	96.8	97.6	3157200	8.9	2.9
A	√			96.8	97.3	98.4	3157212	8.9	3.0
B		√		97.1	96.8	98.2	2801619	7.4	2.5
C	√	√		97.0	97.8	98.5	2801631	7.4	2.7
D			√	95.9	96.5	97.5	1644579	5.0	2.3
E	√		√	96.4	96.7	98.0	1644591	5.0	2.4
VBGS-YOLOv8n	√	√	√	97.1	96.2	98.4	1524943	4.2	2.0

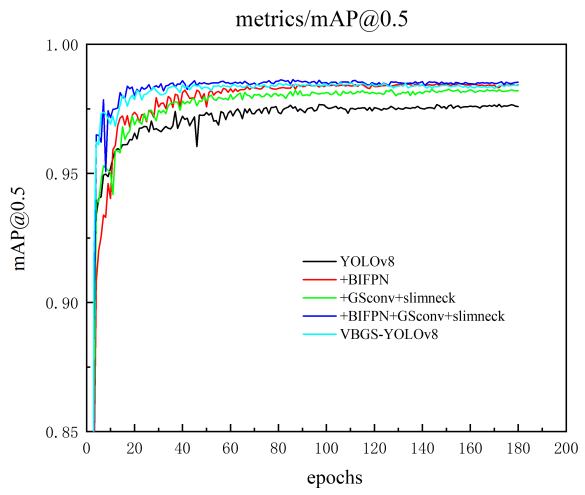


FIGURE 11
Curve of the model improvement training process.

3.3 Comparative horizontal experiment

To further explore the superiority of the VBGS-YOLOv8n network in potato seedling detection, experimental comparisons were conducted between the VBGS-YOLOv8n model and mainstream object detection Network algorithms such as RetinaNet, QueryDet, YOLOv5 and YOLOv8n, as shown in Table 3.

From the table data, it is evident that compared to mainstream models, the VBGS-YOLOv8n network surpasses current mainstream detection models in all performance metrics, with a significant improvement in mAP. More importantly, while maintaining high performance, the VBGS-YOLOv8n model has the lowest parameter count and computational load, further highlighting its superiority and efficiency. RetinaNet, despite using FPN and a new focal loss function to enhance model efficiency and run on low-end devices, faces accuracy issues in small object detection and has high computational load, making it unsuitable for this experiment. QueryDet, a small object detection model that accelerates feature pyramid object detector inference speed using a novel query mechanism, employs the Sparse Cascaded Query (CSQ) mechanism to obtain high-resolution feature maps while minimizing computation on background regions. Comparing QueryDet to RetinaNet in the table data, QueryDet shows improvements in all metrics, with optimal parameter and computational load compared to other mainstream models, with computational load only 3.54 points higher than the VBGS-YOLOv8n model in this study. However, its detection accuracy is 8.3 percentage points lower than the model in this study. YOLOv5, another model in the YOLO series widely used for its good performance and detection results, shows comparable detection accuracy to the method in this study but with increased complexity and lower inference speed, making it unsuitable for mobile deployment and potato seedling detection. YOLOv7-tiny, the latest algorithm in the YOLO series, achieves decent accuracy with fewer parameters and computational load, but its FPS is 48% lower than the proposed new method, indicating slower model detection speed. The experimental data comparison underscores the superiority and efficiency of the VBGS-YOLOv8n network, which not only meets the accuracy requirements but also features a more lightweight network architecture suitable for potato seedling detection scenarios. The comparative detection performance of different models is illustrated in Figure 13.

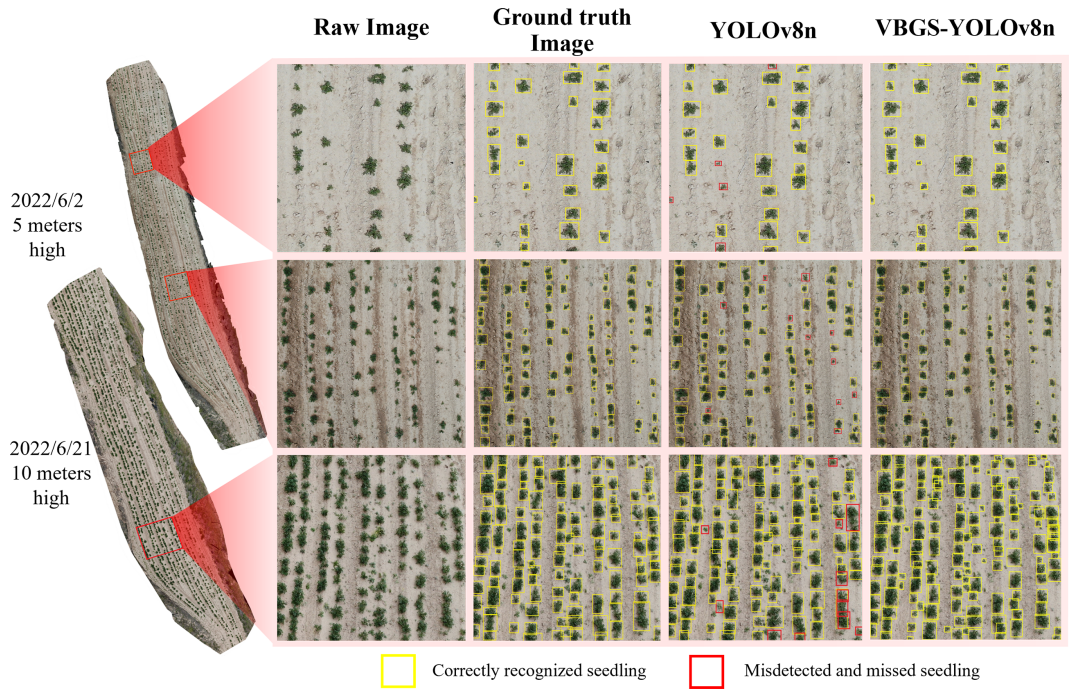


FIGURE 12
Comparison effect of the model before and after the improvement on the detection of potato seedlings at different heights and at different stages.

TABLE 3 Comparison of experimental results of different network models.

Model	mAP (%)	Parameters ($\times 10^6$ M)	Complexity (GFLOPs)	FPS
RetinaNet	82.1	28.27	236.28	29.8
QueryDet	90.3	6.61	7.74	37.4
YOLOv5s	95.8	7.20	16.80	68.3
YOLOv7-tiny	94.3	8.90	13.1	51.5
YOLOv8n	97.9	3.16	8.7	90.1
VBGS-YOLOv8n	98.4	1.52	4.2	98.4

The results indicate that the improved lightweight model outperforms other object detection models in recognizing potato seedlings at different growth stages and heights. It accurately locates potato seedlings, which are dense small targets. In the images, the detection labels and confidence scores were removed for clarity, but in the experiment, detections exhibited high confidence. The predicted bounding boxes fully encapsulate the potato seedlings, even identifying overlapping instances without any missed detections. In the case of the first set of photos with fewer targets at a height of 5 meters, where the potato seedlings are larger and less dense, both YOLOv5 and YOLOv7 in the YOLO series can detect all targets effectively. However, YOLOv7 shows some instances of redundant bounding boxes, indicating slightly inferior detection performance compared to YOLOv5. For small targets at the corners, QueryDet exhibits some missed detections. In the

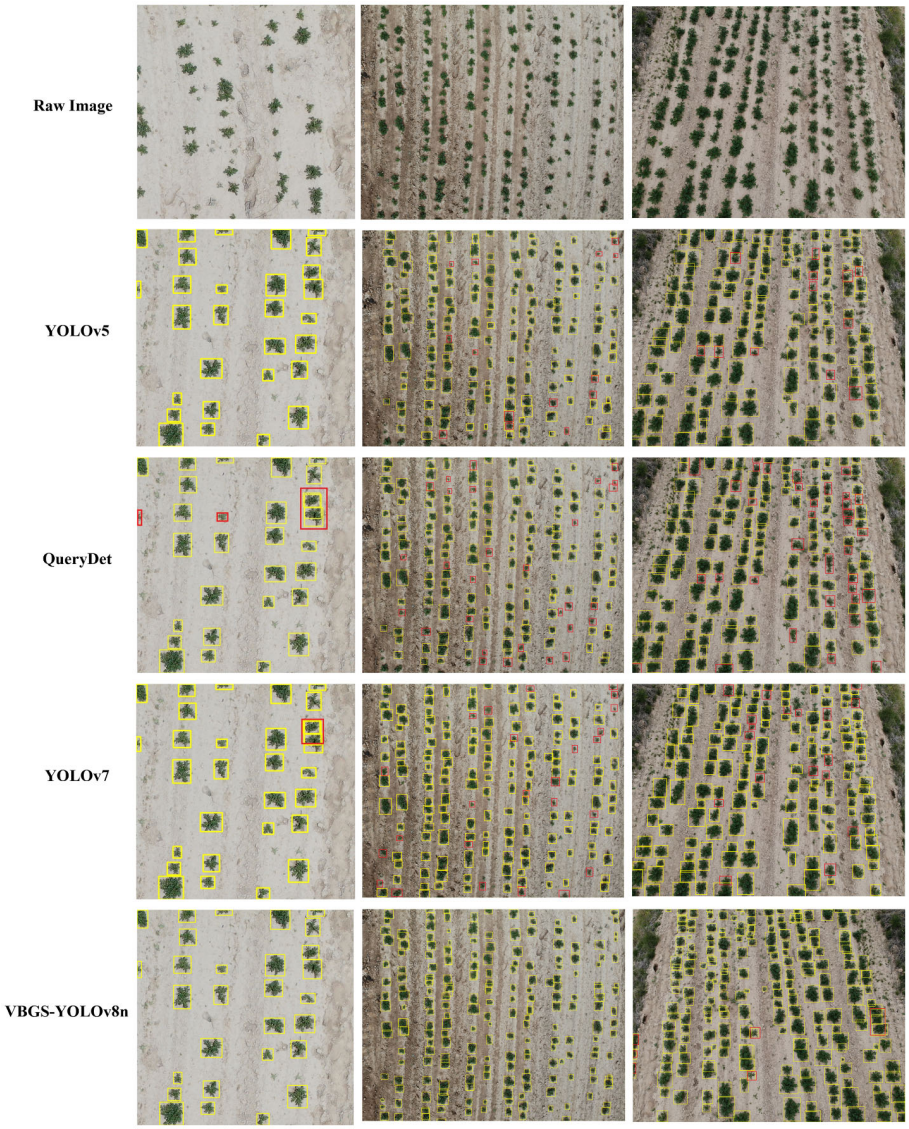


FIGURE 13
Detection results of potato seedlings in different environments by different models.

detection results for the other two environments, it is evident that the proposed VBGS-YOLOv8n model has the fewest missed detections and minimal redundant bounding boxes. This clearly demonstrates the excellent performance and accuracy of the VBGS-YOLOv8n model in recognizing potato seedlings.

4 Conclusion

This study introduces an enhanced VBGS-YOLOv8n network, aimed at addressing the challenge of detecting potato seedlings in drone remote sensing imagery. The model utilizes the lightweight VanillaNet algorithm as its backbone, effectively reducing the model's complexity. It incorporates a BiFPN to improve the retention of detailed features, thereby enhancing the accuracy of small target detection. GSconv convolution is employed in the neck to maintain overall accuracy, and the VoV-GSCSP network replaces all C2f modules in the original YOLOv8n algorithm's neck, significantly reducing the model's parameter count. Experimental validation demonstrates that VBGS-YOLOv8n exhibits exceptional performance in detecting small targets, with accuracy and mAP reaching 97.1% and 98.4%, respectively. Compared to the original YOLOv8 model, there is a 1.4% increase in accuracy and a 0.8% increase in mAP, alongside a 31.0% reduction in computation time. The parameter count is 48.3% of the original model, and the computational load is only 47.2%, with significant reductions in both missed and false detections. To verify its effectiveness, comparative analyses with leading models in the field affirm its superior detection accuracy, efficiency in parameter usage, and overall performance. The VBGS-YOLOv8n model achieves an optimal balance between detection speed, accuracy, and size, rendering it ideal for deployment on agricultural mobile devices. Future work will focus on optimizing the model for practical drone applications and broader datasets, ensuring the feasibility of VBGS-YOLOv8n and its detection capabilities for similar small target crops, offering technical support for precision agriculture.

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

LW: Methodology, Resources, Software, Writing – review & editing. GW: Conceptualization, Formal analysis, Funding acquisition, Methodology, Validation, Writing – original draft, Writing – review & editing. SY: Conceptualization, Funding acquisition, Software, Writing – review & editing. YL: Data curation, Resources, Writing – original draft, Writing – review & editing. XY: Writing – review & editing. BF: Funding acquisition, Visualization, Writing – review & editing. WS: Conceptualization, Formal analysis, Writing – review & editing. HL: Investigation, Validation, 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 Industrial Support Plan (Education Department of Gansu Province, 2023CYZC-42); the National Natural Science Foundation of China (NSFC, 32201663); the National Natural Science Foundation of Gansu (NSFG, 22JR5RA852) and the Gansu Agricultural University Talent Program (GAU-KYQD-2020-33).

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

References

Chen, H., Wang, Y., Guo, J., and Tao, D. (2023). VanillaNet: the power of minimalism in deep learning. *Adv. Neural. Inf. Process. Syst.* 36. doi: 10.48550/arXiv.2305.12972

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*. doi: 10.1007/s11554-024-01436-6

Li, S. J. (2023). Lightweight object detection algorithm for UAV images based on depth Xi. North University of China, Shanxi. Master's thesis.

Li, Y., Fan, Q., and Huang, H. (2023). A modified YOLOv8 detection network for UAV aerial image recognition. *Drones* 7, 304. doi: 10.3390/drones7050304

Li, S., Tao, T., Zhang, Y., Li, M., and Qu, H. (2023). YOLO v7-CS: A YOLO v7-Based Model for Lightweight Bayberry Target Detection Count. *Agronomy* 13, 2952. doi: 10.3390/agronomy13122952

Liang, D., Liu, W., and Zhao, Y. (2022). Optimal models for plant disease and pest detection using UAV image. *Nat. Environ. pollut. Technol.* 21, 1609–1617. doi: 10.46488/NEPT.2022.v21i04.013

Liu, J., Li, Y., Xiao, L. M., Li, W. Q., and Li, H. (2022). Orange fruit identification and localization method based on improved YOLOv4 model. *Trans. Chin. Soc. Agric. Eng.* 38, 173–182. doi: 10.11975/j.issn.1002-6819.2022.12.020

Liu, S. B., Yang, G. J., Zhou, C. Q., Jing, H. T., Feng, H., Xu, B., et al. (2018). Extraction of maize seedling number information based on UAV imagery. *Trans. Chin. Soc. Agric. Eng.* 34, 69–77. doi: 10.11975/j.issn.1002-6819.2018.22.009

Lu, D., Ye, J., Wang, Y., and Yu, Z. (2023). Plant detection and counting: enhancing precision agriculture in UAV and general scenes. *IEEE Access* 11, 116196–116205. doi: 10.1109/ACCESS.2023.3325747

Lun, R., Luo, Q., Gao, M., Li, G., and Wei, T. (2023). How to break the bottleneck of potato production sustainable growth-A survey from potato main producing areas in China. *Sustainability* 15, 12416. doi: 10.3390/su151612416

Osco, L. P., De Arruda, M. D. S., Junior, J. M., Da Silva, N. B., Ramos, A. P. M., Moryia, É.A.S., et al. (2020). A convolutional neural network approach for counting and geolocating citrus-trees in UAV multispectral imagery. *ISPRS J. Photogrammetry Remote Sens.* 160, 97–106. doi: 10.1016/j.isprsjprs.2019.12.010

Saifizi, M., Syauqi, M. A., Vinnoth, R., Mustafa, W. A., Idrus, S. Z. S., and Jamlos, M. A. (2019). "Estimation of paddy plant population using aerial image captured by drone," in *Proceedings of the 2nd Joint International Conference on Emerging Computing Technology and Sports (JICETS)*, Bandung, Indonesia, 25-27 November 2019. doi: 10.1088/1742-6596/1529/2/022085

Sapkota, R., Ahmed, D., and Karkee, M. (2023). Comparing YOLOv8 and Mask RCNN for object segmentation in complex orchard environments. *arXiv*. doi: 10.32388/ZB9SB0

Shi, M., Li, X. Y., Lu, H., and Cao, Z. G. (2022). Background-aware domain adaptation for plant counting. *Front. Plant Sci.* 13, 731816. doi: 10.3389/fpls.2022.731816

Shi, M. Y., and Xu, J. F. (2023). Innovation status and development suggestions of potato varieties in China. *Chin. Vegetables* 8), 1–5. doi: 10.19928/j.cnki.1000-6346.2023.1025

Sishodia, R. P., Ray, R. L., and Singh, S. K. (2020). Applications of remote sensing in precision agriculture: A review. *Remote Sens.* 12, 3136. doi: 10.3390/rs12193136

Tan, M., Pang, R., and Le, Q. V. (2020). "EfficientDet: scalable and efficient object detection," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 05 August 2020. doi: 10.1109/CVPR42600.2020

Wang, G., Chen, Y., An, P., Hong, H., Hu, J., and Huang, T. (2023). UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios. *Sensors* 23, 7190. doi: 10.3390/s23167190

Wang, Y., Tian, Y., Liu, J., and Xu, Y. (2023). Multi-stage multi-scale local feature fusion for infrared small target detection. *Remote Sens.* 15, 4506. doi: 10.3390/rs15184506

Wang, F., Wang, H., and Qin, Z. (2023). UAV target detection algorithm based on improved YOLOv8. *IEEE Access* PP, 1–1. doi: 10.1109/ACCESS.2023.3325677

Wu, B., Meng, J., Zhang, F., Du, X., Zhang, M., and Chen, X. (2010). "Applying remote sensing in precision farming-a case study in Yucheng," in *Proceedings of 2010 World Automation Congress*, Kobe, Japan, 19-23 September 2010.

Zhang, R. H., Ou, J. S., and Li, X. M. (2023). Lightweight pineapple seedling heart detection algorithm based on improved YOLOv4. *Trans. Chin. Soc. Agric. Eng.* 39, 135–143. doi: 10.11975/j.issn.1002-6819.202210133

Zhao, J., Zhang, X., Yan, J., Qiu, X., Yao, X., Tian, Y., et al. (2021). A wheat spike detection method in UAV images based on improved YOLOv5. *Remote Sens.* 13, 3095. doi: 10.3390/rs13163095

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 17 May 2024
DOI 10.3389/fpls.2024.1382416

OPEN ACCESS

EDITED BY
Mohsen Yoosefzadeh Najafabadi,
University of Guelph, Canada

REVIEWED BY
Milind B. Ratnaparkhe,
ICAR Indian Institute of Soybean Research, India
Parvathaneni Naga Srinivasu,
Prasad V. Potluri Siddhartha Institute of Technology, India

*CORRESPONDENCE
Joe I. R. Praveen
praveen.joe@vit.ac.in

RECEIVED 05 February 2024
ACCEPTED 02 April 2024
PUBLISHED 17 May 2024

CITATION
V. P, Kumar AMS, Praveen JIR, Venkatraman S, Kumar SP, Aravintakshan SA, Abeshek A and Kannan A (2024) Improved tomato leaf disease classification through adaptive ensemble models with exponential moving average fusion and enhanced weighted gradient optimization. *Front. Plant Sci.* 15:1382416. doi: 10.3389/fpls.2024.1382416

Improved tomato leaf disease classification through adaptive ensemble models with exponential moving average fusion and enhanced weighted gradient optimization

Pandiyaraju V.[1], A. M. Senthil Kumar[1], Joe I. R. Praveen[1]*, Shravan Venkatraman[1], S. Pavan Kumar[1], S. A. Aravintakshan[1], A. Abeshek[1] and A. Kannan[2]

[1]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India, [2]School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India

Tomato is one of the most popular and most important food crops consumed globally. The quality and quantity of yield by tomato plants are affected by the impact made by various kinds of diseases. Therefore, it is essential to identify these diseases early so that it is possible to reduce the occurrences and effect of the diseases on tomato plants to improve the overall crop yield and to support the farmers. In the past, many research works have been carried out by applying the machine learning techniques to segment and classify the tomato leaf images. However, the existing machine learning-based classifiers are not able to detect the new types of diseases more accurately. On the other hand, deep learning-based classifiers with the support of swarm intelligence-based optimization techniques are able to enhance the classification accuracy, leading to the more effective and accurate detection of leaf diseases. This research paper proposes a new method for the accurate classification of tomato leaf diseases by harnessing the power of an ensemble model in a sample dataset of tomato plants, containing images pertaining to nine different types of leaf diseases. This research introduces an ensemble model with an exponential moving average function with temporal constraints and an enhanced weighted gradient optimizer that is integrated into fine-tuned Visual Geometry Group-16 (VGG-16) and Neural Architecture Search Network (NASNet) mobile training methods for providing improved learning and classification accuracy. The dataset used for the research consists of 10,000 tomato leaf images categorized into nine classes for training and validating the model and an additional 1,000 images reserved for testing the model. The results have been analyzed thoroughly and benchmarked with existing performance metrics, thus proving that the proposed approach gives better performance in terms of accuracy, loss, precision, recall, receiver operating characteristic curve, and F1-score with values of 98.7%, 4%, 97.9%, 98.6%, 99.97%, and 98.7%, respectively.

KEYWORDS

deep learning, machine learning, image processing, ensemble learning, classification

1 Introduction

In the dynamic landscape of modern agriculture, where crop health plays a pivotal role in global food production, the precise and timely management of plant diseases is an ongoing challenge. Among these agricultural adversaries, leaf diseases emerge as intricate and multifaceted adversaries with distinct morphological manifestations. The science of leaf disease classification, a subdomain of plant pathology, is at the forefront of efforts to combat these detrimental afflictions. This research aspires to contribute to the field of leaf disease classification through the incorporation of pioneering technologies, namely, artificial intelligence (AI) and machine learning. The criticality of early detection and accurate classification in disease management cannot be overstated. Therefore, this study seeks to harness the potential of advanced algorithms, including convolutional neural networks (CNNs) and optimization into deep learning methodologies, to revolutionize the existing approaches to leaf disease diagnosis. At its core, this research addresses the challenges posed by leaf diseases by developing a novel classification system. By utilizing image recognition and deep learning techniques, this system aims to empower agriculture practitioners and plant pathologists with a sophisticated tool for disease identification. The impact of this system extends to many applications including crop health, reaching into the realms of global food security, sustainable agricultural practices, and environmental conservation.

Deep learning is an extension to the machine learning methods such as neural networks in AI that trains the computer system to recognize the patterns similar to the human brain. Deep learning models are trained to recognize even complex patterns found in images, text, videos, and voice data to perform accurate classifications and predictions. Deep learning algorithms perform both feature extraction and feature selection automatically without needing human effort as required in machine learning algorithms for training the software based on the algorithms. A CNN is one of the most important and fundamental deep learning neural network-based algorithms used for image recognition as it provides promising and accurate results in computer vision tasks. It has many architectural implementations including LeNet, AlexNet, Visual Geometry Group (VGG), GoogLeNet, and ResNet.

Time and space are important parameters to be considered for prediction-oriented decision-making systems. The temporal and spatial data on the disease growth in tomato leaves need time series analysis on image data with temporal reasoning. Moreover, prediction using time series analysis must focus on the direction of sequence that can be performed more effectively using machine learning-based classifiers. Moving average methods support to smoothen the time series analysis by identifying the temporal data patterns more effectively. Moreover, smoothing or filtering helps to eliminate the random variations that occur in the plotted time series data. An exponential (weighted) moving average method that applies a simple recursive procedure under the hood provides flexibility to the algorithm.

Despite the presence of many works on tomato plant leaf disease detection that are found in the literature, most of the existing systems use a machine learning approach for classification without any optimizer and temporal analysis. Therefore, it is necessary to employ manual preprocessing or to apply additional machine learning-based classification algorithms or clustering algorithms when performing effective feature extraction and feature selection. Moreover, the existing systems that use time series data are not designed to give higher importance to the most recent data and also do not focus on temporal reasoning by applying temporal constraints. Moreover, the convergence of the existing deep learning algorithm employed in the detection of tomato leaf diseases is not supported by an optimization algorithm. Finally, ensemble-based classification algorithms are not employed in the classification process to enhance the detection accuracy. Therefore, it is necessary to propose a new ensemble classifier with an optimization component and a temporal data analysis component.

In this paper, an ensemble model is proposed with an exponential moving average (EMA) function with temporal constraints based on interval analysis and an enhanced weighted gradient optimizer (EWGO) in which the gradient optimizer is enhanced with temporal rules and that is integrated into VGG-16 and Neural Architecture Search Network (NASNet) CNN architectures. VGG-16 is a fine-tuned model with a 16-layer depth developed by the VGG that consists of 13 convolution and max pooling layers with three fully connected layers, and it applies stride 2. The learning rate is fixed here as 0.1. The regression-based and binary classification-based loss functions are used in this work to reduce the errors. Moreover, the NASNet mobile training methods are integrated in this ensemble model for identifying the diseases in tomato leaves by providing improved learning and classification accuracy.

NASNet is also a CNN model that consists of two types of cells, namely, the normal and the reduction cells. The EMA method is used in this ensemble model since it gives more weightage to the current data in the temporally oriented time series data. Moreover, the Plant Village dataset is used in this work to carry out the experiments for testing the ensemble model proposed in this paper. Moreover, the Plant Village dataset is a publicly available dataset consisting of 54,305 images from which 1,000 images related to tomato leaves have been extracted and used in this work for training and testing the system. The main advantages of the proposed ensemble model are the increase in classification accuracy and the reduction in error rate in the detection of tomato leaf diseases.

The main motivation for this research work is that the profession of agriculture is one of the most vital in every world economy. It is the main source of resources in our country. Nowadays, leaf disease has a great impact on the productivity of vegetables. If we cannot control the disease, then it can greatly affect the harvest. These problems provide great motivation in finding out the origin of the disease at an earlier stage to help the tomato plants grow healthily and increase their yield. Another motivation for this research is that it addresses the challenges posed by leaf diseases by developing a novel classification system. By utilizing image recognition and deep learning techniques, this system aims to empower agriculture practitioners and plant pathologists with a sophisticated tool for disease identification. The impact of this

system extends beyond crop health, reaching into the realms of global food security, sustainable agricultural practices, and environmental conservation.

In this work, the Plant Village dataset is used to carry out the experiments for testing the model proposed in this paper. Moreover, the Plant Village dataset is a publicly available dataset consisting of 54,305 images from which 1,000 images related to tomato leaves have been extracted and used in this work for training and testing the system. The Plant Village dataset provides data to detect 39 different plant diseases. Moreover, the dataset contains 61,486 images of plant leaves with backgrounds. The dataset was designed using six different augmentation techniques in order to create more diverse datasets with different background conditions. The augmentations that have been used in this process include scaling, rotation, injection of noise, gamma correction, image flipping, and principal component analysis to perform color augmentation.

The main contributions of this paper are as follows:

- Proposal of an ensemble model using VGG-16 and NASNet mobile training deep learning models with an EMA function.
- Effective time series analysis using the CNN-based deep learning classifier along with an EWGO.
- Use of the Plant Village dataset for validation.
- Evaluation using suitable metrics.

The research unfolds in the following sequence: Section 2 provides a comprehensive exploration of the taxonomy and intricacies of leaf diseases. Section 3 is a detailed methodology section highlighting the technical aspects of image processing and machine learning, and the revelation of a state-of-the-art deep learning classification system designed to improve the accuracy and efficiency of leaf disease identification. In section 4, performance assessment of the proposed approach and results are compared with existing techniques. We conclude the research paper in section 5.

The VGG-16 architecture is a deep CNN designed for image classification tasks. It was introduced by the VGG at the University of Oxford. VGG-16 is characterized by its simplicity and uniform architecture, making it easy to understand and implement.

2 Literature survey

There are many works on tomato leaf detection, machine learning (Uma et al., 2016; Anusha and Geetha, 2022; Harakannanavara et al., 2022), deep learning (Haridasan et al., 2023; Sankareshwaran et al., 2023; Yakkundimath and Saunshi, 2023), optimization techniques, data mining (Das and Sengupta, 2020; Demilie, 2024), regression analysis, image analysis (Ganatra and Patel, 2020; Ngugi et al., 2021), and prediction techniques that are found in the literature. Mustafa et al. (2023) proposed a five-layer CNN model for detecting plant diseases using leaf images. A total of 20,000 images were used to train the model. This model detects the pepper bell plant leaf disease with better accuracy. The results are evaluated in terms of accuracy, precision, and recall, and F1-scores are computed. The model performs better than state-of-the-art models. Seetharaman et al. (Seetharaman and Mahendran, 2022) presented a region-based CNN model to detect a banana leaf disease using Gabor extraction. Images are preprocessed by histogram pixel localization with media filter. The segmentation part is done with region-based edge normalization. Feature extraction is performed using the novel method Gabor-based binary patterns with CNN. A region-based CNN helps in detecting the disease area. The results are evaluated and they perform better than CNN, DCNN, ICNN, and SVM models in terms of precision, recall, accuracy, and sensitivity.

Nerkar et al. (Nerkar and Talbar, 2021) proposed a method to detect leaf disease using a two-level nonintrusive method. This model combines generative adversarial network and reinforcement learning. Cross dataset learning is used. CNN is combined with GAN for better results. Re-enforcement learning retrains the GAN using confidence scores. Classification results are evaluated and results are higher than other models. Mukhopadhyay et al. (2021) proposed a non-dominated sorting genetic algorithm for tea leaf disease detection. Image clustering is the main idea of this model. PCA is used for feature reduction and multi-class SVM is used for disease detection. Five various datasets of tea leaf are used in the work. The proposed model provides better accuracy than traditional models.

Vallabhajosyula et al. (2022) proposed a transfer learning-based neural network for plant leaf disease detection. In this work, pre-trained models were used. The deep ensemble neural network is used along with pre-trained models. Transfer learning and data augmentation are used for parameter tuning. The results are evaluated and provide higher accuracy with lesser number of computations. Huang et al. (2023) discussed a tomato leaf disease detection model using the full convolutional neural network (FCN) with suitable normalization dual path networks. The FCN used to segment the target crop images and improve the dual path network model is used for feature extraction. The results are evaluated on the augmentation dataset and accuracy is better than other models.

Chouhan et al. (2021) proposed a model for leaf disease detection using the fuzzy-based function network. Initially, preprocessing is done and the scale-invariant feature transform method is used for feature extraction. The fuzzy-based function network is used for detecting the leaf disease. Training is done with the help of the firefly algorithm. The model results are evaluated in terms of accuracy and are higher than traditional models. He et al. (2023) presented a maize leaf disease detection model using machine vision. The batch normalization layer is appended with the convolution layer to fasten the convergence speed of the network. Cost function is developed to increase the detection accuracy. Four types of pre-trained CNN models are used for feature extraction network for training. The gradient descent algorithm is applied to optimize the model performance. The results are evaluated in terms of F1-score, recall rate, and accuracy.

Ruth et al. (2022) proposed a deep learning model for disease detection using the meta-heuristic algorithm. CNN is used for feature extraction. The optimal deep neural network is used for disease detection. A two-level weight optimization is used to

increase the performance of the detection model. Two-level weight optimization is achieved using an improved butterfly optimization algorithm, where the genetic algorithm is used to improve the butterfly optimization algorithm. The results are evaluated in terms of sensitivity, accuracy, and specificity. The overall accuracy is higher than other traditional models. Andrushia et al. (Andrushia and Patricia, 2020) presented a leaf disease detection model using the artificial bee colony optimization algorithm. Initially, preprocessing is done by removing noises and background images. Shape, color, and texture are extracted as features and are sent to the support vector machine model for disease detection. The model results are better in terms of recall, precision, and accuracy.

Abed et al. (2021) presented a novel deep learning model for bean leaf disease detection. This model contains two phases: detection and diagnosing. For detection, the U-Net architecture using the ResNet34 encoder is used. In the classification part, results are evaluated for five different deep learning models. The dataset contains 1,295 images of three classes such as healthy, bean rust, and angular leaf spot. The results are evaluated in terms of sensitivity, specificity, precision, F1-score, and area under the curve (AUC). Pandey et al. (Pandey and Jain, 2022) proposed a deep attention residual network using an opposition-based symbiotic organisms search algorithm. In this model, residual learning blocks are used with the attention learning mechanism for feature extraction. A new CNN model, AResNet-50, is designed for disease detection. The opposition-based symbiotic organisms search algorithm is used to tune the parameters of the model. Plants like citrus, guava, eggplant, and mango leaves are considered for the experimental analysis. The results of the model are evaluated in terms of accuracy, and they are better than those of the existing models such as AlexNet, ResNet-50, VGG-16, and VGG-19. Zhao et al. (2020) proposed a multi-context fusion network model for crop disease detection. In this model, standard CNN is used to extract visual features from 50,000 crop disease samples. Contextual features are collected from image acquisition sensors. A deep, fully connected network is proposed by combining contextual features and visual features to detect the leaf disease. The model performance is evaluated in terms of accuracy, which is higher than state-of-the-art methods.

Wang et al. (2017) proposed a new technique for automatic estimation of plant disease severity using image analysis through the effective application of deep learning algorithms. Bracino et al. (2020) explained the development of a new hybrid model based on machine learning techniques for the accurate detection of health using disease classification. Ashwinkumar et al. (2022) proposed an automated plant leaf disease detection model using deep learning classification named optimal MobileNet, which is designed based on CNNs. Khan et al. (2019) developed one optimized method for disease detection using image segmentation and classification for identifying the apple diseases. The authors made the decisions by analyzing whether there is a strong correlation among the features and also using genetic algorithm for feature selection. Most of the works found in the literature on tomato leaf disease detection used the benchmark dataset, namely, the Plant Village dataset (Kaustubh, 2020).

Sanida et al. (2023) proposed a new methodology for the effective detection of tomato leaf diseases by identifying them using a two-stage transfer learning model. Pandiyaraju et al. (2023) proposed an optimal energy utilization technique for reducing the energy consumption via the agricultural sensors used in precision agriculture. These sensors have been connected to a WSN that performs energy optimization by using a multi-objective clustering and deep learning algorithm to reduce the energy consumption. In another related work, Pandiyaraju et al. (2020) developed an energy-efficient routing algorithm for WSNs using clustering of nodes. Moreover, the routing decision has been made in their work using intelligent fuzzy rules that were applied in precision agriculture. In the area of agriculture and gardening, Pandiaraju et al. (Pandiyaraju et al., 2017) proposed a rule-based intelligent roof control algorithm for effective water conservation without affecting the agricultural yield with respect to smart terrace gardening. Such a model can be enhanced to detect the leaf diseases for providing better yield with minimum water.

Shoaib et al. (2023) presented a review of deep learning classification algorithms that have been used in the detection of plant leaf diseases. Santhosh et al. (2014) proposed a farmer advisory system using intelligent rules based on machine learning classifier. Jabez Christopher et al. (Jabez et al., 2015) proposed an optimized classification model that uses rules based on knowledge mining with swarm optimization for providing effective disease diagnosis. Gadade et al. (Gadade and Kirange, 2022) proposed an intelligent approach based on deep learning for the effective detection of tomato leaf diseases from leaf images that have captured with varying capturing conditions. Saeed et al. (2023) proposed one new smart detection methodology for the accurate detection of tomato leaf diseases by using transfer learning-based CNNs. Shoaib Muhammad et al. (Shoaib et al., 2022) proposed a new model for tomato leaf disease detection by using deep learning algorithms for performing both segmentation and classification of leaf images.

Sreedevi and Manike (2024) presented a new solution for identifying the tomato leaf disease based on classification using a modified recurrent neural network through severity computation. Prabhjot Kaur et al. (2024) carried out a performance analysis on the image segmentation models that are used to detect leaf diseases present in the tomato plants. Thai-Nghe et al. (Nguyen et al., 2023) presented a deep learning-based approach for the effective detection of tomato leaf diseases. Chang et al. (2024) developed one general-purpose edge-feature-guided model for the identification of plant diseases by enhancing the power of vision transformers. Li et al. (2023) presented a new lightweight vision transformer model based on shuffle CNNs for the effective diagnosis of leaf diseases in sugarcane plants. Thai et al. (2023) proposed a new vision transformer model designed for the accurate detection of cassava leaf diseases.

Yu et al. (2023) explained the use of inception convolutional vision transformers for the effective identification of plant diseases. Arshad et al. (2023) developed an end-to-end and hybrid model based on the deep learning framework for the accurate prediction of potato leaf diseases. Shiloah et al. (Elizabeth et al., 2012) proposed one new segmentation approach based on machine learning model for improving the diagnostic accuracy of detecting lung cancers from chest computed tomography images. Dhalia Sweetlin et al. (2016) proposed a patient-specific model for the effective

segmentation of lung computed tomographic images. Singh and Misra (2017) proposed a machine learning-based model for the effective detection of plant leaf diseases by performing suitable image segmentation. Agarwal et al. (2020) developed a new system for tomato leaf disease detection by applying the CNN classifier.

Chen et al. (2022) proposed the use of the AlexNet CNN model for the effective detection of tomato leaf diseases by performing accurate classification of tomato leaf images. Ganapathy et al. (2014) proposed an intelligent temporal pattern classification model by using fuzzy temporal rules with particle swarm optimization algorithm. Jaison et al. (Bennet et al., 2014) proposed a discrete wavelet transform-based feature extraction model along with one hybrid machine learning classification algorithm for performing effective microarray data analysis. Elgin Christo et al. (2019) proposed a new correlation-based ensemble feature selection algorithm that has been developed using bioinspired optimization algorithms integrated with a backpropagation neural network-based classifier.

Thangaraj et al. (2021) proposed an automated tomato leaf disease classification algorithm by using a transfer learning-based deep CNN classifier. Al-Gaashani et al. (Al-gaashani et al., 2022) proposed a new model for tomato leaf disease classification by the application of transfer learning with feature concatenation. Han et al. (2017) proposed a new weighted gradient-enhanced classification model not only to provide high-dimensional surrogate modeling but also to perform design optimization. Wu et al. (2021) proposed a new distributed optimization method that uses weighted gradients for solving the economic dispatch problem pertaining to the multi-microgrid systems. Abouelmagd et al. (2024) developed an optimized capsule neural network for the effective classification of tomato leaf diseases. Other approaches that are used in the detection of leaf diseases include those with deep learning and also with explainable AI (Rakesh and Indiramma, 2022; Bhandari et al., 2023; Debnath et al., 2023; Nahiduzzaman et al., 2023).

Despite the presence of all these related work in the literature, most of the segmentation and classification algorithms use a machine learning approach for classification. Therefore, it is necessary to employ either manual work or additional classification algorithms for performing feature extraction and feature selection. Moreover, the time series data are not analyzed by giving higher importance to the most recent data by the application of temporal constraints. The convergence of the existing deep learning algorithm employed in the detection of tomato leaf diseases is not supported by an optimization algorithm. Finally, ensemble-based classification algorithms are not employed in the classification process to enhance the detection accuracy. In order to handle all these limitations that are present in the existing systems developed for accurate tomato leaf disease detection, a new ensemble classification model is proposed in this paper that uses an EMA function with temporal constraints, and it is supported by an EWGO along with fine-tuned VGG-16 and NASNet mobile training methods for enhancing the classification accuracy that can increase the detection accuracy with respect to the detection of tomato leaf diseases.

3 Proposed work

3.1 Method

The data that show the features are initially analyzed using histogram plots and pie charts for better visualization of the data statistics to check for data imbalance among different classes. It has been concluded via complete exploration that there is no data imbalance and that the features of the images have been completely studied.

Next, the images are preprocessed in order to enhance the learning ability of our deep learning models. A median filter is applied on the image to remove noise to improve image quality. Redundant parts of the image that do not contribute to the model's learning process are also removed. Furthermore, the α and β factors in our images are adjusted in order to modify the brightness and contrast, thereby making the region of interest more prominent. The images are finally normalized to have pixel values ranging from 0 to 1, and the data are augmented to ensure a wider scale of learning by the model.

For the initial part of feature extraction, the VGG-16 transfer learning model undergoes fine-tuning by unfreezing its last five layers, enabling to adapt the model that originally contained ImageNet's weights to the specified dataset. By employing the use of Global Average Pooling to pool the CNN layers' features, the data are then passed into two fully connected layers ultimately leading to the output layer. The optimization of the model is achieved using the Adam optimizer with a learning rate of 0.0001, and evaluation metrics such as the F1-score, AUC score, precision, and recall are applied.

The NASNet mobile transfer learning model is employed with ImageNet weights for the next part. A flattened layer is then used to transform the outputs from the CNN layers into a one-dimensional tensor that facilitates the passage through three fully connected layers that ultimately reach the output layer. The optimization of the model is once again achieved using the Adam optimizer with a learning rate of 0.0001, and evaluation metrics such as the F1-score, AUC score, precision, and recall are applied.

The extracted features obtained from the two transfer learning models are now taken and passed on as parameters to a custom ensemble layer that incorporates EMA function that emphasizes the recent data points with greater weights. The resulting ensemble model shows an optimized learning curve by adopting the adaptive rate of learning, which is achieved by using a custom EWGO that modifies the learning rate based on custom ensemble weight suitable for our custom ensemble model.

3.2 Dataset

This research utilizes the dataset (Kaustubh, 2020) that consists of a collection of tomato leaf images, each belonging to one of nine distinct categories, representing various leaf diseases or a healthy state (no disease). The dataset encompasses a total of 10,000 images designated for training and an additional 1,000 images reserved for

testing. To facilitate model development and evaluation, we partitioned the training dataset into a 75%–25% split, resulting in 7,500 images allocated for training and 2,500 images for validation, and the entire additional 1,000 images were reserved for the test set.

This dataset serves as the foundation for the development of the proposed model, which aims to enhance the classification of tomato leaf diseases.

3.3 Preprocessing

The following are the steps involved in preprocessing:

- Median filter
- Image cropping
- Brightness and contrast adjustments
- Normalization

3.3.1 Median filter

The first step of data preprocessing utilizes a median filter, which is a non-linear digital image filtering technique that runs through the signal as one entry after another by replacing the entry value by the median of the neighboring entry values, which depends on the window size, resulting in the removal of the salt-and-pepper noise in an image. In this case, a window size of 3 has been chosen for preprocessing the image.

This median filter is represented mathematically as shown in Equation (1):

$$g(x, y) = Med(f(x, y)) \tag{1}$$

where $f(x,y)$ is the window array and $g(x,y)$ is the median value of the window array. The steps for the median filter are shown in Algorithm 1.

```
function median_filter():
    input: raw tomato_leaf_image;
    output: median_filtered_image;
    image = input;
    l = length of image;
    b = breadth of image;
    c = channels of image;
    w = window_size;
    filtered_image = create_empt;y_image(l,b)
    b_image = img[l][b][1];
    g_image = img[l][b][2];
    r_image = img[l][b][3];
    for i = 0 to l − 1 do:
        for j = 0 to b − 1 do:
            b_img = image[i][j][1]
            g_img = image[i][j][2];
            r_img = image[i][j][3];
        end for
    end for
    apply_median_filter(b_img,w);
    apply_median_filter(g_img,w);
    apply_median_filter(r_img,w);
    for i = 0 to l − 1 do:
        for j = 0 to b − 1 do:
            filtered_image = [b_img[i][j], g_img[i][j], r_img[i][j]];
        end for
    end for
end function
End
Function apply_median_filter():
    input: single_channel_tomato_leaf_image;
    output: median_filtered_single_channel_image;
    len = length of img;
    bt = length of img;
    applied_img = create_array(len,bt);
    wh = w/2 ;
    for x = 0 to len − 1 do:
        for b = 0 to bt − 1 do:
            window = [];
            for i = −wh to wh − 1 do:
                for j = −wh to wh − 1 do:
                    winx = x + i;
                    winy = y + j;
                    if winx >= 0 and winy >= 0 and winx < len and winy < bt then:
                        append value to window (img[winx][winy])
                    end if
                end for
            end for
        end for
    end function
End
```

Algorithm 1. Median filter.

3.3.2 Image cropping

Since the outer areas of the image are not helpful with the tomato disease detection, the size of the image is reduced by 10 pixels on each side, thus reducing the image size from 256 × 256 to 236 × 236 by removing the areas where there are no significant features for disease detection. The steps for image cropping are shown in Algorithm 2.

```
Function crop_image ():
    input: median_filtered_image
    output: cropped_median_filtered_image
    img = median filtered image
    length = length of img
    breadth = breadth of img
    crop_value = 10
    max_crop_length = length − crop_value
    max_crop_breadth = breadth − crop_value
    crop_image = create empty image of dimensions(max_crop_length,max_crop_breadth)
```

```
        crop_image = img[crop_value:max_crop_length][crop_
        value:max_crop_breadth]
    end Function
end
```

Algorithm 2. Image cropping.

3.3.3 Brightness and contrast enhancements of images

For better-quality images and improved ability of the CNN to identify the region of interest, its brightness is reduced and the contrast of the image is increased. This mitigates overexposure of the images, allowing the CNN to extract the features in the region of interest easily due to better visibility.

Brightness and contrast enhancement can be represented mathematically as shown in Equation (2):

$$g(i,j) = \alpha f(i,j) + \beta \tag{2}$$

where α is the contrast factor and β is the brightness factor. $f(i,j)$ represents the pixel of the input image, which is the cropped image, while $g(i,j)$ is the output image where the image's brightness and contrast are adjusted using α and β. The procedure for brightness and contrast enhancements is shown in Algorithm 3.

```
Function adjust_image():
    input:cropped_median_filtered_image,brightness_
    factor,contrast_factor
    output:cropped_filtered_image_with_adjustments
    image = cropped_median_filtered_image
    l←length of image
    b←breadth of image
    c ← channels of image
    adjusted_img←create empty image of dimensions
    l and b
    α←contrast_factor
    β←brightness_factor
    for i = 0 to l − 1 do:
        for j = 0 to b − 1 do:
            for k = 0 to c − 1 do:
                adjusted_img[i][j][k] ← α*image[i][j][k] + β
            end for
        end for
    end for
end Function
end
```

Algorithm 3. Brightness and contrast enhancement.

3.3.4 Image normalization

For better weight initialization and to maintain consistency in the pixel range of the input, the image is normalized so that all pixel values are confined to the interval [0, 1]. Due to this, the deep learning model's convergence is enhanced with the range reduction from 255 to 1 by dividing each pixel value by 255. This process also improves the learning rate of our proposed model and the stability of the model during training. The procedure for image normalization is shown in Algorithm 4.

```
Function normalize_image():
    input:brightness_and_contrast_adjusted_image
    output:normalised_image
    image = input
    l←length of image
    b←breadth of image
    c ← channels of image
    normalisation_value←255
    normalised_image←create_empty_image(l,b)
    for i = 0 to l − 1 do:
        for j = 0 to b − 1 do:
            for k = 0 to c − 1 do:
                normalised_image[i][j][k]←image[i][j][k]/255
            end for
        end for
    end for
end Function
end
```

Algorithm 4. Image normalization.

3.4 Feature extraction and classification

Upon successful completion of preprocessing, the tomato leaf images are subjected to appropriate feature extraction and thereby will be classified using the deep learning model. This, in turn, will support not only the identification of diseases in the leaves but also the severity. The deep learning model used is the VGG-16 fine-tuned model. In addition, a CNN model, namely, NASNet, is also employed for the leaf's disease identification.

Later, an ensemble model consisting of five ensemble blocks and a final output block is used with the input layer being received from the output of the VGG-16 fine-tuned model and the NASNet model as a list. Furthermore, the results are improved for an enhanced performance with the aid of an EMA-based approach and optimized with an EWGO.

3.4.1 VGG-16 fine-tuned model

The last five layers of the VGG-16 model are unfrozen and the weighs of these layers are updated with the data to fine-tune the model. The optimizers do not modify the parameters of the remaining layers, which remain frozen, thereby preserving the weights.

This model, which is made up of five different blocks, is composed of convolution layers with rectified linear unit (ReLU) activation and a max pooling layer, a global average pooling layer, dense layers, batch normalization layers, and an output dense layer with softmax activation. The preprocessed image of size 236 × 236 × 3 is taken as an input into the model, first entering block 1.

Block 1 consists of two convolution layers and a max pooling layer. Each convolution layer consists of 64 filters, each of size 3 × 3. Each layer also has a ReLU activation layer that brings in non-linearity once the feature extraction is done by that layer. The first convolution layer receives the input as 236 × 236 × 3, and the first convolution layer produces the output of shape 236 × 236 × 64 after the activation function. The second convolution layer takes the input as the output of the first convolution layer and performs feature extraction and ReLU activation without making any changes in the shape of the data. Once the output data are produced by the second convolution layer, the max pooling layer that has a filter size of 2 × 2 reduces the size from 236 × 236 × 64 to 118 × 118 × 64, which sends the output to block 2.

Block 2, just like block 1, consists of two convolution layers where each layer has a ReLU activation function and a max pooling layer. The only difference is that the input received by the first convolution layer of this block will be of size 118 × 118 × 64. At the end of the second convolution, the output will be of size 118 × 118 × 128 since the number of filters in the convolution layers of the second block is 128. The max pooling layer reduces the size of the data from 118 × 118 × 128 to 59 × 59 × 128.

Block 3, unlike the previous two blocks, has three convolution layers where each layer has a ReLU activation function and a max pooling layer. The functionality of the block remains the same with the difference here being the presence of a third convolutional layer and the presence of 256 filters in each convolution layer. The first convolution layer receives the input of size 59 × 59 × 128 from the max pooling layer of block 2 and produces an output of size 59 × 59 × 256, which is preserved in the second and third convolution layer. The max pooling layer reduces the size of the data to 29 × 29 × 256.

Blocks 4 and 5 are similar to block 3 with the only difference being all the convolution layers present in blocks 4 and 5 have 512 filters. The input received by the first layer of block 4 will be of dimension 29 × 29 × 256 and the output after the third convolution layer will be of size 29 × 29 × 512, which, in turn, is reduced to 14 × 14 × 512 by the max pooling layer. In case of block 5, the input received by the first convolution layer will be of size 14 × 14 × 512 and the output is preserved even after the third convolution layer. The max pooling layer in block 5 reduces its size from 14 × 14 × 512 to 7 × 7 × 512.

The global average pooling layer takes the output of block 5 as input, which down-samples the multi-dimensional data into single-dimensional data by finding the average of each feature map where the filter is of size 2 × 2, resulting in the reduction of data size from 7 × 7 × 512 to 1 × 1 × 512. After this down-sampling, two dense layers with ReLU activation composed of 128 and 32 neurons, respectively, transform the output obtained by extracting the features of the preceding layers into data, which are suitable for classification. Finally, the output layer, i.e., dense layer with softmax activation, is used to perform multiclass classification. The steps for VGG-16 fine-tuned model is shown in Algorithm 5.

input: *preprocessed tomato leaf image*
output: *trained finetuned_VGG16 classifier for tomato leaf disease classification*
Function *TrainClassifier(preprocessed_tomato_leaf_image):*
  model ← *VGG16 multiclass Classifier*
    k ← *finetuneable layers*
  for *layer in last k model layers* ***do***
    layer ← *trainable*
  end for
  β ← *batch size*
  N ← *total classes of tomato leaf diseases*
  h ← *height of preprocessed_tomato_leaf_image*
  w ← *width of preprocessed_tomato_leaf_image*
  c ← *color channels of preprocessed_tomato_leaf_image*
  for *epoch* = 1 *to* 100 ***do***
    μ ← *learning rate*
    while *performance does not plateau* ***do***
    batch ← *obtain a batch of size* ***β***
    feed ***batch*** *into model through layers* ***L***
    prob ← *predicted tomato leaf disease class probabilities*
    labels ← *ground truth probabilities*
    loss, ***δ*** ← *categorical cross entropy loss*

$$\boldsymbol{\delta} \leftarrow -\log\left(\frac{e^{x_p}}{\sum_{j=1}^{N} e^{x_j}}\right)$$

$$x_i \leftarrow \textit{logit for class } i \in \{1, 2, \ldots, N\}$$

    update model parameters ***θ*** *through backpropagation using loss* ***δ***

$$\boldsymbol{\theta} \leftarrow \theta - \mu \nabla \delta$$

    where $\nabla\boldsymbol{\delta}$ ← *gradient of loss* ***δ*** *with respect to model parameters* ***θ***
    compute ***accuracy***,

$$accuracy = \frac{\sum_{k=1}^{N}(TP_k + TN_k)}{\sum_{k=1}^{N}(TP_k + TN_k + FP_k + FN_k)}$$

    compute ***precision***,

$$precision = \frac{\sum_{k=1}^{n} TP_k}{\sum_{k=1}^{n}(TP_k + FP_k)}$$

    compute ***recall***,

$$recall = \frac{\sum_{k=1}^{n} TP_k}{\sum_{k=1}^{n}(TP_k + FN_k)}$$

    compute $\boldsymbol{F1}$ – ***score***,

$$F1 - Score = \frac{2\sum_{k=1}^{n} TP_k}{\sum_{k=1}^{n}(2TP_k + TN_k + FP_k)}$$

    use ***Adam*** *optimizer to monitor loss* ***δ*** *and tune model learning*;
    end while
    if *performance plateaus* ***then***

update learning rate μ to promote further learning
end if
end for
$output_{VGG16} \leftarrow$ output probabilities from model
return $output_{VGG16}$
end Function
end

Algorithm 5. Tomato leaf classification—fine-tuned VGG-16 training.

input: preprocessed tomato leaf image
output: trained NASNet classifier for tomato leaf disease classification
Function TrainClassifier(preprocessed_tomato_leaf_image):
model $\leftarrow$ NASNet multiclass Classifier
$\beta \leftarrow$ batch size
$N \leftarrow$ total classes of tomato leaf diseases
$h \leftarrow$ height of preprocessed_tomato_leaf_image
$w \leftarrow$ width of preprocessed_tomato_leaf_image
$c \leftarrow$ color channels of preprocessed_tomato_leaf_image
for epoch = 1 to 100 **do**
$\mu \leftarrow$ learning rate
while performance does not plateau **do**
batch $\leftarrow$ obtain a batch of size β
feed **batch** into **model** through layers L
prob $\leftarrow$ predicted tomato leaf disease class probabilities
labels $\leftarrow$ ground truth probabilities
loss, $\delta \leftarrow$ categorical cross entropy loss

$$\delta \leftarrow -\log\left(\frac{e^{x_p}}{\sum_{j=1}^{N} e^{x_j}}\right)$$

$x_i \leftarrow$ logit for class $i \in \{1, 2, \ldots, N\}$
update model parameters θ through back propagation using loss δ

$$\theta \leftarrow \theta - \mu\nabla\delta$$

where $\nabla\delta \leftarrow$ gradient of loss δ with respect to model parameters θ
compute **accuracy**,

$$accuracy = \frac{\sum_{k=1}^{N}(TP_k + TN_k)}{\sum_{k=1}^{N}(TP_k + TN_k + FP_k + FN_k)}$$

compute **precision**,

$$precision = \frac{\sum_{k=1}^{n} TP_k}{\sum_{k=1}^{n}(TP_k + FP_k)}$$

compute **recall**,

$$recall = \frac{\sum_{k=1}^{n} TP_k}{\sum_{k=1}^{n}(TP_k + FN_k)}$$

compute $F1 - score$,

$$F1 - Score = \frac{2\sum_{k=1}^{n} TP_k}{\sum_{k=1}^{n}(2TP_k + TN_k + FP_k)}$$

use **Adam** optimizer to monitor loss δ and tune model learning;
end while
if performance plateaus **then**
update learning rate μ to promote further learning
end if
end for
$output_{NASNet} \leftarrow$ output probabilities from **model**
return $output_{NASNet}$
end Function
end

Algorithm 6. Tomato leaf classification—NASNet training.

3.4.2 NASNet

NASNet is a deep learning architecture where an optimal neural architecture is searched automatically by using the Neural Architecture Search (NAS) method. For the best performance on a specific task, the design of the neural network's topology is automated using the NAS process.

The NAS algorithm can be generalized as an algorithm that searches for the best algorithm to perform a certain task. It involves three different components, namely, the search space, performance estimation strategy, and search strategy. The search space encompasses all the potential architectures that can be looked for within the neural network's subspace. It can be categorized into two primary types: the global search space and the cell-based search space. The global search space offers a high degree of flexibility, accommodating a wide range of architecture due to its ample operation arrangement options. In contrast, the cell-based search space is characterized by recurring fixed structures in effective, manually designed architectures, leading to the assembly of smaller cells into larger architectural structures.

Without construction or training of a possible neural network, the performance is evaluated using the performance estimation strategy, which returns a number or an accuracy value of the possible model architecture, which the NASNet predicts as a possible solution. Different search strategies such as grid search, random search, gradient-based search, evolutionary algorithm, and reinforcement learning can be used to identify the best architectures and avoid bad ones before estimating performance. The steps for NASNet training is shown in Algorithm 6.

3.4.3 Ensemble model

The ensemble consists of five ensemble blocks and a final output block. The input layer receives the output of the VGG-16 fine-tuned model and the NASNet model as a list. This input is then passed through the five ensemble blocks, finally reaching the output layer. Each ensemble block is composed of a fully connected layer, a reshape layer, two convolutional layers, a batch normalization layer, ReLU activation, an ensemble layer, and a max pooling layer.

The ensemble process in the ensemble layer is carried out based on effective moving average. This layer has two parameters, namely,

the decay rate, which is responsible for reducing the effective moving average, and the update rate, which ensures that for every update rate iteration, the weights in the ensemble layer will be modified with the help of the effective moving average.

The output layer is responsible for classification.

The effective moving average is represented mathematically as shown in Equation (3):

$$EMA_{updated} = EMA + (Pred_{NASnet} - Pred_{VGG16FT}) * \delta \quad (3)$$

where $EMA_{updated}$ is the updated effective moving average; EMA denotes the effective moving average before the update operation; $Pred_{NASNet}$ and $Pred_{VGG16FT}$ are the predictions of NASNet and the VGG-16 fine-tuned model, respectively; and δ is the decay rate, which is taken as 0.8 in this case.

The predictions of both models are taken as input. Initially, the prediction of the VGG-16 fine-tuned model was taken as the effective moving average, which is then updated with the help of the above mathematical expression. The update rate ensures that the weights are modified only after a certain number of iterations, which is two in this case. Therefore, for every second iteration, the weights are modified by reshaping the effective moving average tensor for every weight tensor. The reshaped tensor is updated into the weight tensor as the new weight tensor for the next two iterations. The procedure for Ensemble classifier training using EMA is shown in Algorithm 7 and procedure for exponential moving average-based ensemble weight update in a custom ensemble layer is shown in Algorithm 8.

input: *preprocessed tomato leaf image*
output: *trained ensemble with EMA classifier for tomato leaf disease classification*
Function *TrainClassifier(preprocessed_tomato_leaf_image):*
VGG ← *train VGG16 classifier;*
NASNet ← *train NASNet classifier;*
β ← *batch size*
N ← *total classes of tomato leaf diseases*
h ← *height of preprocessed_tomato_leaf_image*
w ← *width of preprocessed_tomato_leaf_image*
c ← *color channels of preprocessed_tomato_leaf_image*
for *epoch* = 1 *to* 100 ***do***
μ ← *learning rate*
while *performance does not plateau* ***do***
batch ← *obtain a batch of size* β
feed ***batch*** *into model through ensemble layer*
$output_{ensemble}$ ← *EMA_ensemble(VGG, NASNet)*
feed ***batch*** *into* ***model*** *through fully connected and reshape layers*
$output_{reshape} \cdot shape \leftarrow (h, w, c)$
perform convolution on $output_{reshape}$

$$O(x,y) \leftarrow \sum_{i}^{m_1}\sum_{j}^{m_2}\sum_{k}^{m_c} I(x-i, y-j, k) * K(i,j,k)$$

flatten convolution output
$output_{flatten} \cdot shape \leftarrow (\beta,\ h*w*d)$
feed $output_{flatten}$ *to output layer*
prob ← *predicted tomato leaf disease class probabilities*
labels ← *ground truth probabilities*
loss, δ ← *categorical cross entropy loss*

$$\delta \leftarrow -\log\left(\frac{e^{x_p}}{\sum_{j=1}^{N} e^{x_j}}\right)$$

x_i ← *logit for class* $i \in \{1, 2, \ldots, N\}$
update model weights θ *using Effective Moving Average,* ***ema***
compute ***accuracy***,

$$accuracy = \frac{\sum_{k=1}^{N}(TP_k + TN_k)}{\sum_{k=1}^{N}(TP_k + TN_k + FP_k + FN_k)}$$

compute ***precision***,

$$precision = \frac{\sum_{k=1}^{n} TP_k}{\sum_{k=1}^{n}(TP_k + FP_k)}$$

compute ***recall***,

$$recall = \frac{\sum_{k=1}^{n} TP_k}{\sum_{k=1}^{n}(TP_k + FN_k)}$$

compute $F1 - $***score***,

$$F1 - Score = \frac{2\sum_{k=1}^{n} TP_k}{\sum_{k=1}^{n}(2TP_k + TN_k + FP_k)}$$

use ***EWG*** *optimizer to monitor loss* δ *and tune model learning;*
end ***while***
if *performance plateaus* ***then***
update learning rate μ *to promote further learning*
end ***if***
end ***for***
$output_{NASNet}$ ← *output probabilities from model*
return $output_{NASNet}$
end ***Function***
end

Algorithm 7. Ensemble classifier training using EMA for tomato leaf disease classification.

Initialize model ← ***EnsembleClassifier***(*tensor*);
Set ***decay rate***, α ← 0.8;
Set ***update rate***, β ← 2;
Set ***counter*** ← 0;
$NasNet_{Outputs}$ ← *TrainNasNet(tensor)*;
$VGG16_{Outputs}$ ← *TrainVGG16(tensor)*;
Function *Custom_EMA_Ensemble():*
Initialize ema_0 ← $VGG16_{Outputs}$
while *EnsembleModel is running do*
ema_i ← $(1 - \alpha) * ema_i - 1 + \alpha * NasNet_{Outputs}$
counter ← ***counter*** + 1

```
        if counter % β ← 0 then
        weights
        ←[reshape(ema_i, weight.shape)for weight
            in current model weights]
        Update Ensemble Layer weights
      end if
    end while
  end Function
end
```

Algorithm 8. Exponential moving average-based ensemble weight update in a custom ensemble layer.

3.4.4 Layer information during feature extraction

There are a total of 12 layers used during feature extraction as enumerated below.

3.4.4.1 *(i)* Convolutional layer

The convolutional layer is the most important layer used in CNNs, which is responsible for extracting features from the input with the use of filters or kernels. The kernel is a matrix consisting of a set of learnable parameters. The convolution process can be defined as the conversion of pixels in its receptive field into a single pixel. This operation is performed as the dot product between the kernel matrix and another matrix, which is the receptive field restricted to a certain portion. Hence, in the input image that is composed of three color channels, the kernel carries out the convolution operation in all the three channels, although the height and width will be spatially small. The kernel slides across the height and width of the receptive region of the image. This sliding size is called a stride. The result is a production of a two-dimensional representation of the kernel at each spatial position of the image. The convolution operation results in a feature map as output, which can be represented mathematically as shown in Equation (4):

$$O(x,y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(x-i, y-j) * K(i,j) \quad (4)$$

where $O(x,y)$ represents the value in the output feature map in the position (x, y) and $I(x–I,y–j)$ represents the pixel value in the input at position $(x–i,y–i)$. $K(i,j)$ represents the value of the kernel at position (i,j).

3.4.4.2 *(ii) Depthwise separable convolutional layer*

Depthwise separable convolution handles both the spatial and depth dimensions. Here, the kernels cannot be factored into smaller units. This process is split into two steps:

- Depthwise convolution: a single convolution filter is applied on each input channel.
- Pointwise convolution: it involves the usage of a 1 × 1 filter that iterates through every single point of the input.

This kernel has a depth equal to the number of channels that the input has. The usage of a depthwise separable convolution layer reduces the number of parameters compared to the standard convolution layer.

3.4.4.3 *(iii)* Max pooling layer

The max pooling layer is one of the largely used layers in CNNs, normally found after the convolutional layer. Its purpose is to reduce the spatial dimensions (length and breadth in this case) of the input feature map resulting from the preceding convolution layer. The feature map is taken by the layer as input, which applies the max pooling operation where a window slides through the feature map the window content with the maximum value in the window, thus down-sampling the feature map. Providing a stride value lets the CNN know the number of pixels to move while sliding through that particular layer. The max pooling layer can be mathematically represented as shown in Equation (5):

$$O(x,y) = max_{i=0}^{k-1} max_{j=0}^{k-1} I(x \cdot s + i,\ y \cdot s + j) \quad (5)$$

where $O(x,y)$ is the value in the output feature map at point (x, y), s is the stride value, and $I(x·s+i,y·s+j)$ is the value in the input feature map at position $(x·s+i,y·s+j)$, and k is the size of the pooling window.

3.4.4.4 *(iv)* Average pooling layer

The purpose of using the average pooling layer is to reduce the spatial dimensions such as the length and depth of the feature map just like the max pooling function, but the difference here is that down-sampling is performed by transforming the window into a single value, which is the average of the values present in it. This returns a smoother feature map compared to the max pooling layer, which returns a feature map focusing on prominent features. The average pooling layer can be mathematically represented as shown in Equation (6):

$$Y[i,j,c] = \frac{1}{k_h * k_w} \sum_{p=0}^{k_h-1} \sum_{q=0}^{k_w-1} X[i * s_h + p\ ,\ j * s_w + q, c] \quad (6)$$

where Y is the output after the pooling function, X is the input feature map, k_h is the height of the feature map, and k_w is the width of the feature map. s_w and s_h are the stride values for height and width while sliding through the input feature map.

3.4.4.5 *(v)* Concatenation layer

The concatenation layer concatenates the inputs having the same size in all dimensions except the concatenation dimension, received by the layer along a specified dimension. This layer is used whenever we want to merge the information from different parts of the network or data modalities. The concatenation operation takes place by combining multiple input tensors by stacking them along the specified axis, resulting in a single tensor with an increase in size. The layer is mathematically expressed as shown in Equation (7):

$$O[i,j,c] = \begin{cases} A[i,j,c] & if\ \ 0 \le c < C_1 \\ B[i,j,c-C_1] & if\ \ C_1 \le c \le C_1 + C_2 \end{cases} \quad (7)$$

where O is the output, A is the first input tensor with C_1 channels and B is the second input tensor with C_2 channels for the concatenation layer, i represents the height dimension and ranges from 0 to H, j represents the width dimension and ranges

from j to W, and c represents the channels and ranges from 0 to C_1+C_2.

3.4.4.6 *(vi)* Addition layer

This layer adds inputs from multiple neural network element-wise. This operation can be performed when the input tensors have the same shape. This is done so that the information flows seamlessly through the network just by the addition of the output of one layer to the output of the previous layer. This layer is mathematically represented as shown in Equation (8):

$$O[i,j,c] = A[i,j,c] + B[i,j,c] \tag{8}$$

where O is the output, A is the first input tensor and B is the second input tensor for the addition layer, i represents the height dimension and ranges from 0 to H, j represents the width dimension and ranges from j to W, and c represents the channels and ranges from 0 to C.

3.4.4.7 *(vii)* Batch normalization layer

This layer helps in making neural networks faster and more stable by performing standardization and normalization operations in the feature map that is provided as input to the layer. The normalization process is carried out in two steps:

- Normalization
- Rescaling and offsetting

Before performing normalization, the data are fed into the layer in the form of mini batches. The mean and standard deviations of these mini batches can be found using the following equations shown in Equations (9, 10):

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{9}$$

and

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2 \tag{10}$$

where μ and σ are the mean of the values in the ith value in the mini-batch x of size m.

The main purpose of normalization is to transform the data to have a mean equal to 0 and standard deviation equal to 1, which is carried out using the expression as shown in Equation (11):

$$x_{i(norm)} = \frac{x_i - \mu}{\sigma + \epsilon} \tag{11}$$

Two learnable parameters γ and β are used for rescaling and offsetting, respectively, thereby normalizing each batch accurately. This is represented using the expression shown in Equation (12):

$$x_i = \gamma x_{i(norm)} + \beta \tag{12}$$

where x_i is the ith value of mini batch x and $x_{i(norm)}$ is the normalized ith value of mini batch x.

3.4.4.8 *(viii)* Dropout layer

The dropout layer acts as a mask to nullify some of the neurons' contributions towards the next layer while the rest of the neurons remain unmodified. It aims to prevent overfitting, avoid dependency on a specific neuron during training, and ensure better generalization from the model. The neurons are nullified using a probability for random exclusion such that they behave like they are not part of the architecture. The layer can be mathematically represented as shown in Equations (13, 14):

$$O = X * M \quad during \quad training \tag{13}$$

and

$$O = X * (1-p) during \quad testing \tag{14}$$

where O is the output, X is the input, and p is the probability, and it is scaled to a factor $(1 - p)$ during output since the dropout will be turned off during the testing phase. M is a binary mask with the shape same as X and each element of M is set as 0 or 1 depending on p.

3.4.4.9 *(ix)* Global average pooling layer

The global average pooling layer is a pooling layer that performs down-sampling. Unlike the usual pooling layer, the global pooling layer condenses the feature maps into a one-dimensional mapping that can easily be read by the single dense classification layer. The mathematical representation is as shown in Equation (15):

$$O = \frac{1}{H*W} * \sum_{i=0}^{H}\sum_{j=0}^{W}(F[i,\ j]) \tag{15}$$

3.4.4.10 *(x)* Flatten layer

This layer performs the flattening operation that reshapes the input received into a single-dimensional feature vector without affecting the batch. It is done to allow the fully connected layers to operate on the multi-dimensional feature maps since the fully connected layers can only be trained with single-dimensional feature vectors.

3.4.4.11 *(xi)* Fully connected layer

The fully connected layer or simply the dense layer is a CNN layer where all the neurons or nodes in one layer is connected to every node to the next layer. This layer works with activation functions such as the ReLU during feature extraction and softmax during multiclass classification. It is represented as a mathematical function as shown in Equation (16):

$$\mathrm{O} = \mathrm{f}(\mathrm{W}*\mathrm{X} + \mathrm{b}) \tag{16}$$

where X is the input, O is the output, W is the weight matrix, b is the bias vector, and f is the activation layer, which would be ReLU in case of feature extraction and softmax in case of classification.

3.4.4.12 *(xii)* ReLU activation layer

The ReLU is a piecewise linear function used to introduce non-linearity into the feature map obtained as output before the activation function is applied. The ReLU function works by applying a simple thresholding operation where the positive values remain the same while the negative values become zero. The ReLU activation function can be expressed mathematically as shown in Equation (17)

$$f = \max(x, 0) \tag{17}$$

where x is the input given into the function and f is the output obtained.

3.4.5 Classification

3.4.5.1 *(i)* Softmax activation

The softmax activation function is responsible for the multi-class classification of the vector obtained from the convolution layers after the feature extraction phase in the output layer. It works by calculating the exponent of each entry in the vector and dividing the value by the sum of all the exponents in the vector as shown in Equation (18).

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} \tag{18}$$

where x is the input vector and i is the ith entry in the input vector with N entries. The denominator of the softmax activation is the sum of the exponents of the entries. This is done for the conversion of N real number entries into a probability distribution of N possible outcomes.

3.4.5.2 *(ii)* Categorical cross-entropy loss function

This loss function (also known as softmax loss) is used with a CNN to provide an output for the probability of each image over N different classes. This function is a combination of softmax activation and the cross-entropy loss function and is thus useful during multi-class classification. Its use allows the comparison of the target and predicted values by the CNN model as an output, thereby measuring the modeling efficiency of the training data by the CNN. The objective of this loss function is to calculate the difference between the ground truth and predicted class distribution. Techniques like gradient descent are used to adjust the weights and biases for minimalization of this loss, thereby improving the predictions. The categorical cross-entropy loss function is written as the negation of logarithmic function of the softmax function as shown in Equation (19):

$$CE = -\log\left(\frac{e^{x_p}}{\sum_{j}^{N} e^{x_j}}\right) \tag{19}$$

where CE is the cross-entropy loss, x_p is the positive class' CNN score, N is the number of classes for classification, and x_j is the jth class' score.

To backpropagate through the network and optimize the defined loss function resulting in tuning the net parameters, the loss' gradient is calculated with respect to the CNN's output neurons given by the gradient of the cross-entropy loss with respect to each CNN's class score. The derivatives are represented mathematically as shown in Equations (20, 21):

Derivative with respect to positive class:

$$\frac{\partial}{\partial x_p}\left(-\log\left(\frac{e^{x_p}}{\sum_{j}^{N} e^{x_j}}\right)\right) = \frac{e^{x_p}}{\sum_{j}^{N} e^{x_j}} - 1 \tag{20}$$

Derivative with respect to negative class:

$$\frac{\partial}{\partial x_n}\left(-\log\left(\frac{e^{x_p}}{\sum_{j}^{N} e^{x_j}}\right)\right) = \frac{e^{x_n}}{\sum_{j}^{N} e^{x_j}} \tag{21}$$

where x_n is the score of any negative class in N other than N_p, which consists of the positive classes.

3.4.6 Optimizer

3.4.6.1 *(i)* Adam optimizer

The Adam optimizer is an extension of the stochastic gradient descent (SGD) algorithm based on adaptive moment estimation, which takes advantage of two principles, namely, the momentum and root mean square propagation (RMSprop). The momentum technique is used to accelerate convergence in gradient descent by adding the fraction of the previous gradient update with the current update, reducing the oscillations. The convergence process speeds up along shadow dimensions, which assists optimization. RMSprop adapts the learning rate for each parameter individually by maintaining a moving average of squared gradients. This helps in scaling learning rates and making the optimization process more robust. With the help of these two methods, the following are obtained as shown in Equations (22, 23):

$$m_t = \beta_1 m_{t_{-1}} + (1 - \beta_1)\left[\frac{\delta L}{\delta W_t}\right] \tag{22}$$

$$v_t = \beta_2 v_{t_{-1}} + (1 - \beta_2)\left[\frac{\delta L}{\delta W_t}\right]^2 \tag{23}$$

where m_t is the estimate of the first-order moment, which is the aggregate of gradients at time t, v_t is the estimate of the second-order moment, which is the sum of the squares of the past gradients at time t, β_1 is the decay rate of average of gradient in the momentum principle, and β_2 is the decay rate of average of gradient in the RMSprop principle. The moment estimates m_t and v_t can be called the weight parameters.

In the Adam optimizer, the bias-corrected weights are considered such that the weight parameters will not be biased towards 0. The bias-corrected weight parameters are as shown in Equations (24, 25):

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t} \tag{24}$$

$$\widehat{v_t} = \frac{v_t}{1-\beta_2^t} \tag{25}$$

These bias-corrected weight parameters are used in the general weight update equation as shown in Equation (26):

$$w_{t+1} = w_t - \widehat{m_t}\left(\frac{\alpha}{\sqrt{\widehat{v_t}}+\epsilon}\right) \tag{26}$$

where α is the learning rate or the step size parameter and ϵ is a small positive constant to avoid division by 0.

3.4.6.2 *(ii)* Enhanced weighted gradient optimizer

This is a modified Adam optimizer that accepts a custom weight as a parameter and incorporates the gradients multiplied by the custom weight into its operation. The custom weights are given as a parameter and are introduced into the gradients with the values being multiplied. The modified values are introduced into the Adam optimizer and then used in our ensemble model. The updated weight with the custom weight parameter before optimization is as shown in Equation (27):

$$\omega = \gamma \cdot w_t \tag{27}$$

This updated weight ω is introduced to the weight update process as shown in Equation (28).

$$w_{t+1} = \omega - \widehat{m_t}\left(\frac{\alpha}{\sqrt{\widehat{v_t}}+\epsilon}\right) \tag{28}$$

where γ is the custom weight parameter, α is the learning rate or the step size parameter, ϵ is a small positive constant to avoid division by 0, w_t is the existing weight before the optimization process, and w_{t+1} is the updated weight after optimization. $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected weight parameters. The procedure for enhanced weighted gradient optimizer is shown in Algorithm 9.

```
Initialize epoch ← 0;
while EnsembleModel is running do
    forward pass
        predictions ← EnsembleModel(batch i);
        loss ← CategoricalCrossEntropy(predictions,
        batch i _labels);
    end forward pass
    backward pass
        gradient, ∇_L ← ∂L/∂θ;
        Custom weights, ∇L custom ←∇L · custom_weight;
        Update EnsembleModel parameters,
        θ ← θ − α · ∇L ;
    end backward pass
    early stopping check
        Monitor validation loss Lval
        Criteria: if Lval does not improve for 4
        consecutive epochs then end training
        if Lval ≤ best loss then
            best loss ← Lval
            patience counter ← 0
        else
            patience counter ← patience counter + 1
            if patience counter ≥ 4 then
                Break training loop;
            end if
        end else
    end early stopping check
end while
end
```

Algorithm 9. Enhanced weighted gradient optimizer.

4 Results and discussion

In this paper, the focus of research starts by addressing a pressing issue in agriculture: the management of plant diseases, with a specific focus on tomato plants. Researchers have employed complex deep learning methodologies and machine learning models to tackle this challenge. This paper strives to revolutionize the ways to identify plant diseases, especially those affecting tomato plants, and manage them accordingly.

The study adopts data analysis and image preprocessing techniques to ensure that the dataset used is well-balanced and that the quality of the images is optimized for deep learning models. It uses methods such as median filtering, resized cropping, and brightness normalization to enhance the features derived from them. This meticulous attention to data quality and balance is crucial in developing a reliable disease classification system. To extract relevant features from the tomato leaf images, the research leverages two transfer learning models, VGG-16 and NASNet. Furthermore, these models are fine-tuned, allowing them to adapt to the specific characteristics of the dataset. This adaptability showcases the potential for pre-trained models to significantly improve classification accuracy when applied to particular datasets.

One of the key novelties is the incorporation of an ensemble model with an EMA function and an EWGO. This innovative approach optimizes the learning process, resulting in a more effective and accurate disease classification system. It stands as a promising method to enhance the performance of machine learning models in agriculture.

4.1 Performance metrics

The evaluation of the models is robust, using a variety of performance metrics, including the confusion matrix, specificity, accuracy, loss, precision, recall, F1-score, ROC curve, AUC, and misclassification rate. These metrics provide a comprehensive assessment of the model's effectiveness, making it clear that the research is backed by rigorous analysis and empirical evidence. The overall proposed architecture is shown in Figure 1, the training data distributions of the dataset is shown in Figure 2, the validated data distributions is shown in Figure 3, images of dataset after preprocessing is shown in Figure 4, images of tomato leaves

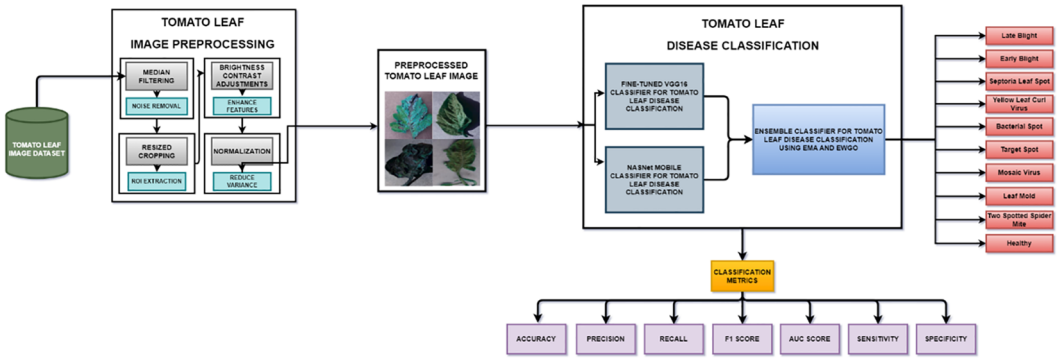


FIGURE 1
Overall proposed architecture for tomato leaf disease classification.

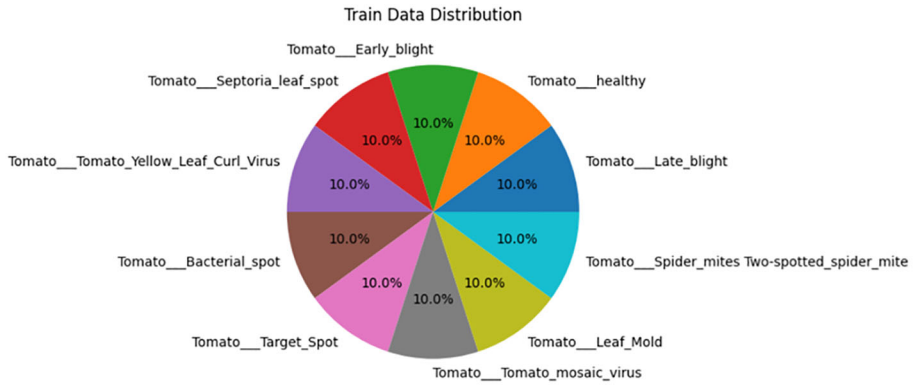


FIGURE 2
Training data distributions of tomato leaf images.

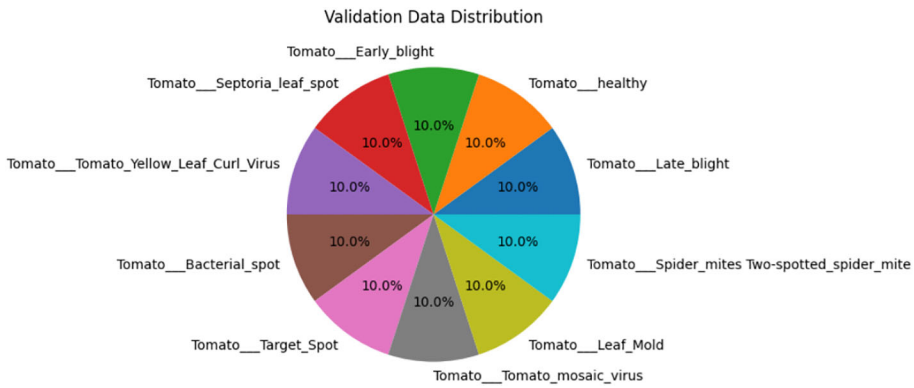


FIGURE 3
Validation data distributions of tomato leaf images.

observed at each preprocessing step in shown in Figure 5, layer architecture for VGG-16 tomato leaf disease classifier is shown in Figure 6, layer architecture for NASNet mobile tomato leaf disease classifier is shown in Figure 7 and layer architecture for ensemble model is shown in Figure 8.

4.1.1 Confusion matrices

The confusion matrix is an $n \times n$ matrix where the rows represent the actual classes while the columns represent the predicted class. The data points are stored in the matrix in cells corresponding to the specific actual class and specific predicted class as count values.

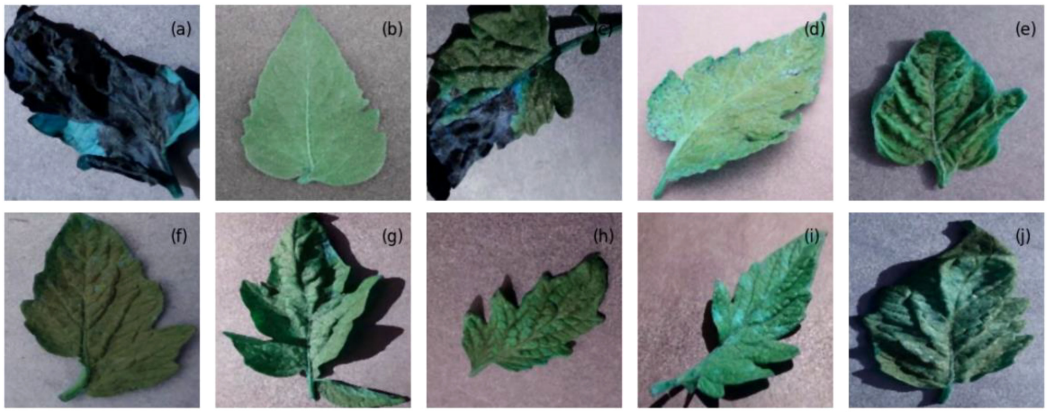


FIGURE 4
The images from the tomato leaf dataset after preprocessing, representing **(A)** late blight, **(B)** healthy, **(C)** early blight, **(D)** septoria leaf spot, **(E)** yellow leaf curl virus, **(F)** bacterial spot, **(G)** target spot, **(H)** mosaic virus, **(I)** leaf mold, and **(J)** two spotted spider mite.

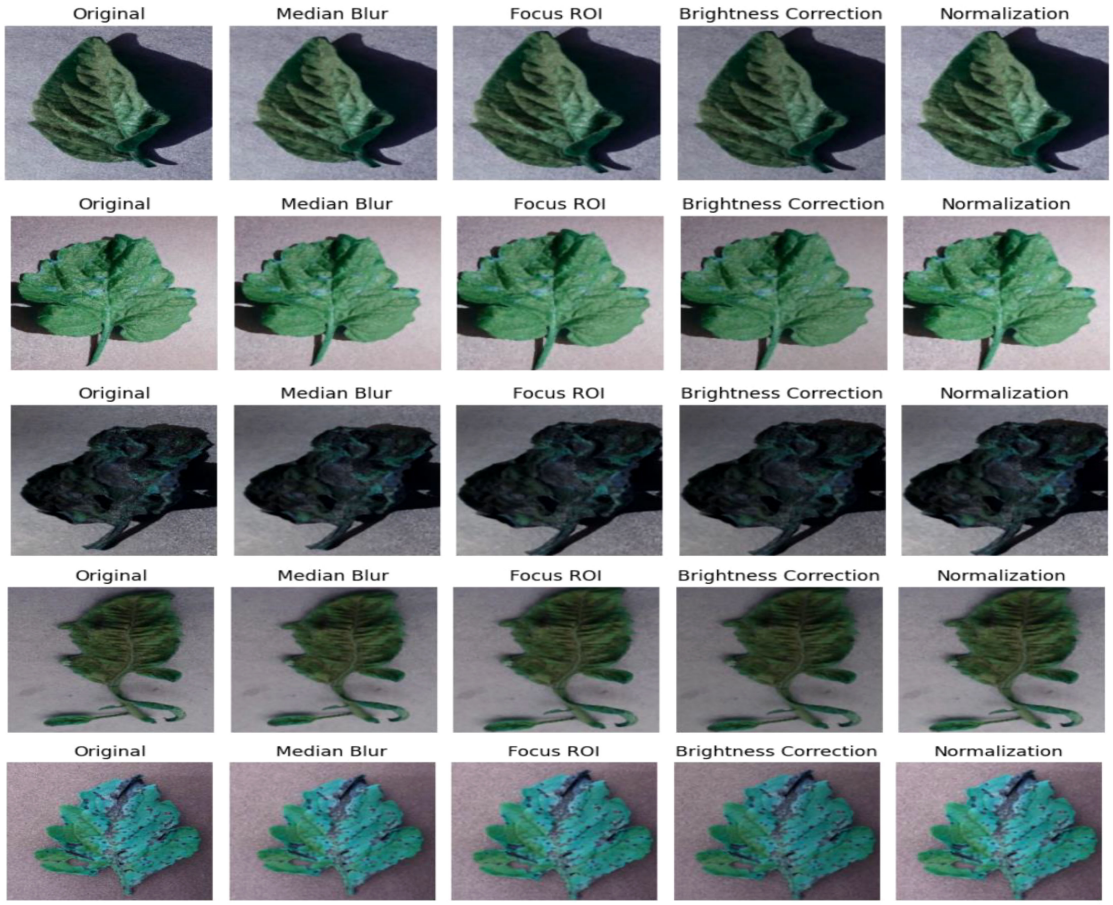


FIGURE 5
Images of tomato leaves observed at each preprocessing step.

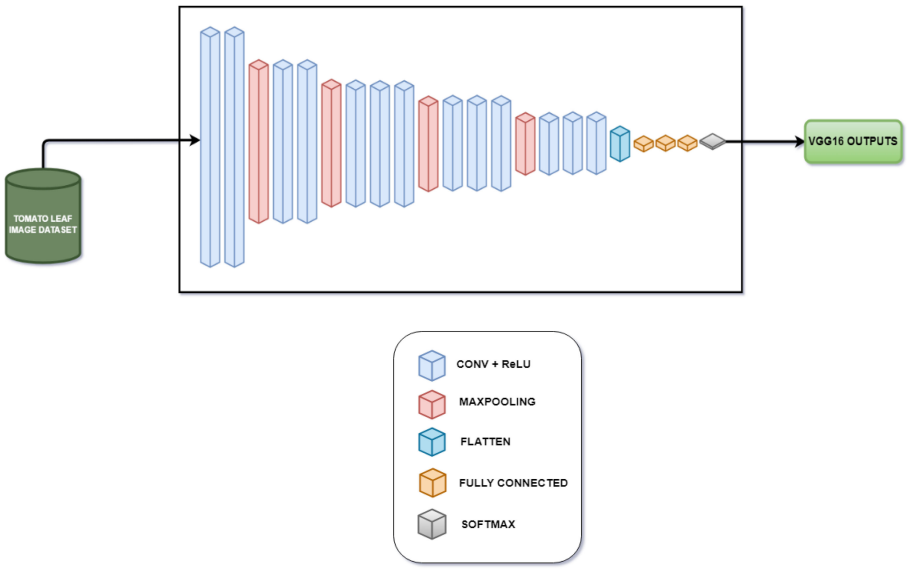


FIGURE 6
Layer architecture for VGG16 tomato leaf disease classifier.

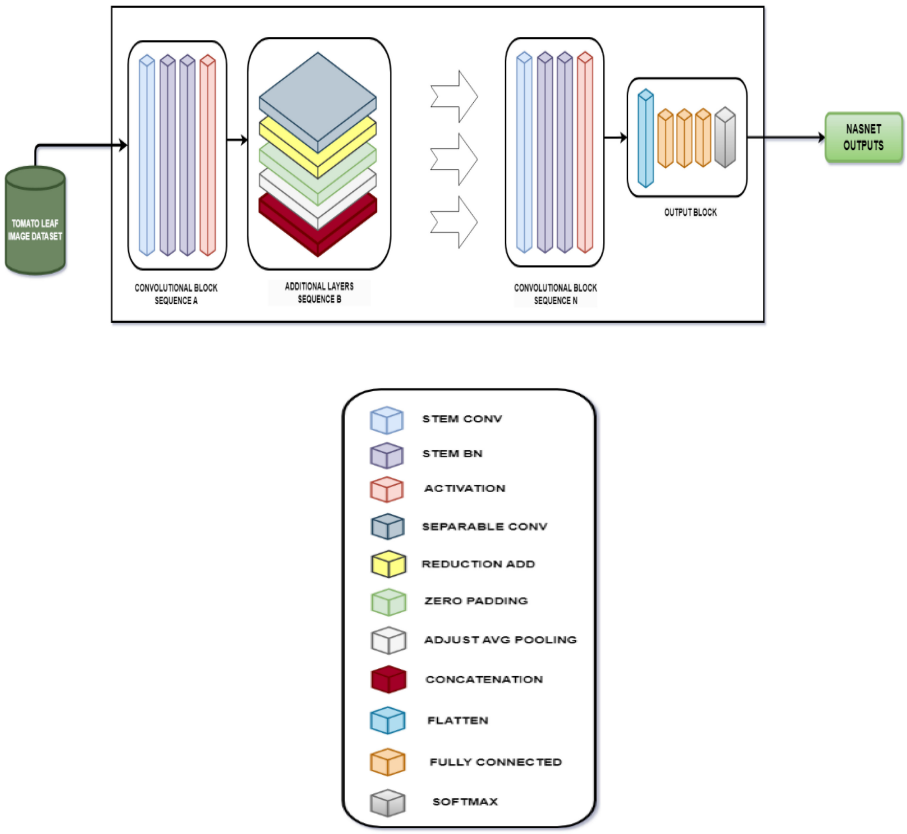


FIGURE 7
Layer architecture for NASNet mobile tomato leaf disease classifier.

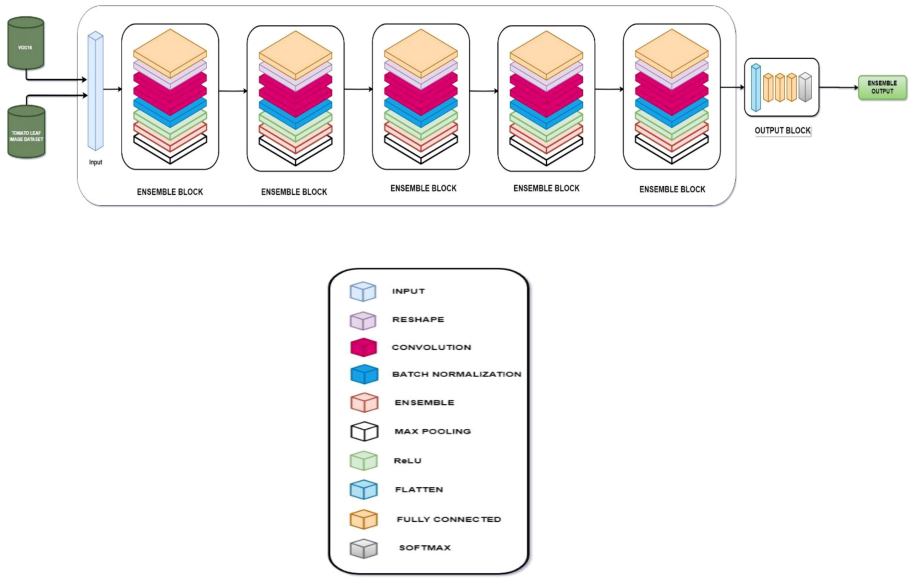


FIGURE 8
Layer architecture for ensemble model.

The above confusion matrix consists of the values predicted by the proposed model corresponding to the actual value. The confusion matrix of the proposed model is shown in Figure 9, the confusion matrix of the VGG-16 fine-tuned model is shown in Supplementary Figure 2, the confusion matrix of the NASNet model is shown in Supplementary Figure 3, the precision values of VGG-16, NASNet, and the proposed model is shown in Supplementary Figure 4.

4.1.2 Specificity

The specificity is the ratio of true negatives to the actual number of negative instances in a specific class. This is a metric to measure the ability of the classifier for correct identification of negative instance within a specific class.

It is mathematically expressed as shown in Equation (29): Specificity of VGG-16, NASNet, and the proposed model is shown in Figure 10.

$$Specificity = \frac{\sum_{k=1}^{n} TN_k}{\sum_{k=1}^{n} (TN_k + FP_k)} \tag{29}$$

4.1.3 Accuracy

Accuracy can be defined as the number of correctly classified images to the total number of images in the dataset. This can be expressed mathematically as shown in Equation (30): Accuracy curves of VGG-16, NASNet, and the proposed model is shown in Figure 11.

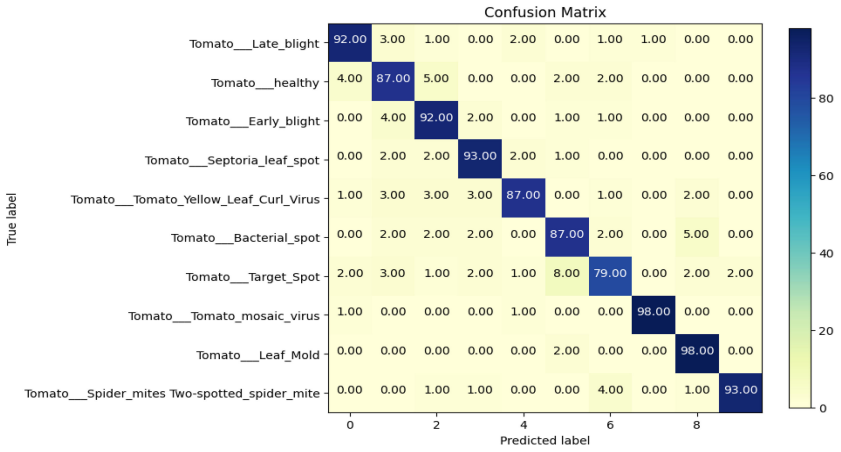


FIGURE 9
Confusion matrix of the proposed model.

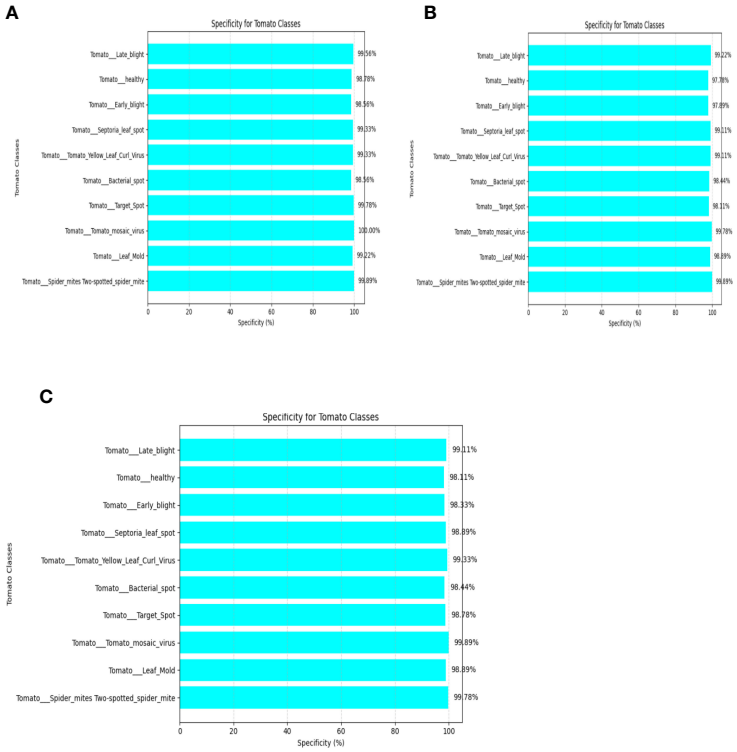


FIGURE 10
Specificity of **(A)** VGG-16, **(B)** NASNet, **(C)** proposed model.

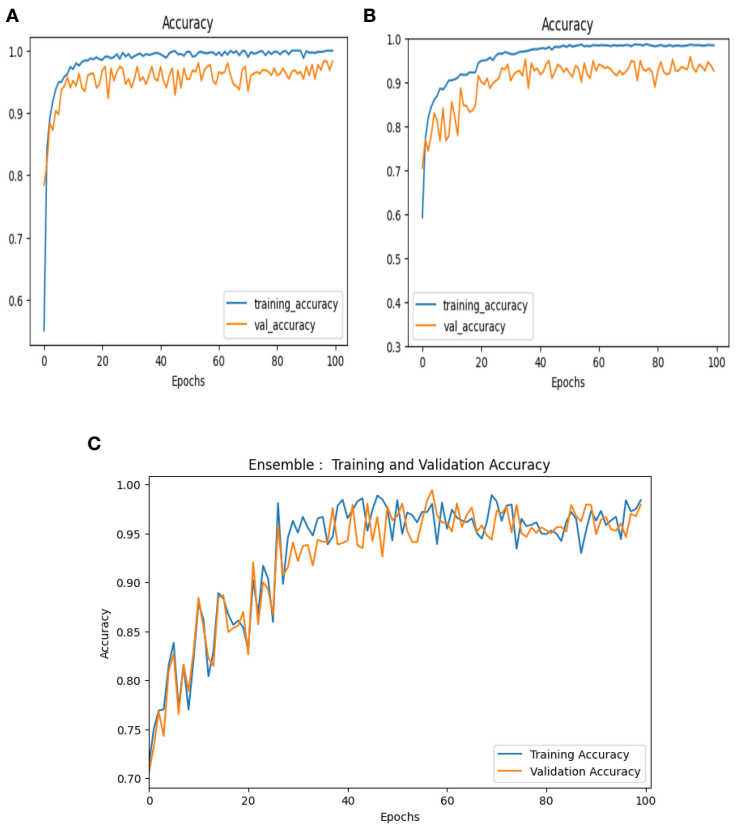


FIGURE 11
Accuracy curves of **(A)** VGG-16, **(B)** NASNet, **(C)** proposed model.

$$Accuracy = \frac{\sum_{k=1}^{n}(TP_k + TN_k)}{\sum_{k=1}^{n}(TP_k + TN_k + FP_k + FN_k)} \quad (30)$$

4.1.4 Loss

Loss is represented as the measure of the model's performance regarding the ability to minimize the difference between the predicted and actual values. In our case, we have used the categorical cross-entropy loss function. Loss Curves of VGG-16, NASNet and the proposed Model are shown in Figure 12.

$$Precision = \frac{\sum_{k=1}^{n} TP_k}{\sum_{k=1}^{n}(TP_k + FP_k)} \quad (31)$$

4.1.5 Precision

Precision is calculated as the ratio of the true total number of instances that are correctly identified as positive by the classifier to the total number of instances identified as positive by the classifier. This is mathematically expressed as shown in Equation (31):

$$Recall = \frac{\sum_{k=1}^{n} TP_k}{\sum_{k=1}^{n}(TP_k + FN_k)} \quad (32)$$

4.1.6 Recall

Recall or sensitivity is the ratio of the number of true positives to the sum of the number of true-positive and false-negative instances in a specific class. This is a metric to measure the ability of the classifier for correct identification of positive instances within a specific class. The recall curves of VGG-16, NASNet, and the proposed model is shown in Figure 13.

It is mathematically expressed as shown in Equation (32):

$$F1\ \ Score = \frac{2\sum_{k=1}^{n} TP_k}{\sum_{k=1}^{n}(2TP_k + TN_k + FP_k)} \quad (33)$$

4.1.7 F1-score

The F1-score is utilized for striking a balance between minimizing the false positives and false negatives and is used as a combination of both precision and recall. Thus, it can be mathematically expressed as shown in Equation (33) and the F1 score curves of VGG-16, NASNet, and the proposed model is shown in Figure 14.

4.1.8 ROC curve and AUC

The receiver operating characteristic (ROC) curve is a graphical representation that consists of the performance of the model in various classification thresholds and is plotted with sensitivity against specificity, thereby visualizing the trade-off between both metrics. AUC helps in quantifying the overall performance of the classifier, which is measured as the area under the ROC curve and the ROC curves of VGG-16, NASNet, and the proposed model is shown in Figure 15.

4.1.9 Misclassification rate

The error rate can be defined as the number of inputs in a particular, which are classified into a wrong class; this can be expressed

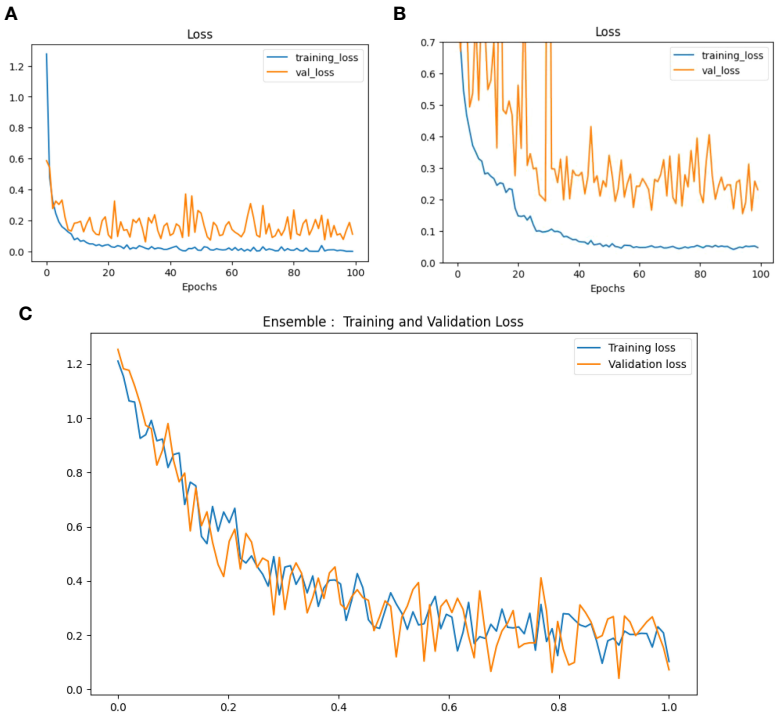


FIGURE 12
Loss curves of **(A)** VGG-16, **(B)** NASNet, **(C)** proposed model.

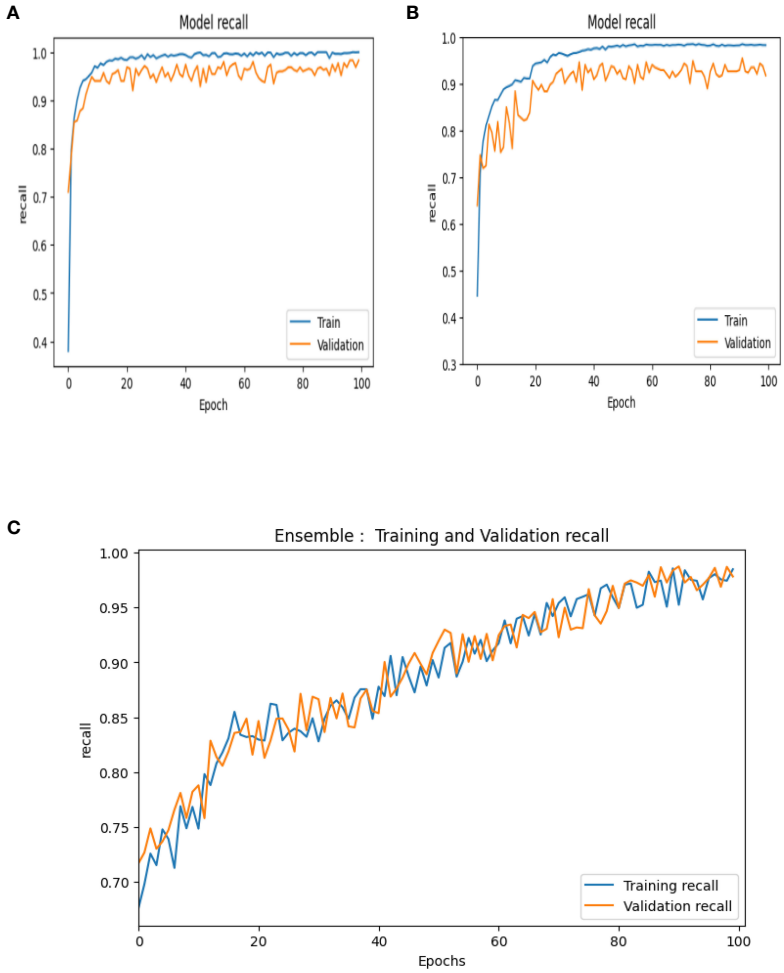


FIGURE 13
Recall curves of **(A)** VGG-16, **(B)** NASNet, **(C)** proposed model.

mathematically as shown in Equation (34): Misclassification rates in VGG-16, NASNet, and proposed model is shown in Figure 16.

$$Error\ \% = \frac{No\ of\ Misclassified\ Instances\ in\ a\ class}{Total\ Number\ of\ Instances\ in\ a\ class} \quad (34)$$

4.2 Performance analysis

The comparison of the three models in the context of the above explained metrics, namely, (a) VGG-16, (b) NASNet, and (c) proposed model, is presented below in graphical representations.

4.3 Interpretation

The above computed performance metrics and the respective graphical representations are proof that the proposed deep learning technique, the suitable application of the ensemble model, and the enhanced classifier and optimizer used have shown a tangible increase of the feasibility in the disease prediction for the given series of input images of tomato leaves. It also proves that the preprocessing procedure applied is a fitting one. The performance values observed for accuracy, loss, precision, recall, ROC, and F1-score are 98.7%,<4%, 97.9%, 98.6%, 99.97%, and 98.7% respectively. It is apparent that the results obtained show significant improvement compared with those shown by conventional and present techniques as explained in the literature. The performance scores recorded for the existing models in the literature are tabulated below. The techniques studied do not record all the performance metrics as in the proposed model in this work. One parameter that is considered in all the models, namely, "accuracy", is exponentially high in the proposed approach. The performance comparison of the proposed model with existing models is shown in Table 1.

4.4 Testing of hypotheses

In order to provide a statistical analysis on the proposed work, testing of hypothesis was carried out in this work. It consists of three hypotheses including a Null hypothesis given in Table 2.

Hypothesis 1: There is a significant influence between season and tomato leaf diseases.

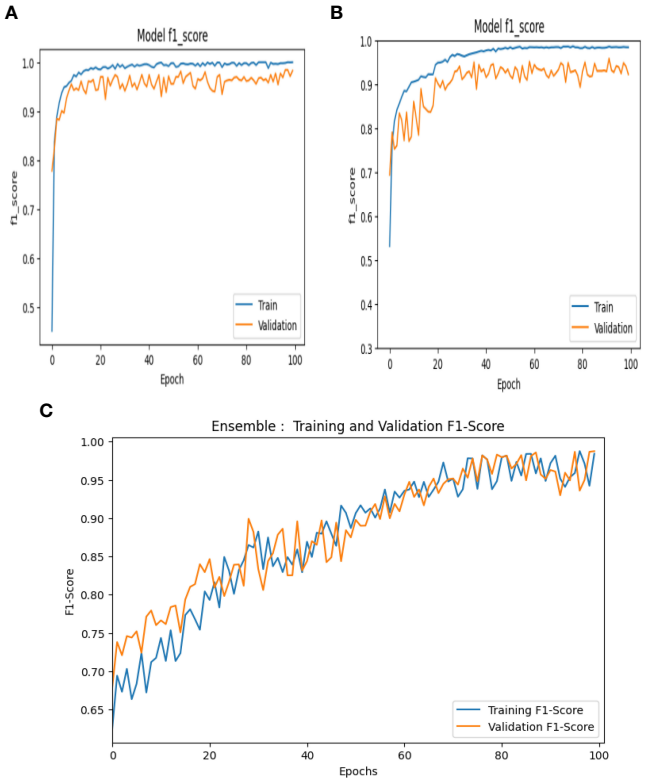


FIGURE 14
F1 score curves of **(A)** VGG-16, **(B)** NASNet, and **(C)** the proposed model.

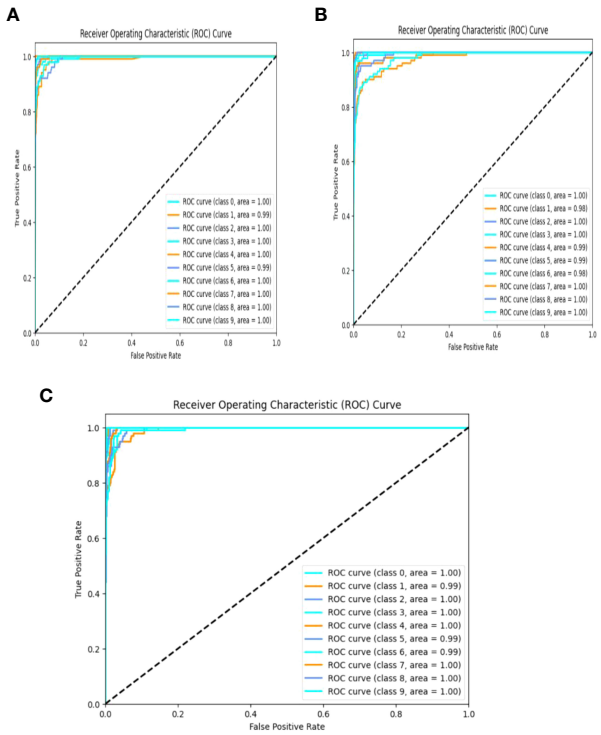


FIGURE 15
ROC curves of **(A)** VGG-16, **(B)** NASNet, and **(C)** the proposed model.

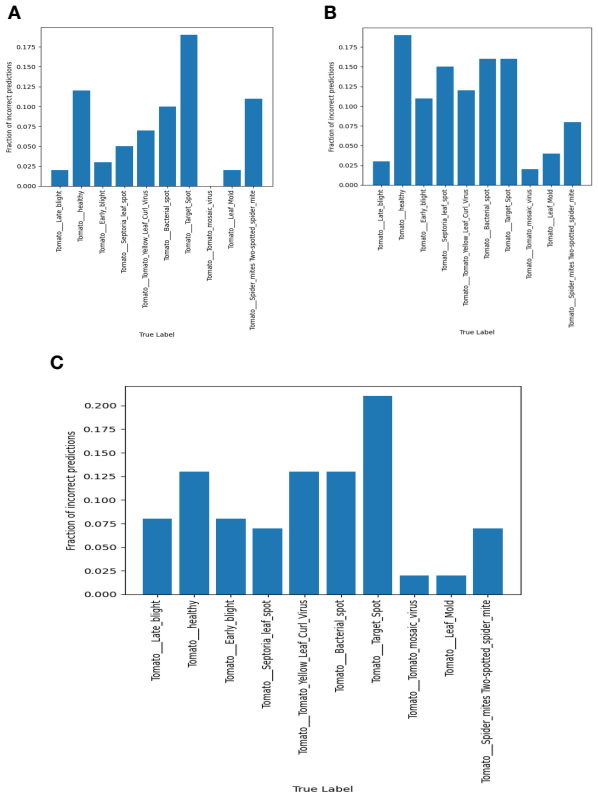


FIGURE 16
Misclassification rates in **(A)** VGG-16, **(B)** NASNet, **(C)** proposed model.

Hypothesis 2: There is no relationship between the occurrence of tomato leaf disease and the environment.

4.5 Testing of Hypothesis 1

As the *p*-value in this test is greater than 0.01, the given null hypothesis can be accepted at the 1% significance level. Hence, there is a significant influence between season and tomato leaf diseases.

Table 3 shows the chi-square test for analyzing the relationship between the deep learning classifier vs. tomato leaf disease detection.

4.6 Testing of Hypothesis 2

H0: There is no relationship between the selection of the deep learning classifier vs. tomato leaf disease detection for performing accurate detection of the disease.

Since the value of *p* is less than 0.5, this hypothesis, which is shown in Table 3, is rejected at the 5% significance level. Therefore, it is concluded that there is a strong and direct relationship between the selection of the deep learning classifier and tomato leaf disease detection from tomato leaf images for performing accurate detection of the disease.

TABLE 1 Performance comparison.

Models	Performance scores (all in %)							
	Specificity	Accuracy	Recall	Precision	F1-score	Loss	ROC	Misclassification
AlexNet (Wang et al., 2017)	–	91.00	91.00	91.0	91.00	-	-	-
GoogLeNet (Wang et al., 2017)	–	94.8	94	94	94	-	-	-
VGG-16 (Wang et al., 2017)	–	95	95	95	95	-	-	-
VGG-16 (Bracino et al., 2020)	–	90.40	-	-	-	-	-	-
LBP M-SVM (Wang et al., 2017)	90.23	97.20	90.75	93.50	-	-	-	-
GPR Quadratic SVM (Ashwinkumar et al., 2022)	–	83.30	-	-	-	-	86.00	-
OMCNN (Khan et al., 2019)	–	98.7	98.2	-	98.5	-	–	-
Proposed adaptive ensemble model	**98.9**	**98.7**	**98.6**	**97.9**	**98.7**	**<4**	**99.97**	**<9**

TABLE 2 H0: There is a significant influence between season and tomato leaf diseases.

Reason for tomato leaf disease	Weighted mean using experiments (observed value *O*)	Weighted mean based on computation (expected value *E*)	$(O - E)^2$	$Value\ is\ \chi^2\ /\ \chi^2 \sum \frac{\leftarrow O\ E \leftarrow^2}{E}$	*p*-value (with 6 dof)
Fungi	7.692	4.649	0.649	4.11	0.65
Fertilizer use	6.329	3.548	0.779		
Bacteria	7.947	4.979	0.612		
Virus	7.309	3.648	0.999		
Viroids	7.519	4.718	0.60		
Geographical Region	6.418	4.269	0.499		

TABLE 3 Analysis of deep learning algorithm's role in tomato leaf disease detection.

Important metric applied on the algorithm	Chi-square value	*p*-value	Mean availability	
			Up to 80%	Above 80%
Accuracy of classification	1.91	0.41	25	11

5 Conclusion and future work

In this research paper, a new ensemble classifier along with an EMA function with temporal constraints, an EWGO that is integrated with two CNN models, namely, VGG-16 and NASNet, has been proposed for the effective detection of diseases in tomato leaves at an early state. This integration of state-of-the-art deep learning CNN technologies with a gradient optimizer and EMA function with temporal constraints provides meticulous data analysis. The proposed model uses image enhancement techniques, and groundbreaking ensemble models underscore a comprehensive approach to tomato leaf disease classification. The amalgamation of image preprocessing, transfer learning, and the pioneering ensemble model with EWGO exhibits promising outcomes in disease classification and increases detection accuracy compared with the existing systems. The main limitation of this work is the lack of time during training. However, an optimizer is added to this work to solve the training time problem. In the future, the implications of this research shall be extended to areas like crop health, global food security, sustainable agriculture, and environmental preservation, underscoring its value within the realm of plant pathology and agriculture.

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

PV: Conceptualization, Methodology, Writing – original draft, Writing – review & editing. AS: Data curation, Investigation, Writing – original draft, Writing – review & editing. JP: Methodology, Supervision, Writing – original draft, Writing – review & editing. SV: Software, Visualization, Writing – original draft, Writing – review & editing. SK: Software, Visualization, Writing – original draft, Writing – review & editing. SA: Data curation, Writing – original draft, Writing – review & editing. AA: Software, Visualization, Writing – original draft, Writing – review & editing. AK: Data curation, Writing – original draft, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

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.2024.1382416/full#supplementary-material

References

Abed, S. H., Al-Waisy, A. S., Mohammed, H. J., and Al-Fahdawi, S. (2021). A modern deep learning framework in robot vision for automated bean leaves diseases detection. *Int. J. Intell. Robot. Appl.* 5, 235–251. doi: 10.1007/s41315-021-00174-3

Abouelmagd, L. M., Shams, M. Y., Marie, H. S., and Hassanien, A. E. (2024). An optimized capsule neural networks for tomato leaf disease classification. *EURASIP J. Image Video Process.* 2024, 2. doi: 10.1186/s13640-023-00618-9

Agarwal, M., Singh, A., Arjaria, S., Sinha, A., and Gupta, S. (2020). ToLeD: Tomato leaf disease detection using convolution neural network. *Proc. Comput. Sci.* 167, 293–301. doi: 10.1016/j.procs.2020.03.225

Al-gaashani, M. S., Shang, F., Muthanna, M. S., Khayyat, M., and Abd El-Latif, A. A. (2022). Tomato leaf disease classification by exploiting transfer learning and feature concatenation. *IET Image Process.* 16, 913–925. doi: 10.1049/ipr2.12397

Andrushia, A. D., and Patricia, A. T. (2020). Artificial bee colony optimization (ABC) for grape leaves disease detection. *Evol. Syst.* 11, 105–117. doi: 10.1007/s12530-019-09289-2

Anusha, B., Geetha, P., and Kannan, A. (2022). "Parkinson's disease identification in homo sapiens based on hybrid ResNet-SVM and resnet-fuzzy SVM models". *J. Intell. Fuzzy Syst.* 43, 2711–2729. doi: 10.3233/JIFS-220271

Arshad, F., Mateen, M., Hayat, S., Wardah, M., Al-Huda, Z., Gu, Y. H., et al. (2023). PLDPNet: End-to-end hybrid deep learning framework for potato leaf disease prediction. *Alexandria Eng. J.* 78, 406–418. doi: 10.1016/j.aej.2023.07.076

Ashwinkumar, S., Rajagopal, S., Manimaran, V., and Jegajothi, B. (2022). Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks. *Mater. Today: Proc.* 51, 480–487. doi: 10.1016/j.matpr.2021.05.584

Bennet, J., Ganaprakasam, C. A., and Arputharaj, K. (2014). A discrete wavelet-based feature extraction and hybrid classification technique for microarray data analysis. *Sci. World J.* 2014, 1–9. doi: 10.1155/2014/195470

Bhandari, M., Shahi, T. B., Neupane, A., and Walsh, K. B. (2023). Botanicx-ai: Identification of tomato leaf diseases using an explanation-driven deep-learning model. *J. Imag.* 9, 53. doi: 10.3390/jimaging9020053

Bracino, A. A., Concepcion, R. S., Bedruz, R. A. R., Dadios, E. P., and Vicerra, R. R. P. (2020). "Development of a hybrid machine learning model for apple (Malus domestica) health detection and disease classification," in *2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*. 1–6.

Chang, B., Wang, Y., Zhao, X., Li, G., and Yuan, P. (2024). A general-purpose edge-feature guidance module to enhance vision transformers for plant disease identification. *Expert Syst. Appl.* 237, 121638. doi: 10.1016/j.eswa.2023.121638

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

Chouhan, S. S., Singh, U. P., and Jain, S. (2021). Automated plant leaf disease detection and classification using fuzzy based function network. *Wireless Pers. Commun.* 121, 1757–1779. doi: 10.1007/s11277-021-08734-3

Das, S., and Sengupta, S. (2020). "Feature extraction and disease prediction from paddy crops using data mining techniques," in *Computational Intelligence in Pattern Recognition* (Springer), 155–163. doi: 10.1007/978-981-15-2449-3

Debnath, A., Hasan, M. M., Raihan, M., Samrat, N., Alsulami, M. M., Masud, M., et al. (2023). A smartphone-based detection system for tomato leaf disease using efficientNetV2B2 and its explainability with artificial intelligence (AI). *Sensors* 23, 8685. doi: 10.3390/s23218685

Demilie, W. B. (2024). Plant disease detection and classification techniques: a comparative study of the performances. *J. Big Data* 11, 5. doi: 10.1186/s40537-023-00863-9

Dhalia Sweetlin, J., Khanna Nehemiah, H., and Kannan, A. (2016). Patient-specific model based segmentation of lung computed tomographic images. *J. Inf. Sci. Eng.* 32, 1373–1394.

Elgin Christo, V. R., Khanna Nehemiah, H., Minu, B., and Kannan, A. (2019). Correlation-based ensemble feature selection using bioinspired algorithms and classification using backpropagation neural network. *Comput. Math. Methods Med.* 7398307, 1–17. doi: 10.1155/2019/7398307

Elizabeth, D. S., Nehemiah, H. K., Raj, C. S. R., and Kannan, A. (2012). A novel segmentation approach for improving diagnostic accuracy of CAD systems for detecting lung cancer from chest computed tomography images. *J. Data Inf. Qual. (JDIQ)* 3, 1–16. doi: 10.1145/2184442.2184444

Gadade, H. D., and Kirange, D. K. (2022). Deep learning for tomato leaf disease detection for images captured in varying capturing conditions 2191–2200.

Ganapathy, S., Sethukkarasi, R., Yogesh, P., Vijayakumar, P., and Kannan, A. (2014). "An intelligent temporal pattern classification system using fuzzy temporal rules and particle swarm optimization," in *Sadhana*, vol. 39. (Springer), 283–302.

Ganatra, N., and Patel, A. (2020). A multiclass plant leaf disease detection using image processing and machine learning techniques. *Int. J. Emerg. Technol.* 11, 1082–1086.

Han, Z. H., Zhang, Y., Song, C. X., and Zhang, K. S. (2017). Weighted gradient-enhanced kriging for high-dimensional surrogate modelling and design optimization. *Aiaa J.* 55, 4330–4346. doi: 10.2514/1.J055842

Harakannanavara, S. S., Rudagi, J. M., Puranikmath, V. I., Siddiqua, A., and Pramodhini, R. (2022). Plant leaf disease detection using computer vision and machine learning algorithms. *Global Transit. Proc.* 3, 305–310. doi: 10.1016/j.gltp.2022.03.016

Haridasan, A., Thomas, J., and Raj, E. D. (2023). Deep learning system for paddy plant disease detection and classification. *Environ. Monit. Assess.* 195, 120. doi: 10.1007/s10661-022-10656-x

He, J., Liu, T., Li, L., Hu, Y., and Zhou, G. (2023). MFaster r-CNN for maize leaf diseases detection based on machine vision. *Arab. J. Sci. Eng.* 48, 1437–1449. doi: 10.1007/s13369-022-06851-0

Huang, X., Chen, A., Zhou, G., Zhang, X., Wang, J., Peng, N., et al. (2023). Tomato leaf disease detection system based on FC-SNDPN. *Multimed. Tools Appl.* 82, 2121–2144. doi: 10.1007/s11042-021-11790-3

Jabez, C. J., Khanna, N. H., and Arputharaj, K. (2015). "A swarm optimization approach for clinical knowledge mining," in *Computer methods and programs in biomedicine*, vol. 121. (Elsevier), 137–148.

Kaur, P., Harnal, S., Gautam, V., Singh, M. P., and Singh, S. P. (2024). "Performance analysis of segmentation models to detect leaf diseases in tomato plant," in *Multimedia Tools and Applications*, vol. 83. , 16019–16043.

Kaustubh, B. (2020). *Tomato Leaf Disease Dataset* (Kaggle). Available at: https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf?select=tomato.

Khan, M. A., Lali, M. I. U., Sharif, M., Javed, K., Aurangzeb, K., Haider, S. I., et al. (2019). An optimized method for segmentation and classification of apple diseases based on strong correlation and genetic algorithm based feature selection. *IEEE Access* 7, 46261–46277. doi: 10.1109/Access.6287639

Li, X., Li, X., Zhang, S., Zhang, G., Zhang, M., and Shang, H. (2023). SLViT: Shuffle-convolution-based lightweight Vision transformer for effective diagnosis of sugarcane leaf diseases. *J. King Saud University-Comput. Inf. Sci.* 35, 101401. doi: 10.1016/j.jksuci.2022.09.013

Mukhopadhyay, S., Paul, M., Pal, R., and De, D. (2021). Tea leaf disease detection using multi-objective image segmentation. *Multimed. Tools Appl.* 80, 753–771. doi: 10.1007/s11042-020-09567-1

Mustafa, H., Umer, M., Hafeez, U., Hameed, A., Sohaib, A., Ullah, S., et al. (2023). Pepper bell leaf disease detection and classification using optimized convolutional neural network. *Multimed. Tools Appl.* 82, 12065–12080. doi: 10.1007/s11042-022-13737-8

Nahiduzzaman, M., Chowdhury, M. E., Salam, A., Nahid, E., Ahmed, F., Al-Emadi, N., et al. (2023). Explainable deep learning model for automatic mulberry leaf disease classification. *Front. Plant Sci.* 14, 1175515. doi: 10.3389/fpls.2023.1175515

Nerkar, B., and Talbar, S. (2021). Cross-dataset learning for performance improvement of leaf disease detection using reinforced generative adversarial networks. *Int. J. Inf. Technol.* 13, 2305–2312. doi: 10.1007/s41870-021-00772-1

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

Nguyen, T.-N., Dong, T. K., Tri, H. X., and Nguyen, C.-N. (2023). "Deep Learning Approach for Tomato Leaf Disease Detection," in *International Conference on Future Data and Security Engineering* (Springer Nature Singapore, Singapore), 572–579.

Pandey, A., and Jain, K. (2022). Plant leaf disease classification using deep attention residual network optimized by opposition-based symbiotic organisms search algorithm. *Neural Comput. Appl.* 34, 21049–21066. doi: 10.1007/s00521-022-07587-6

Pandiyaraju, V., Ganapathy, S., Mohith, N., and Kannan, A. (2023). An optimal energy utilization model for precision agriculture in WSNs using multi-objective clustering and deep learning. *J. King Saud University-Comput. Inf. Sci.* 35, 101803. doi: 10.1016/j.jksuci.2023.101803

Pandiyaraju, V., Logambigai, R., Ganapathy, S., and Kannan, A. (2020). An energy efficient routing algorithm for WSNs using intelligent fuzzy rules in precision agriculture. *Wireless Pers. Commun.* 112, 243–259. doi: 10.1007/s11277-020-07024-8

Pandiyaraju, V., Shunmuga Perumal, P., Kannan, A., and Sai Ramesh, L. (2017). Smart terrace gardening with intelligent roof control algorithm for water conservation 451–455. doi: 10.21162/PAKJAS

Rakesh, S., and Indiramma, M. (2022). "December. Explainable AI for Crop disease detection," in *2022 4th International Conference on Advances in Computing, Communication Control and Networking, ICAC3N*. 1601–1608.

Ruth, J. A., Uma, R., Meenakshi, A., and Ramkumar, P. (2022). Meta-heuristic based deep learning model for leaf diseases detection. *Neural Process. Lett.* 54, 5693–5709. doi: 10.1007/s11063-022-10880-z

Saeed, A., Abdel-Aziz, A. A., Mossad, A., Abdelhamid, M. A., Alkhaled, A. Y., and Mayhoub, M. (2023). Smart detection of tomato leaf diseases using transfer learning-based convolutional neural networks. *Agriculture* 13, 139. doi: 10.3390/agriculture13010139

Sanida, T., Sideris, A., Sanida, M. V., and Dasygenis, M. (2023). Tomato leaf disease identification via two–stage transfer learning approach. *Smart Agric. Technol.* 5, 100275. doi: 10.1016/j.atech.2023.100275

Sankareshwaran, S. P., Jayaraman, G., Muthukumar, P., and Krishnan, A. (2023). Optimizing rice plant disease detection with crossover boosted artificial hummingbird algorithm based AX-RetinaNet. *Environ. Monit. Assess.* 195, 1070. doi: 10.1007/s10661-023-11612-z

Santhosh, D., Pandiyaraju, V., and Kannan, A. (2014). "Non-naive Bayesian classifier for farmer advisory system," in *2014 Sixth International Conference on Advanced Computing (ICoAC)*. 248–254 (IEEE).

Seetharaman, K., and Mahendran, T. (2022). Leaf disease detection in banana plant using gabor extraction and region-based convolution neural network (RCNN). *J. Instit. Eng. (India): Ser. A* 103, 501–507. doi: 10.1007/s40030-022-00628-2

Shoaib, M., Babar, S., Ihsan, U., Ali, F., and Park, S. H. (2022). Deep learning-based segmentation and classification of leaf images for detection of tomato plant disease. *Front. Plant Sci.* 13, 1031748. doi: 10.3389/fpls.2022.1031748

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, 1158933. doi: 10.3389/fpls.2023.1158933

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

Sreedevi, A., and Manike, C. (2024). A smart solution for tomato leaf disease classification by modified recurrent neural network with severity computation. *Cybernet. Syst.* 55, 409–449. doi: 10.1080/01969722.2022.2122004

Thai, H. T., Le, K. H., and Nguyen, N. L. T. (2023). FormerLeaf: An efficient vision transformer for Cassava Leaf Disease detection. *Comput. Electron. Agric.* 204, 107518. doi: 10.1016/j.compag.2022.107518

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

Uma, K., Geetha, P., and Kannan, A. (2016). A novel segmentation of scanned compound images using fuzzy logic. *J. Med. Imaging Health Inf.* 6, 763–768. doi: 10.1166/jmihi.2016.1754

Vallabhajosyula, S., Sistla, V., and Kolli, V. K. K. (2022). Transfer learning-based deep ensemble neural network for plant leaf disease detection. *J. Plant Dis. Prot.* 129, 545–558. doi: 10.1007/s41348-021-00465-8

Wang, G., Sun, Y., and Wang, J. (2017). Automatic image-based plant disease severity estimation using deep learning. *Comput. Intell. Neurosci.*, 2917536. doi: 10.1155/2017/2917536

Wu, K., Li, Q., Chen, Z., Lin, J., Yi, Y., and Chen, M. (2021). Distributed optimization method with weighted gradients for economic dispatch problem of multi-microgrid systems. *Energy* 222, 119898. doi: 10.1016/j.energy.2021.119898

Yakkundimath, R., and Saunshi, G. (2023). Identification of paddy blast disease field images using multi-layer CNN models. *Environ. Monit. Assess.* 195, 646. doi: 10.1007/s10661-023-11252-3

Yu, S., Xie, L., and Huang, Q. (2023). Inception convolutional vision transformers for plant disease identification. *Internet Things* 21, 100650. doi: 10.1016/j.iot.2022.100650

Zhao, Y., Liu, L., Xie, C., Wang, R., Wang, F., Bu, Y., et al. (2020). An effective automatic system deployed in agricultural Internet of Things using Multi-Context Fusion Network towards crop disease recognition in the wild. *Appl. Soft Comput.* 89, 106128. doi: 10.1016/j.asoc.2020.106128

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 05 June 2024
DOI 10.3389/fpls.2024.1421381

OPEN ACCESS

EDITED BY
Mohsen Yoosefzadeh Najafabadi,
University of Guelph, Canada

REVIEWED BY
Jieli Duan,
South China Agricultural University, China
Yunchao Tang,
Dongguan University of Technology, China

*CORRESPONDENCE
Hailong Zhu
20100009@kust.edu.cn

RECEIVED 22 April 2024
ACCEPTED 16 May 2024
PUBLISHED 05 June 2024

CITATION
Wang F, Tang Y, Gong Z, Jiang J, Chen Y, Xu Q, Hu P and Zhu H (2024) A lightweight Yunnan Xiaomila detection and pose estimation based on improved YOLOv8. *Front. Plant Sci.* 15:1421381. doi: 10.3389/fpls.2024.1421381

A lightweight Yunnan Xiaomila detection and pose estimation based on improved YOLOv8

Fenghua Wang[1], Yuan Tang[1], Zaipeng Gong[1], Jin Jiang[1], Yu Chen[1], Qiang Xu[1], Peng Hu[1] and Hailong Zhu[2]*

[1]Faculty of Modern Agricultural Engineering, Kunming University of Science and Technology, Kunming, Yunnan, China, [2]Engineering Training Center, Kunming University of Science and Technology, Kunming, Yunnan, China

Introduction: Yunnan Xiaomila is a pepper variety whose flowers and fruits become mature at the same time and multiple times a year. The distinction between the fruits and the background is low and the background is complex. The targets are small and difficult to identify.

Methods: This paper aims at the problem of target detection of Yunnan Xiaomila under complex background environment, in order to reduce the impact caused by the small color gradient changes between xiaomila and background and the unclear feature information, an improved PAE-YOLO model is proposed, which combines the EMA attention mechanism and DCNv3 deformable convolution is integrated into the YOLOv8 model, which improves the model's feature extraction capability and inference speed for Xiaomila in complex environments, and achieves a lightweight model. First, the EMA attention mechanism is combined with the C2f module in the YOLOv8 network. The C2f module can well extract local features from the input image, and the EMA attention mechanism can control the global relationship. The two complement each other, thereby enhancing the model's expression ability; Meanwhile, in the backbone network and head network, the DCNv3 convolution module is introduced, which can adaptively adjust the sampling position according to the input feature map, contributing to stronger feature capture capabilities for targets of different scales and a lightweight network. It also uses a depth camera to estimate the posture of Xiaomila, while analyzing and optimizing different occlusion situations. The effectiveness of the proposed method was verified through ablation experiments, model comparison experiments and attitude estimation experiments.

Results: The experimental results indicated that the model obtained an average mean accuracy (mAP) of 88.8%, which was 1.3% higher than that of the original model. Its F1 score reached 83.2, and the GFLOPs and model sizes were 7.6G and 5.7MB respectively. The F1 score ranked the best among several networks, with the model weight and gigabit floating-point operations per second (GFLOPs) being the smallest, which are 6.2% and 8.1% lower than the original model. The loss value was the lowest during training, and the convergence speed was the fastest. Meanwhile, the attitude estimation results of 102 targets showed that the orientation was correctly estimated exceed 85% of the cases, and the average error angle was 15.91°. In the occlusion condition, 86.3% of the attitude

estimation error angles were less than 40°, and the average error angle was 23.19°.

Discussion: The results show that the improved detection model can accurately identify Xiaomila targets fruits, has higher model accuracy, less computational complexity, and can better estimate the target posture.

KEYWORDS

improved YOLOv8, Xiaomila fruits, target detection, lightweight, pose estimation

1 Introduction

Pepper is one of the three major vegetable crops in the world. Its fruit has rich polyphenols, flavonoids, vitamin C, and other natural active ingredients, with high food value, economic value, and health care value (Zhang, 2023). Currently, pepper-picking equipment mainly consists of various forms of harvesters, such as rod and comb harvesters, unfolding double helix harvesters, drum finger harvesters, and strip comb harvesters (Fan et al., 2023). Xiaomila is a smaller, lighter, crispy, and tender variety of pepper, and its flowers and fruits have the same characteristics. Traditional picking equipment is not only prone to damaging Xiaomila fruits but also cannot adapt to the characteristics of Xiaomila flowers and fruits that are contemporaneous.

In recent years, picking robots have gradually become popular (Ye et al., 2023; Wang et al., 2023a; Tang et al., 2024), different from traditional mechanical picking equipment, picking robots have the capability of non-one-time picking and can reduce uncontrollable damage caused by traditional mechanical equipment. This enables the picking robot to adapt well to the characteristics of Xiaomila flowers and fruits that are contemporaneous and easily damaged. The spatial attitude estimation of Xiaomila objects is the to accurate and collision-free picking, and Xiaomila grows in different directions in the natural farmland environment, as illustrated by the arrows in Figure 1.

FIGURE 1
Xiaomila grows in different directions in the natural farmland environment.

Attitude estimation is to infer the three-dimensional translation and rotation information of the target in the camera coordinate system from images or videos (Guo et al., 2023). Traditional attitude estimation methods have low applicability in weak texture target detection and real-time detection, while deep learning methods learn feature information in input images through deep neural networks and have high robustness in real-time applications (Lin et al., 2022a). Therefore, current research on target attitude estimation during picking mainly focuses on deep learning methods.

Methods based on RGB-D images generally collect image data containing target depth information through a depth sensor and extract corresponding features for posture regression. Luo et al. obtained the grape cluster image mask and point cloud information using a depth camera, constructed a region of interest based on the mapping relationship between the two, and utilized the LOWESS algorithm and geometric method to fit the pedicel surface and estimate the posture of the pedicel. This estimation method is highly sensitive to point cloud data (Luo et al., 2022). Eizentals et al. obtained green pepper surface point information through a laser rangefinder and obtained the attitude information of the green pepper fruit in space through model fitting, but the accuracy and success rate were not high (Eizentals and Oka, 2016). Yin et al. obtained the grape mask by using the Mask Region Convolutional Neural Network (Mask R-CNN); meanwhile, they combined the RANSAC algorithm to fit the point cloud into a cylindrical model, estimated the grape posture with its axis, and estimated the posture of each bunch of grapes. The approach took about 1.7s to complete the task (Yin et al., 2021). Zhang et al. proposed a tomato bunch attitude detection method for continuous tomato harvesting operations. The method consists of *a priori* model, cascade network, and three-dimensional reconstruction. It fully exploits the advantages

of convolutional neural networks while avoiding complex point cloud calculations, but it cannot make correct predictions for fruits with heavy occlusion (Zhang et al., 2022). Lin et al. used RGB-D sensors to obtain binary images of guava and branches through a fully convolutional network, adopted Euclidean clustering to separate different groups of point clouds, and used the guava center and nearest branch information for attitude estimation. However, the success rate and accuracy still need to be improved (Lin et al., 2019). Wang et al. designed a geometric perception network that uses point cloud information and RGB images to detect, segment, and grasp targets. It can better perceive targets, but changes in distance have a greater impact on the estimation accuracy (Wang et al., 2022). Li et al. calculated the local plane normal of each point in the point cloud, scored each candidate plane, took the lowest-scoring plane as the symmetry plane of the point cloud, and calculated the symmetry axis based on this plane to realize attitude estimation of bell peppers. However, the estimation effect is not good for occluded bell peppers (Li et al., 2018).

The input data of the method based on RGB images does not contain depth information, and the features of the image are directly extracted for analysis. Sun et al. constructed a multi-task learning model that locates the position of the citrus navel point and predicts the rotation vector of the citrus by performing RGB image analysis of citrus. However, for citrus whose navel point is invisible, the model needs to be further improved (Sun et al., 2023). Zhang et al. used 3D detection results to regress the 2D key point coordinates of objects in the image. By using the perspective n-point algorithm to estimate the pose of an object, this method enhances the accuracy and efficiency of pose estimation (Zhang et al., 2019). Kim et al. developed a deep learning network for determining robot cutting poses during harvesting, which can perform ripeness classification and pose estimation of fruits and lateral stems. The study results indicate that this method performs well in detecting tomatoes in a smart farm environment. However, the detection effect in complex farmland environments has not been verified (Kim et al., 2022). Based on the growth characteristics of grapes, Wu et al. combined human pose estimation, key point detection models, and target detection algorithms to identify grape clusters and estimate poses. However, this method is not effective for complex image processing (Wu et al., 2023a). Lin et al. analyzed a single RGB image based on key points and estimated the pose of the object by regressing the size of the boundary cuboid, but the network was not sufficiently lightweight (Lin et al., 2022b).

To sum up, the method of using RGB-D images or point cloud data to estimate the pose of a target requires a large amount of calculation and is not suitable for transplantation to mobile devices. Additionally, objects to be identified in farmland are basically occluded. The above methods are usually combined with the stems of the identified objects to realize pose estimation. However, the diameter of Xiaomila stems is very small (1–3 mm), and the background is complex. Traditional stereo cameras and depth sensors such as lidar have been proven to be unable to provide reliable depth information (Coll-Ribes et al., 2023). To solve these problems, this study mainly makes the following contributions:

1) We propose a lightweight, multiscale detection model, called PAE-YOLO, for Xiaomila target detection in complex farmland environments. The EMA attention mechanism can effectively enhance the feature extraction capability of the model, while DCNv3 can significantly reduce the computational complexity of the model and improve the portability of the model.

2) We used a depth camera to detect pepper skins and caps to determine the posture of Xiaomi spicy. We also analyze and optimize Xiaomi target detection and posture determination under different occlusion situations.

3) We determined the effectiveness of the improved model through ablation experiments and comparison experiments, and determined the effectiveness of attitude detection through attitude estimation experiments. Among several classic detection models, our proposed model has higher accuracy, the smallest model size, and the lowest computational effort than several classical models.

2 Materials and methods

2.1 Image acquisition

This study takes Xiaomila fruits in the green and mature stage of farmland as the research object. All images used in the experiment were taken in 2023 at a Xiaomila plantation in Qiubei County, Wenshan City, Yunnan Province, China. The Intel realsense D435i device was utilized to collect RGB images. During the image collection process, the camera was placed about 15–30 cm away from the Xiaomila plants and photographed directly above the Xiaomila plants. The image resolution was 1920×1080 pixels, and a total of 1060 images were collected.

2.2 Dataset construction and annotation

In the natural farmland environment, Xiaomila fruits have a similar color to pepper leaves, with small individuals and complex backgrounds. Considering the difference in images obtained under actual changing lighting and occlusion conditions, the original images are collected at different times, under varying lighting, and with diverse occlusion levels. However, these images typically cannot encompass all real-world conditions. Furthermore, they differ somewhat from actual Xiaomila images. Hence, collected RGB images underwent expansion through random rotation, brightness adjustment, and noise addition to harmonize and mitigate these disparities. In the real environment, the pepper's orientation varies, and random rotation and flipping primarily serve to diversify its orientation, enhancing the model's generalization ability. Random clipping accounts for the impact of various occlusion scenarios, ensuring data diversity. Noise addition and brightness adjustment aim to mitigate factors such as brightness deviations among different sensors (Akbar et al., 2022; Bosquet et al., 2023).

Of course, there will still be some differences between the enhanced dataset and the actual changing lighting and occlusion conditions. To minimize such differences, more factors from real scenes can be incorporated when collecting data, such as weather

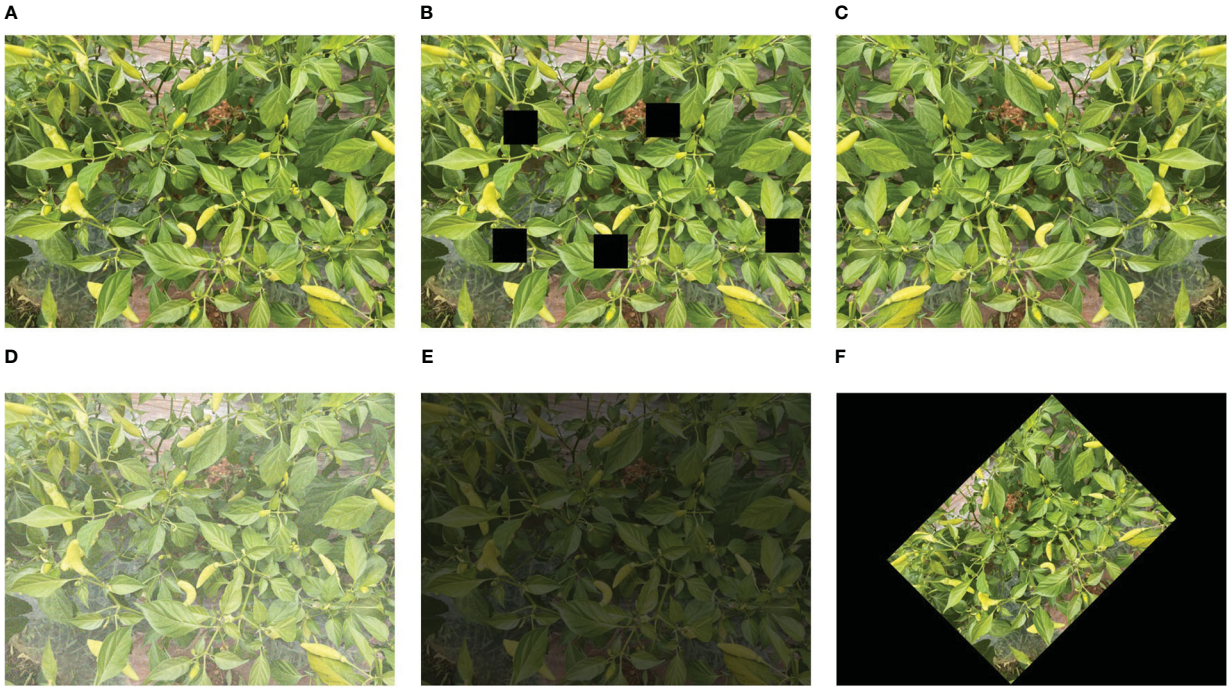


FIGURE 2
Image expansion effect. **(A)** Original image, **(B)** random cropping, **(C)** flipping, **(D)** noise adding, **(E)** brightness adjustment, **(F)** random rotation.

changes, varying occlusion, etc. Additionally, ensuring a similar distribution between training and test data reflects the actual scene more accurately.

The expansion effect is demonstrated in Figure 2. The final target detection data set consists of 2500 images, of which 1750 are used as a training set and the remaining 750 are used as a verification set. The labeling tool was used to label Xiaomi fruits and convert the labeled xml file into the txt file required by the model.

2.3 YOLOv8 network structure

The YOLO series algorithm is an efficient method with limited computational parameters, making it a key research focus in target detection (Wang et al., 2023b). Wu et al. proposed a segmentation and counting algorithm for banana bunches based on YOLOv5-Banana (Wu et al., 2023b). Song et al. introduced the YOLOv7-ECA model, which offers fast detection speed, specifically designed for the similar color and small size of young apple leaves (Song et al., 2023). Yao et al. presented the SCR-YOLO model for detecting the germination rate of wild rice (Yao et al., 2024). Ranjan et al. utilized the YOLOv8 network to detect and adjust green apples in orchards (Sapkota et al., 2024). YOLOv8 is the latest version of the YOLO series network. According to the scaling coefficient, the network is divided into five scales: n/s/m/l/x. The main updates of the YOLOv8 network lie in the C3 module, head network, and loss function. Specifically, the C3 module is replaced by the C2f module, which improves the backbone network's ability to fuse the detailed information and semantic information of feature maps at different scales. The original coupling head is replaced with a decoupling head, and the regression branch and prediction branch are separated, leading to better recognition results. Regarding the loss function, YOLOv8 adopts the task-aligned allocator positive sample distribution strategy to optimize the calculation process of the loss function. Figure 3 shows the overall structure of the YOLOv8 network.

2.4 YOLOv8 model improvement strategy

Though YOLOv8 has strong capabilities in target detection, it still has limitations in the detection of Xiaomila fruits. Compared with other crop fruits, Xiaomila fruits exhibit irregular distribution, there is little change in the color gradient between the fruit area and the background, and it is more susceptible to interference from background information. Considering the above limitations, this study improves YOLOv8 in two aspects: attention mechanism and convolutional neural network.

First, the EMA attention mechanism is combined with the C2f module in the YOLOv8 network. The C2f module can well extract local features from the input image, and the EMA attention mechanism can control the global relationship. The two complement each other, thereby enhancing the model's expression ability; Meanwhile, in the backbone network and head network, the DCNv3 convolution module is introduced, which can adaptively adjust the sampling position according to the input feature map, contributing to stronger feature capture capabilities for targets of different scales and a lightweight network. The test results suggest that the improved model has better performance in

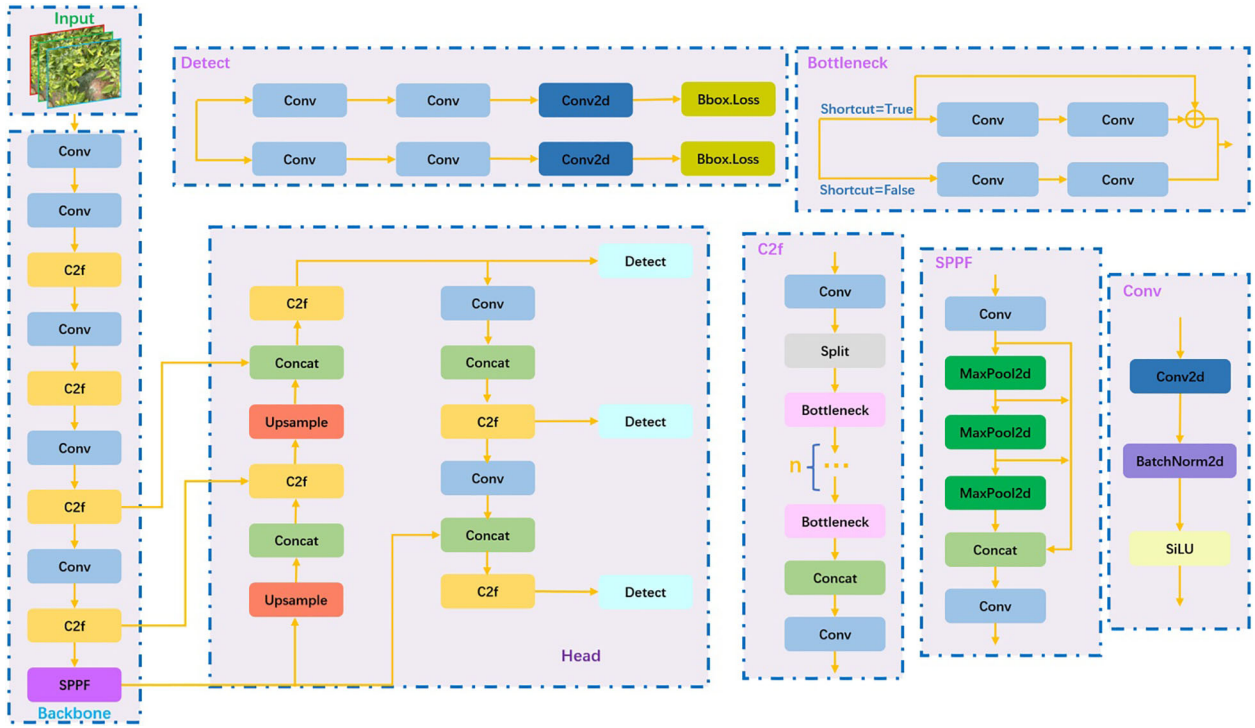


FIGURE 3
YOLOv8 network architecture.

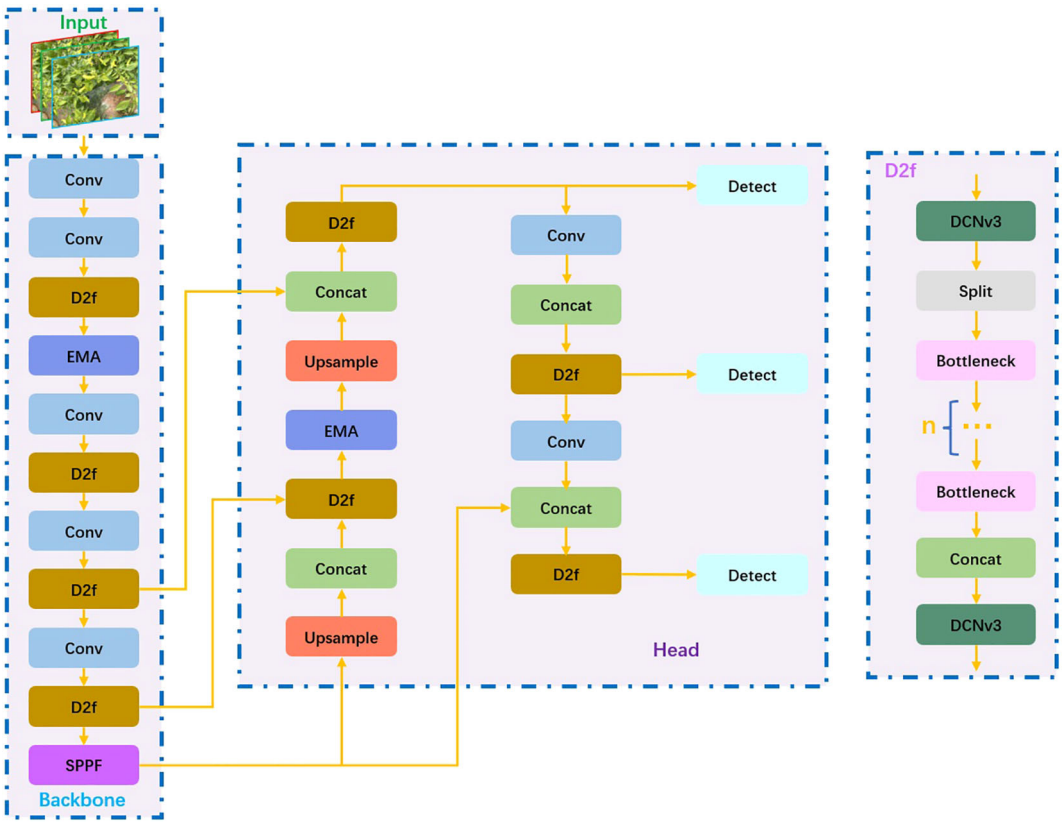


FIGURE 4
The architecture of the PAE-YOLO network.

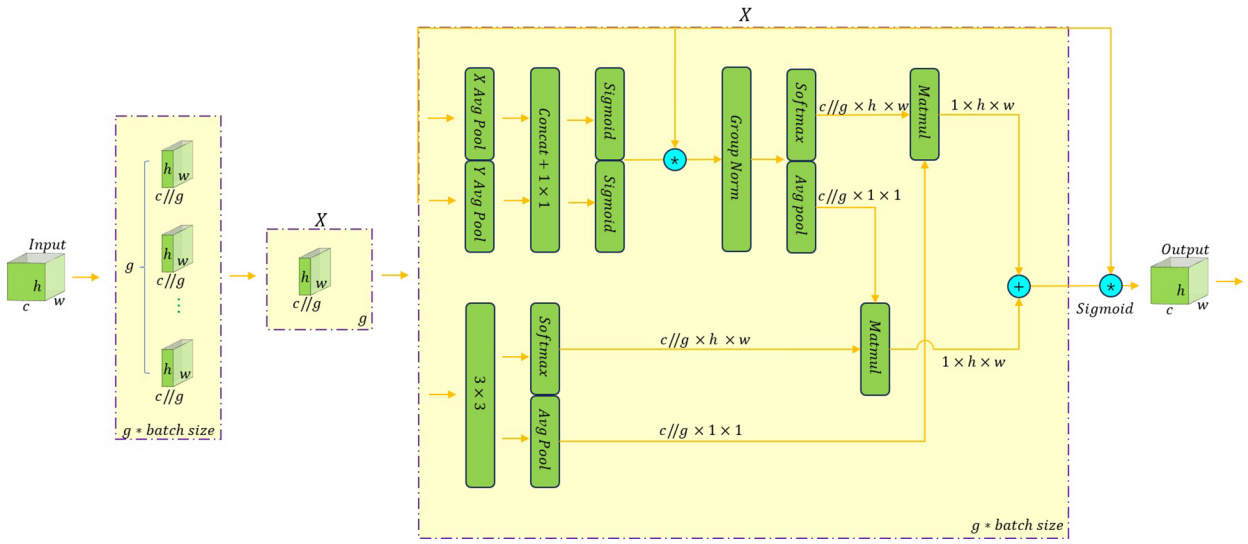


FIGURE 5
The structure of the EMA attention mechanism.

identifying Xiaomila fruits. Since this model is established based on YOLOv8, the improved model is called PAE-YOLO. Figure 4 demonstrates the entire network framework of PAE-YOLO.

2.4.1 EMA attention mechanism

The attention mechanism is employed to help the model distinguish important channels and enhance the feature information in the channels, thereby improving the model's perception and generalization ability of feature information. Traditional attention mechanisms usually produce clear feature information by reducing channel dimensions. However, the reduction of channel dimensions may result in partial information loss and increased errors.

EMA is a multiscale attention mechanism for calculating attention weights (Ouyang et al., 2023). This mechanism introduces the concept of exponential moving average, which divides each channel of the input image into groups containing multiple sub-features. In the process, the EMA attention mechanism only requires one learning accumulation factor, and the number of added parameters is small, which can guarantee that the spatial semantic features are evenly distributed in each feature group without changing the channel dimension. The specific structure of the attention mechanism is shown in Figure 5.

2.4.2 Deformable convolutional network DCNv3

Deformable convolution is a non-fixed sampling convolution network with stronger generalization ability and feature capture ability than ordinary convolution networks. DCNv3 (Wang et al., 2023c) introduces the concept of convolution separation to divide the original convolution weight into two parts: the depth direction and the point direction. The point direction part is taken as the shared projection weight between sampling points to improve the overall efficiency of the model. Meanwhile, DCNv3 divides the process of spatial aggregation into multiple groups with independent sampling offsets and modulation scales. All modulation scalars between sampling points are normalized through softmax, and their sum is constrained to 1, thereby enhancing the training stability of the model. The specific expression is given in Formula 1.

$$y(p_0) = \sum_{f=1}^{F} \sum_{h=1}^{H} w_f m_{fh} X_f (p_0 + p_h + \Delta p_{fh}) \quad (1)$$

where, F denotes the total number of aggregated groups, H represents the number of dimensions, w_f represents the position –independent projection weight of the current group, m_{fh} represents the h sampling points in the f group, X_f denotes a slice of the input feature map, p_0 denotes the current pixel, p_h represents the grid sampling position of the current group, and Δp_{fh} stands for the offset corresponding to p_h.

Figure 6 compares different core operators. (a) shows the global attention operator, which has high computational complexity and memory cost. (b) shows a local attention operator. Although the calculation amount is reduced, it cannot handle long-distance dependencies. (c) shows a large kernel operator, but it cannot adapt to spatial aggregation. (d) shows the dynamic sparse kernel operator used in DCNv3 deformable convolution. It has low computational cost and memory costs, has the capability to handle long-distance dependencies, and can adapt to spatial aggregation.

2.5 Posture estimation for Xiaomila fruits

In the natural farmland environment, affected by leaves, branches, and other fruits, the attitude of Xiaomila fruits has little correlation with the fruit itself. Coupled with complex background factors, it is difficult to directly estimate the posture of Xiaomila fruits. This paper adopts the idea of mapping and uses the detection network to identify all the peppers in the image and takes the single

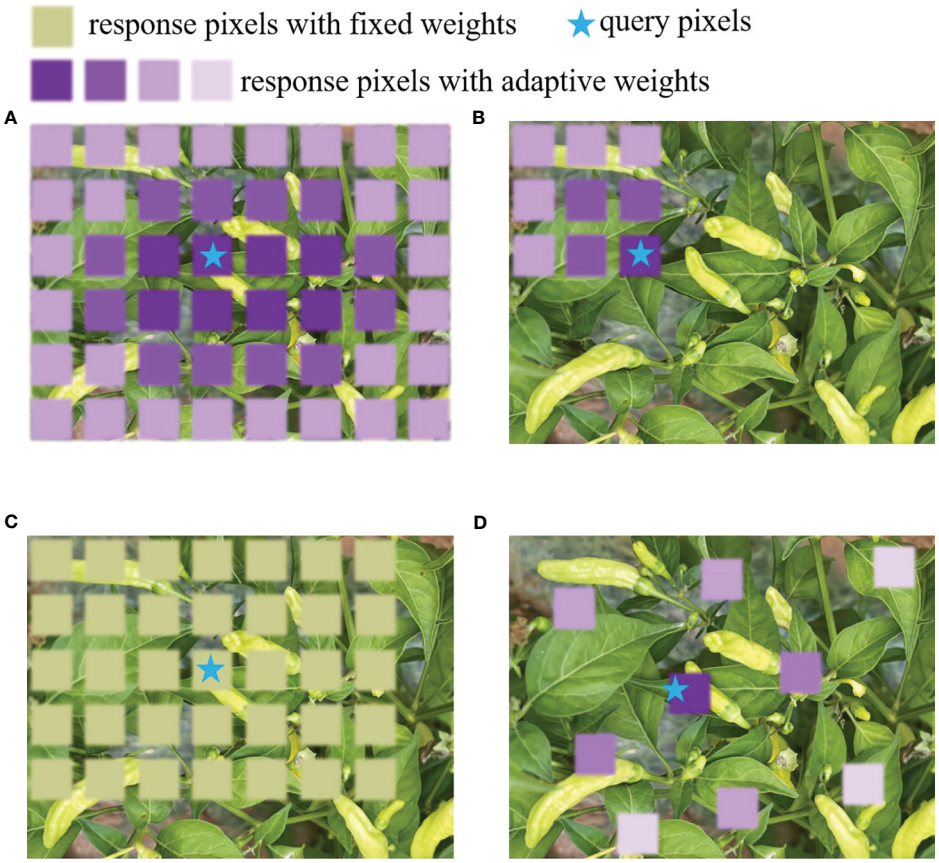


FIGURE 6
The schematic diagram of different core operators. **(A)** global attention operator, **(B)** local attention operator, **(C)** large kernel operator, **(D)** dynamic sparse kernel operator.

pepper image in the recognition frame as the region of interest (RoI). Then, the data of the RoI is passed to the segmentation network, which segments the area target and outputs a binary mask. Next, based on the pixel information of the segmented individual Xiaomila fruits, two-dimensional pose estimation is performed on the Xiaomila fruits, and the pose estimation effect is mapped back to the original image. Finally, combined with the depth information, the spatial posture of Xiaomila fruits is obtained.

2.5.1 Xiaomila 2D fitting

Xiaomila fruits are very light. Unlike heavier crops such as grapefruit and apples, the fruit stems are generally facing downward

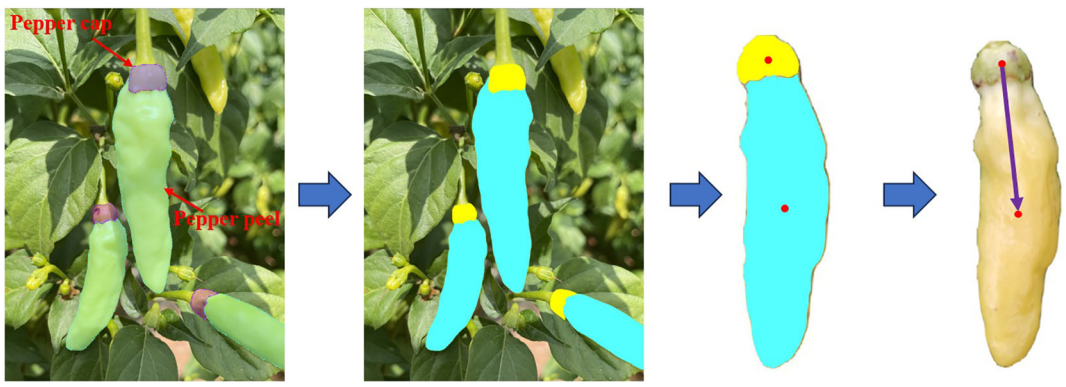


FIGURE 7
Posture estimation of unoccluded Xiaomila fruits.

(Kang et al., 2020; Zeng et al., 2021). Meanwhile, the fruit stems of Xiaomila are very thin and subject to greater interference. These factors make it difficult to directly identify and fit the fruit stems like tomatoes, grapes, lychees, etc (Zhong et al., 2021; Li et al., 2023; Zhang et al., 2023).

There is an obvious gradient change in the color of the pepper peel and the color of the pepper cap. Based on this characteristic, this paper segments the pepper peel and the pepper cap respectively, calculates the moments of the masks of these two parts, and then takes the two-dimensional vector composed of these two moment points as the two-dimensional image posture of Xiaomila fruits, as shown in Figure 7.

In the farmland environment, part of the pepper caps are blocked, and the moment points of the pepper caps cannot be successfully obtained. Considering that the Xiaomila fruit is strip-shaped, this paper employs the least squares method (de Jong, 1993) to optimally fit the mask data of the Xiaomila fruit. The relevant parameters and definitions of Xiaomila fruit fitting are given in Formulas 2–4:

$$y = \hat{k}x + \hat{b} \tag{2}$$

$$\hat{k} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3}$$

$$\hat{b} = \bar{y} - \hat{k}\bar{x} \tag{4}$$

where, x_i is the x-direction coordinate of the mask outline pixel in the Xiaomila image coordinate system, y_i is the y-direction coordinate of the mask outline pixel, n denotes the number of mask outline pixel points, $\bar{x}$ is the x coordinate of all outline pixels. $\bar{y}$ represents the mean of all y-coordinates of the contour pixel. $\hat{k}$ denotes the slope of the mask profile fitting straight line, and $\hat{b}$ is the intercept of the straight line.

The final fitting effect is illustrated in Figure 8. Specifically, (a) shows the original Xiaomila image; (b) shows the mask image of Xiaomila; (c) shows the extracted mask contour binary image; (d) shows a schematic diagram of contour fitting; (e) shows a fitting effect diagram, where the green line represents the Xiaomila contour line, the blue line AB represents the fitting straight line, and the red dot indicates the estimated tip of Xiaomila; (f) shows the posture effect.

Finally, by comparing the sum of the Euclidean distances between the two end points of the contour and other points on the contour to determine which end is the tip, two-dimensional pose estimation of Xiaomila fruits with the pepper cap occluded is realized.

2.5.2 Estimating space posture for Xiaomila fruits

The Xiaomila fruit fitting line is obtained based on a two-dimensional image, and its description method is based on the image pixel coordinate system. To obtain its posture in real space, the points in the pixel coordinate system need to be converted to the world coordinate system. The pixel coordinate system ($o - uv$) takes the upper left corner of the image as the origin of the coordinate system, and the unit is pixel; meanwhile, the image coordinate system ($o - xy$) takes the center point of the image as the origin of the coordinate system, and the unit is millimeter (mm); additionally, the camera coordinate system ($o_c - x_c y_c z_c$) takes the optical center of the depth camera as the origin, and the unit is meter (m); moreover, the world coordinate system coincides with the camera coordinate system, as shown in Figure 9.

Before performing coordinate conversion, the Matlab-Camera Calibrator toolbox is utilized to calibrate the depth camera to obtain

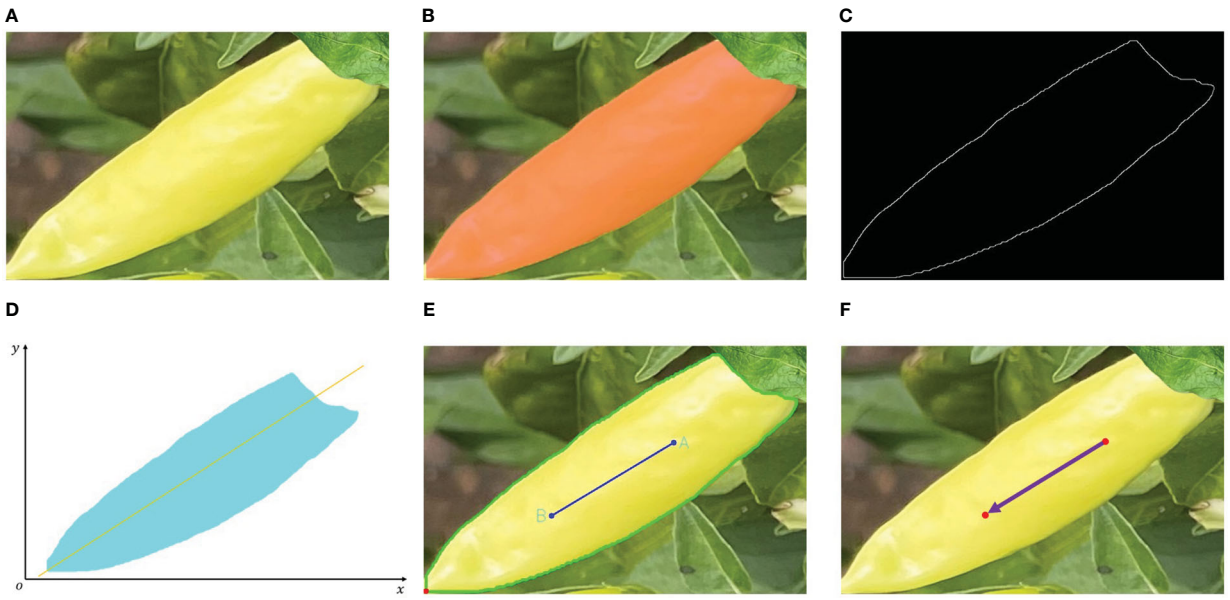


FIGURE 8
Posture fitting of occluded pepper caps. **(A)** original Xiaomila image, **(B)** mask image of Xiaomila, **(C)** extracted mask contour binary image, **(D)** schematic diagram of contour fitting, **(E)** fitting effect, **(F)** posture effect.

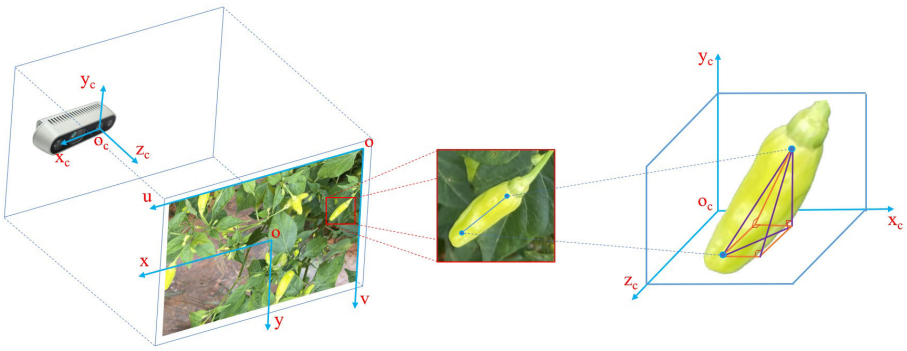


FIGURE 9
Schematic diagram of the coordinate systems.

the camera's internal parameter matrix and external parameter matrix. Then, the spatial point coordinates corresponding to the pixel point coordinates are calculated through Equation 5.

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = KP \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{5}$$

where, z_c represents the axial distance of the camera in the Z-axis, (u, v) is the pixel coordinate, K is the camera internal parameter matrix, P is the camera external parameter matrix, and (X_w, Y_w, Z_w) is the point coordinate corresponding to the world coordinate system.

After the depth camera coordinate system is determined, a 3×1 translation matrix can be used to locate any point in the camera coordinate system. The conversion between the camera coordinate system and the Xiaomila coordinate system is represented by a 3×3 rotation matrix. Then, the position and attitude of the Xiaomila fruit can be determined by combining the translation matrix and rotation matrix. In this approach, the spatial position and spatial vector of the Xiaomila fruit are now known. Through inverse solution, the translation matrix and rotation matrix are obtained, thereby obtaining the rotation angle and translation distance of each joint. Finally, based on the rotation angle and translation information, the end effector is controlled to reach the designated position to complete the picking task. The translation matrix and rotation matrix are shown in Equations 6 and 7.

$${}^{S}\boldsymbol{P} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{6}$$

$${}^{S}_{L}\boldsymbol{R} = \left({}^{S}\hat{\boldsymbol{X}}_L \quad {}^{S}\hat{\boldsymbol{Y}}_L \quad {}^{S}\hat{\boldsymbol{Z}}_L\right) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{7}$$

where, ${}^{S}P$ is the translation matrix, ${}^{S}_{L}R$ is the rotation matrix, S represents the depth camera coordinate system, and L represents the Xiaomila coordinate system; p_x, p_y, and p_z are the center of gravity of the Xiaomila fruit relative to the camera, respectively. ${}^{S}\hat{X}_L$, ${}^{S}\hat{Y}_L$, and ${}^{S}\hat{Z}_L$ respectively represent the distance information of the Xiaomila coordinate system relative to the camera coordinate system along the x, y, and z axes.

2.6 Evaluation metrics

2.6.1 Evaluation of detection and segmentation

This paper takes detection precision (P), mean average precision (mAP), recall rate (R), F1 score, gigabit floating point operations per second (GFlops), and model weight file size as evaluation indicators. Precision is the ratio of the actual number of positives to the number of predicted positives. The higher the precision, the lower the false detection rate. The mean average precision is the mean of the average accuracy across all categories, and it is used to evaluate the accuracy of the entire model. Recall rate is used to evaluate the missed detection rate of the model. The F1 score measures the impact of precision and recall and is used to evaluate the stability of the model. GFlops represent the number of

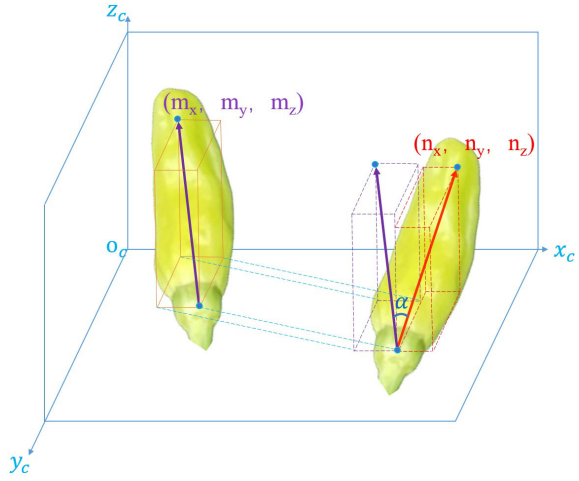


FIGURE 10
Diagram of error angle.

floating-point operations performed per second, and it is used to evaluate the computing performance of the model. The calculation formulas for these evaluation indicators are shown in Equations 8–11.

$$P = \frac{TP}{TP + FP} \tag{8}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{9}$$

$$R = \frac{TP}{TP + FN} \tag{10}$$

$$F = \frac{2 \times P \times R}{P + R} \tag{11}$$

where, TP represents the number of true positive values; FP represents the number of false positive values; FN represents the number of false negative values; n represents the number of categories of identified objects, and AP_i represents the average accuracy for category i.

2.6.2 Evaluation of pose estimation

The error angle α is the angle between the actual space vector and the predicted space vector of the Xiaomila fruit. It is used to represent the error of the posture prediction algorithm, as shown in Figure 10. The calculation formula of α is shown in Equation 12:

$$\alpha = arccos\frac{n_x m_x + n_y m_y + n_z m_z}{\sqrt{n_x^2 + n_y^2 + n_z^2} \times \sqrt{m_x^2 + m_y^2 + m_z^2}} \tag{12}$$

where, $m = (n_x, n_y, n_z)$ is the spatial vector of the Xiaomila fruit predicted by the attitude estimation algorithm, and $m = (m_x, m_y, m_z)$ is the actual spatial vector of the Xiaomila fruit. The smaller the error angle α, the closer the predicted posture is to the real situation.

2.7 Software

The hardware platform used for the experiment is a computer equipped with Intel Xeon W-2145 (16GB memory) and NVIDIA GeForce RTX2080Ti (11 GB video memory) and running 64-bit Windows 11 operating system. The Xiaomila target detection and segmentation model is trained using CUDA 11.6, Opencv, Pytorch framework, Python3.9 programming language, etc.

3 Results and discuss

3.1 Analysis of detection and segmentation

3.1.1 Ablation experiment

To evaluate the impacts of the EMA attention mechanism and the DCNv3 convolution module on improving the detection performance of Xiaomila fruits, these two structures were introduced into the official YOLOv8 respectively. Table 1 presents the impact of each module on the overall detection effect of the algorithm. The model performance was evaluated in terms of precision, recall, average precision, F1 score, floating point operations (FLOPs), and model weight size.

As shown in Table 1, several improvement strategies are effective in improving the model's detection effect. Compared with the original YOLOv8n model, the recall rate and average precision of the model with the attention mechanism were increased by 0.7% and 1.4%, respectively. Meanwhile, the model

TABLE 1 Ablation experiments of different modules of PAE-YOLO.

Model	EMA	DCNv3	P (%)	R (%)	mAP (%)	F1 Score (%)	GFLOPs	Model Size/MB
YOLOv8n	×	×	86.5	78.8	87.5	82.5	8.1	6.2
	√	×	87.1	79.5	88.9	83.1	8.4	6.3
	×	√	87.3	78.1	87.6	82.4	7.4	5.7
	√	√	87.2	79.5	88.8	83.2	7.6	5.7

TABLE 2 Recognition results of different models on Xiaomila images.

Model	P (%)	R (%)	mAP (%)	F1 Score (%)	GFLOPs	Model Size/MB
Mobilenetv3	85.0	76.7	85.4	80.6	11.2	10.5
YOLOv5s	88.8	75.3	85.0	81.5	15.8	14.4
YOLOv7-tiny	85.7	82.8	89.5	84.2	13.0	12.3
YOLOv8n	86.5	78.8	87.5	82.5	8.1	6.2
PAE-YOLO	87.2	79.5	88.8	83.2	7.6	5.7

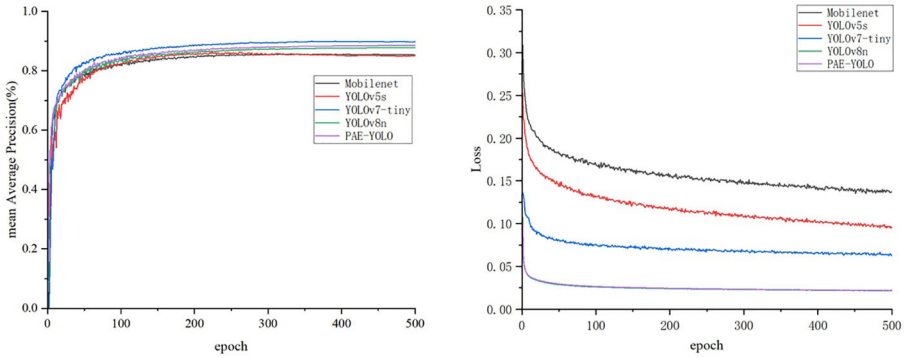


FIGURE 11
mAP and loss curves.

weight was slightly increased, and the FLOPs reached 8.4G. After the convolution in the c2f module of the original model was replaced, the recall rate and average precision of the model were slightly improved compared to the original model, the model weight decreased by 8.1%, and the FLOPs dropped to 7.4G. Compared with the original YOLOv8n model, the average precision of the final PAE-YOLO model increased by 1.3%, the recall rate increased by 0.7%, GFLOPs decreased by 6.2%, the model size decreased by 8.1%, and the F1 score reached 83.2%.The results suggest that the EMA attention mechanism can improve the feature extraction capability of the model while adding a small number of parameters, and the DCNv3 convolution module enhances the portability and real-time detection performance of the model.

By combining the EMA attention mechanism and the DCNv3 deformable convolution network, PAE-YOLO not only improved the detection performance of Xiaomila fruits but also reduced the model's calculation amount from 8.4G to 7.6G, and the model weight size dropped from 6.3M to 5.7M. Compared with the original YOLOv8n model, the FLOPs of PAE-YOLO were reduced by 6.2%, the model weight was reduced by 8.1%, the precision reached 87.2%, the recall rate reached 79.5%, the average precision reached 88.8%, and the F1 score was 83.2. Therefore, our method

FIGURE 12
PAE-YOLO detection and segmentation results. **(A)** detection results of the xiaomila object, **(B)** segmentation results of the pickable xiaomila object.

TABLE 3 Error angle analysis.

Limit angle	Frequency	Average error	Standard deviation
Unlimited	1	18.63	13.89
<30°	0.844	13.75	4.94
<20°	0.711	11.98	2.91
<15°	0.556	10.63	1.44

improves the algorithm performance in various indicators and reduces the algorithm's computational complexity, which helps integrate the algorithm into picking robots for real-time applications.

3.1.2 Comparative experiment

To verify the advantages of the model proposed in this paper in detecting Xiaomila targets, this study selected five classic detection models based on deep learning for performance comparison. Table 2 shows the experimental results of Mobilenetv3, YOLOv5s, YOLOv7-tiny, YOLOv8n, and PAE-YOLO.

As illustrated in Table 2 and Figure 11, compared with Mobilenetv3 and YOLOv5s networks, the recall rate of the PAE-YOLO model increased by 2.8% and 4.2% respectively, the mAP value increased by 3.4% and 3.8% respectively, and the model weight decreased by 45.7% and 60.4% respectively. Compared with the YOLOv7-tiny model, although the PAE-YOLO model had a slight decrease in precision and recall, the GFLOPs and weight decreased by 41.5% and 53.7%, respectively. The F1 score of PAE-YOLO ranked the best among the above-mentioned series of networks, with the smallest model weight and GFLOPs. Additionally, the PAE-YOLO model exhibited the lowest loss value and the fastest convergence speed during the training process.

These test results suggest that the PAE-YOLO network has a stronger overall performance in visual recognition of Xiaomila fruits. Figure 12 shows the detection and segmentation results of the PAE-YOLO model. Specifically, (a) shows the detection results of the xiaomila object, in which the xiaomila with purple contour is the pickable object; (b) shows the segmentation results of the pickable xiaomila object.

3.2 Analysis of pose estimation effects

3.2.1 Error angle analysis

In the actual farmland picking environment, if the error angle of Xiaomila's attitude estimation falls within a certain range, the end effector of the picking equipment can achieve accurate picking. This study analyzes the error angles at different angles, as listed in Table 3.

An example of the spatial pose estimated by the proposed pose estimation method is demonstrated in Figure 13. In this figure, the burgundy arrow represents the actual posture of the manually annotated pepper, the dark purple arrow represents the preliminary posture of the pepper estimated by the algorithm based on the surface points of the pepper, and the blue arrow represents the optimized posture of the pepper.

Figure 13A shows the spatial pose estimation of a Xiaomila fruit without bending, while Figure 13B shows the spatial attitude estimation of a Xiaomila fruit in a curved state. The posture of the Xiaomila fruit with a small curvature estimated based on surface points is basically consistent with the actual situation, while the estimation of the Xiaomila fruit with a large curvature based on surface points produces an error. This error may be ignored in complex farmland environments, resulting in an inability to correctly estimate the posture. This paper uses the two-

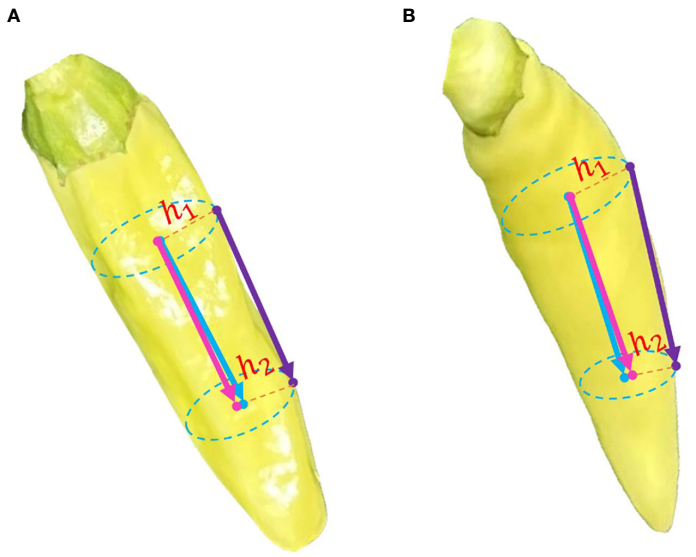


FIGURE 13
Example of spatial pose estimation of Xiaomila fruits. **(A)** spatial pose estimation of a Xiaomila fruit without bending, **(B)** spatial attitude estimation of a Xiaomila fruit in a curved state.

10.3389/fpls.2024.1421381

FIGURE 14
Classification of Xiaomila fruits occlusion. (1) Xiaomila pose estimation with the pepper cap not occluded, (2) (3) (4) Xiaomila pose estimation with the pepper cap occluded.

TABLE 4 Pose estimation error under different occlusion situations.

Occlusion situation	Frequency	Average error	Standard deviation
a	0.667	15.68	5.87
b	0.196	16.69	5.63
c	0.059	160.97	6.37
d	0.078	122.31	55.11

dimensional Xiaomila fitting straight line as the symmetry axis to calculate the radial pixels at both end points of the estimated posture and then determines the inward offset distances h1 and h2 through the depth camera, thereby performing spatial analysis on the estimated posture.

3.2.2 Analysis of different occlusion situations

This paper discusses the pose estimation results under four different occlusion situations: the pepper cap is not occluded (a), the pepper cap is occluded but the occlusion does not produce a tip on the Xiaomila fruit (b), the pepper cap is occluded and the occlusion produces a tip on the Xiaomila fruit (c), and the pepper cap and tip are both occluded (d). In Figure 14, (1) shows the Xiaomila pose estimation with the pepper cap not occluded. (2)(3)(4) show the Xiaomila pose estimation with the pepper cap occluded. (3)(4) did not correctly determine the direction of the Xiaomila fruit. This is because (i) The pepper cap is occluded, and the tip angle formed by the occluded on the pepper cap part is smaller than the pepper tip angle. The attitude estimation algorithm makes an error when judging the orientation of the Xiaomila fruit. (ii) Both the pepper tip and pepper cap are occluded, and the algorithm cannot correctly identify and predict the specific orientation of the Xiaomila fruit.

The attitude estimation error results under four different occlusion situations are presented in Table 4. The attitude estimation error when the pepper cap is not occluded is smaller than the attitude estimation error when the pepper cap is occluded. The average error angle is 23.19°. When the occluded cap is occluded, the algorithm fails to correctly identify the specific orientation of the pepper, thus affecting the attitude estimation effect. Since there are fewer situations (c) and (d) in practice, these two occlusion situations have less impact on the overall pose

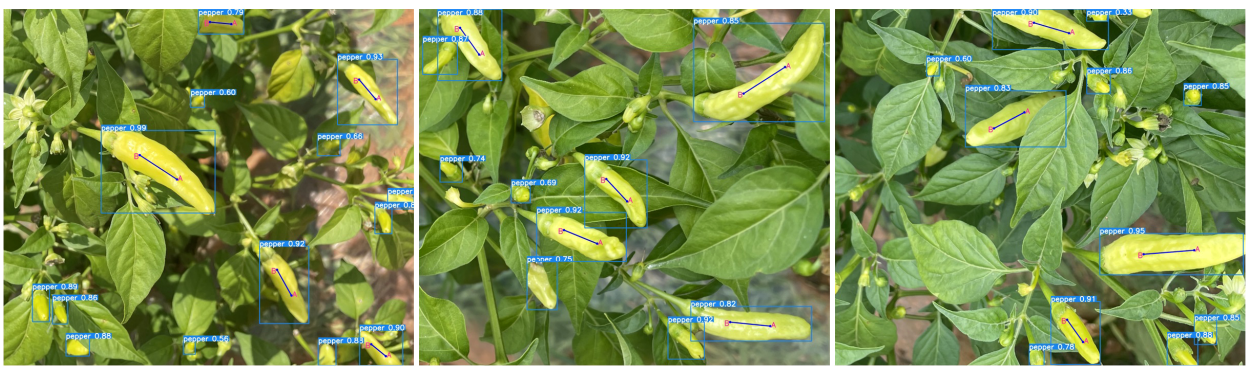


FIGURE 15
Attitude estimation renderings.

estimation effect. The final attitude estimation effect is shown in Figure 15, where end A represents the pepper tip, and end B represents the pepper cap.

4 Conclusion and future work

To solve the problems due to complex background, similar fruit color and background color, and different growth directions in the natural farmland environment, this paper constructed a Xiaomila target recognition data set, proposed an improved Xiaomila target detection model, and the spatial posture and occlusion of Xiaomila were analyzed. Specifically, the existing YOLOv8 target detection algorithm has been improved. The addition of the EMA attention mechanism can better capture the characteristic information of targets of different scales, and the deformable convolution module makes the model more lightweight. At the same time, the spatial position information of the pepper was exploited to describe the translation part of Xiaomila's posture, and the transformation information of the fitted Xiaomila spatial vector relative to the depth camera coordinate system was utilized to describe the rotation part of Xiaomila's posture. The advantage of this work is that no complex annotation model and calculations is required to obtain the expected estimation results, and can be better transplanted to embedded devices. In experiments, the mAP value of the improved PAE-YOLO model reached 88.8%, which was 1.3% higher than the original model. The model weight and GFLOPs were 7.6G and 5.7MB respectively, which are 6.2% and 8.1% lower than the original model, the loss value was the lowest during training, and the convergence speed was the fastest. Finally, an experimental analysis was conducted on Xiaomila's posture and occlusion conditions. More than 85% of the cases where Xiaomila's orientation was correctly estimated, with an average error angle of 15.91°. Under occlusion situations, 86.3% of the attitude estimation error angles were less than 40°, and the average error angle was 23.19°. Therefore, the improved detection model can accurately identify Xiaomila targets in complex environments, and can better estimate the target posture, and is suitable for visual picking of Xiaomila fruits.

However, current detection models still have some limitations. Some severely occluded Xiaomila targets cannot be correctly identified and estimated. For example, the pepper cap and the pepper peel are covered at the same time or the pepper cap is covered and the covering splits the pepper in two. Meanwhile, it remains to be seen whether the target recognition algorithm and attitude estimation method proposed in this article are applicable to other fruits. In future work, we will integrate the improved model into the robot motion control system to realize the automatic harvesting of Xiaomila in natural farmland environments.

Data availability statement

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

Author contributions

FW: Writing – review & editing. YT: Writing – original draft, Software, Validation, Writing – review & editing. ZG: Methodology, Writing – original draft. JJ: Data curation, Writing – original draft. YC: Resources, Writing – original draft. QX: Conceptualization, Writing – original draft. PH: Investigation, Writing – original draft. HZ: 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 supported by the National Natural Science Foundation of China (51975265) and the Scientific Research Fund of Education Department of Yunnan Province (2023J0136, 2024Y134) and the Yunnan Provincial Innovation Training Project (S202310674109).

Acknowledgments

The authors would like to thank all the reviewers who participated in the review.

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

References

Akbar, M., Ullah, M., Shah, B. B., Khan, R. U., Hussain, T., Ali, F., et al. (2022). An effective deep learning approach for the classification of Bacteriosis in peach leave. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.1064854

Bosquet, B., Cores, D., Seidenari, L., Brea, V. M., Mucientes, M., and Del Bimbo, A. (2023). A full data augmentation pipeline for small object detection based on generative adversarial networks. *Pattern Recognit.* 133. doi: 10.1016/j.patcog.2022.108998

Coll-Ribes, G., Torres-Rodríguez, I. J., Grau, A., Guerra, E., and Sanfeliu, A. (2023). Accurate detection and depth estimation of table grapes and peduncles for robot harvesting, combining monocular depth estimation and CNN methods. *Comput. Electron. Agric.* 215. doi: 10.1016/j.compag.2023.108362

de Jong, S. (1993). SIMPLS: An alternative approach to partial least squares regression. *Chemom. Intell. Lab. Sys.* 18, 251–263. doi: 10.1016/0169-7439(93)85002-X

Eizentals, P., and Oka, K. (2016). 3D pose estimation of green pepper fruit for automated harvesting. *Comput. Electron. Agricult.* 128, 127–140. doi: 10.1016/j.compag.2016.08.024

Fan, S., Sun, R., and Lou, H. (2023). Technical status and research strategies of pepper harvest mechanization in my country. *J. Zhongzhou Univ.* 40, 116–120. doi: 10.13783/j.cnki.cn41–1275/g4.2023.02.021

Guo, N., Li, J., and Ren, X. (2023). Survey of rigid object pose estimation algorithms based on deep learning. *Comput. Sci.* 50, 178–189. doi: 10.11896/jsjkx.211200164

Kang, H. W., Zhou, H. Y., Wang, X., and Chen, C. (2020). Real-time fruit recognition and grasping estimation for robotic apple har-vesting. *Sensors* 20. doi: 10.3390/s20195670

Kim, J., Pyo, H., Jang, I., Kang, J., Ju, B., and Ko, K. E. (2022). Tomato harvesting robotic system based on Deep-ToMaToS: Deep learning network using transformation loss for 6D pose estimation of maturity classified tomatoes with side-stem. *Comput. Electron. Agric.* 201. doi: 10.1016/j.compag.2022.107300

Li, Y. J., Feng, Q. C., Liu, C., Xiong, Z. C., Sun, Y. H., Xie, F., et al. (2023). MTA-YOLACT: Multitask-aware network on fruit bunch identification for cherry tomato robotic harvesting. *Eur. J. Agron.* 146. doi: 10.1016/j.eja.2023.126812

Li, H., Zhu, Q. B., Huang, M., Guo, Y., and Qin, J. W. (2018). Pose estimation of sweet pepper through symmetry axis detection. *Sensors* 18. doi: 10.3390/s18093083

Lin, G. C., Tang, Y. C., Zou, X. J., Xiong, J. T., and Li, J. H. (2019). Guava detection and pose estimation using a low-cost RGB-D sensor in the field. *Sensors* 19. doi: 10.3390/s19020428

Lin, Y. Z., Tremblay, J., Tyree, S., Vela, P. A., and Birchfield, S. (2022b). "Single-stage keypoint-based category-level object pose estimation from an RGB image," in *IEEE International Conference on Robotics and Automation (ICRA)*, Philadelphia, May 23–27. PA2022. doi: 10.48550/arXiv.2109.06161

Lin, L., Wang, Y., and Sun, H. (2022a). Target 6D attitude 000estimation algorithm based on improved heat map loss function. *Liquid Crystal Display* 37, 913–923. doi: 10.37188/CJLCD.2021-0317

Luo, L. F., Yin, W., Ning, Z. T., Wang, J. H., Wei, H. L., Chen, W. L., et al. (2022). In-field pose estimation of grape clusters with combined point cloud segmentation and geometric analysis. *Comput. Electron. Agric.* 200. doi: 10.1016/j.compag.2022.107197

Ouyang, D., He, S., Zhang, G., Luo, M., Guo, H., and Zhan, J. (2023). "Efficient multi-scale attention module with cross-spatial learning," in *ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 1–5 (IEEE).

Sapkota, R., Ahmed, D., Churuvija, M., and Karkee, M. (2024). Immature green apple detection and sizing in commercial orchards using YOLOv8 and shape fitting techniques. *IEEE Access.* 12, 43436–43452. doi: 10.1109/ACCESS.2024.3378261

Song, H., Ma, B., Shang, Y., Wen, Y., and Zhang, S. (2023). Detection of young apple fruits based on YOLO v7-ECA model. *Trans. Chin. Soc. Agric. Machine.* 54, 233–242. doi: 10.6041/j.issn.1000-1298.2023.06.024

Sun, Q. X., Zhong, M., Chai, X. J., Zeng, Z. K., Yin, H. S., Zhou, G. M., et al. (2023). Citrus pose estimation from an RGB image for automated harvesting. *Comput. Electron. Agric.* 211. doi: 10.1016/j.compag.2023.108022

Tang, Y., Qi, S., Zhu, L., Zhuo, X., Zhang, Y., and Meng, F. (2024). Obstacle avoidance motion in mobile robotics. *J. Sys. Simul.* 36, 1. doi: 10.16182/j.issn1004731x.joss.23-1297E

Wang, L., Bai, J., Li, W., and Jiang, J. (2023b). Research progress of YOLO series target detection algorithms. *Comput. Eng. Appl. (China)*, 15–29. doi: 10.3778/j.issn.1002-8331.2301-0081

Wang, W., Dai, J., Chen, Z., Huang, Z., Li, Z., and Zhu, X. (2023c). "Internimage: Exploring large-scale vision foundation models with deformable convolutions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14408–14419. doi: 10.48550/arXiv.2211.05778

Wang, X., Kang, H. W., Zhou, H. Y., Au, W., and Chen, C. (2022). Geometry-aware fruit grasping estimation for robotic harvesting in apple orchards. *Comput. Electron. Agric.* 193. doi: 10.1016/j.compag.2022.106716

Wang, C., Li, C., Han, Q., Wu, F., and Zou, X. (2023a). 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

Wu, Z. W., Xia, F., Zhou, S. Y., and Xu, D. Y. (2023a). A method for identifying grape stems using keypoints. *Comput. Electron. Agric.* 209. doi: 10.1016/j.compag.2023.107825

Wu, F., Yang, Z., Mo, X., Wu, Z., Tang, W., Duan, J., et al. (2023b). 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

Yao, Q., Zheng, X. M., Zhou, G. M., and Zhang, J. H. (2024). SGR-YOLO: a method for detecting seed germination rate in wild rice. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1305081

Ye, L., Wu, F., Zou, X., and Li, J. (2023). Path planning for mobile robots in unstructured orchard environments: An improved kinematically constrained bi-directional RRT approach. *Comput. Electron. Agric.* 215, 108453. doi: 10.1016/j.compag.2023.108453

Yin, W., Wen, H., Ning, Z., Ye, J., Dong, Z., and Luo, L. (2021). Fruit detection and pose estimation for grape cluster-harvesting robot using binocular imagery based on deep neural networks. *Front. robot. AI.* 8, 626989. doi: 10.3389/frobt.2021.626989

Zeng, J., Hong, T., and Yang, Z. (2021). Research on pomelo pose recognition and location based on instance segmentation. *J. Henan Agric. Univers.* 55, 287–294. doi: 10.16445/j.cnki.1000-2340.20210326.001

Zhang, Z. (2023). Development status, main challenges and countermeasures of my country's pepper industry. *North. Horticult.* 14), 153–158.

Zhang, F., Gao, J., Song, C. Y., Zhou, H., Zou, K. L., Xie, J. Y., et al. (2023). TPMv2: An end-to-end tomato pose method based on 3D key points detection. *Comput. Electron. Agric.* 210. doi: 10.1016/j.compag.2023.107878

Zhang, F., Gao, J., Zhou, H., Zhang, J. X., Zou, K. L., and Yuan, T. (2022). Three-dimensional pose detection method based on keypoints detection network for tomato bunch. *Comput. Electron. Agric.* 195. doi: 10.1016/j.compag.2022.106824

Zhang, X., Jiang, Z. G., and Zhang, H. P. (2019). Real-time 6D pose estimation from a single RGB image. *Image Vision Comput.* 89, 1–11. doi: 10.1016/j.imavis.2019.06.013

Zhong, Z., Xiong, J. T., Zheng, Z. H., Liu, B. L., Liao, S. S., Huo, Z. W., et al. (2021). A method for litchi picking points calculation in natural environment based on main fruit bearing branch detection. *Comput. Electron. Agric.*, 189. doi: 10.1016/j.compag.2021.106398

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 20 June 2024
DOI 10.3389/fpls.2024.1409194

OPEN ACCESS

EDITED BY
Mohsen Yoosefzadeh Najafabadi,
University of Guelph, Canada

REVIEWED BY
Jinling Zhao,
Anhui University, China
Pappu Kumar Yadav,
South Dakota State University, United States

*CORRESPONDENCE
Pan Gao
gp_inf@shzu.edu.cn
Li Guo
gl_inf@shzu.edu.cn

†These authors have contributed equally to this work

RECEIVED 29 March 2024
ACCEPTED 31 May 2024
PUBLISHED 20 June 2024

CITATION
Zhang M, Chen W, Gao P, Li Y, Tan F, Zhang Y, Ruan S, Xing P and Guo L (2024) YOLO SSPD: a small target cotton boll detection model during the boll-spitting period based on space-to-depth convolution. *Front. Plant Sci.* 15:1409194. doi: 10.3389/fpls.2024.1409194

YOLO SSPD: a small target cotton boll detection model during the boll-spitting period based on space-to-depth convolution

Mengli Zhang[1†], Wei Chen[2†], Pan Gao[1*], Yongquan Li[1], Fei Tan[1], Yuan Zhang[1], Shiwei Ruan[1], Peng Xing[1] and Li Guo[1*]

[1]College of Information Science and Technology, Shihezi University, Shihezi, China, [2]School of Information Science and Engineering, Xinjiang College of Science and Technology, Korla, China

Introduction: Cotton yield estimation is crucial in the agricultural process, where the accuracy of boll detection during the flocculation period significantly influences yield estimations in cotton fields. Unmanned Aerial Vehicles (UAVs) are frequently employed for plant detection and counting due to their cost-effectiveness and adaptability.

Methods: Addressing the challenges of small target cotton bolls and low resolution of UAVs, this paper introduces a method based on the YOLO v8 framework for transfer learning, named YOLO small-scale pyramid depth-aware detection (SSPD). The method combines space-to-depth and non-strided convolution (SPD-Conv) and a small target detector head, and also integrates a simple, parameter-free attentional mechanism (SimAM) that significantly improves target boll detection accuracy.

Results: The YOLO SSPD achieved a boll detection accuracy of 0.874 on UAV-scale imagery. It also recorded a coefficient of determination (R^2) of 0.86, with a root mean square error (RMSE) of 12.38 and a relative root mean square error (RRMSE) of 11.19% for boll counts.

Discussion: The findings indicate that YOLO SSPD can significantly improve the accuracy of cotton boll detection on UAV imagery, thereby supporting the cotton production process. This method offers a robust solution for high-precision cotton monitoring, enhancing the reliability of cotton yield estimates.

KEYWORDS

cotton boll detection, cotton yield estimation, transfer learning, YOLOv8, UAV

1 Introduction

Cotton yield estimation is essential for the cotton production process and can influence the price trend in the cotton market (Sarkar et al., 2023). Cotton yield estimation can be carried out by boll detection during the cotton fluffing period (Pokhrel et al., 2023; Torgbor et al., 2023). The quantity of cotton bolls directly affects the cotton harvest and is also a key indicator for assessing the quality of cotton (Shi et al., 2022). A high precision boll number detection model can quickly and accurately model yield estimation before harvesting and make planting management related decisions, which is economically vital for cotton production (Thorp et al., 2020; Naderi Mahdei et al., 2023).

Traditional cotton production information detection methods require sampling and frequent manual observation of cotton fields (Tian et al., 2022; Kurihara et al., 2023). With the continuous improvement of land transfer rate, large-scale planting rate and technological content, driven by the whole mechanization, many new technologies have been applied to the field of cotton production, improving the development of cotton production process intelligence (Muruganantham et al., 2022; Yan et al., 2022). Although high-resolution, ground-shot images are not suitable for cotton boll detection in field environments due to their high acquisition costs. As remote sensing technology develops, satellite positioning system and geographic information system (GIS), unmanned aerial vehicle (UAV) remote sensing technology has found broad applications (Dhaliwal and Williams, 2023; Hu et al., 2023; Kumar et al., 2023; Priyatikanto et al., 2023). Due to the small scale of cotton bolls and the complexity of the field background, large-scale monitoring methods such as satellite remote sensing cannot describe the detailed changes of cotton bolls in a small-scale range. Low-altitude UAV remote sensing acquires the benefit of short cycle time and fast speed, so it provides technical support for small- and medium-scale crop growth monitoring (Eskandari et al., 2020; Fernandez-Gallego et al., 2020; Hassanzadeh et al., 2021; Palacios et al., 2023).

UAVs provide excellent image acquisition flexibility at flight altitude, flight area and various weather conditions for fast and accurate crop monitoring (Bouras et al., 2023; X. Wang, Lei, et al., 2023). UAV remote sensing combined with machine learning algorithms is an essential area of re-search in target detection studies based on UAV remote sensing images. In the study of automated agave detection, the utilization of UAV image data has demonstrated remarkable accuracy (Flores et al., 2021). Moreover, red, green, blue (RGB) aerial imagery from UAV, coupled with the faster regions with convolutional neural network (Faster R-CNN) object detection model, prove effective in estimating plant density (Velumani et al., 2021). The application of UAV image data for training convolutional neural networks (CNNs) shows superior performance compared to traditional machine learning methods (Impollonia et al., 2022; Amarasingam et al., 2024; Skobalski et al., 2024; Zou et al., 2024). High-resolution images significantly enhance the accuracy of target detection. Collection of high-resolution UAV RGB images provides a methodology for counting plants at different growth stages of sunflowers and corn seedlings (Bai et al., 2022). High-resolution UAV images, when combined with suitable image segmentation algorithms, serve as a basis for detection counting and analysis. In a study focused on the detection and counting of citrus trees using high-resolution UAV images, the connected component labelling (CCL) algorithm was employed to segment and label individual citrus trees in images (Donmez et al., 2021). The relationship between image based manual counting and algorithmic counting demonstrates high precision and efficiency through the utilization of frequency filters, segmentation, and feature extraction techniques (Azizi et al., 2024; Liu et al., 2024). Given sufficient data, pre-trained deep learning models offer enhanced generalization performance in target detection tasks. The pre-trained ResNet 17 model, when applied to UAV-captured RGB images of cotton seedlings, enables real-time estimation of the quantity and canopy size of the seedlings in each frame (Feng et al., 2020). Building on the success of this method, researchers have further integrated transfer learning techniques into a new framework that combines remote sensing and deep learning to enhance processing efficiency. This integrated framework has proven particularly effective in sparse counting tasks for different plant species, such as potatoes and lettuce (Machefer et al., 2020). Estimating crop density using vegetation indices is applicable in the early and middle stages of crop growth, but its performance is limited at maturity due to the gradual onset of plant senescence, wilting leaves, and the impact of crop fruits (Huang et al., 2018).

Following the analysis of various multispectral and RGB vegetation indices, a neural network model can integrate the analytical results to estimate vegetation coverage and crop density (García-Martínez et al., 2020). Remote sensing imagery has been widely employed for crop segmentation in the later stages of crop growth, yielding significant results. UAV images are also utilized in computing the elevation difference between the crop canopy and exposed soil surface, extracting cotton height during the boll spitting period, and using it as a basis for estimating cotton yield. The specific process involves validating UAV-based height measurements using known ground reference points, segmenting crop rows, and obtaining a plant height map based on global positioning system (GPS) and image features (Feng et al., 2019). Remote sensing image segmentation can be employed to detect the quantity of target cotton bolls since cotton often exhibits distinct optical features (such as color and morphology) from branches and leaves in the later stages of growth. A cotton boll threshold segmentation detection algorithm based on UAV remote sensing images is proposed. Initially, spectral thresholds are derived from input images through adaptive applications, automatically distinguishing cotton bolls from other non-target items. Subsequently, the derived thresholds and other morphological filters are utilized for binary cotton boll classification to reduce result noise (Yeom et al., 2018). Combining UAV remote sensing data with multispectral images and cotton boll pixel coverage enables the construction of a high precision cotton boll detection model. This model primarily utilizes a Bayesian regularized back propagation (BP) neural network to predict cotton yield from the quantity of cotton bolls and spectral indices(R. Xu et al., 2018; W. Xu et al., 2021).

Due to the extension and interlacing of cotton leaves in the background of the cotton field and the complex changes in the external environment, the morphological characteristics of cotton bolls in the field are variable and overlapping. Therefore, for cotton boll detection in a field environment, the boll-spitting period is considered the ideal phase. However, due to the large number and small size of cotton bolls, a specific detection model is required to achieve accurate applications (Fue et al., 2018). The YOLO series has undergone multiple updates and iterations, making it suitable for detection and segmentation in agriculture. This model builds upon the YOLOv8 architecture with added modules for feature processing, significantly improving the detection accuracy of small objects in UAV images (G. Wang, Chen, et al., 2023). The real-time YOLOv8 model has been effectively applied for detecting kiwifruit diseases, providing real-time disease estimates (Xiang et al., 2023). Additionally, to address the challenge of strawberry ripeness detection, the YOLOv8s model and the LW-Swin Transformer module have been employed to support the strawberry picking process in orchard management (Yang et al., 2023).

This study introduces an enhanced detection model, YOLO small-scale pyramid depth-aware detection (SSPD), based on YOLOv8 to improve the accuracy of UAV-based cotton boll detection during the boll spitting period. High-resolution ground cotton boll images were initially captured and utilized to train data on network models such as YOLO SSPD. The trained model was subsequently transferred to UAV remote sensing images for the detection and counting of cotton bolls. The Detailed Process Overview is Shown in Figure 1.

2 Materials and methods

2.1 Dataset acquisition and preprocessing

This research was carried out from August to October 2021 in the Second Company of Experimental Field of Xinjiang Shihezi University (latitude 44°18′N, longitude 85°58′E, average altitude 443 m), as shown in Figure 2. The experimental area was planted

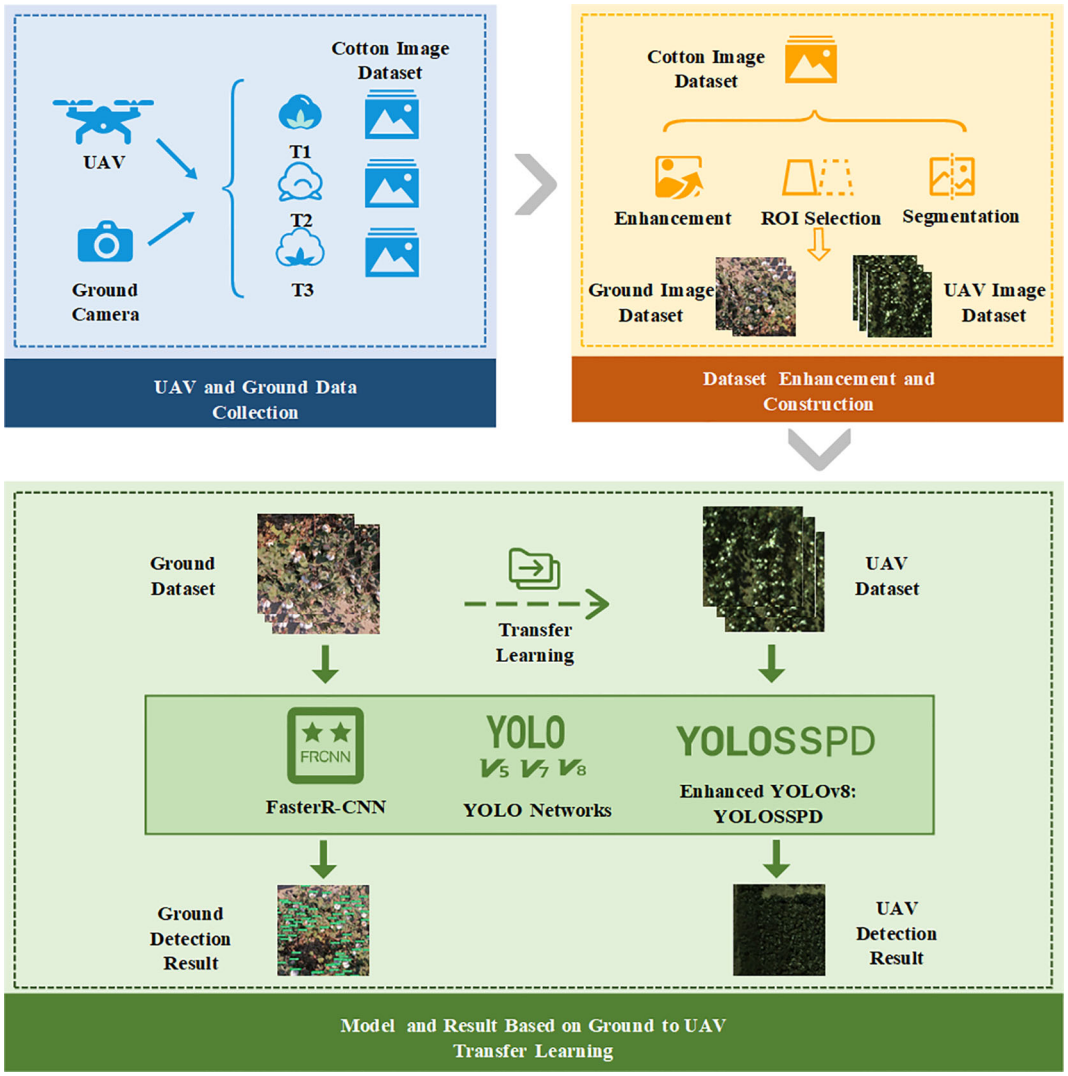


FIGURE 1
The abstract process framework of this study.

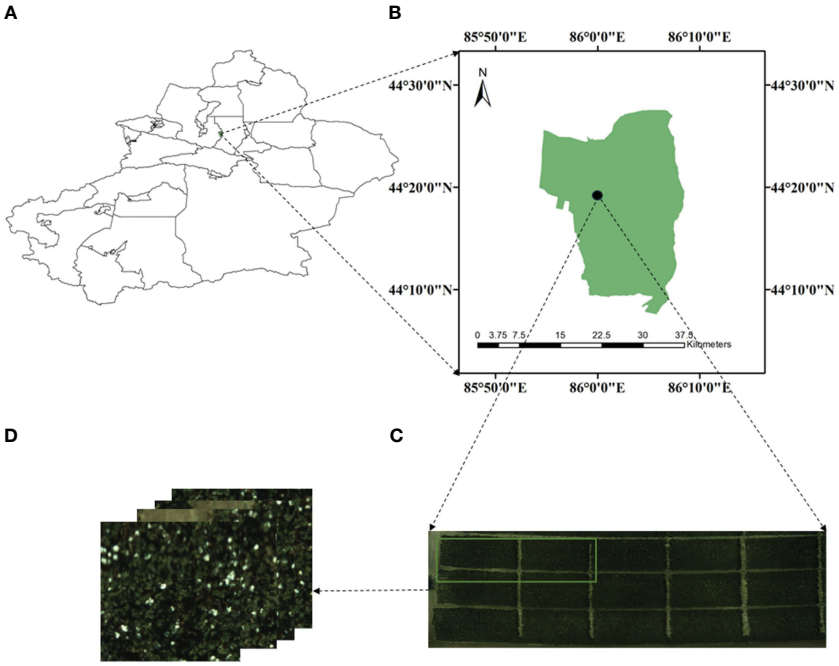


FIGURE 2
Overview of study area: **(A)** illustrates the graph of Xinjiang, **(B)** represents the area of Shihezi, **(C)** represents the testing region, Cotton boll image acquisition experimental area, the photos in **(D)** are the RGB images taken by a drone.

with "Xinlu Early No. 53" and "Xinlu Early No. 74", utilizing the planting pattern "one film, three cylinders and six rows" with the design of a comprehensive and cramped row. The chosen cotton variety was "Xinlu Early No. 53", and the planting density is 20 plants per square meter. The image data collection activities were carried out in three stages of the cotton fluffing period. The three stages of filming were 5 days after the first defoliant spraying (T1, September 8th), 3 days after the second defoliant spraying (T2, September 15th) and 7 days before cotton picking (T3, September 25th).

2.2 UAV image data acquisition and processing

This study uses a DJI M Atrice M600 PRO UAV (Shenzhen, China) with third-party hardware and software extensions, a global positioning system (GPS) positioning system, a flight imaging receiver, an a3 Pro flight controller, a Lightbridge 2 high definition (HD) digital mapper, and a remote control, with a load capacity of 6.0 kg and an Isuzu Optics real-time camera (Hsinchu County, Taiwan, China). The UAV captured datasets were all RGB images, and the real-time camera parameters are shown in Table 1. Each time the images were taken, three altitudes were flown, 60 meters, 40 meters and 20 meters from the ground. The UAV flight speed was 2.8 m/s, the camera was oriented parallel to the main flight path, the heading overlap rate was 70%, the side overlap rate was 60%, the gimbal pitch angle was -80°, and the camera mode was set to isometric intervals to increase the efficiency of the shooting as well as to obtain a clear image of the UAV. The camera configured and carried by the UAV is shown in Figure 3.

Pictures taken by UAVs are characterized by small image size, large data volume, and rich spatial information. Still, environmental factors also directly affect, such as sunshine, wind direction, etc. Therefore, even if multiple pictures are acquired in the same environment, there will be differences in sensitivity and color, which will directly affect the accuracy of the subsequent detection of feature points, thus directly affecting the final use of remote sensing data from UAVs for target detection using UAV remote sensing data. In this paper, the steps of UAV remote sensing image processing include UAV flight parameter setting, raw image acquisition, remote sensing imaging stitching, region of interest (ROI) selection and datasets cropping, and the remote sensing image processing steps are shown in Figure 4.

TABLE 1 Configuration of the hyperspectral camera carried by the drone.

Parameter	Value
Spectral bandwidth	<15nm,collimated
Base imager type	CMOS[1] imager, CMOSIS CMV[2] 2000based
Spatial resolution	408*216 per band
Frame rate	Up to 340 hyperspectral cubes/second
Pixel pitch	5.5μm
Bit depth	7or10bit
RGB pixel	4 million

[1]CMOS-complementary metal-oxide semiconductor. [2]CMV-CMOSIS machine vision.

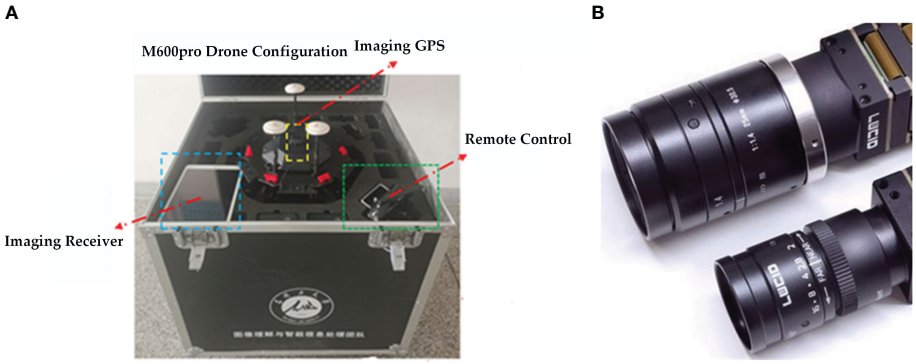


FIGURE 3
The DJI drone that collected the data, where **(A)** is the configuration of the DJI M600pro drone and **(B)** the RGB camera carried by the drone.

2.3 Datasets enhancement and construction

The image annotation tool LabelImg (free and open source, Taiwan, China) was installed, and each cotton bolls were annotated. An extensible markup language (XML) record file was generated for the training images output from each boll for better image data management and analysis in subsequent studies. In this study, the entirety of six training datasets was prepared, including 600 images of each of T1, T2 and T3 randomly selected from the ground data set and 50 segmented images of each of T1, T2 and T3 irrelevantly chosen from the UAV images. The training images were randomly cropped from the UAV RGB composite images, each with a size of 640 x 640 pixels. Ground images of 7,000, 7,500, and 6,000 were acquired for the three periods, and UAV cropped images of 250, 450, and 800 were acquired for the three flight altitudes. The above two different scales of images were randomly assigned in the proportion of 3:1:1 for the training, validation and testing of the cotton bolls detection model.

During the construction of the cotton bolls datasets, due to the direct influence of various reasons such as shooting time, climate, flight speed, camera viewpoint, etc. The cotton boll image data varied greatly, resulting in data imbalance, so it is necessary to carry out data enhancement on the cotton bolls image datasets. To further enhance the quality of the datasets, methods, for example, image rotation, image panning, image mirroring and adding image noise, are used to perform data enhancement on the existing datasets. The way the UAV enhanced the RGB image data is shown in Figure 5.

2.4 Cotton boll detection models

The models were trained on a platform equipped with an NVIDIA GeForce RTX 3060 laptop graphics processing unit (GPU) with 16GB of random-access memory (RAM). This setup provides powerful graphics processing, which is critical for handling complex computations in deep learning models. The system runs on Windows 10 x64 with a 12th generation Intel® Core™

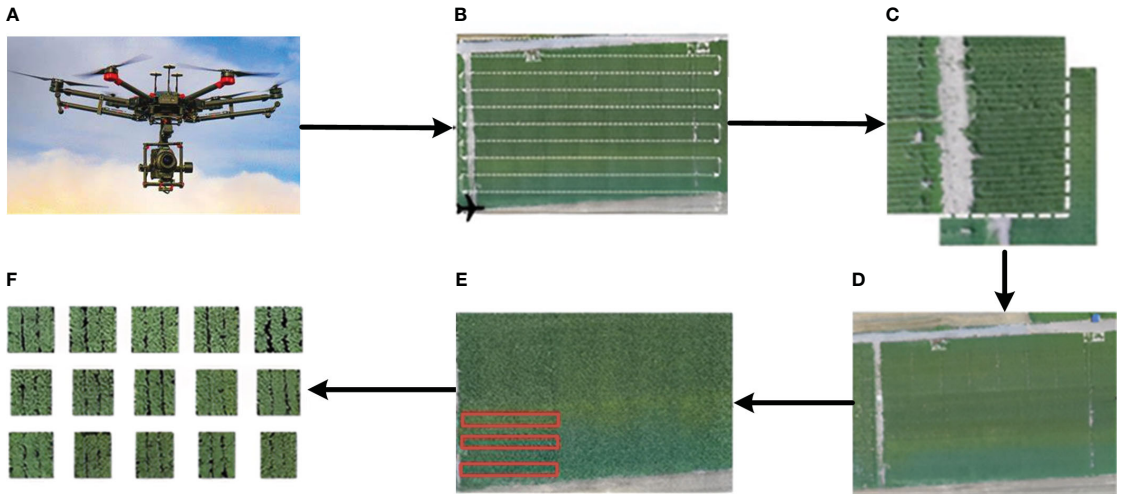


FIGURE 4
Remote sensing image processing flow: **(A)** UAV commissioning, **(B)** UAV flight parameter setting, **(C)** raw image acquisition, **(D)** remote sensing imaging stitching, **(E)** ROI selection and **(F)** datasets cropping.

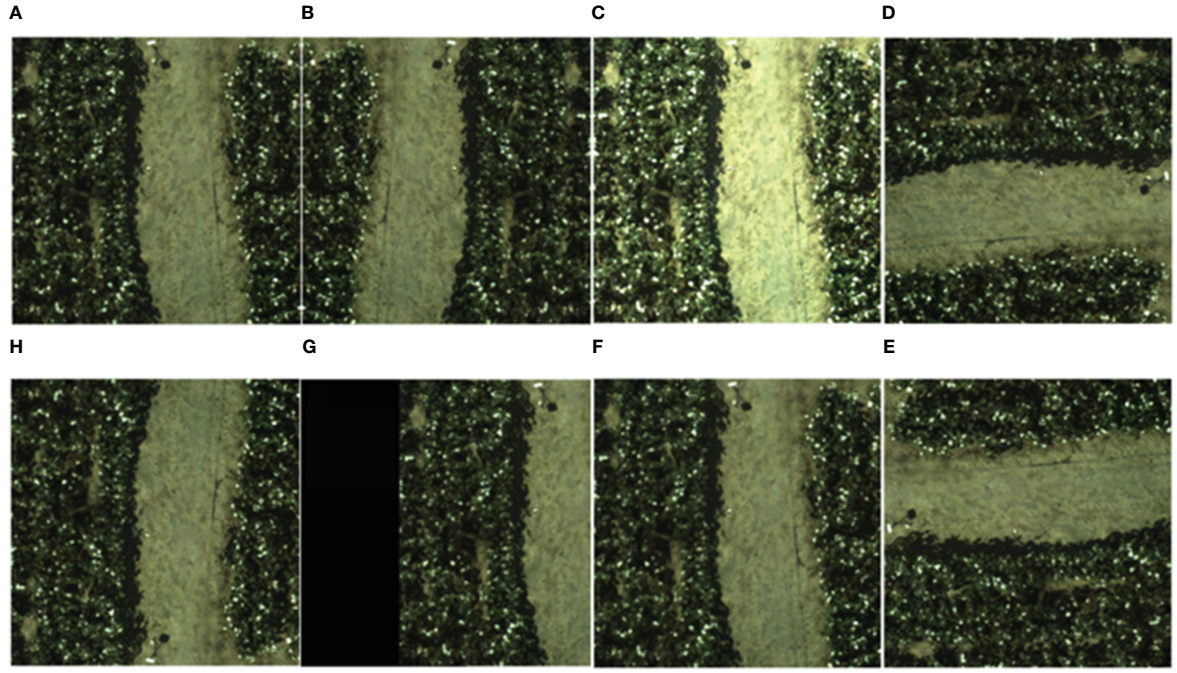


FIGURE 5
UAV expanded RGB image datasets methods: **(A)** original image, **(B)** horizontal mirroring, **(C)** increasing brightness, **(D)** rotating 90° to the right, **(E)** vertical mirroring, **(F)**image panning, **(G)** increasing noise, and **(H)** rotating 90 to the left.

i5–12500H central processing unit (CPU), which supports efficient multitasking and fast data processing. In addition, the device features 1.0TB of storage capacity, allowing for extensive data processing and model training without storage limitations. The Pytorch framework version used is 1.7.1, which is known for its flexibility and efficiency in model development. Optimized computational performance with compute unified device architecture (CUDA) 11.0 and CUDA deep neural network (cuDNN) 8.0.5 ensures faster training times and enhanced reproducibility of results.

2.4.1 Faster R-CNN

Faster R-CNN (https://github.com/jwyang/faster-rcnn.pytorch) (Mai et al., 2020) is an improved version of fast regions with convolutional neural network (Fast R-CNN) that draws features straight from the original input image. It then uses ROI Pooling to extract feature vectors of a specific length for each ROI on the feature map of the whole image. It regresses the feature vectors directly on them using multiple full convolution (FC) layers. Two FC branches are then used to predict the ROI-related categories and boxes separately, which significantly improving speed and prediction. The first part of the network architecture uses convolution layer stacking to extract the feature map from the image, then fixes the data dimensions using region pooling. The Region Proposal Network (RPN) network is the second part, which mainly serves to generate alternate regions. The third part of ROI Pooling is primarily responsible for the feature maps of the convolutional network inputs, and the exact proposals generated by the RPN training (Duan et al., 2019; Chen et al., 2020; Zhang et al., 2021), and the pooling process is used to implement edge regression and region classification. In this study, the image input size is set to 640 × 640, the learning rate is configured to 0.001, the step size is adjusted to 5, the batch size is fixed at 16, and the number of iteration rounds is 500.

2.4.2 YOLOv5

On the input side of YOLOv5 (https://github.com/ultralytics/yolov5), the mosaic data information boost technique replaces the traditional single-cut mix data information enhancement method of the previous generations. It employs the self-fitting stroke frame method and self-fitting image compression (Ghiasi et al., 2021). Cross stage partial (CSP) and focus structures are introduced in the Backbone part of the network to expand the input channels for subsequent slicing operations. The neck part of the network greatly improves the deep learning capability of the network by combining feature pyramid networks (FPN) and path aggregation network (PAN), and applies PAN to the three effective feature layers for better fusion of features from different layers. In addition, in order to obtain more accurate output results, the neck also adopts generalized intersection over union (GIOU) loss as the loss function for edge regression to achieve more efficient model analysis. In this study, the image input size is 640×640, because it is cotton boll single target detection, the output category of the network, nb_classes, is changed to 1, the training weights are yolov5s, the optimizer chosen is stochastic gradient descent (SGD), the batch size is 16, the iteration rounds epoch is 500, and the learning rate is set as 0.001, and the rest are default settings.

2.4.3 YOLOv7

YOLOv7 (https://github.com/WongKinYiu/yolov7) inherits the architecture of YOLOv5, including the configuration information settings, training process, inference and testing procedures. Additionally, YOLOv7 adopts the structure and methods of hyperparameter tuning and implicit knowledge learning from YOLOR. It also incorporates YOLOX's Optimal Transport Assignment (OTA) strategy for positive sample matching strategy. YOLOv7 itself also features an efficient aggregation network, reparametrized convolution, extra training module and model scaling (C.-Y. Wang, Bochkovskiy, and Liao 2023). Among these, the efficient aggregation network enhances the learning efficiency and aggregation ability of the network system by controlling the shortest and longest gradient paths (Zhao et al., 2023). The auxiliary training method and deep supervision in the YOLOv7 model add additional neurons to the network system to enhance the model's accuracy. Notably, the auxiliary training method is only employed during the training process and does not degrade the accuracy of the model validation and testing (Jiang et al., 2022). In this study, the parameters are set as follows, the pre-training weight is YOLOv7-tiny, the optimizer is Adam, the batch size is 8, and the epoch is 500.

2.4.4 YOLOv8

YOLOv8 (https://github.com/ultralytics/ultralytics) represents the latest advancement in the YOLO series of object detection models, showcasing superior performance in terms of both speed and accuracy compared to its predecessors. Building upon the foundation of earlier versions, YOLOv8 introduces notable enhancements. In the backbone architecture, YOLOv8 refines the C3 structure of YOLOv5 to the C2f structure. The C2f modification not only preserves the lightweight nature but also facilitates the acquisition of more informative features during the gradient descent process. Within the head component, YOLOv8 transitions from a coupled head to a decoupled head, departing from the anchor box structure employed in prior iterations in favor of an Anchor-Free approach. Moreover, YOLOv8 incorporates an outstanding dynamic allocation strategy in the design of its loss function. This strategic approach enhances the adaptability of the model during training. Notably, YOLOv8 demonstrates versatility by extending its applicability to earlier versions of the YOLO series, delivering commendable performance across image detection, segmentation, and classification tasks. The structure of Yolov8 is shown in Figure 6.

2.4.5 YOLO SSPD

YOLO SSPD is designed based on the YOLOv8 architecture to address the challenges of small and dense cotton boll targets and complex field backgrounds in UAV-scale scenarios. SPD-Conv (https://github.com/LabSAINT/SPD-Conv) is a combination of space-to-depth layer and non-strided convolution. To mitigate the loss of image information during network propagation, the SPD-Conv structure is introduced (Sunkara and Luo, 2022). Equations 1–3 elucidate the principles of SPD convolution. The input feature map X with dimensions $S\times S\times C_1$. The SPD transformation downsamples X using a scale parameter *scale*. For each position (i, j) in X, X is sliced into $scale^2$ sub-feature maps $f_{x,y}$, where $x, y \in \{0, 1, \ldots, \text{scale}-1\}$. The sub-feature maps are extracted as follows:

$$f_{x,y} = X[x:S:scale, y:S:scale] \quad (1)$$

Each sub-feature map $f_{x,y}$ downsamples X by extracting pixels at intervals of *scale*, and the dimensions of each $f_{x,y}$ are $(\frac{S}{\text{scale}}, \frac{S}{\text{scale}}, C_1)$. These sub-feature maps are then concatenated along the channel dimension to form a new feature map X':

$$X' = concatenate(\{f_{x,y} \mid x, y \in \{0, 1, \ldots, scale-1\}\}, axis = channel) \quad (2)$$

The main purpose of this transformation is to increase the channel dimension while reducing the spatial dimensions of the feature map. The dimensions of the new feature map X' are $(\frac{S}{\text{scale}}, \frac{S}{\text{scale}}, scale^2 \times C_1)$. A non-strided (stride=1) convolution operation

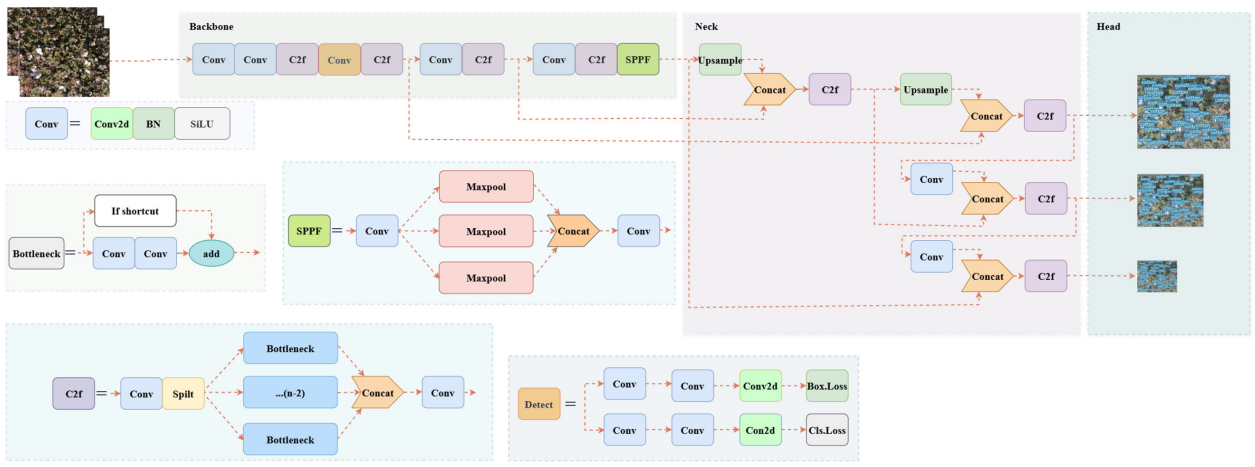


FIGURE 6
YOLOv8 model structure.

is applied to X' using $C2$ filters. This convolution transforms X' into X'' as follows:

$$X = Convolution(X', filters = C_2, stride = 1) \quad (3)$$

This convolution operation aims to retain as much discriminative feature information as possible, preventing the loss of information. The dimensions of the output feature map X'' are: $(\frac{S}{\text{scale}}, \frac{S}{\text{scale}}, C_2)$. By scaling the image proportion before inputting it into the detection network, the space-to-depth layer preserves channel dimension information throughout the feature mapping process, effectively preventing information loss (Wan et al., 2024). Additionally, non-strided convolutions are added after the space-to-depth layer to expedite image processing. The simple parameter-free attention mechanism (SimAM), while not increasing computational parameters, serves as a versatile attention mechanism, enhancing model performance. When dealing with UAV images, this not only accelerates computation speed but also improves overall model efficiency. The small target detection head finds widespread applications in the industry, addressing challenges related to inconspicuous features and potential information loss during training, thereby enhancing detection capabilities. Integrating the small target detection head into YOLO SSPD contributes to improved accuracy in identifying small target cotton bolls. Figure 7 illustrates the network structure of the YOLO SSPD.

2.5 Transfer learning based cotton boll detection from UAV RGB images

Transfer learning involves improving performance in a newly acquired task by leveraging knowledge gained from a closely related task that has already been mastered. To address the issue of limited training instances and low resolution of UAV remote sensing images, we first train the model on ground boll image data. Then, the trained model is applied to the boll recognition and detection task on UAV RGB images. Image size, quantity and quality are essential factors affecting the setting of training parameters, and in order to achieve the best training effect, these parameters must be refined to improve further the correctness and credibility of modelling (Tedesco-Oliveira et al., 2020; Park and Yu, 2021). In this study, the transfer learning model is configured with a learning rate of 0.0005, a batch size of 8, and a total of 500 iteration rounds.

2.6 Evaluation indicators

In this paper, single target detection of cotton bolls was investigated, so the model evaluation metrics selected included precision, recall, F1 score, average precision, average precision (AP) for a single class, and coefficient of determination (R^2), relative root mean square error (RMSE) and root mean square error (RRMSE), which were calculated using the formulas shown below. Equations 4–10 are introduced as metrics for subsequent model performance evaluation.

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + recall} \quad (6)$$

$$AP = \int_0^1 P(r)dr \quad (7)$$

$$R^2 = 1 - \frac{\sum_1^n (p_i - c_i)^2}{\sum_1^n (p_i - \bar{p}_i)^2} \quad (8)$$

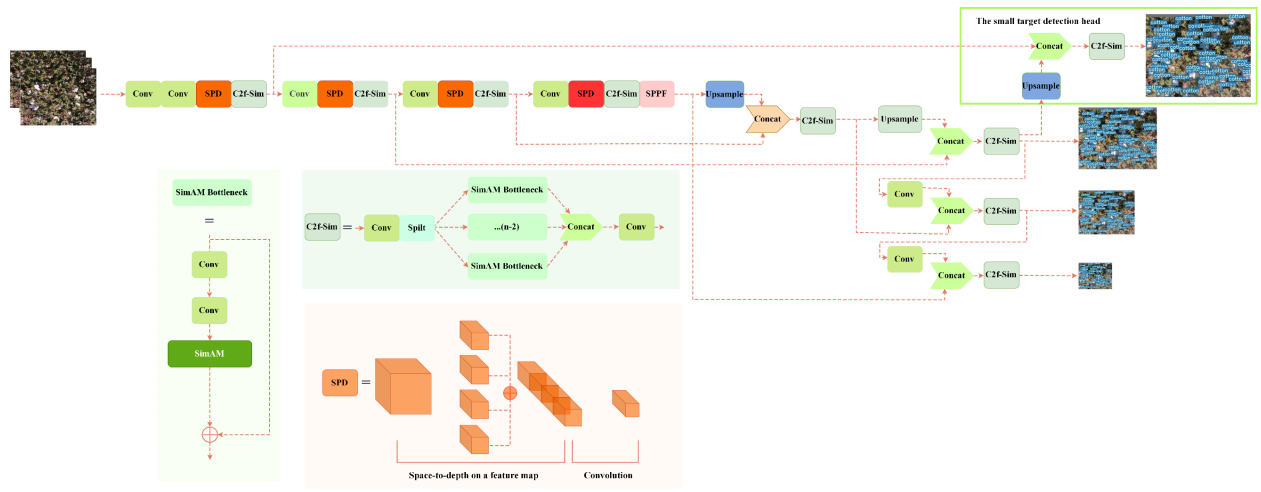


FIGURE 7
YOLO SSPD model structure.

$$RMSE = \sqrt{\frac{\sum_1^n (p_i - c_i)^2}{n}} \tag{9}$$

$$RRMSE = \frac{\sqrt[n]{\frac{1}{n}\sum_1^n (p_i - c_i)}}{\sum_1^n p_i} \times 100\,\% \tag{10}$$

Where True positive (*TP*) represents correct prediction of cotton bolls, False positive (*FP*) represents misidentification of background noise as cotton bolls, and False negative (*FN*) represents misidentification of cotton bolls as background noise. The value range of *Precision* and *Recall* is between 0 and 1, so the value range of *AP* is also in the range of [0,1]. p_i, $\bar{p}_i$ and c_i are the quantity of manually labelled bolls in the *i*-th image, the mean of the amount of manually labelled bolls in the *i*-th image and the count of bolls obtained by prediction, correspondingly. n is the total of test images.

3 Results

3.1 Results of ground cotton boll detection models

Table 2 displays the outcomes of cotton boll recognition and detection in ground image data at different time intervals utilizing various object detection networks. When employing models like Faster R-CNN, a consistent performance trend is observed across different time periods, with T2 > T1 > T3. This phenomenon is attributed to the suboptimal effect of defoliant spraying during the T1 period. In the T3 period, when cotton flowers are fully open, distinguishing targets becomes challenging, resulting in instances where a single cotton boll is identified as multiple ones. Additionally, due to the proximity of cotton bolls, multiple instances are detected as a single cotton boll. The second phase, occurring after the second defoliant spraying, emerges as the optimal period for cotton boll detection. During this phase, there is minimal interference from leaves, and the branching of cotton plants is less pronounced, resulting in relatively independent cotton bolls. Therefore, it is recommended to select T2 as the golden period for cotton boll detection in subsequent studies involving transfer learning. Figure 8 illustrates the detection results of different networks on ground cotton boll images at time interval T2, with magenta boxes indicating missed detections. Despite achieving higher detection recall rates in ground cotton boll image data, the Faster R-CNN model tends to experience overfitting due to its robust deep feature extraction capabilities. This results in an increased false positive rate, significantly impacting the balance between precision and recall. The YOLO v5 model exhibits some shortcomings, with less evident features and smaller cotton bolls going unrecognized. YOLOv7 employs multi-layer modification techniques in the model, halving aspect ratios, doubling channels, and reducing downsampling. Consequently, at the same volume, YOLOv7 outperforms YOLOv5 in efficiently detecting targets with higher accuracy and faster speed. However, there are still some shadowed and concealed cotton bolls that go undetected. The YOLOv8 model provides a scaled-down version based on scaling factors, catering to the requirements of cotton boll detection scenes. Nevertheless, further improvements are needed for low-resolution small target detection. The proposed YOLO SSPD in this study evidently demonstrates high-precision cotton boll recognition at the ground scale.

TABLE 2 Model testing results for ground image datasets.

Model	Time	Precision(%)	Recall(%)	F1-Score(%)	AP_{50}(%)
Faster R-CNN	T1	80.3	85.2	82.7	83.9
	T2	**81.6**	**86.9**	**84.2**	**83.0**
	T3	78.2	82.1	80.1	81.1
YOLOv5	T1	81.1	84.8	81.9	82.2
	T2	**81.7**	**83.4**	**82.5**	**83.1**
	T3	79.2	81.6	80.4	81.0
YOLOv7	T1	83.1	85.2	84.1	84.8
	T2	**83.8**	**85.8**	**85.0**	**85.6**
	T3	80.2	82.6	81.4	81.3
YOLOv8	T1	81.8	83.8	83.7	82.1
	T2	**84.6**	**86.0**	**84.3**	**82.6**
	T3	80.9	81.7	82.6	82.3
YOLO SSPD	T1	84.1	87.3	85.7	86.5
	T2	**85.2**	**88.9**	**87.0**	**88.1**
	T3	81.1	84.6	82.8	83.9

The values are bolded to emphasize that the best-performing models for each period consistently peaked in T2.

10.3389/fpls.2024.1409194

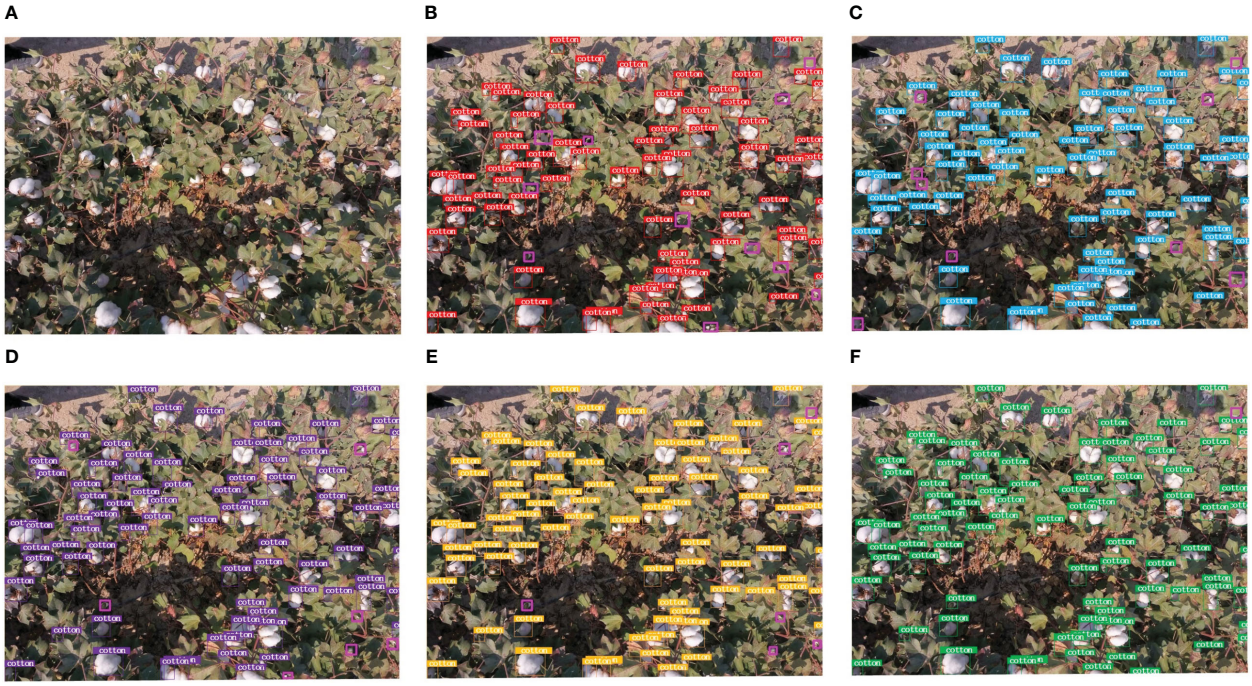


FIGURE 8
The model detection results (Pinkish-purple boxes show missed bolls): **(A)** Original image, **(B)** Faster R-CNN, **(C)** YOLOv5, **(D)** YOLOv7, **(E)** YOLOv8, **(F)** YOLO SSPD.

3.2 Results of UAV image cotton boll detection and transfer learning

The images captured by the UAV at flight altitudes of 20 meters, 40 meters, and 60 meters all exhibit distinct features of open cotton bolls, with the images obtained at a 20-meter flight altitude having the highest resolution. The contrast between the target cotton bolls and the background is more pronounced, resulting in the highest detection accuracy. Subsequent research focuses on the UAV image dataset obtained at a 20-meter altitude. When evaluating the impact of transfer learning, Tables 3, 4 present the cotton boll detection results using the five aforementioned detection models on the UAV RGB image dataset during the T2 period, along with the results after transfer learning on the UAV images during the same period. The detection results of different models on cotton boll images are depicted in Figure 9. Due to the small scale of detection targets on the drone, a portion of the region enclosed by red rectangles in the original image detection results was cropped for comparison. Comparative analysis of detection results before and after model transfer indicates overall improvement in the detection efficiency of all model's post-transfer, with the YOLO SSPD model exhibiting the highest detection efficiency. Before model transfer, the detection time for each image in the drone RGB image dataset was 51ms, while after model transfer, the average detection time for each image in the drone RGB image dataset was reduced to 22ms. These results signify the effectiveness of model transfer. The optimal YOLO SSPD model achieves an optimal balance between detection accuracy and detection rate.

3.3 Validation of cotton boll detection models

Neural networks are often perceived as black-box models with limited interpretability. However, employing class activation maps (CAM) on a trained model allows for a visual understanding of its

TABLE 3 UAV image datasets models testing results.

Model	Time	Precision(%)	Recall(%)	F1-Score(%)	AP_{50}(%)
Faster R-CNN	T2	77.6	84.3	80.8	83.2
YOLOv5	T2	80.3	84.2	82.2	83.6
YOLOv7	T2	82.1	85.6	83.9	84.1
YOLOv8	T2	82.6	86.1	83.8	84.6
YOLO SSPD	**T2**	**85.3**	**88.0**	**86.6**	**86.9**

The bolding is used to highlight the superior metrics of the best-performing models.

TABLE 4 Testing results after models transfer.

Model	Time	Precision(%)	Recall(%)	F1-Score(%)	AP_{50}(%)
Faster R-CNN	T2	79.9	85.6	82.7	83.9
YOLOv5	T2	81.1	86.4	84.8	84.3
YOLOv7	T2	83.8	87.1	85.4	86.0
YOLOv8	T2	84.1	87.2	85.6	86.4
YOLO SSPD	**T2**	**87.4**	**89.3**	**87.8**	**88.0**

The bolding is used to highlight the superior metrics of the best-performing models.

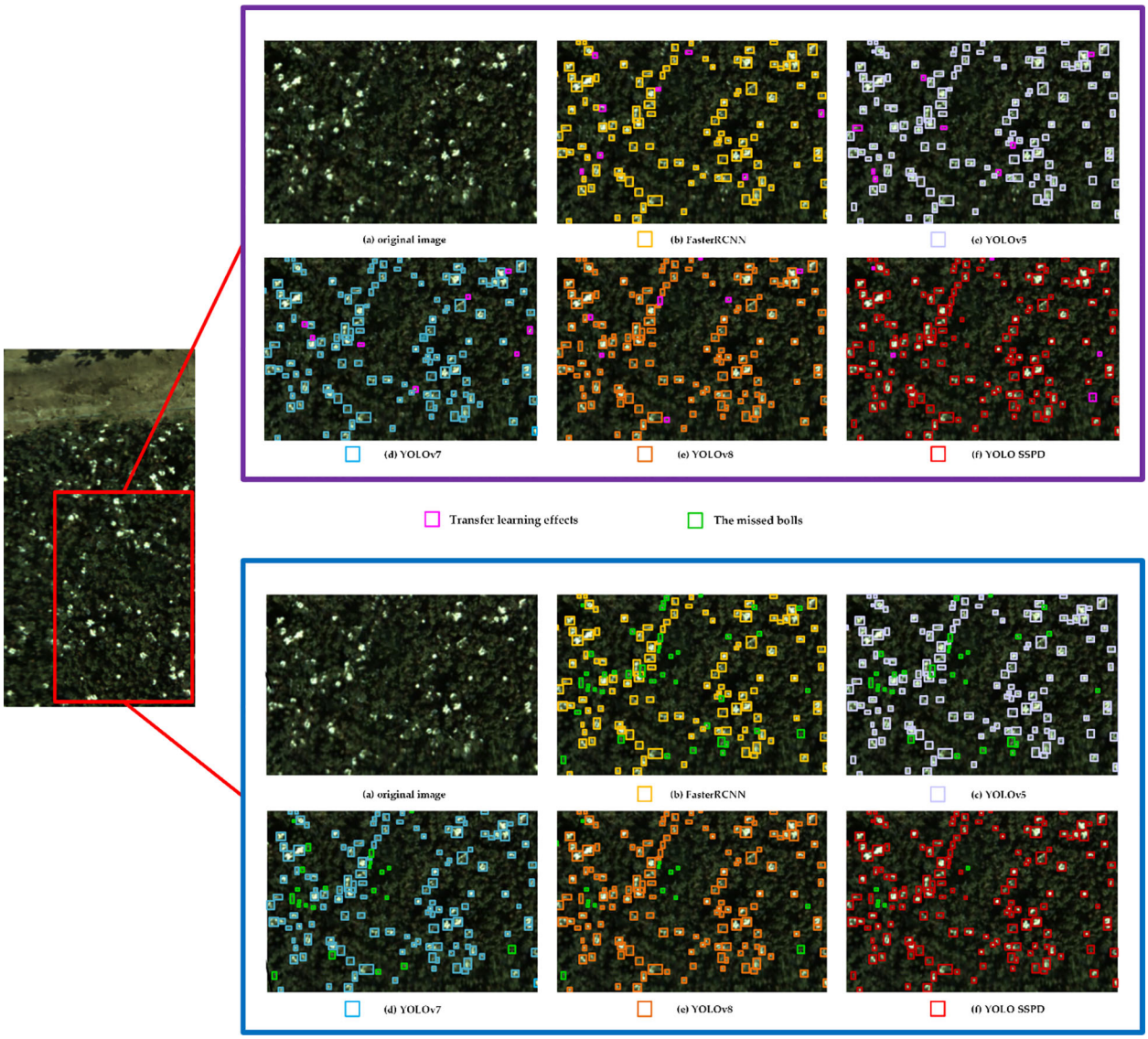


FIGURE 9
Below is a comprehensive comparison of the five object detection models before and after transfer learning. Purple boxes represent detection results before transfer learning, while blue boxes represent results after transfer learning. Different colored boxes in the images denote the effectiveness of different detection models, with yellow indicating Faster R-CNN detection, light purple for YOLOv5, blue for YOLOv7, orange for YOLOv8, and red for YOLO SSPD detection results.

principles. CAM (https://github.com/jacobgil/pytorch-grad-cam) typically operates on the last convolutional layer of the model to extract class activation maps corresponding to input images (Zhou et al., 2016). These CAMs, which are the same size as the input images, facilitate the visualization of predicted class scores and highlight detected objects. The generation of heatmaps involves overlaying weighted feature maps obtained from CAM. Within these heatmaps, the degree of network response in different regions of the input image can be observed. Larger heatmap ranges indicate the presence of more predicted class targets in the corresponding regions, while darker colors signify greater contributions to the predicted results. To further enhance cotton boll detection, a visual analysis of the detection results for each model was conducted through heatmap visualization, providing insights into the neural network models. As shown in Figure 10, Faster R-CNN focuses on prominent features of cotton bolls, making it susceptible to information loss in small target detection, evident in the discrete distribution of the heatmap. YOLOv5's feature pyramid structure exhibits limitations in recognizing obscured and smaller cotton boll features accurately. While YOLOv7 has a larger model width and depth compared to YOLOv5, resulting in the extraction of more features, the heatmap's predominantly light colors indicate that these positions contribute less to the network output, indicating insufficient feature extraction for practical applications. YOLOv8, with its ability to adjust the model scale for detection, outperforms the first three models in small target scenarios. However, the large-scale field images captured by the UAV exhibit diverse characteristics of open cotton bolls and suffer from lower resolution issues. This leads to YOLOv8's focus on concentrated open cotton bolls, indicating a need for further attention to the discrete small cotton boll targets. YOLO SSPD, by introducing SPD convolution and a small target detection head onto the YOLOv8 model, significantly captures a broader target range in low-resolution small target images, achieving precise detection in the images.

3.4 Validation of cotton boll counting model

This study employed the determination coefficient, RMSE, and RRMSE as metrics to evaluate the counting effectiveness of the model. Combining the YOLO SSPD detection model with transfer learning, counting was performed on UAV RGB image data. The results demonstrate that the detection model, after being fine-tuned through a transfer learning approach, achieved an R^2 of 0.86, RMSE

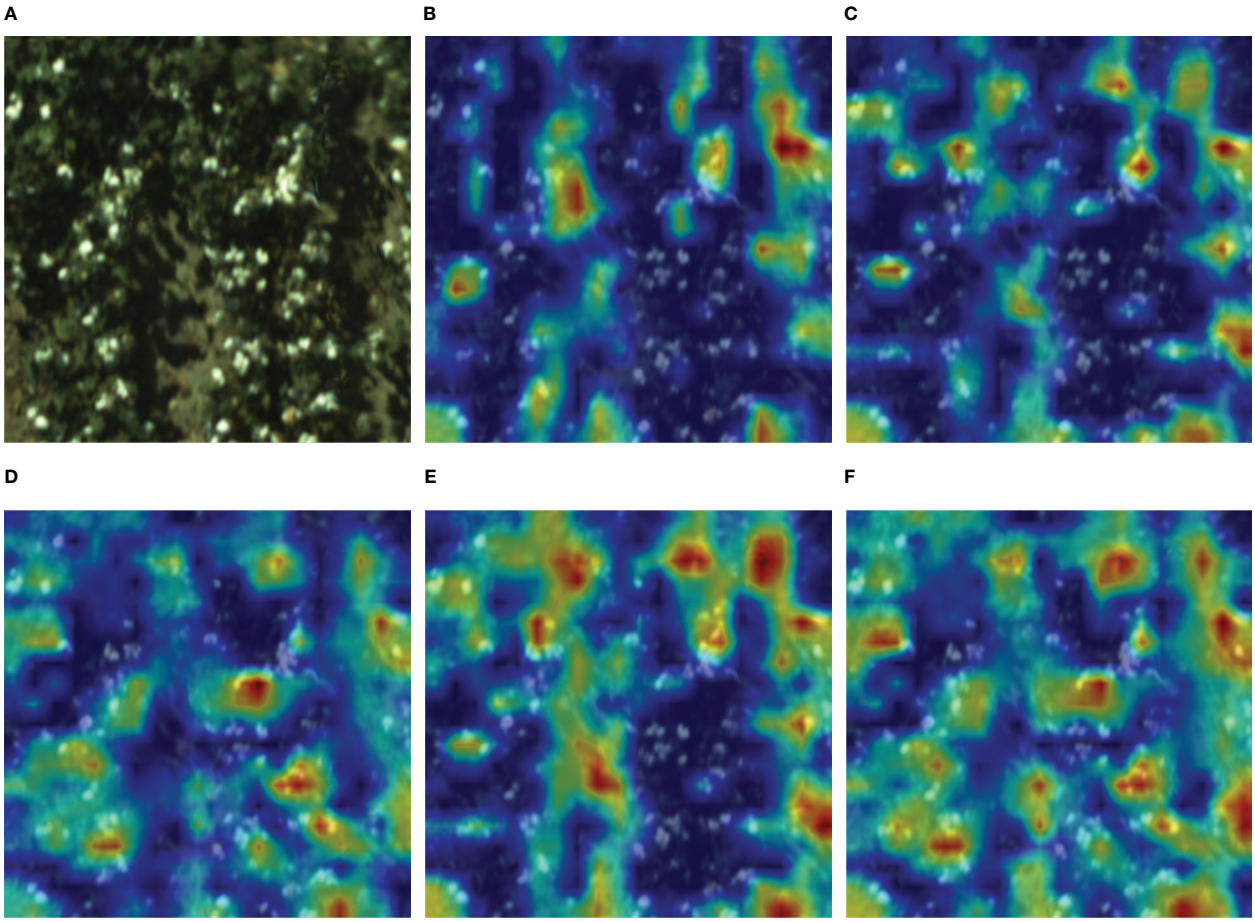


FIGURE 10
Five object detection models' heatmaps: **(A)** Original image, **(B)** Faster R-CNN, **(C)** YOLOv5, **(D)** YOLOv7, **(E)** YOLOv8, **(F)** YOLO SSPD.

of 12.38, RRMSE of 11.19%, and an AP of 88.9%, thus indicating a robust counting performance. Figure 11 showcases how the integration of the YOLO SSPD model with transfer learning techniques enhances its ability to detect and count cotton bolls accurately in 20m resolution UAV images during the T2 period.

4 Discussion

Boll detection in the pre-harvest stage of cotton can realize the assessment of cotton yield, so as to provide scientific and effective resource allocation and management strategies. As cotton bolls are not obvious in the early growth stage in a complex field background environment, the stages of cotton flocculation can be selected to accurately and reliably identify and locate cotton bolls. In this study, the three stages of cotton flocculation were selected to be captured by UAV and on the ground. In order to reduce the interference of cotton leaves and achieve better detection conditions, 5 days after the first spraying of defoliant (T1), 3 days after the second spraying of defoliant (T2), and 7 days before the cotton picking (T3) were selected, and the image of T2 got the best detection accuracy in the subsequent experimental results. In the process of cotton boll data acquisition, although the effects of UAV shooting time stage, weather conditions, UAV flight speed, camera shooting angle and other factors on the quality of ground image data and remotely sensed data were taken into account, factors such as different degrees of shading and background clutter in the cotton field in the natural environment still have a significant impact on the detection accuracy (Kang et al., 2022, 2023; Li et al., 2022; Li et al., 2020). Data enhancement can balance and enrich the cotton boll image datasets, better realize the acquisition of cotton boll features, and also reduce the workload of manual labelling.

For the case of boll detection by UAV in small-scale cotton fields, which is limited in resolution and insufficient in the number of samples obtained, ground photography was conducted to obtain sufficient ground open boll data. From the perspective of transfer learning, many ground images were used to train the deep learning model. After reaching a higher accuracy, the model was transferred so that the model could achieve a good detection accuracy on UAV images with a smaller dataset. The specific steps were, on the ground cotton boll image datasets, to investigate the cotton boll detection effect of different target detection networks in different periods through comparative experiments. Then, on UAV RGB image data, the performance of different target detection networks on cotton boll detection at UAV scale and different periods were investigated through comparison and transfer learning (Meng et al., 2019). In terms of model performance, Faster R-CNN based on Region Proposal Networks could extract target cotton bolls, but the model was complex, had slow training speed, and was prone to overfitting. Due to different growth conditions, cotton bolls during the boll spitting period exhibit varying shapes and color characteristics. The feature extraction capability of Faster R-CNN was too strong, leading to failures of recognizing some cotton bolls. YOLOv5 introduced CSPDarknet53 as the backbone network and employed the PANet structure to enhance feature fusion, demonstrating good performance in both accuracy and speed. However, when applied to cotton boll detection in UAV images, the YOLOv5 model produces numerous instances of false negatives. YOLOv7 builds on YOLOv5 by introducing architectures such as the Efficient Layer Aggregation Network, but it exhibits weak generalization, with variations in different scenes and poor performance in small object detection tasks. YOLOv8 was the latest achievement in the YOLO series at the time, featuring adjustable scaling coefficients and excellent application in practical scenarios with small targets. The proposed YOLO SSPD object detection model further improves the detection accuracy of small cotton bolls from UAVs by building upon YOLOv8. Experimental results indicate that YOLO SSPD performs best on both the ground cotton boll image dataset (T2) and the UAV RGB image dataset(T2). The accuracy of cotton boll detection in UAV scale is enhanced through the transfer model, contributing to improved accuracy in cotton yield prediction (Wang et al., 2021; Rodriguez-Sanchez et al., 2022). The combination of the YOLO SSPD detection model and transfer learning methods excels in

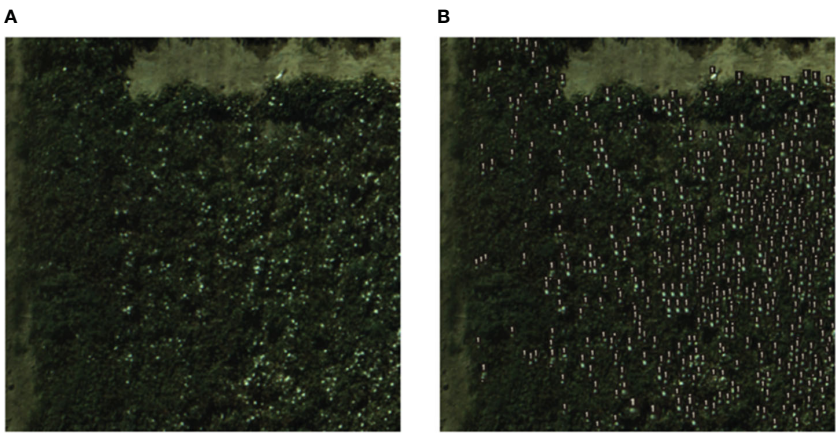


FIGURE 11
The model detection results: **(A)** Real ground boll counts, **(B)** YOLO SSPD results (UAV imagery).

detecting cotton bolls in complex environments from UAV RGB image data, providing a more precise representation of the specific locations of targets. The counting results accurately reflect the number of cotton bolls during the boll spitting stage, closely matching actual counting results (Siegfried et al., 2023). Utilizing the YOLO SSPD model for counting cotton bolls in UAV-scale images can be appropriately applied in practical cotton production processes (Qiu et al., 2022; Lang et al., 2023).

Although some progress has been made in this study, there are still many issues that need to be explored and solved in depth. (1) This study is based on cotton boll image datasets collected by ground and UAV at three altitudes (20 m, 40 m and 60 m). The image resolution of the images collected at 40 m and 60 m flight altitudes is not high, which impacts the precision of cotton boll detection and recognition. The UAV can be upgraded subsequently in terms of the camera pixels and the frame rate. High-resolution UAV images are able to achieve higher accuracy using the method proposed in this paper. (2) In the future, with a focus on enhancing the efficacy of cotton boll detection, multi-scale image fusion algorithms can be targeted to expand the detection area while improving the image resolution. Further, the large-scale cotton field yield estimation combined with satellite remote sensing images can be practically applied to a broader range of production research.

5 Conclusions

This study proposes a target detection network, YOLO SSPD, based on YOLOv8, specifically designed for detecting cotton bolls during the boll spitting period. In ground-based cotton boll image detection, the model was trained alongside four other object detection models until convergence. Subsequently, transfer learning was employed to apply these models to UAV-based cotton boll image detection. A comparison with four other models shows that YOLO SSPD outperforms them all. In the T2 period, the detection accuracy on UAV cotton boll images reaches 0.874, and the cotton boll count R^2 is 0.86. The results indicate that utilizing transfer learning and the YOLO SSPD detection model significantly improves the accuracy of cotton boll detection. The outcomes of this study serve as a practical tool in the cotton production process, enhancing the efficiency of cotton information detection. They also provide a basis for agricultural researchers to make timely decisions in cotton management, ultimately improving cotton yield and quality.

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

MZ: Conceptualization, Investigation, Methodology, Writing – original draft. WC: Conceptualization, Resources, Software, Writing – review & editing. PG: Conceptualization, Funding acquisition, Project administration, Supervision, Writing – review & editing. YL: Methodology, Writing – review & editing. FT: Validation, Visualization, Writing – review & editing. YZ: Data curation, Validation, Writing – review & editing. SR: Validation, Writing – review & editing. PX: Formal analysis, Writing – review & editing. LG: Data curation, Validation, 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. 62265015) and Eight division Shihezi City key areas of innovation team plan (2023TD01).

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

Amarasingam, N., Vanegas, F., Hele, M., Warfield, A., and Gonzalez, F. (2024). Integrating artificial intelligence and UAV-acquired multispectral imagery for the mapping of invasive plant species in complex natural environments. *Remote Sens.* 16, 15825. doi: 10.3390/rs16091582

Azizi, A., Zhang, Z., Rui, Z., Li, Y., Igathinathane, C., Flores, P., et al. (2024). Comprehensive wheat lodging detection after initial lodging using UAV RGB images. *Expert Syst. Appl.* 238, 121788. doi: 10.1016/j.eswa.2023.121788

Bai, Y., Nie, C., Wang, H., Cheng, M., Liu, S., Yu, X., et al. (2022). A fast and robust method for plant count in sunflower and maize at different seedling stages using high-resolution UAV RGB imagery. *Precis. Agric.* 23, 1720–17425. doi: 10.1007/s11119-022-09907-1

Bouras, El h., Olsson, P.-O., Thapa, S., Díaz, JesúsM., Albertsson, J., and Eklundh, L. (2023). Wheat Yield Estimation at High Spatial Resolution through the Assimilation of Sentinel-2 Data into a Crop Growth Model. *Remote Sens.* 15, 44255. doi: 10.3390/rs15184425

Chen, K., Lin, W., Li, J., See, J., Wang, J., and Zou, J. (2020). AP-loss for accurate one-stage object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 3782–37985. doi: 10.1109/TPAMI.2020.2991457

Dhaliwal, D. S., and Williams, M. M. (2023). Sweet corn yield prediction using machine learning models and field-level data. *Precis. Agriculture*. doi: 10.1007/s11119–023-10057–1

Donmez, C., Villi, O., Berberoglu, S., and Cilek, A. (2021). Computer vision-based citrus tree detection in a cultivated environment using UAV imagery. *Comput. Electron. Agric.* 187, 106273. doi: 10.1016/j.compag.2021.106273

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., and Tian, Q. (2019). "Centernet: Keypoint triplets for object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*. doi: 10.1109/ICCV43118.2019

Eskandari, R., Mahdianpari, M., Mohammadimanesh, F., Salehi, B., Brisco, B., and Homayouni, S. (2020). Meta-analysis of unmanned aerial vehicle (UAV) imagery for agro-environmental monitoring using machine learning and statistical models. *Remote Sens.* 12, 35115. doi: 10.3390/rs12213511

Feng, A., Zhang, M., Sudduth, K. A., Vories, E. D., and Zhou, J. (2019). Cotton yield estimation from UAV-based plant height. *Trans. ASABE* 62, 393–4045. doi: 10.13031/trans.13067

Feng, A., Zhou, J., Vories, E. D., Sudduth, K. A., and Zhang, M. (2020). Yield estimation in cotton using UAV-based multi-sensor imagery. *Biosyst. Eng.* 193, 101–114. doi: 10.1016/j.biosystemseng.2020.02.014

Fernandez-Gallego, J. A., Lootens, P., Borra-Serrano, I., Derycke, V., Haesaert, G., Roldán-Ruiz, I., et al. (2020). Automatic wheat ear counting using machine learning based on RGB UAV imagery. *Plant J.* 103, 1603–16135. doi: 10.1111/tpj.14799

Flores, D., González-Hernández, I., Lozano, R., Vazquez-Nicolas, J. M., and Toral, J. L. H. (2021). Automated agave detection and counting using a convolutional neural network and unmanned aerial systems. *Drones* 5, 45. doi: 10.3390/drones5010004

Fue, K. G., Porter, W. M., and Rains, G. C. (2018). "Deep Learning based Real-time GPU-accelerated Tracking and Counting of Cotton Bolls under Field Conditions using a Moving Camera," in *2018 ASABE Annual International Meeting* (St. Joseph, MI).

García-Martínez, H., Flores-Magdaleno, H., Ascencio-Hernández, R., Khalil-Gardezi, A., Tijerina-Chávez, L., Mancilla-Villa, O. R., et al. (2020). Corn grain yield estimation from vegetation indices, canopy cover, plant density, and a neural network using multispectral and RGB images acquired with unmanned aerial vehicles. *Agriculture* 10, 2775. doi: 10.3390/agriculture10070277

Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.-Y., Cubuk, E. D., et al. (2021). "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. doi: 10.1109/CVPR46437.2021.00294

Hassanzadeh, A., Zhang, F., Aardt, J. v., Murphy, S. P., and Pethybridge., S. J. (2021). Broadacre crop yield estimation using imaging spectroscopy from unmanned aerial systems (UAS): A field-based case study with snap bean. *Remote Sens.* 13, 32415. doi: 10.3390/rs13163241

Hu, T., Zhang, X., Bohrer, G., Liu, Y., Zhou, Y., Martin, J., et al. (2023). Crop yield prediction via explainable AI and interpretable machine learning: Dangers of black box models for evaluating climate change impacts on crop yield. *Agric. For. Meteorol.* 336, 109458. doi: 10.1016/j.agrformet.2023.109458

Huang, H., Lan, Y., Deng, J., Yang, A., Deng, X., Zhang, L., et al. (2018). A semantic labeling approach for accurate weed mapping of high resolution UAV imagery. *Sensors* 18. doi: 10.3390/s18072113

Impollonia, G., Croci, M., Ferrarini, A., Brook, J., Martani, E., Blandinières, H., et al. (2022). UAV remote sensing for high-throughput phenotyping and for yield prediction of miscanthus by machine learning techniques. *Remote Sens.* 14, 29275. doi: 10.3390/rs14122927

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* 12, 16595. doi: 10.3390/agriculture12101659

Kang, X., Huang, C., Zhang, L., Wang, H., Zhang, Z., and Lv, X. (2023). Regional-scale cotton yield forecast via data-driven spatio-temporal prediction (STP) of solar-induced chlorophyll fluorescence (SIF). *Remote Sens. Environ.* 299, 113861. doi: 10.1016/j.rse.2023.113861

Kang, X., Huang, C., Zhang, L., Zhang, Z., and Lv, X. (2022). Downscaling solar-induced chlorophyll fluorescence for field-scale cotton yield estimation by a two-step convolutional neural network. *Comput. Electron. Agric.* 201, 107260. doi: 10.1016/j.compag.2022.107260

Kumar, C., Mubvumba, P., Huang, Y., Dhillon, J., and Reddy, K. (2023). Multi-stage corn yield prediction using high-resolution UAV multispectral data and machine learning models. *Agronomy* 13, 12775. doi: 10.3390/agronomy13051277

Kurihara, J., Nagata, T., and Tomiyama, H. (2023). Rice yield prediction in different growth environments using unmanned aerial vehicle-based hyperspectral imaging. *Remote Sens.* 15, 20045. doi: 10.3390/rs15082004

Lang, P., Zhang, L., Huang, C., Chen, J., Kang, X., Zhang, Z., et al. (2023). Integrating environmental and satellite data to estimate county-level cotton yield in Xinjiang Province. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.1048479

Li, F., Bai, J., Zhang, M., and Zhang, R. (2022). Yield estimation of high-density cotton fields using low-altitude UAV imaging and deep learning. *Plant Methods* 18, 555. doi: 10.1186/s13007-022-00881-3

Li, N., Lin, H., Wang, T., Li, Y., Liu, Y., Chen, X., et al. (2020). Impact of climate change on cotton growth and yields in Xinjiang, China. *Field Crops Res.* 247, 107590. doi: 10.1016/j.fcr.2019.107590

Liu, P., Qian, W., and Wang, Y. (2024). YWnet: A convolutional block attention-based fusion deep learning method for complex underwater small target detection. *Ecol. Inf.* 79, 102401. doi: 10.1016/j.ecoinf.2023.102401

Machefer, M., Lemarchand, F., Bonnefond, V., Hitchins, A., and Sidiropoulos, P. (2020). Mask R-CNN refitting strategy for plant counting and sizing in UAV imagery. *Remote Sens.* 12, 30155. doi: 10.3390/rs12183015

Mai, X., Zhang, H., Jia, X., and Meng, M. Q.-H. (2020). Faster R-CNN with classifier fusion for automatic detection of small fruits. *IEEE Trans. Automation Sci. Eng.* 17, 1555–15695. doi: 10.1109/TASE.8856

Meng, L., Liu, H., Zhang, X., Ren, C., Ustin, S., Qiu, Z., et al. (2019). Assessment of the effectiveness of spatiotemporal fusion of multi-source satellite images for cotton yield estimation. *Comput. Electron. Agric.* 162, 44–52. doi: 10.1016/j.compag.2019.04.001

Muruganantham, P., Wibowo, S., Grandhi, S., Samrat, N. H., and Islam, N. (2022). A systematic literature review on crop yield prediction with deep learning and remote sensing. *Remote Sens.* 14, 19905. doi: 10.3390/rs14091990

Naderi Mahdei, K., Esfahani, S. M. J., Lebailly, P., Dogot, T., Passel, S. V., and Azadi, H. (2023). Environmental impact assessment and efficiency of cotton: the case of Northeast Iran. *Environment Dev. Sustainability* 25, 10301–103215. doi: 10.1007/s10668-022-02490-5

Palacios, F., Diago, M. P., Melo-Pinto, P., and Tardaguila, J. (2023). Early yield prediction in different grapevine varieties using computer vision and machine learning. *Precis. Agric.* 24, 407–4355. doi: 10.1007/s11119-022-09950-y

Park, J., and Yu, W. (2021). A sensor fused rear cross traffic detection system using transfer learning. *Sensors* 21, 60555. doi: 10.3390/s21186055

Pokhrel, A., Virk, S., Snider, J. L., Vellidis, G., Hand, L. C., et al. (2023). Estimating yield-contributing physiological parameters of cotton using UAV-based imagery. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1248152

Priyatikanto, R., Lu, Y., Dash, J., and Sheffield, J. (2023). Improving generalisability and transferability of machine-learning-based maize yield prediction model through domain adaptation. *Agric. For. Meteorol.* 341, 109652. doi: 10.1016/j.agrformet.2023.109652

Qiu, R., He, Y., and Zhang, M. (2022). Automatic detection and counting of wheat spikelet using semi-automatic labeling and deep learning. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.872555

Rodriguez-Sanchez, J., Li, C., and Paterson, A. H. (2022). Cotton yield estimation from aerial imagery using machine learning approaches. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.870181

Sarkar, S., Zhou, J., Scaboo, A., Zhou, J., Aloysius, N., and Lim, T. T. (2023). Assessment of soybean lodging using UAV imagery and machine learning. *Plants* 12, 28935. doi: 10.3390/plants12162893

Shi, G., Du, X., Du, M., Li, Q., Tian, X., Ren, Y., et al. (2022). Cotton yield estimation using the remotely sensed cotton boll index from UAV images. *Drones* 6, 254. doi: 10.3390/drones6090254

Siegfried, J., Adams, C. B., Rajan, N., Hague, S., Schnell, R., and Hardin, R. (2023). Combining a cotton 'Boll Area Index' with in-season unmanned aerial multispectral and thermal imagery for yield estimation. *Field Crops Res.* 291, 108765. doi: 10.1016/j.fcr.2022.108765

Skobalski, J., Sagan, V., Alifu, H., Akkad, O. A., Lopes, F. A., and Grignola, F. (2024). Bridging the gap between crop breeding and GeoAI: Soybean yield prediction from multispectral UAV images with transfer learning. *ISPRS J. Photogrammetry Remote Sens.* 210, 260–281. doi: 10.1016/j.isprsjprs.2024.03.015

Sunkara, R., and Luo, T. (2022). "No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects," in *ECML/PKDD*.

Tedesco-Oliveira, D., da Silva, R. P., Maldonado, W., and Zerbato, C. (2020). Convolutional neural networks in predicting cotton yield from images of commercial fields. *Comput. Electron. Agric.* 171, 105307. doi: 10.1016/j.compag.2020.105307

Thorp, K. R., Thompson, A. L., and Bronson, K. F. (2020). Irrigation rate and timing effects on Arizona cotton yield, water productivity, and fiber quality. *Agric. Water Manage.* 234, 106146. doi: 10.1016/j.agwat.2020.106146

Tian, Z., Zhang, Y., Liu, K., Li, Z., Li, M., Zhang, H., et al. (2022). UAV remote sensing prediction method of winter wheat yield based on the fused features of crop and soil. *Remote Sens.* 14, 50545. doi: 10.3390/rs14195054

Torgbor, B. A., Rahman, M. M., Brinkhoff, J., Sinha, P., and Robson, A. (2023). Integrating remote sensing and weather variables for mango yield prediction using a machine learning approach. *Remote Sens.* 15, 30755. doi: 10.3390/rs15123075

Velumani, K., Lopez-Lozano, R., Madec, S., Guo, W., Gillet, J., Comar, A., et al. (2021). Estimates of maize plant density from UAV RGB images using faster-RCNN detection model: Impact of the spatial resolution. *Plant Phenomics*. doi: 10.34133/2021/9824843

Wan, S., Lin, S., Yuan, Q., and He, Z. (2024). A novel defect detection method for color printing fabrics based on attention mechanism and space-to-depth transformation. *Signal Image Video Processing*. doi: 10.1007/s11760-024-03146-9

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*. doi: 10.1109/CVPR52729.2023.00721

Wang, G., Chen, Y., An, P., Hong, H., Hu, J., and Huang, T. (2023). UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios. *Sensors* 23, 71905. doi: 10.3390/s23167190

Wang, L., Liu, Y., Wen, M., Li, M., Dong, Z., He, Z., et al. (2021). Using field hyperspectral data to predict cotton yield reduction after hail damage. *Comput. Electron. Agric.* 190, 106400. doi: 10.1016/j.compag.2021.106400

Wang, X., Lei, H., Li, J., Huo, Z., Zhang, Y., and Qu, Y. (2023). Estimating evapotranspiration and yield of wheat and maize croplands through a remote sensing-based model. *Agric. Water Manage.* 282, 108294. doi: 10.1016/j.agwat.2023.108294

Xiang, Y., Yao, J., Yang, Y., Yao, K., Wu, C., Yue, X., et al. (2023). Real-time detection algorithm for kiwifruit canker based on a lightweight and efficient generative adversarial network. *Plants* 12, 30535. doi: 10.3390/plants12173053

Xu, R., Li, C., Paterson, A. H., Jiang, Y., Sun, S., and Robertson, J. S. (2018). Aerial images and convolutional neural network for cotton bloom detection. *Front. Plant Sci.* 8. doi: 10.3389/fpls.2017.02235

Xu, W., Chen, P., Zhan, Y., Chen, S., Zhang, L., and Lan, Y. (2021). Cotton yield estimation model based on machine learning using time series UAV remote sensing data. *Int. J. Appl. Earth Observation Geoinformation* 104, 102511. doi: 10.1016/j.jag.2021.102511

Yan, P., Han, Q., Feng, Y., and Kang, S. (2022). Estimating LAI for cotton using multisource UAV data and a modified universal model. *Remote Sens.* 14, 42725. doi: 10.3390/rs14174272

Yang, S., Wang, W., Gao, S., and Deng, Z. (2023). Strawberry ripeness detection based on YOLOv8 algorithm fused with LW-Swin Transformer. *Comput. Electron. Agric.* 215, 108360. doi: 10.1016/j.compag.2023.108360

Yeom, J., Jung, J., Chang, A., Maeda, M., and Landivar, J. (2018). Automated open cotton boll detection for yield estimation using unmanned aircraft vehicle (UAV) data. *Remote Sens.* 10, 18955. doi: 10.3390/rs10121895

Zhang, Y., Wang, C., Wang, X., Zeng, W., and Liu, W. (2021). Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vision* 129, 3069–3087. doi: 10.1007/s11263-021-01513-4

Zhao, H., Zhang, H., and Zhao, Y. (2023). "Yolov7-sea: Object detection of maritime uav images based on improved yolov7," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. doi: 10.1109/WACVW58289.2023.00029

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. doi: 10.1109/CVPR.2016.319

Zou, M., Liu, Y., Fu, M., Li, C., Zhou, Z., Meng, H., et al. (2024). Combining spectral and texture feature of UAV image with plant height to improve LAI estimation of winter wheat at jointing stage. *Front. Plant Sci.* 14, 1272049. doi: 10.3389/fpls.2023.1272049

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 05 July 2024
DOI 10.3389/fpls.2024.1415006

OPEN ACCESS

EDITED BY
Mohsen Yoosefzadeh Najafabadi,
University of Guelph, Canada

REVIEWED BY
Orly Enrique Apolo-Apolo,
KU Leuven, Belgium
Rismen Sinambela,
Christian University of Indonesia, Indonesia
Raja Hamza,
University of Sfax, Tunisia

*CORRESPONDENCE
Lijia Xu
xulijia@sicau.edu.cn

RECEIVED 09 April 2024
ACCEPTED 18 June 2024
PUBLISHED 05 July 2024

CITATION
Tang Z, Xu L, Li H, Chen M, Shi X, Zhou L, Wang Y, Wu Z, Zhao Y, Ruan K, He Y, Ma W, Yang N, Luo L and Qiu Y (2024) YOLOC-tiny: a generalized lightweight real-time detection model for multiripeness fruits of large non-green-ripe citrus in unstructured environments.
Front. Plant Sci. 15:1415006.
doi: 10.3389/fpls.2024.1415006

YOLOC-tiny: a generalized lightweight real-time detection model for multiripeness fruits of large non-green-ripe citrus in unstructured environments

Zuoliang Tang[1,2], Lijia Xu[1]*, Haoyang Li[1], Mingyou Chen[3], Xiaoshi Shi[1,2], Long Zhou[1], Yuchao Wang[1], Zhijun Wu[1], Yongpeng Zhao[1], Kun Ruan[2], Yong He[4], Wei Ma[5], Ning Yang[6], Lufeng Luo[3] and Yunqiao Qiu[7]

[1]College of Mechanical and Electrical Engineering, Sichuan Agriculture University, Ya'an, China, [2]College of Resources, Sichuan Agriculture University, Chengdu, China, [3]School of Mechatronic Engineering and Automation, Foshan University, Foshan, China, [4]College of Biosystems Engineering and Food Science, Zhejiang University, Hangzhou, China, [5]Institute of Urban Agriculture, Chinese Academy of Agriculture Sciences, Chengdu, China, [6]School of Electrical and Information Engineering, Jiangsu University, Zhenjiang, China, [7]Sichuan Academy of Agricultural Machinery Sciences, Chengdu, China

This study addresses the challenges of low detection precision and limited generalization across various ripeness levels and varieties for large non-green-ripe citrus fruits in complex scenarios. We present a high-precision and lightweight model, YOLOC-tiny, built upon YOLOv7, which utilizes EfficientNet-B0 as the feature extraction backbone network. To augment sensing capabilities and improve detection accuracy, we embed a spatial and channel composite attention mechanism, the convolutional block attention module (CBAM), into the head's efficient aggregation network. Additionally, we introduce an adaptive and complete intersection over union regression loss function, designed by integrating the phenotypic features of large non-green-ripe citrus, to mitigate the impact of data noise and efficiently calculate detection loss. Finally, a layer-based adaptive magnitude pruning strategy is employed to further eliminate redundant connections and parameters in the model. Targeting three types of citrus widely planted in Sichuan Province—navel orange, Ehime Jelly orange, and Harumi tangerine—YOLOC-tiny achieves an impressive mean average precision (mAP) of 83.0%, surpassing most other state-of-the-art (SOTA) detectors in the same class. Compared with YOLOv7 and YOLOv8x, its mAP improved by 1.7% and 1.9%, respectively, with a parameter count of only 4.2M. In picking robot deployment applications, YOLOC-tiny attains an accuracy of 92.8% at a rate of 59 frames per second. This study provides a theoretical foundation and technical reference for upgrading and optimizing low-computing-power ground-based robots, such as those used for fruit picking and orchard inspection.

KEYWORDS

non-green-ripe citrus, multiripeness fruits, YOLOv7, EfficientNet, CBAM, agricultural robot

1 Introduction

Citrus is one of the most widely cultivated and highest-yielding fruit crops globally, generating significant economic value (The United States Department of Agriculture, 2024). However, the citrus industry faces immense pressure due to skilled labor shortages, rising production costs, market demand fluctuations, and extreme climate changes (Castro-Garcia et al., 2019; Apolo-Apolo et al., 2020a). Agricultural robots can mitigate these pressures by reducing reliance on skilled labor, lowering economic and environmental costs, and enhancing orchard management and productivity (Bargoti and Underwood, 2017; Fu et al., 2020a). Accurate fruit detection is essential for automated harvesting and early yield prediction (Zhuang et al., 2018; Apolo-Apolo et al., 2020b; Lu et al., 2023). Consequently, the detection of citrus fruits has become a research hotspot (Wang et al., 2022c; Ma et al., 2024). Particularly, there is an urgent need for high-performance detection models that can be deployed on resource-limited robots and other edge devices (Tang et al., 2020; Xu et al., 2023).

Multispectral cameras, optical digital cameras, 3D stereoscopic cameras, and RGB-D depth cameras are the primary devices used for fruit detection (Chen et al., 2020; Condotta et al., 2020). Multispectral cameras can capture spectral information across various bands from visible to near-infrared and are commonly mounted on unmanned aerial vehicles for large-scale crop health, yield estimation, and disease monitoring (Huang et al., 2020; Lan et al., 2020). However, their high cost and complex data processing requirements limit their application in ground-based agricultural robots. Optical digital cameras, 3D stereoscopic cameras, and RGB-D depth cameras typically produce RGB images with three visible light bands: red, green, and blue. Many studies have shown that RGB images are sufficient for fruit detection (Lu et al., 2018; Yu et al., 2019; Gené-Mola et al., 2020; Liu et al., 2023). These devices are cost-effective and require less computational power, making them more suitable for the practical needs of real-time monitoring and automated harvesting robots.

Over the past few decades, methods combining digital image processing with traditional machine learning (ML) techniques have been used for fruit detection, including citrus (Liu et al., 2018), kiwifruit (Fu et al., 2019), and apples (Lu et al., 2022). However, the pixel values in RGB images are highly sensitive to changes in lighting and background interference. Traditional ML algorithms, such as support vector machines and decision trees, rely on complex feature extraction and manual rules to handle these variations (Fu et al., 2018). Consequently, these algorithms exhibit performance fluctuations in complex environments and fail to meet the need for stable citrus fruit detection by robots in real-world scenarios.

In recent years, the advancement of deep learning (DL) technology has significantly impacted the field of agricultural detection due to its exceptional feature learning capability, robust generalization performance, and substantial computational power (Gené-Mola et al., 2020; Maheswari et al., 2021). DL methods for fruit detection are broadly categorized into two main approaches: region-based two-stage methods (Girshick et al., 2014; Shaoqing et al., 2016) and end-to-end single-stage methods (Redmon et al., 2016; Wei et al., 2016).

Two-stage detection methods first extract a large number of regions of interest (RoIs) that potentially contain target fruits. These RoIs are then passed through a convolutional neural network (CNN) for detection, with final detection results obtained after post-processing (Girshick et al., 2014; Shaoqing et al., 2016). Although this process is time-consuming, these methods typically achieve high detection precision due to the utilization of CNNs for fruit detection on RoIs (Redmon et al., 2016). For example, C.H. Yang et al. (Yang et al., 2020) developed a citrus fruit detection algorithm based on Mask R-CNN, achieving a detection precision of 88.15%. Longsheng Fu et al. (Fu et al., 2020b) proposed an apple detection algorithm based on Faster R-CNN, achieving a detection precision of 89.3%. However, the inherent characteristics of two-stage methods, including slower detection speed and high memory consumption, limit their suitability for applications such as harvesting robots, which require rapid detection and are constrained by computational resources.

In contrast, the YOLO series of single-stage detection methods, introduced in 2015, offers faster detection speeds and high detection accuracy (Redmon et al., 2016). YOLO models perform target detection in a single pass through a CNN, eliminating the need for separate stages and reducing redundant operations (Redmon et al., 2016; Wang et al., 2022a). While early YOLO models had lower detection accuracy compared to two-stage models like R-CNN, subsequent optimizations, and improvements by numerous researchers have led to the development of several highly effective fruit detection methods. For instance, Longsheng Fu et al. (Fu et al., 2021) proposed a kiwifruit detection algorithm, DY3TNet, by improving the YOLOv3-tiny model, achieving a detection precision of 90.05%. Shenglian Lu et al. (Lu et al., 2022) developed the CA-YOLOv4 detection algorithm for apples in orchard environments, achieving a detection precision of 92.6% for Envy apples during harvest. Additionally, Lijia Xu et al. (Xu et al., 2023) proposed the HPL-YOLOv4 citrus detection model for complex environments, achieving a detection precision of 93.45%.

Citrus is a general term for fruits belonging to the Citrus genus of the Rutaceae family, with major types including grapefruit, lemon, tangerine, and orange (Liu et al., 2012). Among these, navel oranges, Ehime Jelly oranges, and Harumi tangerines are widely cultivated in the southwestern regions of China, and their fruits turn orange-red upon ripening. In this study, we refer to them as non-green-ripe citrus. While existing models can detect single-variety or single-degree ripeness fruits, such as apples or certain citrus fruits (Lu et al., 2018, 2022), there remains an urgent need for a real-time and accurate detection model for multi-ripeness fruits of different non-green-ripe citrus varieties in complex orchards. To address this issue, we first collected and created a custom image dataset of non-green-ripe citrus, covering the detection needs of unripe, semi-ripe, and ripe fruits. We then proposed a lightweight, single-stage citrus detection model suitable for deployment on edge devices such as robots. The main contributions of this work are as follows:

(1) We designed a comprehensive image dataset, RC3025, which includes images of non-green-ripe citrus fruits of various varieties and ripeness levels in complex scenarios.

(2) We discovered and proposed that incorporating a small number of pure citrus fruit images into the training set enhances the model's detection performance in real orchards.

(3) We developed a general, lightweight, and high-performance multi-ripeness citrus recognition model, YOLOC-tiny, based on YOLOv7.

(4) We validated the practical performance and advantages of YOLOC-tiny through robot deployment application experiments, demonstrating its effectiveness in detecting non-green-ripe citrus fruits in complex environments.

2 Materials and methods

2.1 Multiripeness non-green-ripe citrus fruit image dataset

2.1.1 Raw image acquisition and labeling

To properly train the developed DL model, a number of raw images are required as an initial dataset (Apolo-Apolo et al., 2020a). Research suggests that 2,500 annotated instances are adequate for training deep networks to recognize a certain type of fruit (Wang et al., 2022b). From 2020 to late 2023, we continuously collected raw images over four years using both manual and robotic photography, as shown in Figure 1. Various imaging devices, including a 3D stereoscopic camera (ZED), a Canon 80D camera, and four different mobile phones (VIVO Y97, Mi 10, Redmi K40, and iPhone Xs), were employed to capture images of citrus fruits at different ripeness levels and varieties in three non-green-ripe citrus orchards. These orchards are located in three different counties in the western part of Sichuan Province: Yucheng District, Ya'an City (29°58′N, 102°59′E); Jintang County, Chengdu City (30°43′N, 104°29′E); and Danling County, Meishan City (29°58′N, 103°32′E). To meet the operational needs of ground-based agricultural robots, the shooting distance ranged from 0.3 to 1.2 meters.

Experienced researchers screened the raw images and collected a total of 2,905 image data samples covering three citrus varieties (navel orange, Ehime Jelly orange, and Harumi tangerine) at different ripeness levels in unstructured orchards. Additionally, to investigate whether pure citrus images could enhance model detection performance, 120 images featuring pure citrus fruits were captured in the laboratory using both ZED and digital cameras. Specifically, the sections pertaining to citrus images in complex orchards and pure citrus images were separately labeled as RC2905 and RC120. The labeling process for the 3,025 images was completed using the open-source tool LabelImg (Tzutalin, 2015), and the citrus image dataset RC3025 (raw citrus dataset with 3,025 images) was created, comprising a total of 10,653 labeled instances. Details of the dataset are shown in Supplementary Table 1.

2.1.2 Dataset partitioning

Several studies have successfully used a 10% validation split (Lu et al., 2022; Liu et al., 2023; Xu et al., 2023), achieving significant detection results. To balance computational resources and maintain training efficiency, the RC2905 dataset was partitioned into the raw training set (TRAIN-R), the raw validation set (VAL-R), and the raw test set (TEST-R) in an 8:1:1 ratio, as illustrated in Figure 2. The RC120 dataset was employed to investigate the impact of pure citrus images on model performance by randomly substituting 120 images in TRAIN-R, defining the refined training set as FTRAIN-R after fine-tuning. TEST-R was further categorized based on background complexity and citrus occlusion, resulting in a complex background test set comprising 166 images (TEST-RCE) and a simple background test set with 124 images (TEST-RSE). Given the variations in light intensity, the test set was further stratified into a set containing 228 images with normal light intensity (TEST-RNL) and another set containing 62 images under low-light conditions (TEST-RWL).

2.1.3 Image dataset augmentation

Many studies demonstrated that enhancing raw images can improve the model's generalization ability. In the present study, seven enhancement methods, including affine transformation,

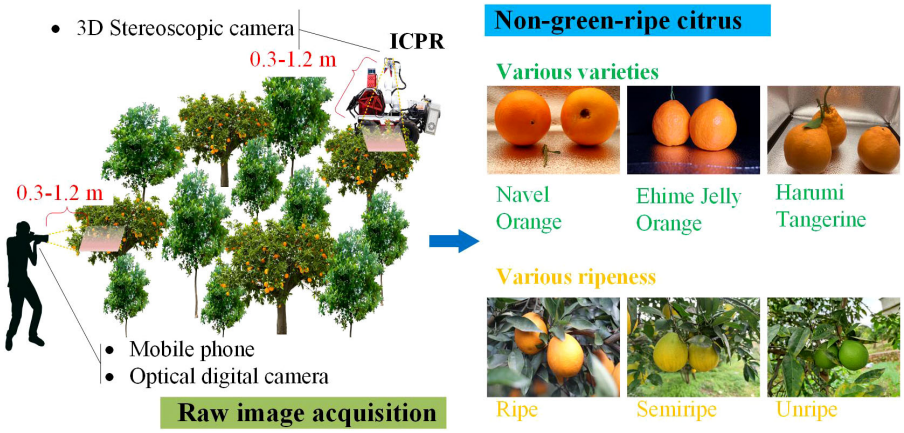


FIGURE 1
Schematic diagram of how to capture the images of non-green-ripe citrus fruits and some examples.

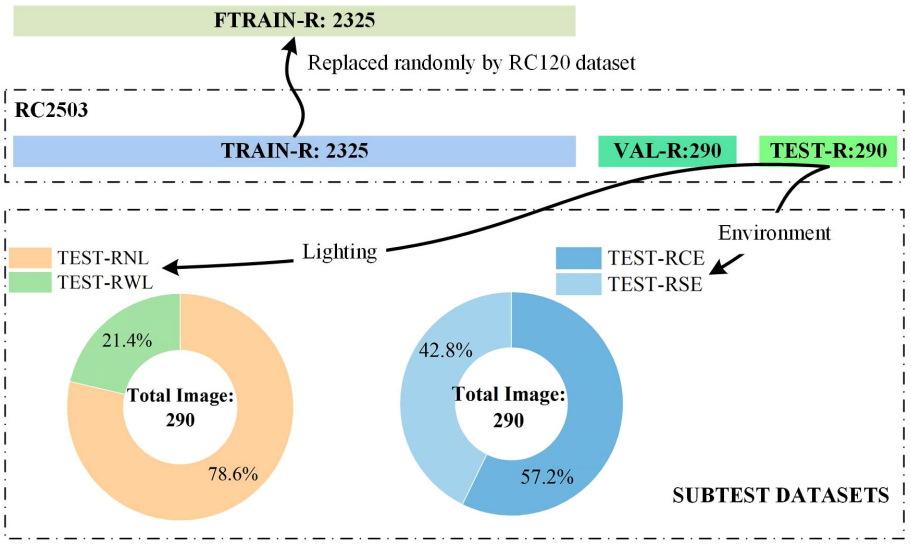


FIGURE 2
Diagram of dataset partitions.

luminance adjustment, cut-out, coarse dropout, Gaussian noise, motion blur, and salt–pepper noise, were employed to augment the training and validation sets, as illustrated in Figure 3. These enhancement operations were executed on TRAIN-R, VAL-R, TEST-R and FTRAIN-R, resulting in the corresponding enhanced datasets TRAIN-A, VAL-A, TEST-A and FTRAIN-A, respectively. Three enhancement methods—up and down flip, contrast adjustment, and Gaussian blur—were simultaneously applied to the test set. Two datasets, TEST-ANL and TEST-AWL, were generated by enhancing the original test set of normal and weak light environments. Additionally, the TEST-ACE and TEST-ASE datasets were created by augmenting the original test sets of complex and simple environments, respectively. These enhanced datasets aim to simulate accurately the diverse lighting conditions, backgrounds, and fruit states in real-life orchard scenes, thereby bolstering the robustness and accuracy of the detection models in practical scenarios. An overview of the augmented dataset and the number of images is provided in Supplementary Table 2.

2.2 Design of the YOLOC-tiny model

Orchard operation robots face constraints due to limited computational resources, whereas traditional DL models pose challenges with their high computational complexity and demanding hardware requirements. To ensure that robots can reliably, accurately, and efficiently detect various types and ripeness levels of citrus fruits in complex non-green-ripe citrus orchards, we initially used our custom datasets TRAIN-A, VAL-A, and TEST-A to train and test most of the popular SOTA object detectors, including YOLOv7 and YOLOv8. Based on the practical needs of robotic operations and the experiment results, we chose YOLOv7 (Wang et al., 2022a) as the foundational network and conducted a series of optimizations and improvements.

Given the large size of the YOLOv7 backbone network, we selected a lightweight feature extraction network, EfficientNet-B0, to replace the original backbone. After comparing various advanced attention mechanisms, we introduced a composite efficient attention mechanism, CBAM, to enhance target perception. Subsequently, after numerous experiments, we carefully designed an extended efficient aggregation network module incorporating CBAM, called Efficient Layer Aggregation Networks in the Head with CBAM (ELAN-HC). Considering the phenotypic features of the targets, we proposed an adaptive and efficient complete intersection over the union regression loss function (ACIoU). This function allows for adjustments to the aspect ratio regression loss penalty factor, enhancing the perception ability of citrus fruits and consequently improving detection accuracy.

We integrated these measures to develop a generalized base network, YOLOC, where C stands for citrus, to recognize non-green-ripe citrus varieties such as navel orange, Ehime jelly orange, and Harumi tangerine in complex environments, particularly in the hilly areas of southwest China. The structure of YOLOC is depicted in Figure 4, where RepConv denotes reparametrized convolution. Furthermore, we leveraged transfer learning to train YOLOC on FTRAIN-A and employed sparse training and Layer-based Adaptive Magnitude Pruning (LAMP), a quantized pruning technique, to derive a lightweight recognition model, YOLOC-tiny.

2.2.1 ACIoU

The accuracy of target detection and localization is significantly influenced by the choice of the loss function (Yu et al., 2022). The loss function was computed based on the intersection over union (IoU), and the CIoU utilized by YOLOv7 comprehensively considered the variations in the overlap area, center distance, and aspect ratio between the predicted box and the ground truth box (Zheng et al., 2020), as illustrated in (Equations 1–3).

10.3389/fpls.2024.1415006

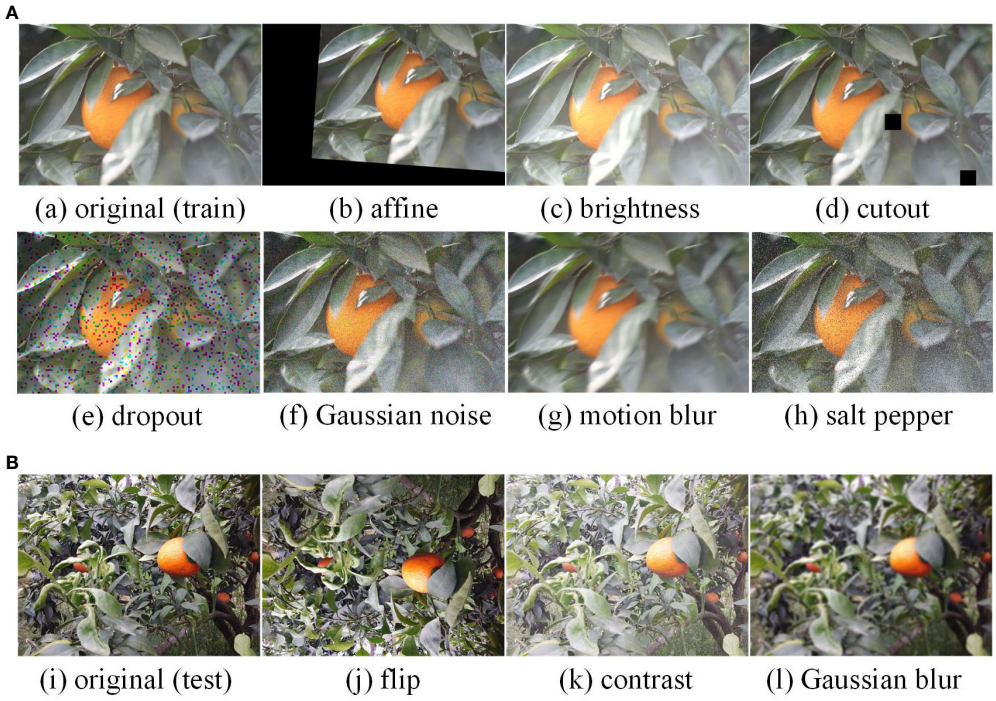


FIGURE 3
Image augmentations. **(A)** Augmentation methods for TRAIN-R, FTRAIN-R, and VAL-R: affine transformation, brightness adjustment, cutout, coarse dropout, Gaussian noise, motion blur, and salt and pepper noise. **(B)** Augmentation methods for TEST-R: up and down flip, contrast adjustment, and Gaussian blur.

$$Loss_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v, \tag{1}$$

$$\alpha = \frac{v}{1 - IoU + v}, \tag{2}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2, \tag{3}$$

where $Loss_{CIoU}$ denotes the loss value, IoU represents the IoU ratio between the ground truth box and the predicted box, $\rho^2(b, b^{gt})$ signifies the Euclidean pixel distance between the ground truth box and the predicted, c represents the diagonal length of the smallest enclosing area that surrounds both the predicted and ground truth bounding boxes, α is the acquired trade-off coefficient, v denotes the consistency factor of the width and height of the predicted box and the ground truth box, w^{gt} and h^{gt} are the width and height of the ground truth box, respectively, and w and h are the width and height of the predicted box, respectively.

The CIoU loss function is commonly employed in target detection tasks; however, it exhibits the following drawbacks (Yu et al., 2022): (1) The use of an inverse tangent function in CIoU makes it highly sensitive to outliers, resulting in poor robustness. (2) The value domain $(0, \pi/2)$ of the inverse tangent function cannot directly fulfill the normalization requirements of the loss function. (3) Adaptability to adjust the corresponding features of the loss function based on the detection object is lacking. Hence, considering the phenotypic features of the large non-green-ripe citrus fruits, we proposed ACIoU. This function can dynamically adjust the length and width regression loss penalty factor based on the phenotypic parameters of citrus fruits, as depicted in (Equations 4–6).

$$Loss_{ACIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha\gamma, \tag{4}$$

$$s(a, b, x) = \frac{1}{1 + e^{-a(x-b)}}, \tag{5}$$

$$\gamma = \left(s\left(a, b, \frac{w^{gt}}{h^{gt}}\right) - s\left(a, b, \frac{w}{h}\right)\right)^2, \tag{6}$$

where $Loss_{ACIoU}$ represents the value of the ACIoU function, a and denotes the adaptive Sigmoid deformation parameters that can be adjusted based on different aspect ratios of the detection targets, and γ signifies the adaptive consistency factor of the width and height of the predicted box and the ground truth box.

The variation curves of the width and height difference loss penalty terms corresponding to the real and predicted boxes for different deformation parameters, a and b, are presented in Figure 5. The disparity between the length and width of the ground truth box of citrus fruits is smaller than that in Microsoft Common Objects in Context (COCO). We randomly selected 47 citrus fruits with different maturity levels from the orchard of Ya'an Yucheng District, and their average transverse and longitudinal diameters were measured using vernier calipers. The transverse diameter

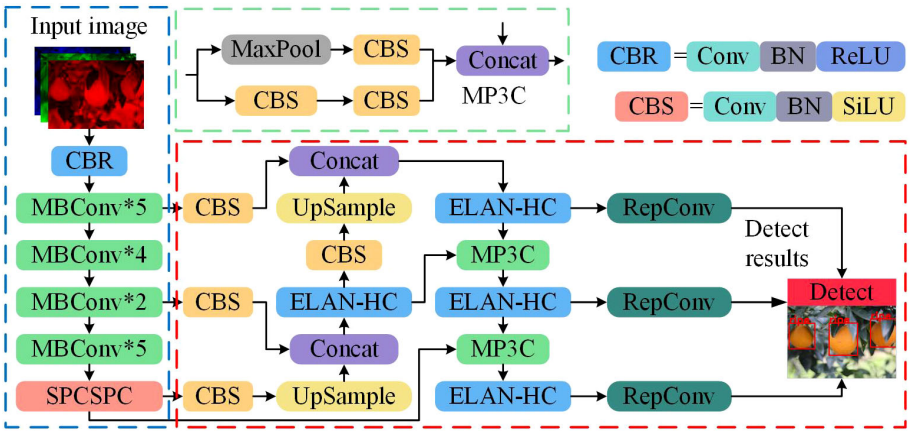


FIGURE 4
Structure of the YOLOC network.

represents the maximum equatorial diameter of the mandarin orange. The longitudinal diameter is the straight-line distance between the pith (at the stalk) and the center of the top of the fruit, as shown in Figure 6. The measured average longitudinal diameter of citrus was 68.04 mm, the average transverse diameter was 64.94 mm, and the average aspect ratio was 1.05. These measurements can serve as a reference for adaptively adjusting the loss function width and high consistency evaluation index.

2.2.2 Efficient feature extraction backbone

This study utilized EfficientNet-B0 as the feature extraction backbone to optimize the model parameters for practical deployment in orchard robots for recognizing large non-green-ripe citrus. EfficientNet-B0, a lightweight and high-performance neural network, was designed using neural architecture search. The architecture primarily consists of mobile inverted bottleneck convolutions (MBConv), as illustrated in Supplementary Figure 1. MBConv integrates depthwise separable convolutions (DWConv) with Squeeze-and-Excitation (SE) blocks and inverse residual blocks. The SE module within MBConv dynamically recalibrates channel-wise feature responses by explicitly modeling interdependencies between the channels. With its DWConv and SE modules, MBConv offers a lightweight structure while maintaining good detection performance.

2.2.3 ELAN-HC

The spatial attention mechanism amplifies the model's capability to concentrate on specific regions within the image, facilitating the extraction of features crucial for target detection. The channel attention mechanism guides the model to prioritize significant features in the image, thereby contributing to an overall enhancement in target detection accuracy. CBAM integrates the channel attention mechanism and the spatial attention mechanism. YOLOv7 introduces efficient layer aggregation networks in the detection head (ELAN-H), leading to significant performance improvements. Empirically drawing on engineering experience, we incorporated CBAM into the ELAN-H network module, resulting in the formation of the ELAN-HC module, as depicted in Figure 7. This integration is aimed at optimizing further the model's detection performance for non-green-ripe citrus in the unstructured orchards.

2.2.4 Lightweight pruning strategy

We employed the LAMP pruning method on the trained YOLOC model to eliminate redundant parameters and connections, thereby enhancing the deployable performance and detection efficiency of YOLOC-tiny (Supplementary Figure 2). Subsequently, the pruned model underwent retraining in FTRAIN-A, resulting in the development of a lightweight detection model, YOLOC-tiny. The calculation for the LAMP

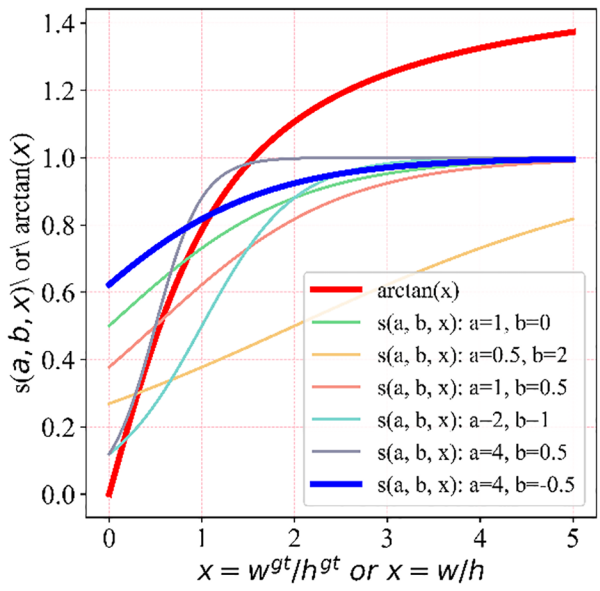


FIGURE 5
Loss penalty curves of the width and height differences with different deformation parameters.

FIGURE 6
Measurement methods of longitudinal and transverse diameters and samples. **(A)** Measurement method of citrus fruit longitudinal diameter. **(B)** Measurement method of citrus fruit transverse diameter. **(C)** Samples of citrus fruits.

score is expressed in Equation 7. The LAMP score was used to measure the importance of all weights in each layer of the YOLOC network to the citrus detection performance. During each round of pruning iterations, we removed the weights that contributed the least to the detection performance until the global sparsity constraint was satisfied. Thus, the model size was compressed, with little impact on its detection accuracy. LAMP retained at least one connection in each layer to ensure that at least one surviving connection was retained in each layer, thereby avoiding the loss of neurons and helping to maintain the ability to perceive non-green-ripe large citrus.

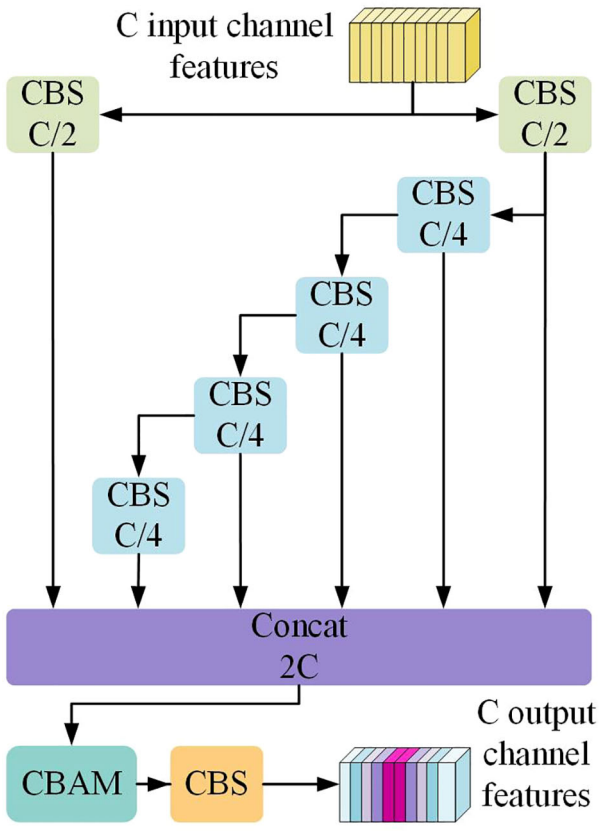


FIGURE 7
The structure of ELAN-HC.

$$score(u_i) = \frac{u_i^2}{\sum_{i \geq j} u_j^2}, \tag{7}$$

where u_i is the ith weight magnitude in the kth ($k = 0, \cdots 359$) layer of the YOLOC network after ascending order, and $score(u_i)$ is the LAMP score of u_i.

3 Experiments and results

In this study, three computers, namely, PC1, PC2 and PC3, were employed for model training, testing, and deploying applications, respectively. PC3 is a lightweight industrial control mainframe computer integrated into the self-developed intelligent citrus picking robot (ICPR), as shown in Figure 1. The detailed hardware and software configurations of the three computers are provided in Table 1.

3.1 Model training and performance evaluation metrics

The model training was performed on PC1 and initialized with pre-trained weights from the COCO dataset. The stochastic gradient descent algorithm was used as the optimizer for model training. The training parameters included an initial learning rate of 0.01, momentum decay of 0.937, weight decay of 0.0005, a model input image size of 640 × 640, and a training epoch count of 300. A label smoothing strategy was implemented to address potential network overfitting resulting from incorrect data labeling by improving the model's generalization ability. Additionally, online data augmentation using the mosaic method at each iteration was employed to enrich the citrus image data and further enhance the model's generalization ability.

The model evaluation tests were conducted on PC2. The batch size for model test inputs was set to 1, the confidence threshold was 0.001, the IOU threshold was 0.6, and the model input image size was 640 × 640 by aligning with the practical conditions of the orchard robot. Given the constraints of the robot's limited hardware resources, the models were comprehensively evaluated in this study based on three aspects, namely, basic detection performance, degree of lightweight, and detection speed, to assess the detection

10.3389/fpls.2024.1415006

TABLE 1 Key hardware and software configurations of the experimental environment.

Hardware/ Software	PC1	PC2	PC3
CPU	Intel(R) Core (TM) i9-10920X CPU @ 3.50 GHz	Intel(R) Core (TM) i9-10920X CPU @ 3.50 GHz	Intel(R) Core (TM) i7-1165G7 CPU @ 2.80 GHz
GPUs	NVIDIA GeForce RTX 3090 (24576 MB) × 2	NVIDIA GeForce RTX 3090 (24576MB) × 2	NVIDIA GeForce MX450 (2048 MB)× 1
RAM	32 GB 3200 MHz × 4	32 GB 3200 MHz × 4	16 GB 3200 MHz × 1
Motherboard	ASUS WS X299 SAGE	ASUS WS X299 SAGE	HP 87E2
Operating system	Microsoft Windows 10 Pro (64-bit)	Microsoft Windows 10 Pro (64-bit)	Microsoft Windows 10 Pro (64-bit)
CUDA	11.7	11.8	11.8
cuDNN	8.5.0	8.7.0	8.7.0
PyTorch	2.0.0	2.0.1	2.0.1
OpenCV	4.7.0	4.8.0	4.8.0
Python	3.8.16	3.8.17	3.9.18
VS code	1.83.1	1.84.1	1.84.1

performance of different models. The evaluation of basic detection performance includes detection precision (P), recall (R), and mean average precision (mAP), which were calculated according to (Equations 8–10).

$$P = \frac{TP}{TP + FP} \times 100\,\%, \tag{8}$$

$$R = \frac{TP}{TP + FN} \times 100\,\%, \tag{9}$$

$$mAP = \frac{1}{k}\sum_{i=1}^{k}\int_{0}^{1} P_i(R_i)d(R_i) \times 100\,\%, \tag{10}$$

where TP (true positive) represents the count of accurately detected citrus fruits, FP (false positive) signifies the count of erroneously identified objects or backgrounds as citrus fruits, FN (false negative) corresponds to the count of undetected or inaccurately identified citrus fruits, and k denotes the specific fruit type to be detected. In this study, k is 3, indicating the three categories of ripe, semi-ripe, and unripe citrus fruits.

The evaluation metrics for lightweight degree encompass the memory size occupied by the model (model size), the number of parameters (params), and the model detection speed measured by the number of FPS. Additionally, we introduced four normalized evaluation indicators, including the compound evaluator (CEval), which provides a holistic assessment of the model considering basic performance, the degree of lightweight, and detection speed. The CEval, model size score, model parameter score, and frame rate score are calculated as depicted in (Equations 11–14).

$$CEval = \alpha_1 P + \alpha_2 mAP + \alpha_3 SizeScore + \alpha_4 \text{ParamsScore} + \alpha_5 FPSScore, \tag{11}$$

$$SizeScore = \frac{1}{1 + \exp\left(0.1 \times (ModelSize - t_1)\right)}, \tag{12}$$

$$ParamsScore = \frac{1}{1 + \exp\left(0.1 \times (Params - t_2)\right)}, \tag{13}$$

$$FPSScore = \frac{1}{1 + \exp\left(-(FPS - t_3)\right)}, \tag{14}$$

where α_1, α_2, α_3, α_4, and α_5 are the weight coefficients, and their sum is 1.0. They differentiate the importance of various indicators for intelligent operation robots in orchards. t_1, t_2 and t_2 control the thresholds for each evaluation indicator.

The curves illustrating the model size score, model parameter score, and frame rate score are presented in Figure 8. Slight variations in the FPS of each model were observed across experiments; thus, the FPS rates of all models were averaged over five tests after completing the graphics card warm-up. Aligned with the real-time target detection task and the goal of maintaining lightweight models, the threshold values (t_1, t_2, and t_3) in this study were set at 50, 50, and 30, respectively. A high frame rate score indicates proximity to 1. However, the frame rate beyond the robot's real-time monitoring need of 30 FPS becomes barely crucial. Conversely, parameter and model size scores approach 1 as they decrease and approach 0 as they increase.

3.2 Comparative experiments of different attention mechanisms

YOLOv7 was employed as a baseline to elucidate the impact of attention mechanisms on the detection performance of the YOLOC

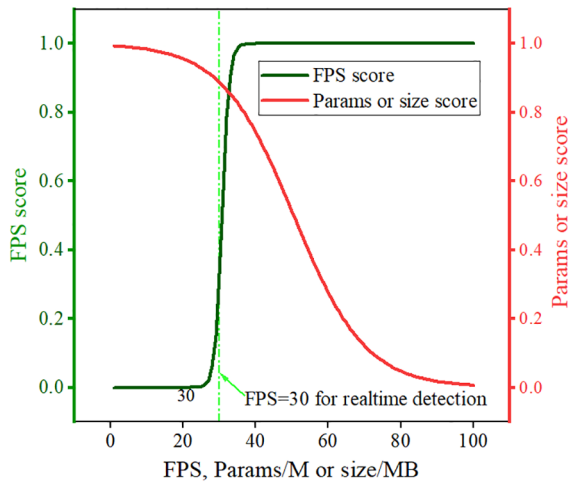


FIGURE 8
Evaluation metric score curves of model performance.

TABLE 2 Experimental results of incorporating various attention mechanisms.

Model	P/%	mAP/%	Size/MB	Params/M	FPS
YOLOv7	82.6	81.3	71.3	36.5	91
YOLOv7 +CA	84.3	81.9	72.5	37.1	86
YOLOv7 +ECA	83.5	82.2	71.3	36.5	90
YOLOv7 +SE	80.8	82.1	72.8	37.3	91
YOLOv7 +SimAM	83.6	82.0	71.3	36.5	89
YOLOv7 +CBAM	83.8	82.5	72.9	37.3	86

All attention mechanisms were implemented in the same position within ELAN-H. CA refers to coordinate attention, ECA refers to efficient channel attention, and SimAM refers to a simple and effective attention module.

model. Various attention mechanisms were incorporated into the ELAN-H module, and comparative experiments were conducted to assess the detection performance for non-green-ripe large citrus. The results are presented in Table 2.

Table 2 reveals that the introduction of different attention mechanisms impacts the model's detection performance to varying degrees. Regarding performance metrics, YOLOv7 +CBAM exhibits a detection accuracy of 83.8%, marking a 1.2 percentage point improvement over YOLOv7. Thus, it only ranks second to the YOLOv7+CA model. Compared with the average accuracy of YOLOv7, that of YOLOv7+CBAM reaches 82.5%, indicating a 1.2 percentage point increase, whereas that of YOLOv7+CA is only 81.9%. This finding suggests that YOLOv7 +CBAM excels in capturing citrus image features. Although the model size and number of parameters of YOLOv7+CBAM experience a slight increase compared with those of YOLOv7, its detection accuracy is enhanced. The frame rate of YOLOv7+CBAM is 86 FPS, satisfying the real-time target detection requirements. We employed the GradCAM algorithm to generate detection heat maps for multiripeness citrus images and gain deep insights into the suitability of the CBAM attention mechanism in citrus fruit detection. The corresponding detection results are presented in Figure 9. All heat maps were generated at the same layer above the detection head of the detect network layer of the model.

Figure 9 shows that different attention mechanisms allocate varying degrees of focus to citrus fruits, leading to differences in the detection performance of fruits at various ripeness levels. YOLOv7 +CBAM exhibits the highest attention to citrus fruits with diverse ripeness, surpassing the attention given by YOLOv7, which allocates minimal attention to citrus fruits. For green unripe citrus, YOLOv7 distributes attention across the surroundings evenly. Despite the improvement in the model's attention to citrus fruits with the introduction of other attention mechanisms, it still falls short of the performance achieved by the YOLOv7+CBAM model.

In terms of detection results, YOLOv7+CBAM and YOLOv7 exhibit no misdetections or omissions. By contrast, YOLOv7+CA has one omission and one misdetection in ripe citrus detection, YOLOv7+ECA has two omissions, and YOLOv7+SE and YOLOv7 +SimAM have one omission in ripe citrus detection. In summary, the CBAM attention mechanism demonstrates superior performance in detecting multiripeness citrus fruits, particularly for unripe green citrus. Thus, it maintains high-precision results with no false or missed detections. Therefore, the CBAM attention mechanism proves to be well-suited for citrus fruit detection.

3.3 Ablation experiments

We conducted ablation experiments to assess comprehensively the impact of various enhancement measures on the model's detection performance by incrementally introducing these measures with YOLOv7 as the baseline. The experimental results are presented in Table 3.

Table 3 reveals that all proposed enhancements in this study lead to varying degrees of improvement in the model's detection accuracy or lightweight characteristics. Compared with the training set without the use of the fine-tuned training set, the YOLOv7 model with the fine-tuned training set FTRAIN-A exhibits a 2.4% improvement in P and a 0.9% improvement in mAP. Furthermore, incorporating the lightweight backbone network EfficientNet-B0 further enhances the model's detection accuracy and increases its lightweight profile. Compared with YOLOv7, the model with the EfficientNet-B0 backbone shows a 2.9% increase in P and a 0.8% increase in mAP value whilst maintaining only 32.5% and 32.1% of the model size and number of parameters of YOLOv7, respectively.

The introduction of the ELAN-HC module with the CBAM attention mechanism leads to a slight increase in model size and a decrease in frame rate. However, the P and mAP of the model show significant improvements, reaching 85.7% and 83.1%, respectively, representing a 3.1% and 1.8% increase compared with those of YOLOv7. Given these improvements, the YOLOC model with ACIoU experiences a marginal decrease in detection accuracy by 0.5% but improves in mAP and frame rate by 0.4% and 6 FPS, respectively.

The YOLOC-tiny model, derived through pruning and retraining on top of YOLOC, excels not only in accuracy but also in achieving an extremely compact model size. In particular, the P and mAP of the model are 85.3% and 83.0%, respectively, representing 2.7% and 1.7% increases compared with those of YOLOv7. The model size of YOLOC-tiny is 8.4 MB, with only 11.8% and 11.5% of the model size and number of parameters of YOLOv7, respectively.

3.4 Comparison experiments of different detectors

YOLOC-tiny was compared with the leading SOTA target detection models. The experimental results are presented in Table 4.

Table 4 reveals that YOLOC-tiny achieves an accuracy of 85.3% in detecting multiripeness citrus fruits. This finding indicates that

10.3389/fpls.2024.1415006

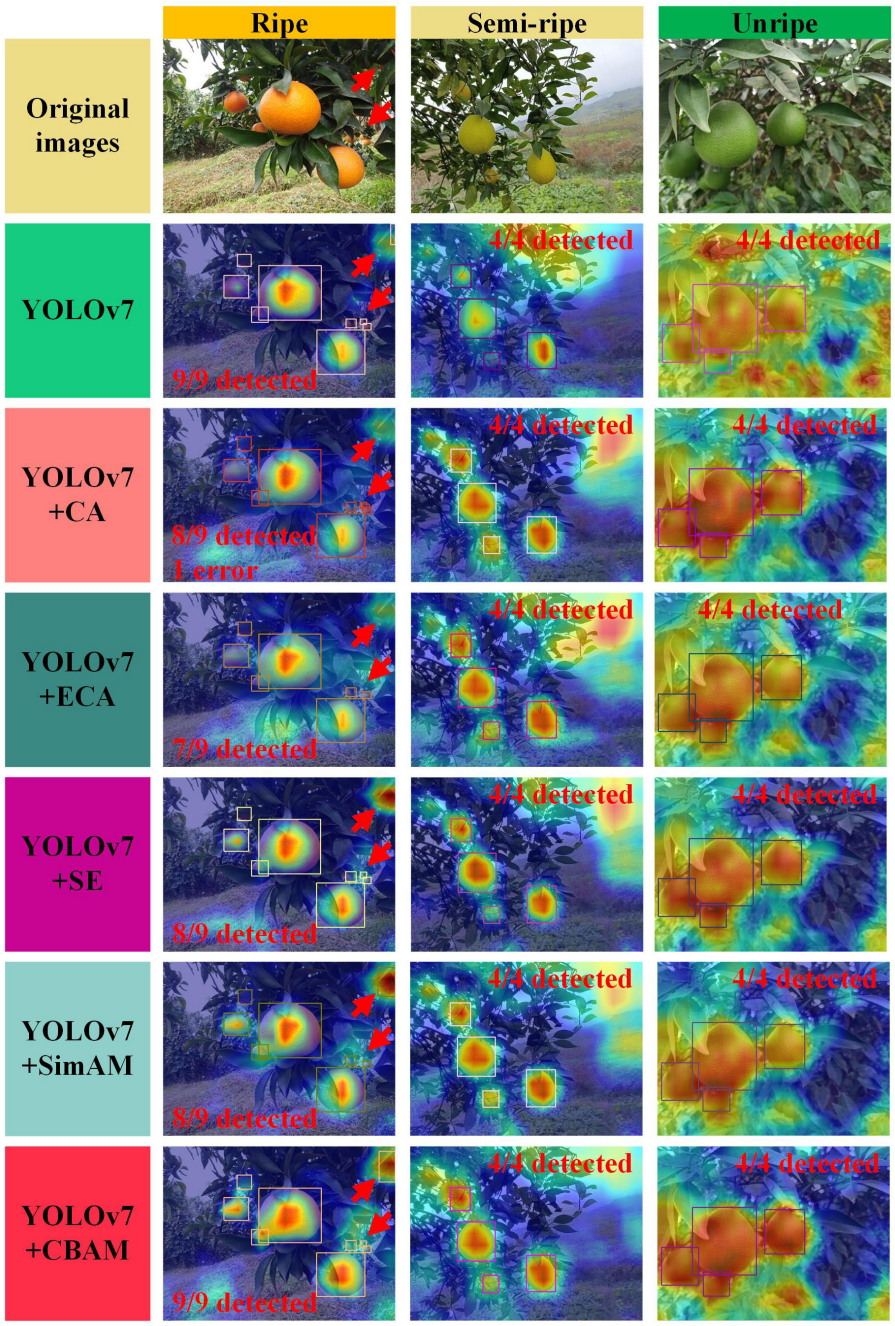


FIGURE 9
Thermograms and detection results of models integrated with various attention mechanisms. The red arrows indicate the locations where the detection results of different detectors are significantly different.

YOLOC-tiny outperforms most SOTA models and even surpasses YOLOC. Additionally, YOLOC-tiny attains an 83.0% mAP, ranking only second to YOLOC. YOLOC-tiny occupies a mere 8.4 MB of storage space, making it significantly more lightweight than YOLOv7x and YOLOv8x. The model's parameter count is only 4.2 M, rendering it suitable for deployment in edge devices and resource-limited environments. Furthermore, YOLOC-tiny achieves a frame rate of 80 FPS, surpassing YOLOv8l and YOLOv8x. Thus, it is well-suited for real-time performance-critical scenarios.

We utilized the previously mentioned model lightweight, frame rate, and comprehensive performance indexes to analyze the detection performance of YOLOC series models comprehensively and thoroughly in complex environments for multiripeness and multispecies citrus fruits. The comprehensive performance diagrams of the aforementioned models were plotted, as shown in Figure 10. Figure 10 shows that YOLOC and YOLOC-tiny exhibit excellent detection performance for citrus fruits. YOLOC and YOLOC-tiny demonstrate commendable average detection accuracies, with YOLOC-tiny being more compact than other

TABLE 3 Results of ablation experiments.

Model	P (%)	mAP (%)	Size (MB)	Params (M)	FPS
YOLOv7	82.6	81.3	71.3	36.5	91
YOLOv7+FTRAIN-A	85.0	82.2	71.3	36.5	90
YOLOv7+FTRAIN-A+EfficientNet-B0	85.5	82.1	23.2	11.7	88
YOLOv7+FTRAIN-A+EfficientNet-B0+CBAM	85.7	83.1	23.8	12.0	81
YOLOv7+FTRAIN-A+EfficientNet-B0+CBAM+ACIoU +ACIoU (YOLOC)	85.2	83.5	23.8	12.0	87
YOLOv7+FTRAIN-A+EfficientNet-B0+CBAM+ACIoU +LAMP+ACIoU+LAMP (YOLOC-tiny)	85.3	83.0	8.4	4.2	80

models. This compactness contributes to reduced storage requirements on edge devices. YOLOC-tiny outperforms all other SOTA models, including YOLOC, in terms of the total score. Therefore, YOLOC-tiny has significant advantages in various aspects, including detection accuracy, lightweight design, frame rate, and overall performance. It exhibits the strongest overall performance, making it highly suitable for target detection in citrus orchard scenarios.

TABLE 4 Experimental results of different SOTA models.

Model	P/%	mAP/ %	Size/ MB	Params/ M	FPS	Total Score
YOLOv5n	82.3	80.0	3.6	1.8	97	4.60
YOLOv5s	85.6	79.7	13.6	7.0	91	4.61
YOLOv5m	83.4	80.0	40.1	20.9	92	4.31
YOLOv5l	83.4	78.9	88.4	46.1	84	3.24
YOLOv5x	86.0	79.8	165.0	86.2	70	2.67
YOLOv6n	85.4	81.1	10.0	4.6	24	3.99
YOLOv6s	85.0	81.2	38.7	18.5	20	3.65
YOLOv6m	85.1	82.8	72.5	34.8	23	2.93
YOLOv6l	83.1	83.0	114.0	59.5	22	2.37
YOLOv7-tiny	80.6	82.1	11.6	6.0	101	4.59
YOLOv7	82.6	81.3	71.3	36.5	91	3.54
YOLOv7x	84.6	81.8	135.0	70.8	86	2.77
YOLOv8n	81.8	81.2	5.9	3.0	114	4.61
YOLOv8s	85.4	81.5	21.4	11.1	117	4.59
YOLOv8m	84.3	81.8	49.5	25.8	103	4.09
YOLOv8l	85.0	81.8	83.5	43.6	79	3.35
YOLOv8x	86.0	81.1	130.0	68.1	62	2.77
YOLOC	85.2	83.5	23.8	12.0	87	4.59
YOLOC-tiny	85.3	83.0	8.4	4.2	80	4.65

3.5 Comparison experiments in different environments

We extensively validated the YOLOC and YOLOC-tiny models across multiple validation subsets, encompassing various scenarios, to address varying lighting conditions and environmental complexities. These subsets comprise the test subsets TEST-ANL and TEST-AWL for diverse lighting conditions, along with the test subsets TEST-ACE and TEST-ASE representing varying environmental complexities. Table 5 presents the average detection accuracies of different SOTA detectors on the respective test sets.

Table 5 reveals that YOLOC-tiny exhibits notable mAP performance across all test sets, with impressive results on TEST-ANL and TEST-ACE. It achieves a substantial advantage on TEST-ANL, boasting a mAP of 84.0%, slightly below YOLOC's 84.7%. This finding suggests that YOLOC-tiny excels in detection under

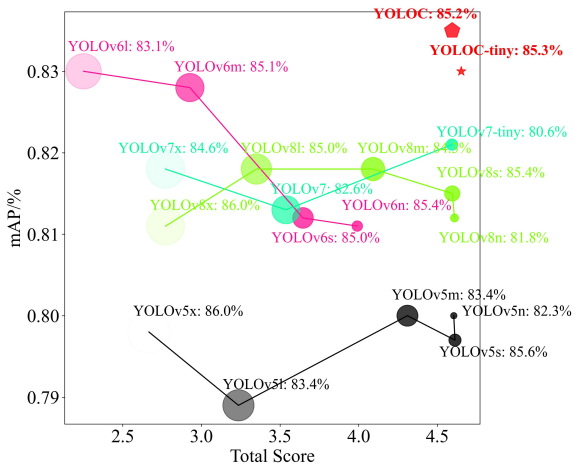


FIGURE 10
Detection performance charts for various SOTA models. The size of each geometric shape corresponds to the model size, with large shapes indicating large model sizes. The darkness of the geometric shape color represents the model parameter score, with dark colors indicating high parameter counts. The detection precision on the test set is provided for each model following its respective name.

TABLE 5 Detection accuracy of SOTA detectors in different validation subsets.

Model	mAP/% (TEST-ANL)	mAP/% (TEST-AWL)	mAP/% (TEST-ACE)	mAP/% (TEST-ASE)
YOLOv5n	81.1	90.6	79.3	86.0
YOLOv5s	80.9	91.7	79.2	83.9
YOLOv5m	81.0	89.4	79.4	83.5
YOLOv5l	80.1	90.1	77.4	83.4
YOLOv5x	80.4	90.9	79.1	83.5
YOLOv6n	82.1	87.3	82.1	82.1
YOLOv6s	82.2	90.8	82.2	82.2
YOLOv6m	83.8	85.8	83.8	83.8
YOLOv6l	83.9	89.0	83.9	83.9
YOLOv7-tiny	83.0	88.5	81.9	85.1
YOLOv7	82.0	92.4	81.1	84.5
YOLOv7x	82.8	90.7	81.6	84.2
YOLOv8n	81.9	91.6	80.4	84.8
YOLOv8s	81.7	91.0	80.9	83.9
YOLOv8m	82.6	91.0	81.4	85.0
YOLOv8l	82.8	90.8	81.3	85.3
YOLOv8x	81.5	91.5	80.2	84.6
YOLOC	84.7	91.3	83.9	84.7
YOLOC-tiny	84.0	90.7	82.6	85.4

regular lighting conditions. On TEST-AWL, the mAP of YOLOC-tiny is slightly lower than that of some algorithms. However, it still maintains a high level of performance. YOLOC-tiny achieves mAP scores of 85.4% and 82.6% on TEST-ASE and TEST-ACE, respectively, indicating its robustness in complex environments. These experimental results underscore the strong adaptability and practicality of YOLOC-tiny across various application scenarios.

3.6 Performance assessment in practical applications

Comparative experiments for real-world applications involving the YOLOC, YOLOC-tiny, YOLOv7, and YOLOv7-tiny models were conducted on ICPR, with deployment tests performed on PC3. The necessary software for model deployment includes onnx 1.14.0, onnxruntime-gpu 1.51.1, onnx-simplifier 0.4.33, and tensorrt 8.5.3.1. We initially exported the PyTorch models as general-purpose network models in the ONNX format. Then, we exported the ONNX model as a TensorRT model for ICPR deployment. Specific parameters, such as a confidence threshold of 0.4, an IOU threshold of 0.5, a model input image size of 640 × 640, and 32-bit floating-point precision, were set. Detection and labeling of images in the TEST-A dataset were performed on PC3 (Figure 11). Key metrics, including inference time, frame rate, detection accuracy, and the number of correctly detected citrus, were recorded. The accuracy rate was derived by sampling 29 images from the 290-image TEST-R test set for detection and manually verifying them multiple times. The detailed results are presented in Table 6.

Table 6 reveals that the inference times for YOLOC and YOLOC-tiny are 27.2 and 17.1 ms, respectively. Although slightly higher than the 12.7 ms of YOLOv7-tiny, the values mentioned are significantly lower than the 78.1 ms of YOLOv7 (Figure 12A). YOLOC and YOLOC-tiny achieve frame rates exceeding the 30 FPS threshold required for the real-time detection needs of the robot, with YOLOC-tiny reaching an impressive 59 FPS. Although YOLOv7 demonstrates high detection accuracy, its FPS falls far below real-time requirements (Figure 12B). YOLOC-tiny detects 3852 citruses, outperforming the other three models (Figure 12C). This finding indicates its ability to capture targets comprehensively. Moreover, YOLOC-tiny exhibits superior real-time performance in citrus fruit detection. The detection accuracies of YOLOC and YOLOC-tiny are 92.9% and 92.8%, respectively, slightly lower than the detection accuracy of YOLOv7 (93.8%) but higher than that of YOLOv7-tiny (91.5%). This finding suggests that both models boast high detection accuracy and offer fast inference speeds and a good balance (Figure 12D). These experimental results further confirm the exceptional performance of the YOLOC series in real robotics applications.

4 Discussion

Detecting and localizing fruits are crucial for the agronomic management of fruit crops, including yield prediction and automated harvesting (Fu et al., 2020a; Lu et al., 2023). Fruit harvesting operations typically account for 25% of the total production cost and 50% of the total labor force (Castro-Garcia et al., 2019). Developing lightweight, high-precision detection models suitable for deployment on robots with limited computational power can ensure operational efficiency in complex orchard environments (Liu et al., 2023; Xu et al., 2023). This also can provide stable visual information for early yield prediction and fruit thinning operations.

Although excellent algorithms for detecting ripe fruits such as citrus fruits (Xu et al., 2023), apples (Wang and He, 2021), and kiwifruit (Fu et al., 2021), and for detecting apples at different growth stages (Ma et al., 2024), have been proposed, rapid detection of multi-variety and multi-ripeness citrus fruits in complex orchards remains challenging. Additionally, balancing detection performance, speed, and model parameters on edge devices with limited computational power has yet to be achieved satisfactorily.

Based on engineering experience and experimental results, we compared and analyzed various SOTA object detectors. We selected YOLOv7 as the base network and implemented a series of optimizations and improvements, including using a lightweight backbone network and embedding the attention mechanism CBAM. We also designed metrics to comprehensively evaluate

10.3389/fpls.2024.1415006

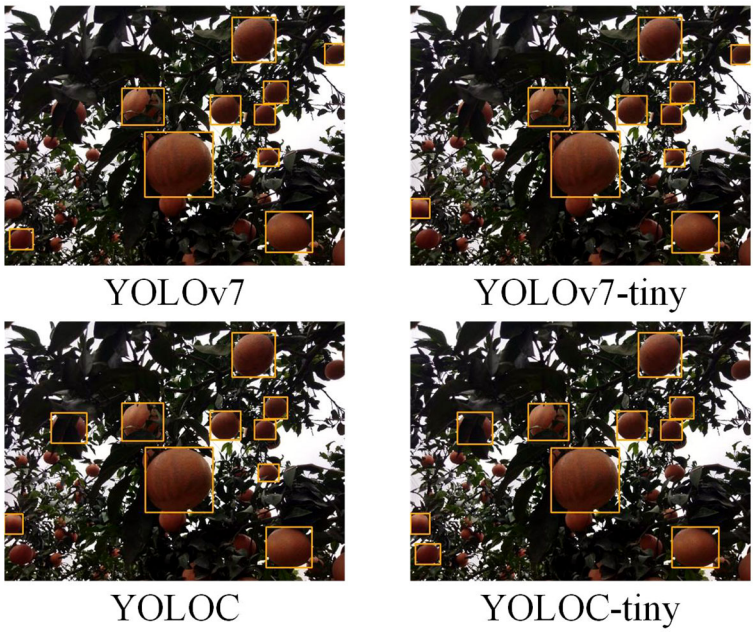


FIGURE 11
Detection results of various models in dark, complex environments.

the model's detection performance on edge devices with limited computational power (see Equations 11–14). Consequently, we proposed the lightweight detection model YOLOC-tiny.

While YOLOC-tiny demonstrated excellent detection performance in tasks involving multi-variety and multiripeness non-green-ripe citrus fruits, several limitations remain. First, as shown in Figure 9 of the revised manuscript, the model's detection capability for citrus fruits that are either distant or severely occluded is insufficient. Although these fruits can be detected as the robot moves, detecting small, distant citrus fruits and severely occluded citrus fruits requires further attention. Second, in distinguishing between different citrus varieties and maturities, YOLOC-tiny's detection accuracy is lower compared to algorithms that detect single-variety, single-maturity fruits (Fu et al., 2019; Apolo-Apolo et al., 2020a). As shown in Table 5, the mAP of YOLOC-tiny is slightly lower than that of YOLOv5n in simple environments, although YOLOC-tiny outperforms YOLOv5n in complex orchards and varying lighting conditions.

TABLE 6 Results of robot application experiments.

Model	YOLOv7	YOLOv7-tiny	YOLOC	YOLOC-tiny
Inference time/ms	78.1	12.7	27.2	17.1
FPS	13	79	37	59
Accuracy/%	93.8	91.5	92.9	92.8
Number of citrus	3723	3801	3758	3852

The accuracy values in the table were calculated by comparing the model's output results with the manual detection results.

Moreover, in this study, we only verified the impact of adding pure citrus image datasets on enhancing the detection performance of citrus fruits in unstructured environments, without conducting quantitative and qualitative research. Considering the basic conditions of the robot's operating environment, we used only seven data augmentation methods. Furthermore, transformers have proven effective in large language models and have recently been applied to object detection tasks, suggesting promising avenues for improving model performance (Zhu et al., 2022; Yang et al., 2023). We also note the recent advancements with YOLOv9 and YOLOv10.

We will further expand the dataset, enrich the images with various scenes and lighting conditions, or increase the image resolution. We will explore the effectiveness of generative adversarial networks and MixUp in robot applications in future research. Therefore, future work will focus on enriching the dataset and incorporating more efficient network architectures and modules to further enhance the model's detection performance and lightweight characteristics. In our future research, we plan to optimize the model structure further to improve the detection performance of citrus fruits in low-light environments.

To address these issues, we will expand the dataset, enhance image diversity with various scenes and lighting conditions, and increase the image resolution (Wang et al., 2022b). Additionally, we will explore the effectiveness of generative adversarial networks and MixUp in dataset augmentation. Future work will focus on incorporating more efficient network architectures and modules to enhance the model's detection performance and lightweight characteristics. We also plan to optimize the model structure to improve the detection of citrus fruits in low-light environments.

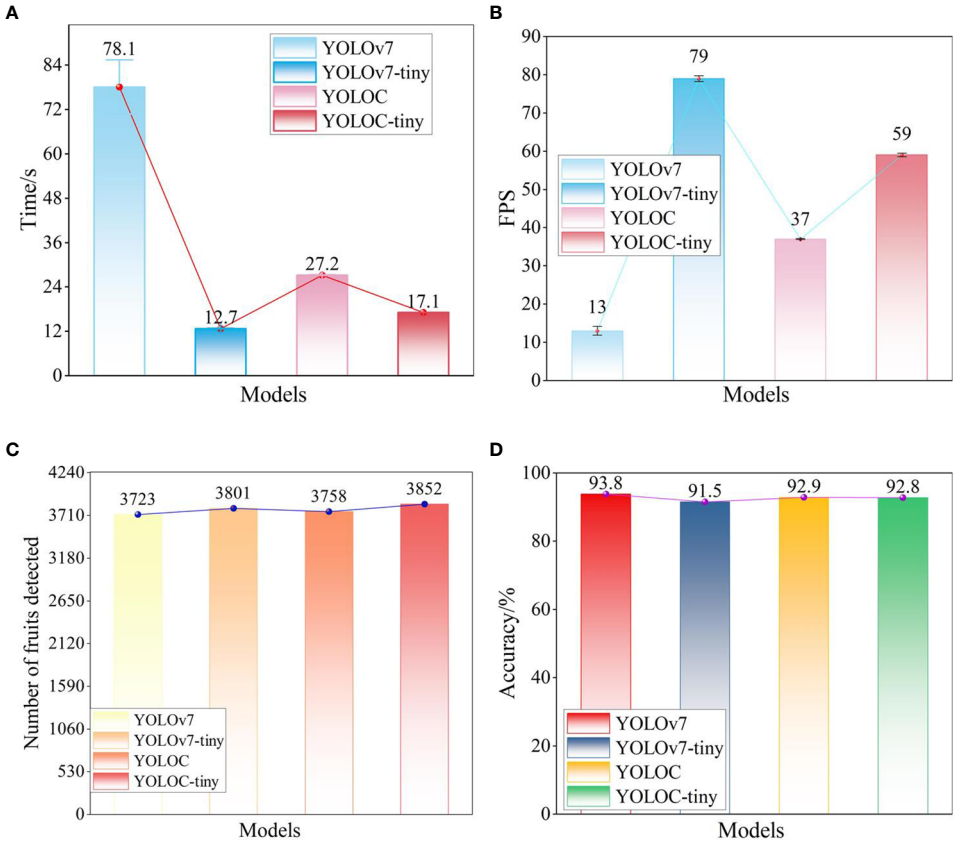


FIGURE 12
Detection results of different models deployed on ICPR. **(A)** Inference time of each model. **(B)** FPS of each model. **(C)** Number of citrus fruits detected. **(D)** Accuracy of each model.

5 Conclusions

We introduce a generalized lightweight detection model, YOLOC-tiny, tailored for large non-green-ripe citrus of different varieties with multiripeness in complex environments by optimizing the network structure and reducing the model size to enhance computational efficiency. Our methodology begins with the curation of image datasets featuring citrus fruits in various environments and ripeness stages, encompassing navel orange, Ehime Jelly orange, and Harumi tangerine. YOLOC-tiny utilizes the EfficientNet-B0 feature extraction backbone, streamlining model parameters whilst augmenting feature extraction capabilities. Furthermore, it integrates a spatial and channel hybrid attention mechanism, CBAM, to enhance access to contextual information, intensify focus on diverse citrus fruits, and achieve superior detection performance. Additional parameter reduction is achieved by implementing the LAMP strategy.

The key findings from our study include the following: (1) Ablation experiments confirm the effectiveness of our enhancement measures in improving network performance for non-green-ripe citrus fruit detection. (2) Compared with TRAIN-A, YOLOv7 based on the F-TRAIN-A dataset exhibits a 2.4% and 0.8% improvement in P and mAP, respectively. This finding validates the benefit of replacing citrus images in real scenes with a small number of pure citrus images in complex environments to enhance model detection performance. (3) Compared with other SOTA models, such as YOLOv8, YOLOC-tiny surpasses real-time detection requirements with an impressive frame rate. It also demonstrates superior detection performance. YOLOC-tiny achieves an 85.3% P and an 83.0% mAP at a frame rate of 80 FPS, with a parametric count of merely 4.2 M. (4) In a real-world deployment with a citrus-picking robot, ICPR, YOLOC-tiny attains 92.8% accuracy at a frame rate of 59. Thus, YOLOC-tiny provides real-time, accurate information on multiripeness and diverse citrus fruits for orchard robots.

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

ZT: Funding acquisition, Resources, Writing – original draft, Writing – review & editing, Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Software, Supervision, Validation, Visualization. LX: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Writing – review & editing. HL: Data curation, Formal analysis, Software, Validation, Visualization, Writing – review & editing. MC: Conceptualization, Data curation, Funding acquisition, Supervision, Validation, Writing – review & editing. XS: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Writing – review & editing. LZ: Data curation, Validation, Writing – review & editing, Methodology, Software. YW: Conceptualization, Data curation, Funding acquisition, Writing – review & editing. ZW: Funding acquisition, Methodology, Resources, Writing – review & editing. YZ: Data curation, Formal analysis, Resources, Writing – review & editing. KR: Software, Validation, Writing – review & editing. YH: Funding acquisition, Resources, Writing – review & editing. WM: Funding acquisition, Resources, Writing – review & editing. NY: Funding acquisition, Resources, Writing – review & editing. LL: Funding acquisition, Resources, Writing – review & editing. YQ: Funding acquisition, 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 founded by the National Natural Science Foundation of China (Grant No.32301704 and Grant No.32171909), the Department of Science and Technology of Sichuan province (Grant No. 2021JDRC0091 and Grant No. 22ZDYF0095), and the Sichuan Smart Agricultural Engineering Technology Research Centre.

Acknowledgments

We acknowledge the students who contributed to labeling raw non-green-ripe images for this research project, as well as the student and the EnPapers editors for their dedication to refining our research paper in impeccable English. We also extend our appreciation to the journal reviewers for their valuable contributions towards improving the 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

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2024.1415006/full#supplementary-material

SUPPLEMENTARY FIGURE 1
The MBConv convolutional network.

SUPPLEMENTARY FIGURE 2
Diagram of LAMP for PL-YOLO-tiny.

SUPPLEMENTARY TABLE 1
Details of citrus fruit image datasets.

SUPPLEMENTARY TABLE 2
Overview of the dataset and the number of images used in this study.

References

Apolo-Apolo, O. E., Martínez-Guanter, J., Egea, G., Raja, P., and Pérez-Ruiz, M. (2020a). Deep learning techniques for estimation of the yield and size of citrus fruits using a UAV. *Eur. J. Agron.* 115, 126030. doi: 10.1016/j.eja.2020.126030

Apolo-Apolo, O. E., Pérez-Ruiz, M., Martínez-Guanter, J., and Valente, J. (2020b). A cloud-based environment for generating yield estimation maps from apple orchards using UAV imagery and a deep learning technique. *Front. Plant Sci.* 11. doi: 10.3389/fpls.2020.01086

Bargoti, S., and Underwood, J. P. (2017). Image segmentation for fruit detection and yield estimation in apple orchards. *J. Field Robotics* 34, 1039–1060. doi: 10.1002/rob.21699

Castro-Garcia, S., Aragon-Rodriguez, F., Sola-Guirado, R. R., Serrano, A. J., Soria-Olivas, E., and Gil-Ribes, J. A. (2019). Vibration monitoring of the mechanical harvesting of citrus to improve fruit detachment efficiency. *Sensors* 19, 1760. doi: 10.3390/s19081760

Chen, M., Tang, Y., Zou, X., Huang, K., Huang, Z., Zhou, H., et al. (2020). Three-dimensional perception of orchard banana central stock enhanced by adaptive multi-vision technology. *Comput. Electron. Agric.* 174, 105508. doi: 10.1016/j.compag.2020.105508

Condotta, I. C. F. S., Brown-Brandl, T. M., Pitla, S. K., Stinn, J. P., and Silva-Miranda, K. O. (2020). Evaluation of low-cost depth cameras for agricultural applications. *Comput. Electron. Agric.* 173, 105394. doi: 10.1016/j.compag.2020.105394

Fu, L., Feng, Y., Tola, E., Liu, Z., Li, R., and Cui, Y. (2018). Image recognition method of multi-cluster kiwifruit in field based on convolutional neural networks. *Trans. Chin. Soc. Agric. Eng.* 34, 205–211. doi: 10.11975/j.issn.1002-6819.2018.02.028

Fu, L., Feng, Y., Wu, J., Liu, Z., Gao, F., Majeed, Y., et al. (2021). Fast and accurate detection of kiwifruit in orchard using improved YOLOv3-tiny model. *Precis. Agric.* 22, 754–776. doi: 10.1007/s11119-020-09754-y

Fu, L., Gao, F., Wu, J., Li, R., Karkee, M., and Zhang, Q. (2020a). Application of consumer RGB-D cameras for fruit detection and localization in field: A critical review. *Comput. Electron. Agric.* 177, 105687. doi: 10.1016/j.compag.2020.105687

Fu, L., Majeed, Y., Zhang, X., Karkee, M., and Zhang, Q. (2020b). Faster R–CNN–based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting. *Biosyst. Eng.* 197, 245–256. doi: 10.1016/j.biosystemseng.2020.07.007

Fu, L., Tola, E., Al-Mallahi, A., Li, R., and Cui, Y. (2019). A novel image processing algorithm to separate linearly clustered kiwifruits. *Biosyst. Eng.* 183, 184–195. doi: 10.1016/j.biosystemseng.2019.04.024

Gené-Mola, J., Sanz-Cortiella, R., Rosell-Polo, J. R., Morros, J.-R., Ruiz-Hidalgo, J., Vilaplana, V., et al. (2020). Fruit detection and 3D location using instance segmentation neural networks and structure-from-motion photogrammetry. *Comput. Electron. Agric.* 169, 105165. doi: 10.1016/j.compag.2019.105165

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. Available online at: https://openaccess.thecvf.com/content_cvpr_2014/html/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.html (Accessed May 20, 2024).

Huang, Y., Ren, Z., Li, D., and Liu, X. (2020). Phenotypic techniques and applications in fruit trees: a review. *Plant Methods* 16, 107. doi: 10.1186/s13007-020-00649-7

Lan, Y., Huang, Z., Deng, X., Zhu, Z., Huang, H., Zheng, Z., et al. (2020). Comparison of machine learning methods for citrus greening detection on UAV multispectral images. *Comput. Electron. Agric.* 171, 105234. doi: 10.1016/j.compag.2020.105234

Liu, C., Feng, Q., Sun, Y., Li, Y., Ru, M., and Xu, L. (2023). YOLACTFusion: An instance segmentation method for RGB-NIR multimodal image fusion based on an attention mechanism. *Comput. Electron. Agric.* 213, 108186. doi: 10.1016/j.compag.2023.108186

Liu, T.-H., Ehsani, R., Toudeshki, A., Zou, X.-J., and Wang, H.-J. (2018). Detection of citrus fruit and tree trunks in natural environments using a multi-elliptical boundary model. *Comput. Industry* 99, 9–16. doi: 10.1016/j.compind.2018.03.007

Liu, Y., Heying, E., and Tanumihardjo, S. A. (2012). History, global distribution, and nutritional importance of citrus fruits. *Compr. Rev. Food Sci. Food Saf.* 11, 530–545. doi: 10.1111/j.1541-4337.2012.00201.x

Lu, J., Chen, P., Yu, C., Lan, Y., Yu, L., Yang, R., et al. (2023). Lightweight green citrus fruit detection method for practical environmental applications. *Comput. Electron. Agric.* 215, 108205. doi: 10.1016/j.compag.2023.108205

Lu, J., Lee, W. S., Gan, H., and Hu, X. (2018). Immature citrus fruit detection based on local binary pattern feature and hierarchical contour analysis. *Biosyst. Eng.* 171, 78–90. doi: 10.1016/j.biosystemseng.2018.04.009

Lu, S., Chen, W., Zhang, X., and Karkee, M. (2022). Canopy-attention-YOLOv4-based immature/mature apple fruit detection on dense-foliage tree architectures for early crop load estimation. *Comput. Electron. Agric.* 193, 106696. doi: 10.1016/j.compag.2022.106696

Ma, B., Hua, Z., Wen, Y., Deng, H., Zhao, Y., Pu, L., et al. (2024). Using an improved lightweight YOLOv8 model for real-time detection of multi-stage apple fruit in complex orchard environments. *Artif. Intell. Agric.* 11, 70–82. doi: 10.1016/j.aiia.2024.02.001

Maheswari, P., Raja, P., Apolo-Apolo, O. E., and Pérez-Ruiz, M. (2021). Intelligent fruit yield estimation for orchards using deep learning based semantic segmentation techniques—A review. *Front. Plant Sci.* 12. doi: 10.3389/fpls.2021.684328

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)*. (Las Vegas, NV, USA: IEEE), 779–788. doi: 10.1109/CVPR.2016.91

Shaoqing, R., He, K., Girshick, R., and Sun, J. (2016). Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Tang, Y., Chen, M., Wang, C., Luo, L., Li, J., Lian, G., et al. (2020). Recognition and localization methods for vision-based fruit picking robots: A review. *Front. Plant Sci.* 11. doi: 10.3389/fpls.2020.00510

The United States Department of Agriculture (2024). Citrus: World Markets and Trade. Available online at: https://apps.fas.usda.gov/psdonline/circulars/citrus.pdf (Accessed June 9, 2024).

Tzutalin (2015). LabelImg. Git code, (2015). Available online at: https://github.com/tzutalin/labelImg.

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2022a). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. Available online at: http://arxiv.org/abs/2207.02696. doi: 10.1109/CVPR52729.2023.00721

Wang, D., and He, D. (2021). Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning. *Biosyst. Eng.* 210, 271–281. doi: 10.1016/j.biosystemseng.2021.08.015

Wang, X., Tang, J., and Whitty, M. (2022b). Data-centric analysis of on-tree fruit detection: Experiments with deep learning. *Comput. Electron. Agric.* 194, 106748. doi: 10.1016/j.compag.2022.106748

Wang, Y., Yan, G., Meng, Q., Yao, T., Han, J., and Zhang, B. (2022c). DSE-YOLO: Detail semantics enhancement YOLO for multi-stage strawberry detection. *Comput. Electron. Agric.* 198, 107057. doi: 10.1016/j.compag.2022.107057

Wei, L., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. (2016). "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016*. Eds. B. Leibe, J. Matas, N. Sebe and M. Welling (Springer International Publishing, Cham), 21–37. doi: 10.1007/978-3-319-46448-0_2

Xu, L., Wang, Y., Shi, X., Tang, Z., Chen, X., Wang, Y., et al. (2023). Real-time and accurate detection of citrus in complex scenes based on HPL-YOLOv4. *Comput. Electron. Agric.* 205, 107590. doi: 10.1016/j.compag.2022.107590

Yang, C. H., Xiong, L. Y., Wang, Z., Wang, Y., Shi, G., Kuremot, T., et al. (2020). Integrated detection of citrus fruits and branches using a convolutional neural network. *Comput. Electron. Agric.* 174, 105469. doi: 10.1016/j.compag.2020.105469

Yang, S., Wang, W., Gao, S., and Deng, Z. (2023). Strawberry ripeness detection based on YOLOv8 algorithm fused with LW-Swin Transformer. *Comput. Electron. Agric.* 215, 108360. doi: 10.1016/j.compag.2023.108360

Yu, J., Wu, T., Zhang, X., and Zhang, W. (2022). An efficient lightweight SAR ship target detection network with improved regression loss function and enhanced feature information expression. *Sensors* 22, 3447. doi: 10.3390/s22093447

Yu, Y., Zhang, K., Yang, L., and Zhang, D. (2019). Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. *Comput. Electron. Agric.* 163, 104846. doi: 10.1016/j.compag.2019.06.001

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., and Ren, D. (2020). "Distance-ioU loss: faster and better learning for bounding box regression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. Palo Alto, California, USA: Association for the Advancement of Artificial Intelligence (AAAI), 12993–13000. doi: 10.1609/aaai.v34i07.6999

Zhu, J., Yang, G., Feng, X., Li, X., Fang, H., Zhang, J., et al. (2022). Detecting wheat heads from UAV low-altitude remote sensing images using deep learning based on transformer. *Remote Sens.* 14, 5141. doi: 10.3390/rs14205141

Zhuang, J. J., Luo, S. M., Hou, C. J., Tang, Y., He, Y., and Xue, X. Y. (2018). Detection of orchard citrus fruits using a monocular machine vision-based method for automatic fruit picking applications. *Comput. Electron. Agric.* 152, 64–73. doi: 10.1016/j.compag.2018.07.004

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 05 July 2024
DOI 10.3389/fpls.2024.1412988

OPEN ACCESS

EDITED BY
Mohsen Yoosefzadeh Najafabadi,
University of Guelph, Canada

REVIEWED BY
Maciej Przybyłek,
Nicolaus Copernicus University in Toruń, Poland
Ali Taheri,
Tennessee State University, United States

*CORRESPONDENCE
Amith Khandakar
amitk@qu.edu.qa

RECEIVED 06 April 2024
ACCEPTED 07 June 2024
PUBLISHED 05 July 2024

CITATION
Prince RH, Mamun AA, Peyal HI, Miraz S, Nahiduzzaman M, Khandakar A and Ayari MA (2024) CSXAI: a lightweight 2D CNN-SVM model for detection and classification of various crop diseases with explainable AI visualization.
Front. Plant Sci. 15:1412988.
doi: 10.3389/fpls.2024.1412988

CSXAI: a lightweight 2D CNN-SVM model for detection and classification of various crop diseases with explainable AI visualization

Reazul Hasan Prince[1], Abdul Al Mamun[2], Hasibul Islam Peyal[1,3], Shafiun Miraz[3], Md. Nahiduzzaman[1], Amith Khandakar[4]* and Mohamed Arselene Ayari[5,6]

[1]Department of Electrical and Computer Engineering, Rajshahi University of Engineering and Technology, Rajshahi, Bangladesh, [2]Department of Computer Science and Engineering, Tejgaon College, Dhaka, Bangladesh, [3]Department of Computer Science and Engineering, Rajshahi University of Engineering and Technology, Rajshahi, Bangladesh, [4]Department of Electrical Engineering, College of Engineering, Qatar University, Doha, Qatar, [5]Department of Civil and Architectural Engineering, Qatar University, Doha, Qatar, [6]Technology Innovation and Engineering Education Unit, Qatar University, Doha, Qatar

Plant diseases significantly impact crop productivity and quality, posing a serious threat to global agriculture. The process of identifying and categorizing these diseases is often time-consuming and prone to errors. This research addresses this issue by employing a convolutional neural network and support vector machine (CNN-SVM) hybrid model to classify diseases in four economically important crops: strawberries, peaches, cherries, and soybeans. The objective is to categorize 10 classes of diseases, with six diseased classes and four healthy classes, for these crops using the deep learning-based CNN-SVM model. Several pre-trained models, including VGG16, VGG19, DenseNet, Inception, MobileNetV2, MobileNet, Xception, and ShuffleNet, were also trained, achieving accuracy ranges from 53.82% to 98.8%. The proposed model, however, achieved an average accuracy of 99.09%. While the proposed model's accuracy is comparable to that of the VGG16 pre-trained model, its significantly lower number of trainable parameters makes it more efficient and distinctive. This research demonstrates the potential of the CNN-SVM model in enhancing the accuracy and efficiency of plant disease classification. The CNN-SVM model was selected over VGG16 and other models due to its superior performance metrics. The proposed model achieved a 99% F1-score, a 99.98% Area Under the Curve (AUC), and a 99% precision value, demonstrating its efficacy. Additionally, class activation maps were generated using the Gradient Weighted Class Activation Mapping (Grad-CAM) technique to provide a visual explanation of the detected diseases. A heatmap was created to highlight the regions requiring classification, further validating the model's accuracy and interpretability.

KEYWORDS

convolutional neural network (CNN), support vector machine (SVM), gradient-weighted class activation mapping (GRAD-CAM), pre-trained models, plant diseases

1 Introduction

In Bangladesh, agriculture is crucial due to a growing population and higher food demand. Besides, the gross national income of the country and the families of the farmers depend on the agriculture field. Many countries rely on agricultural products and allied businesses as their primary source of income. One of the most basic and crucial necessities for any country is the safety and security of agricultural products Akbar et al. (2022). As plants are the health of agricultural development, so it is essential to increase the production of crops by ensuring the health of plant leaves. To boost plant yield, it's essential to address the issue of low yield caused by diseases from bacteria, viruses, and fungi. Moreover, Plant leaf diseases not only impact our daily lives but also have a terrible impact on farmers whose families depend on the production of plants. Identifying and classifying these diseases manually is both time-consuming and prone to errors. To address this, we suggest a deep learning approach for accurate and efficient identification and classification of plant leaf diseases. This method utilizes neural networks to extract characteristics of diseased parts, enhancing the accuracy of disease area classification. Detecting these plant diseases can help prevent them, and deep learning methods are effective for identification because they analyze data directly, focusing on specific task outcomes. This paper outlines the steps in a plant disease detection system and compares deep learning techniques for detecting plant diseases. To identify diseases by applying deep learning techniques, this paper introduces four kinds of crop leaves - Cherry, Peach, Strawberry, and Soybean.

Cherries hold notable importance in human health due to their rich nutritional profile and potential health benefits. Packed with antioxidants, particularly anthocyanins, cherries contribute to combating oxidative stress and inflammation, potentially promoting heart health and reducing the risk of chronic diseases. However, the cultivation of cherries is not without challenges, as various diseases, such as bacterial canker, brown rot, and powdery mildew, can pose significant drawbacks. The cherry leaves infected by Podosphaera pannosa will suffer powdery mildew, which is a serious disease threatening the cherry production industry Zhang et al. (2019). Thus, identifying a cherry leaf infected by Podosphaera pannosa only needs to identify whether the cherry leaf is healthy or diseased. To identify the diseased cherry leaves in the early stage, a combined technique of machine learning and deep learning have been used.

Peaches, both delicious and nutritious, hold significant importance in the realm of nutrition and well-being. Several diseases can attack peaches, including Bacterial spots, also known as Bacteriosis or shot holes. This disease also can be called peach spot. However, Bacteriosis severely affects peach crop production. Bacteriosis typically develops on the peach leaves first; therefore, the leaves are the primary source for recognizing plant disease Ebrahimi et al. (2017). The diseases reduce the yield of peaches and cause harm to human health. Thus, it is important to find rapid and accurate methods to identify peach diseases and further locate and segment the areas of the lesion in earlier stages Yao et al. (2022).

In many parts of the world, soybeans are the main food crop for people and an important source of oil for human consumption. But in recent years, some factors such as natural disasters, soil erosion, and fertilizer unreasonably lead to the occurrence of crop diseases. These diseases seriously affect soybean yield and quality in some aspects Gui et al. (2015). Traditional diagnosis of these diseases relies on disease symptom identification based on naked-eye observation by pathologists, which can lead to a high rate of false recognition. With the help of machine learning and deep learning knowledge, this infection of leaves can be identified, and take necessary steps in an earlier stage. This will lead to the prevention of the infection rate of other leaves. In this proposed article, three types of soybean diseases such as soybean sudden death, soybean yellow mosaic, and soybean bacterial blight which are significant threats to soybean plant production, have been classified as providing one healthy class.

Strawberries are one of the most sensitive and important crops in the world. Strawberries have high nutritional content and commercial value. So, it is a major fruit for daily consumption Skrovankova et al. (2015). Strawberries are easily infected by several plants' phytopathogenic fungi, bacteria, and viruses Maas (2012); Pan et al. (2014); Husaini and Neri (2016). That's why the diseases in strawberry leaves become the main interruption in its yield. Strawberry diseases are manually identified by growers, which is laborious and time-consuming. The shrinking workforce in agricultural counties also complicates this issue, since it is harder to accurately predict disease severity over a large scale. Therefore, it's urgent to develop an automatic system to identify the diseases in strawberry leaves Xiao et al. (2020). To accomplish the automatic identification of diseases, this article introduces a smart identification system using an image recognition technique for the detection of strawberry diseases using a Convolutional Neural Network (CNN) model. The traditional pathology method involves visually observing diseases, but it is labor-intensive, time-consuming, and heavily dependent on plant pathologists. To address these challenges, the Enzyme-linked Immunosorbent Assay (ELISA) has been suggested, capable of detecting viral protein content in plant extracts Clark and Bar-Joseph (1984). However, it proves less effective for diagnosing fungal and bacterial diseases. Another method, real-time polymerase chain reaction (PCR), is employed for testing plant pathogens, offering superior speed and accuracy compared to the aforementioned techniques Schaad and Frederick (2002). Nevertheless, widespread implementation is hindered by the requirement for skilled operators and the high cost of equipment. Consequently, we propose an image-based diagnostic method using deep learning. This approach is characterized by high accuracy, ease of implementation and the potential for real life implementation. The research offers some contributions. The contributions are –

- Building a deep learning CNN-based model to extract the most relevant features of the plant leaf images.
- Use of machine learning SVM model to classify the diseased and healthy plant leaf images.
- Keeping the model's parameters low, will produce a low-size model to use comfortably on any device.
- Comparison of the proposed CNN Model with some pre-trained model to show its acceptance and feasibility, as the

proposed model is superior to the transfer learning models in terms of parameters and accuracy.
- Comparison with the existing research works by providing the model's performance in terms of training accuracy, validation accuracy, precision, recall, F1-score, Receiver Operating Characteristics (ROC) curve, precision Vs recall curve, and the number of trainable parameters.
- Use of explainable AI to visualize the diseased areas that classify the plant leaves.

2 Related works

The early identification of the plant leaf disease is vital for profitable harvest yield in the agricultural field. Numerous types of research have been carried out to detect the leaf disease on the agricultural land. To achieve this goal, Hang et al. (2019) developed an integrated CNN-based model using squeeze and the Squeeze-and-Excitation module to classify 10 classes of plant leaves for 3 crops - apple, cherry, and corn. To achieve a good classification accuracy and lightweight model, the model was trained using global average pooling layers instead of dense layers. With a dataset containing less number of images, the proposed research work achieved 91.7% accuracy in identifying the diseases in cherries. Zhang et al. (2019) proposed a CNN model which was built based on a pre-trained model named GoogleNet. The model was applied in a binary classification with only 1200 images of cherry plant leaves. The experiment got an accuracy of 99.6% by adopting 5-fold cross-validation.

In order to detect bacteriosis in peach leaves, Akbar et al. (2022) looked for a novel lightweight CNN model based on VGG-19 and got the experimental result with 99% accuracy. The research was a binary classification of healthy and diseased peach leaves with a large dataset. The dataset consists of 1000 images, of which 70% are used for training and 30% for testing the Models. The LWNet Model uses 13 convolutional layers, the count of max-pooling is 7, and the dropout rate is 0.5 with the ReLu activation function. Alosaimi et al. (2021) proposed an innovative method for the binary classification of peach leaves and fruits with 3,199 images. The novel method consists of a CNN-based model and can also locate the region of disease and help farmers find appropriate treatments to protect peach crops. This innovative model got only 94% accuracy.

Soybean is another plant that needs to be identified whether it is infected or not. Wallelign et al. (2018) designed a CNN model based on LeNet architecture to classify four classes including a healthy class of soybean leaf. The authors collected a huge dataset of 12,673 samples and got an impressive accuracy of 99.32%. The research work was classified by only four classes of soybean leaves. Wu et al. (2023) proposed a classification method based on the improved ConvNeXt model where an attention module was used to generate feature maps at various depths and increase the network's focus on discriminative features as well as reduced background noise. The authors got an experimental accuracy of 85.42% which was comparatively poor in terms of AI-based disease detection. Although the research mentioned some evaluation metrics and a method to visualize the images, the number of model parameters was not satisfactory, as the model was not lightweight. Moreover, the model classified only three classes of soybean leaves including one healthy class. Yu et al. (2022) designed a model by constructing a residual attention layer (RAL) using attention mechanisms and shortcut connections, which further embedded into the residual neural network 18 (ResNet18) model to establish a new model of RANet based on attention mechanism and idea of residuals. The model achieved 98.49% accuracy for the recognition of three types of soybean leaf disease without providing a healthy class. Moreover, their proposed model was not lightweight. Jadhav et al. (2019) presented a novel system using the support vector machine (SVM) and K-Nearest Neighbor (KNN) classifiers used for classifying soybean diseases using color images of diseased leaf samples. The research was applied to the four classes of soybean leaves - blight, brown spot, frog eye leaf spot diseases, and Healthy samples with an accuracy of 87.3% and 83.6%. Besides, the authors didn't mention the lightweightness of their model and there was no method of visualization through explainable AI in terms of detecting strawberry diseases. The automation of agriculture and image recognition techniques are indispensable.

Xiao et al. (2020) proposed a CNN model based on ResNet50 that achieves a classification accuracy rate of 100% for leaf blight cases affecting the crown, leaf, and fruit; 98% for gray mold cases and 98% for powdery mildew cases. The overall accuracy rate for the feature images of the dataset was 99.60%. The dataset was not augmented as the number of total images was just 1306 and the feature images were built up manually. Moreover, the authors didn't use some performance evaluation metrics such as confusion metrics, ROC curves, and PR curves to compare the experimental results. Besides, there was no talk about visualization techniques. With the 5 types of classes, the authors managed to get a decent accuracy. Dhivya and Shanmugavadivu (2021) proposed a work that was more concentrated on image pre-processing for the reduction of noise using various filtering methods. The image preprocessing helps to enhance the feature extraction and classification of the leaf disease. The experimental results on the proposed separating model have been assessed regarding PSNR and MSE incentive to clarify and demonstrate the precision of the sifting models by using some image filters based on gradients. Abbas et al. (2021) worked with four pre-trained CNN models to detect the diseases of strawberry scorch with just only 2 types including one healthy class. All the trained CNN models were integrated with a machine vision system for real-time image acquisition. The authors showed an impressive comparison between the transfer learning models and tried to implement the best one for the identification of strawberry disease where EfficientNet-B3 achieved 92% and 97% classification accuracy for initial and severe stage leaf scorch disease respectively. SqueezeNet recorded the lowest disease classification accuracy values in comparison with AlexNet, VGG-16 and EfficientNet-B3. Shoaib et al. (2023) proposed a CNN model that can identify four prevalent diseases: powdery mildew, rust, leaf spot, and blight from 8000 images. The model was trained with multiple hyperparameters, such as the learning rate, number of hidden layers, and dropout rate, and attained a test set accuracy of 95.5%. The authors presented a comparison by changing different

hyperparameters and displayed hyperspectral images representing four prevalent types of plant diseases. The results demonstrate that the proposed CNN model performed better when compared with other machine learning image classifiers such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree, and Random Forest.

Based on the literature reviews, the following gaps have been identified:

- Many studies highlight challenges with limited dataset sizes, impacting the model's ability to generalize effectively. There is a need for larger and more diverse datasets to enhance model robustness and performance across various environmental conditions.
- The pursuit of lightweight models is emphasized in some studies; however, achieving both high accuracy and model simplicity remains a challenge. Research gaps exist in the development of efficient yet accurate lightweight models suitable for resource-constrained environments, such as on-field applications.
- Several studies achieve high accuracy in disease classification but lack in explaining the affected regions within plant images. Future research should focus on integrating explainable AI techniques to visualize and interpret model decisions, aiding farmers in targeted disease management.
- Some studies fall short in providing a comprehensive set of evaluation metrics, such as confusion matrices, ROC curves, and PR curves. A standardized and thorough evaluation approach is essential for comparing models and understanding their performances.
- Many studies focus on binary or limited multiclass classification, potentially overlooking a broader spectrum of plant diseases. Research gaps exist in addressing challenges associated with an increased number of disease classes and ensuring accurate identification within diverse plant species.
- While several studies propose innovative models, there is often a lack of emphasis on the lightweight nature of these models, critical for practical on-field applications. Future research should prioritize the development of lightweight models without compromising accuracy.
- Certain studies lack comprehensive comparisons between different models or hyperparameters, limiting insights into the effectiveness of various approaches.
- While some studies explore hyperparameters, there was no room for more systematic investigations into the impact of hyperparameter variations on model performance.

With the advancement of machine learning, all the traditional techniques of observing plant diseases have been considered time-consuming and complex. To assist farmers in increasing crop production and identifying diseases at earlier stages, this research proposed a CNN-based technique that combines machine learning and deep learning models. Our research purpose is to make the farmers familiar with the advancement of modern technology easily and identify plant diseases without any confusion. To achieve this goal, different performance evaluation metrics have been added to this research that represent the acceptance of our CNN-SVM model.

3 Materials and methods

For the identification of four plant leaf diseases, a 2D CNN-SVM model has been proposed in this research. The model was trained using the Kaggle platform to get the advantages of a Graphics Processing Unit (GPU). To implement the model, various Python libraries like numpy, and pandas and machine learning frameworks like tensorFlow, and keras were applied. Additionally, an explainable AI technique Grad-CAM was used to know the explanation of the outcome performed by the proposed model.

3.1 Overall process of establishing the recognition model

Firstly, a large dataset containing ten classes of four types of crop images was collected combined from Kaggle datasets named 'PlantVillage' and 'Soybean Diseased Leaf Dataset'. In the final dataset, we collected four plants (peach, cherry, soybean, and strawberry) healthy and diseased data. After collecting the dataset, we did feature scaling (Normalization) to make our picture size similar and data augmentation like rotating those pictures in different positions to train our model correctly. So, data augmentation is used to increase the diversity and size of a training dataset by applying various transformations to the existing data. By generating new samples from the original data through transformations such as rotation, flipping, cropping, scaling, or adding noise, data augmentation helps improve the robustness and generalization of deep learning models. After data augmentation and scaling, the dataset was ready to be trained by our proposed CNN-SVM model.

As demonstrated in Figure 1, for the identification of four plant leaf diseases, a 2D CNN-SVM model has been proposed in this research. CNN model has the power of extracting features efficiently which helps in the classification system. The CNN model has been fed an enormous dataset that was also augmented to get a generalized and reliable model. In this research, for classification, we used a machine learning model Support Vector Machine (SVM) that works with numerical data. Therefore, CNN works as the collector of featured data for the SVM model. Moreover, Convolutional Neural Networks (CNNs) have revolutionized image analysis and pattern recognition, offering several advantages over traditional observation methods. By implementing the CNN, we extracted features from the dataset, Now, we need to detect and classify key classes from those features in this step we used SVM, a machine learning method for classification. By implementing SVM, we successfully classified the healthy and diseased classes of the cherry, peach, soybean, and strawberry. After correctly classifying the healthy and disease

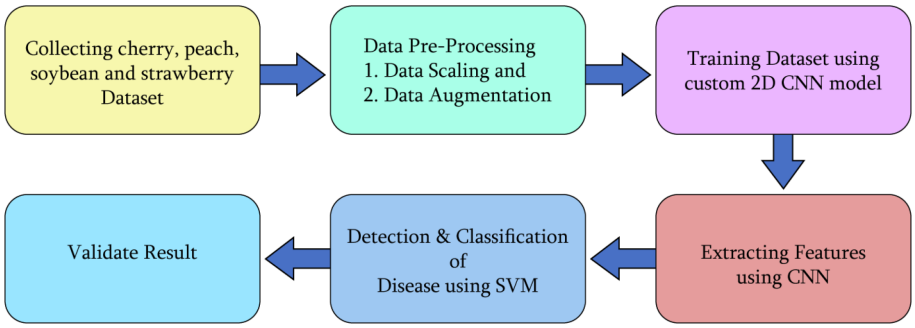


FIGURE 1
An overview of the whole methodology of the research.

classes, we validate the result by obtaining some performance evaluation metrics - training and validation accuracy curve, loss curve, ROC and confusion matrix.

3.2 Dataset description

The importance of a well-curated and representative dataset in deep learning research cannot be overstated. A dataset serves as the foundation upon which deep learning models are built, trained, and evaluated. The quality, diversity, and size of the dataset directly influence the performance, generalization, and reliability of the models developed.

To maintain the good performance, generalization, and reliability of the proposed model, a dataset with four types of plant leaves was collected from the publicly available 'PlantVillage' dataset and public available Kaggle 'Soybean Diseased Leaf Dataset'. The following Table 1 shows that a total 11,504 numbers of plant leaf images were used as the dataset to feed the proposed novel model. The merged dataset consists of four plant leaves – Cherry, Peach, Strawberry, and Soybean. Each type of plant includes healthy and some diseased classes. To make the model well-trained, a total 9,220 numbers of images have been used as training datasets, and 2,304 images for testing purposes are organized into 10 classes (Six diseased classes and four healthy classes). Therefore, the split ratio of the training and testing dataset is approximately 4:1. Here, Figure 2 depicts example images from all the classes of the dataset.

TABLE 1 Dataset details.

Plant	Disease Type	Training	Testing
Cherry	Cherry Mildew Cherry Healthy	842 682	210 171
Peach	Peach Spot Peach Healthy	1,838 288	459 72
Strawberry	Strawberry Scorch Strawberry Healthy	887 365	222 91
Soybean	Soybean Bacterial Blight Soybean Sudden Death Soybean Yellow Mosaic Soybean Healthy	71 88 88 4,071	17 22 22 1,018
Total		9,220	2,304

3.3 Data preprocessing

Image processing plays a pivotal role in enhancing the effectiveness of deep learning models by facilitating the extraction of meaningful features from visual data. In the area of computer vision, where deep learning models are commonly employed for image classification, object detection, and segmentation tasks, raw images often contain an abundance of information. In this research, for the processing of images, two steps have been followed.

3.3.1 Data scaling/resizing

Data scaling or resizing is a crucial preprocessing step in the realm of deep learning, especially for models designed to extract features from diverse datasets. Resizing involves adjusting the dimensions of input data to a uniform size. By bringing input features to a standardized scale, the optimization process becomes more efficient. In this study, the images were resized into 120 X 120 for both the proposed 2D CNN-SVM model and the transfer learning models. Therefore, it becomes ideal to measure the performance of the proposed model and transfer the learning model on a uniform scale.

3.3.2 Image augmentation

Augmentation is a useful technique to make our model more adaptable and avoid getting too focused on specific details. We applied augmentation to generate more images and increase the dataset's size. The main goal of augmentation is to add some variety to the images quantitatively, which aids the model in avoiding overfitting during training. Overfitting happens when the model starts memorizing random details instead of grasping the actual patterns in the data. Augmentation achieves this by introducing distortions to the images. As demonstrated in Figure 3, data augmentation includes different tricks like zooming, shearing, rotating, shifting in height and width, and flipping horizontally or vertically. These techniques create a diverse set of images for our model to learn from, promoting better generalization. For this purpose, some augmentation techniques have been applied in the

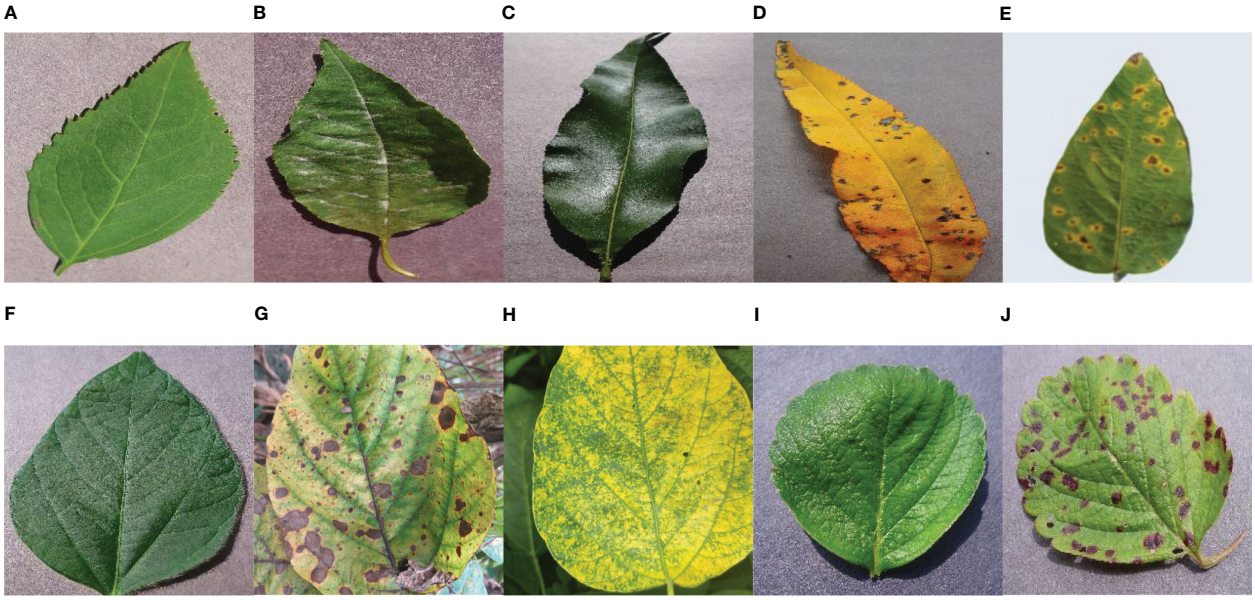


FIGURE 2
Visual Description of the dataset – **(A)** Cherry Healthy Leaf **(B)** Cherry Mildew **(C)** Peach Healthy Leaf **(D)** Peach Spot **(E)** Soybean Bacterial Blight **(F)** Soybean Healthy **(G)** Soybean Sudden Death **(H)** Soybean Yellow Mosaic **(I)** Strawberry Healthy **(J)** Strawberry Scorch.

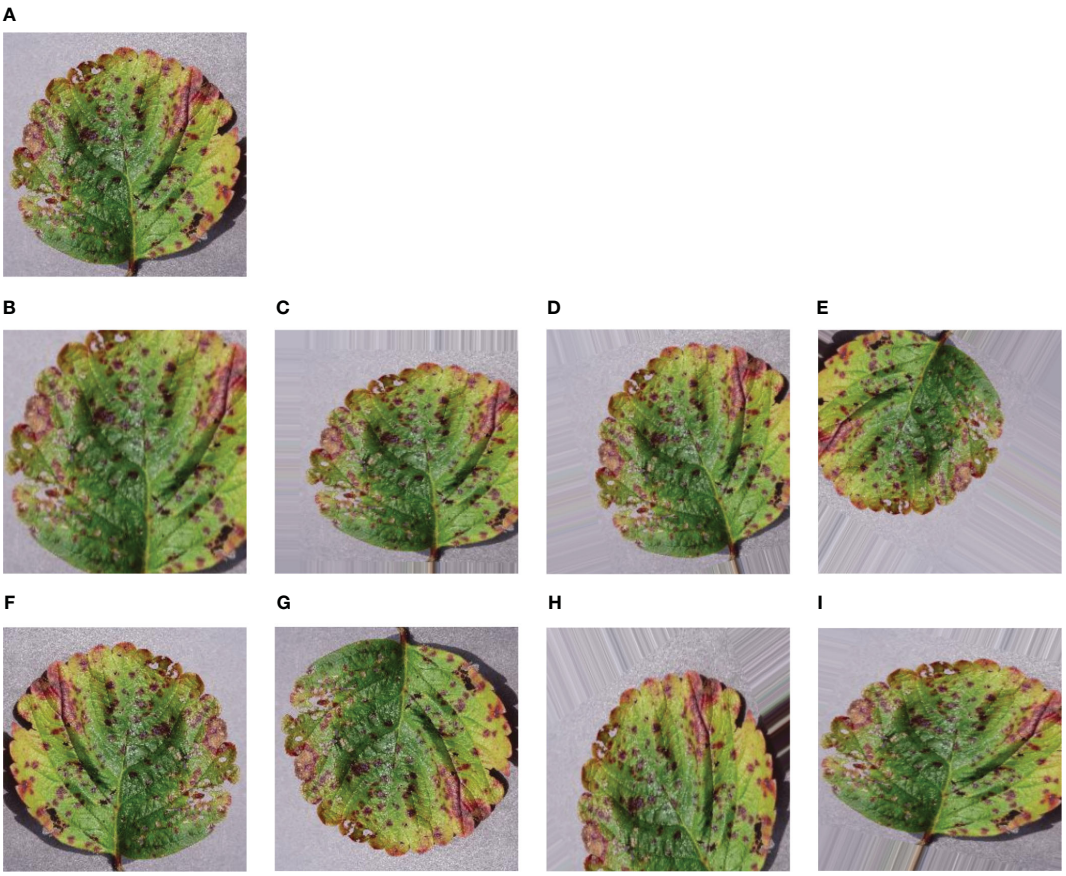


FIGURE 3
Sample of some augmented leaf images – **(A)** Original leaf image **(B)** Zoomed image **(C)** Sheared image **(D)** Rotated image **(E)** Fill Mode image **(F)** Horizontally Flipped image **(G)** Vertically Flipped image **(H)** Height Shifted image **(I)** Width Shifted image.

training images so that the model can observe the dataset from various aspects and validate the dataset from the memorized features. After applying eight techniques of data augmentation, our training dataset gathered a huge collection of datasets. So, a total of 73, 760 images were achieved from the augmentation.

3.4 Proposed hybrid method of CNN and SVM

The proposed hybrid (CNN-SVM) model is designed to combine both CNN & SVM advantages for the good classification of plant diseases. In this research, a simple structured 2D CNN model has been proposed to absorb the most important features in the plant leaf images. As CNN is a powerful tool for extracting features and taking two-dimensional inputs, we chose the CNN model to reach our goal. Moreover, enhancing the classification performance of the model relies on extracting distinctive features specific to different leaf diseases. These distinctive attributes play a crucial role in effectively categorizing leaf diseases. The architecture of the suggested 2D CNN model is depicted in Figure 4. The model has been formed using four convolutional and max-pooling layers. A max-pooling layer was added following each convolutional layer. Each layer is followed by a batch normalization layer.

The batch normalization layer speeds up the training process of the model. The utilization of batch normalization was implemented to enhance and expedite the model's performance by readjusting and rescaling the inputs of the layers Santurkar et al. (2018). Besides, the max-pooling layer assumes a pivotal role in the feature extraction process within convolutional neural networks (CNNs). Its primary function involves reducing the spatial dimensions of input feature maps and effectively downsizing them while preserving essential information. This downsampling operation facilitates the identification of prominent features by emphasizing the most significant values within local regions and removing useless data. This process is called subsampling.

In essence, MaxPooling contributes to the extraction of dominant features by highlighting the highest values, resulting in a more refined and condensed representation. Another important step used in the model is to flatten the layer. when the pooling layer is applied and the all-important feature is mapped, the flatten layer converts 2D arrays to 1D arrays before applying a fully connected layer (CNN-SVM) and is followed by batch normalization. In this context, the utilization of dropout aimed to mitigate overfitting by intermittently excluding the training of all nodes within each layer throughout the training process. This strategic approach led to a notable acceleration in training speed, contributing to more efficient model training Peyal et al. (2023). After accelerating the training speed, it is crucial to note that the fully connected layer represents the final layer of a neural network. In all neural networks, every node in this layer is properly connected, and the last layer of the model works as a machine learning classifier named Support Vector Machine (SVM). This layer classifies our research goal using the numerical features collected from the CNN model. This layer ensures that the information learned and processed through the preceding layers is synthesized to produce the final prediction or classification output.

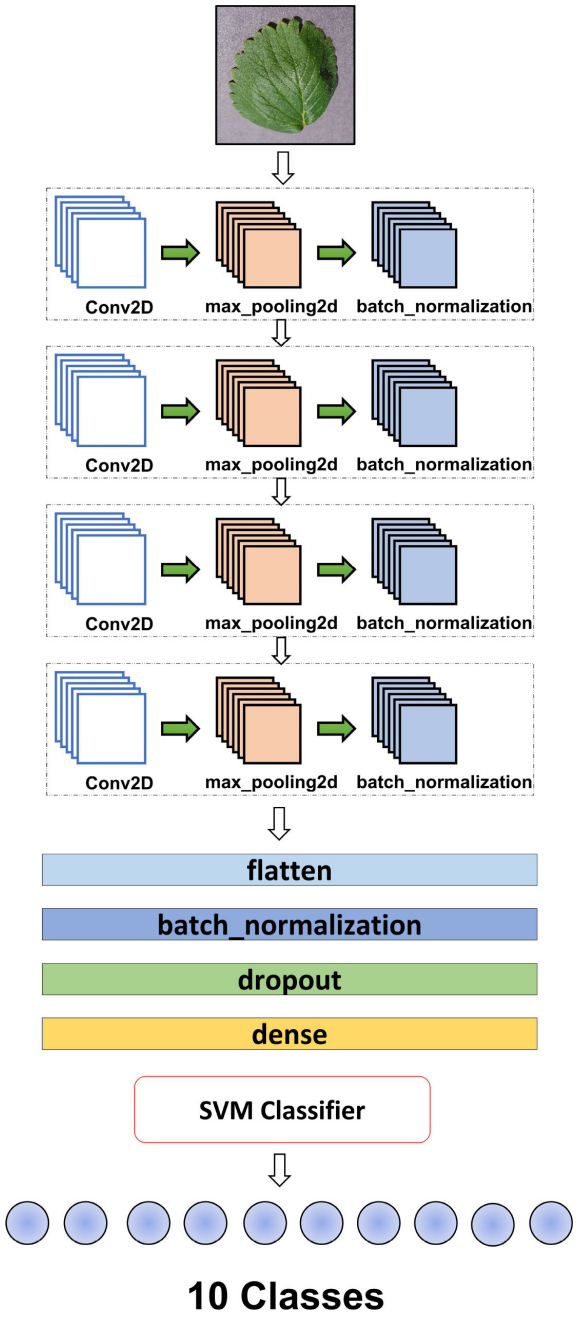


FIGURE 4
An overview of the whole methodology of the research.

The following Figure 4 depicts the proposed CNN-SVM model where the CNN model acts as the most relevant feature extractor and the SVM model as the disease classifier. The summary of the proposed model has been drawn in Table 2. The table also shows the lightweightness of the model where the number of total parameters is just only 393k which is very impressive and outperforms that of the transfer learning models mentioned in this research. Table 3 describes all the hyperparameters of the models including 2D CNN-SVM and transfer learning models – VGG16, VGG19,

TABLE 2 Summary of proposed simple 2D CNN model.

Layer (type)	Output Shape	Parameters
L1 (Conv2D)	(None, 120, 120, 16)	448
max_pooling2d (MaxPooling2D)	(None, 60, 60, 16)	0
batch_normalization (Batch Normalization)	(None, 60, 60, 16)	64
L2 (Conv2D)	(None, 60, 60, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 32)	0
batch_normalization_1 (Batch Normalization)	(None, 30, 30, 32)	128
L3 (Conv2D)	(None, 30, 30, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 15, 15, 64)	0
batch_normalization_2 (Batch Normalization)	(None, 15, 15, 64)	256
L4 (Conv2D)	(None, 15, 15, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 8, 8, 128)	0
Flatten (Flatten)	(None, 8192)	0
batch_normalization_4 (Batch Normalization)	(None, 8192)	32768
dropout (Dropout)	(None, 8192)	0
dense (Dense)	(None, 32)	262176
dense_1 (Dense)	(None, 10)	330
Total parameters:		393674
Trainable parameter:		376810
Non-trainable parameter:		16864

DenseNet, Inception V3, MobileNet, MobileNet V2, ShuffleNet and Xception used in this research.

To show the acceptance of the 2D CNN-SVM model, the hyper-parameters were kept the same for the training purpose of all transfer learning models. Overall, the experiment helped to detect the plant leaf diseases impressively.

4 Experiment and results

4.1 Experimental environment

The experimental environment for image classification using Convolutional Neural Network (CNN) and Support Vector Machine (SVM) involved the utilization of the Kaggle platform, leveraging its available Nvidia P100 GPU with specifications including 16 GB of GPU memory, a clock speed of 1.32 GHz, and a performance capability of 9.3 TFLOPS. To enhance model training efficiency, the input sample size for plant disease images was adjusted to 120 × 120 pixels to match the real-world operating conditions. The training process employed a batch size of 32 for training samples over 350 epochs. The Rectified Linear Unit (ReLU) activation function was applied, and batch normalization was incorporated to normalize batch data. The RMSprop optimizer with a learning rate of 0.001 was chosen for model optimization. Both the proposed CNN-SVM model and transfer learning models shared the same training and validation set sample sizes, training batch configuration, and activation function in the experiment.

4.2 Performance metrics

A classification report serves as a comprehensive overview of how well a model performs by highlighting crucial metrics like precision, recall, and F1-score for individual classes. Precision assesses the accuracy of positive predictions, while recall measures the model's capability to identify all relevant instances. The F1 score combines precision and recall, presenting a consolidated metric. Additional metrics such as accuracy, indicating overall correctness, and the confusion matrix, which breaks down true positives, true negatives, false positives, and false negatives, contribute to a thorough evaluation. Besides the Precission-Recall curve (PR), Region of Convergence (ROC) and loss curve were also used indicating the overall impressive function of the research. These metrics together provide a detailed insight into a model's strengths and weaknesses, enabling practitioners to make well-informed decisions regarding model improvement and selection based on

TABLE 3 Evaluation metrics comparison with transfer learning models.

Models	Accuracy	Precision	Recall	F1-Score	Parameters	Model Size (MB)
DenseNet	53.82%	76%	78%	70%	7053642	26.91
Inception V3	97.70%	98%	97%	97%	47521706	181.28
MobileNet V2	77.65%	97%	97%	97%	3579978	13.66
MobileNet	69.70%	94%	83%	83%	3250058	12.40
ShuffleNet	98.83%	100%	100%	100%	967874	3.70
VGG 16	98.35%	96%	94%	95%	24683850	94.16
VGG 19	97.61%	97%	96%	96%	20106314	76.70
Xception	84.85%	88%	80%	82%	20881970	79.66
Proposed model	99.09%	99%	99%	99%	393674	1.50

the specific demands of the image classification task. Thus, the performance of the CNN models was evaluated with these different evaluation metrics. Precision, recall, F1Score, and test accuracy metrics were used to evaluate the performance of the convolutional neural network models that were used in training. Validation and test outcomes for all CNN models were adapted in matrices of binary confusion, which are true positive (TP), false positive (FP), true negative (TN), and false negative (FN) Skrovankova et al. (2015). The first performance evaluation criterion, Accuracy rate, is used to evaluate the performance of network models. The accuracy rate refers to the proportion of the number of corrected positive predictions to that of the whole positive predictions Hang et al. (2019). It signifies the ratio of accurately identified images to the total number of images and is expressed by:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision measures how accurate your model is when it predicts positive instances. It's calculated by taking the number of true positive predictions and dividing it by the total number of positive predictions (both true positives and false positives). It can be quantified as,

$$\text{Precision} = \frac{TP}{TP + FP}$$

The Recall measures the efficiency of the neural network in identifying and categorizing the target, determined through the following calculation:

$$\text{Recall} = \frac{TP}{TP + FN}$$

The F1-score serves as the harmonic mean of precision and recall, providing a balanced metric that considers both false positives and false negatives. It is calculated by taking the reciprocal of the average of precision and recall through the following equation:

$$F_1 - \text{Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

4.3 Multiclass classification results

4.3.1 Accuracy graphs

Accuracy is defined as the sum of correct classifications divided by the total number of classifications. The sum of all diagonal elements is divided by the sum of all items in the confusion metrics. Accuracy gives the overall correctness of the predicted model. The accuracy of the model is drawn across the number of epochs which is called the accuracy graph. The accuracy graph contains both the training and validation accuracy (99.15% and 99.09%) in terms of epoch numbers. According to our research, the first adoption of the proposed CNN-SVM model has been clear from the accuracy graphs of our proposed CNN-SVM model which is shown in Figure 5.

From Table 3, it is observed that the validation accuracy of VGG16, VGG19, Inception V3, shuffleNet, MobileNet, MobileNet V2, DenseNet and Xception are 98.35%, 97.61%, 97.70%, 98.83%, 69.70%, 77,65%, 53.82% and 84.85% respectively. On the other hand, we checked our model in various epochs and environments (Table 4) and got the accuracy of 99.09%. In Figure 6, the accuracy comparison bar graph has also been shown to observe the outcome of various transfer learning models and the proposed model. Therefore, it is evident that the evaluation metrics accuracy, precision, recall, and F1-score of the proposed model are significantly higher than the transfer learning models which is a very good indicator of the reliability of the proposed model's performance in classifying 10 categories of plant leaf diseases from a huge dataset using CNN-SVM combined model.

4.3.2 Confusion matrix

The confusion matrix is a table that gives information about how the test dataset performs on the trained model Sharma et al. (2022). Various performance measures like accuracy, precision, recall, or sensitivity and specificity of the model can be calculated using the confusion matrix Tripathy et al. (2015). The diagonal values of the confusion matrix represent true positives (TP). To obtain false negatives, we have to add the values in the corresponding row items ignoring the true positive values. The total number of testing samples belonging to a given class can be calculated by the sum of all items of rows corresponding to that class (TP + FN). Similarly, the number of false positives (FP) for a class is obtained by adding the values of the corresponding column ignoring true positives TP for that class. The total number of true negative TN for a certain class will be the sum of all columns and row values ignoring that class's column and row. However, this study considered a 10-class problem, which consisted of four healthy classes and six different unhealthy classes of Cherry, Peach, Soybean and strawberry leaves. It is noticeable that out of 2304 images, only 21 images were misclassified by the proposed CNN-SVM model. Therefore, from Figure 7, it is clear that the proposed model can classify 10 numbers of classes accurately rather than the existing works.

4.3.3 ROC and PR curves

The ROC curve is a graphical representation of the trade-off between true positive rate and false positive rate at various thresholds. It is created by plotting the true positive rate against the false positive rate across different classification thresholds. The area under the ROC curve (AUC-ROC) quantifies the overall performance of the model. A higher AUC-ROC indicates better discrimination ability. From Figure 8, it is noticeable that the AUC score for the proposed model is almost nearly one and also it has surpassed the other transfer learning model's AUC. It is also known that a model with a higher AUC-ROC generally performs better. Besides, ROC curves provide insights into the model's ability to discriminate between classes. On the contrary, The PR curve represents the trade-off between precision and recall at different classification thresholds. Precision is the ratio of true positives to the sum of true positives and false positives. Recall is the ratio of true

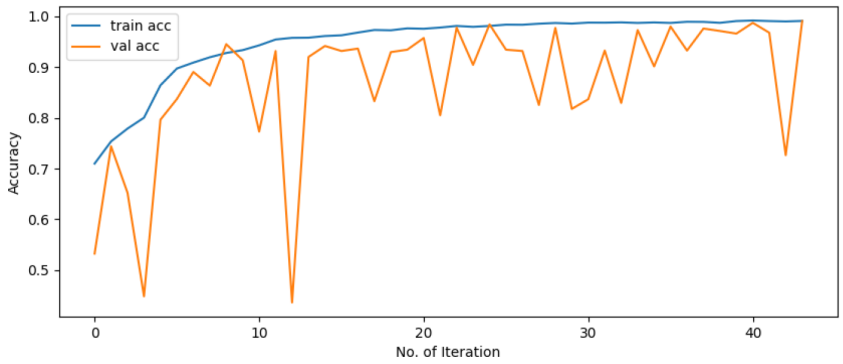


FIGURE 5
Accuracy graph of the proposed CNN-SVM model.

positives to the sum of true positives and false negatives. From the figure, both the ROC and PR curves show an impressive outcome of the proposed model. In summary, both ROC and PR curves provide valuable insights into different aspects of model performance.

4.3.4 Experimental research from different parameters

In order to achieve a reliable and robust classification model, the research was carried out using different optimizers such as Adam, SGD and RMSprop. The research was done at 350 epochs but we got our expected result within 50 epochs to train the model. In this environment, RMSprop Optimizer has given the best outcome. So, our proposed model gave 99.09% by using RMSprop as an optimizer whereas the SGD and Adam optimizer were not capable of giving this result. The following Table 5 shows the experimental results in the case of accuracy and AUC score for various optimizers.

From the table, it is proved that RMSprop performs better than other optimizers. Overall, the adaptability of RMSprop's learning rate, its stability during training, efficient memory usage, and rapid convergence made it a favored option across various scenarios, especially when handling complicated deep learning models and extensive datasets.

4.3.5 Matthews correlation coefficient

The MCC is crucial as it considers sensitivity, specificity, precision, and negative predictive value simultaneously, providing a holistic assessment of binary classification models. Matthews Correlation Coefficient (MCC) can also be used in multi-class classification problems but is typically used for binary classification tasks. Unlike the ROC AUC, the MCC generates a high score only when the classifier performs well across all four basic rates of the confusion matrix, ensuring a reliable evaluation.

A high MCC value always corresponds to high values for sensitivity, specificity, precision, and negative predictive value, making it a superior performance indicator compared to other metrics like F1 score and accuracy Chicco and Jurman (2023). The MCC ranges from -1 to +1, with -1 indicating perfect misclassification and +1 indicating perfect classification, while the DOR ranges from 0 to + Chicco et al. (2021).

In our proposed model, We have managed to acquire an impressive outcome of Matthews Correlation Coefficient (MCC) that is 0.987, which signifies a near-perfect classification performance. In summary, the attainment of an MCC value of 0.98 underscores the efficacy and reliability of our model's classification capabilities. It provides strong evidence that our model has learned meaningful patterns from the data and can generalize well to unseen instances, thereby instilling confidence in its practical utility and real-world deployment.

TABLE 4 Accuracy comparison in various epochs.

Epochs	Callback Function	Accuracy
42	Yes	99.09%
100	No	96.79%
200	No	97.98%

4.3.6 Mn/Mg deficient leaf vs. soybean sudden death

The symptoms of Mn/Mg deficient leaves and Soybean sudden death leaves are almost similar. These two can look alike, making it hard to distinguish them by eye since they have almost the same features in the images. In this case, we tried to classify them through our proposed model and got an outcome.

To separate the two species through the model, we collected pictures of Mn/Mg-deficient soybean leaves from Google and added those to our dataset after augmenting them.

After adding a new class of Mn/Mg-deficient soybean leaves to our original dataset, the proposed model was applied to the merged dataset. Figure 9 shows that the model achieved an impressive training accuracy of 99.11% and validation accuracy of 98.74% over the merged dataset. From the Figure 10 of the confusion matrix, it is seen that our model successfully classified all the images of Mn/Mg deficiency. To make the model recognize the difference among the soybean diseased classes, we increased the number of images in the dataset from 'DRYAD' dataset which contains high quality images of the same classes. Eventually, our model became successful in classifying them. In summary, our model has achieved a success rate

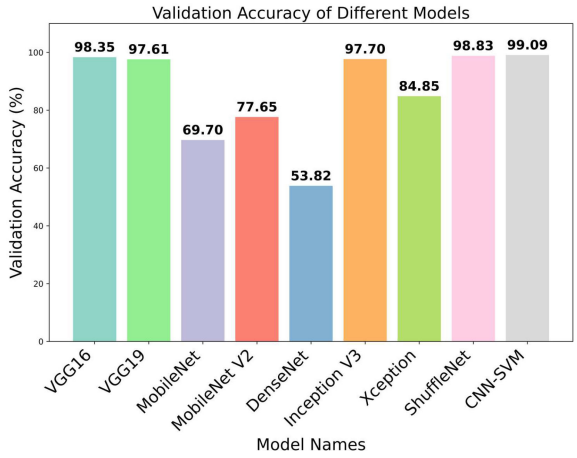


FIGURE 6
Bar graph of different transfer learning models for validation accuracy.

in distinguishing differences, even when they're hard to see with the naked eye.

4.3.7 Reusability of the proposed CNN-SVM model

The proposed CNN-SVM model was applied to a new, more extensive dataset comprising larger images of Soybean Rust, Soybean Frogeye Spot, and Soybean Healthy classes, collected from 'SoyNet', 'Soybean Leaf Disease Prediction', and 'Roboflow' datasets. After merging new classes to our proposed dataset, the model was trained on it and achieved an impressive validation accuracy of 99.04%, closely matching our original dataset's performance. Additionally, as shown in Figure 11, the classification for each class was satisfactory like before, maintaining the model's robust performance.

In Table 6, we have evaluated the performance variations of our proposed CNN-SVM model across different criteria. Initially, we observed that certain classes in our proposed dataset contained images that were relatively small in size. To address this, we replaced those classes with new ones featuring comparatively larger images from 'DRYAD' dataset which contains great quality images. Additionally, we noted potential confusion between the Mn/Mg deficient class and the Soybean Sudden Death class. To clarify this, we replaced the Soybean Yellow Mosaic class with the Mn/Mg deficient class in our dataset as we wanted to keep the similar types of soybean classes together and reassessed the model's performance. Finally, we showed the performance of our proposed dataset. Therefore, Table 6 represents an analysis of using the proposed model across different criteria. This analysis indicates the model's robustness and effectiveness across various datasets in the desired classification tasks.

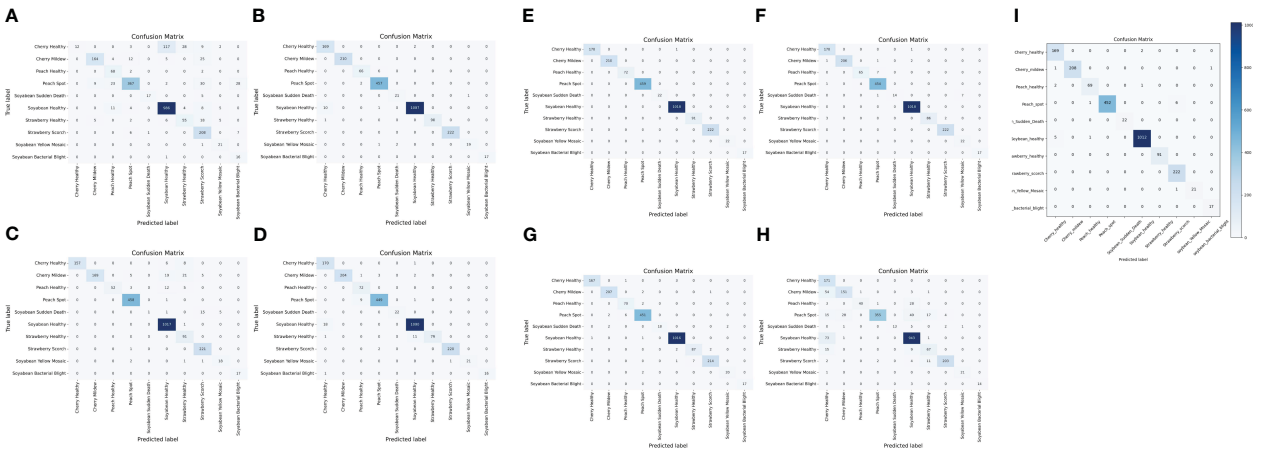
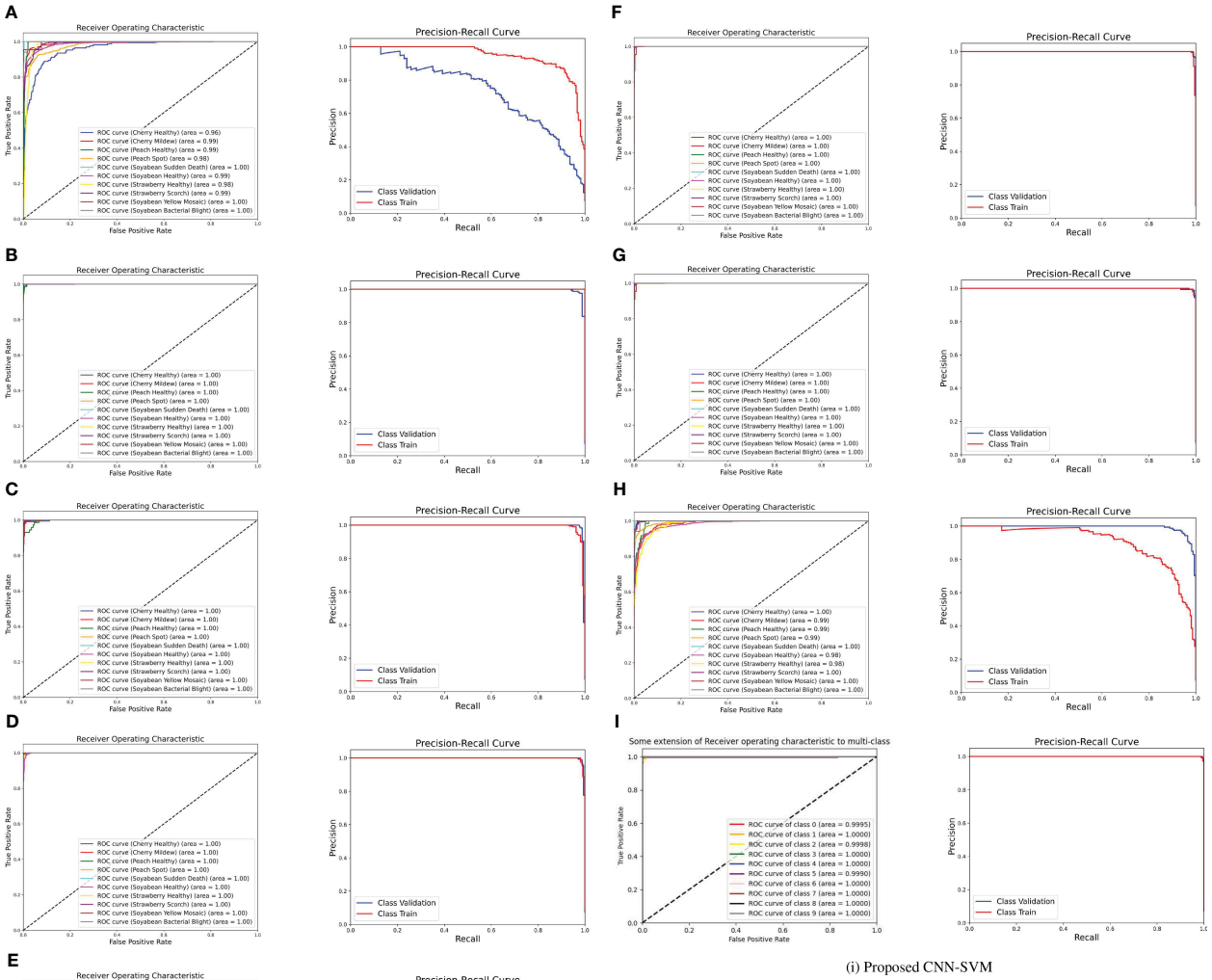


FIGURE 7
Confusion matrix of **(A)** DenseNet **(B)** Inception-V3 **(C)** MobileNet-V2 **(D)** MobileNet **(E)** ShuffleNet **(F)** VGG16 **(G)** VGG19 **(H)** Xception **(I)** Proposed CNN-SVM model.



(i) Proposed CNN-SVM

FIGURE 8
ROC curve and PR curve of **(A)** DenseNet **(B)** Inception-V3 **(C)** MobileNet-V2 **(D)** MobileNet **(E)** ShuffleNet **(F)** VGG16 **(G)** VGG19 **(H)** Xception **(I)** Proposed CNN-SVM model.

5 Comparative analysis

In conclusion, the proposed CNN-SVM model stands as a pioneering solution in the realm of plant disease classification, showcasing a unique fusion of CNN and SVM for optimal feature extraction and classification. The model's exceptional performance, as evidenced by its accuracy, evaluation metrics, lightweight design, and the incorporation of explainable AI techniques, underscores its superiority. Notably, when compared to well-established transfer learning models such as VGG16, VGG19, MobileNet, MobileNet-V2, DenseNet, Inception-V3, Xception and ShuffleNet, our model emerges as the clear frontrunner. Table 3 shows that our model performs better than other transfer learning models in terms of accuracy, precision, recall, F1-score, number of parameters and model size. Even when compared to strong competitors like VGG16, VGG19, Inception V3, and ShuffleNet, our model outperforms them across all evaluation measures. Impressively, it achieves superior precision, recall and F1-score metrics, further

TABLE 5 Comparison of various optimizers.

Optimizer	Accuracy	AUC Score
Adam	99%	99.96%
SGD	95%	99.87%
RMSprop	99.09%	99.98%

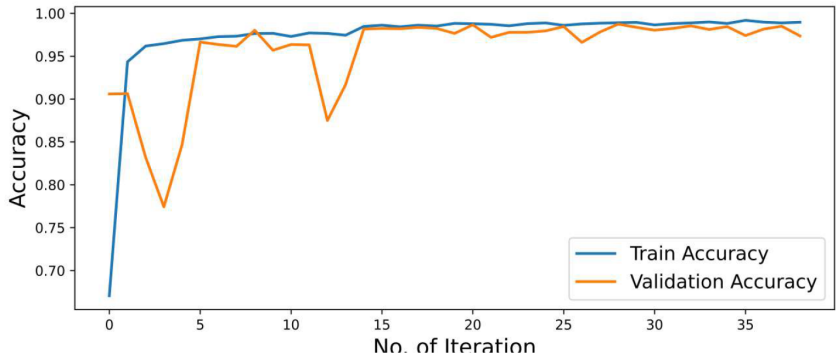


FIGURE 9
Accuracy graph including Mn/Mg deficiency class.

validating its standard and reliability. Additionally, our model is highly efficient. It's half the size of the ShuffleNet pre-trained model but still achieves almost similar accuracy. Compared to other transfer learning models, it has the fewest parameters, with some popular models having up to eight times more parameters and larger sizes. This means our model runs fast, making it perfect for various mobile devices. These results suggest that our model is not only effective for diagnosing plant diseases but also has great potential for use by farmers on a large scale. Therefore, its economic feasibility and exceptional performance collectively contribute to its greatness, making it a valuable asset for agricultural practitioners seeking advanced yet accessible solutions.

The proposed CNN-SVM model's significance is also evaluated against several related research works, where it holds a notable position. Zhang et al. (2019) aimed to develop automatic image-based diagnostic methods for identifying cherry diseases using only two types of cherry leaves – diseased and healthy. The research achieved a high accuracy rate, outperforming other works, and demonstrated its superiority through ROC curves, comparing with various machine learning models. However, they encountered challenges in creating a lightweight model and explaining their model's visualization technique, such as Grad-CAM. Additionally, they lacked some evaluation metrics like classification reports, confusion matrix, and PR curve. Hang et al. (2019) proposed a model which was compared with numerous transfer learning models regarding accuracy, model size, and training time. Despite having the same number of classes as ours, the paper aimed to structure automatic cherry disease identification with two types of diseased cherry classes and one healthy class. Although the authors visualized the model's performance, the accuracy rate fell short of

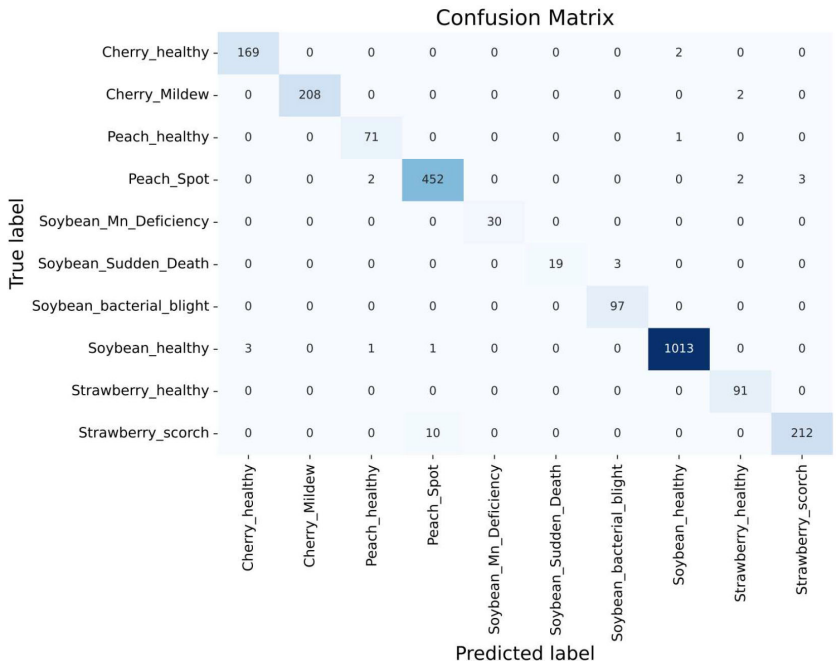


FIGURE 10
Confusion matrix including Mn/Mg deficiency class.

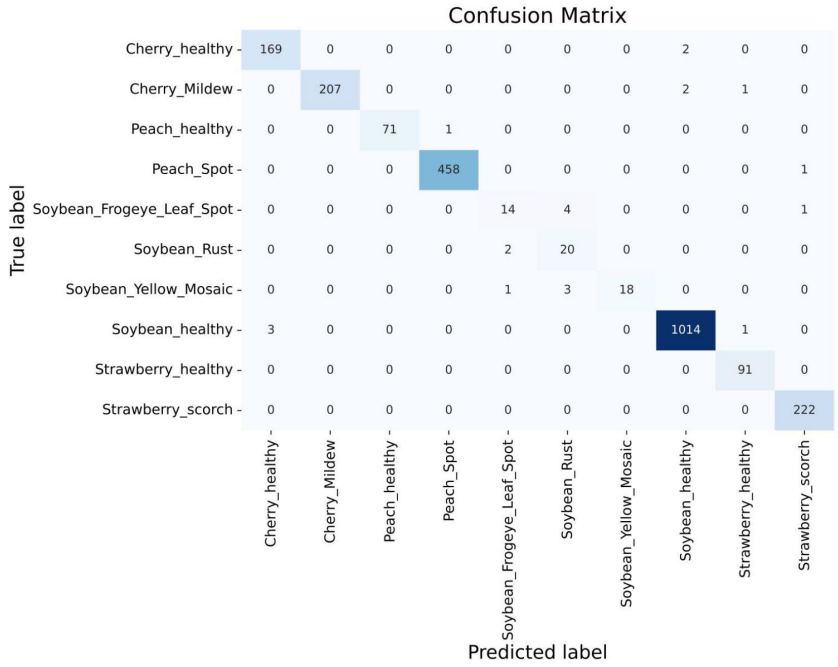


FIGURE 11
Confusion matrix including soybean frogeye spot and soybean rust classes.

expectations and they struggled to develop a lightweight model efficient for farmers.

Alosaimi et al. (2021) showcased impressive results through accuracy graphs, confusion matrices, classification reports and ROC curves, applied to 12 types of peach diseases in a CNN model. They have worked with several peach diseases, but they could also apply their model for the other crops. Besides, their accuracy rate was not as satisfactory as ours and their model lacked visualization technique. Akbar et al. (2022) proposed a novel lightweight and parameters-concerned model for classifying two types of peach leaves, with noticeable experimental outcomes providing various comparisons of performance evaluation metrics and transfer learning models. But, while the accuracy was high, it couldn't maintain the same accuracy as our proposed model obtained with ten classes. Besides, they could increase the number of peach classes or the types of crops and explain the model by using explainable AI. To sum up, they could increase the dataset by providing more number of classes and trying to achieve the same accuracy as before.

Xiao et al. (2020) proposed research that was conducted with two datasets, utilizing original and feature images to detect strawberry diseases like leaf blight, gray mold, and powdery mildew. Their customized CNN model, based on ResNet50 achieved 99.6% accuracy, but they could have explored more evaluation metrics instead of modifying a transfer learning model. Moreover, they also needed to focus on the number of parameters as ResNet50 has a higher number of parameters. In summary, they have achieved a higher accuracy but with a heavyweight transfer learning model as it has a higher number of parameters. Dhivya and Shanmugavadivu (2021) showed an impressive comparison among various CNN models where EfficientNet-B3 achieved a remarkable outcome than others. However, they haven't proposed their own built model to compare with various transfer learning models. Moreover, the research paper does not mention the use of visualization techniques like explainable AI and the authors could do the same research for more crops instead of only strawberries. Besides, the authors didn't show some performance evaluation matrices like the ROC curve, PR curve and Confusion matrix. Another drawback of this research is that the research did not mention the lightweightness of the models.

Wu et al. (2023), the researchers proposed an improved ConvNeXt model with an attention module for generating feature maps at different depths, achieving an accuracy of 85.42% on three types of soybean leaves. Though the number of classes was limited, the accuracy was unsatisfactory, suggesting room for improvement. Jadhav et al. (2019), the authors used SVM and KNN algorithms to classify four types of soybean leaf diseases, achieving 87.3% and 83.6% accuracy, respectively. However, their accuracy value seems to be a limitation due to the use of a small dataset and only one type of crop. Wallelign et al. (2018) managed to achieve 99.32% accuracy with four classes of soybean leaves using a CNN model based on the LeNet architecture, with visualization of the model's outcome.

TABLE 6 Analysis of applying CNN-SVM over various datasets.

No of Classes	Total Images	Accuracy	MCC
10 (Two new classes - Soybean Rust and Frogeye Spot)	11,532	99.04%	0.98
10 (Replaced Yellow Mosaic with Mn/Mg deficient class)	11,957	98.74%	0.98
10 (Proposed dataset)	11,524	99.09%	0.98

Although the dataset size was satisfactory, the limited number of disease types was a drawback. Overall, the limitations in existing research, particularly the absence of a combined CNN-SVM model with the Grad-CAM visualization technique have been noticed.

In summary, the discussion highlights our proposed CNN-SVM model having both the advancements and the remaining challenges in automating disease identification in crops. From Table 7, our proposed model has mitigated all the research gaps of the existing works mentioned above and showed its acceptance for real-world-based plant disease detection.

6 Explainable-AI application

Significant efforts are underway to enhance the interpretability and comprehensibility of deep learning, particularly in applications related to the imaging of plant diseases. Ensuring a clear understanding of deep learning models is crucial in such contexts. The Gradient Weighted Class Activation Mapping (Grad-CAM) method, introduced by Selvaraju et al. (2017) plays a pivotal role in elucidating deep learning models as an explainable AI application. Grad-CAM produces a visually interpretable representation of any intricately connected neural network, thereby aiding in model comprehension during task detection or prediction. In the majority of cases, Grad-CAM was primarily applied to the final convolutional layer. Grad-CAM produces a heatmap, highlighting essential areas within an image by leveraging gradients derived from the target class in the last convolutional layer. The regions used for classification become apparent when superimposing this heatmap onto the original image. In this research, Grad-CAM was utilized to asses if leaf sections in the input image significantly influence the diagnostic process to visually depict the diagnosis. The calculation entails evaluating the target class gradient on each feature map and averaging them to determine the relative significance of each map. The computation involves determining a weighted sum of activations from each feature map, where the importance of each is associated with the input image, resulting in the visualization. Grad-CAM proves to be an effective technique that does not hinder performance, as it doesn't necessitate any additional custom components Fujita et al. (2018). As depicted in Figure 12, the proposed model utilized Grad-CAM for detection techniques on a basic image received as input.

7 Conclusions

Crop diseases are a major threat to food security, but their rapid identification remains difficult in many parts of the world due to the lack of the necessary infrastructure. The rise in global smartphone usage, along with advancements in computer vision powered by deep learning, has opened doors to smartphone-enabled disease diagnosis. To accomplish this goal, in the proposed work, a 2D CNN-based model has been constructed to detect the 6 disease classes and 4 healthy classes in Peach, Cherry, Soybean, and Strawberry. The suggested 2D CNN-based architecture has four convolutional and four max-pooling layers, two fully connected layers, two dropout layers, and batch normalization in each layer makeup. The suggested model uses

TABLE 7 Comparison of existing related works.

Reference	Method	Accuracy	Precision	Recall	F1-Score	Classes	Plant
Zhang et al. (2019)	GoogleNet	99.6%	–	–	–	2	Cherry
Hang et al. (2019)	VGG16	91.7%	–	–	–	10	Apple, Cherry, Corn
Alosaimi et al. (2021)	CNN	94%	94%	94%	94%	12	Peach
Akbar et al. (2022)	LWNet	99%	100%	99%	99%	2	Peach
Xiao et al. (2020)	ResNet50	99.6%	–	–	–	3	Strawberry
Dhivya and Shanmugavadivu (2021)	EfficientNet-B3	97%	98%	97%	97%	2	Strawberry
Wu et al. (2023)	Improved ConvNeXt	85.42%	88.35%	88.44%	88.37%	3	Soybean
Jadhav et al. (2019)	SVM and KNN classifiers	83.6%, 87.3%	–	–	–	4	Soybean
Wallelign et al. (2018)	LeNet	99.32%	99%	99%	99%	4	Soybean
Proposed model	CNN-SVM	99.09%	99%	99%	99%	10	Peach, cherry, soybean, strawberry

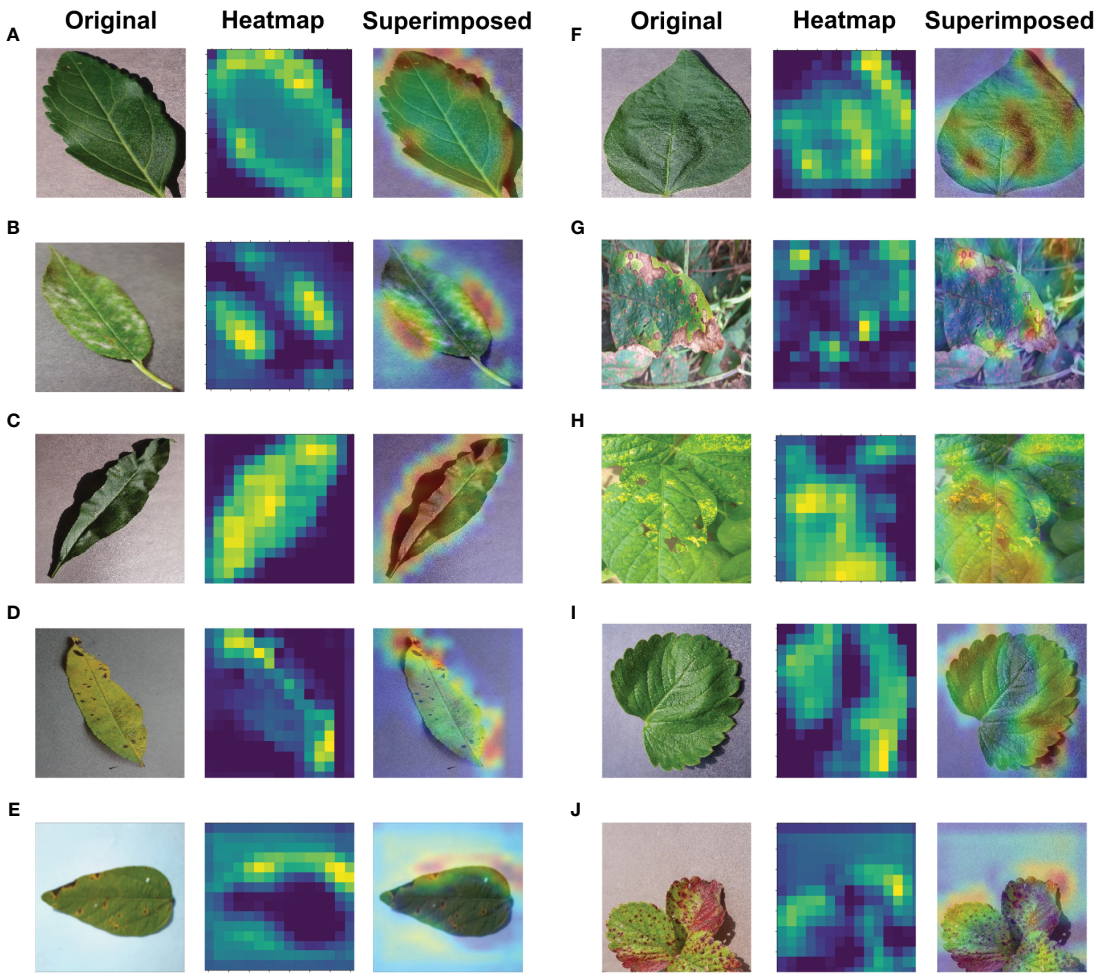


FIGURE 12
Application of Explainable-AI for **(A)** Cherry Healthy Leaf **(B)** Cherry Mildew **(C)** Peach Healthy Leaf **(D)** Peach Spot **(E)** Soybean Bacterial Blight **(F)** Soybean Healthy **(G)** Soybean Sudden Death **(H)** Soybean Yellow Mosaic **(I)** Strawberry Healthy **(J)** Strawberry Scorch leaves.

less storage capacity and has fewer parameters than transfer learning models because of this kind of shallow structure, which has surpassed heavyweight transfer learning architectures (VGG16, VGG19, and Inception V3) and lightweight transfer learning architectures (MobileNet, MobileNetV2, DenseNet and ShuffleNet) which have an average accuracy range from 54% to 97%. Along with the transfer learning models, the model's performance has also been evaluated using the confusion matrix, ROC curve, AUC score, and Matthews Correlation Coefficient. The model also showed an impressive performance over various datasets. The outcome shows that the model has achieved a high level of performance that will assist plant doctors and farmers in accurately identifying a variety of diseases affecting cherry, peach, strawberry, and soybean plants. This can help plant doctors take appropriate action to prevent the disease and save money for the farmers. Additionally, this can benefit the economy of the nation. Because the suggested model has significantly fewer parameters than transfer learning models, it requires between three and four times less storage space than transfer learning models. This concept can be easily applied to smartphones and other devices due to its lightweight structure. Grad-CAM class activation maps and a heatmap were created to visualize the detection the trained model was able to achieve to symbolize the area in charge of classification. However, there can be several obstacles and limitations when implementing a model in real-world situations. Besides, our model should have classified Mn/Mg deficient images and Soybean sudden death images without any misclassification although both of the classes have very similar type of features between them. In the future, we have a plan to increase the classification rate more and remove the collision between those two classes. Furthermore, we are planning to explore different hybrid models to handle upcoming challenges better.

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

RP: Writing – original draft, Software, Methodology, Data curation, Conceptualization, Writing – review & editing, Investigation. AM: Writing – review & editing, Writing – original draft, Methodology, Investigation, Data curation, Conceptualization. HP: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Data curation, Conceptualization. SM: Writing – review & editing, Writing – original draft, Methodology, Data curation. MN: Writing – review & editing, Validation, Supervision, Investigation. AK: Writing – review & editing, Validation, Supervision, Project administration, Funding acquisition, Formal analysis. MA: Writing – review & editing, Validation, Supervision, Project administration, Funding acquisition, Formal analysis.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. The paper publication has been possible with the support of Qatar National Library funding and the work was supported by Qatar Research, Development and Innovation (QRDI) fund HSREP05-1012-230035.

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

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2024.1412988/full#supplementary-material

References

Abbas, I., Liu, J., Amin, M., Tariq, A., and Tunio, M. H. (2021). Strawberry fungal leaf scorch disease identification in real-time strawberry field using deep learning architectures. *Plants* 10, 2643. doi: 10.3390/plants10122643

Akbar, M., Ullah, M., Shah, B., Khan, R. U., Hussain, T., Ali, F., et al. (2022). An effective deep learning approach for the classification of bacteriosis in peach leave. *Front. Plant Sci.* 13, 4723. doi: 10.3389/fpls.2022.1064854

Alosaimi, W., Alyami, H., and Uddin, M.I. (2021). Peachnet: Peach diseases detection for automatic harvesting. *Comput. Mater. Continua* 67. doi: 10.32604/cmc.2021.014950

Chicco, D., and Jurman, G. (2023). The matthews correlation coefficient (mcc) should replace the roc auc as the standard metric for assessing binary classification. *BioData Min.* 16, 4. doi: 10.1186/s13040-023-00322-4

Chicco, D., Starovoitov, V., and Jurman, G. (2021). The benefits of the matthews correlation coefficient (mcc) over the diagnostic odds ratio (dor) in binary classification assessment. *IEEE Access* 9, 47112–47124. doi: 10.1109/ACCESS.2021.3068614

Clark, M. F., and Bar-Joseph, M. (1984). Enzyme immunosorbent assays in plant virology. *Methods Virol.* 7, 51–85. doi: 10.1016/B978-0-12-470207-3.50009-7

Dhivya, S., and Shanmugavadivu, R. (2021). Performance evaluation of image processing filters towads strawberry leaf disease. *Turkish J. Comput. Math. Educ. (TURCOMAT)* 12, 3776–3784. doi: 10.17762/turcomat.v12i11.6487

Ebrahimi, M. A., Khoshtaghaza, M. H., Minaei, S., and Jamshidi, B. (2017). Vision-based pest detection based on svm classification method. *Comput. Electron. Agric.* 137, 52–58. doi: 10.1016/j.compag.2017.03.016

Fujita, E., Uga, H., Kagiwada, S., and Iyatomi, H. (2018). A practical plant diagnosis system for field leaf images and feature visualization. *Int. J. Eng. Technol.* 7, 49–54. doi: 10.14419/ijet.v7i4.11

Gui, J., Hao, L., Zhang, Q., and Bao, X. (2015). A new method for soybean leaf disease detection based on modified salient regions. *Int. J. multimed. ubiquitous Eng.* 10, 45–52. doi: 10.14257/ijmue

Hang, J., Zhang, D., Chen, P., Zhang, J., and Wang, B. (2019). Classification of plant leaf diseases based on improved convolutional neural network. *Sensors* 19, 4161. doi: 10.3390/s19194161

Husaini, A. M., and Neri, D. (2016). *Strawberry: growth, development and diseases* (CABI). doi: 10.1079/9781780646633.0000

Jadhav, S. B., Udup, V. R., and Patil, S. B. (2019). Soybean leaf disease detection and severity measurement using multiclass svm and knn classifier. *Int. J. Electric. Comput. Eng.* 9, 4092. doi: 10.11591/ijece.v9i5

Maas, J. L. (2012). "Strawberry diseases and pests-progress and problems," in *VII International Strawberry Symposium*, Leuven, Belgium. Vol. 1049. 133–142.

Pan, L., Zhang, W., Zhu, N., Mao, S., and Tu, K. (2014). Early detection and classification of pathogenic fungal disease in post-harvest strawberry fruit by electronic nose and gas chromatography– mass spectrometry. *Food Res. Int.* 62, 162–168. doi: 10.1016/j.foodres.2014.02.020

Peyal, H. I., Nahiduzzaman,, Pramanik, A. H., Syfullah, K., Shahriar, S. M., Sultana, A., et al. (2023). Plant disease classifier: Detection of dual-crop diseases using lightweight 2d cnn architecture. *IEEE Access* 11, 110627-110643. doi: 10.1109/ACCESS.2023.3320686

Sharma, R., Singh, A., Jhanjhi, N. Z., Masud, M., Jaha, E. S., Verma, S., et al. (2022). Plant disease diagnosis and image classification using deep learning. *Comput. Mater. Continua* 71. doi: 10.32604/cmc.2022.020017

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. (2018). How does batch normalization help optimization? *Adv. Neural Inf. Process. Syst.* 31.

Schaad, N. W., and Frederick, R. D. (2002). Real-time pcr and its application for rapid plant disease diagnostics. *Can. J. Plant Pathol.* 24, 250–258. doi: 10.1080/07060660209507006

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 localization," in *Proceedings of the IEEE international conference on computer vision*. headquarter in Piscataway, New Jersey, USA. 618–626.

Shoaib, M., Shah, B., Sayed, N., Ali, F., Ullah, R., and Hussain, I. (2023). Deep learning for plant bioinformatics: an explainable gradient-based approach for disease detection. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1283235

Skrovankova, S., Sumczynski, D., Mlcek, J., Jurikova, T., and Sochor, J. (2015). Bioactive compounds and antioxidant activity in different types of berries. *Int. J. Mol. Sci.* 16, 24673–24706. doi: 10.3390/ijms161024673

Tripathy, A., Agrawal, A., and Kumar Rath, S. (2015). Classification of sentimental reviews using machine learning techniques. *Proc. Comput. Sci.* 57, 821–829. doi: 10.1016/j.procs.2015.07.523

Wallelign, S., Polceanu, M., and Buche, C. (2018). "Soybean plant disease identification using convolutional neural network," in *FLAIRS conference*. Palo Alto, California, USA. 146–151.

Wu, Q., Ma, X., Liu, H., Bi, C., Yu, H., Liang, M., et al. (2023). A classification method for soybean leaf diseases based on an improved convnext model. *Sci. Rep.* 13, 19141. doi: 10.1038/s41598-023-46492-3

Xiao, J.-R., Chung, P.-C., Wu, H.-Y., Phan, Q.-H., Yeh, J.-L. A., and Hou, M. T.-K. (2020). Detection of strawberry diseases using a convolutional neural network. *Plants* 10, 31. doi: 10.3390/plants10010031

Yao, N., Ni, F., Wu, M., Wang, H., Li, G., and Sung, W.-K. (2022). Deep learning-based segmentation of peach diseases using convolutional neural network. *Front. Plant Sci.* 13, 876357. doi: 10.3389/fpls.2022.876357

Yu, M., Ma, X., Guan, H., Liu, M., and Zhang, T. (2022). A recognition method of soybean leaf diseases based on an improved deep learning model. *Front. Plant Sci.* 13, 878834. doi: 10.3389/fpls.2022.878834

Zhang, K., Zhang, L., and Wu, Q. (2019). Identification of cherry leaf disease infected by podosphaera pannosa via convolutional neural network. *Int. J. Agric. Environ. Inf. Syst. (IJAEIS)* 10, 98–110. doi: 10.4018/IJAEIS

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 11 July 2024
DOI 10.3389/fpls.2024.1404772

OPEN ACCESS

EDITED BY
Mohsen Yoosefzadeh Najafabadi,
University of Guelph, Canada

REVIEWED BY
Parvathaneni Naga Srinivasu,
Prasad V. Potluri Siddhartha Institute of Technology, India
Parismita Sarma,
Gauhati University, India

*CORRESPONDENCE
Chengzhong Liu
liucz@gsau.edu.cn

RECEIVED 21 March 2024
ACCEPTED 18 June 2024
PUBLISHED 11 July 2024

CITATION
Sun K, Liu C, Han J, Zhang J and Qi Y (2024) Phenotypic detection of flax plants based on improved Flax-YOLOv5. *Front. Plant Sci.* 15:1404772. doi: 10.3389/fpls.2024.1404772

Phenotypic detection of flax plants based on improved Flax-YOLOv5

Kai Sun[1], Chengzhong Liu[1]*, Junying Han[1], Jianping Zhang[2] and Yanni Qi[2]

[1]College of Information Science and Technology, Gansu Agricultural University, Lanzhou, China, [2]Crop Research Institute, Gansu Academy of Agricultural Sciences, Lanzhou, China

Accurate detection and counting of flax plant organs are crucial for obtaining phenotypic data and are the cornerstone of flax variety selection and management strategies. In this study, a Flax-YOLOv5 model is proposed for obtaining flax plant phenotypic data. Based on the solid foundation of the original YOLOv5x feature extraction network, the network structure was extended to include the BiFormer module, which seamlessly integrates bi-directional encoders and converters, enabling it to focus on key features in an adaptive query manner. As a result, this improves the computational performance and efficiency of the model. In addition, we introduced the SIoU function to compute the regression loss, which effectively solves the problem of mismatch between predicted and actual frames. The flax plants grown in Lanzhou were collected to produce the training, validation, and test sets, and the detection results on the validation set showed that the average accuracy (mAP@0.5) was 99.29%. In the test set, the correlation coefficients (R) of the model's prediction results with the manually measured number of flax fruits, plant height, main stem length, and number of main stem divisions were 99.59%, 99.53%, 99.05%, and 92.82%, respectively. This study provides a stable and reliable method for the detection and quantification of flax phenotypic characteristics. It opens up a new technical way of selecting and breeding good varieties.

KEYWORDS

flax, YOLOv5, target detection, phenotypic data, variety breeding

1 Introduction

Flax (*Linum usitatissimum*) is one of the most important oil and fiber crops in the world. Flax is mainly divided into oil flax, fiber flax, and dual-purpose oil and flax varieties according to their uses (Zhang et al., 2011). Recently, the results of studies emphasizing the anticancer properties of substances present in flaxseed and oil have attracted great attention (Praczyk and Wielgusz, 2021) and are widely cultivated worldwide (Kauser et al., 2024). Selection and breeding of flax varieties are crucial for progress in flax production (Gong et al., 2020). Obtaining the phenotypic data required for flax breeding is the basis of breeding; only rapid

and accurate access to flax plant phenotypic data and the breeding of flax varieties will have a qualitative leap. The traditional acquisition of flax phenotypic data is through manually counting the number of flax fruits and the number of main stems divided into stems, measuring the plant height and main stem length, and manually recording data; this traditional method of flax production has made a significant contribution to the progress of flax production, but with the advancement of science and technology, these methods have become more and more inefficient and expensive. As a result, these traditional methods often fail to meet the stringent requirements of modern breeding practices. To address these challenges, there is an urgent need to explore innovative techniques that are more efficient, cost-effective, and compatible with contemporary sub-breeding acquisition of data.

Currently, computer vision technology is widely used in agriculture and has made great progress in the accuracy and efficiency of extracting plant phenotypic data. Currently, there are two main detection methods for obtaining plant phenotypic data: traditional target detection methods and target detection methods based on deep learning (Zhang et al., 2023). Among them, the traditional target detection process is more complex, requiring multiple steps to be completed together and time-consuming, with higher requirements for images, different algorithms for different detection objects, and greater difficulty in extracting different information at the same time; deep learning has a powerful feature extraction capability, which can make up for the shortcomings of the traditional methods, and therefore, more and more researchers are using it for agricultural target detection.

In recent years, many scholars have begun to apply deep learning in the field of agriculture, such as identifying plants, pests, and diseases, to improve crop yields. Zhu et al. (2024) proposed a CBF-YOLO network for the detection of common soybean pests in complex environments. Pei et al. (2022) proposed a maize field weed detection framework based on crop row pretreatment and improved YOLOv4 in UAV images. Li et al. (2023) proposed an apple leaf disease detection method based on the improved YOLOv5s model. Bai et al. (2024) proposed an improved YOLO algorithm to detect the flowers and fruits of strawberry seedlings. Wang et al. (2024) developed a new deep learning network, YOLO-DCAM, which effectively facilitates single-wood detection in complex scenarios. Du et al. (2023) proposed a method for detecting strawberry fruit planted in fields under different shade levels. Su et al. (2023) proposed an improved YOLOv5-SE-BiFPN model, which could more effectively detect brown spot lesion areas in kidney beans. Zhang et al. (2024) proposed a multi-task learning method named YOLOMS for mango recognition and rapid location of major picking points.

YOLO series is a single-stage algorithm that ensures high precision and faster speed, especially in the GPU environment, and real-time detection can be realized. Due to its excellent performance, it has achieved great results in the extraction of plant phenotype data and the application of detection objects. Guo et al. (2022) proposed a method to obtain phenotypic parameters of soybean plants based on Re-YOLOv5 and detection region search algorithms, and the results showed that the average absolute errors of plant height, stem node count, and soybean branch count were 2.06 cm, 1.37 cm, and 0.03 cm, respectively. The results were better, and a specialized black box for filming was developed, but this is time-consuming in the face of a large number of films to obtain phenotypic data and does not apply to realistic breeding requirements. Chen et al. (2024) proposed an efficient, fast, and real-time seedling counting method for cabbages, which replaced the C2f block in the main stem network of YOLOv8n with a Swin-conv block and added a ParNet block to both the main stem and neck portions of the network. ParNet attention modules were added to the neck section to accurately track cabbage seedlings in the field and count them using an unmanned aerial vehicle (UAV), achieving 90.3% mAP50–95, but its recognition progress needs to be further improved. She et al. (2022) introduced the ECA attention mechanism into the YOLOv5s model to improve the accuracy of trap vial detection and counting, but the recognition accuracy needs to be further improved. Gao et al. (2022) proposed the YOLOv4-tiny network combined with the channel spatial reliability discriminant correlation filtering (CSR-DCF) algorithm for training, and the correlation coefficient R^2 between apple number prediction and manual counting was 0.9875. The counting accuracy of the orchard video is 91.49%, so the accuracy of fruit recognition in the video needs to be further improved.

While deep learning has applications in acquiring plant phenotypic data, it has received limited attention for the accurate detection of organs in flax plants. In real-world detection scenarios, complex flax fruit overlap and branching pose significant challenges to fruit occlusion. This often leads to incomplete detection, as existing models ignore occluded flax fruits. In addition, less characterization of flax plant main stem length and main stem branching increases the complexity of identification. In addition, the shapes of flax fruits, plant heights, industrial lengths, and main stem meristems varied, increasing the difficulty of designing a fusion model for identification. To solve these problems and improve the accuracy of phenotypic information, this study proposes a pioneering method to recognize phenotypic organs of flax plants, and this technological breakthrough is expected to improve the efficiency of breeding and open up a new way for precision agriculture. The main contributions are summarized as follows.

(1) Establishing a new flax plant dataset.

(2) Deepening the original YOLOv5x network layer and adding the BiFormer attention mechanism to its network layer significantly improve the extraction of flax features and reduce the risk of overfitting (Yang et al., 2023). In addition, the SIoU loss function replaces the original CIoU loss function, which effectively solves the problem of mismatch between the prediction and the actual bounding box and improves the accuracy of the model (Qian et al., 2024).

(3) After the model is fully trained, it is loaded onto the test set for identification and compared with the manual test data to obtain a good correlation. The model has been embedded into PC software and put into use.

The rest of the paper is organized as follows. Section 2 discusses the methods involved in the flax plant dataset, the improved Flax-ylolv5, the experimental setup, and the evaluation criteria. The conclusions are explained and discussed in Section 3. The design of

the improved Flax-YOLOv5 application software is presented in Section 4. Section 5 summarizes the conclusions of the paper.

2 Materials and methods

2.1 Phenotypic dataset of flax plants

The experimental study used manually collected samples of mature and intact plants of flax from the Lanzhou Flax Planting Base of Gansu Provincial Academy of Agricultural Sciences. A total of 630 flax plants were collected to ensure phenotypic diversity. These samples were carefully selected to include a range of plant types, such as single main-stem split-stem flax plants, multiple main stem split-stem flax plants, flax plants with different numbers of fruits, and plants with complex branching patterns.

Images were captured using an MV-HS2000GM/C2 industrial camera. To eliminate potential interference from natural light, which can lead to exposure problems and complex backgrounds, the shoot was conducted indoors. A LED light source was used to provide supplemental lighting during the shoot, while a black light-absorbing cloth was used as a backdrop to simplify the test background and minimize interference. Additionally, the branches of the flax plants were hand-arranged to prevent excessive fruit overlap. To ensure accurate measurement of plant height and main stem length, the flax plant was placed horizontally below the camera lens. The camera height was set to 140 cm, and the image resolution was set to 5,472 pixels × 3,000 pixels to capture high-quality images for subsequent analysis.

2.2 Labeling of phenotypic feature datasets

The image features obtained were carefully measured and annotated for specific phenotypic traits, including the number of flax fruits, plant height, length of the main stem, and number of divisions within the main stem. Length measurements were made in centimeters with accuracy maintained to one decimal place. Considering the irregularity of traits such as number of flax fruits, plant height, length of the main stem, and branching of the main stem, we aimed to minimize measurement errors. Therefore, all phenotypic traits of flax plants were labeled to represent the average of three separate measurements. The labeling process utilized a dedicated labeling tool to generate the dataset in text format. The number of fruits on the flax plant, recorded as complete fruits, was labeled as “flax”. Plant height, which represents the vertical extension of the plant from root to tip, was labeled as “height”. The length of the main stem, i.e., the distance from the root to the first main branch, is labeled as “length”. In addition, the number of divisions, representing the number of branches emanating from the prominent main stem, was labeled “n” (n = 1, 2, …), and the maximum number of main stem divisions observed in a single plant was six.

2.3 Data expansion

A traditional data enhancement method was used to enrich the diversity of flax plant image samples, thus enhancing the generalization ability and robustness of the model. The enhancement process was carried out in five different ways: downward brightness adjustment, mirror operation, rotating the image, a combination of mirroring and brightness reduction, and a combination of mirroring and noise addition. Figure 1 shows an illustrative example of this data enhancement process, which demonstrates the effectiveness of these techniques in generating a diverse and representative sample of images to be used for model training.

2.4 Original YOLOv5x

As shown in Figure 2, the original network structure of YOLOv5x is divided into an input network, a backbone network, a neck network, and a head network. The input integrates mosaic data enhancement, adaptive anchoring, and adaptive image scaling of 1.33 depth and 1.25 width. The backbone is a convolutional

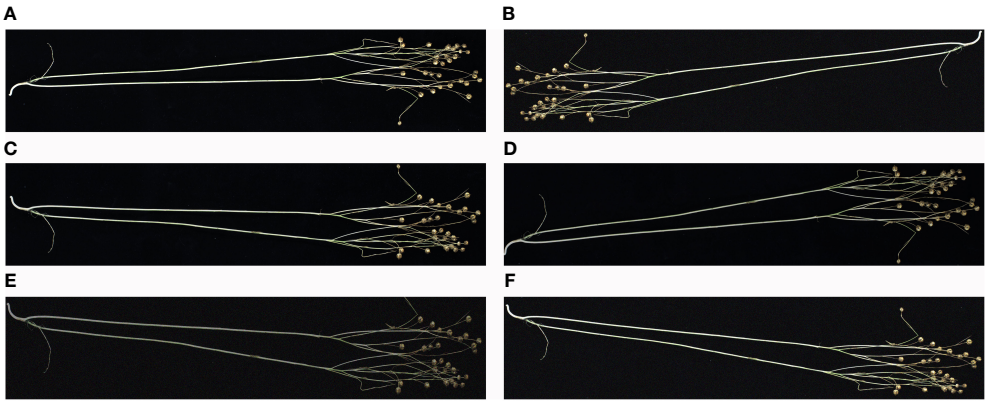


FIGURE 1
Example of data enhancement: **(A)** original, **(B)** rotated, **(C)** mirrored, **(D)** reduced brightness, **(E)** mirrored and reduced brightness, and **(F)** mirrored and added noise. The image has been cropped for ease of viewing.

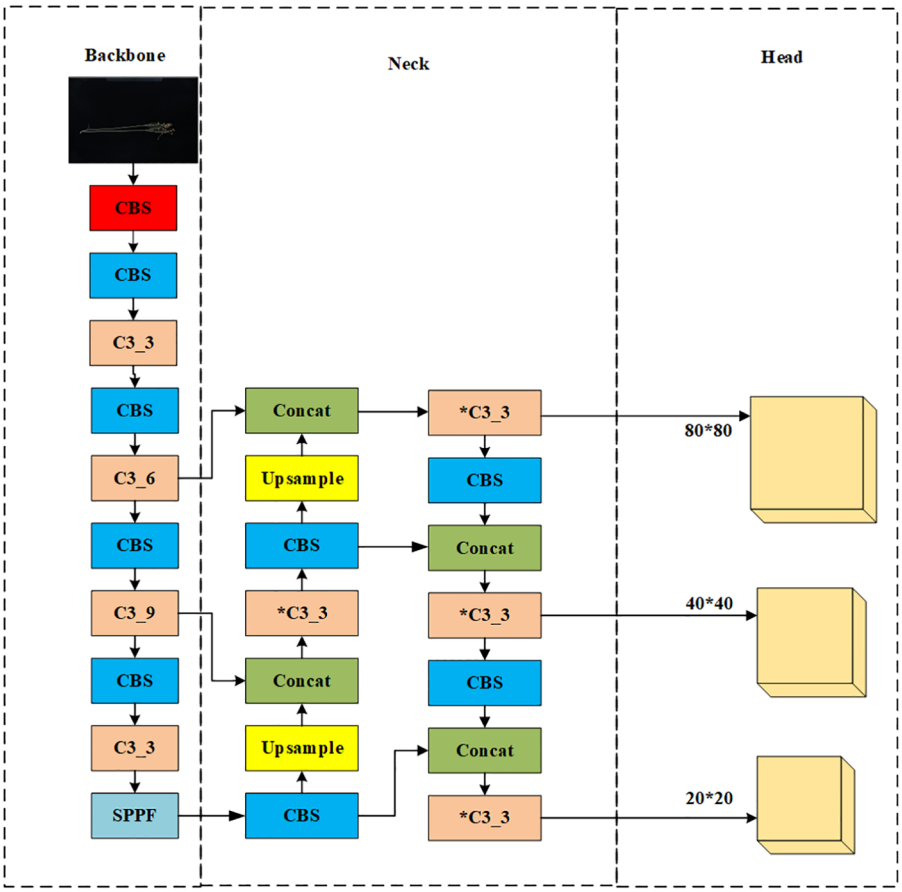


FIGURE 2
YOLOv5x model structure.

neural network that accumulates fine-grained images and generates feature maps. It contains CBS, C3, and Spatial Pyramid Pooling (SPPF) for feature extraction as shown in Figure 3. The YOLOv5x neck part uses a PANet structure for multi-scale feature fusion. The neck network combines the feature maps collected by the backbone network and then passes the integrated feature maps to the head network, which generates predictions from the anchor box for target detection (Rahman et al., 2022). The head network outputs a vector containing the class probability of the target, the target score, and the location of the bounding box around the target.

2.5 Improved Flax-YOLOv5

To accurately identify the phenotypic organs of flax plants, a Flax-YOLOv5 network structure model with high detection accuracy and detection speed was proposed. First, in the Flax-YOLOv5 network shown in Figure 3, the adaptive image scaling of Flax-YOLOv5 is 1.0 times depth and 1.0 times width. This adjusts the depth and width of the network to meet the needs of different scenes and improve detection accuracy.

Second, the Flax-YOLOv5 backbone network is improved based on the inheritance of the YOLOv5x backbone network. In the improvement of Flax-YOLOv5, the BiFormer module is added after the CBS module at layer 10 in the original YOLOv5x necking network. The CBS module, Upsample, Concat, and C3 modules are added at the end of the 18th layer, and the CBS, Concat, and C3 layers are added at the end of the 28th layer to improve the model's ability to extract target features.

Finally, the improved Flax-YOLOv5 head network in Figure 3 generates feature maps with sizes of 160 × 160, 80 × 80, 40 × 40, and 20 × 20 with different scale target detection; the improved network model is named Flax-YOLOv5, and its structure is shown in Figure 3.

Flax-YOLOv5 is divided into three parts. The backbone is used for feature extraction of input Flax plant images, the Neck is used for feature fusion of acquired feature mappings, and the Head is used for regression prediction. BiFormer is introduced into the feature fusion network Neck to improve the feature extraction capability of the model. Second, the SIoU function is introduced into the output Head to calculate the regression loss and improve the convergence ability of the model. Among them, the CBS module is a basic convolutional neural network module, used to extract and transmit image features; it is composed of Conv (CONvolution layer), BN (Batch Normalization layer), and SiLU (activation function) in three parts. The Conv layer is responsible for the convolution operation of the input feature graph to extract higher-level features. The BN layer is used to normalize the data, which helps accelerate training and improve the performance of the

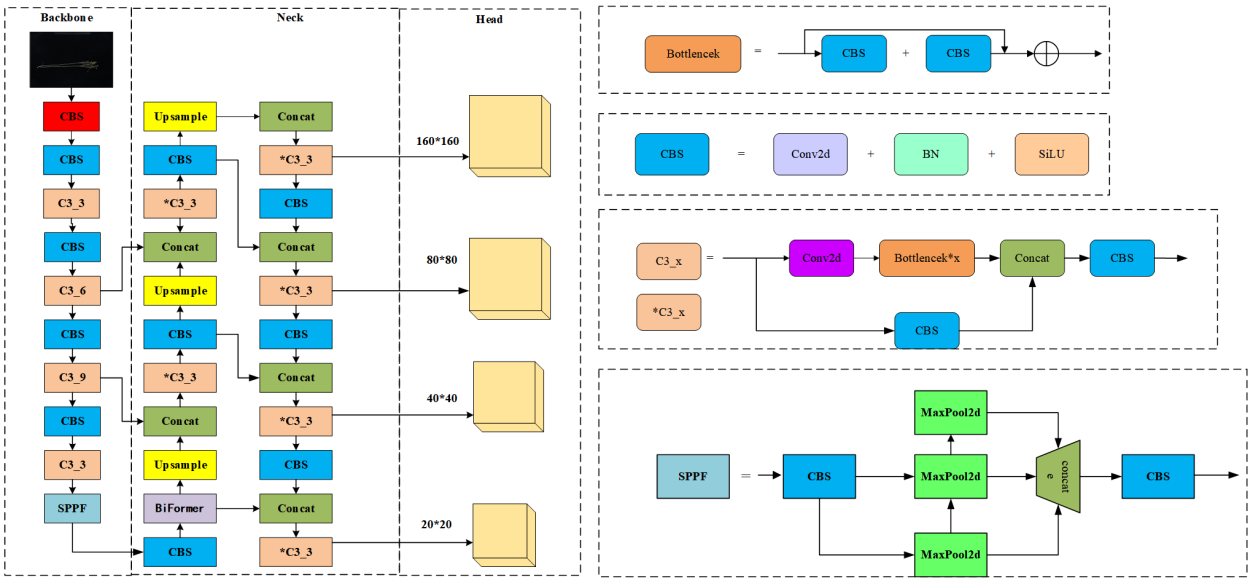


FIGURE 3
Flax-YOLOv5 model structure.

model. SiLU (Sigmoid-weighted Linear Unit) is an activation function to increase the non-linearity of the model. The C3_x module is composed of a series of multiple residual network structures. The inner Bottleneck module can be programmed to divide C3_x into two different structures, which are applied in the Backbone network and Neck network. The outer layer of the C3_x module connects to the CBS module to form a large residual edge. These residual components enhance the feature extraction capability of convolutional networks, and the stacking of residual blocks solves the difficult balance between network depth and gradient. C3_3 indicates that the C3 module has three Bottleneck modules. The SPPF module is an improved version of the Spatial Pyramid Pooling (SPP) module. SPP module is mainly used for image recognition and target detection, which can extract and encode image features at different scales, re-scale input images of any size to a fixed size, and generate fixed-length feature vectors. The SPPF module changes the parallel structure of SPP to a serial structure, which significantly reduces the amount of computation and makes the speed faster. This improvement not only maintains the function of SPP but also significantly improves the speed.

2.5.1 BiFormer attention mechanism

In the original image, the flax fruit is a small target with fewer features in terms of main stem length and number of main stem branches. For better extraction of effective features, the BiFormer module is introduced. BiFormer focuses on a small number of relevant markers in a query-adaptive manner without distracting other irrelevant markers, thus providing good performance and high computational efficiency. BiFormer is used in the first stage using overlapping block embedding, and in the second stage through the fourth stage, it uses a block merging module to reduce the input spatial resolution while increasing the number of channels and then uses consecutive BiFormer blocks for feature transformation. Note that the relative position information is implicitly encoded at the beginning of each block using 3 × 3 deep convolution. Subsequently, the (Bi-level routing attention, BRA) module and the 2-layer Multi-Layer Perceptron (MLP) module with an expansion rate of e are sequentially applied for cross-positional relation modeling and position-by-position embedding, with the BiFormer attention mechanism shown in Figure 4 (Kong et al., 2023).

2.5.2 SIoU

YOLOv5x uses the CIoU loss function, which is a traditional loss function for target detection that relies on the aggregation of bounding box regression metrics and does not take into account the desired orientation mismatch between the real and predicted frames, resulting in slower convergence and lower efficiency. To solve this problem, the loss function SIoU is introduced in the improved model, which considers not only the overlap region, distance, and orientation but also the angle between the predicted frame and the true frame. The SIoU formula is defined by Equations 1–5, where IoU is the regular regression loss, Δ is the distance loss, Ω is the shape loss, B denotes the prediction frame, B^{gt} denotes the ground truth box, ω^{gt} and h^{gt} denote the width and height of the ground truth box, respectively, and ω and h denote the width and height of the prediction box. b and b^{gt} denote the centroid of the predicted truth box and the true truth box, respectively, and b_{cx}^{gt} and b_{cy}^{gt} denote the horizontal and vertical coordinates of the center of the ground truth box, respectively. b_{cx} and b_{cy} are the corresponding coordinates of the predicted box. θ is an adjustable parameter used to control how much to focus on the shape cost, which is set to 4 in this study (Zhang et al., 2024).

$$\mathrm{Loss}_{SIoU} = 1 - IoU + \frac{\Delta + \Omega}{2} \tag{1}$$

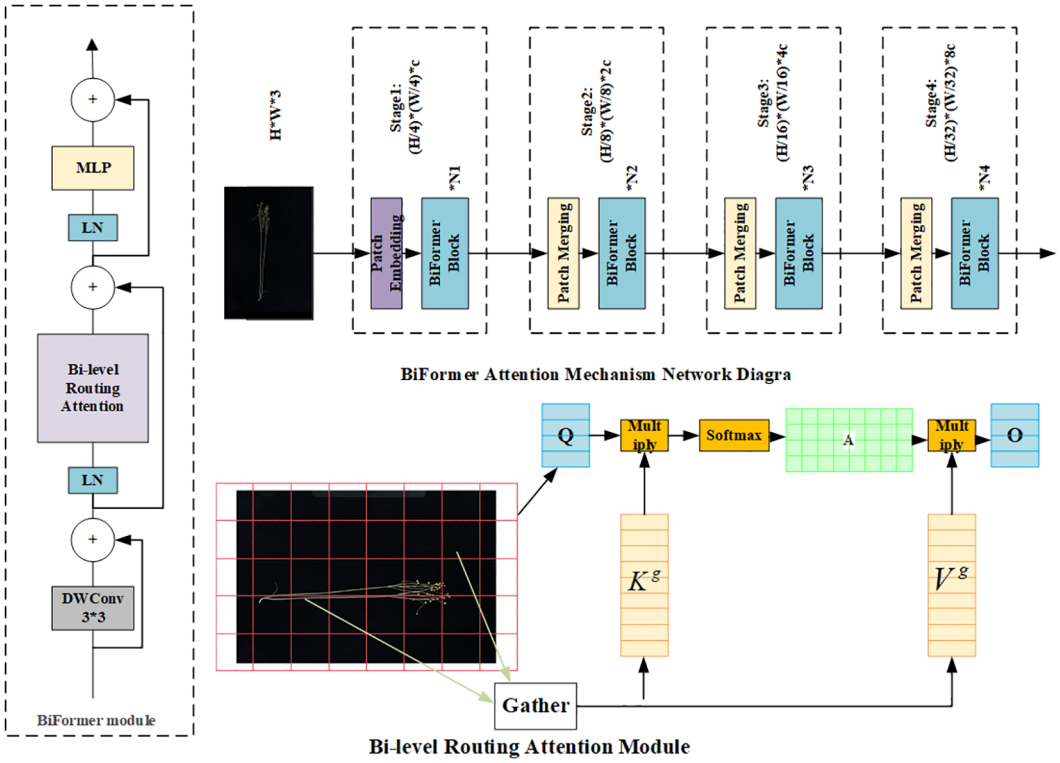


FIGURE 4
BiFormer attention mechanism architecture.

$$Iou = \frac{B \cap B^{gt}}{B \cup B^{gt}}, \beta = \frac{v}{(1-IOU)+v}, \quad v = \frac{4}{\pi^2}\left(tan^{-1}\frac{\omega^{gt}}{h^{gt}} - tan^{-1}\frac{\omega}{h}\right)^2 \tag{2}$$

$$\Delta = \sum_{t=x,y}(1 - e^{-\gamma p t}), \rho_x = \frac{b_{cx}^{gt} - b_{cx}}{c_w}, \quad \rho_x = \frac{b_{cy}^{gt} - b_{cy}}{c_h}, \gamma = 2 - \Lambda \tag{3}$$

$$\Lambda = 1 - 2 \ * \ sin^2(\arcsin(x) - \frac{\pi}{4}, x = \frac{c_h}{\sigma} = sin\alpha \tag{4}$$

$$\Omega = \sum_{t=\omega,h}(1 - e^{-\omega_t})^\theta, \omega_\omega = \frac{|\omega - \omega^{gt}|}{\max(\omega, \omega^{gt})}, \omega_h = \frac{|h - h^{gt}|}{\max(h, h^{gt})} \tag{5}$$

3 Improved model identification results and analysis

3.1 Experimental process

The specific steps of the experiment are shown in Figure 5.

As shown in Figure 5, data collection was carried out first. Of the 630 images collected, 100 were selected as the test set, and the remaining 530 images, that is, 3,180 images obtained through five data enhancement methods, were randomly divided into the training set and the verification set according to the ratio of 8:2, among which 2,544 were the training set. The verification set was 636 pieces. Second, the YOLO series model was trained on the training set. Finally, the model weight obtained from the above model on the training set was loaded onto the corresponding model and then tested on the test set. The optimal model was obtained by comparing the obtained results, and the optimal model was embedded in the developed software for the convenience of flax breeders.

3.2 Experimental environment

All models completed training on a server configured with CPU: Intel® Xeon® W-2123 CPU @ 3.60GHz and GPU: RTX 1080Ti with 8-GB video memory. The model training environments were PyTorch 1.10.0, python 3.8, and Cuda 10.2. The training parameters were 300 epochs (Ajayi et al., 2023); batch size was 4; the learning rate was set to 0.01, 0.937 momentum, 0.0005 weight decay, 0.2 IoU, 0.015 hue, 0.7 saturation, 0.4 lightness, 1.0 mosaic, 0.5 scale, and 0.1 translate; image input resolution was 640 pixels × 640 pixels; other original default parameters were used. The shooting instrument is shown in Figure 6.

3.3 Evaluation metrics

In this study, in addition to using the target detection algorithm to evaluate the precision and recall metrics, as well as the metrics for F1, we evaluated the Mean Average Precision (mAP) performance of

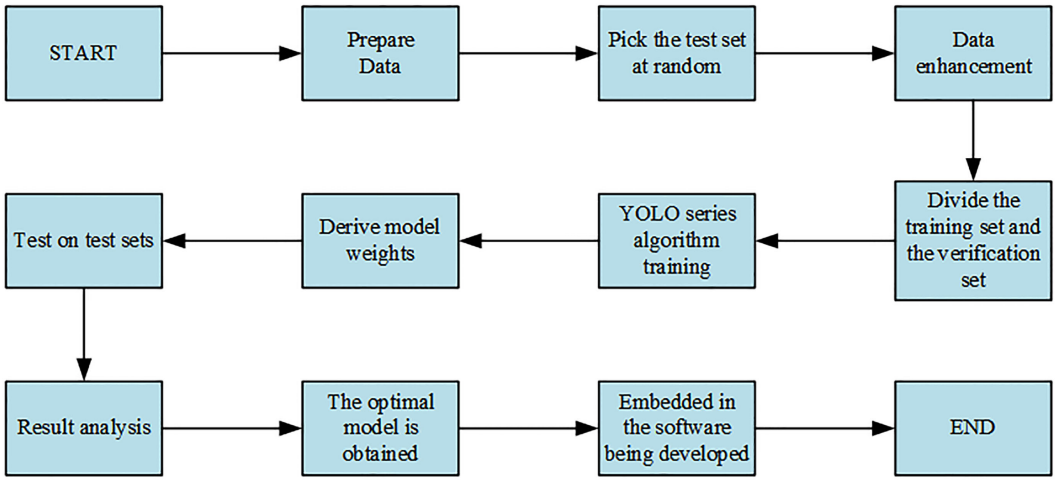


FIGURE 5
Experimental flowchart.

the model at an Intersection over Union (IoU) threshold of 0.5. In addition, to assess the accuracy of the phenotypic parameters extracted from flax plants using the Flax-YOLOv5 model, four evaluation metrics were used: mean absolute error (MAE), maximum absolute error (HAE), root mean square error (RMSE), and correlation coefficient (R). The above evaluation metrics can be defined by Equations 6–15. tP is true positive (correctly detected), FN is false negative (not detected), FP is false positive (incorrectly detected), F1 is the trade-off between precision and recall, mAP is the average of all the AP values of the different categories, MAE is the average of all the absolute errors, and HAE is the maximum absolute error. RMSE is very sensitive to the magnitude error of a set of measurements and gives a good indication of the precision of the measurements. r is the degree of correlation between the manually measured flax plant phenotypic data and the model-predicted data, N is the number of experimental images, Ti is the manually measured ith plant phenotypic data, and mi is the model-predicted ith plant phenotypic data. These metrics were chosen to comprehensively evaluate the phenotypic data extraction ability of the directed search algorithm (Abyaneh et al., 2011).

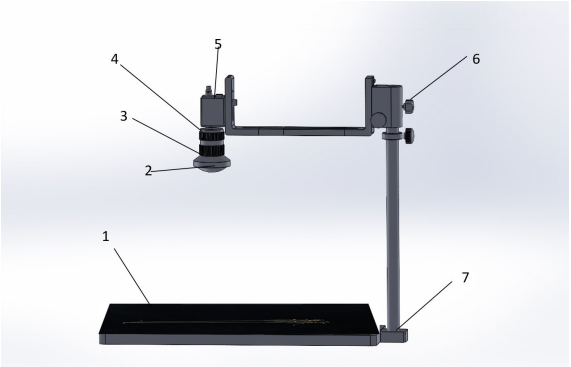


FIGURE 6
Shooting instrument. (1) Flax plant carrier table, (2) industrial camera wide-angle lens, (3) exposure time adjustment, (4) focal length adjustment, (5) computer data cable connection, (6) height adjustment, and (7) removable metal tube.

$$Precision = \frac{TP}{TP + FP} \times 100\,\% \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \times 100\,\% \tag{7}$$

$$F = \frac{(\alpha^2 + 1)^2 Recall \times Precision}{Recall + \Pr ecision} \tag{8}$$

$$F1 = \frac{2 \ *Recall \times Precision}{Recall + Precision} \tag{9}$$

$$AP = \int_0^1 Precision_{(Recall)} dR \tag{10}$$

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{11}$$

$$MAE = \frac{1}{N}\sum_{1}^{N}|m_i - T_i| \tag{12}$$

$$HAE = Max(|mi - Ti|) \tag{13}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{1}^{N}(m_i - T_i)^2} \tag{14}$$

$$R = \sqrt{1 - \frac{\sum_{i=1}^{N}(m_i - T_i)^2}{\sum_{i=1}^{N}(m_i - \overline{m})^2}} \tag{15}$$

3.4 Calculate the number of flax fruits, plant height, length of main stem, and number of main stem divisions

(1) Number of flax fruits

The number of flax fruits is determined by the number of "Flax" labels.

(2) Plant height and main stem length

In the same environment, Formulas 16 and 17 define the flax plant height and main stem length: H_{true} is the manually measured value of plant height and main stem length of the flax plant, H_{pi} is the plant height and main stem length of the pixel of the identification frame, H_{rate} is the ratio of the actual length of the one-dollar coin to the length of the pixel, H_{rate2} is the actual length of the one-dollar coin, and H_{pi2} is the pixel length of the one-dollar coin.

$$H_{true} = H_{pi} * H_{rate} \tag{16}$$

$$H_{rate} = \frac{H_{true2}}{H_{pi2}} \tag{17}$$

The actual diameter of the one-dollar coin was measured using 0.02-mm Vernier calipers, and the pixel diameter of the one-dollar coin was calculated using digital image technology.

(3) Number of main stem divisions

The label "n" (n= 1, 2, …) indicates that the main stem of the flax plant is n sub-stems, from which the number of sub-stems of the main stem is calculated.

3.5 Model identification results

The phenotypic organs of 100 flax plant images from the test set were recognized using the improved Flax-YOLOv5 model. The results of flax plant phenotypic organ recognition are shown in Figure 7. In addition, Figure 8A demonstrates the case of some flax fruits occluding each other, while Figure 8B demonstrates the case of branches occluding flax fruits, from which it can be seen that the model proposed in this paper has better recognition results.

The phenotypic data of 100 flax plants obtained from manual measurements were thoroughly compared with the phenotypic prediction data generated by the algorithm proposed in this study. To assess the reliability and stability of the algorithm in this paper, a correlation analysis was performed, and the results are shown in Figure 9.

From Figure 9A, it can be seen that most of the flax plants had between five and 40 fruits with a strong correlation and a mean absolute error of 1.37 fruits, although the maximum absolute error was seven fruits, but this was for very few plants with complex branching. As can be seen in Figure 9B, the height of most plants ranged from 50 cm to 75 cm, with a mean absolute error of 0.80 cm. As can be seen in Figure 9C, the craft length of the majority of plants was essentially in the range of 30 cm to 50 cm, with a mean absolute error of 2.24 cm. It is worth noting in Figure 9D that the intensity of the bubble color in the graphs reflects the number of main stem divisions of the repeat frequency, the vast majority of the main stem split number predicted accurately, with an average absolute error of 0.12. In summary, the number of fruits, plant height, main stem length, and the number of main stem split R of flax plants was 99.59%, 99.53%, 99.05%, and 92.82%, respectively, and the results were better and in line with the actual production needs.

3.6 Validation set test results and analysis

To evaluate the performance of the Flax-YOLOv5 model, we performed tests on a validation set. We chose the YOLOv3-tiny (Redmon and Farhadi, 2018), YOLOv5x (Jocher et al., 2022), YOLOv7-tiny (Wang et al., 2023), YOLOv7x, YOLOv8n (Lou et al., 2023), and YOLOv9c (Wang et al., 2024) models for comparison. Changes in training curves of different models mAP@0.5 are shown in Figure 10. It can be seen from the figure that mAP@0.5 of the YOLOv3, YOLOv5x, YOLOv7-tiny, YOLOv8n, and YOLOv9c models is significantly lower than that of the improved model Flax-YOLOv5. Although mAP@0.5 of the YOLOv7x model is close to that of the Flax-YOLOv5 model, it does not exceed it, and mAP@0.5 of the Flax-YOLOv5 model tends to 1 in a more stable trend with stronger convergence.

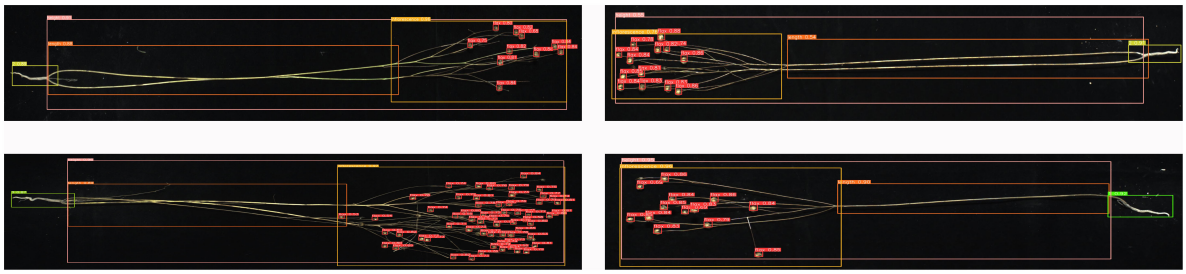


FIGURE 7
Results of phenotypic organ recognition in flax plants. The image has been cropped for ease of viewing.

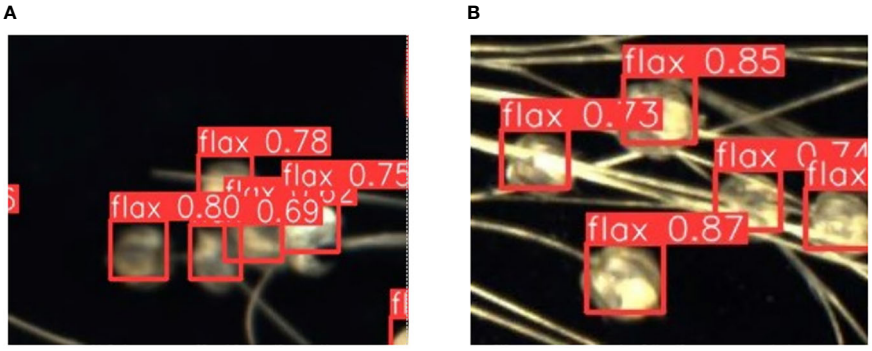


FIGURE 8
Recognition results of partially obscured fruits. The label "flax" in the picture stands for flax fruit; Numbers are confidence rates. **(A)** demonstrates the case of some flax fruits occluding each other, while **(B)** demonstrates the case of branches occluding flax fruits.

The experimental results comparing the recognition accuracy of the improved model Flax-YOLOv5 model with other models are shown in Table 1. As can be seen from Table 2, F1 and mAP@0.5 values of YOLOv3, YOLOv5x, and YOLOv7-tiny models are lower than 90%, which indicates that the performance is not ideal and does not meet the requirements of actual applications. Compared with the YOLOv7x model, the Flax-YOLOv5 model has an increase of 0.56 percentage points on F1 and 0.22 percentage points on mAP@0.5. However, the Flax-YOLOv5 model is 36.22 MB less than the YOLOv7x model. Although the YOLOv8n and YOLOv9c models are smaller than the improved model, the F1 evaluation shows that the improved model has more advantages. Overall, the improved Flax-YOLOv5 model exhibits superior performance compared to the YOLOv3, YOLOv5x, YOLOv7-tiny, YOLOv7x, YOLOv8n, and YOLOv9c models, providing a balance between accuracy and model size.

3.7 Test set test results and analysis

In this study, four phenotypic data points for each flax plant sample corresponding to the images in the dataset were successfully obtained

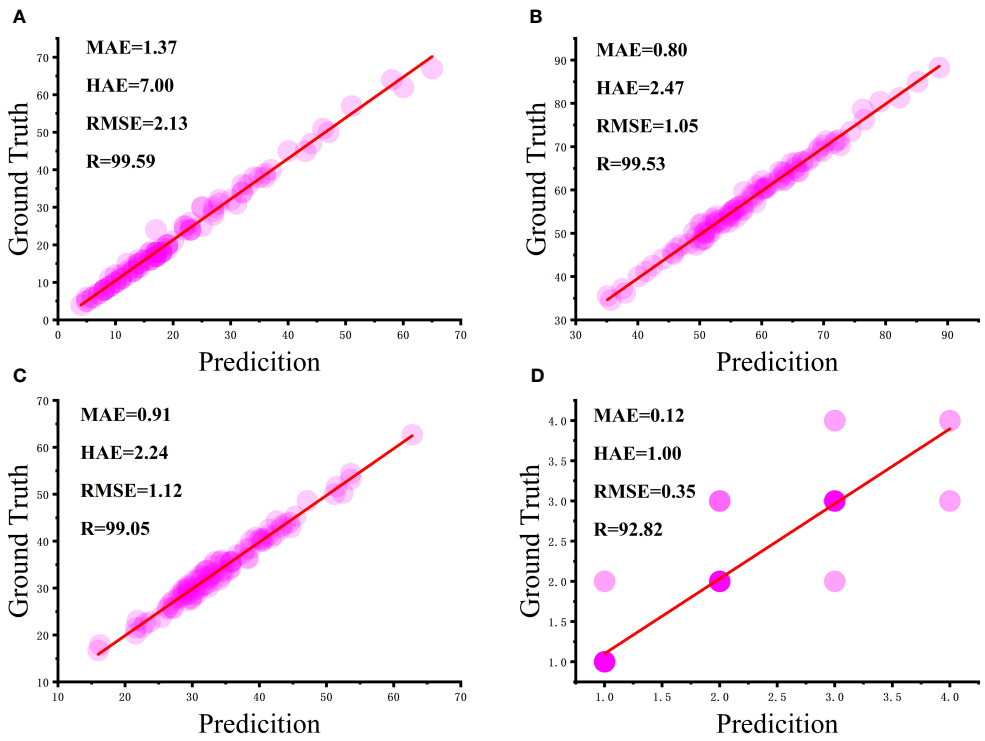


FIGURE 9
Correlation analysis between manual and algorithmic measurements: **(A)** number of flax fruits, **(B)** plant height, **(C)** length of main stem, and **(D)** number of main stem divisions.

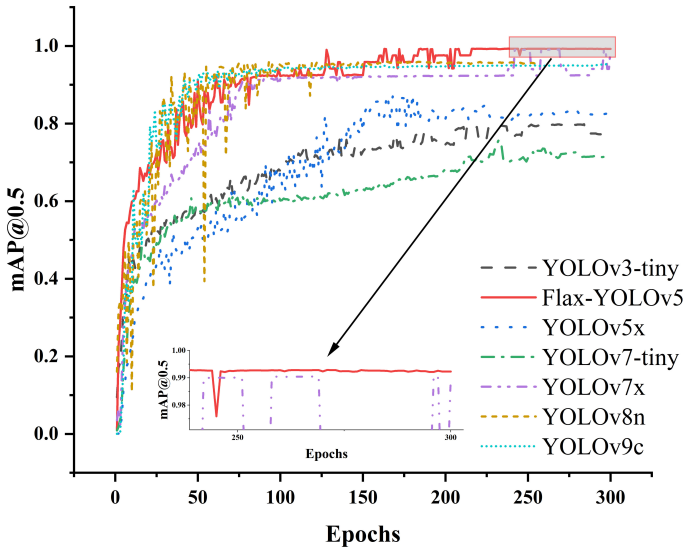


FIGURE 10
mAP@0.5 change curves of different models.

TABLE 1 Results predicted by different models in the test set.

Model	Number of flax fruits/pieces				Plant height/cm				Main stem length/cm				Number of main stem divisions/pieces			
	MAE	HAE	RMSE	R	MAE	HAE	RMSE	R	MAE	HAE	RMSE	R	MAE	HAE	RMSE	R
YOLOv3-tiny	21.16	67.00	25.18	7.69	/	/	/	/	/	/	/	/	/	/	/	/
YOLOv5x	18.76	61.00	23.00	26.76	2.01	5.84	2.51	97.91	8.27	54.40	14.65	37.41	1.58	4.00	1.92	12.85
YOLOv7-tiny	9.37	39.00	12.87	89.03	1.40	5.57	1.78	99.04	5.60	51.60	12.21	45.40	1.28	4.00	1.74	19.24
YOLOv7x	5.97	24.00	8.60	94.55	1.28	6.22	1.60	98.94	4.40	42.90	9.99	63.78	0.32	4.00	0.73	70.47
YOLOv8n	19.14	62.00	23.01	53.94	2.01	23.60	4.06	92.76	6.59	51.60	13.65	38.71	0.55	4.00	1.07	48.56
YOLOv9c	19.43	60.00	23.01	72.41	1.21	4.86	1.55	99.15	3.74	44.3	9.25	66.00	0.34	3.00	0.72	74.55
Flax-YOLOv5	1.37	7.00	2.13	99.59	0.80	2.47	1.05	99.53	0.91	2.24	1.12	99.05	0.12	1.00	0.35	92.82

MAE, mean absolute error; HAE, maximum absolute error; RMSE, root mean square error; R, correlation coefficient.

TABLE 2 Comparison of recognition results of different models.

Model	Precision (%)	Recall (%)	F1 (%)	mAP@0.5 (%)	Model size (MB)
YOLOv3-tiny	81.90	75.92	78.80	79.73	17.15
YOLOv5x	88.01	62.68	73.22	87.60	169.22
YOLOv7-tiny	92.61	66.31	77.28	71.26	12.03
YOLOv7x	92.82	98.15	95.41	99.07	138.88
YOLOv8n	94.58	91.31	92.92	95.75	6.14
YOLOv9c	95.51	90.77	93.08	95.35	50.44
Flax-YOLOv5	93.25	98.86	95.97	99.29	102.66

FIGURE 11
Original image.

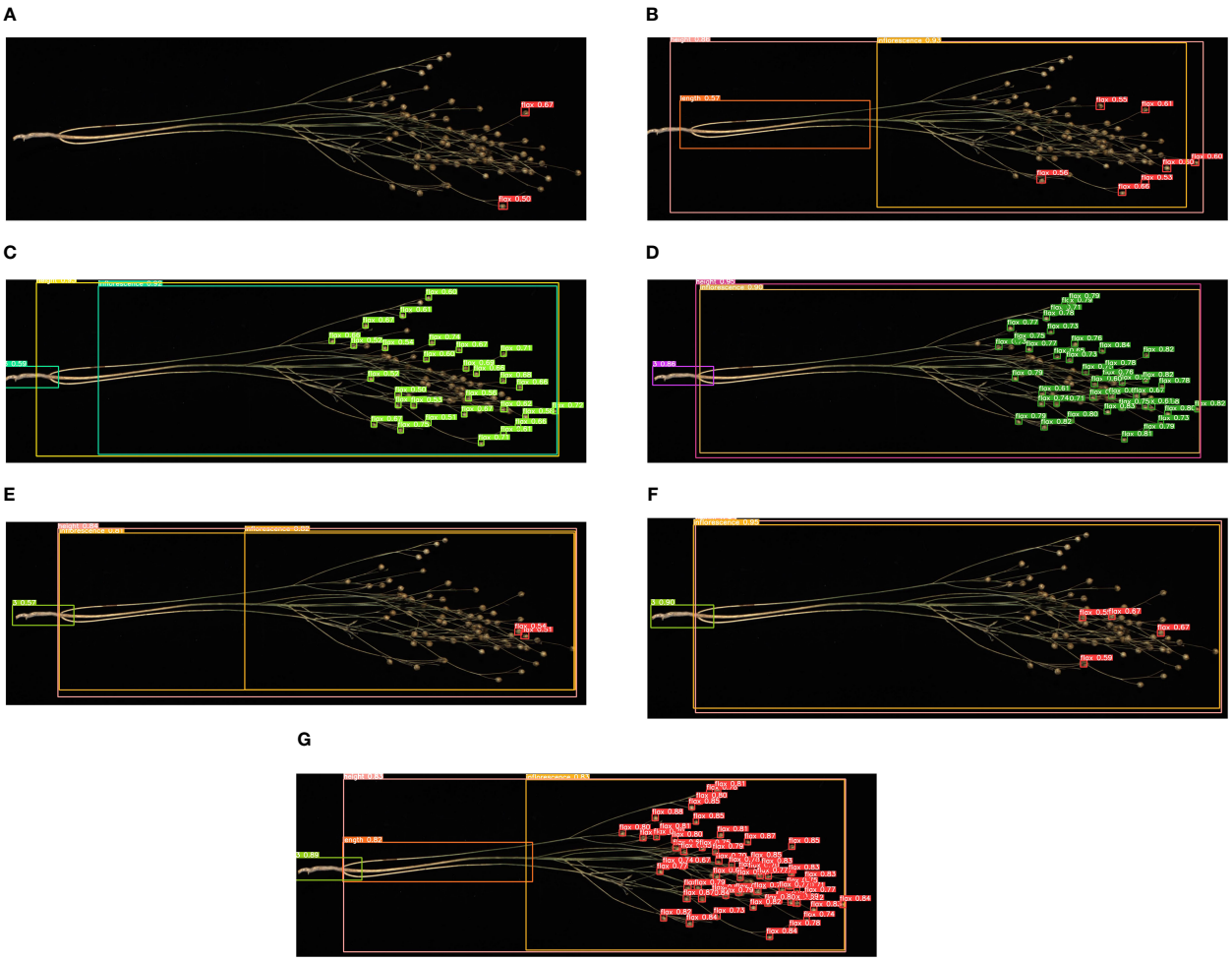


FIGURE 12
Comparison of recognition of different models: **(A)** YOLOv3-tiny, **(B)** YOLOv5x, **(C)** YOLOv7-tiny, **(D)** YOLOv7x, **(E)** YOLOv8n, **(F)** YOLOv9c, and **(G)** Flax-YOLOv5. The image has been cropped for ease of viewing.

10.3389/fpls.2024.1404772

through rigorous testing of the test set. These phenotypic measurements were then compared with manual measurements for validation. The results predicted by the different models in the test set are given in Table 1.

The YOLOv3-tiny model showed limited discrimination, recognizing only the fruits of the flax plant with a correlation coefficient of only 7.69%, indicating a large margin of error. Similarly, the identification results of the YOLOv5x model showed correlation coefficients of less than 50% for the number of flax fruits, main stem length, and number of main stem meristems, reflecting considerable inaccuracy.

The YOLOv7-tiny, YOLOv8n, and YOLOv9c models also performed poorly in the identification of flax fruit number, main stem length, and main stem branching number. The correlation coefficient of the YOLOv7x model in identifying the main stem length and the main stem branching number was less than 50%, and the identification accuracy was poor, with correlation coefficients of identifying the main stem length and the main stem branching number being 63.78% and 70.04%, which were unsatisfactory.

The improved Flax-YOLOv5 model, in contrast, showed better prediction results, with correlation coefficients of 99.59%, 99.53%, 99.05%, and 92.82% for flax fruit, plant height, main stem length, and number of main stem branches, respectively. These results were significantly better than those of the YOLOv3-tiny, YOLOv5x, YOLOv7-tiny, YOLOv7x, YOLOv8n, and YOLOv9c models.

To verify the effectiveness of the model improvement, we selected a flax plant with multiple flax fruits and branches from the test set and tested it using the above model and the Flax-YOLOv5 model; the original image is shown in Figure 11, and the comparative results of the recognition by different models are shown in Figure 12.

As can be seen in Figure 12, the YOLOv3-tiny model has limited recognition ability and can only accurately recognize two flax fruits. Similarly, the YOLOv7-tiny, YOLOv7x, YOLOv8n, and YOLOv9c models were defective in recognizing the main stem length of flax plants, accompanied by a considerable number of missing fruit detection. The improved Flax-YOLOv5 model, in contrast, has better recognition ability and can accurately recognize flax fruits, plant height, main stem length, and number of main stem divisions.

3.8 Ablation experiments and analysis

To verify the effectiveness of the improved model Flax-YOLOv5, it is necessary to compare and analyze the models through ablation experiments, and the results of the ablation experiments are shown in Table 3.

As can be seen in Table 3, the correlation coefficients of flax fruits with plant height, main stem length, and number of main stem divisions in Model 2 are higher than the values of Model 1. This observation emphasizes the advantages of the BiFormer network in extracting the target features, which improves the performance of the network in the plant detection task. Model 3 plant height correlation coefficients were significantly higher than those of Model 2 by 34.09 percentage points, which indicates that the integration of SIoU significantly enhanced the model fitting ability, which led to an overall improvement in the accuracy of the model recognition framework.

TABLE 3 Results of ablation experiments.

Model			Number of flax fruits/pieces				Plant height/cm				Main stem length/cm				Number of main stem divisions/pieces			
	BiFormer	SIOU	MAE	HAE	RMSE	R	MAE	HAE	RMSE	R	MAE	HAE	RMSE	R	MAE	HAE	RMSE	R
1	×	×	1.42	10.00	2.27	99.31	4.02	71.60	14.06	53.57	6.52	51.60	14.04	45.74	0.21	4.00	0.62	76.28
2	√	×	1.38	8.00	2.21	99.39	3.39	71.30	11.88	65.15	4.48	51.60	10.52	60.93	0.19	3.00	0.50	85.00
3	×	√	1.37	9.00	2.23	99.19	1.12	3.03	1.37	99.24	6.01	51.60	13.51	45.47	0.17	1.00	0.41	89.67
4	√	√	1.37	7.00	2.13	99.59	0.80	2.47	1.05	99.53	0.91	2.24	1.12	99.05	0.12	1.00	0.35	92.82

MAE, mean absolute error; HAE, maximum absolute error; RMSE, root mean square error; R, correlation coefficient.

4 Application

To facilitate researchers in selecting flax varieties, it is simple to obtain key phenotypic indicators such as the number of fruits, plant height, main stem length, and the number of main stem divisions of flax plants. Using the improved Flax-YOLOv5 model, the statistical software for flax plant phenotypic data was elaborately designed and developed. This software system is based on PyQt5 technology, which ensures its robustness and scalability. Deployment was effectively accomplished using the PyInstaller toolkit.

The software has a variety of features that greatly assist in phenotypic data analysis. Specifically, users can upload photos and videos and turn on the camera for real-time recognition. By entering data, the software automatically recognizes each organ of the flax plant and provides comprehensive statistics on its phenotypic data. This comprehensive approach ensures accurate and efficient data collection, which is essential for accurate flax variety selection and subsequent breeding programs.

5 Conclusion

The acquisition of flax plant phenotype data is the cornerstone of flax breeding. The traditional method is manual technical testing, which is not only time-consuming but also expensive. Therefore, we propose a Flax-YOLOV5 model specifically designed to obtain Flax phenotypic data. The experimental results show that in the verification set, mAP@0.5 is 99.29%. In the test set, the correlation analysis between the predicted value of the model and the key phenotypic traits (fruit number, plant height, main stem length, and main stem number) generated 99.59%, 99.53%, 99.05%, and 92.82%, respectively, and their MAEs were 1.37 pieces, 0.80 cm, 0.91 cm, and 0.12 pieces, respectively, all of which were within the acceptable range. These results show that our method can accurately capture the phenotypic data of flax plants, which provides convenience for the selection of flax varieties. On this basis, a PC-based flax phenotype data collection platform was designed and developed. The platform can efficiently collect key phenotypic traits such as fruit number, plant height, main stem length, and main stem number. This practical application highlights the practicability and effectiveness of our proposed method in supporting flax plant breeding, improves the efficiency of flax plant phenotype data acquisition, and greatly reduces the cost of data acquisition, which provides a solid foundation for flax breeding to become digital. In future research, for plants with complex branches and a large number of fruits, the recognition rate should be further improved, the recognition effect of the number of main stems should be more accurate, and the model parameters should be reduced. At present, the statistics of the secondary branches of the primary branches of flax plants are difficult, and we will further study and solve the problems.

Data availability statement

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

Author contributions

KS: Conceptualization, Data curation, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft. CL: Formal Analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Writing – review & editing. JH: Methodology, Resources, Supervision, Writing – review & editing. JZ: Data curation, Resources, Supervision, Validation, Writing – review & editing. YQ: Data curation, 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 funded by the National Natural Science Foundation of China (No. 32360437), the Innovation Fund for Higher Education of Gansu Province (No. 2021A-056), and the Industrial Support Program for Higher Education Institutions of Gansu Province (No. 2021CYZC-57).

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

References

Abyaneh, H. Z., Nia, A. M., Varkeshi, M. B., Marofi, S., and Kisi, O. (2011). Performance evaluation of ANN and ANFIS models for estimating garlic crop evapotranspiration. *J. Irrig. Drain Eng.* 137, 280–286. doi: 10.1061/(ASCE)IR.1943-4774.0000298

Ajayi, O. G., Ashi, J., and Guda, B. (2023). Performance evaluation of YOLO v5 model for automatic crop and weed classification on UAV images. *Smart Agric. Technol.* 5, 100231. doi: 10.1016/j.atech.2023.100231

Bai, Y., Yu, J., Yang, S., and Ning, J. (2024). An improved YOLO algorithm for detecting flowers and fruits on strawberry seedlings. *Biosyst. Engineering* 237, 1–12. doi: 10.1016/j.biosystemseng.2023.11.008

Chen, X., Liu, T., Han, K., Jin, X., Wang, J., Kong, X., et al. (2024). SP-yolo-based deep learning method for monitoring cabbage seedling emergence. *Eur. J. Agron.* 157, 127191. doi: 10.1016/j.eja.2024.127191

Du, X., Cheng, H., Ma, Z., Lu, W., Wang, M., Meng, Z., et al. (2023). DSW-YOLO: A detection method for ground-planted strawberry fruits under different occlusion levels. *Comput. Electron. Agric.* 214, 108304. doi: 10.1016/j.compag.2023.108304

Gao, F., Fang, W., Sun, X., Wu, Z., Zhao, G., Li, G., et al. (2022). A novel apple fruit detection and counting methodology based on deep learning and trunk tracking in the modern orchard. *Comput. Electron. Agric.* 197, 107000. doi: 10.1016/j.compag.2022.107000

Gong, W., Kang, X., Ma, M., Duan, H., and Jiang, G. (2020). Research progress on QTL mapping of flax. *Plant Fiber Sci. China* 42, 187–192. doi: 10.3969/j.issn.1671-3532.2020.04.008

Guo, X., Li, J., Zheng, L., Zhang, M., and Wang, M. (2022). Acquiring soybean phenotypic parameters using Re-YOLOv5 and area search algorithm. *Trans. Chin. Soc. Agric. Engineering* 38, 186–194. doi: 10.11975/j.issn.1002-6819.2022.15.020

Jocher, G., Stoken, A., and Borovec, J. (2022). *ultralytics/yolov5*. Available at: https://github.com/ultralytics/yolov5.

Kauser, S., Hussain, A., Ashraf, S., Fatima, G., Ambreen, Javaria, S., et al. (2024). Flaxseed (Linum usitatissimum); phytochemistry, pharmacological characteristics, and functional food applications. *Food Chem. Advances* 4, 100573. doi: 10.1016/j.focha.2023.100573

Kong, D., Wang, J., Zhang, Q., Li, J., and Rong, J. (2023). Research on fruit spatial coordinate positioning by combining improved YOLOv8s and adaptive multi-resolution model. *Agronomy* 13 (8), 2122. doi: 10.3390/agronomy13082122

Li, H., Shi, L., Fang, S., and Yin, F. (2023). Real-time detection of apple leaf diseases in natural scenes based on YOLOv5. *Agriculture* 13, 878. doi: 10.3390/agriculture13040878

Lou, H., Duan, X., Guo, J., Liu, H., Gu, J., Bi, L., et al. (2023). DC-YOLOv8: small-size object detection algorithm based on camera sensor. *Electronics* 12, 2323. doi: 10.3390/electronics12102323

Pei, H., Sun, Y., Huang, H., Zhang, W., Sheng, J., and Zhang, Z. (2022). Weed detection in maize fields by UAV images based on crop row preprocessing and improved YOLOv4. *Agriculture* 12, 975. doi: 10.3390/agriculture12070975

Praczyk, M., and Wielgusz, K. (2021). Agronomic assessment of fibrous flax and linseed advanced breeding lines as potential new varieties. *Agronomy* 11, 1917. doi: 10.3390/agronomy11101917

Qian, L., Zheng, Y., Cao, J., Ma, Y., Zhang, Y., and Liu, X. (2024). Lightweight ship target detection algorithm based on improved YOLOv5s. *Real-Time Image Proc.* 21, 3. doi: 10.1007/s11554-023-01381-w

Rahman, R., Bin Azad, Z., and Bakhtiar Hasan, M. (2022). "Densely-populated trafficdetection using YOLOv5 and non-maximum suppression ensembling," in *Proceedingsof the International Conference on Big Data, IoT, and Machine Learning. Lecture Noteson Data Engineering and Communications Technologies* (Singapore: Springer) 95. doi: 10.1007/978-981-16-6636-0_43

Redmon, J., and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv* 1804, 2767. doi: 10.48550/arXiv.1804.02767

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

Su, P., Li, H., Wang, X., Wang, Q., Hao, B., Feng, M., et al. (2023). Improvement of the YOLOv5 model in the optimization of the brown spot disease recognition algorithm of kidney bean. *Plants* 12, 3765. doi: 10.3390/plants12213765

Wang, C., Bochkovskiy, A., and Liao, H. 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: 10.48550/arXiv.2207.02696

Wang, C., Yeh, I., and Liao, H. M. (2024). YOLOv9: learning what you want to learn using programmable gradient information. *arXiv 2402.13616 [cs.CV]*. doi: 10.48550/arXiv.2402.13616

Wang, J., Zhang, H., Liu, Y., Zhang, H., and Zheng, D. (2024). Tree-level Chinese fir detection using UAV RGB imagery and YOLO-DCAM. *Remote Sensing* 16, 335. doi: 10.3390/rs16020335

Yang, Z., Feng, H., Ruan, Y., and Weng, X. (2023). Tea tree pest detection algorithm based on improved yolov7-tiny. *Agriculture* 13, 1031. doi: 10.3390/agriculture13051031

Zhang, B., Xia, Y., Wang, R., Wang, Y., Yin, C., Fu, M., et al. (2024). Recognition of mango and location of picking a point on stem based on a multi-task CNN model named YOLOMS. *Precis. Agric* 25, 1454–1476. doi: 10.1007/s11119-024-10119-y

Zhang, M., Zhao, D., Sheng, C., Liu, Z., and Cai, W. (2023). Long-strip target detection and tracking with autonomous surface vehicle. *JMSE* 11, 106. doi: 10.3390/jmse11010106

Zhang, Z.-S., Wang, L.-J., Li, D., Li, S.-J., and Özkan, N. (2011). Characteristics of flaxseed oil from two different flax plants. *Int. J. Food Properties* 14, 1286–1296. doi: 10.1080/10942911003650296

Zhu, L., Li, X., Sun, H., and Han, Y. (2024). Research on CBF-YOLO detection model for common soybean pests in complex environments. *Comput. Electron. Agriculture* 216, 108515. doi: 10.1016/j.compag.2023.108515

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 22 August 2024
DOI 10.3389/fpls.2024.1451018

OPEN ACCESS

EDITED BY
Mohsen Yoosefzadeh Najafabadi,
University of Guelph, Canada

REVIEWED BY
Ana María Mendez-Espinoza,
Agricultural Research Institute (Chile), Chile
Yunchao Tang,
Dongguan University of Technology, China

*CORRESPONDENCE
Xiangyang Sun
sxy586977@163.com

RECEIVED 18 June 2024
ACCEPTED 05 August 2024
PUBLISHED 22 August 2024

CITATION
Sun X (2024) Enhanced tomato detection in greenhouse environments: a lightweight model based on S-YOLO with high accuracy. *Front. Plant Sci.* 15:1451018. doi: 10.3389/fpls.2024.1451018

Enhanced tomato detection in greenhouse environments: a lightweight model based on S-YOLO with high accuracy

Xiangyang Sun*

College of Information Science and Engineering, Shandong Agricultural University, Tai'an, China

Introduction: Efficiently and precisely identifying tomatoes amidst intricate surroundings is essential for advancing the automation of tomato harvesting. Current object detection algorithms are slow and have low recognition accuracy for occluded and small tomatoes.

Methods: To enhance the detection of tomatoes in complex environments, a lightweight greenhouse tomato object detection model named S-YOLO is proposed, based on YOLOv8s with several key improvements: (1) A lightweight GSConv_SlimNeck structure tailored for YOLOv8s was innovatively constructed, significantly reducing model parameters to optimize the model neck for lightweight model acquisition. (2) An improved version of the α-SimSPPF structure was designed, effectively enhancing the detection accuracy of tomatoes. (3) An enhanced version of the β-SIoU algorithm was proposed to optimize the training process and improve the accuracy of overlapping tomato recognition. (4) The SE attention module is integrated to enable the model to capture more representative greenhouse tomato features, thereby enhancing detection accuracy.

Results: Experimental results demonstrate that the enhanced S-YOLO model significantly improves detection accuracy, achieves lightweight model design, and exhibits fast detection speeds. Experimental results demonstrate that the S-YOLO model significantly enhances detection accuracy, achieving 96.60% accuracy, 92.46% average precision (mAP), and a detection speed of 74.05 FPS, which are improvements of 5.25%, 2.1%, and 3.49 FPS respectively over the original model. With model parameters at only 9.11M, the S-YOLO outperforms models such as CenterNet, YOLOv3, YOLOv4, YOLOv5m, YOLOv7, and YOLOv8s, effectively addressing the low recognition accuracy of occluded and small tomatoes.

Discussion: The lightweight characteristics of the S-YOLO model make it suitable for the visual system of tomato-picking robots, providing technical support for robot target recognition and harvesting operations in facility environments based on mobile edge computing.

KEYWORDS

greenhouse tomatoes, YOLOv8, object detection, deep learning, high accuracy, fast detection, lightweight, computer vision

1 Introduction

Tomatoes are one of the most extensively cultivated vegetables in Chinese agriculture. China not only leads globally in tomato production but also serves as a major exporter (Huo, 2016). Manual tomato harvesting requires a significant amount of labor and time. Mechanized harvesting not only cuts down on labor expenses but also boosts efficiency in the harvesting process (Li et al., 2021). Harvesting robots initially utilize computer vision systems for fruit detection, followed by guiding mechanical arms based on the detection results for harvesting operations. Therefore, fruit detection stands as a pivotal aspect throughout the entire harvesting process, with its accuracy and speed directly influencing the efficiency of harvesting robots. However, tomato fruits exhibit diverse growth postures, overlap with each other, and are heavily obscured by leaves, branches, and stems, presenting certain challenges for robot recognition. Rapid and precise identification of tomato fruits in complex greenhouse environments is a pressing issue in the development of tomato harvesting robots (Liu, 2017). Moreover, deploying models with excessively high complexity proves challenging in practical scenarios. Thus, enhancing fruit detection accuracy, speed, and lightweight improvements are crucial for bolstering the performance of harvesting robots.

Traditional methods for tomato fruit recognition in greenhouse environments rely on extracting and analyzing information based on color and shape features. Feng et al. (2015) extracted the color features of red ripe tomato fruits using the 2R-G-B color difference model and identified red ripe tomato fruits using dynamic threshold segmentation. However, this method is time-consuming and does not consider factors such as leaf occlusion in complex environments during tomato fruit recognition. Ma et al. (2016)) introduced a technique for recognizing objects by combining saliency detection with the circular random Hough transform, achieving a correct recognition rate of 77.6% for immature tomato fruits. Despite the achievements in feature design in the above studies, they suffer from slow recognition speed, low detection accuracy, and poor robustness of traditional machine vision algorithms in complex scenes, making them difficult to meet practical requirements. Although these studies have achieved certain success in feature design and tomato recognition to some extent, their slow recognition speed, low detection accuracy, and poor robustness in complex scenes cannot meet practical requirements. Additionally, they often depend on static color characteristics to recognize desired fruits. This reliance can make them less adaptable to variations in lighting and color discrepancies, resulting in reduced effectiveness when dealing with unstable color conditions. In summary, traditional methods for tomato fruit recognition fail to meet the requirements of high accuracy and real-time performance. Additionally, most of the above studies have not considered the influencing factors in complex greenhouse environments, lack robustness to diverse feature changes, and therefore, are unable to meet practical requirements.

In recent times, deep convolutional neural networks have emerged as a pivotal domain within deep learning research, attracting considerable interest. Their increasing utilization in greenhouse settings for tomato recognition has offered novel perspectives on tomato fruit identification. The detection methods of deep convolutional neural networks can be divided into two types: single-stage and two-stage detection. Region-based methods, the first type, create a set of candidate boxes and subsequently classify the targets contained within these boxes. Representative models include RCNN (Girshick et al., 2014), Fast-RCNN (Girshick, 2015), and Faster-RCNN (Ren et al., 2016). Although these methods exhibit excellent recognition accuracy with relatively low error rates and miss rates, their complex processing leads to slow detection speeds, making it difficult to meet real-time detection requirements. The second type is regression-based methods, where targets are directly classified while being located. The YOLO series networks (Redmon et al., 2016; Redmon and Farhadi, 2018; Ge et al., 2021) are typical representatives of this category. These methods have the advantage of fast recognition speed, meeting real-time requirements, and achieving accuracy levels close to the first type of methods. Given their strong real-time performance, the second type of object detection methods is beneficial for improving the efficiency of harvesting robots and monitoring devices, suitable for real-time target detection in complex environments. (Su et al. (2022)) used a lightweight YOLOv3 model in greenhouse environments, combined with lightweight networks, successfully applied it to classify tomato ripeness, achieving a 97.5% mAP. However, the model still had a large volume, making deployment challenging. Liu et al. (2020) proposed an improved tomato detection model, YOLO-Tomato, based on YOLOv3, achieving good performance. Nevertheless, the YOLOv3 model they used was large. Appe et al. (2023)) introduced a tomato detection model based on YOLOv5, which incorporates the CBAM attention mechanism into the network architecture, effectively detecting overlapping small tomatoes with an average precision of 88.1%. However, this study also faced issues with low detection accuracy. Tian et al. (2024)) proposed the TF-YOLOv5s model for detecting tomato flowers and fruits in natural environments, replacing the complete intersection over union (CIoU) loss with the efficient intersection over union (EIoU) loss and incorporating the SE attention module. Bai et al. (2024)) improved the YOLOv7 model to accurately identify strawberry seedling flowers and fruits by addressing issues such as small size, similar colors, and overlapping occlusion. They also applied the GSConv structure to optimize the model neck, achieving a 92.1% mAP with a frame rate of 45 frames per second, meeting real-time detection requirements. Li (Li et al., 2024) et al. proposed a lightweight improved YOLOv5s model for detecting dragon fruit in illuminated environments during both day and night. Meng (Meng et al., 2023) et al. proposed a spatiotemporal convolutional neural network model that utilizes a shifted window Transformer to integrate a regional convolutional neural network model for detecting pineapple fruits. Chen (Chen et al., 2024) et al. proposed a set of visual algorithms for

motion target estimation, real-time self-localization, and dynamic harvesting. They also established a reliable coordination mechanism for continuous movement and picking actions. This study, inspired by previous research, addresses issues such as large model volumes, low accuracy, and difficulty in deploying actual robot vision systems. It proposes a lightweight and accurate S-YOLO model, considering tomato recognition in complex environments. Establishing a high-performance, lightweight target detection model suitable for tomato harvesting robot vision systems remains a significant challenge.

In actual greenhouse environments, tomato fruits often overlap and are heavily occluded, varying in sparsity and size, posing challenges for rapid and accurate tomato fruit recognition. Therefore, this paper introduces a novel S-YOLO model to address the aforementioned issues. This model can rapidly and accurately identify greenhouse tomato fruits while maintaining lightweight characteristics, addressing some of the limitations faced by current research and providing new technical support for the visual systems of tomato harvesting robots. This study focuses on the target detection problem for automated tomato harvesting in greenhouse environments. The core of the research is to develop and optimize a lightweight tomato target detection model, S-YOLO, aimed at enhancing the accuracy of tomato detection in complex environments. The model features high precision, a lightweight design, and rapid detection capabilities. However, the cost-effectiveness of model deployment and its practical impact on agricultural production require further discussion and analysis in future research to provide more robust support for agricultural production. This paper makes the following key contributions:

1. Introducing a S-YOLO model suitable for complex environment tomato detection, characterized by high accuracy, lightweight design, and fast speed, suitable for the visual systems of tomato harvesting robots.
2. Constructing a lightweight GSConv_SlimNeck structure suitable for YOLOv8s to optimize the model's neck section, thereby improving model performance.
3. Creating an enhanced version of the α-SimSPPPF structure to optimize the network architecture, effectively improving detection accuracy with better performance.
4. Proposing a new enhanced version of the β-SIoU loss function, optimizing the training process, and improving tomato recognition accuracy.
5. Integrating the SE attention module into the network structure for more effective tomato feature extraction.

The paper's structure is as follows: Section 2 covers dataset acquisition and processing. Section 3 outlines the principles of the proposed S-YOLO network structure and details improvement methods for each module. In Section 4, experimental setups are explained, and the performance of each enhanced module is thoroughly analyzed, evaluating and comparing results with other mainstream models. Finally, Sections 5 and 6 discuss and summarize the paper's findings.

2 Experimental data and processing methods

2.1 Datasets

The dataset utilized in this research was originally obtained from the Kaggle platform, which provides resources for developers and data scientists to participate in machine learning competitions, host databases, and write and share code. The tomato dataset used in this study consists of images collected by the authors from the glass greenhouse at the National Engineering Research Center for Facility Agriculture in Chongming Base (Li et al., 2019). All images were captured in real agricultural environments, not under laboratory conditions, thus exhibiting complex backgrounds and varying brightness. The dataset comprises a total of 895 image samples. Example images from the tomato dataset in complex environments are shown in Figure 1, which mainly include large tomato targets, small tomato targets, occluded tomatoes, and clustered tomatoes.

2.2 Data preprocessing

For deep learning tasks, dataset annotation is crucial. In the case of complex greenhouse tomato images, variations in lighting conditions due to different weather and angles result in significant color differences in the collected tomato fruit images. Additionally, the diverse growth postures and severe overlapping and occlusion of greenhouse tomato fruits make it challenging to extract shape features. In this study, the LabelImg tool was used for manual annotation of tomato images, and the annotation data for each image was stored in the form of Extensible Markup Language files, following the VOC format (Everingham et al., 2010). To meet the training requirements of the detection model, the images were resized to a uniform size of 640×640 pixels and converted to RGB three-channel images. Since the YOLOv8 network incorporates online data augmentation during the training process, including techniques such as Mosaic and Mixup augmentation, and given that the dataset is not particularly small, additional offline data augmentation is generally unnecessary to save training time. Therefore, this study did not perform additional offline data augmentation.

To facilitate subsequent model training, 80% of the original 895 tomato images were allocated to the training set, 10% to the validation set, and 10% to the test set. The specific distribution is shown in Table 1. Finally, these datasets were utilized for training the network models, followed by additional Mixup and Mosaic data augmentation.

3 Methods

3.1 Proposed S-YOLO object detection model

Figure 2 illustrates the architecture of YOLOv8 (Reis et al., 2023). The neck and backbone parts of YOLOv8 may have drawn

FIGURE 1
Tomato datasets. **(A)** Big tomatoes, **(B)** Small tomatoes, **(C)** Occlusion, **(D)** Clusters of tomatoes.

inspiration from the ELAN module in YOLOv7 (Wang et al., 2023). It utilizes the C2f structure to replace the C3 structure in YOLOv5 while adjusting the number of channels for various scale models. This meticulous adjustment of the model structure significantly enhances its performance. The head part adopts the current mainstream decoupled head structure, separating the classification and detection heads. It also transitions from Anchor-Based to Anchor-Free. Although the YOLOv8s model shows significant improvements, it still involves substantial computational complexity. Moreover, accurately detecting tomato fruits in complex environments remains a huge challenge.

This study introduces a novel lightweight network, termed S-YOLO, which is built upon the enhancements made to the YOLOv8s architecture. This entails a meticulous optimization of the model architecture to strike a delicate balance between model complexity and performance metrics. This also involves optimizing the architecture while maximizing the model's capability to accurately identify objects in real-time scenarios. To achieve this, four key strategies are employed. Firstly, we utilize the GSConv_SlimNeck structure to optimize the model's neck section, effectively reducing the parameter count while ensuring performance remains intact. Secondly, we replace the original SPPF module with the newly proposed α-SimSPPF module, enhancing the model's capabilities. Thirdly, a novel loss function, β-SIoU, is introduced to refine the training process and enhance overall model performance. Lastly, the integration of the SE attention module into the YOLOv8s' neck network facilitates better focus on crucial features, thereby further improving the accuracy of tomato fruit target identification. Figure 3 illustrates the architecture of the S-YOLO model proposed in this study.

TABLE 1 Tomato images.

Dataset	Number
training	724
validation	81
test	90
total	895

3.2 The GSConv_SlimNeck design for YOLOv8s

GSConv (Li et al., 2022) is a novel lightweight convolutional operation designed to reduce model complexity while maintaining accuracy. The structure of GSConv is shown in Figure 4. The computational cost of GSConv is approximately 60% to 70% of that of standard convolution (SC), while its contribution to model learning ability is comparable to SC. By leveraging GSConv, we can effectively utilize the advantages of Depthwise Separable Convolution (DSC) while mitigating its drawbacks on the model.

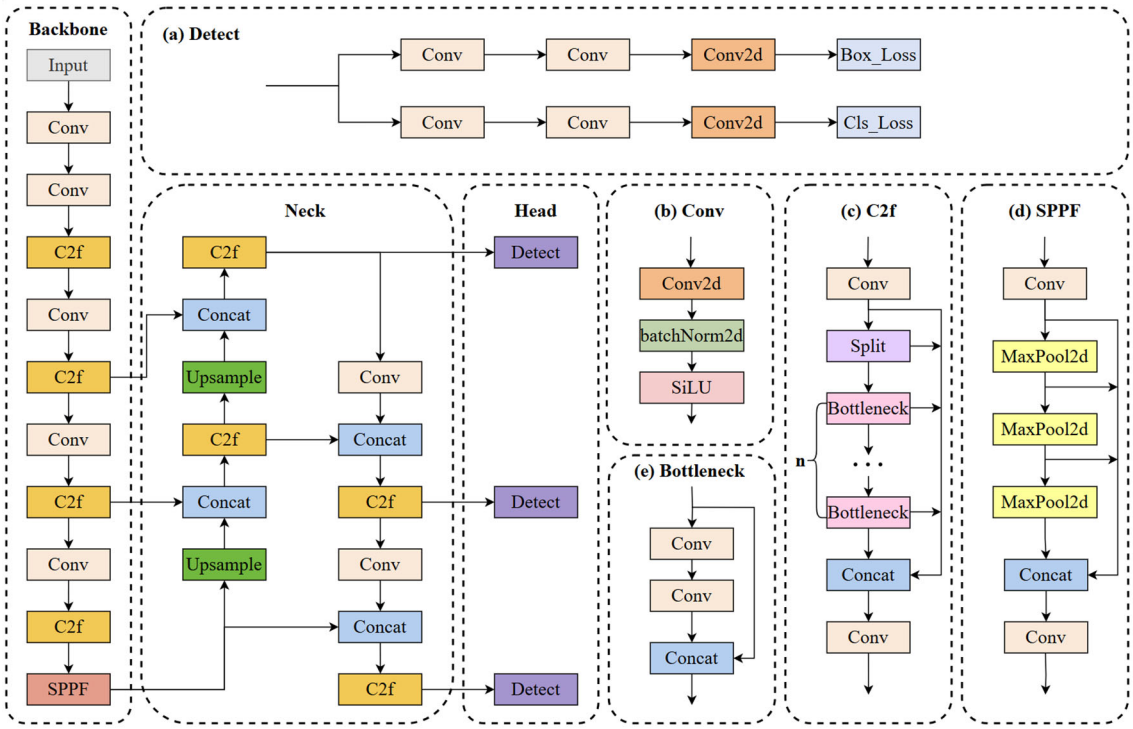


FIGURE 2
YOLOv8 algorithm model.

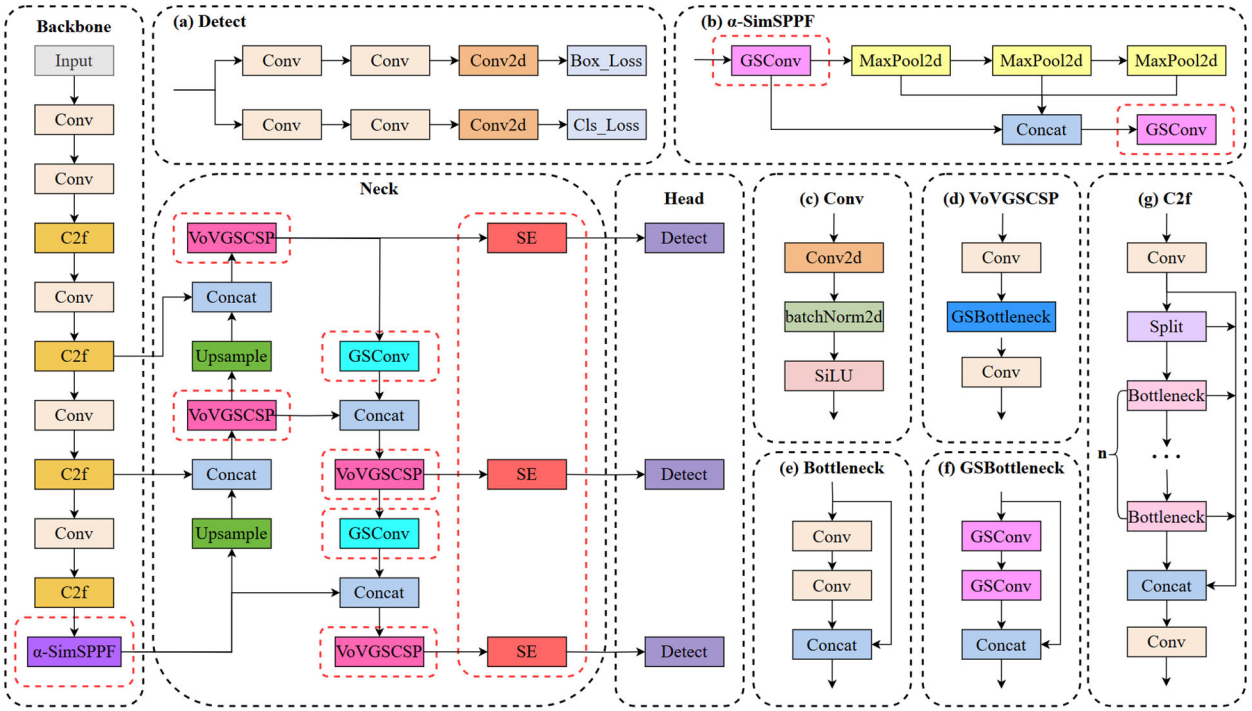


FIGURE 3
The proposed S-YOLO algorithm model. The red dashed line represents the added improvement module.

SlimNeck is a design paradigm aimed at achieving higher cost-effectiveness for detectors. The core idea of SlimNeck is to use GSConv in the Neck part of the detector while maintaining a standard Backbone, which maximally reduces the impact of DSC drawbacks on the model while maintaining high accuracy. SlimNeck also introduces other modules, such as GSbottleneck and VoVGSCSP, to further improve model performance.

The original Neck structure of YOLOv8s is not sufficiently lightweight, so this paper proposes a lightweight structure, GSConv_SlimNeck, suitable for the YOLOv8s model. The construction process is as follows: Firstly, the conventional Conv structure in the Neck component is substituted with the GSConv structure. Subsequently, the terminal C2f structure within the Neck is substituted with the VoVGSCSP structure. With these two improvements, we successfully construct a lightweight GSConv_SlimNeck structure suitable for YOLOv8s, making the model more lightweight while maintaining higher detection performance.

3.3 The improved α-SimSPPF structure

SimSPPF is an improved spatial pyramid pooling method proposed in YOLOv6 (Li et al., 2022), which is an upgraded version of SPPF. SPPF (Spatial Pyramid Pooling Function) is a technique used for feature map pooling, commonly employed in Convolutional Neural Networks (CNNs), to pool features at different scales, thereby better capturing spatial information in images. It solves the multi-scale problem by extracting features using pooling kernels of different sizes at different scales. The fundamental concept behind SPPF involves parallel processing of the input through multiple MaxPool layers of varying sizes, followed by fusion to enhance the detector's performance. In YOLOv5, SPPF is used to achieve feature-level fusion of local and global features. SimSPPF is an improved version of SPPF. Compared to SPPF, SimSPPF can improve the performance of the detector without increasing computational cost. SimSPPF uses ReLU activation function, while SPPF uses SiLU activation function. Structurally, SimSPPF maintains the original parallel structure of SPPF but with higher computational efficiency.

The SimSPPF structure was enhanced in this study by substituting the Conv structure with the more lightweight GSConv structure, resulting in an improved version termed α-SimSPPF. Compared to both the SPPF structure and SimSPPF, α-SimSPPF boasts higher detection accuracy with fewer parameters.

3.4 The enhanced β-SIoU algorithm

YOLOv8 by default utilizes the CIoU (Qiu et al., 2022) loss function, which introduces additional calculations for the distance between center points and diagonal distances. Therefore, compared to traditional IoU, the computational complexity increases, potentially adding some computational cost. CIoU's computation method is relatively complex, requiring more processing and calculation of bounding box coordinates. Traditional methods like CIoU, DIoU (Zheng et al., 2020), etc., match IoU, center point distance, aspect ratio, etc., between real and predicted boxes but do not consider the mismatched orientation between them. This inadequacy results in slow convergence and lower efficiency, ultimately leading to poorer models.

Gevorgyan (2022) proposed the SIoU loss function, which incorporates angle considerations and scale sensitivity, introducing a more complex bounding box regression method to address the limitations of previous loss functions. By integrating these aspects, better training speed and prediction accuracy can be achieved. The aim of the SIoU is to reduce the gap between predicted and actual bounding boxes, accounting for variations in shape and angle. The SIoU schematic is shown in Figure 5.

The process of angle loss calculation is as follows:

$$Angle_{Loss} = 1 - 2 * \sin^2(\arcsin(\frac{ch}{d}) - \frac{\pi}{4}) \tag{1}$$

$$Distance_{Loss} = 2 - e^{-\gamma p_x} - e^{-\gamma p_y} \tag{2}$$

$$p_x = (\frac{cw}{Cw})^2 \tag{3}$$

$$p_y = (\frac{ch}{Ch})^2 \tag{4}$$

$$\gamma = 2 - Angle_{Loss} \tag{5}$$

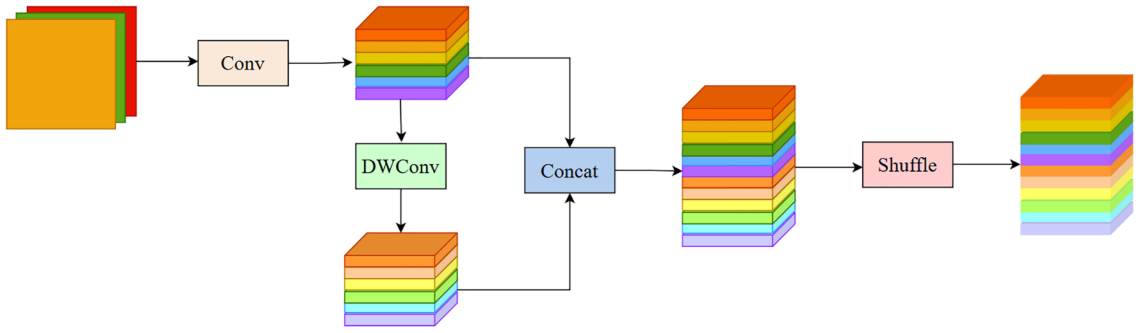


FIGURE 4
The structure of the GSConv module.

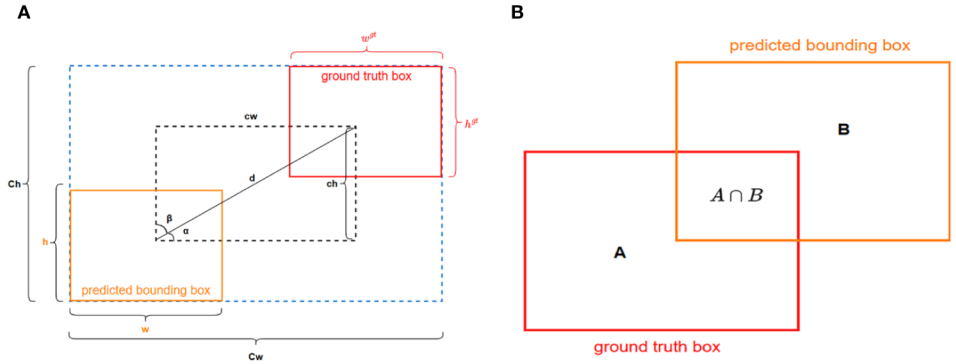


FIGURE 5
The SIoU loss function **(A)** and the IoU loss function **(B)**.

In this equation, "cw" represents the disparity in width between the centers of the two bounding boxes, and "Ch" represents the height of the minimum bounding rectangle of the ground truth bounding box, while "Cw" represents the width of the minimum bounding rectangle of the predicted bounding box. The calculation process for shape loss is as follows:

$$Shape_{Loss} = (1 - e^{-W_w})^{\theta} + (1 - e^{-W_h})^{\theta} \tag{6}$$

$$W_w = \frac{|w - w^{gt}|}{max(w, w^{gt})} \tag{7}$$

$$W_h = \frac{|h - h^{gt}|}{max\,(h, h^{gt})} \tag{8}$$

In this equation, "w", "h", "w^{gt}", and "h^{gt}" respectively represent the width and height of the predicted bounding box and the true bounding box. θ controls the emphasis on shape loss. To avoid overly focusing on shape loss and thus reducing the movement of the predicted bounding box, the authors used a genetic algorithm to compute a value close to 4. The calculation process for IoU loss is as follows:

$$IoU = \frac{A \cap B}{A \cup B} \tag{9}$$

Where A∩B represents the intersection of the predicted bounding box and the ground truth bounding box, and A∪B represents the union of the predicted bounding box and the ground truth bounding box. The SIoU can be expressed using the following formula:

$$SIoU_{Loss} = 1 - IoU + \frac{DistanceLoss + ShapeLoss}{2} \tag{10}$$

He et al. (2022) proposed the α-IoU method, which enhances bounding box regression by incorporating a power transformation into the conventional IoU loss function. Inspired by this, to bolster the robustness of SIoU towards bounding boxes and attain higher accuracy in the regression of overlapping bounding boxes, this study enhances SIoU by introducing a power of 1.5 to each of its terms. We refer to this enhanced version as β-SIoU, and its effectiveness will be demonstrated through experiments in Section 4.3.5. The computation formula is shown as follows:

$$\beta - SIoU_{Loss} = 1 - IoU^{1.5} + (\frac{DistanceLoss + ShapeLoss}{2})^{1.5} \tag{11}$$

3.5 SE attention module

Attention mechanisms facilitate models in comprehensively grasping the structure and attributes of input data, thus advancing the precision and efficiency of object detection. Attention mechanisms empower the model to discern the significance of diverse local details in the image, allowing it to concentrate more effectively on crucial features and thereby enhance the accuracy of tomato fruit detection.

The SE (Squeeze-and-Excitation) attention mechanism (Hu et al., 2018) enhances model performance by modeling the correlation between different channels. Channel-wise attention assigns different weights to different channels, focusing on channels that are crucial for recognizing specific objects. The SE module captures channel relationships through Squeeze and Excitation operations. In the Squeeze phase, it condenses the output feature map from the convolutional layer into a feature vector via global average pooling. This vector captures comprehensive statistical data from the entire feature map. During the Excitation phase, the SE module employs a fully connected layer and a nonlinear activation function to determine the significance of each channel by learning their respective weights. By incorporating Squeeze and Excitation operations, the model autonomously learns the weight and significance of individual channels, enhancing the network's expressive power and performance. By automatically learning the weight and significance of individual channels, the network can prioritize crucial feature channels, enhancing overall model performance.

After comparing different attention mechanisms, this study selected the SE attention module with the highest accuracy and incorporated it into the model's neck. The SE attention structure is shown in Figure 6.

4 Experimental design and results analysis

4.1 Experimental environment and parameter setting

The experiments were conducted using PyTorch as the deep learning framework. Table 2 provides a detailed description of the experimental setup. To optimize model training, cosine annealing was employed to update the learning rate and network weight parameters. The entire process comprised 300 iterations. The momentum factor was set at 0.937 to effectively smooth gradient updates, facilitating faster convergence and stabilizing the training process. The weight decay was set at 0.0005 to help limit the model's complexity, prevent overfitting on the training data, and enhance the model's generalization ability to new data. The initial learning rate was set at 0.01 to quickly reduce the loss function during the initial training phase while avoiding excessively large steps that could lead to an unstable training process. The SGD optimizer was employed, which is suitable for large deep learning models. Using the SGD optimizer simplifies the computation process, and combined with the momentum factor, effectively speeds up convergence. During the first 50 iterations, the training of the backbone network was frozen, with a batch size of 8. Freezing the backbone network's training leverages the general features extracted by the pretrained model. This approach helps to quickly train the model with fewer computational resources and prevents disruption of the existing feature extraction capabilities. Setting the batch size to 8 improves training parallelism and efficiency within the limits of GPU memory. In the subsequent 250 iterations, the backbone network was unfrozen for training, and the batch size was adjusted to 4. Unfreezing the backbone network in the later training stage allows fine-tuning of the entire model to better adapt to the specific task's data distribution. Adjusting the batch size to 4 helps maintain training stability and efficiency as the model complexity increases. Freezing the training is also a concept in transfer learning, as the features extracted by the neural network backbone are general. Freezing the backbone during training can accelerate the training process and prevent the weights from being disrupted.

TABLE 2 Hardware and software environment.

Configuration Item	Value
CPU	Intel i9-12900H
GPU	NVIDIA GeForce RTX 3060
CUDA	12.0
Memory	32GB
Operating system	Windows11×64
Deep learning frame	PyTorch

4.2 Evaluation indicators

This research selected mean Average Precision (mAP), Average Precision (AP), precision, recall, F1 score, GFLOPs, model parameters, and frames per second (FPS) as performance metrics for evaluating the deep-learning model. The evaluation metrics were calculated using the formulas below.

$$Precision = \frac{TP}{TP + FN} \times 100\,\% \tag{12}$$

$$Recall = \frac{TP}{TP + FP} \times 100\,\% \tag{13}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\,\% \tag{14}$$

$$\mathrm{AP} = \int_0^1 \mathrm{P}(R)\mathrm{dR} \tag{15}$$

$$\mathrm{mAP} = \frac{\sum_{i=1}^{n} AP_i}{n} \tag{16}$$

$$\mathrm{FPS} = 1/T \tag{17}$$

Where TP represents the number of images where tomato fruit targets were correctly detected by the model, FP represents the number of images where non-tomato fruit targets were incorrectly detected by the model, and FN represents the number of images where tomato fruit targets were missed by the model. Precision

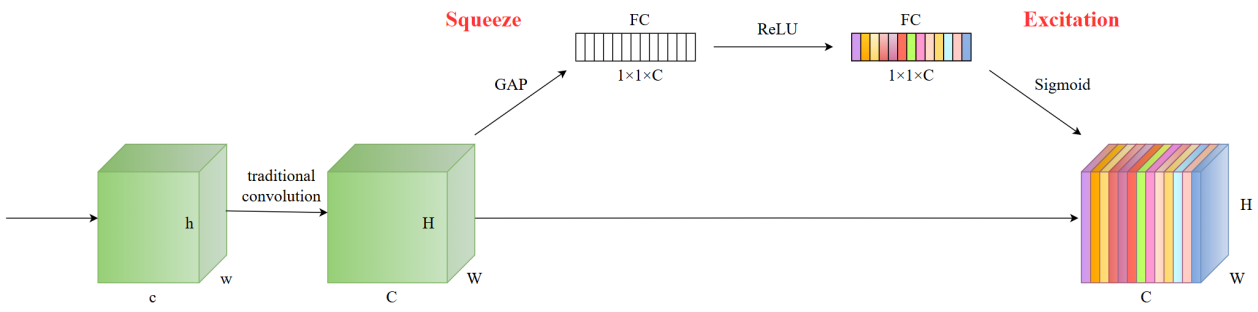


FIGURE 6
The structure of the SE attention.

indicates the precision rate, while Recall represents the recall rate. F1-score serves as a means to strike a balance between precision and recall. Precision and recall values are utilized to construct the precision-recall curve (PR curve), with the area under this curve denoted as AP (Average Precision). The mAP refers to the average AP. T denotes the detection time for a single image. FPS represents the number of images detected per second. The model parameters were calculated considering the input and output channel counts along with the convolutional kernel sizes, aiding in estimating the model's size. GFLOPs are used to measure model complexity.

4.3 Results and analysis

4.3.1 Training and validation of the S-YOLO algorithm

Figure 7A displays the training loss progression of the S-YOLO algorithm. During the initial training phase, the model exhibits relatively high learning efficiency, as indicated by the rapid decline in the training loss curve, suggesting that the model is quickly learning new features. As training progresses, the rate of decrease in the loss curve gradually slows down, implying that the model is gradually stabilizing and approaching convergence. Throughout this process, both the training and validation set losses fluctuate but eventually stabilize, indicating that the model has reached the expected stable state.

In Figure 7B, the fluctuation of the mean Average Precision (mAP) throughout each training epoch is depicted. It can be observed that mAP rapidly increases at the beginning of training, corresponding to the rapid decline in the training loss curve. As training continues, the change in mAP stabilizes, indicating a continuous improvement in the model's accuracy. At the 150th training epoch, mAP reaches its peak, indicating that the model is very close to its optimal performance at this point. These two figures together depict the training process of the model, from rapid learning to eventual convergence, demonstrating the effectiveness and stability of the S-YOLO model.

4.3.2 Ablation experiments

We conducted ablation experiments on the tomato dataset to evaluate the performance of GSConv_SlimNeck, α-SimSPPF, β-SIoU, and SE components integrated into the model. Based on YOLOv8s, the subsequent models progressively integrated the improved modules. Model1 optimized the model's neck structure using the GSConv_SlimNeck architecture. Model2 replaced the original SPPF structure with the enhanced version of α-SimSPPF based on Model1. Model3 introduced the proposed β-SIoU loss function on top of Model2. Ultimately, the SE attention module was embedded within the network's neck in Model3, leading to the formulation of the S-YOLO model.

As shown in Table 3, based on YOLOv8s, Model1 achieved improvements in several metrics by introducing the GSConv_SlimNeck structure. Precision, mAP@0.5, and FPS increased by 0.86%, 0.86%, and 2.81FPS, respectively, while model complexity and parameters decreased by 3.35G and 1.78M. The addition of the α-SimSPPF module further improved model accuracy and mAP@0.5, while reducing computational overhead. However, this improvement also slightly decreased detection speed by 0.33FPS. After adding β-SIoU to Model2, the detection rate increased by 0.45FPS compared to Model2 and exceeded YOLOv8s and Model1, compensating for the shortcomings of α-SimSPPF. This indicates a noticeable improvement in model performance with the enhanced β-SIoU loss function. The introduction of the SE attention module further improved precision, mAP@0.5, and FPS by 1.92%, 0.48%, and 0.56FPS, respectively, compared to Model3, despite a slight increase of 0.04M in model parameters. This suggests the effectiveness of the attention mechanism in extracting features relevant to tomato detection. Figure 8 shows the experimental curves and bar charts for different models.

In summary, the lightweight S-YOLO model surpasses the original YOLOv8s model significantly. Not only does it achieve model lightweighting, but it also maximizes the enhancement in detection accuracy. The model exhibits improvements across various metrics: precision, mAP@0.5, and FPS see increases of 5.25%, 2.1%, and 3.49FPS, respectively. Furthermore, the model complexity (measured in GFLOPs) and parameters are reduced by

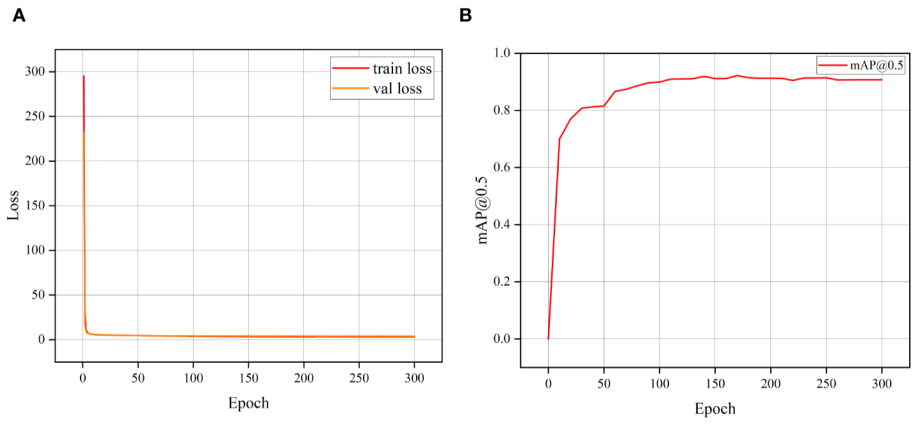


FIGURE 7
The training loss curve variation **(A)** and mAP training variation **(B)**.

3.6G and 2.06M, respectively, showcasing its efficiency and effectiveness in practical applications.

4.3.3 Comparison of different lightweight modules

In constructing the lightweight structure GSConv_SlimNeck, both GSConv and GhostConv (Han et al., 2020) modules were compared and analyzed to validate their effectiveness. The experimental results in Table 4 show that both GSConv and GhostConv modules contribute equally to model lightweighting, resulting in a reduction of model complexity and parameters by 3.35G and 1.78M, respectively.

However, utilizing the GSConv module to build the GSConv_SlimNeck structure exhibits superior model performance compared to using the GhostConv module. Although there is a slight decrease in recall, precision, and F1 score experience significant improvements. Specifically, compared to using GhostConv, using GSConv increases precision by 0.84%, mAP by 0.12%. Overall, the GSConv_SlimNeck structure built using GSConv demonstrates superior performance.

4.3.4 Comparison of SPPF, SimSPPF, and α-SimSPPF

To verify the efficacy of the proposed α-SimSPPF structure, this study conducted a comparative analysis involving SPPF, SimSPPF, and α-SimSPPF. These three modules were placed at the same position in the model and trained accordingly. Table 5 presents the experimental results. From various metrics, it is evident that the performance of SimSPPF is significantly lower than that of the SPPF module. However, following the enhancement from SimSPPF to α-SimSPPF, the model's performance saw significant improvement. In comparison to the SPPF module, precision increased by 0.65% and mAP@0.5 increased by 0.48%. Additionally, the model complexity and parameters were reduced by 0.25G and 0.32M, respectively.

Although using the α-SimSPPF structure resulted in a slight decrease of 0.33FPS in detection speed compared to using the SPPF structure, the accuracy and mAP@0.5 were significantly improved. Moreover, the model complexity was lower, and the model parameters were reduced, aligning with the research goal of this study. α-SimSPPF demonstrated superior performance on the dataset used in this study, with higher accuracy and lighter model, making it more suitable for tomato fruit detection and deployment in tomato harvesting robot visual systems.

4.3.5 Comparison of different IoU loss functions

This study delved deeper into the influence of integrating the β-SIoU algorithm on the model's performance, with a primary focus on comparing CIoU, DIoU, SIoU, and the β-SIoU algorithm. As shown in Table 6, compared to CIoU, DIoU achieved higher precision but slightly decreased mAP@0.5, while increasing the inference speed by 0.25FPS. SIoU resulted in varying degrees of decrease in precision, mAP@0.5, and FPS. However, the proposed β-SIoU algorithm demonstrated improvements across all metrics.

Among all these algorithms, Model3 stood out in multiple key metrics, particularly in precision, mAP, and processing speed. Compared to CIoU, DIoU, and SIoU, precision increased by 1.82%, 1.25%, and 2.16%, respectively, while mAP@0.5 increased by 0.28%, 0.69%, and 1.15%, respectively. Detection speed also increased by 0.45FPS, 0.2FPS, and 2.22FPS, respectively. These improvements significantly enhance model performance, making it suitable for handling overlapping and densely packed tomato objects, as well as deployment in tomato harvesting robot visual systems. Figure 9 illustrates the experimental curves for different loss functions.

In addition, this study explored the optimal loss function for the dataset by examining different exponent values for individual terms in SIoU. As shown in Table 7, varying the exponent values for individual terms in SIoU had no impact on the model's complexity.

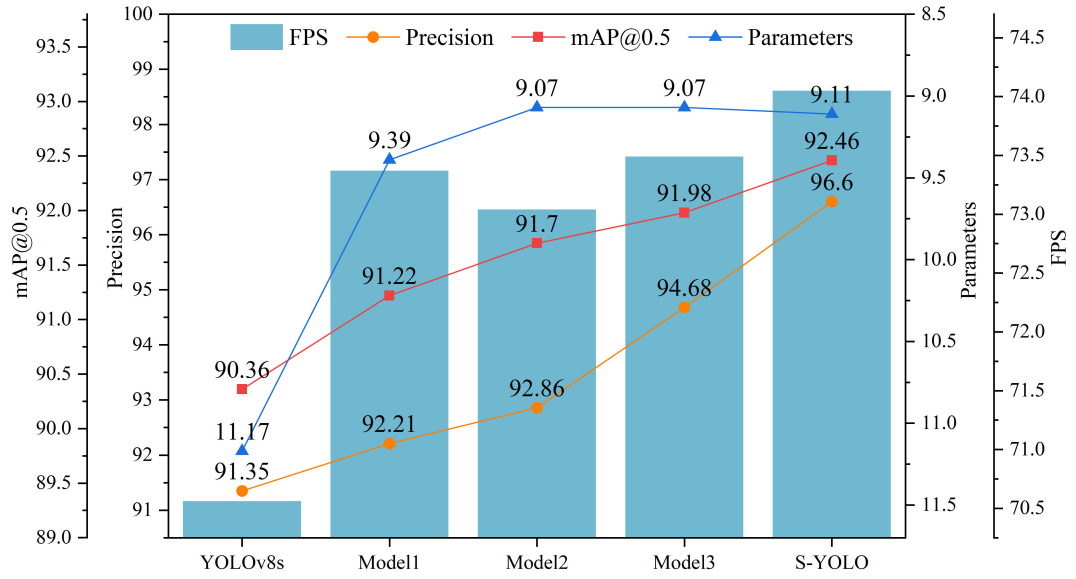


FIGURE 8
Experimental curves for different models.

TABLE 3 Ablation experiments on the proposed S-YOLO algorithm.

Model	GSConv_ SlimNeck	α-SimSPPF	β-SIoU	SE	Precision	mAP@0.5	GFLOPs (G)	Parameters (M)	FPS
YOLOv8s Model1	√				91.35 92.21	90.36 91.22	28.82 25.47	11.17 9.39	70.56 73.37
Model2	√	√			92.86	91.70	25.22	9.07	73.04
Model3	√	√	√		94.68	91.98	25.22	9.07	73.49
S-YOLO	√	√	√	√	**96.60**	**92.46**	**25.22**	9.11	**74.05**

Bold values represent the best experimental results compared to other models.

TABLE 4 Experimental results for the lightweight modules.

Model	Precision	Recall	F1-Score	mAP@0.5	GFLOPs (G)	Parameters (M)	FPS
YOLOv8s	91.35	81.72	0.86	90.36	28.82	11.17	70.56
YOLOv8s + GhostConv_SlimNeck YOLOv8s + GSConv_SlimNeck	91.37 **92.21**	81.94 81.51	0.86 **0.87**	91.10 **91.22**	25.47 **25.47**	9.39 **9.39**	72.86 **73.37**

Bold values represent the best experimental results compared to other models.

When the exponent value for individual terms in SIoU was set to 1.5, precision reached 94.68%, mAP@0.5 reached 91.98%, and the detection rate reached 73.49FPS. When each exponent in the SIoU function is set to 1.5, the model demonstrates its optimal performance.

4.3.6 Comparison of different attention modules

To delve deeper into the influence of the SE attention module and its placement within the model architecture, this study explored inserting various attention mechanisms, including ECA (Wang et al., 2020), CBAM (Woo et al., 2018), CA (Hou et al., 2021), SimAM (Yang et al., 2021), GAM (Liu et al., 2021), Shuffle (Zhang and Yang, 2021), and EMA (Ouyang et al., 2023), at the same position. Additionally, three SE attention modules were inserted into the backbone network after the third, fourth, and fifth Conv structures.

As shown in Table 8, adding any attention mechanism led to an improvement in accuracy. However, except for the SE attention module, which increased mAP@0.5, the other attention modules resulted in varying degrees of decrease in mAP@0.5. This suggests that the SE attention module is most suitable for incorporation into this model structure. The decrease in mAP@0.5 when adding other attention mechanisms may be due to model overfitting or neglect of certain features of tomato fruits. The SE attention module significantly improved model performance, with accuracy and mAP@0.5 increasing by 1.92% and 0.48%, respectively, compared to Model3. Moreover, the detection rate increased by 0.56FPS. Compared to Model3, adding the GAM attention module not only increased the model complexity by

TABLE 5 Experimental results of SPPF, SimSPPF, and α-SimSPPF.

Model	Precision	Recall	F1-Score	mAP@0.5	GFLOPs (G)	Parameters (M)	FPS
Model1+SPPF Model1+ SimSPPF Model1+α-SimSPPF (Model2)	92.21	81.51	0.87	91.22	25.47	9.39	73.37
	91.35	81.72	0.86	91.14	25.47	9.39	72.77
	92.86	81.08	**0.87**	**91.70**	**25.22**	**9.07**	73.04

Bold values represent the best experimental results compared to other models.

TABLE 6 Comparison of different loss functions.

Model	Precision	mAP@0.5	GFLOPs (G)	Parameters (M)	FPS
Model2 + CIoU	92.86	91.70	25.22	9.07	73.04
Model2 + DIoU	93.43	91.29	25.22	9.07	73.29
Model2 + SIoU	92.52	90.83	25.22	9.07	71.27
Model2 + β-SIoU(Model3)	**94.68**	**91.98**	**25.22**	**9.07**	**73.49**

Bold values represent the best experimental results compared to other models.

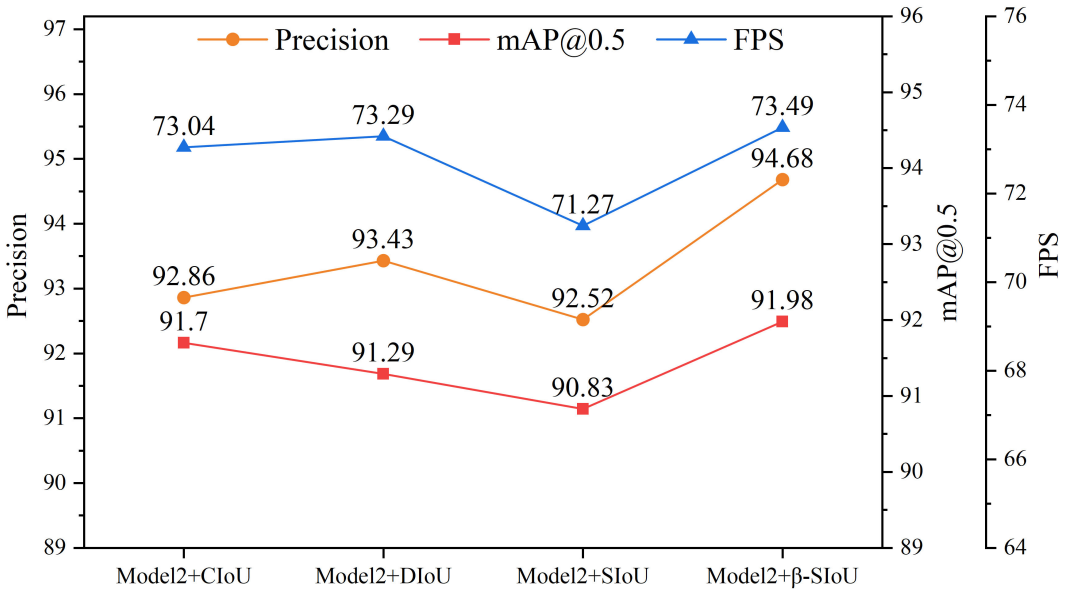


FIGURE 9
Experimental curves for different loss functions.

15.74G and the model parameter quantity by 8.6M but also decreased mAP@0.5 and the detection rate by 1.36% and 21.51FPS, respectively, severely reducing model performance. Although the EMA attention module achieved 97.31% accuracy, both mAP@0.5 and the detection rate were significantly lower than those with the SE attention mechanism. In general, the SE attention module exhibited the most impressive performance, leading to the most substantial enhancement in the S-YOLO model's performance.

As demonstrated in Table 9, incorporating the SE attention module into the backbone network resulted in a decline in model evaluation metrics. In comparison to models lacking attention mechanisms, integrating the SE attention module into the backbone network led to reductions of 0.14% and 2.15% in accuracy and mAP@0.5, respectively. However, when employing the SE attention module at the model's neck, the accuracy and mAP@0.5 increased by 2.06% and 2.63%, respectively, compared to inserting it into the backbone network. The performance decrease resulting from inserting the module into the backbone network may be attributed to the compression of spatial and channel dimensions of the feature maps caused by introducing attention mechanisms in the backbone network. Attention mechanisms typically selectively emphasize certain features, which may lead to the neglect of other features, resulting in the loss of semantic information. This loss of information could weaken the model's feature extraction ability. After inserting SE into the backbone network, the model's detection speed decreased by 0.06 FPS compared to Model3, and the decrease was more significant when compared to inserting it into the neck network, reaching 0.62 FPS. Figure 10 displays the experimental curves and bar charts for different attention modules.

4.3.7 Comparative analysis of various object-detection models' performance

To further substantiate the model's effectiveness, this study conducted an extensive comparison between the S-YOLO model and other prominent convolutional neural network object detection models, including the two-stage object detection model Faster RCNN, as well as the single-stage object detection algorithms CenterNet (Duan et al., 2019), YOLOv3 (Tian et al., 2019), YOLOv4 (Bochkovskiy et al., 2020), YOLOv5m (Yang et al., 2023), YOLOv7, YOLOv7x, YOLOv8m, and YOLOv8s. The experimental results are presented in Table 10.

TABLE 7 Experimental results for different exponential powers of SIoU.

Model	Exponent	mAP@0.5	GFLOPs (G)	Parameters (M)	FPS
Model2+SIoU	0.5	91.89	25.22	9.07	71.37
	1.0	90.83	25.22	9.07	71.27
	1.5	**91.98**	**25.22**	**9.07**	**73.49**
	2.0	91.23	25.22	9.07	72.03
	2.5	91.29	25.22	9.07	71.66
	3.0	90.97	25.22	9.07	72.62

Bold values represent the best experimental results compared to other models.

TABLE 8 Comparison of different attention models' performance.

Model	Precision	mAP@0.5	GFLOPs (G)	Parameters (M)	FPS
Model3	94.68	91.98	25.22	9.07	73.49
Model3 + ECA	96.10	90.94	25.22	9.07	69.30
Model3 + CBAM	95.56	91.91	25.22	9.16	69.06
Model3 + CA	95.05	90.71	25.22	9.13	73.33
Model3 + SimAM	96.01	90.83	25.22	9.07	73.62
Model3 + GAM	95.03	90.62	40.96	17.67	51.98
Model3 + Shuffle	96.67	87.17	25.22	9.07	71.47
Model3 + EMA	97.31	88.17	25.22	9.07	73.19
Model3 + SE(S-YOLO)	96.60	**92.46**	**25.22**	9.11	**74.05**

Bold values represent the best experimental results compared to other models.

TABLE 9 Experimental results on the effects of inserting attention modules at different positions.

Model	Embedding position	Precision	GFLOPs (G)	Parameters (M)	mAP@0.5	FPS
Model3	\	94.68	25.22	9.07	91.98	73.49
Model3 + SE	Backbone	94.54	25.22	9.07	89.83	73.43
Model3 + SE (S-YOLO)	Neck	**96.60**	**25.22**	**9.07**	**92.46**	**74.05**

Bold values represent the best experimental results compared to other models.

Faster RCNN is a typical two-stage object detection algorithm, but its model size is large, with model complexity and parameters much higher than other single-stage object detection algorithms. Its detection speed is only 10.57 FPS, which is only 14.27% of S-YOLO's. The model complexity is as high as 370.21G, about 15 times that of S-YOLO, and the model parameters are as high as 137.10M, about 14 times that of S-YOLO. S-YOLO's accuracy, mAP@0.5, and FPS are 45.18%, 14.04%, and 63.48FPS higher than Faster RCNN, respectively. Overall, the performance of the S-YOLO model far exceeds that of Faster RCNN.

In comparison to other models, S-YOLO outperforms other models across all metrics. The model accuracy, mAP@0.5, and detection speed are 96.60%, 92.46%, and 74.05FPS, respectively, with model complexity and parameters of only 25.22G and 9.11M. Compared to CenterNet, the S-YOLO model shows advantages in mAP@0.5, model complexity, model parameters, and FPS, with mAP@

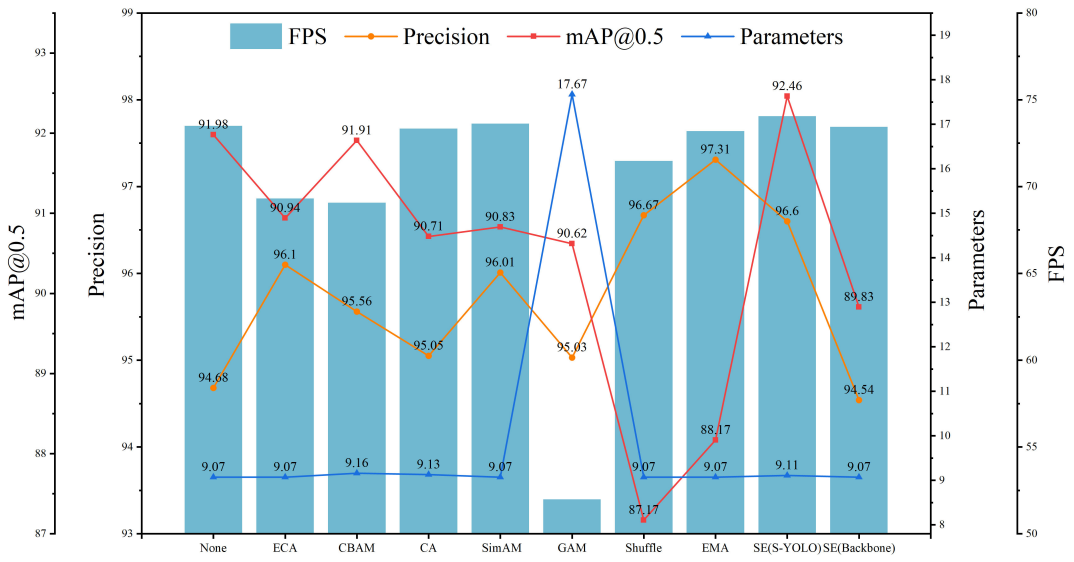


FIGURE 10
Experimental curves for different attention mechanisms.

TABLE 10 Comparison of different mainstream object detection models.

Model	Precision	mAP@0.5	GFLOPs (G)	Parameters (M)	FPS
Faster-RCNN	51.42	78.42	370.21	137.10	10.57
CenterNet	95.88	85.14	70.22	32.67	69.77
YOLOv3	87.62	86.55	66.17	61.95	46.42
YOLOv4	66.96	72.63	60.53	64.36	36.93
YOLOv5m	88.30	86.69	51.62	21.38	44.84
YOLOv7	87.20	89.61	106.47	37.62	28.39
YOLOv7x	91.56	88.84	190.58	71.34	17.94
YOLOv8m	93.19	91.69	79.32	25.90	37.57
YOLOv8s	91.35	90.36	28.82	11.17	70.56
S-YOLO	**96.60**	**92.46**	**25.22**	**9.11**	**74.05**

Bold values represent the best experimental results compared to other models.

0.5 7.32% higher, FPS 4.28FPS higher, and model complexity and parameters only 35.91% and 27.88% of CenterNet, respectively. YOLOv3 and YOLOv5m have similar model complexities and detection speeds, but their overall performance is much lower than S-YOLO. YOLOv4 has the lowest accuracy and mAP@0.5 among all models. Due to the higher model complexity of YOLOv7, YOLOv7x, and YOLOv8m, they also have a certain impact on detection speed, which is 45.66FPS, 56.11FPS, and 36.48FPS lower than S-YOLO, respectively, indicating that the lightweight improvements of S-YOLO have a certain effect on improving detection speed. Compared to the YOLOv8s model, the S-YOLO model has higher accuracy by 5.25%, mAP@0.5 by 2.1%, and FPS by 3.49FPS, with model complexity and parameters reduced by 3.6G and 2.06M, respectively, indicating that the improved S-YOLO model has improved in all indicators, and the model performance has been significantly improved. The following Figure 8 provides a more

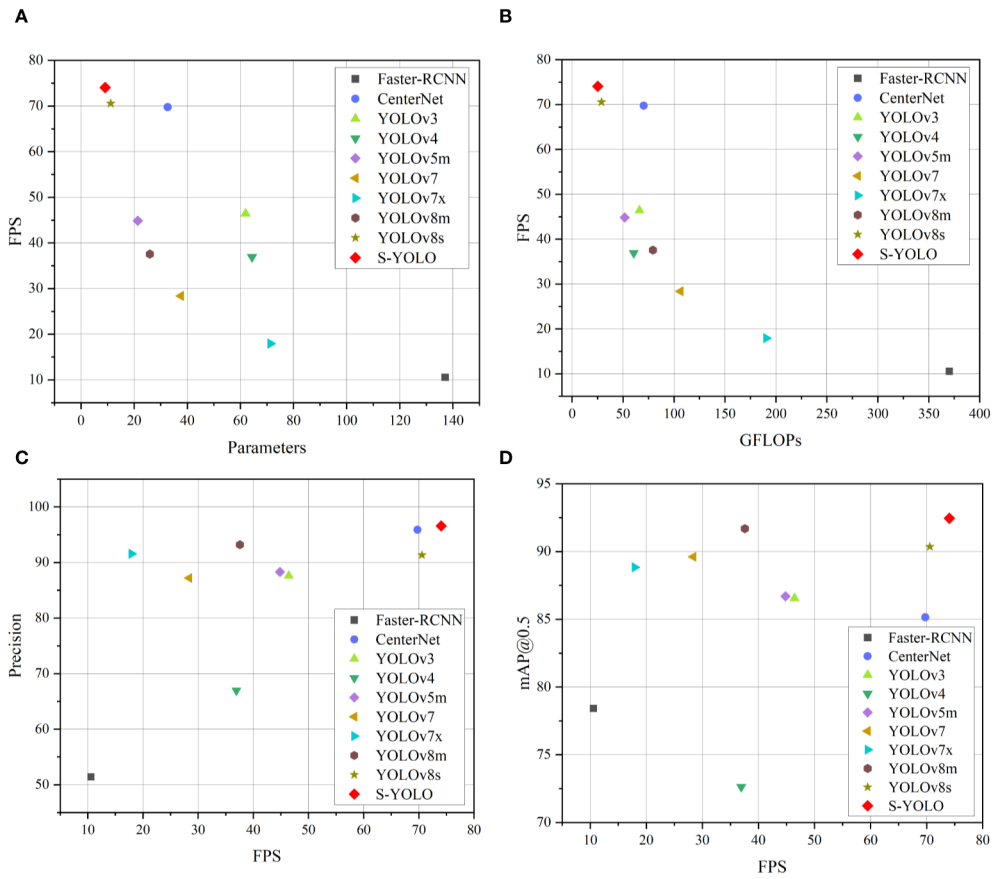


FIGURE 11
Scatter plots of the experiments for different models. **(A)** FPS-Parameters, **(B)** FPS-GFLOPs, **(C)** Precision-FPS, **(D)** mAP@0.5-FPS.

intuitive illustration of the unique advantages of S-YOLO compared to other models, achieving the optimal balance between model detection speed, lightweight, and accuracy. Figure 11 illustrates that the S-YOLO model excels over other models in various aspects.

In summary, the S-YOLO model performs significantly better than current mainstream object detection models, with high accuracy while being lightweight, providing technical references for the deployment of tomato harvesting robot vision systems.

4.3.8 Model visualization results

The detection performance of CenterNet, YOLOv4, YOLOv5m, YOLOv7, YOLOv7x, YOLOv8s, and S-YOLO models is illustrated in Figure 12. For the YOLOv4 model, there are numerous detection errors, incorrectly identifying tomato leaves and other objects as tomato fruits. The YOLOv5m model exhibits poor detection performance for occluded tomatoes, resulting in missed detections and overall poor recognition. YOLOv7x also struggles

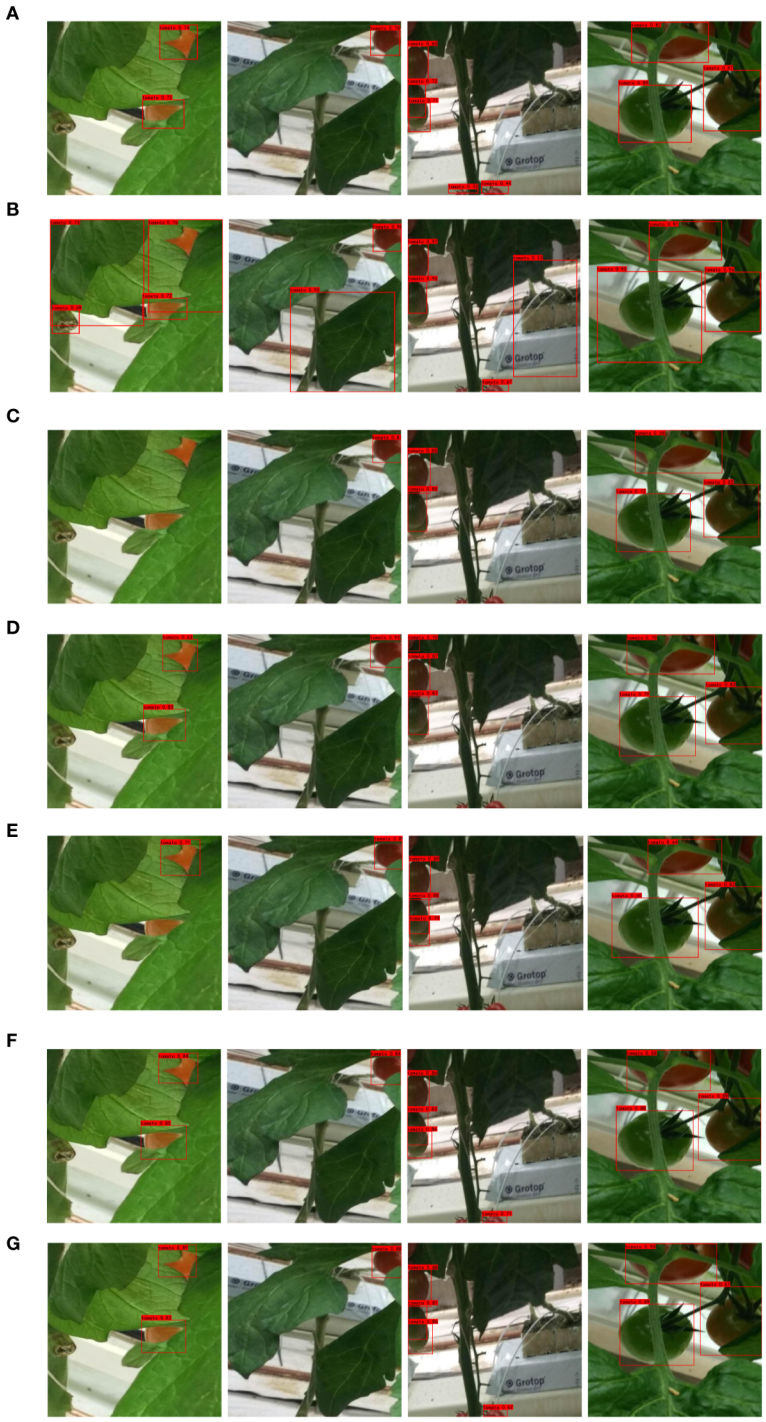


FIGURE 12
Visual detection comparison results of different models. **(A)** CenterNet, **(B)** YOLOv4, **(C)** YOLOv5m, **(D)** YOLOv7, **(E)** YOLOv7x, **(F)** YOLOv8s, **(G)** S-YOLO.

with accurately detecting occluded tomatoes. The overall detection accuracy of CenterNet, YOLOv7, and YOLOv8s is lower than that of the S-YOLO model, with S-YOLO achieving higher accuracy overall. In summary, the S-YOLO model not only achieves lightweight design but also significantly outperforms other models in tomato fruit detection.

5 Discussion

This study investigates an improved lightweight S-YOLO model designed for accurately detecting tomato fruits in greenhouse environments, including occluded and small target tomatoes. It provides a technical reference for the visual system of tomato harvesting robots, addressing issues such as low detection efficiency and accuracy, thus holding considerable practical value.

Previous research has shown limitations in terms of accuracy, lightweight design, or detection speed. In this work, a lightweight GSConv_SlimNeck structure is constructed to optimize the model's neck region. To enhance detection accuracy, the α-SimSPPF structure and β-SIoU loss function are proposed. Additionally, the incorporation of the SE attention module enhances the accuracy of the model. By implementing these enhancements, the proposed S-YOLO model significantly outperforms other object detection models, achieving substantially improved accuracy in tomato detection while maintaining lightweight characteristics. Ultimately, the S-YOLO model achieves 96.60% accuracy, 92.46% mAP@0.5, with a parameter count of only 9.11M and a detection speed of 74.05FPS, demonstrating excellent detection performance.

While this study has made progress in tomato detection in greenhouse environments, there are still limitations to address. For instance, the proposed model may face significant limitations in detection speed when running on low-cost devices. Considering the cost limitations of harvesting robot hardware and the pressing need for real-time detection, future studies should prioritize further size reduction of the model to expedite its processing speed. This will ensure real-time tomato detection and enhance its suitability for integration into the visual systems of tomato harvesting robots.

6 Conclusions

This study introduces a novel model named S-YOLO, characterized by its lightweight design and exceptional accuracy. It effectively addresses the low accuracy in detecting occluded and small tomatoes, providing technical guidance for the visual systems of tomato harvesting robots. Through experimental research and result analysis, the main contributions can be summarized as follows:

1. Lightweight Design: A GSConv_SlimNeck structure suitable for YOLOv8s is constructed to optimize the model's neck region, achieving model lightweightness.
2. Accuracy Improvement: The substitution of the SPPF module with the upgraded α-SimSPPF structure and the replacement of the CIoU loss function with the enhanced β-SIoU loss function contributed to the improved accuracy of the model's detection capabilities.
3. Effective Feature Extraction: Additional SE attention module is introduced to focus on crucial information, further enhancing feature extraction for occluded and small target tomatoes.

Compared to traditional object detection algorithms, S-YOLO demonstrates robustness, lightweight design, and outstanding detection performance, providing technical support for efficiently identifying tomato fruits in tomato harvesting robots. In the future, more tomato fruit images captured in greenhouse environments will be collected, and the model will be further improved in a more lightweight manner to provide stronger technical support for the visual systems of tomato robots.

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

XS: Software, Validation, Writing – original draft, Writing – review & editing, Conceptualization, Data curation, Investigation, Methodology, Resources.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Acknowledgments

Thank you for the guidance from Wenqing Ji, Fenghang Zhang, and Yuxuan Jiang.

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

References

Appe, S. N., Arulselvi, G., and Balaji, G. N. (2023). CAM-YOLO: tomato detection and classification based on improved YOLOv5 using combining attention mechanism. *PeerJ Comput. Sci.* 9, e1463. doi: 10.7717/peerj-cs.1463

Bai, Y., Yu, J., Yang, S., and Ning, J. (2024). An improved YOLO algorithm for detecting flowers and fruits on strawberry seedlings. *Biosyst. Eng.* 237, 1–12. doi: 10.1016/j.biosystemseng.2023.11.008

Bochkovskiy, A., Wang, C. Y., and Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv. arXiv:2004.10934*. doi: 10.48550/arXiv.2004.10934

Chen, M., Chen, Z., Luo, L., Tang, Y., Cheng, J., Wei, H., et al. (2024). Dynamic visual servo control methods for continuous operation of a fruit harvesting robot working throughout an orchard. *Comput. Electron. Agric.* 219, 108774. doi: 10.1016/j.compag.2024.108774

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., and Tian, Q. (2019). "Centernet: Keypoint triplets for object detection," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. (Seoul, Korea (South)). 6569–6578. doi: 10.1109/ICCV.2019.00667

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.* 88, 303–338. doi: 10.1007/s11263-009-0275-4

Feng, Q., Chen, W., and Yang, Q. (2015). Identification and localization of overlapping tomatoes based on linear structured vision system. *J. China Agric. Univ.* 20, 100–106.

Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J. (2021). Yolox: Exceeding yolo series in 2021. *arXiv. arXiv:2107.08430*. doi: 10.48550/arXiv.2107.08430

Gevorgyan, Z. (2022). SIoU loss: more powerful learning for bounding box regression. *arXiv. arXiv:2205.12740*. doi: 10.48550/arXiv.2205.12740

Girshick, R. (2015). "Fast R-CNN," in *2015 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 *2014 IEEE Conference on Computer Vision and Pattern Recognition*. ((Columbus, OH, USA)) 580–587. doi: 10.1109/CVPR.2014.81

Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., and Xu, C. (2020). GhostNet: more features from cheap operations. *arXiv. arXiv:1911.11907*. doi: 10.48550/arXiv.1911.11907

He, J., Erfani, S., Ma, X., Bailey, J., Chi, Y., and Hua, X. S. (2022). Alpha-ioU: A family of power intersection over union losses for bounding box regression. *arXiv. arXiv:2110.13675*. doi: 10.48550/arXiv.2110.13675

Hou, Q., Zhou, D., and Feng, J. (2021). "Coordinate attention for efficient mobile network design," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (Nashville, TN, USA). 13713–13722. doi: 10.1109/CVPR46437.2021.01350

Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (Salt Lake City, UT, USA). 7132–7141. doi: 10.1109/CVPR.2018.00745

Huo, J. Y. (2016). Current situation and safety precaution of tomato industry in China. *Vegetables* 6, 1–4.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., et al. (2022). YOLOv6: A single-stage object detection framework for industrial applications. *arXiv. arXiv:2209.02976*. doi: 10.48550/arXiv.2209.02976

Li, H. L., Li, J., Wei, H. B., Liu, Z., Zhan, Z., and Ren, Q. (2022). Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles. *arXiv. arXiv:2206.02424*. doi: 10.48550/arXiv.2206.02424

Li, H., Gu, Z., He, D., Wang, X., Huang, J., Mo, Y., et al. (2024). A lightweight improved YOLOv5s model and its deployment for detecting pitaya fruits in daytime and nighttime light-supplement environments. *Comput. Electron. Agric.* 220, 108914. doi: 10.1016/j.compag.2024.108914

Li, T. H., Sun, M., Ding, X., Li, Y., Zhang, G., Shi, G., et al. (2021). Tomato recognition method at the ripening stage based on YOLO v4 and HSV. *Trans. Chin. Soc Agric. Eng.* 37, 183–190.

Li, Z., Xu, L., and Zhu, S. (2019). Pruning of network filters for small dataset. *IEEE Access.* 8, 4522–4533. doi: 10.1109/ACCESS.2019.2963080

Liu, J. Z. (2017). Research progress analysis of robotic harvesting technologies in greenhouse. *Trans. Chin. Soc Agric. Mach.* 48, 1–18.

Liu, G., Nouaze, J. C., Touko Mbouembe, P. L., and Kim, J. H. (2020). YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3. *Sensors* 20, 2145. doi: 10.3390/s20072145

Liu, Y. C., Shao, Z. R., and Hoffmann, N. (2021). Global attention mechanism: Retain information to enhance Channel-spatial interactions. *arXiv arXiv:2112.05561*. doi: 10.48550/arXiv.2112.05561

Ma, C., Zhang, X., Li, Y., Lin, S., Xiao, D., and Zhang, L. (2016). Identification of immature tomatoes based on salient region detection and improved Hough transform method. *Trans. Chin. Soc Agric. Eng.* 32, 219–226.

Meng, F., Li, J., Zhang, Y., Qi, S., and Tang, Y. (2023). Transforming unmanned pineapple picking with spatio-temporal convolutional neural networks. *Comput. Electron. Agric.* 214, 108298. doi: 10.1016/j.compag.2023.108298

Ouyang, D., He, S., Zhang, G., Luo, M., Guo, H., Zhan, J., et al. (2023). "Efficient multi-scale attention module with cross-spatial learning," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. (Rhodes Island, Greece). 1–5. doi: 10.1109/ICASSP49357.2023.10096516

Qiu, Z., Zhao, Z., Chen, S., Zeng, J., Huang, Y., and Xiang, B. (2022). Application of an improved YOLOv5 algorithm in real-time detection of foreign objects by ground penetrating radar. *Remote Sens.* 14, 1895. doi: 10.3390/rs14081895

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)*. (Las Vegas, NV, USA). 779–788. doi: 10.1109/CVPR.2016.91

Redmon, J., and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv. arXiv:1804.02767*. doi: 10.48550/arXiv.1804.02767

Reis, D., Kupec, J., Hong, J., and Daoudi, A. (2023). Real-time flying object detection with YOLOv8. *arXiv. arXiv:2305.09972*. doi: 10.48550/arXiv.2305.09972

Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. intelligence.* 39, 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Su, F., Zhao, Y., Wang, G., Liu, P., Yan, Y., and Zu, L. (2022). Tomato maturity classification based on SE-YOLOv3-MobileNetV1 network under nature greenhouse environment. *Agronomy* 12, 1638. doi: 10.3390/agronomy12071638

Tian, S., Fang, C., Zheng, X., and Liu, J. (2024). Lightweight detection method for real-time monitoring tomato growth based on improved YOLOv5s. *IEEE Access.* 12, 29891–29899. doi: 10.1109/ACCESS.2024.3368914

Tian, Y. N., Yang, G. D., Wang, Z., Wang, H., Li, E., and Liang, Z. Z. (2019). Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comput. Electron. Agric.* 157, 417–426. doi: 10.1016/j.compag.2019.01.012

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (Vancouver, BC, Canada). 7464–7475. doi: 10.1109/CVPR52729.2023.00721

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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (Seattle, WA, USA). 11534–11542. doi: 10.1109/CVPR42600.2020.01155

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). 3–19. doi: 10.1007/978-3-030-01234-2_1

Yang, R. J., Li, W. F., Shang, X. N., Zhu, D. P., and Man, X. Y. (2023). KPE-YOLOv5: an improved small target detection algorithm based on YOLOv5. *Electronics* 12, 817. doi: 10.3390/electronics12040817

Yang, L., Zhang, R. Y., Li, L., and Xie, X. (2021). "Simam: A simple, parameter-free attention module for convolutional neural networks," in *International conference on machine learning*. PMLR. 11863–11874.

Zhang, Q. L., and Yang, Y. B. (2021). "SA-net: shuffle attention for deep convolutional neural networks," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. (Toronto, ON, Canada). 2235–2239. doi: 10.1109/ICASSP39728.2021.9414568

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., and Ren, D. (2020). "Distance-ioU loss: faster and better learning for bounding box regression," in *Proceedings of the AAAI Conference on Artificial Intelligence*. (Palo Alto, California USA). Vol. 34. 12993–13000. doi: 10.1609/aaai.v34i07.6999

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 16 September 2024
DOI 10.3389/fpls.2024.1409544

OPEN ACCESS

EDITED BY
Mohsen Yoosefzadeh Najafabadi,
University of Guelph, Canada

REVIEWED BY
Jakub Nalepa,
Silesian University of Technology, Poland
Sreedevi A.,
K L University, India

*CORRESPONDENCE
Diye Xin
21013184@mail.ecust.edu.cn

†These authors have contributed equally to this work

RECEIVED 30 March 2024
ACCEPTED 22 August 2024
PUBLISHED 16 September 2024

CITATION
Xin D and Li T (2024) Revolutionizing tomato disease detection in complex environments. *Front. Plant Sci.* 15:1409544. doi: 10.3389/fpls.2024.1409544

Revolutionizing tomato disease detection in complex environments

Diye Xin[1*†] and Tianqi Li[2†]

[1]East China University of Science and Technology, School of Information Science and Engineering, Shanghai, China, [2]East China University of Science and Technology, School of Biotechnology, Shanghai, China

In the current agricultural landscape, a significant portion of tomato plants suffer from leaf diseases, posing a major challenge to manual detection due to the task's extensive scope. Existing detection algorithms struggle to balance speed with accuracy, especially when identifying small-scale leaf diseases across diverse settings. Addressing this need, this study presents FCHF-DETR (Faster-Cascaded-attention-High-feature-fusion-Focaler Detection-Transformer), an innovative, high-precision, and lightweight detection algorithm based on RT-DETR-R18 (Real-Time-Detection-Transformer-ResNet18). The algorithm was developed using a carefully curated dataset of 3147 RGB images, showcasing tomato leaf diseases across a range of scenes and resolutions. FasterNet replaces ResNet18 in the algorithm's backbone network, aimed at reducing the model's size and improving memory efficiency. Additionally, replacing the conventional AIFI (Attention-based Intra-scale Feature Interaction) module with Cascaded Group Attention and the original CCFM (CNN-based Cross-scale Feature-fusion Module) module with HSFPN (High-Level Screening-feature Fusion Pyramid Networks) in the Efficient Hybrid Encoder significantly enhanced detection accuracy without greatly affecting efficiency. To tackle the challenge of identifying challenging samples, the Focaler-CIoU loss function was incorporated, refining the model's performance throughout the dataset. Empirical results show that FCHF-DETR achieved 96.4% Precision, 96.7% Recall, 89.1% mAP (Mean Average Precision) 50-95 and 97.2% mAP50 on the test set, with a reduction of 9.2G in FLOPs (floating point of operations) and 3.6M in parameters. These findings clearly demonstrate that the proposed method improves detection accuracy and reduces computational complexity, addressing the dual challenges of precision and efficiency in tomato leaf disease detection.

KEYWORDS

tomato leaf disease, Cascaded Group Attention, Real-Time-Detection-Transformer, lightweight backbone, feature fusion, Focaler-CIoU loss function

1 Introduction

Tomatoes, rich in nutritional and medicinal value, are among the most significant crops cultivated globally. China ranks as a leading tomato producer globally (Coelho et al., 2023). In 2023, China, leveraging its vast agricultural landscape and favorable climate, solidified its status as the top tomato producer worldwide, contributing 67 million tons to the global total of approximately 190 million tons. This substantial output underscores China's dominance in the global tomato market (Min, 2023). Moreover, China's 2023 tomato production (Lu et al., 2023) exceeded initial forecasts, reaching 8 million tons, up from the predicted 7.3 million tons.

However, tomatoes face threats from various leaf diseases, including spot disease and leaf mold (Lee, 2022), caused by fungi, bacteria, and environmental stressors (Hernandez et al., 2021). Untimely detection and prevention can drastically reduce tomato yield and quality, resulting in significant economic losses for farmers.

Traditionally, tomato leaf disease detection has been manual, presenting numerous limitations and challenges. First, it depends on professional inspectors, leading to significant human resource constraints (Geisseler and Horwath, 2014). Second, factors like visual fatigue compromise the method's accuracy. In large-scale settings like tomato plantations, manual detection becomes labor-intensive, increasing the risk of missed detections and false alarms (Lambooij et al., 2009). Consequently, automating tomato leaf detection has emerged as a key research focus to enhance efficiency and accuracy (Azim et al., 2014).

Advancements in computer technology have facilitated the incorporation of machine learning into agricultural research (Pallathadka et al., 2022)'s study preprocesses images with histogram equalization, followed by principal component analysis for feature extraction. Support vector machines and naive Bayesian classifiers are then employed for rice leaf disease classification. However (Sujatha et al., 2021), notes that machine learning's extensive computational demands in preprocessing and feature extraction limit its practical application. Comparative studies have shown deep learning's superior efficacy in plant leaf disease recognition, with convolutional neural networks (LeCun et al., 1998) and residual structures (He et al., 2016) leading to significant advancements in object detection algorithms, including the evolution to one-stage approaches like DETR with transformers. DETR (Detection Transformer) is an innovative object detection approach that utilizes transformers, which are originally designed for natural language processing tasks. By leveraging transformers, DETR simplifies the object detection pipeline, eliminating the need for hand-crafted components such as anchor generation and non-maximum suppression, and allows for direct end-to-end object detection with improved accuracy and efficiency.

Notably, two-stage models such as Faster RCNN (Region-based Convolutional Neural Network) (Ren et al., 2016) and Mask RCNN (He et al., 2017) have been typical (Teng et al., 2022). enhances pest detection with super-resolution modules (Dong et al., 2015) and Soft IoU (Rahman and Wang, 2016) mechanisms, achieving 67.4% accuracy on a pest dataset (Saleem et al., 2022). optimizes weed detection using Faster RCNN ResNet-101, with an enhanced anchor box method (Redmon and Farhadi, 2018) that refines region proposals and improves accuracy. RCNN3's Mask RCNN-based algorithm (Wang et al., 2021) for crop images introduces path aggregation and feature enhancements (Liu et al., 2018), increasing edge accuracy with a micro fully connected layer (Lin et al., 2013). Despite these improvements, the large size, numerous parameters, and high computational costs challenge the practicality of two-stage algorithms.

Common one-stage algorithms encompass SSD (Single Shot MultiBox Detector) (Liu et al., 2016), YOLO v5(You-Only-Look-Once) (Jocher et al., 2022), YOLOv7 (Wang et al., 2023), and YOLOv9 (Wang et al., 2024) (Wang et al., 2022)'s YOLOv5 significantly enhances weed detection accuracy and speed via data augmentation (Simard et al., 2003) and converter encoder modules (Zhang et al., 2022). Experimental results indicate that the improved network surpasses the baseline YOLOv5 in F1 score, AP, and mAP@0.5 by 11.8%, 11.3%, and 5.9%, respectively (Zhang et al., 2023)'s study introduced a lightweight agricultural pest identification method using an enhanced Yolov5s, merged with MobileNetV3 (Howard et al., 2019), significantly lowering the network's parameter count. Additionally, the study integrated the ECA (Efficient Channel Attention) attention (Wang et al., 2020) mechanism into MobileNetV3's shallow network to boost performance. Experimental results reveal that compared to Yolov5s, their model cuts parameters by 80.3% with only a 0.8% drop in mAP, achieving a real-time detection speed of 15.2 FPS on embedded devices, outperforming the original model by 5.7 FPS.

The aforementioned one-stage algorithms have seen substantial optimization in speed and scale, yet their accuracy falls short of two-stage algorithms, rendering them less suited for high-precision applications in sectors like industry, agriculture, and emerging technologies (Agarwal et al., 2020). introduces a deep learning model with three convolutional layers and three max pooling layers for tomato leaf disease detection and classification. Outperforming established models like VGG (Visual Geometry Group)16, InceptionV3, and MobileNet, it achieves a classification accuracy of 91.2%. The study employs data augmentation and hyperparameter tuning to aid farmers in managing tomato diseases, enhancing crop yield and quality. Additionally, the DETR algorithm has shown significant accuracy in crop detection. The recent DETR (Carion et al., 2020) algorithm has also demonstrated notable accuracy in crop detection (Yang et al., 2023). introduces a DETR-based rice leaf disease detection algorithm, leveraging an enhanced detection transformer for diagnosis and recognition. Introducing the Neck structure and the Dense Higher Level Composition Feature Pyramid Network (Gao et al., 2019), based on FPN (Feature Pyramid Network), improves small disease target detection accuracy. However, DETR's computational intensity, exacerbated by enhanced feature extraction, results in less favorable detection speeds and model parameters.

To facilitate a clearer understanding of the progress in this field, the methods utilized in the referenced literature are summarized in Table 1.

TABLE 1 Summary of detection methods for tomato leaf disease.

Method	Dataset	Train & Test	mAP50	FPS
Manual detection (Geisseler and Horwath, 2014)				
Automated detection technology (Azim et al., 2014)				
Support vector machines and Naive Bayesian classifiers (Pallathadka et al., 2022)	Rice Leaf Disease	Not mentioned		
Inception V3 (Sujatha et al., 2021)	Citrus leaf disease dataset	9:1	89.2	
Multi-Scale Super-Resolution RCNN (Teng et al., 2022)	Capured by Chinese Intelligent Machines Institute	8:2	67.4	
Enhanced Anchor Box-RCNN (Saleem et al., 2022)	DeepWeeds dataset	9:1	96.2	
Segmentation and Extraction Algorithm Based on Mask RCNN (Wang et al., 2021)	Fruit 360 dataset	9:1	94.9	
Real-time detection YOLOv5 (Wang et al., 2022)	Sugarbeet image dataset	9:1	90.0	20.8
Lightweight detection YOLOv5 (Zhang et al., 2023)	Large-scale open-source dataset IP102	9:1	98.6	15.2
CNN disease detection (Agarwal et al., 2020)	Tomato leaves dataset from plantvillage	20:1	91.2	
Dense Higher-Level Composition DETR (Yang et al., 2023)	IDADP dataset	Not mentioned	93.5	24.4

The motivation for developing the FCHF-DETR model arises from the serious economic losses and social impacts resulting from global crop diseases. Many farmers depend on the yield and quality of their crops for their livelihoods, and disease outbreaks not only threaten their food security, but can also inflict serious damage on the economic structure of entire regions.

In this context, the need for precise and timely disease detection is critical. The FCHF-DETR model employs advanced deep learning and real-time detection techniques to rapidly and precisely identify plant diseases in the field. This capability not only enables farmers to take timely measures to mitigate losses, but also offers a more stable and reliable management approach for agricultural production, thus effectively reducing the economic and social pressures arising from diseases.

Furthermore, the lightweight design of the FCHF-DETR model allows it to operate efficiently in resource-limited environments, critical for resource-poor agricultural areas. This design permits unrestricted model deployment across various hardware platforms, enabling farmers worldwide to utilize this technology and thereby enhance the sustainability and resilience of global agricultural production.

In summary, researchers have introduced numerous innovative methods and technologies in the field of object detection, which have significantly advanced the progress of plant disease management technology. To enhance applicability in crop production environments, this study introduces an accurate and lightweight tomato leaf disease detection model based on RT-DETR-R18. This model is characterized by its lightweight design, high detection accuracy, and rapid processing speed, facilitating easy deployment on farm detection equipment. The main contributions of this study include:

1. The integration of the lightweight and efficient Fasternet in lieu of the ResNet18 backbone network enhances the feature extraction speed by mitigating memory access and computational redundancy through the use of PConv (Partial Convolution) in Fasternet. This modification not only optimizes memory efficiency but also reduces the overall size of the model.
2. The substitution of the Attention-based Intra-scale Feature Interaction (AIFI) module with Cascaded Group Attention (CGA) within the Efficient Hybrid Encoder not only curtails computational expenditure but also enriches attention diversity. This is achieved by layering attention maps from different heads, facilitating a dual enhancement in both efficiency and accuracy.
3. The replacement of the High-Level Screening-feature Fusion Pyramid Networks (HSFPN) module with the CNN-based Cross-scale Feature-fusion Module (CCFM) module for inter-scale feature fusion within the Efficient Hybrid Encoder incorporates a channel attention mechanism. Given the dataset's variety in terms of the types and sizes of diseased leaves, HSFPN adeptly assimilates global features across varying scales, synergizing with the decoder to accurately pinpoint locations.
4. Acknowledging the dataset's heterogeneity and the varying levels of detection difficulty presented by diseased leaves,

the model adopts the Focaler-IoU loss function in place of the conventional IoU loss. This strategic alteration aims at honing the focus on more challenging samples without amplifying the parameter count or computational complexity, thereby enhancing accuracy.

In the second section, we will delve into the dataset and the overarching architecture of FCHF-DETR. Moving on to the third section, we will undertake a series of ablation studies to dissect the impact of different modules on FCHF-DETR's performance, complemented by visual illustrations. The fourth section is dedicated to a comparative analysis, highlighting the merits of our model vis-à-vis the prevalent RT-DETR-R18, and discussing prospective avenues for refinement. We will conclude by encapsulating the essence of our model and exploring its potential implications for practical applications.

2 Materials and methods

2.1 Data collection

To improve the model's generalization, the dataset includes tomato leaves photographed from multiple perspectives, backgrounds, lighting conditions, and featuring different disease types. A large collection of images was curated to enable accurate detection of minor diseases. However, due to the scarcity of public tomato leaf disease datasets, this study utilized the Tomato Leaf Diseases Detection Computer Vision dataset (Figure 1A) and the Tomato Disease Multiple Sources dataset (Figure 1B) from Kaggle. Despite their usefulness, these datasets have limitations, especially the oversimplified backgrounds with isolated leaves, which differ from real-world scenarios.

To overcome this and enhance the model's ability to detect small-scale diseases, we augmented these datasets with 512 additional tomato leaf photos we collected (Figure 1C), creating a comprehensive dataset of 3147 images for this experiment. This carefully curated image collection features specimens of various resolutions and sizes, taken from many angles to ensure data diversity (Figure 1). The detailed presentation of tomato leaves closely mirrors actual detection settings, including the effects of natural elements like lighting and shadows. To simulate rainy-day detection conditions, we deliberately reduced the clarity of some images, emulating real-world challenges and enhancing the model's robustness and applicability.

Images in the dataset were classified into five categories using LabelMe software: 'Late blight leaf', 'Early blight leaf', 'Septoria leaf spot', 'Mold leaf', and 'Yellow virus leaf'. In the experimental setup, the dataset was divided into training, validation, and testing sets in an 8:1:1 ratio.

2.2 Date preprocessing

During data preprocessing, we utilized the Mosaic data augmentation technique (Bochkovskiy et al., 2020) to combine four unique images into one composite image. This composite image undergoes random scaling, flipping, shifting, and color adjustments to enhance the model's generalization ability. This technique enriches the dataset with extensive contextual details and various object instances in each synthesized image, as shown in Figure 2.

In tomato leaf disease detection, the uneven distribution of smaller target samples could hinder the model's training efficiency. Using the Mosaic augmentation not only increases the sample volume but also balances the distribution of smaller targets, improving the model's ability to detect them. Visualizing the disease targets and bounding boxes clarifies the spatial distribution of label centroids, with 'x' and 'y' axes representing the centroids' coordinates and color intensity indicating proximity to the image center.

The visualization (Figure 3) highlights the distribution of target box sizes in the dataset, showing a relatively uniform color gradient across the image. This uniform color gradient suggests a balanced mix of large and small targets, achieved by careful preprocessing of the defect bounding boxes. This processing ensures fair representation of all target sizes in the dataset, counteracting any original bias towards larger defects. Aiming for a uniform distribution of defect sizes enhances the model's ability to detect

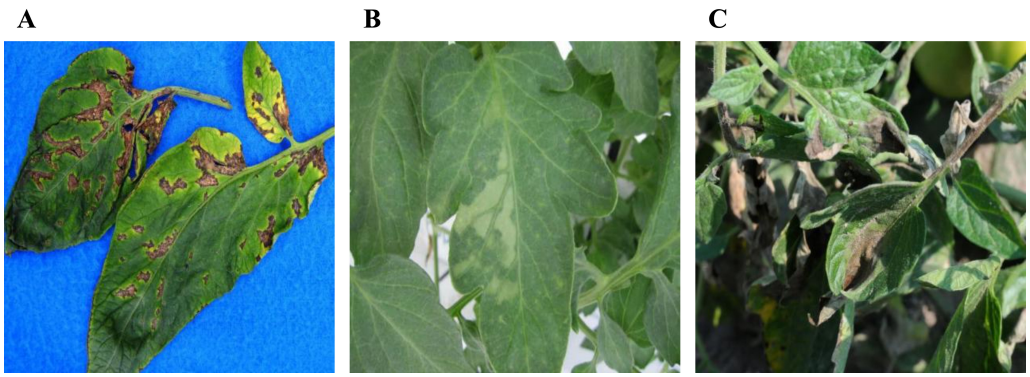


FIGURE 1
Samples of dataset, where **(A)** is the data from Tomato Leaf Diseases Detection Computer Vision dataset, **(B)** comes from Tomato Disease Multiple Sources dataset, **(C)** is the data collected for this paper's research.

FIGURE 2
Mosaic data augmentation, randomly combining four pictures together.

anomalies at various scales. This approach reduces size-related bias during training, enabling the model to accurately identify defects of different sizes in real scenarios. Ultimately, this preprocessing effort boosts the model's generalization and balances its detection ability, leading to enhanced overall performance.

2.3 Overall structure of FCHF-DETR

This study presents the FCHF-DETR model (Figure 4), a streamlined yet precise detection network for various tomato leaf diseases, based on the RT-DETR-R18 (Lv et al., 2024) framework. The detailed structure of the proposed FCHF-DETR model is outlined below.

RT-DETR-R18 and the newly introduced FCHF-DETR are based on three main components: the Backbone, the Hybrid Encoder, and the Transformer Decoder. The Backbone acts as a feature extraction unit, effectively distilling multi-level features from input images, especially from the last three stages, S3, S4, and S5. These features are then fed into the Hybrid Encoder for further processing, which includes the AIFI module focusing on S5 feature maps to enhance precision and reduce complexity, and the CCFM module working with S3 and S4 features, using fusion blocks for feature amalgamation, refined by 1x1 convolutions.

RT-DETR-R18's original backbone, based on ResNet18, contained numerous convolutional modules, hindering real-time detection and mobile deployment. Additionally, early versions of the AIFI module did not significantly improve accuracy. To address these challenges, this study introduces the FCHF-DETR approach, carefully crafted for efficient and accurate tomato leaf disease detection. Key improvements include integrating FasterNet instead of ResNet18 and adding PConv layers to enhance feature extraction speed and reduce model size; replacing the AIFI module with Cascaded Group Attention for increased efficiency; substituting the CCFM module with HSFPN for better feature fusion; and adopting the Focaler-IoU loss function to improve accuracy for difficult samples without increasing complexity.

2.3.1 Lightweight network establishment

RT-DETR-R18's ResNet-18 backbone, filled with convolutional modules, results in high computational needs and a large parameter count. Targeting mobile device deployment, this study prioritizes precise detection, faster inference, fewer

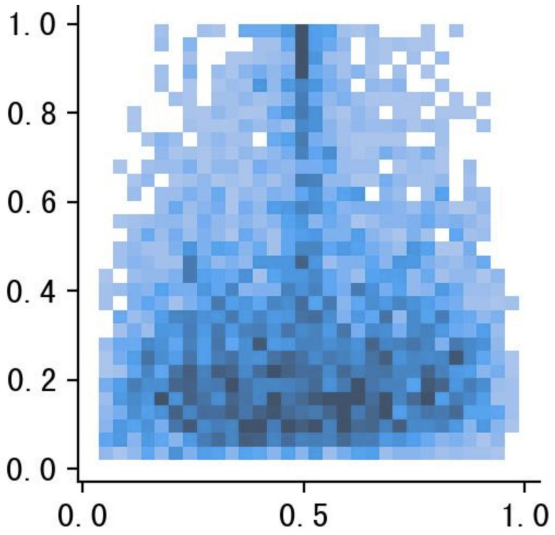


FIGURE 3
Tomato leaf disease size quantification framework based on target detection.

parameters, and improved device compatibility. FCHF-DETR features a streamlined Backbone with FasterNet (Chen et al., 2023), balancing quick processing and accuracy, as depicted in Figure 5. FasterNet's core includes FasterNet Blocks and PConv layers, dynamically adjusting convolution ranges based on data relevance for efficient processing.

2.3.1.1 Partial convolution

Partial convolution, or PConv, uses a unique binary masking technique to accurately distinguish valid from invalid data points. It dynamically adjusts the convolution kernel's reach according to this distinction, focusing the convolution process on valid data. This method greatly enhances the model's resilience in data incompleteness scenarios, preserving maximum information and minimizing data gap impacts. Compared to traditional convolutions (Figure 6A), PConv provides greater flexibility, efficiency, and precision in processing datasets with missing entries. Unlike Depth-Wise (Figure 6B) separable convolution (Chollet, 2017), known for fewer parameters and efficiency, PConv excels in managing complex imaging tasks with missing areas (Figure 6C). This suitability makes PConv ideal for applications like image restoration and content filling, where it effectively addresses image voids.

Given the similarity across feature maps of different channels, PConv efficiently performs convolution on a subset of input channels to extract spatial features, as shown in the Figure 6C. This method leaves the other channels unchanged. Assuming equal channel counts for input and output feature maps, PConv's computational complexity, in terms of FLOPs, is significantly reduced:

$$FLOPs_{PConv} = h \times w \times k^2 \times c_p^2$$

Where:
h, w are the width and height of the feature map,
k is the size of the convolution kernel,
c_p is the number of channels for conventional convolution.

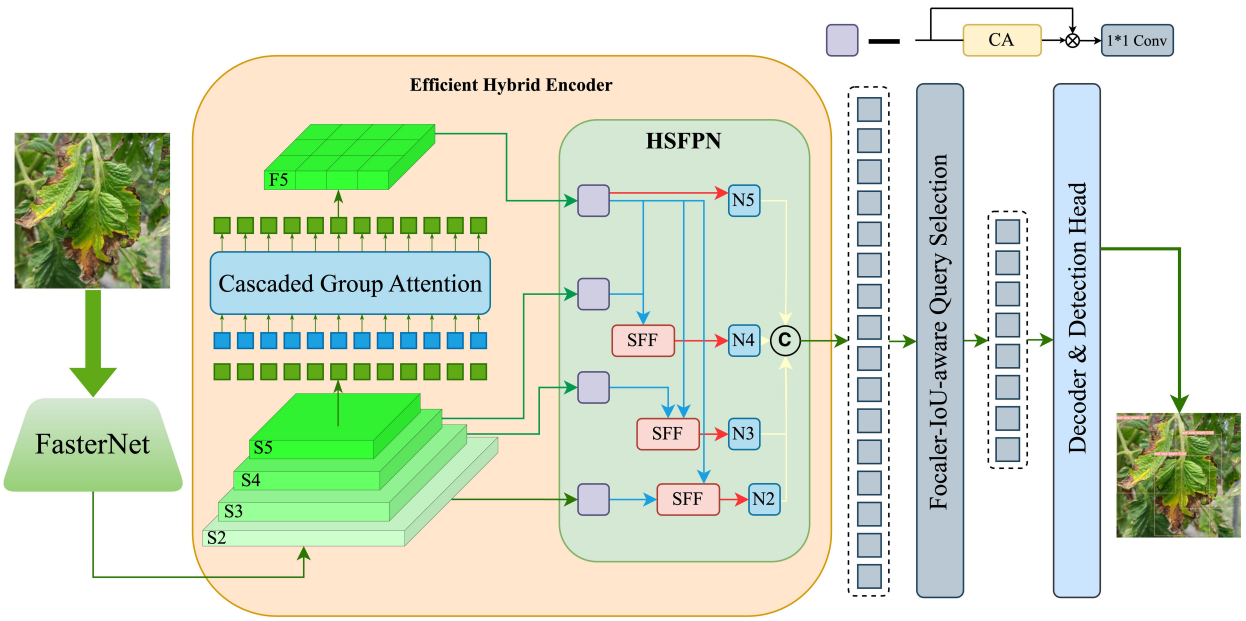


FIGURE 4
The overall architecture of FCHF-DETR, which contains Backbone, Cascaded group attention (CGA), High-level screening feature-fusion pyramid network (HSFPN), Focaler-CIoU loss function and Detection head.

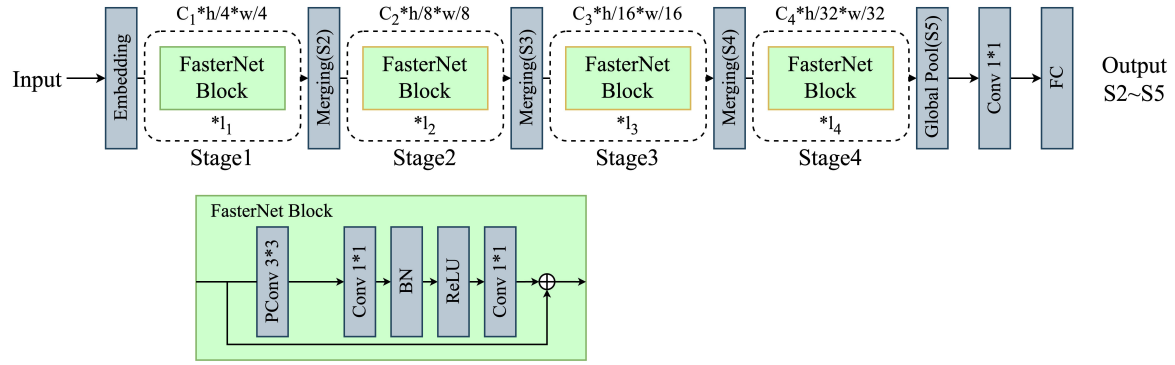


FIGURE 5
Fasternet's backbone leverages deep learning for efficient feature extraction and accelerated neural network computations.

In practical implementation, there is generally $r = c_p/c = 1/4$, so the FLOPs of PConv are only 1/16 of those of conventional convolutions.

Memory access status of PConv:

$$MEM_{PConv} = h \times w \times 2c_p + k^2 \times c_p^2 \approx h \times w \times 2c_p$$

Where:

h, w are the width and height of the feature map,

k is the size of the convolution kernel,

c_p is the number of channels for conventional convolution.

The memory access count of PConv is only 1/4 of that of regular convolution, and the remaining $(c - c_p)$ channels do not participate in the calculation, so there is no need for memory access.

RT-DETR-R18's backbone network focuses on improving detection accuracy with a complex structure and more parameters for slightly enhanced capabilities. However, this approach may impact computational and memory efficiency. In fast-processing and resource-limited scenarios, like tomato leaf disease detection, FasterNet's streamlined architecture could provide a better balance of accuracy and efficiency.

2.3.2 Cascaded group attention

The attention mechanism is pivotal in tomato leaf disease recognition, with its primary capability being the substantial enhancement of recognition accuracy and processing efficiency through the focus on and emphasis of key features related to diseases in images. In environments characterized by complex backgrounds or varied disease manifestations, traditional image recognition techniques can overlook important details or result in misjudgments due to information overload. In contrast, the

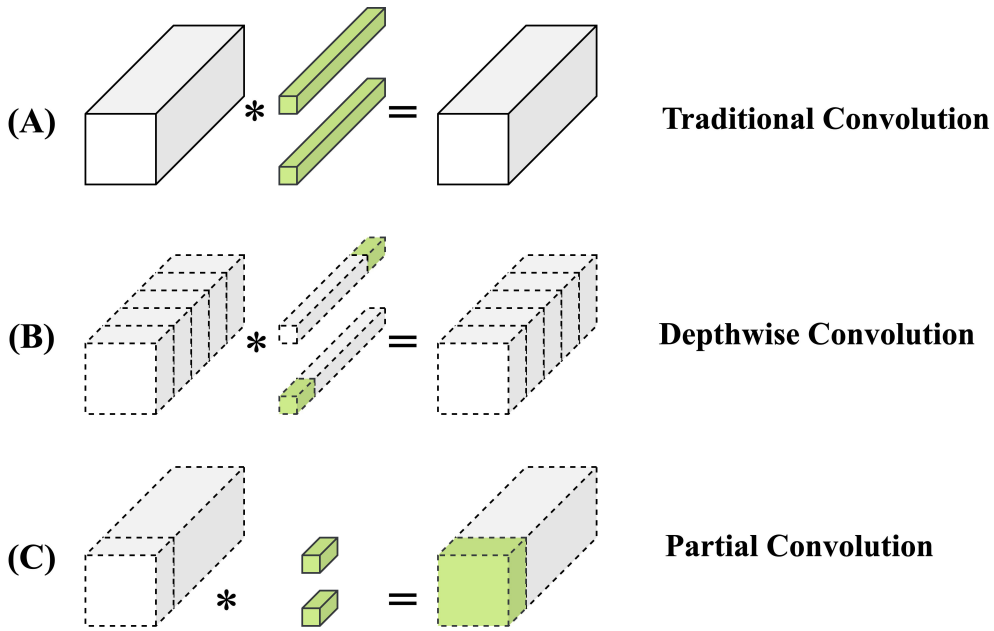


FIGURE 6
(A) Standard convolution applies filters across the entire input. **(B)** Depth-Wise convolution separates channels for independent processing. **(C)** Partial convolution dynamically adapts to missing data areas.

attention mechanism significantly improves the model's effectiveness in distinguishing between healthy and diseased leaves through the construction of rich feature interactions and the optimization of importance allocation. This mechanism guarantees that the model maintains high recognition accuracy even amidst complex backgrounds or in cases of unclear symptoms.

We've incorporated the Cascaded Group Attention (CGA) (Chen et al., 2023) mechanism, shown in Figure 7, to effectively address the computational efficiency challenges often found with the SE attention (Hu et al., 2018) approach. Traditional mechanisms such as SimAM (Yang et al., 2021) falter in complex scenes, and CBAM's (Woo et al., 2018) complexity may overload the model, slowing down inference. Unlike SE, CA, and CBAM, CGA excels in nuanced feature processing via systematic grading and grouping, enhancing feature differentiation. CGA highlights inter-channel and spatial relationships and uses a cascaded framework to enrich layers with informative attention outputs. This progressive approach makes CGA highly adaptable and effective in managing complex features, providing a balanced depth and breadth in analysis.

$$\tilde{x}_{ij} = Attn(X_{ij}W_{ij}^{Q}, X_{ij}W_{ij}^{K}, X_{ij}W_{ij}^{V})$$

$$\tilde{x}_{i+1} = Concat[\tilde{x}_{i,j}]_{j=1:h}W_{i}^{P}$$

Where:

j-th head computes the self-attention over X_{ij} represents the j-th split of the input feature X_i, $i.e.$,

$X_i = [X_{i1}, X_{i2}, \ldots\ldots, X_{ih}]$ and $1 \le j \le h$, h represents the total number of heads,

W_{ij}^{Q}, W_{ij}^{K}, W_{ij}^{V} represent projection layers mapping the input feature split into different subspaces,

W_i^P represents a linear layer that projects the concatenated output features back to the dimension consistent with the input.

Using feature segmentation instead of the full feature set for each attention head is more efficient and reduces computational cost. While effective, this approach can be improved by enabling the Q, K, and V layers to project richer features, thus enhancing their capabilities. A cascading strategy for attention maps, as shown in the Figure 7, involves incrementally adding each head's output to the next, enhancing feature refinement. This systematic accumulation enables progressive refinement of feature representation:

$$X_{ij}^{'} = X_{ij} + \tilde{X}_{i(j-1)}, \quad 1 < j \le h$$

Where:

$X_{ij}^{'}$ represents the addition of the $j-th$ input split X_{ij} and the $(j-1)$-th head output $\tilde{X}_{i(j-1)}$.

In the self-attention computation, we redefine X_{ij} as the novel input feature for the $j-th$ attention head. Furthermore, we've introduced an additional Token Interaction layer post Q-projection, enriching the self-attention mechanism's capability to concurrently apprehend local and global relationships, thereby amplifying the feature representation.

In our work, we replaced RT-DETR-R18's original AIFI module with the CGA approach, yielding two key advantages. Firstly, varied feature segmentation for each head enhances attention map diversity. This is similar to group convolution, where cascaded group attention can save Flops and parameters by a factor of h. Secondly, layering the attention heads deepens the network, enhancing capacity without additional parameters. With reduced channel dimensions for Q and K in attention map computations, the resulting latency overhead is minimal. This refined approach enables precise disease localization across sizes in tomato leaf disease detection, significantly improving detection accuracy.

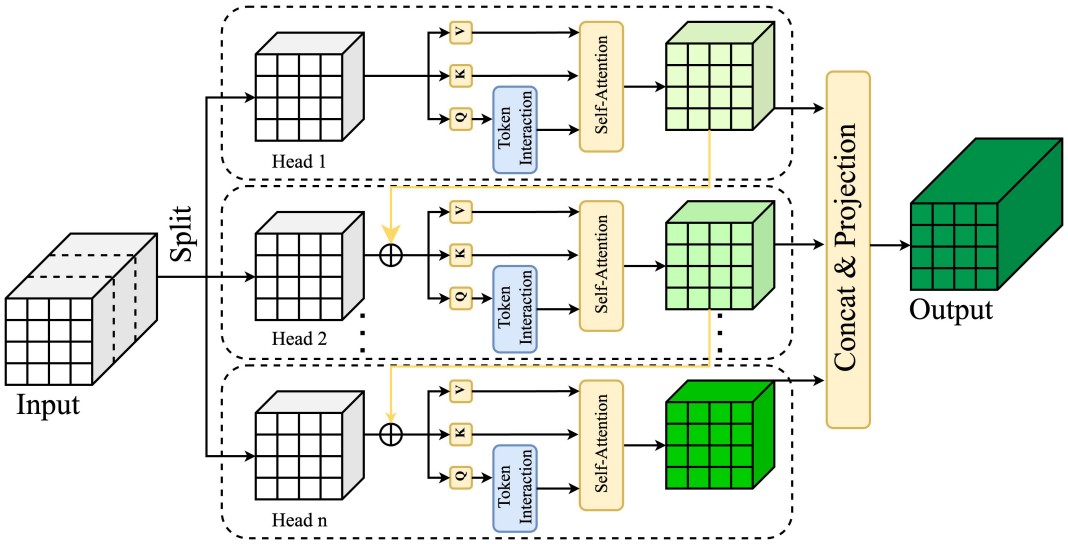


FIGURE 7
Cascaded Group Attention employs sequential attention layers, grouping features to focus progressively, enhancing representation by refining attention at multiple scales for improved contextual learning.

2.3.3 High-Level Screening-feature Fusion Pyramid Networks

The High-Level Screening-feature Fusion Pyramid Network (HSFPN) (Chen et al., 2024) is crafted to build hierarchical feature pyramids attuned to scale variations, as shown in Figure 8. This design allows HSFPN to precisely detect disease features on tomato leaves, varying in size and shape, thus improving detection accuracy and robustness. Furthermore, HSFPN's layered approach to feature fusion preserves detailed information, crucial for identifying early-stage or subtle leaf disease indicators. Consequently, HSFPN outperforms CCFM, particularly in complex agricultural settings and in detecting finely detailed objects.

2.3.3.1 Selective Feature Fusion

Selective Feature Fusion (SFF), key to HSFPN, shown in Figure 9, crucially combines feature maps from various scales. The SFF module uses higher-level features as weights to filter through and selectively extract relevant information from low-level features. This involves scaling higher-level features to match low-level feature dimensions, using methods like transposed convolution and bilinear interpolation. Then, these scaled higher-level features act as attention weights to highlight valuable insights from low-level features. This fusion strategy effectively combines the semantic depth of high-level features with the detailed nuances of low-level features, greatly improving the model's ability to handle multi-scale data challenges.

Given a high-level feature $f_{high} \in R^{C*H*W}$ and a low-level feature $f_{low} \in R^{C*H_1*W_1}$, the process begins by expanding f_{high} through a transposed convolution operation. This operation utilizes a stride of 2 and a kernel size of 3×3, enlarging f_{high} to a new dimension $R^{C*2H*2W}$.

Following this, to reconcile the dimensions of the high-level and low-level features, bilinear interpolation is employed to either upscale or downscale the high-level features. This adjustment results in a feature f_{att} that matches the low-level feature dimensions in $R^{C*H_1*W_1}$, thus facilitating their subsequent integration:

$$f_{att} = BL(T - Conv(f_{high}))$$

$$f_{att} = f_{low} * CA(f_{att}) + f_{att}$$

Next, use the CA module to convert advanced features into corresponding attention weights to filter out low-level features, after obtaining features with the same dimension. Finally, the filtered low-level features are fused with high-level features to enhance the feature representation of the model and obtain $f_{out} \in R^{C*H_1*W_1}$.

Integrating HSFPN with CCFM significantly enhances disease detection precision in tomato leaf images, especially for size-varying disease manifestations. HSFPN's layered feature pyramid architecture skillfully captures and defines features across scales, greatly improving the model's sensitivity and accuracy in identifying disease stages, from small lesions to widespread areas.

HSFPN's strategic use of multi-scale features not only strengthens the model's ability to detect small targets but also maintains accuracy for larger ones. This dual strength effectively addresses traditional challenges in detecting varying disease sizes, offering robust support for precision agriculture's complex requirements.

2.3.4 Focaler-CIoU

Sample imbalance is a common issue in object detection, typically appearing as simple and difficult samples, categorized by target size. Simple samples involve easier-to-detect targets, while difficult samples include very small targets, challenging accurate localization.

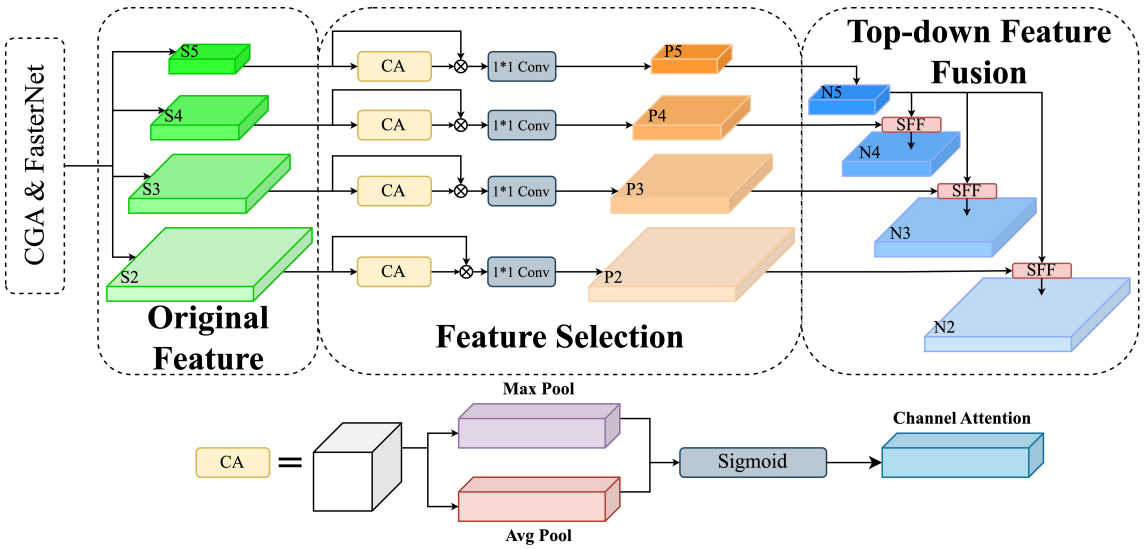


FIGURE 8
High-Level Screening-Feature Fusion Pyramid Networks (HSFPN) integrate multi-scale features with high-level screening for enhanced object detection, achieving superior performance through hierarchical feature fusion.

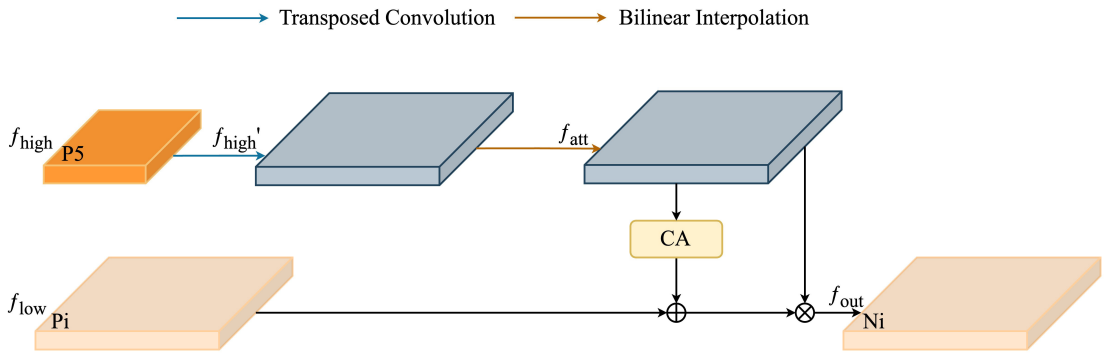


FIGURE 9
Selective feature fusion in High-Level Screening-Feature Fusion Pyramid Networks intelligently merges critical high-level features, enhancing object detection by optimizing feature representation.

In tasks with mainly simple samples, focusing on bounding box regression for these targets can significantly improve detection. Conversely, in scenarios with prevalent difficult samples, refining regression for these targets becomes essential. To address this variance, the IoU loss function can be adapted using a linear interval mapping method (Zhang and Zhang, 2024). This method enables flexible adjustment between simple and difficult samples, fine-tuning bounding box regression accuracy and improving detection performance. The modified IoU loss function, designed to address sample imbalance, is mathematically defined as follows:

$$IoU = \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|}$$

$$IoU^{focaler} = \begin{cases} 0, & IoU < d \\ \frac{IoU-d}{u-d}, & d \ll IoU \ll u \\ 1, & IoU > u \end{cases}$$

Where:
B represents the predicted box
B^{gt} represents the GT (goal target) box
$IoU^{focaler}$ is the reconstructed Focaler-IoU
IoU is the original IoU value

$$[d, u] \in [0, 1]$$

Applying Focaler-IoU loss to existing IoU based bounding box regression loss function CIoU:

$$CIoU = IoU - \frac{\rho^2(b, b^{gt})}{c^2} - \alpha v$$

$$\alpha = \frac{v}{(1 - IoU) + v}$$

$$v = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2$$

$$L_{Focaler-CIoU} = L_{CIoU} + IoU - IoU^{Focaler}$$

Where:
b represents the center points of anchor box
b^{gt} represents the center points of GT box
$\rho(\cdot)$ represents the Euclidean distance
c represents the diagonal minimum distance enclosing bounding box between b and b^{gt}
w^{gt} represents the width of GT box
h^{gt} represents the height of GT box
w represents the width of anchor box
h represents the height of anchor box

In the field of tomato leaf disease detection, the Focaler-CIoU loss function offers significant advantages over the loss function originally used in RTDETR. Focaler-CIoU enhances the model's ability to recognize challenging samples by adjusting the loss function to focus on samples of varying difficulty levels, particularly for disease samples that are challenging to distinguish or have indistinct boundaries, by assigning higher weights. This is particularly important when dealing with lesions of varied sizes and shapes on tomato leaves, as accurately identifying these diseases in their early stages is often challenging. The characteristic of Focaler-CIoU can significantly enhance the sensitivity in detecting early or minor lesions, lower the rate of missed detections, and thus boost the overall detection efficiency while maintaining high accuracy. It holds considerable importance in enhancing the early prevention and control of tomato leaf diseases.

3 Results

This section details the experimental, hyperparameter settings, and training strategies in Section 3.1. Section 3.2 describes the indicators and calculation formulas employed to evaluate model performance. Sections 3.3 and 3.4 discuss the study's results, utilizing ablation experiments and visual displays, respectively.

3.1 Experimental setup

The experiment utilized an OpenBayes cloud server equipped with an Nvidia A100 80GB MIG 1g.10g graphics card, boasting

16GB of graphics memory, and ran on a Linux operating system. This experiment was implemented using Python 3.10 and Cuda11.8.

The model training strategy entailed: For IoU-aware query selection, the first 300 encoder features were selected to initialize the decoder's object query. Training employed the AdamW optimizer, with a base learning rate of 0.0001, weight decay of 0.0001, global gradient clipping norm of 0.0001, 2000 linear warm-up steps, and spanned 100 epochs.

3.2 Evaluation indicators

In the field of object detection, performance is primarily evaluated by Precision (P), Recall (R), and Mean Average Precision (mAP). Precision represents the ratio of correctly predicted positive samples to all samples labeled as positive by the model. Recall measures the proportion of correctly identified positive samples among all actual positive samples. mAP denotes the mean of the average precisions across all categories. The corresponding formulas for Recall, Precision, and mAP are provided below:

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

$$AP = \int_0^1 P(R)dR$$

$$mAP = \frac{\sum_{i=1}^{n} AP_i}{n}$$

TP (True Positive) refers to correctly identified positives, FN (False Negative) to positives incorrectly labeled as negatives, and FP (False Positive) to negatives incorrectly labeled as positives. Precision (P) is the ratio of correctly predicted positive observations to the total predicted positives, while Recall (R) is the ratio of correctly predicted positive observations to all actual positives. The area under the curve drawn through Precision (P) and Recall (R) values on the PR graph represents the Average Precision (AP), and the mean of AP values across all categories yields the Mean Average Precision (mAP).

Beyond the aforementioned performance metrics, model size and computational cost are assessed using the number of parameters and FLOPs, to facilitate the selection of a lightweight network for deployment on mobile devices. A reduction in parameters and FLOPs enhances model efficiency under identical computational resources, concurrently minimizing memory consumption and boosting computational speed.

3.3 Ablation experiment

Each module within FCHF-DETR was evaluated through ablation experiments to discern which modules enhance detection performance and which reduce computational and parameter costs. RT-DETR-R18 served as the benchmark model, with the introduction of the lightweight network structure, FasterNet, as FCHF-DETR's backbone to assess its capacity to reduce model parameters and enhance inference speed effectively. Subsequently, the AIFI module in the Efficient Hybrid Encoder was replaced with Cascaded Group Attention to extract finer features. Additionally, the CCFM module was substituted with HSFPN, capable of capturing and expressing multi-scale features, thereby enhancing network accuracy. Ultimately, the model's original loss function was optimized to the Focaler-CIoU loss function, adept at efficiently capturing edge information of tomato leaf diseases.

Initially, we evaluated the impact of integrating lightweight backbone networks versus not integrating them on the test set. Comparison of the benchmark model RT-DETR-R18 with RT-DETR FasterNet (Experiments 1 and 2) was performed. The introduction of lightweight backbone networks led to decreases of 1.9% and 0.5% in Precision and Recall, respectively. The mAP50-95 and mAP50 values decreased by 0.6% and 0.3%, respectively, while the number of Parameters decreased by 21%, the FPS increased by 1.8, and the FLOP decreased by 13.6%. These results suggest that FasterNet, as the backbone network of RT-DETR-R18, effectively reduces computational complexity and parameter count, and significantly enhances inference speed. Although the accuracy has marginally decreased, the improvement in efficiency renders this loss acceptable.

A lightweight network structure significantly trims model size and elevates detection speed, albeit at the expense of detection accuracy. Consequently, methods that enhance accuracy without incurring substantial computational costs are crucial.

Subsequently, employing the lightweight RT-DETR model with FasterNet as the backbone, we examined the performance alterations resulting from the integration of various modules. Experiments 3, 4, and 5 involved the replacement of the AIFI module in the original Efficient Hybrid Encoder with the SimAM, SE, and CGA attention mechanisms, respectively, each contributing to an improvement in accuracy. However, given the focus on lightweight networks in this study, the CGA attention mechanism was selected for further investigation. In Experiments 6 and 7, the CCFM module in the Efficient Hybrid Encoder was replaced by HSFPN without the SFF module and HSFPN with the SFF module, respectively. Upon comparison, the HSFPN with the SFF module, which offered greater accuracy improvements, was chosen. Building on Experiment 7, the loss function of the benchmark RT-DETR-R18 model was optimized, with both CIoU and Focaler-CIoU loss functions being employed for training. Table 2 illustrates the enhancement in detection performance attributable to the lightweight DETR model.

- Experiments 3, 4, and 5 evaluated the integration of SimAM, SE, and CGA attention mechanisms, respectively, into the RT-DETR-R18 model with FasterNet as the backbone network. Compared to Experiment 2, the additions of SimAM, SE, and CGA resulted in increases of 0.9%, 1.8%, and 2.3% in the mAP50-95 index, respectively, and changes of -0.1%, 0.4%, and 0.7% in the

TABLE 2 Ablation experiment results: comparative analysis of all modules used in FCHF-DETR.

	Model	P	R	mAP50-95	mAP50	Parameters	FPS	GFLOPs
1	RTDETR-R18	94.7	93.6	83.1	96.2	19,880,748	21.9	57.0
2	RTDETR-FasterNet	92.8	93.1	82.5	95.9	15,792,928	23.7	49.5
3	RTDETR-FasterNet-SimAM	94.1	93.7	83.4	95.8	15,621,884	24.8	47.3
4	RTDETR-FasterNet-SE	94.9	95.2	84.3	96.3	16,882,972	21.7	54.5
5	RTDETR-FasterNet-CGA	95.1	95.1	84.8	96.6	15,812,212	24.5	48.3
6	RTDETR-FasterNet-CGA-HSFPN	95.8	95.2	85.8	96.7	16,101,128	24.2	47.8
7	RTDETR-FasterNet-CGA-HSFPN_SFF	96.1	96.4	87.4	96.9	16,314,816	24.1	47.9
8	RTDETR-FasterNet-CGA-HSFPN_SFF-CIoU	95.8	96.1	87.3	97.0	16,307,482	24.1	47.8
9	RTDETR-FasterNet-CGA-HSFPN_SFF-Focaler-CIoU	96.4	96.7	89.1	97.2	16,265,580	24.1	47.8

mAP50 index for SimAM, SE, and CGA, respectively. The performance metrics suggest that SimAM, likely a non-parametric attention mechanism, notably improved the model's size and inference speed. However, given that SimAM only slightly improved, or even reduced, accuracy, despite a comprehensive comparison, the CGA attention mechanism was ultimately selected due to its significant accuracy improvements, despite a slight increase in model parameters. Additionally, substituting the AIFI module with the selected attention mechanism enhanced the accuracy of tomato leaf disease detection, albeit with a minor reduction in inference speed and a slight increase in model parameters, aligning with the initial objective of replacing the AIFI module.

- Experiments 6 and 7 demonstrate that replacing the CCFM module in the RT-DETR model with HSFPN and HSFPN_SFF leads to significant improvements in the detection accuracy of the model. In the test set, HSFPN and HSFPN_SFF increased the parameter count by 0.3M and 0.5M, respectively, and reduced inference speed by 0.3 and 0.4, respectively. In Experiment 6, incorporating the HSFPN module yielded a 7% increase in Precision, a 1% increase in mAP50-95, and a 0.5G reduction in FLOPs. However, considering the increase in model parameters and the decrease in inference speed, the improvement in detection accuracy is deemed insufficient. In Experiment 7, the integration of the SFF module into feature fusion resulted in increases of 1% in P, 1.3% in Recall, 2.6% in mAP50-95, and 0.3% in mAP50. Although the model parameters have increased slightly and the inference speed is slower compared to FasterNet+CGA in Experiment 5, the significant improvement in detection accuracy relative to the benchmark network satisfies the lightweight standard.
- In Experiments 8 and 9, the loss functions of the benchmark network were substituted with CIoU and Focaler CIoU, respectively. Although the impact on inference speed, parameter count, and computational complexity is minimal, the CIoU loss function fails to yield a significant improvement in detection accuracy. However, optimization of the Focaler CIoU loss function led to increases of 0.3% in Precision and Recall, and 1.7% and 0.3% in mAP50-95 and mAP50, respectively. The uneven distribution of tomato leaf disease and the presence of small or edge targets in the images pose challenges to the detection capabilities of the model, which is expected. The introduction of the Focaler CIoU loss function significantly enhances the localization and detection of challenging targets, thereby enhancing the accuracy and robustness of the model for small, overlapping, and edge targets.

In conclusion, compared to RT-DETR-R18, the proposed FCHF-DETR demonstrates a 1.7% increase in Precision, a 3.1% increase in Recall, a 6% increase in mAP50-95, and a 1% increase in mAP50. The number of parameters decreased by 3.6M, FPS increased by 2.2, FLOP decreased by 9.2G, thereby significantly improving the speed and accuracy of tomato leaf disease detection. Therefore, FCHF-DETR is highly suitable for deployment on terminal devices in agricultural environments, such as cameras, offering the high detection performance necessary for real-world applications.

3.4 Visual display

Across a test set comprising 3147 images, FCHF-DETR precisely identified eight types of tomato leaf diseases, alongside healthy leaves, attaining an overall mAP50-95 of 89.1% and an mAP50 of 97.2%.

To illustrate the detection performance benefits of the proposed method, a visual representation of the detection results for tomato leaf diseases under various conditions is provided. Figure 10 depicts the model's detection capability in straightforward settings, characterized by favorable shooting conditions, a simple background, clearly visible affected areas on the tomato leaves, and a minimal number of leaves in the image. Figures 10A–H

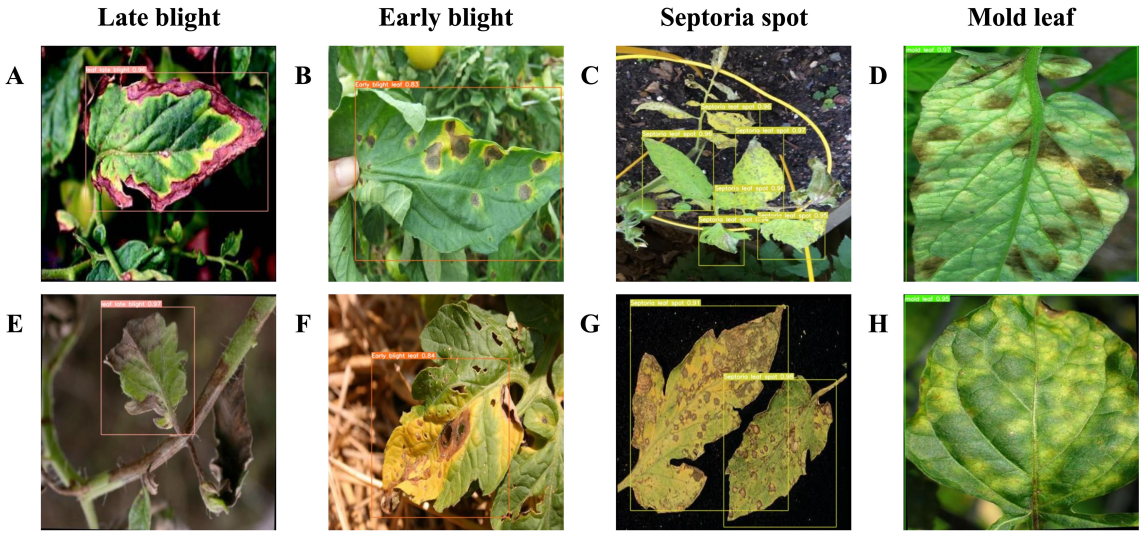


FIGURE 10
(A–H) demonstrate the detection of four distinct types of plant leaf diseases under controlled conditions. The bounding box within the figure highlights the location and specific types of tomato leaf diseases.

demonstrate the model's ability to concurrently and accurately detect four distinct tomato leaf diseases in uncomplicated environments: late blight, early blight, Septoria spot, and mold leaf. Given that yellow viruses typically cluster and are found in complex settings, their detection results were not showcased in the depiction of simple environments.

The integration of the CGA attention mechanism and HSFPN feature fusion module endows the model with a robust capability to extract pivotal information from images, ensuring high detection accuracy across various tomato leaf diseases. Figure 11 illustrates the model's detection performance in complex scenarios, including situations where leaves are at the image's edge or partially obscured. Figures 11A–D reveal that the FCHF-DETR model precisely identifies occluded diseased leaves. Figures 11I–L demonstrate that, with the Focaler-CIoU loss function integrated, the model enhances the detection accuracy of challenging edge targets, mitigating the original model's limitation in identifying partially visible diseased leaves. In the other images, the enhanced model is shown to effectively identify edge targets, even those obscured by surrounding foliage.

To underscore the strengths of the proposed model in complex scenarios, Figure 12 illustrates its detection capabilities in densely populated environments. Given the dense distribution and potential for small spots on tomato leaves in real-world settings, detecting diseased leaves in such environments is paramount. Despite these challenges, the model maintains robust performance. Figure 12 demonstrates the model's efficacy in identifying diseased tomato leaf areas within dense foliage, under varied conditions such as intense illumination area A, D, shadow area B, E, or high-dense area C, F.

Acknowledging weather-related challenges at tomato cultivation sites, pixel reduction was applied to part of the test set data to simulate the effects of rain or dense fog on camera imagery. Figure 13 reveals that, even with reduced pixel quality, the FCHF-DETR model reliably detects most tomato leaf diseases, with only a minor impact on detection accuracy. The sustained performance in simulated rainy and foggy conditions is credited to the Cascaded Group Attention and HSFPN feature fusion mechanisms within the Efficient Hybrid Encoder, capable of extracting key features from blurred images. Additionally, the incorporation of the Focaler-CIoU loss function enables the detection of leaf diseases that pose challenges for the RT-DETR-R18 model, significantly aiding practical deployment.

The visual evidence from Figures 10–13 confirms that FCHF-DETR adeptly addresses a range of challenges typical in real agricultural settings for tomato leaf disease detection, effectively resolving longstanding issues in the sector.

4 Discussion

In contemporary agricultural practices, numerous tomato plants are afflicted by leaf diseases, making manual detection excessively time-consuming and labor-intensive. Current technologies frequently fail to balance processing speed with detection accuracy, particularly when identifying small disease spots, presenting clear drawbacks. To address this challenge, this study introduced FCHF-DETR, a high-precision, lightweight detection algorithm derived from the RT-DETR-R18 framework. A dataset comprising 3147 images of tomato leaf diseases was compiled, encompassing diverse scenes and levels of image clarity. To streamline the model and enhance memory efficiency, the traditional ResNet18 was substituted with FasterNet in the backbone network. Concurrently, within efficient hybrid encoders, replacing the AIFI module with a cascaded group attention mechanism and the CCFM module with HSFPN notably boosted detection accuracy with minimal impact on speed.

Furthermore, to better identify challenging samples, the Focaler-CIoU loss function was introduced, enhancing the model's

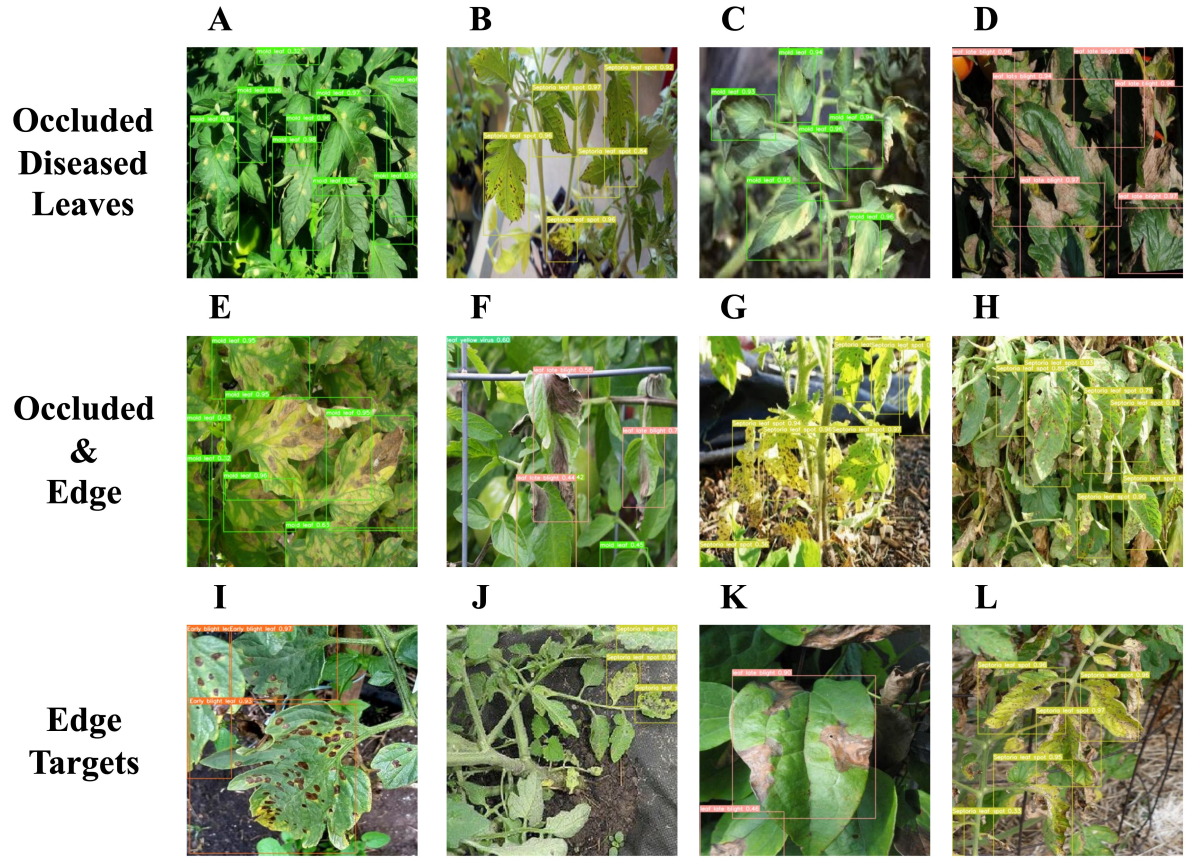


FIGURE 11
(A–D) illustrate the detection performance of the targeted leaf disease in scenarios where it is obscured by other leaves. **(E–H)** demonstrate the detection performance of the targeted leaf disease when situated at the periphery of the image and simultaneously obscured by other foliage. **(I–L)** reveal the detection performance of the targeted leaf disease at the image's edge.

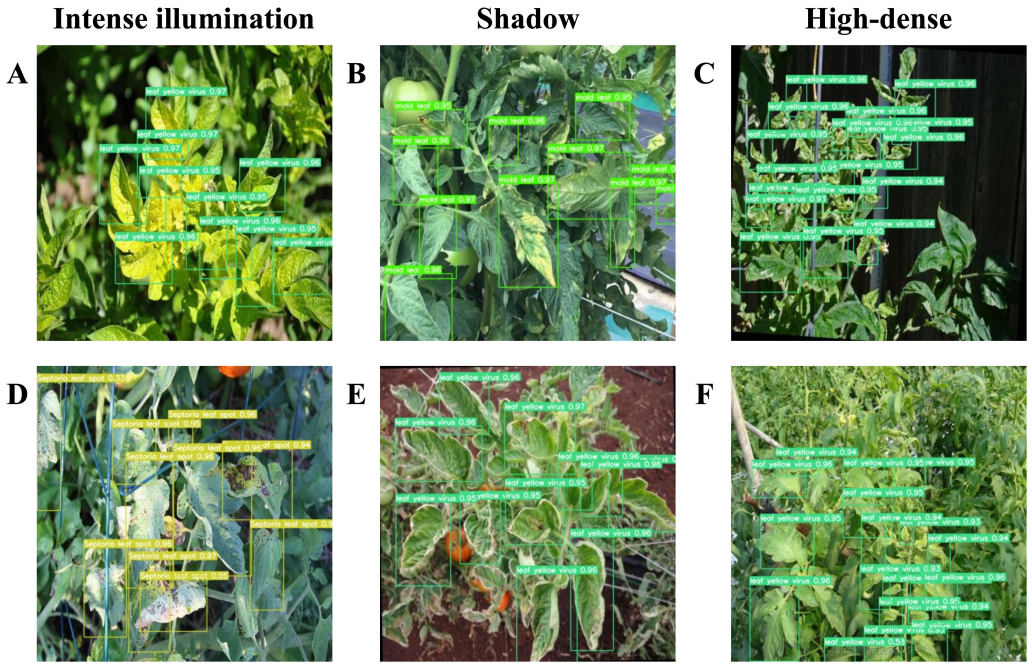


FIGURE 12
(A, D) present the detection results of leaf disease under conditions of intense illumination. **(B, E)** depict the detection results of leaf disease within shaded environments. **(C, F)** illustrate the detection effectiveness of leaf disease in highly dense settings.

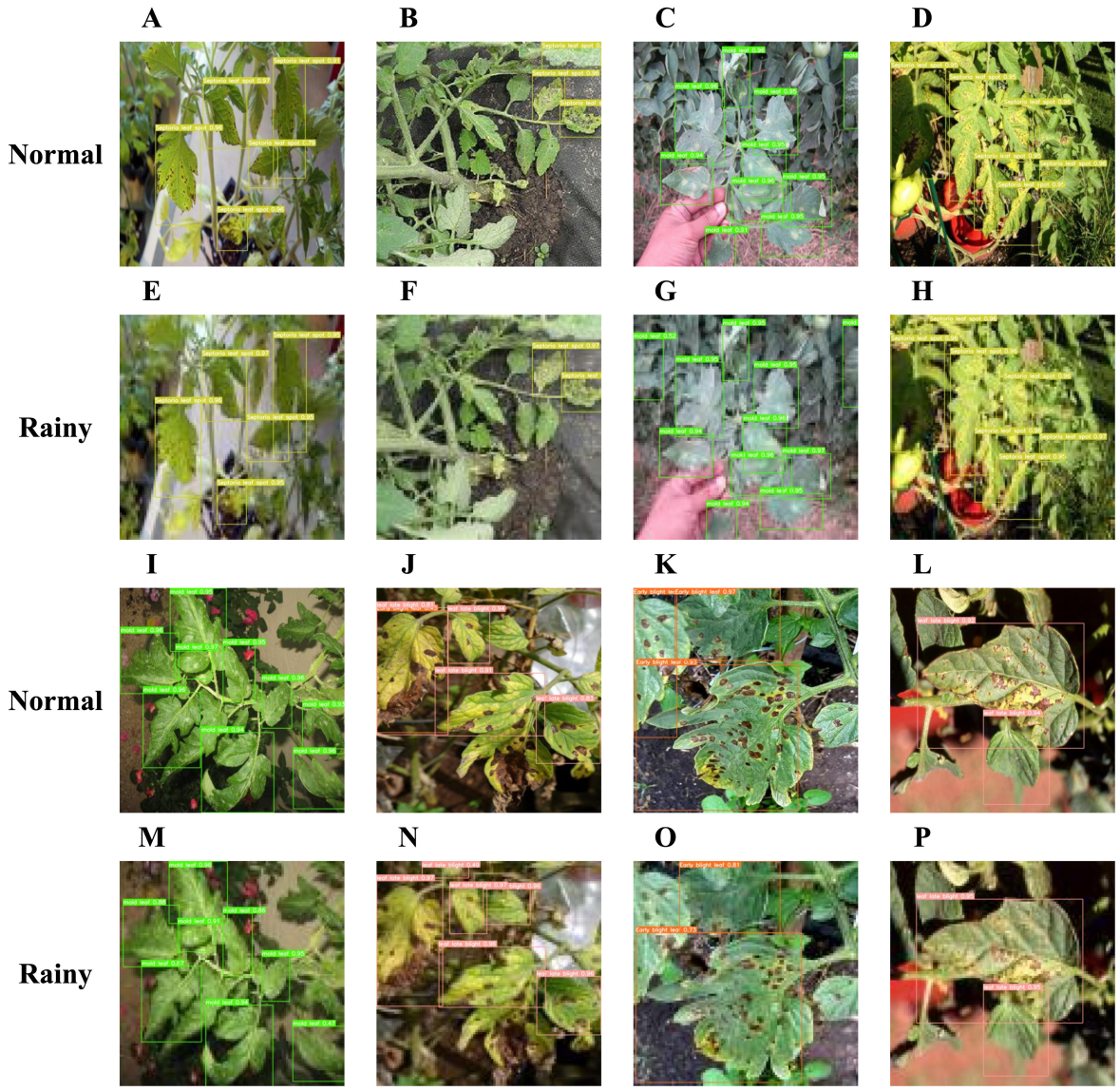


FIGURE 13
(A–D) and **(I–L)** demonstrate the detection efficacy of tomato leaf disease in standard conditions, while **(E–H)** and **(M–P)** exhibit the comparative detection efficacy of the model on the test set following pixel reduction processing and the simulation of rainy conditions within an authentic plantation setting.

performance across the dataset. Experimental results indicated that FCHF-DETR achieved an mAP50-95 of 89.1% on the test set, marking a 6% improvement, and an mAP50 of 97.2%, a 1% increase. Concurrently, FLOPs decreased by 9.2G, and the model's parameter count was reduced by 3.6M. These achievements showcase the method's enhancement of detection accuracy and successful reduction in the model's computational load, illustrating an effective balance between accuracy and efficiency.

In practical agricultural settings, particularly on diverse farmlands, a common challenge arises: the overlapping or obstruction of leaves from different crops, markedly impacting tomato leaf disease detection. For instance, in fields where tomatoes coexist with taller crops like corn or legumes, the foliage of these crops can obscure tomato leaves, masking critical disease features. Under these conditions, the effectiveness of even high-precision detection algorithms like FCHF-DETR may be markedly limited. Leaf occlusion not only diminishes the available feature information for algorithmic recognition but can also lead to errors, like mistaking occluded edges or shadows for disease spots.

This issue underscores the limitations of current visual-based object detection algorithms in navigating complex agricultural scenes. Addressing this challenge necessitates a deeper comprehension of crop interactions and growth patterns to develop algorithms capable of adapting to such diversity and complexity. Furthermore, employing multiperspective or multimodal data acquisition techniques, like integrating aerial and lateral imagery or additional sensor data, could mitigate these issues and enhance lesion detection in occluded conditions.

Meanwhile, we also investigated that the manifestation of tomato leaf disease may vary in different natural environments due to various factors such as climate, soil type, and humidity, resulting in certain types of leaf diseases being more common in specific environments. For example, in high humidity and warm environments, the incidence of downy mildew may be much higher than that of early or late blight in arid environments. The impact of these environmental factors on disease occurrence requires the detection system to adjust the weight of various leaf disease detection according to different natural conditions, in order to improve the detection accuracy and efficiency in specific environments. However, even the high-precision and high-efficiency detection algorithm FCHF-DETR invented in this article adopts the same detection strategy for all types of leaf diseases, failing to fully consider the diversity of natural environmental factors. This may lead to insufficient sensitivity of algorithms to detecting high-risk diseases in certain specific environments, thereby reducing overall detection efficiency and accuracy.

In order to solve this problem, future detection algorithms need to introduce environmental awareness mechanisms, analyze and learn the occurrence patterns of diseases under different natural environmental conditions, and dynamically adjust the detection weights for different leaf diseases. This may involve complex data collection and analysis, such as combining meteorological data, soil conditions, and crop growth data, using machine learning algorithms to predict the probability of disease occurrence under different environmental conditions, and optimizing the parameters of the detection model accordingly. Through this approach, the detection system can adapt more intelligently to different natural environments, improve the detection accuracy of key diseases, and provide more reliable technical support for agricultural production.

5 Conclusion

This study introduces FCHF-DETR, a lightweight model for detecting tomato leaf diseases, effectively balancing accuracy and speed. It employs data augmentation and reduction techniques to adapt to real-world environments for detecting tomato leaf diseases. FCHF-DETR enhances the RT-DETR-R18 framework by integrating the lightweight FasterNet backbone, boosting detection speed and reducing model parameters without compromising accuracy. Additionally, it introduces the Cascaded Group Attention mechanism, replacing the AIFI module, and substitutes the CCFM module with HSFPN in the original network. Despite a minor increase in computational speed and model parameters, there's a significant enhancement in detection accuracy. The adoption of the Focaler-CIOU loss function, replacing the original, further refines the accuracy for challenging samples without altering parameters or computational complexity. Experimental results reveal that FCHF-DETR surpasses RT-DETR-R18 with a 1.7% increase in precision, 3.1% in recall, 1% in mAP50 and 6% in mAP50-95, and reductions in parameters, FPS, and FLOPs. This signifies not just a notable boost in accuracy but also a substantial decrease in the model's parameter count, thus offering robust support for contemporary tomato leaf disease detection.

Future research will aim to refine detection accuracy in diverse farmlands affected by overlapping leaf occlusion. We plan to leverage multiperspective or multimodal data to develop more adaptive detection algorithms. Additionally, to accommodate varying tomato leaf disease patterns across different environments, future algorithms will incorporate environmental awareness mechanisms. Dynamic adjustments to the detection priorities of different diseases will enhance the accuracy and efficiency in specific environments, broadening the algorithm's applicability in complex real-world scenarios.

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

DX: Writing – original draft, Writing – review & editing. TL: Data curation, Resources, Validation, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

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

References

Agarwal, M., Singh, A., Arjaria, S., Sinha, A., and Gupta, S. (2020). ToLeD: Tomato leaf disease detection using convolution neural network. *Proc. Comput. Sci.* 167, 293–301. doi: 10.1016/j.procs.2020.03.225

Azim, T., Jaffar, M. A., and Mirza, A. M. (2014). Fully automated real time fatigue detection of drivers through fuzzy expert systems. *Appl. Soft Computing* 18, 25–38. doi: 10.1016/j.asoc.2014.01.020

Bochkovskiy, A., Wang, C. Y., and Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint*. doi: 10.48550/arXiv.2004.10934

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). "End-to-end object detection with transformers," in *European conference on computer vision* (Springer), 213–229. doi: 10.1007/978-3-030-58452-8_13

Chen, J., Kao, S.-h., He, H., Zhuo, W., Wen, S., Lee, C.-H., et al. (2023). "Run, Don't walk: Chasing higher FLOPS for faster neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (Vancouver, Canada: IEEE), 12021–12031. doi: 10.1109/cvpr52729.2023.01157

Chen, Y., Zhang, C., Chen, B., Huang, Y., Sun, Y., Wang, C., et al. (2024). Accurate leukocyte detection based on deformable-DETR and multi-level feature fusion for aiding diagnosis of blood diseases. *Comput. Biol. Med.* 170, 107917. doi: 10.1016/j.compbiomed.2024.107917

Chollet, F. (2017). "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. (Honolulu, Hawaii, USA: IEEE), 1251–1258. doi: 10.48550/arXiv.1610.02357

Coelho, M. C., Rodrigues, A. S., Teixeira, J. A., and Pintado, M. E. (2023). Integral valorisation of tomato by-products towards bioactive compounds recovery: Human health benefits. *Food Chem.* 410, 135319. doi: 10.1016/j.foodchem.2022.135319

Dong, C., Loy, C. C., He, K., and Tang, X. (2015). Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 38, 295–307. doi: 10.1109/TPAMI.2015.2439281

Gao, S., Cheng, M., Zhao, K., Zhang, X., Yang, M., 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

Geisseler, D., and Horwath, W. R. (2014). Production of processing tomatoes in California. *Assess. Plant fertility fertilizer requirements Agric. Crops California* 1.

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: IEEE), 2961–2969. doi: 10.48550/arXiv.1703.06870

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, USA: IEEE), 770–778. doi: 10.1109/cvpr.2016.90

Hernandez, D. J., David, A. S., Menges, E. S., Searcy, C. A., and Afkhami, M. E. (2021). Environmental stress destabilizes microbial networks. *ISME J.* 15, 1722–1734. doi: 10.1038/s41396-020-00882-x

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., et al. (2019). "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*. (Seoul, South Korea: IEEE), 1314–1324. doi: 10.1109/iccv.2019.00140

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, Utah, USA: IEEE), 7132–7141. doi: 10.48550/arXiv.1709.01507

Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., Kwon, Y., Michael, K., et al. (2022). ultralytics/yolov5: v6. 2-yolov5 classification models, apple m1, reproducibility, clearml and deci. ai integrations. *Zenodo*. doi: 10.5281/zenodo.7002879

Lambooij, M., IJsselsteijn, W., Fortuin, M., and Heynderickx, I. (2009). Visual discomfort and visual fatigue of stereoscopic displays: A review. *J. Imaging Sci. Technol.* 53, 30201–30201. doi: 10.2352/J.ImagingSci.Technol.2009.53.3.030201

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324. doi: 10.1109/5.726791

Lee, M. H. (2022). Forecasting leaf mold and gray leaf spot incidence in tomato and fungicide spray scheduling. *J. Bio-Environment Control* 31, 376–383. doi: 10.12791/KSBEC.2022.31.4.376

Lin, M., Chen, Q., and Yan, S. (2013). Network in network. *arXiv preprint*. doi: 10.48550/arXiv.1312.4400

Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018). "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. (Salt Lake City, Utah, USA: IEEE), 8759–8768. doi: 10.48550/arXiv.1803.01534

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. 21–37, Proceedings, Part I 14: (Amsterdam, The Netherlands: Springer). doi: 10.1007/978-3-319-46448-0_2

Lu, Y., Wang, S., Yao, G., and Xu, J. (2023). Green total factor efficiency in vegetable production: A comprehensive ecological analysis of China's practices. *Agriculture* 13, 2021. doi: 10.3390/agriculture13102021

Lv, W., Xu, S., Zhao, Y., Wang, G., Wei, J., Cui, C., et al. (2024). "Detrs beat yolos on real-time object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 16965–16974. doi: 10.48550/arXiv.2304.08069

Min, X. (2023). *Bio-economic modelling of high-tech greenhouse production systems in China* (Holland Wageningen: Wageningen University).

Pallathadka, H., Ravipati, P., Sajja, G. S., Phasinam, K., Kassanuk, T., Sanchez, D. T., et al. (2022). Application of machine learning techniques in rice leaf disease detection. *Materials Today: Proc.* 51, 2277–2280. doi: 10.1016/j.matpr.2021.11.398

Rahman, M. A., and Wang, Y. (2016). "Optimizing intersection-over-union in deep neural networks for image segmentation," in *International symposium on visual computing*. (Las Vegas, Nevada, USA: Springer), 234–244. doi: 10.1007/978-3-319-50835-1_22

Redmon, J., and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*. doi: 10.48550/arXiv.1804.02767

Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 1137–1149. doi: 10.1109/tpami.2016.2577031

Saleem, M. H., Potgieter, J., and Arif, K. M. (2022). Weed detection by faster RCNN model: An enhanced anchor box approach. *Agronomy* 12, 1580. doi: 10.3390/agronomy12071580

Simard, P. Y., Steinkraus, D., and Platt, J. C. (2003). *Best practices for convolutional neural networks applied to visual document analysis* (Edinburgh: Icdar).

Sujatha, R., Chatterjee, J. M., Jhanjhi, N., and Brohi, S. N. (2021). Performance of deep learning vs machine learning in plant leaf disease detection. *Microprocessors Microsystems* 80, 103615. doi: 10.1016/j.micpro.2020.103615

Teng, Y., Zhang, J., and Liu, L. (2022). MSR-RCNN: a multi-class crop pest detection network based on a multi-scale super-resolution feature enhancement module. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.810546

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, Canada: IEEE), 7464–7475. doi: 10.48550/arXiv.2207.02696

Wang, A., Peng, T., Cao, H., Xu, Y., Wei, X., and Cui, B. (2022). TIA-YOLOv5: An improved YOLOv5 network for real-time detection of crop and weed in the field. *Front. Plant Sci.* 13. doi: 10.3389/fpls.2022.1091655

Wang, S., Sun, G., Zheng, B., and Du, Y. (2021). A crop image segmentation and extraction algorithm based on mask RCNN. *Entropy* 23, 1160. doi: 10.3390/e23091160

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*. (IEEE), 11534–11542. doi: 10.1109/cvpr42600.2020.01155

Wang, C. Y., Yeh, I. H., and Liao, H. Y. M. (2024). YOLOv9: learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*. doi: 10.48550/arXiv.2402.13616

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), 3–19. doi: 10.48550/arXiv.1807.06521

Yang, H., Deng, X., Shen, H., Lei, Q., Zhang, S., and Liu, N. (2023). Disease detection and identification of rice leaf based on improved detection transformer. *Agriculture* 13, 1361. doi: 10.3390/agriculture13071361

Yang, L., Zhang, R.-Y., Li, L., and Xie, X. (2021). "Simam: A simple, parameter-free attention module for convolutional neural networks," in *International conference on machine learning: PMLR* (PMLR), 11863–11874.

Zhang, J., Wang, J., and Zhao, M. (2023). A lightweight crop pest detection algorithm based on improved yolov5s. *Agronomy* 13, 1779. doi: 10.3390/agronomy13071779

Zhang, Y., Yang, P., Xiao, J., Bai, Y., Che, H., and Wang, X. (2022). "K-converter: An unsupervised singing voice conversion system," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. (Singapore: IEEE), 6662–6666. doi: 10.1109/ICASSP43922.2022.9746562

Zhang, H., and Zhang, S. (2024). Focaler-ioU: more focused intersection over union loss. *arXiv preprint arXiv:2401.10525*. doi: 10.48550/arXiv.2401.10525

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 26 September 2024
DOI 10.3389/fpls.2024.1452821

OPEN ACCESS

EDITED BY
Mohsen Yoosefzadeh Najafabadi,
University of Guelph, Canada

REVIEWED BY
Parvathaneni Naga Srinivasu,
Amrita Vishwa Vidyapeetham University, India
Seyed-Hassan Miraei Ashtiani,
Dalhousie University, Canada

*CORRESPONDENCE
Philippe Lyonel Touko Mbouembe
lyoneltouko@gmail.com

RECEIVED 21 June 2024
ACCEPTED 26 August 2024
PUBLISHED 26 September 2024

CITATION
Liu G, Zhang Y, Liu J, Liu D, Chen C, Li Y, Zhang X and Touko Mbouembe PL (2024) An improved YOLOv7 model based on Swin Transformer and Trident Pyramid Networks for accurate tomato detection.
Front. Plant Sci. 15:1452821.
doi: 10.3389/fpls.2024.1452821

An improved YOLOv7 model based on Swin Transformer and Trident Pyramid Networks for accurate tomato detection

Guoxu Liu[1], Yonghui Zhang[1], Jun Liu[2], Deyong Liu[3], Chunlei Chen[1], Yujie Li[1], Xiujie Zhang[1] and Philippe Lyonel Touko Mbouembe[4]*

[1]School of Computer Engineering, Weifang University, Weifang, China, [2]Shandong Provincial University Laboratory for Protected Horticulture, Weifang University of Science and Technology, Weifang, China, [3]School of Computer Science, Weifang University of Science and Technology, Weifang, China, [4]Department of Electronics Engineering, Pusan National University, Busan, Republic of Korea

Accurate fruit detection is crucial for automated fruit picking. However, real-world scenarios, influenced by complex environmental factors such as illumination variations, occlusion, and overlap, pose significant challenges to accurate fruit detection. These challenges subsequently impact the commercialization of fruit harvesting robots. A tomato detection model named YOLO-SwinTF, based on YOLOv7, is proposed to address these challenges. Integrating Swin Transformer (ST) blocks into the backbone network enables the model to capture global information by modeling long-range visual dependencies. Trident Pyramid Networks (TPN) are introduced to overcome the limitations of PANet's focus on communication-based processing. TPN incorporates multiple self-processing (SP) modules within existing top-down and bottom-up architectures, allowing feature maps to generate new findings for communication. In addition, Focaler-IoU is introduced to reconstruct the original intersection-over-union (IoU) loss to allow the loss function to adjust its focus based on the distribution of difficult and easy samples. The proposed model is evaluated on a tomato dataset, and the experimental results demonstrated that the proposed model's detection recall, precision, F_1 score, and AP reach 96.27%, 96.17%, 96.22%, and 98.67%, respectively. These represent improvements of 1.64%, 0.92%, 1.28%, and 0.88% compared to the original YOLOv7 model. When compared to other state-of-the-art detection methods, this approach achieves superior performance in terms of accuracy while maintaining comparable detection speed. In addition, the proposed model exhibits strong robustness under various lighting and occlusion conditions, demonstrating its significant potential in tomato detection.

KEYWORDS

tomato detection, YOLOv7, Swin Transformer, Trident Pyramid Network, Focaler-IoU

1 Introduction

Fruit harvesting is a critical step in the agricultural production process. However, traditional manual methods are costly, time-consuming, and inefficient, complicating meeting large-scale cultivation demands. Due to the advancement of smart agriculture, the transition from manual labor to automated fruit harvesting has become an inevitable trend. For fruit harvesting robots, accurate fruit identification and localization are essential for efficient harvesting. Therefore, it is very important to develop robust and accurate fruit detection algorithms for the robotic vision systems.

Over the past few decades, numerous researchers have explored various fruit detection methods. These approaches are generally categorized into threshold discrimination and machine learning-based methods. Initially, the fruit targets in images are segmented by setting thresholds based on simple features such as color (Wei et al., 2014), shape (Kelman and Linker, 2014), texture (Rakun et al., 2011), or a combination of these features (Payne et al., 2014), to complete the detection process. Although these methods yield reasonable results, the sensitivity of the thresholds to environmental variations limits their generalization capabilities. The introduction of machine learning has mitigated these limitations. Traditional techniques, which integrate handcrafted features such as Histogram of Oriented Gradients and Haar features with machine learning models like Support Vector Machine (SVM) (Liu et al., 2019) and AdaBoost (Zhao et al., 2016), have been employed to locate and recognize fruits. Following the success of deep learning in computer vision (Krizhevsky et al., 2012), it has been applied to smart agriculture (Sa et al., 2016; Fuentes et al., 2017). Deep learning models are adept at directly extracting features from data and facilitating end-to-end training, significantly enhancing the models' detection performance and efficiency.

Despite the significant advancements in deep learning-based fruit detection methods, several shortcomings persist. These models are typically trained on data from controlled conditions, resulting in reduced robustness against unconstrained factors in real-world environments, such as illumination variations and occlusion or overlap occurrences. In addition, the traditional IoU-based regression loss function utilized in the YOLO model cannot accurately predict the position of fruit targets. Due to the limitations inherent in traditional regression methods, which neglect the distribution of objects across different scales, they can fail to accurately identify the location of fruit targets, particularly in challenging scenarios.

In order to address these challenges, this study introduces a novel YOLO-SwinTF model, designed to enhance the accuracy of tomato detection in complex environments while maintaining high detection efficiency. Based on the YOLOv7 architecture, the model's backbone, neck, and loss function are refined to improve feature extraction and target-focusing capabilities. Specifically, Swin Transformer blocks are incorporated into the backbone to aid the model in capturing long-range visual dependencies while maintaining computational efficiency, thereby enhancing the semantic information of the features. Then, the original PANet is replaced with the TPN architecture by embedding multiple SP modules between the traditional top-down and bottom-up architectures. This modification allows the feature mapping to generate new information for propagation. In addition, a Focaler-IoU loss is constructed using a linear interval mapping method to adjust its focus based on sample difficulty, improving the model's detection performance.

The main contributions to this study are as follows:

1. A novel network architecture, YOLO-SwinTF, is proposed, which incorporates the Swin Transformer attention mechanism and Trident Pyramid Network architectures to enhance feature extraction capabilities.
2. The Focaler-IoU loss is introduced to accurately identify tomato locations. This method enhances the detection performance of the model by dynamically adjusting the focus of the loss among samples of varying difficulty.
3. Extensive experiments on tomato datasets demonstrate that the proposed YOLO-SwinTF model achieves excellent performance compared to the current state-of-the-art methods for tomato detection.

The remainder of this paper is organized as follows: Section 2 reviews the literature on fruit detection methods, which include threshold-based discriminant analysis, machine learning, and deep learning approaches. Section 3 introduces the proposed tomato detection model. The experimental results obtained through the proposed method are presented and discussed in Section 4. Finally, Section 5 concludes the study.

2 Related work

2.1 Threshold-based discriminant methods

In the early days, researchers employed simple features such as color, shape, and texture to detect fruits. Kurtulmus et al. (2011) developed a method for detecting and counting green citrus fruits in natural environments using color images. They introduced a novel "eigenfruit" approach that incorporated color, circularity, and Gabor texture analysis to identify the fruits. Then, a shifting sub-window technique was applied at three different scales to scan the image and localize the fruits. In their study, 73% of green fruits were correctly identified. Ji et al. (2012) established an automatic vision recognition system to guide apple harvesting robots. Images of the apples were captured using a color charge-coupled device camera. An industrial computer processed and recognized the apples. A vector median filter removed noise from the color images of the apples, and an image segmentation algorithm based on region and color features was applied. The study reported an accuracy of 89% with an average detection time of 352 ms. Chaivivatrakul and Dailey (2014) developed a texture-based fruit detection approach. This method utilizes interest-point feature extraction and descriptor computation. A low-cost web camera suitable for mechanized systems evaluated 24 combinations of interest-point features and descriptors for pineapples and bitter melons. The method achieved an accuracy of

85% for the single-image detection of pineapples and 100% for bitter melons. Jana and Parekh (2017) proposed a shape-based fruit recognition approach, which included a pre-processing step that normalizes fruit images to account for translation, rotation, and scaling differences. This method then employed features unaffected by variations in distance, growth phase, and surface appearance of the fruits for detection. The proposed method was applied to a dataset of 210 images covering seven different fruit classes, achieving an overall recognition accuracy between 88% and 95%.

Although threshold-based discriminant methods have demonstrated reasonable effectiveness in detecting fruits, their performance significantly depends on the appropriateness of the selected thresholds. This dependence can result in limited model generalization and diminish detection robustness.

2.2 Traditional machine learning-based methods

Due to the development of machine learning, many researchers have attempted to apply it to fruit detection. Methodologies include Adaboost (Payne et al., 2014), Random Forests (Samajpati and Degadwala, 2016), and SVM (Behera et al., 2020). Using machine vision and SVM, Peng et al. (2018) conducted a study on detecting different classes of fruit, such as apples, bananas, citruses, carambolas, pears, and pitaya. The process involved using a Gaussian filter and histogram equalization for image processing, followed by segmentation with the Otsu method. To extract features, researchers employed shape-invariant moments and synthesized the color and shape of fruits. An SVM was then used to classify and detect the fruits, achieving detection rates of 95% for apples, 80% for bananas, 97.5% for citrus fruits, 86.7% for carambola, 92.5% for pears, and 96.7% for pitaya. Jiao et al. (2020) proposed a detection and localization method for overlapping apples, which began with the transformation and segmentation of apple images using the Lab color space and K-means algorithm. Morphological processes such as erosion and dilation were applied to delineate the apple edges. In addition, a fast algorithm calculated the minimum distance from each interior point to the apple outline, determining the radii by identifying the shortest distance from the center to the edge. Zhu et al. (2021) developed a carrot detection method by extracting deep features from a three-layer fully connected layer of network models and integrating these with an SVM. Their most effective model combined ResNet101 with an SVM, achieving an accuracy of 98.17%. Yu et al. (2021) proposed a method for identifying ripe litchi using an RGB-D camera in natural environments. Their approach utilized both color and depth images for litchi detection. Initially, depth image segmentation was employed to eliminate redundant image information outside the effective range of the manipulator. A random forest binary classification model was then trained using color and texture features to detect litchi fruits, achieving detection accuracies of 89.92% for green litchis and 94.50% for red litchis.

Although machine learning has significantly advanced fruit detection, these methods predominantly rely on handcrafted features and possess inherent limitations. Their capacity to abstract features is restricted, confining them to simple scenarios and limiting their generalization capabilities. In addition, the models lack end-to-end learning, which diminishes learning efficiency.

2.3 Deep learning-based methods

In recent years, deep learning-based approaches have emerged as powerful alternatives. In particular, convolutional neural networks (CNN) have shown remarkable success in learning discriminative features directly from raw image data without needing handcrafted features. CNN-based architectures such as Faster R-CNN (Ren et al., 2015), YOLO (Redmon et al., 2016; Redmon and Farhadi, 2017, 2018; Bochkovskiy et al., 2020; Wang et al., 2023), and SSD (Liu et al., 2016) have been widely used for fruit detection. Bargoti and Underwood (2017) proposed a deep model for detecting fruits in orchards, based on Faster R-CNN (Ren et al., 2015), to detect mangoes, almonds, and apples. This method achieved an F_1 score of 90% for mangoes and apples. Ganesh et al. (2019) utilized Mask R-CNN (He et al., 2017) to detect individual fruits and obtain pixel-wise masks for each detected fruit in an image, achieving an overall F_1 score of approximately 89%. Despite the advancements in two-stage methods that use separate networks to predict bounding boxes and class probabilities from an input image, these are not well suited for real-time applications. Recently, YOLO algorithms have been proposed to address this issue using a single CNN to predict and classify objects. Hernández et al. (2023) developed a tomato detection and classification method based on YOLOv3-tiny (Redmon and Farhadi, 2018), achieving an F_1 score of 90% for detecting ripe tomatoes. Guo et al. (2023) employed YOLOv7 for the real-time detection of ripe tomatoes, using an improved RepLKNet (Ding et al., 2022) to enhance the receptive field. In addition, the head structure of YOLOv7 was redesigned to address the issue of low FLOPS, and FasterNet (Chen et al., 2023) was used to optimize the structure between the Concat and CBS in the head. ODConv (Li et al., 2022) was added to improve the feature extraction for small tomatoes, achieving an mAP (0.5:0.95) of 56.8% with a detection time of 0.0127 s. Zeng et al. (2023) proposed a lightweight modified YOLOv5 for real-time localization and ripeness detection of tomatoes, achieving an mAP of 96.9% with a detection speed of 42.5 ms. Mbouembe et al. (2023) developed an efficient tomato detection method based on YOLOv4, incorporating an improved BottleneckCSP, a modified CSP-SPP, and CARAFE into the YOLOv4 architecture to enhance the feature expression capabilities of the model. This method achieved an mAP of 98.5%. Wang et al. (2024c) developed a grape detection algorithm based on YOLOv5s, introducing a dual-channel feature extraction attention mechanism (Li et al., 2017) and a dynamic snake convolution (Qi et al., 2023) in the backbone network to improve feature extraction. The mAP (0.5:0.95) was 69.3%. Gao et al. (2024) established an improved binocular calyx localization method based on YOLOv5x to detect kiwifruit, achieving an mAP of 93.5% with a detection speed of 105 ms per image.

Despite advances in deep learning-based fruit detection, several challenges remain. Variability in fruit appearance due to uneven

illumination, overlap, and occlusion poses a challenge for accurate detection. In addition, the presence of similar-looking objects and background clutter further complicates this task.

3 Materials and methods

3.1 Image acquisition

The tomato dataset for this study was collected at the Shouguang Vegetable High-Tech Demonstration Park in Shandong Province, China between 2017 and 2019. The acquisition equipment utilized was a Sony digital camera (Sony DSC-W170, Tokio, Japan) with a resolution of 3648 × 2056 pixels. This study collected 966 tomato images under various environmental conditions, including sunlight, shade, overlap, and occlusion. Considering that the dataset is not large, additional splitting could lead to a smaller training set, which is prone to overfitting (Ashtiani et al., 2021). Therefore, we divided the data into training and test sets at a ratio of 3:1, following (Liu et al., 2022; Jia et al., 2023). The training dataset comprised 725 images featuring 2553 tomatoes, whereas the test set included 241 images with 912 tomatoes. Figure 1 displays a selection of example images captured under various environmental conditions.

3.2 Image augmentation

The study applied data augmentation techniques to the collected images to enhance the generalization capability of the trained model and prevent overfitting. This resulted in a final set of 4350 enhanced images, achieved through horizontal flipping, scaling and cropping, brightness transformation, color balancing and blurring, as shown in Figure 2. For brightness transformation, a random factor ranging from 0.6 to 1.4 was employed to modulate pixel intensity, simulating the effects of diverse weather conditions on image brightness. Scaling and cropping were performed according to the methods described by Liu et al. (2020). During this phase, images without labels were discarded. The Gray World algorithm (Lam, 2005) was employed for color balancing to mitigate the impact of lighting on color rendering. Then, random blurring was applied to the augmented images to mimic the indistinct visuals that can result from camera motion. Table 1 lists the total number of resulting images after data augmentation.

3.3 YOLOv7 model

YOLOv7 (Wang et al., 2023) is an anchor-based detection method among the widely used YOLO algorithms. Like other iterations in the YOLO series, this version comprises three components: a backbone network for feature extraction; a neck network that fuses and refines the extracted features, yielding large, medium, and small feature sets; and a head network that utilizes these features from the neck to generate prediction outputs.

YOLOv7 developed a new backbone network called EfficientRep, which is a redesigned and improved version of the EfficientNet architecture (Tan and Le, 2019). This new backbone network includes different modules: E-ELAN, MPConv, and SPPCSPC. The E-ELAN module is an extended version of the ELAN (Wang et al., 2022). The original ELAN was designed to address the problem of convergence in deep models, which can gradually deteriorate as the models scale. E-ELAN maintains the same gradient flow as ELAN, but increases cardinality through group convolution. The MPConv module strikes a balance between increasing representational capacity and maintaining computational efficiency. The SPPCSPC module is a combination of the SPP module (He et al., 2015) and the CSP module (Wang et al., 2020). The SPP module captures features at different spatial resolutions, which is beneficial for detecting objects of various sizes. The CSP module then facilitates the flow of information between different stages and concatenates the output of the SPP module with

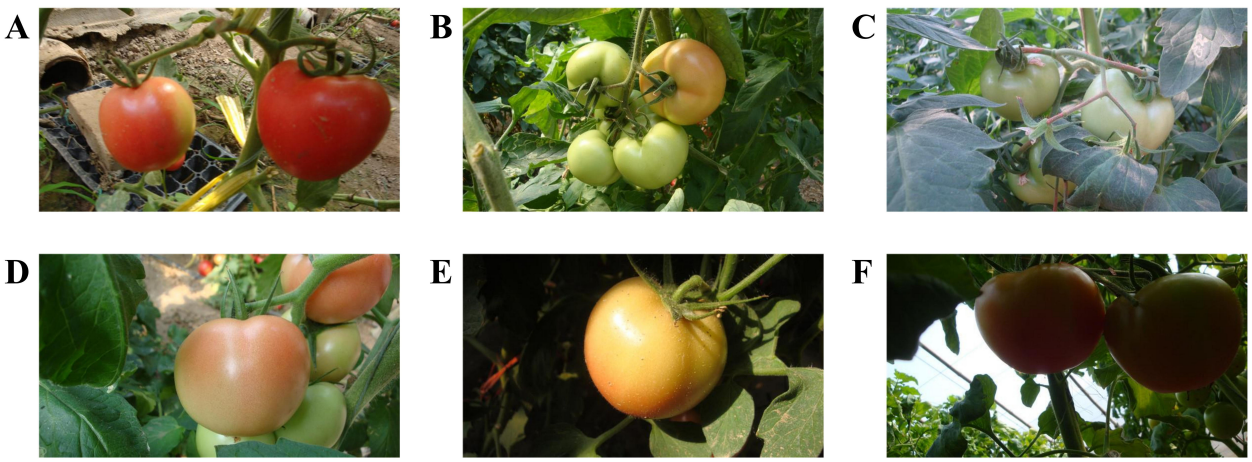


FIGURE 1
Tomato samples with different growing circumstances: **(A)** separated tomatoes, **(B)** a cluster of tomatoes, **(C)** occlusion case, **(D)** overlap case, **(E)** sunlight case, and **(F)** shade case.

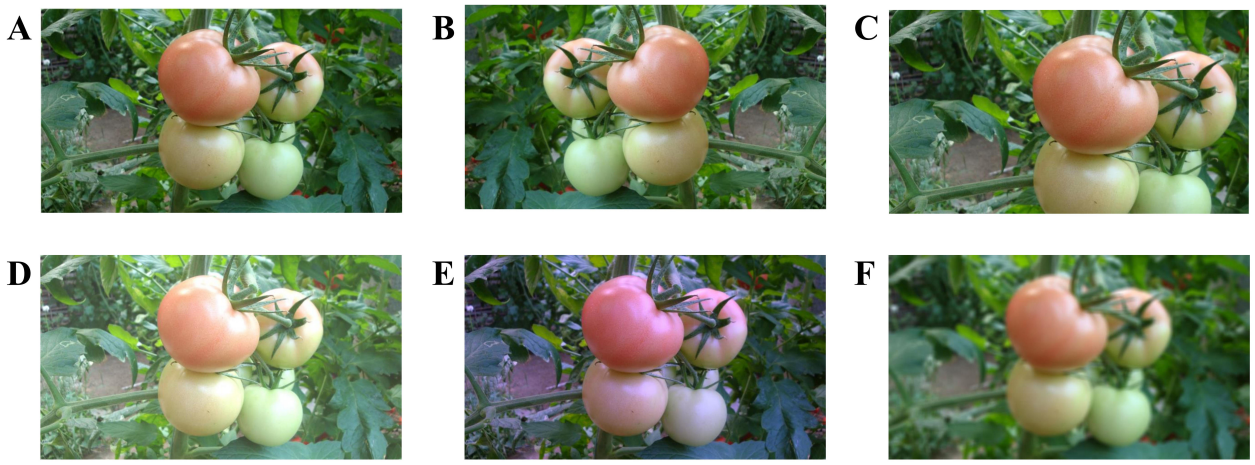


FIGURE 2
Data augmentation of tomato images: **(A)** original image, **(B)** horizontal flip, **(C)** scaling and cropping, **(D)** brightness transformation, **(E)** color balancing, and **(F)** image blurring.

the previous stage's feature maps, creating a richer and more diverse feature representation.

The neck network combines relevant feature maps from the backbone network using the PANet architecture (Liu et al., 2018) for feature fusion. In addition, YOLOv7 uses the RepConv technique (Ding et al., 2021) to address the challenges of detecting objects at various scales by enhancing the representability of feature maps. This technique also improves the inference results, although it increases the training time by introducing gradient diversity and allowing for more complex feature representations.

The head network uses anchor boxes to predict the objects' position, size, and class in the input image. Subsequently, a post-processing technique known as Non-Maximum Suppression (NMS) is employed to refine the predicted object boxes by eliminating redundant detections, enhancing the accuracy of YOLOv7.

3.4 The proposed YOLO-SwinTF

This study introduces the YOLO-SwinTF model, an advancement based on YOLOv7, to enhance the accuracy and robustness of tomato detection in complex environments. Figure 3 illustrates the architecture of the proposed YOLO-SwinTF model. It integrates three innovative modules to enhance the feature expression capability, improving the detection accuracy. Initially, ST blocks were incorporated into the backbone network, enabling the model to capture long-range dependencies efficiently. Subsequently, the TPN architecture replaced the original PANet in the neck network. This replacement was achieved by embedding multiple SP modules within the existing top-down and bottom-up architectures, facilitating the generation and effective propagation of new information within the feature maps. Finally, a Focaler-IoU loss was constructed using a linear interval mapping method. This method dynamically adjusts its focus based on the difficulty of the samples, significantly enhancing the detection capabilities of the model. Further details are provided in Sections 3.4.1 – 3.4.4.

3.4.1 Swin Transformer block

Although CNN networks can effectively extract local features, they are limited in capturing global features, impacting the final detection performance. In order to address this limitation, the current study introduces the attention mechanism of the Swin Transformer (Liu et al., 2021) to enhance the model's long-range dependencies. Unlike traditional Transformer structures, the Swin Transformer employs a hierarchical attention mechanism. In this structure, a sliding window performs attention computations separately at different layers, diverging from the standard multi-head self-attention (MSA) module. This approach not only facilitates the extraction of global information through long-distance modeling but also reduces the computational complexity of the original attention method. Figure 4 indicates that a Swin Transformer module primarily comprises a LayerNorm (LN) layer, a window-based multi-head self-attention (W-MSA) module, a shifted window-based multi-head self-attention (SW-MSA) module, a two-layer multi-layer perceptron (MLP) with a GELU non-linear activation function between layers, and a residual connection.

TABLE 1 The number of training images after data augmentation.

	Original	Honrizontal flip	Scaling and cropping	Brightness transformation	Color balancing	Blurring	Total
No. of images	725	725	725	725	725	725	4350

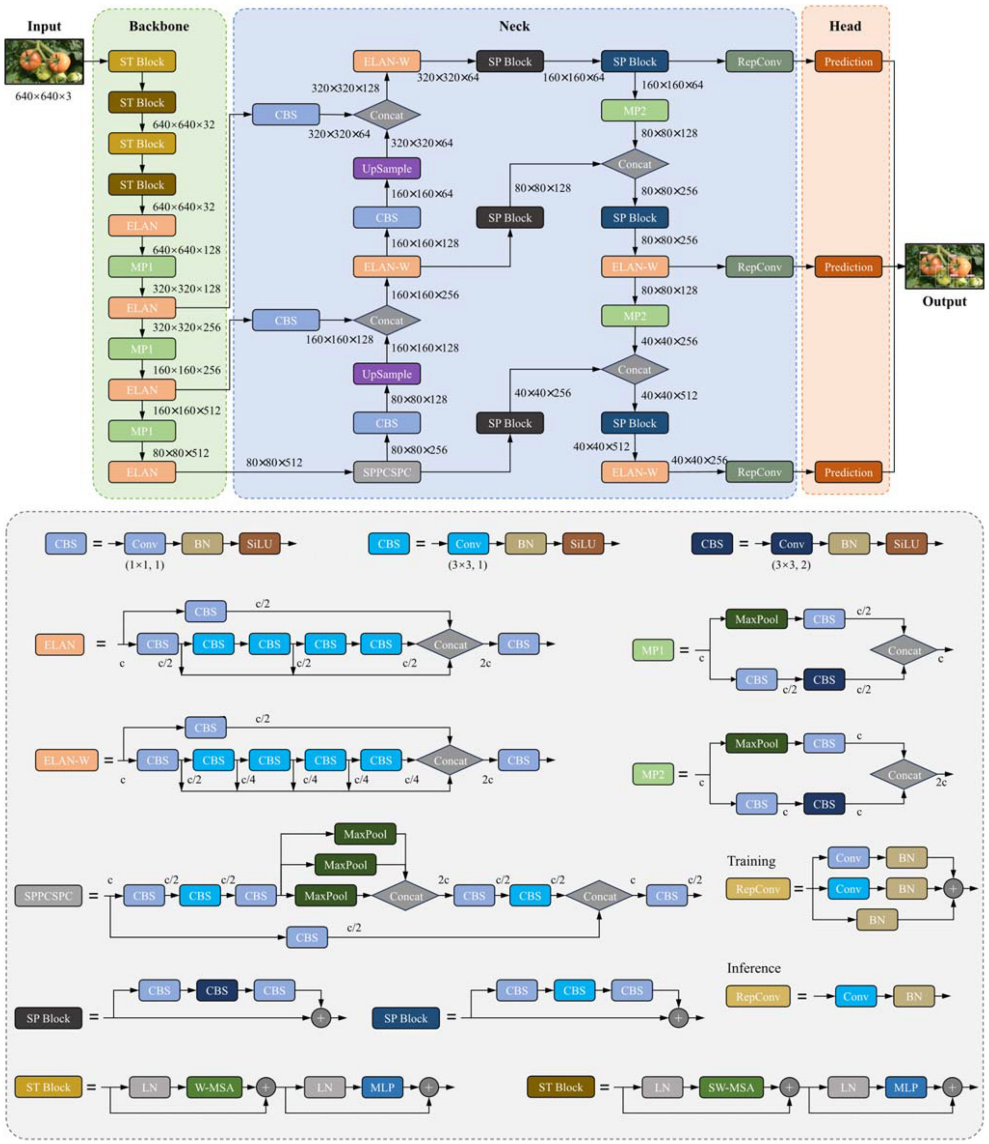


FIGURE 3
The architecture of the proposed YOLO-SwinTF.

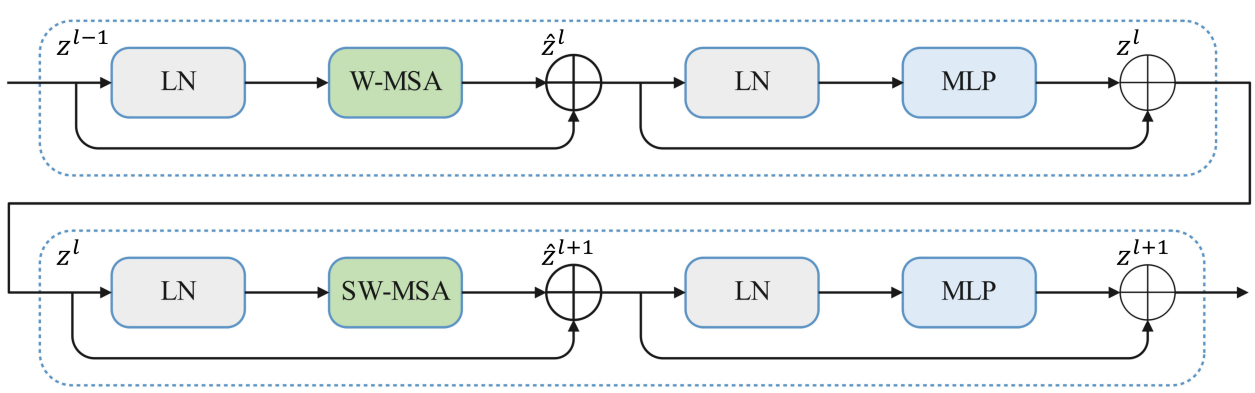


FIGURE 4
The Swin Transformer blocks.

Figure 4 shows that two consecutive Swin Transformer blocks are computed using Equations 1-4 (Liu et al., 2021):

$$\hat{z}^{l} = W - MSA(LN(z^{l-1})) + z^{l-1} \quad (1)$$

$$z^{l} = MLP(LN(\hat{z}^{l})) + \hat{z}^{l} \quad (2)$$

$$\hat{z}^{l+1} = SW - MSA(LN(z^{l})) + z^{l} \quad (3)$$

$$z^{l+1} = MLP(LN(\hat{z}^{l+1})) + \hat{z}^{l+1} \quad (4)$$

where $\hat{z}^{l}$ denotes the output of the (S)W-MSA module and z^{l} denotes the output of the MLP module of the lth block.

In order to enable the model to capture global information, the first four CBS modules in YOLOv7 were replaced with four successive ST blocks, thus expanding the network's receptive field and enriching contextual information, as depicted in Figure 3.

3.4.2 Trident Pyramid Network architecture

As discussed by Picron and Tuytelaars (2022), existing feature pyramid networks (FPN, PANet, and BiFPN) primarily focus on communication-based processing, enhancing feature fusion through top-down and bottom-up operations. These networks can become saturated with communication when multiple communication-based operations are performed consecutively, reducing efficiency. Accordingly, this study introduces the TPN architecture to replace PANet in YOLOv7, which achieves a better balance between communication-based processing and self-processing by alternating top-down and bottom-up operations and parallel self-processing mechanisms.

Specifically, the TPN architecture consists of traditional top-down and bottom-up operations and parallel SP modules, as illustrated in Figure 5. An SP module consists of several consecutive base self-processing layers, each designated as a bottleneck layer, as depicted in Figure 6.

Multiple SP modules were explicitly embedded between the original top-down and bottom-up architectures. As shown in Figure 3, the SP module was added after the SPPCSPC and ELAN-W modules in the bottom-up architecture. In addition, the SP module processed the features again after being merged into the top-down architecture. In this manner, communication-based processing is alternated with self-processing, enabling feature mapping to generate new information for delivery. The TPN architecture controls the amount of self-processing through the hyperparameter, the number of layers in the SP module, N, which is set to 2 in this study.

3.4.3 Focaler-IoU-based regression loss

The accuracy of bounding box localization is critical to target detection performance. However, existing studies often overlook the impact of the distribution of difficult samples (small targets that are difficult to accurately localize) and easy samples (targets that are easy to detect) on bounding box regression. This oversight can result in suboptimal performance and a lack of robustness in challenging scenarios. To address this issue, this study introduces Focaler-IoU (Zhang and Zhang, 2024) to enhance detector performance in the tomato detection task by effectively focusing on different regression samples.

Specifically, the Focaler-IoU reconstructs the original IoU loss through a linear interval mapping method that allows the loss function to adjust its focus according to the distribution of difficult and easy samples. The reconstructed Focaler-IoU $IoU^{focaler}$ is expressed as follows (Zhang and Zhang, 2024):

$$IoU^{focaler} = \begin{cases} 0, & IoU < d \\ \frac{IoU-d}{u-d}, & d \le IoU \le u \\ 1, & IoU > u \end{cases} \quad (5)$$

where IoU is the original IoU value, and d and u are both in the range of [0,1]. Adjusting the values of d and u can guide $IoU^{focaler}$ to focus on different regression samples. In this study, d and u were set to 0.1 and 0.9, respectively. Accordingly, the Focaler-IoU loss $L_{Focaler-IoU}$ is defined below:

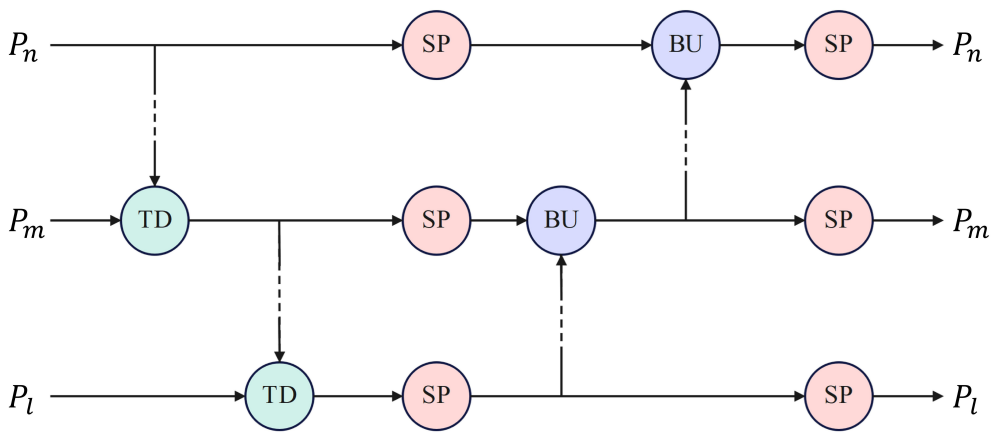


FIGURE 5
The TPN architecture. TD, BU and SP denotes top down, bottom up and self-processing modules, respectively.

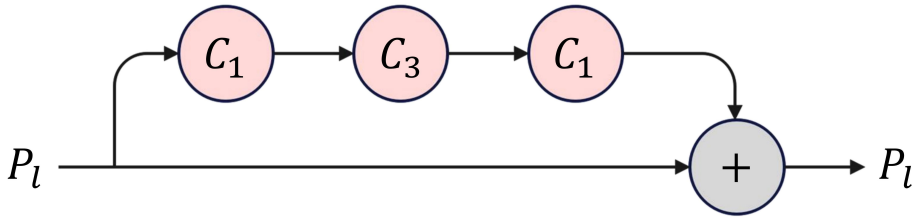


FIGURE 6
The architecture of a base self-processing layer. C_1 and C_3 denote convolution operations with kernel sizes of 1 and 3, respectively.

$$L_{Focaler-IoU} = 1 - IoU^{focaler} \tag{6}$$

Referring to Zhang and Zhang (2024), the Focaler-IoU loss is applied to the original CIoU-based bounding box regression loss used in YOLOv7, resulting in a novel regression loss as follows:

$$L_{reg} = L_{CIoU} + IoU - IoU^{focaler} \tag{7}$$

Where L_{CIoU} is expressed as follows (Zheng et al., 2020):

$$L_{CIoU} = 1 - IoU + \frac{d^2(b, b_{gt})}{c^2} + \beta v \tag{8}$$

where $d(\cdot)$ denotes Euclidean distance. b and b_{gt} denote the central points of the predicted and ground truth bounding boxes, respectively. β represents a positive trade-off parameter and v quantifies the consistency of the aspect ratio, as detailed below.

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w_{gt}}{h_{gt}} - \arctan\frac{w}{h}\right)^2 \tag{9}$$

$$\beta = \frac{v}{(1 - IoU) + v} \tag{10}$$

Combining Equations 7 and 8, we obtain the final regression loss as follows:

$$L_{reg} = 1 - IoU^{focaler} + \frac{d^2(b, b_{gt})}{c^2} + \beta v \tag{11}$$

This approach enables the loss function to dynamically adjust its focus between easy and difficult samples, enhancing the performance of the model in the detection task. Simultaneously, the adjustment of the loss function allows the model to concentrate more on positive samples that are difficult to classify and less on negative samples that are easy to classify. This adjustment effectively improves the model's response to the imbalance between difficult and easy samples.

3.4.4 Loss function

As in YOLOv7 (Wang et al., 2023), the loss function of the proposed model consists of three parts, i.e., the regression loss L_{reg}, confidence loss L_{conf}, and classification loss L_{cls}, and is expressed as follows:

$$L_{total} = \lambda_{reg} L_{reg} + \lambda_{conf} L_{conf} + \lambda_{cls} L_{cls} \tag{12}$$

where λ_{reg}, λ_{conf} and λ_{cls} were set to 5, 1, and 1, respectively, to balance the different losses. L_{reg}, L_{conf}, and L_{cls} are expressed in Equations 11, 13 and 14, respectively.

$$L_{conf} = \sum_{i=1}^{s\times sNB}\sum_{j=1}\lambda_{i,j}\left[-C_i \log \tilde{C}_i\right] - \sum_{i=1}^{s\times sNB}\sum_{j=1}(1 - \lambda_{i,j})\left[-(1 - C_i)\log(1 - \tilde{C}_i)\right] \tag{13}$$

$$L_{cls} = \sum_{i=1}^{s\times sNB}\sum_{j=1}\lambda_{i,j}\sum_{a\in \text{classes}}\left[p_i(a)\log \tilde{p}_i(a) + (1 - p_i(a))\log(1 - \tilde{p}_i(a))\right] \tag{14}$$

where $s \times s$ denotes the grid cell size, and NB is the number of bounding boxes. $\tilde{C}_i$ and C_i represent the confidence of the predicted box and the confidence threshold, respectively. $\lambda_{i,j}$ equals 1 if the jth bounding box falls in the ith grid cell and 0 otherwise. $\tilde{p}_i$ and p_i denote the predicted and ground truth class probabilities, respectively.

4 Experimental results and discussion

4.1 Experimental environment

Our experiments were conducted on a server with a 43GB Intel (R) Xeon(R) Platinum 8255C CPU operating at 2.50GHz and an NVIDIA GeForce RTX 3090 GPU. The server runs Ubuntu 18.04 as its underlying operating system. The proposed model was implemented using the PyTorch framework.

The model was trained with an input resolution of 640 × 640 pixels and a batch size of 32. The SGD optimizer was employed for training with a momentum of 0.937 and a weight decay of 0.0005. A cosine annealing schedule was applied to control changes in learning rates, starting with an initial learning rate of 0.001. The training was carried out over 160 epochs. The hyperparameters used in this study are listed in Table 2.

4.2 Evaluation metrics

For a thorough evaluation of the performance of the proposed method, the recall (R), precision (P), and F_1 score (Sa et al., 2016) were adopted as evaluation metrics. These metrics are defined as follows.

TABLE 2 The hyperparameter settings of YOLO-SwinTF.

Hyperparameter	Value
Initial learning rate	0.001
Weight decay	0.0005
Momentum	0.937
Batch size	32
Epochs	160

$$P = \frac{TP}{TP + FP} \tag{15}$$

$$R = \frac{TP}{TP + FN} \tag{16}$$

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{17}$$

where TP, FP, and FN denote true positive (correct detection), false positive (false detection), and false negative (missing detection), respectively.

In addition, this study employed Average Precision (AP) (Everingham et al., 2010) to assess the overall performance of the detection system. AP is defined as follows:

$$AP = \sum_{n} (r_{n+1} - r_n) p_{interp}(r_{n+1}) \tag{18}$$

$$p_{interp}(r_{n+1}) = \max_{\tilde{r}:\tilde{r} \geq r_{n+1}} p(\tilde{r}) \tag{19}$$

where $p(\tilde{r})$ is the measured precision at a recall level of $\tilde{r}$.

4.3 Ablation study

This study integrated three components, ST block, TPN, and Focaler-IoU, into the detection model to enhance its performance. An ablation study was conducted to assess the effectiveness of each modification within the proposed model. The results are presented in Table 3 and Figure 7. When the ST block, TPN, and Focaler-IoU are implemented individually, the detection performance improves regarding recall, precision, and AP. Due to the incorporation of the ST block, recall, precision, and AP increased by 0.49%, 0.23%, and 0.19%, respectively, compared to the original YOLOv7 model. This improvement results from the ability to learn global contextual features by establishing long-range dependencies. Including TPN raised the F_1 score and AP by 0.45% and 0.36%, respectively. Replacing the original IoU with Focaler-IoU led to increases in the F_1 score and AP of 0.28% and 0.31%, respectively, attributed to the effectiveness of the reconstructed regression loss in handling difficult small targets. The simultaneous use of the ST block and TPN in the model resulted in the F_1 score and AP of 95.81% and 98.33, increases of 0.51% and 0.35% over using the ST block alone, and 0.42% and 0.18% over using TPN alone. Combining the ST block and Focaler-IoU yielded an increase of 0.21% in both F_1 score and AP compared to using the ST block alone. When the TPN module was paired with the Focaler-IoU, the F_1 score and AP reached 95.71% and 98.20%, improvements of 0.32% and 0.05% over using TPN alone and 0.49% and 0.1% over using Focaler-IoU alone. Ultimately, integrating all three modules simultaneously enabled the proposed model to achieve optimal detection performance, with F_1 score and AP reaching 96.22% and 98.67%, respectively. Therefore, the effectiveness of the three enhancement methods – ST block, TPN, and Focaler-IoU-based regression loss – is verified.

4.4 Comparison of different models

A comparative study was conducted alongside leading detection algorithms currently utilized in the field to assess the effectiveness of the newly proposed YOLO-SwinTF model. This study included sophisticated models such as Faster R-CNN (Ren et al., 2015), CenterNet (Zhou et al., 2019), YOLOv4 (Bochkovskiy et al., 2020), YOLO-Tomato (Liu et al., 2020), YOLOv5 (Jocher, 2020), TomatoDet (Liu et al., 2022), YOLOv7 (Wang et al., 2023), YOLOv8 (Jocher et al., 2023), YOLOv9 (Wang et al., 2024b), and YOLOv10 (Wang et al., 2024a). Among these models, Faster R-CNN belongs to the two-stage detection models, whereas the others belong to the single-stage detection models. In addition, CenterNet and TomatoDet are categorized as anchor-free models, while the remaining models rely on anchors for detection. The

TABLE 3 Ablation study on different components of YOLO-SwinTF.

ST Block	TPN	Focaler-IoU	Recall (%)	Precision (%)	F_1 (%)	AP (%)
			94.63	95.25	94.94	97.79
✓			95.12	95.48	95.30	97.98
	✓		95.37	95.41	95.39	98.15
		✓	95.05	95.40	95.22	98.10
✓	✓		95.81	95.82	95.81	98.33
✓		✓	95.42	95.60	95.51	98.19
	✓	✓	95.72	95.70	95.71	98.20
✓	✓	✓	96.27	96.17	96.22	98.67

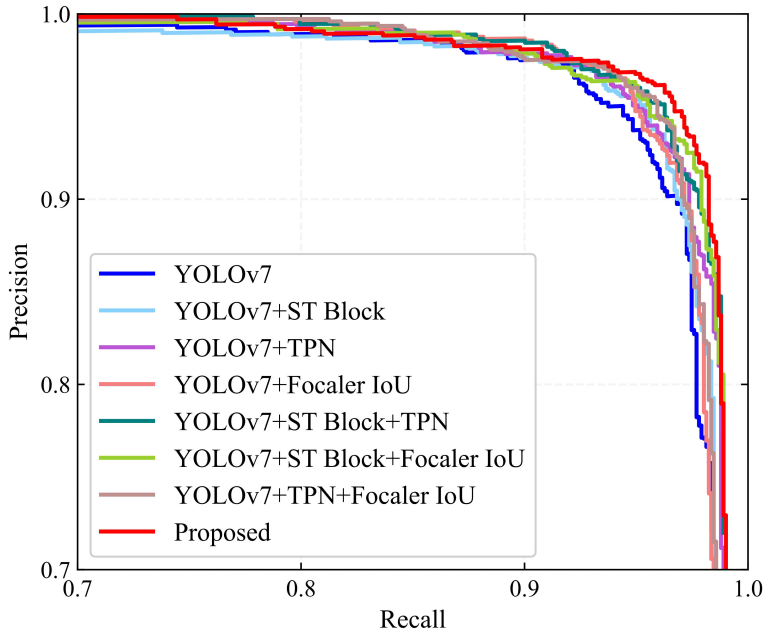


FIGURE 7
PR curves of the major components of YOLO-SwinTF for ablation study.

hyperparameters used for the comparative study, as specified in the original papers (Ren et al., 2015; Zhou et al., 2019; Bochkovskiy et al., 2020; Jocher, 2020; Liu et al., 2020, 2022; Jocher et al., 2023; Wang et al., 2023, 2024a, b), are listed in Table 4. Table 5 displays the detection performance metrics for all detection models, including recall, precision, F_1 score, AP, and average detection time. Precision-recall (PR) curves are illustrated in Figure 8. Table 5 shows that the proposed model outperforms other methods in all detection metrics, with the exception of detection time. In particular, the YOLO-SwinTF model excels in the F_1 score and AP, outperforming the second-ranked YOLOv10 by 0.53% and 0.21%, respectively. This improvement primarily benefits from integrating the attention mechanism, TPN architecture, and Focaler-IoU-based loss. However, in terms of detection speed, the YOLO-SwinTF model is 12 ms slower than YOLOv10, primarily due to YOLOv10's elimination of the post-processing step involving NMS, facilitated by the introduction of dual label assignments. This finding paves the way for our future research. Compared to the baseline model, YOLOv7, the YOLO-SwinTF model shows increases of 1.64% in recall, 0.92% in precision, 1.28% in F_1 score, and 0.88% in AP, demonstrating the effectiveness of the integrated modules in YOLOv7. The average detection time of the proposed

TABLE 4 The hyperparameter settings of different algorithms for comparison.

Models	Batch size	Momentum	Weight decay	Initial learning rate	Learning rate decay strategy	Epochs
Faster R-CNN	16	0.9	5×10^{-4}	10^{-3}	Divided by 10 after 90 epochs	120
CenterNet TomatoDet	32	0.9	10^{-4}	1.25×10^{-4}	Divided by 10 after 90 and 120 epochs	140
YOLO-Tomato	32	0.9	5×10^{-4}	10^{-3}	Divided by 10 after 60 and 90 epochs	160
YOLOv4 YOLOv5 YOLOv7 YOLOv8	32	0.937	5×10^{-4}	10^{-3}	Cosine annealing	160
YOLOv9 YOLOv10	32	0.937	5×10^{-4}	10^{-3}	Linear decay	160

TABLE 5 Tomato detection results of different algorithms.

Methods	Recall (%)	Precision (%)	F_1 (%)	AP (%)	Time (ms)
CenterNet	91.56	92.98	92.26	95.75	32
Faster R-CNN	91.78	92.89	92.33	94.37	231
YOLOv4	92.76	94.11	93.43	93.91	25
YOLO-Tomato	93.09	94.75	93.91	96.40	54
YOLOv5	93.64	94.57	94.10	97.79	22
TomatoDet	94.30	95.77	95.03	98.16	35
YOLOv7	94.63	95.25	94.94	97.79	15
YOLOv8	95.06	95.59	95.32	97.95	12
YOLOv9	95.19	95.71	95.45	98.21	12
YOLOv10	95.55	95.84	95.69	98.46	9
Proposed	96.27	96.17	96.22	98.67	21

model is 21 ms per image, fulfilling the requirements for real-time tomato detection in complex environments.

4.5 Network visualization

The Grad-CAM technique (Selvaraju et al., 2017) was employed to visualize the features of raw images to illustrate the superiority of the proposed YOLO-SwinTF intuitively. Specifically, ten images from the tomato dataset were selected, and visual experiments were conducted, as shown in Figure 9. The experimental results demonstrate that the image feature extractor, enhanced by the ST block, can capture global information by modeling long-range dependencies and extracting the most significant descriptive content from the raw samples. This capability is primarily attributed to the multi-head self-attention mechanism, which excels in capturing semantic information. In addition, the incorporation of TPN architecture facilitates a better balance between communication-based processing and self-processing, resulting in generating new information for propagation.

4.6 Performance of the proposed model under different lighting conditions

The tomato dataset used in this study was divided into sunlight and shade groups to evaluate the detection performance of the proposed model under different lighting conditions. Of all the test data, 425 tomatoes were in the shade, while the remaining 487 tomatoes were under sunlight. We used the correct identification rate (or recall), false identification rate, and missing rate as the evaluation metrics. The falsely identified tomatoes refer to the detected tomatoes that are actually background, and the term 'missed tomatoes' denotes tomatoes that the model did not detect. The detection results are listed in Table 6. As shown in Table 6, under sunlight conditions, 470 out of 487 tomatoes were correctly detected, with a detection rate of 96.51%. For the shade condition, the detection rate was 96.00%. In addition, under sunlight conditions, some backgrounds were incorrectly identified as tomatoes, with a total of 17 such instances, resulting in an incorrect identification rate of 3.49%. Under the shade condition, the false identification rate was 4.23%.

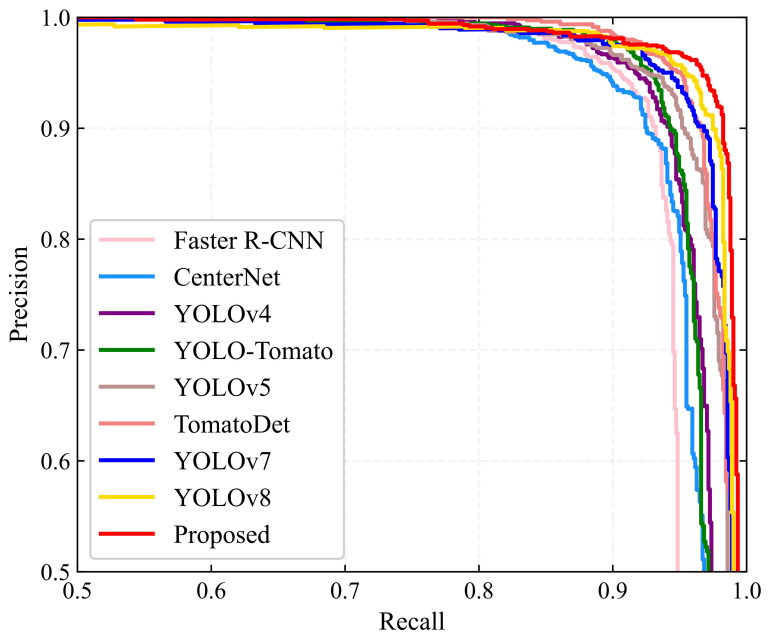


FIGURE 8
PR curves of different detection algorithms.

FIGURE 9
Visual features of images from the tomato dataset.

An analysis of the results indicated that these false identifications typically occurred when the tomatoes were similar in shape and color to the background. The above results show that the detection performance of the proposed model is comparable under both sunlight and shade conditions, verifying the robustness of the model to illumination variations. The detection results are shown in Figure 10.

4.7 Performance of the proposed model under different occlusion conditions

This study also evaluated the detection performance of the proposed model under different occlusion conditions, which are common in real environments. According to the degree of occlusion of the tomatoes by other objects, tomato data can be categorized into slight and severe occlusion cases. Severe occlusion is defined as the tomatoes being more than 50% occluded by leaves, branches, or other tomatoes, and conversely, they are recognized as slight cases, as defined by Liu et al. (2020). We used the same performance evaluation metrics as in the experiments under different lighting conditions. Table 7 lists the test results for different occlusion conditions. As shown in Table 7, 588 out of 609 tomatoes were correctly identified in the slight occlusion condition, with a detection rate of 96.55%, slightly better than in the severe occlusion condition. The false identification rates in the slight and severe occlusion conditions were 3.45% and 4.61%, respectively, indicating that overlap or occlusion can affect the model's detection performance. Almost all tomatoes can be detected when the degree of overlap or occlusion is not very severe. The semantic loss of images resulting from overlap or occlusion can be compensated by the model's attention mechanism and the implicit contextual information mining of hierarchical feature extraction. The model's performance in

TABLE 6 Performance of the proposed model under different lighting conditions.

Illumination	Tomato Count	Correctly Identified		Falsely Identified		Missed	
		Amount	Rate (%)	Amount	Rate (%)	Amount	Rate (%)
Sunlight	487	470	96.51	17	3.49	17	3.49
Shade	425	408	96.00	18	4.23	17	4.00

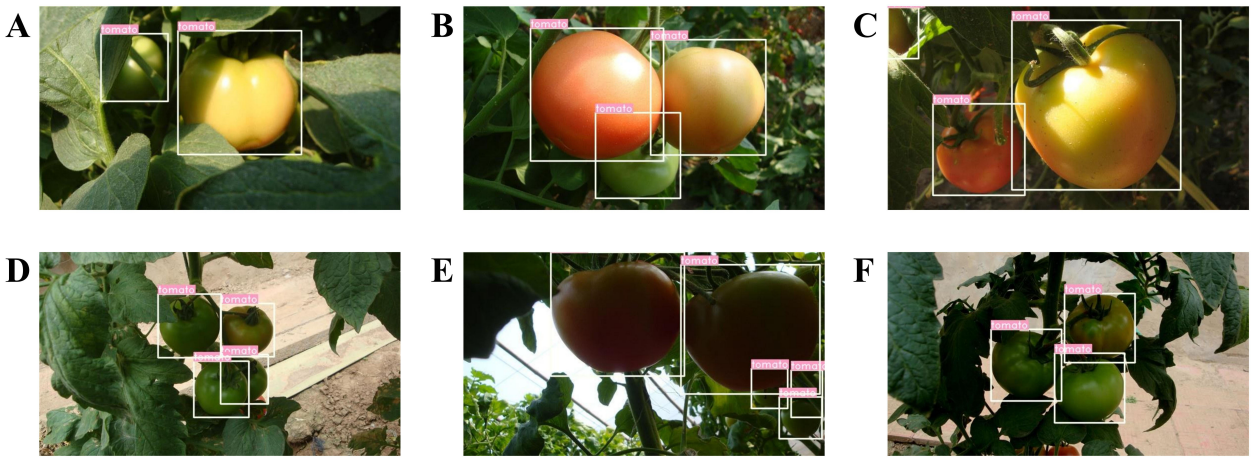


FIGURE 10
Some examples of the detection results under different lighting conditions: **(A-C)** sunlight conditions, and **(D-F)** shade conditions.

TABLE 7 **Performance of the proposed model under different occlusion conditions.**

Occlusion Condition	Tomato Count	Correctly Identified		Falsely Identified		Missed	
		Amount	Rate (%)	Amount	Rate (%)	Amount	Rate (%)
Slight case	609	588	96.55	21	3.45	21	3.45
Severe case	303	290	95.71	14	4.61	13	4.29

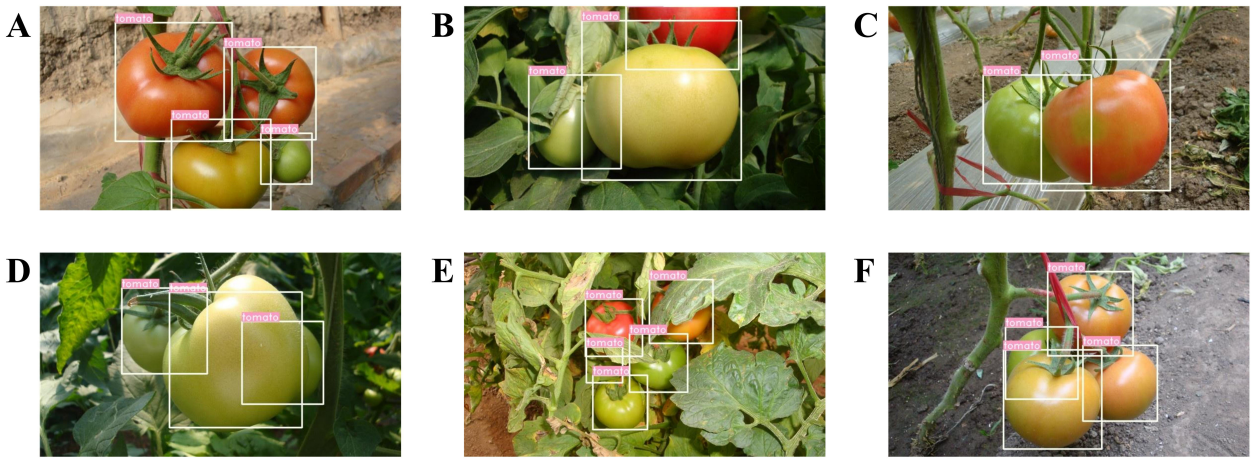


FIGURE 11
Some examples of detection results under different occlusion conditions: **(A-C)** slight cases and **(D-F)** severe cases.

detecting overlapping and occluded tomatoes can be further improved by explicitly modeling the contextual environment of tomatoes. Figure 11 shows some of the detection results.

5 Conclusion

This study proposes a YOLO-SwinTF model designed to enhance the feature expression capabilities of YOLOv7 to achieve accurate tomato detection in complex environments. Initially, the backbone network of the proposed framework incorporates Swin Transformer modules to represent global information by modeling long-range visual dependencies. Subsequently, in the neck network, the TPN architecture replaces the PANet to better balance communication-based processing and self-processing, generating new information for delivery in the feature map. Finally, a novel regression loss based on Focaler-IoU is constructed in bounding box regression to allow the loss function to dynamically adjust its

focus between easy and difficult samples, enhancing the model's detection performance.

Extensive experiments are conducted to verify the performance of the proposed method. The F_1 score and AP of the proposed YOLO-SwinTF model reached 96.22% and 98.67%, respectively, surpassing other state-of-the-art detectors. Ablation studies are performed to verify the effectiveness of each modification. In addition, the model demonstrates strong robustness in detecting tomatoes under various illumination and occlusion conditions. The experimental results confirm the proposed model is highly suitable for tomato detection in complex environments.

In the future, the ripeness information of tomatoes at different growth stages will be utilized to achieve multi-stage tomato detection. In addition, we plan to implement explicit context modeling for tomatoes to improve the detection performance of overlapping and occluded tomatoes.

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

GL: Conceptualization, Methodology, Writing – original draft, Funding acquisition. YZ: Writing – original draft, Investigation. JL: Validation, Writing – review & editing. DL: Writing – review & editing, Formal analysis. CC: Writing – review & editing, Software. YL: Writing – review & editing, Data curation. XZ: Writing – review & editing, Software. PT: Supervision, Writing – review & editing, Writing – original draft.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by Shandong Provincial Natural Science Foundation (ZR2021MC044, ZR2021MF085, ZR2021QC173, ZR2023QC116), Weifang Science and Technology Development Plan (2023GX016), Weifang Soft Science Project (2023RKX184), and Doctoral Research Foundation of Weifang University (2022BS70, 2024BS39).

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

Ashtiani, S.-H. M., Javanmardi, S., Jahanbanifard, M., Martynenko, A., and Verbeek, F. J. (2021). Detection of mulberry ripeness stages using deep learning models. *IEEE Access* 9, 100380–100394. doi: 10.1109/ACCESS.2021.3096550

Bargoti, S., and Underwood, J. (2017). "Deep fruit detection in orchards," in *2017 IEEE international conference on robotics and automation (ICRA)*, Singapore, 3626–3633 (IEEE).

Behera, S. K., Rath, A. K., and Sethy, P. K. (2020). Fruit recognition using support vector machine based on deep features. *Karbala Int. J. Modern Sci.* 6, 16. doi: 10.33640/2405-609X.1675

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

Chaivivatrakul, S., and Dailey, M. N. (2014). Texture-based fruit detection. *Precis. Agric.* 15, 662–683. doi: 10.1007/s11119-014-9361-x

Chen, J., Kao, S.-h., He, H., Zhuo, W., Wen, S., Lee, C.-H., et al. (2023). "Run, don't walk: Chasing higher flops for faster neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Vancouver, Canada: IEEE, 12021–12031.

Ding, X., Zhang, X., Han, J., and Ding, G. (2022). "Scaling up your kernels to 31x31: Revisiting large kernel design in cnns," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. Lousiana, New Orleans: IEEE, 11963–11975.

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*. Nashville, TN, USA: IEEE, 13733–13742.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision* 88, 303–338. doi: 10.1007/s11263-009-0275-4

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

Ganesh, P., Volle, K., Burks, T., and Mehta, S. (2019). Deep orange: Mask r-cnn based orange detection and segmentation. *Ifac-papersonline* 52, 70–75. doi: 10.1016/j.ifacol.2019.12.499

Gao, C., Jiang, H., Liu, X., Li, H., Wu, Z., Sun, X., et al. (2024). Improved binocular localization of kiwifruit in orchard based on fruit and calyx detection using yolov5x for robotic picking. *Comput. Electron. Agric.* 217, 108621. doi: 10.1016/j.compag.2024.108621

Guo, J., Yang, Y., Lin, X., Memon, M. S., Liu, W., Zhang, M., et al. (2023). Revolutionizing agriculture: Real-time ripe tomato detection with the enhanced tomato-yolov7 system. *IEEE Access* 11, 133086–133098. doi: 10.1109/ACCESS.2023.3336562

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: IEEE, 2961–2969.

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

Hernández, G. A. A., Olguin, J. C., Vasquez, J. I., Uriarte, A. V., and Torres, M. C. V. (2023). Detection of tomato ripening stages using yolov3-tiny. *arXiv preprint arXiv:2302.00164*. doi: 10.48550/arXiv.2302.00164

Jana, S., and Parekh, R. (2017). "Shape-based fruit recognition and classification," in *Computational Intelligence, Communications, and Business Analytics: First International Conference, CICBA 2017*, Kolkata, India, March 24–25, 2017. 184–196 (Springer), *Revised Selected Papers, Part II*.

Ji, W., Zhao, D., Cheng, F., Xu, B., Zhang, Y., and Wang, J. (2012). Automatic recognition vision system guided for apple harvesting robot. *Comput. Electric. Eng.* 38, 1186–1195. doi: 10.1016/j.compeleceng.2011.11.005

Jia, W., Xu, Y., Lu, Y., Yin, X., Pan, N., Jiang, R., et al. (2023). An accurate green fruits detection method based on optimized yolox-m. *Front. Plant Sci.* 14, 1187734. doi: 10.3389/fpls.2023.1187734

Jiao, Y., Luo, R., Li, Q., Deng, X., Yin, X., Ruan, C., et al. (2020). Detection and localization of overlapped fruits application in an apple harvesting robot. *Electronics* 9, 1023. doi: 10.3390/electronics9061023

Jocher, G. (2020). *YOLOv5 by Ultralytics*. doi: 10.5281/zenodo.3908559

Jocher, G., Chaurasia, A., and Qiu, J. (2023). *Ultralytics YOLO*.

Kelman, E. E., and Linker, R. (2014). Vision-based localisation of mature apples in tree images using convexity. *Biosyst. Eng.* 118, 174–185. doi: 10.1016/j.biosystemseng.2013.11.007

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 25, 1097–1105. doi: 10.1145/3065386

Kurtulmus, F., Lee, W. S., and Vardar, A. (2011). Green citrus detection using 'eigenfruit', color and circular gabor texture features under natural outdoor conditions. *Comput. Electron. Agric.* 78, 140–149. doi: 10.1016/j.compag.2011.07.001

Lam, E. Y. (2005). "Combining gray world and retinex theory for automatic white balance in digital photography," in *Proceedings of the Ninth International Symposium on Consumer Electronics, 2005.(ISCE 2005)*, Macau, China. 134–139 (IEEE).

Li, C., Zhou, A., and Yao, A. (2022). Omni-dimensional dynamic convolution. *arXiv preprint arXiv:2209.07947*. doi: 10.48550/arXiv.2209.07947

Li, L., Tang, S., Deng, L., Zhang, Y., and Tian, Q. (2017). "Image caption with global-local attention," in *Proceedings of the AAAI conference on artificial intelligence*. San Francisco, California, USA: AAAI Press, Vol. 31.

Liu, G., Hou, Z., Liu, H., Liu, J., Zhao, W., and Li, K. (2022). Tomatodet: Anchor-free detector for tomato detection. *Front. Plant Sci.* 13, 942875. doi: 10.3389/fpls.2022.942875

Liu, G., Mao, S., and Kim, J. H. (2019). A mature-tomato detection algorithm using machine learning and color analysis. *Sensors* 19, 2023. doi: 10.3390/s19092023

Liu, G., Nouaze, J. C., Touko Mbouembe, P. L., and Kim, J. H. (2020). Yolo-tomato: A robust algorithm for tomato detection based on yolov3. *Sensors* 20, 2145. doi: 10.3390/s20072145

Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018). "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. Salt Lake City, UT, USA: IEEE, 8759–8768.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. (2016). "Ssd: Single shot multibox detector," in *European conference on computer vision*. 21–37 (Amsterdam, The Netherlands: Springer), Proceedings, Part I 14.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *Proc. IEEE/CVF Int. Conf. Comput. vision.*, 10012–10022. doi: 10.1109/ICCV48922.2021.00986

Mbouembe, P. L. T., Liu, G., Sikati, J., Kim, S. C., and Kim, J. H. (2023). An efficient tomato-detection method based on improved yolov4-tiny model in complex environment. *Front. Plant Sci.* 14, 1150958. doi: 10.3389/fpls.2023.1150958

Payne, A., Walsh, K., Subedi, P., and Jarvis, D. (2014). Estimating mango crop yield using image analysis using fruit at 'stone hardening'stage and night time imaging. *Comput. Electron. Agric.* 100, 160–167. doi: 10.1016/j.compag.2013.11.011

Peng, H., Shao, Y., Chen, K., Deng, Y., and Xue, C. (2018). Research on multi-class fruits recognition based on machine vision and svm. *IFAC-PapersOnLine* 51, 817–821. doi: 10.1016/j.ifacol.2018.08.094

Picron, C., and Tuytelaars, T. (2022). Trident pyramid networks for object detection. *Proc. BMVC*, p. 241.

Qi, Y., He, Y., Qi, X., Zhang, Y., and Yang, G. (2023). "Dynamic snake convolution based on topological geometric constraints for tubular structure segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Paris, France: IEEE, 6070–6079.

Rakun, J., Stajnko, D., and Zazula, D. (2011). Detecting fruits in natural scenes by using spatial-frequency based texture analysis and multiview geometry. *Comput. Electron. Agric.* 76, 80–88. doi: 10.1016/j.compag.2011.01.007

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*. Las Vegas, NV, USA: IEEE, 779–788.

Redmon, J., and Farhadi, A. (2017). Yolo9000: better, faster, stronger. *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 7263–7271. doi: 10.1109/CVPR.2017.690

Redmon, J., and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint 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. *Adv. Neural Inf. Process. Syst.* 39 (6), 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., and McCool, C. (2016). Deepfruits: A fruit detection system using deep neural networks. *sensors* 16, 1222. doi: 10.3390/s16081222

Samajpati, B. J., and Degadwala, S. D. (2016). "Hybrid approach for apple fruit diseases detection and classification using random forest classifier," in *2016 International conference on communication and signal processing (ICCSP) (IEEE)*. Melmaruvathur, India: IEEE, 1015–1019.

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 localization," in *Proceedings of the IEEE international conference on computer vision*. Venice, Italy: IEEE, 618–626.

Tan, M., and Le, Q. (2019). "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning (PMLR)*. Long Beach, California, USA: PMLR Press, 6105–6114.

Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., et al. (2024a). Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*. doi: 10.48550/arXiv.2405.14458

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, Canada: IEEE, 7464–7475.

Wang, C.-Y., Liao, H.-Y. M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., and Yeh, I.-H. (2020). "Cspnet: A new backbone that can enhance learning capability of cnn," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. Seattle, WA, USA: IEEE, 390–391.

Wang, C.-Y., Liao, H.-Y. M., and Yeh, I.-H. (2022). Designing network design strategies through gradient path analysis. *arXiv preprint arXiv:2211.04800*. doi: 10.48550/arXiv.2211.04800

Wang, C.-Y., Yeh, I.-H., and Liao, H.-Y. M. (2024b). Yolov9: Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*. doi: 10.48550/arXiv.2402.13616

Wang, W., Shi, Y., Liu, W., and Che, Z. (2024c). An unstructured orchard grape detection method utilizing yolov5s. *Agriculture* 14, 262. doi: 10.3390/agriculture14020262

Wei, X., Jia, K., Lan, J., Li, Y., Zeng, Y., and Wang, C. (2014). Automatic method of fruit object extraction under complex agricultural background for vision system of fruit picking robot. *Optik* 125, 5684–5689. doi: 10.1016/j.ijleo.2014.07.001

Yu, L., Xiong, J., Fang, X., Yang, Z., Chen, Y., Lin, X., et al. (2021). A litchi fruit recognition method in a natural environment using rgb-d images. *Biosyst. Eng.* 204, 50–63. doi: 10.1016/j.biosystemseng.2021.01.015

Zeng, T., Li, S., Song, Q., Zhong, F., and Wei, X. (2023). Lightweight tomato real-time detection method based on improved yolo and mobile deployment. *Comput. Electron. Agric.* 205, 107625. doi: 10.1016/j.compag.2023.107625

Zhang, H., and Zhang, S. (2024). Focaler-iou: More focused intersection over union loss. *arXiv preprint arXiv:2401.10525*. doi: 10.48550/arXiv.2401.10525

Zhao, Y., Gong, L., Zhou, B., Huang, Y., and Liu, C. (2016). Detecting tomatoes in greenhouse scenes by combining adaboost classifier and colour analysis. *Biosyst. Eng.* 148, 127–137. doi: 10.1016/j.biosystemseng.2016.05.001

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., and Ren, D. (2020). "Distance-iou loss: Faster and better learning for bounding box regression," in *Proceedings of the AAAI conference on artificial intelligence*. New York, USA: AAAI Press, Vol. 34. 12993–13000.

Zhou, X., Wang, D., and Krähenbühl, P. (2019). Objects as points. *arXiv preprint arXiv:1904.07850*. doi: 10.48550/arXiv.1904.07850

Zhu, H., Yang, L., Fei, J., Zhao, L., and Han, Z. (2021). Recognition of carrot appearance quality based on deep feature and support vector machine. *Comput. Electron. Agric.* 186, 106185. doi: 10.1016/j.compag.2021.106185

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 29 November 2024
DOI 10.3389/fpls.2024.1442968

OPEN ACCESS

EDITED BY
Mohsen Yoosefzadeh Najafabadi,
University of Guelph, Canada

REVIEWED BY
Seyed-Hassan Miraei Ashtiani,
Dalhousie University, Canada
Maciej Przybyłek,
Nicolaus Copernicus University in Toruń, Poland

*CORRESPONDENCE
Guangyao Pang
pangguangyao@gmail.com

RECEIVED 03 June 2024
ACCEPTED 04 November 2024
PUBLISHED 29 November 2024

CITATION
Zhu X, Pang G, He X, Chen Y and Yu Z (2024) A segmentation-combination data augmentation strategy and dual attention mechanism for accurate Chinese herbal medicine microscopic identification. *Front. Plant Sci.* 15:1442968. doi: 10.3389/fpls.2024.1442968

A segmentation-combination data augmentation strategy and dual attention mechanism for accurate Chinese herbal medicine microscopic identification

Xiaoying Zhu[1,2], Guangyao Pang[1,2]*, Xi He[1,2], Yue Chen[1,2] and Zhenming Yu[1]

[1]Guangxi Colleges and Universities Key Laboratory of Intelligent Software, Wuzhou University, Wuzhou, China, [2]Guangxi Key Laboratory of Machine Vision and Intelligent Control, Wuzhou University, Wuzhou, China

Introduction: Chinese Herbal Medicine (CHM), with its deep-rooted history and increasing global recognition, encounters significant challenges in automation for microscopic identification. These challenges stem from limitations in traditional microscopic methods, scarcity of publicly accessible datasets, imbalanced class distributions, and issues with small, unevenly distributed, incomplete, or blurred features in microscopic images.

Methods: To address these challenges, this study proposes a novel deep learning-based approach for Chinese Herbal Medicine Microscopic Identification (CHMMI). A segmentation-combination data augmentation strategy is employed to expand and balance datasets, capturing comprehensive feature sets. Additionally, a shallow-deep dual attention module enhances the model's ability to focus on relevant features across different layers. Multi-scale inference is integrated to process features at various scales effectively, improving the accuracy of object detection and identification.

Results: The CHMMI approach achieved an Average Precision (AP) of 0.841, a mean Average Precision at IoU=.50 (mAP@.5) of 0.887, a mean Average Precision at IoU from .50 to .95 (mAP@.5:.95) of 0.551, and a Matthews Correlation Coefficient of 0.898. These results demonstrate superior performance compared to state-of-the-art methods, including YOLOv5, SSD, Faster R-CNN, and ResNet.

Discussion: The proposed CHMMI approach addresses key limitations of traditional methods, offering a robust solution for automating CHM microscopic identification. Its high accuracy and effective feature processing capabilities underscore its potential to modernize and support the growth of the CHM industry.

KEYWORDS

Chinese herbal medicine, deep learning, attention mechanism, cell recognition, data augmentation

1 Introduction

Chinese Herbal Medicine (CHM) is a cornerstone of traditional Eastern healthcare and has been integrated into disease treatment. With roots deeply embedded in ancient Chinese science, CHM symbolizes Eastern medicine's cultural heritage and underscores a comprehensive medical paradigm that has garnered global recognition for its efficacy. This acknowledgement has notably surged during the COVID-19 pandemic, highlighting the potential of CHM in contributing to contemporary medical practices and prompting a broader international acceptance and trust in its remedies. The burgeoning trust in CHM has catalyzed a substantial expansion of its market, with recent data indicating an annual output reaching 4,555 million tons and daily testing frequencies surpassing 22 million instances. CHM includes plant, animal, and mineral medicines, and according to the Chinese Materia Medica, there are 8,980 kinds of herbs in total. With the addition of medicines used by ethnic minorities, the number of varieties has reached more than 28,000 so far (Li, 1999). These figures reflect the growing reliance on CHM for healthcare purposes and underscore the potential of the fast inspection market within this domain. However, the predominant methodologies employed for CHM identification, particularly through traditional manual microscopy, present numerous challenges. These methods are labor-intensive, require extensive expert knowledge, suffer from low throughput due to the microscopic equipment's limited field of view, and are prone to human error from tester fatigue, potentially leading to misjudgments.

There are four traditional identification methods for CHM: original plant (i.e., animal) identification, character identification, microscopic identification, and physical and chemical identification. Original plant (i.e., animal) identification Yin et al. (2019) was performed by observing the appearance of plants, animals, and minerals in morphological form and classifying herbs using knowledge of taxonomy. Character identification Thongkhao et al. (2020) was carried out by eyes, hand, nose, mouth taste, water test, fire test, and other simple ways to identify medicinal materials. Microscopic identification Ichim et al. (2020) uses microscopy to observe tissue structure, cell shape, and the features of inclusions of medicinal herbs to determine the nature of cell walls and cell inclusions or the distribution of active ingredients of certain species in tissues, and finally to achieve the identification of authenticity of herbal medicines. Physical and chemical identification Peng and Tsa (2020) is to use certain physical, chemical, or instrumental analysis methods to identify the authenticity, purity, and quality of traditional Chinese medicines. Generally, the first three conventional identification techniques rely primarily on abundant working experience, making distinguishing similar or analogous substances difficult.

However, physical and chemical identification is a highly advanced technique, particularly tedious, requiring specialized equipment and high costs. The need for an advanced, reliable, and less subjective method is evident, particularly to keep pace with the increasing scale of CHM testing and support the industry's growth and modernization efforts.

The development of artificial neural networks has opened up new avenues for image recognition, and deep learning-based methods have shown great success in various applications Chen et al. (2022); Jiang et al. (2022). As shown in Figure 1, several key challenges hinder the development of automated CHM microscopic identification systems:

1) **Data collection difficulties and class imbalance:** We found no publicly available herbal microscopic image datasets after

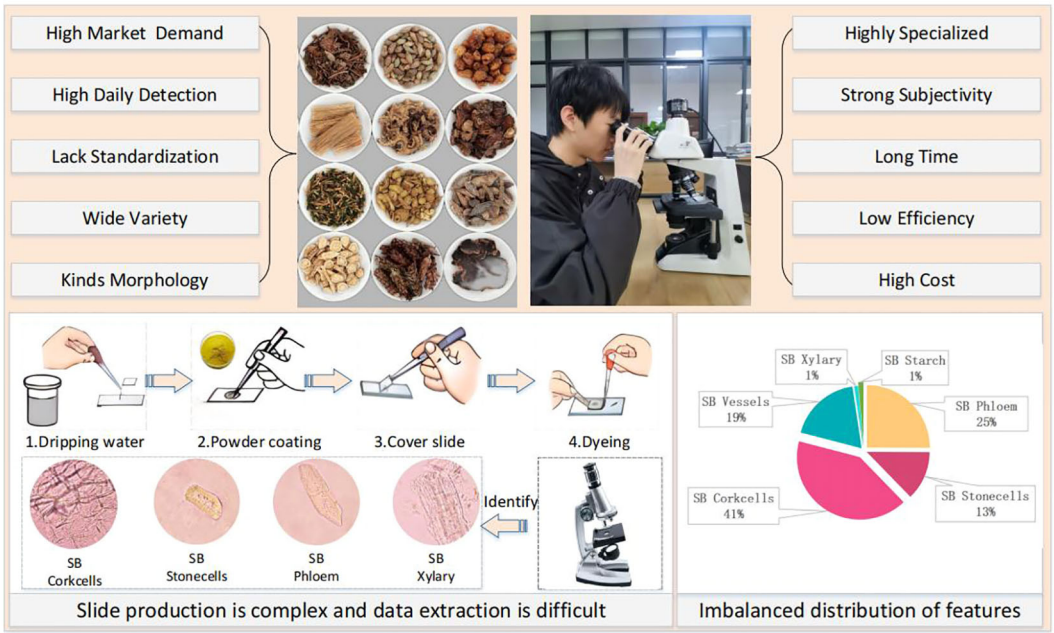


FIGURE 1
Challenges in Chinese herbal medicine microscopy identification.

reviewing the literature and searching search engines. CHM image datasets often exhibit significant class imbalance, where certain cell types or features are underrepresented. This can lead to biased models that perform poorly on rare classes.

2) **Small and Uneven Features:** CHM microscopic images contain small and unevenly distributed features, making it difficult for traditional object detection algorithms to locate and classify them accurately.

3) **Incomplete and Blurriness Cell Structures:** The grinding process used to prepare CHM samples can damage cell structures, resulting in incomplete or ambiguous features that further complicate identification.

This paper proposes a novel methodology, CHMMI, which innovatively applies a segmentation-combination method for data augmentation, allowing the model to capture more comprehensive feature sets from the available microscopic images. Furthermore, by integrating attention mechanisms, CHMMI enhances the model's focus on relevant features across different layers, thereby improving the accuracy of CHM identification. Finally, features across multiple scales and dimensions effectively detect and identify herbal microscopic images. The contributions of this paper can be summarized as follows:

- We propose a data augmentation strategy for generating more datasets by random cutting and random combination for the problem that a single image in CHM micrographs includes many different cells, which can extend and balance the datasets and provide a solid foundation for the training and prediction of the actual model.
- We develop a shallow-deep dual attention module that effectively captures valid auxiliary information from different channels in shallow and deep layers. This facilitates the processing of small, uneven features and incomplete and blurry cell structures in CHM.
- In the final prediction stage, we integrate three features with different object scales through a multi-scale inference module to predict objects in the image.
- We evaluate the performance of CHMMI through a series of comparison experiments with existing state-of-the-art approaches, such as YOLOv5 Zhu et al. (2021), SSD Liu et al. (2016), Faster R-CNN Khan et al. (2022), and ResNet He et al. (2016). The experimental results demonstrate that CHMMI achieves higher accuracy than these approaches, highlighting its potential for practical application in CHM microscopic identification.

2 Related work

Image recognition has significantly advanced by integrating deep learning techniques, predominantly categorized into one-stage and two-stage detection algorithms. These methodologies have been extensively employed across various sectors, including healthcare, autonomous driving, and precision agriculture, progressively encompassing microscopic image analysis for CHM.

2.1 Deep learning-based image recognition methods

Several image recognition approaches based on deep learning have been proposed, including two-stage detection algorithms (e.g., Faster RCNN, SSD) and one-stage detection algorithms (e.g., RetinaNet, YOLO). These algorithms have achieved state-of-the-art performance in various image recognition tasks, such as face detection, object detection, and image classification. For example, Sun et al. (2018) improved the state-of-the-art Faster RCNN framework by combining several strategies, proposed a new face detection scheme using Deep Learning, and achieved the state-of-the-art detection performance on the well-known FDDB face detection benchmark evaluation. Zhai et al. (2020) proposed an improved SSD object detection algorithm based on Dense Convolutional Network (DenseNet) and feature fusion; the algorithm replaces the original backbone network VGG-16 of SSD with DenseNet-S-32-1 to enhance the feature extraction ability of the model. Wang et al. (2020) proposed an automatic ship detection model based on RetinaNet, the model solves the problem that ships have multi-scale shape features in SAR images due to the diversity of SAR imaging patterns and the diversity of ship shapes, resulting in poor recognition rates. Yu et al. (2021) proposed a Deep Learning model named YOLOv4-FPM to realize real-time detection for bridge cracks by unmanned aerial vehicles. Yan et al. (2021) proposed an improved yolov5-based lightweight apple target detection approach for apple picking robots to address the problem that existing apple detection algorithms cannot distinguish between apples obscured by tree branches and apples obscured by other apples, leading to picking failure. Kim et al. (2022) proposed an approach with Maritime Dataset on modified YOLO-V5 with the SMD-Plus, the approach solves the problem of poor recognition rates due to the presence of noisy labels and imprecisely positioned bounding boxes in SMD.

The YOLO series of algorithms have been widely used in various applications, including object detection, pedestrian detection, and facial recognition. YOLOv5, in particular, has been shown to be effective in detecting objects in images with varying sizes, scales, and orientations.

2.2 Microscopic image recognition for Chinese herbal medicine

In microscopic image recognition for CHM, researchers focus on several challenges, including the uneven distribution of sample classes and small differences between classes, stereoscopic features of cells, and the effect of background color on recognition rate.

For the first type of problem, Wang et al. (2020b) used techniques such as dynamic ReLU function and multi-channel color space to use Xception with obvious classification effect as the base network, and replaced the static ReLU in the network with dynamic ReLU so that each small sample has a unique ReLU parameter. For the second type of problem, Ying et al. (2012) analyzed the differences in the characteristics of cross-sections and powders of stems and leaves of two herbs, Buddleja albiflora Hemsl

and Buddleja davidii Franch, which provided important criteria for the recognition of these two herbs. Ye et al. (2014) used a method of fusion of coaxial X-ray and micro-CT imaging techniques for three-dimensional nondestructive *in situ* microscopic imaging of the microscopic image of Amomi Rotundus Fructus and Alpiniae Katsumadai Semen seeds. This method obtained information on the microscopic image's internal microstructure and different cross-sectional orientations. For the third type of problem, Wang et al. (2017) used MATLAB software to program the stitching of the cross-sectional tissue images of the CHM Achyranthes bidentata and Cyathula officinalis. The features such as texture, color, and invariant moment of the microscopic image were extracted to recognize the two herbs effectively. Wang et al. (2020a) used a multi-channel and improved attention method to stitch the microscopic image data of 34 herbal catheters with images of different color spaces of the images themselves before inputting them into the network, and the method effectively improved the accuracy of recognition.

The above work mainly focuses on researching a single problem. However, three types of problems simultaneously exist in detecting CHM microscopic images. Our CHMMI method shows promising results.

3 Problem statement

CHM identification relies on the microscopic examination of herbal powders to verify their authenticity. Each herb can be identified by specific cellular structures, termed "feature cells", as illustrated in Figure 2. For example, identifying *Scutellaria baicalensis* requires detecting six distinct feature cells in microscopic images. We believe that the features of herbal microscopic images have a direct relationship with the accuracy of cell recognition. Therefore, we formulate the problem: How can we achieve automated herbal microscopic identification on an insufficient data-level scale and with an unbalanced distribution of sample data?

To systematically approach the problem, we define the terms and notations used in this study: Given the dataset of microscopic images X and their corresponding annotations Y, the objective is to develop a fitted model $f(X)$ that accurately identifies and classifies the feature cells in new, unseen microscopic images of CHM.

Let $X = \{X_1, X_2, \ldots, X_i, \ldots, X_N\}$ represents the set of microscopic images used in the dataset, where each image X_i may contain one or more cell features and N is the total number of images. Associated with each image are target bounding boxes $Y = \{Y_1, Y_2, \ldots, Y_i, \ldots, Y_N\}$, where each Y_i contains one or more bounding boxes indicating the location of feature cells within the image X_i. For each feature cell j in image X_i, the bounding box is represented as $Y_i^j = \left\{ \left[x_{i1}^j, y_{i1}^j\right], \left[x_{i2}^j, y_{i2}^j\right] \right\}$ and $\left[x_{i1}^j, y_{i1}^j\right]$ are the coordinates of the upper-left and lower-right corners of the bounding box, respectively.

4 Methods

This section presents three main modules: the Microscopic Image Data Augmentation (MIDA) Module, the Shallow-Deep Dual Attention (SDDA) Module, and the Multi-scale Inference (MI) Module, as shown in Figure 3. These modules are designed to

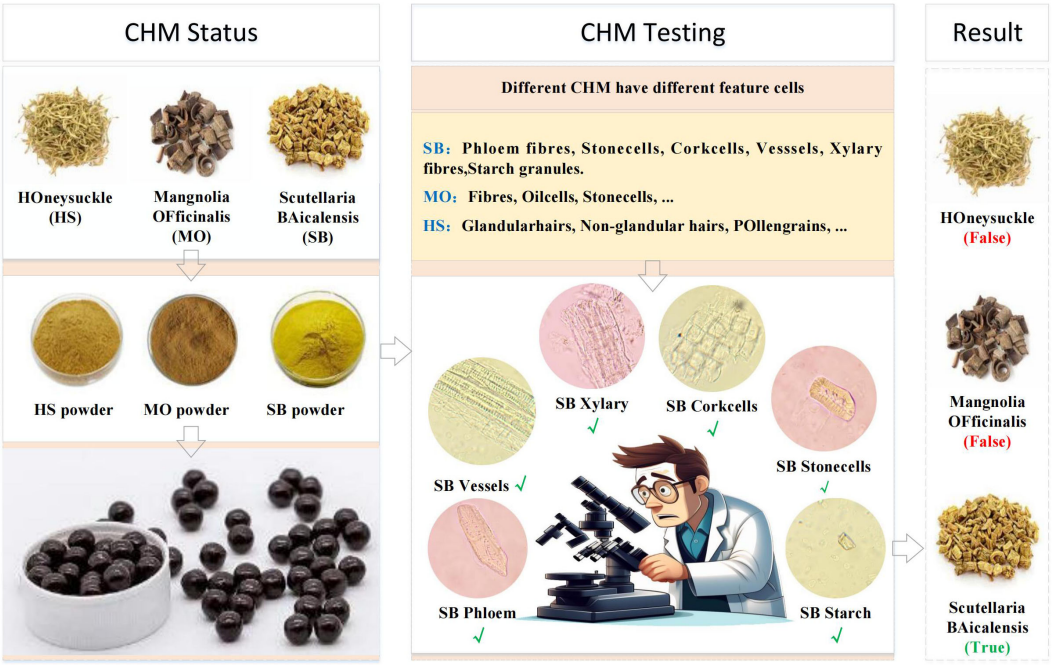


FIGURE 2
Quality testing process of herbal medicine by feature cells.

improve the accuracy and reliability of microscopic image analysis in the study of Chinese herbal medicine.

4.1 Microscopic image data augmentation module

The MIDA module is used to augment and balance the available dataset for training and predicting herbal microscopic images. Our MIDA module associates some of the images with features that are only partially or partially clear, enhancing the representation of specific cell types or features. The detailed steps of MIDA are listed as follows:

1. **Random Selection:** Randomly select two images from the original dataset, such as Figures 4A, B.
2. **Horizontal Segmentation:** Each image is segmented into two halves along the horizontal axis.
3. **Recombination:** Two distinct segments are chosen and stitched together to form four new images from the pool of segmented halves. This ensures that the resultant image differs from the original images (a) and (b), thus enhancing feature representation and diversity.
4. **Augmentation Techniques:** Beyond simple recombination, MIDA incorporates advanced image processing techniques inspired by YOLOv5, such as mirroring, translation, and rotation. These techniques enhance the dataset's diversity further, enabling the model to generalize better across unseen images during inference.

4.2 Shallow-deep dual attention module

The SDDA module addresses several prevalent issues in the microscopic examination of CHM cells, such as the uneven distribution of cells with distinct morphological features and incomplete and blurry cell structures. This module integrates two attention mechanisms: the Shallow Channel Attention Mechanism (SCAM) and the Deep Channel Attention Mechanism (DCAM).

4.2.1 Shallow channel attention mechanism

The core concept of SCAM is to address the problem of uneven cell distribution in CHM cell images by assigning more weights to cell information with significant morphological features while ignoring unimportant feature information, thus improving the image feature recognition rate. The SCAM mechanism consists of three main components: Squeeze, Excitation, and Scale, as shown in Figure 3A. The Squeeze operation performs a global average pooling on the image features to compress the features and reduce the dimensionality. The Excitation operation predicts the importance of each channel using a gating mechanism of the

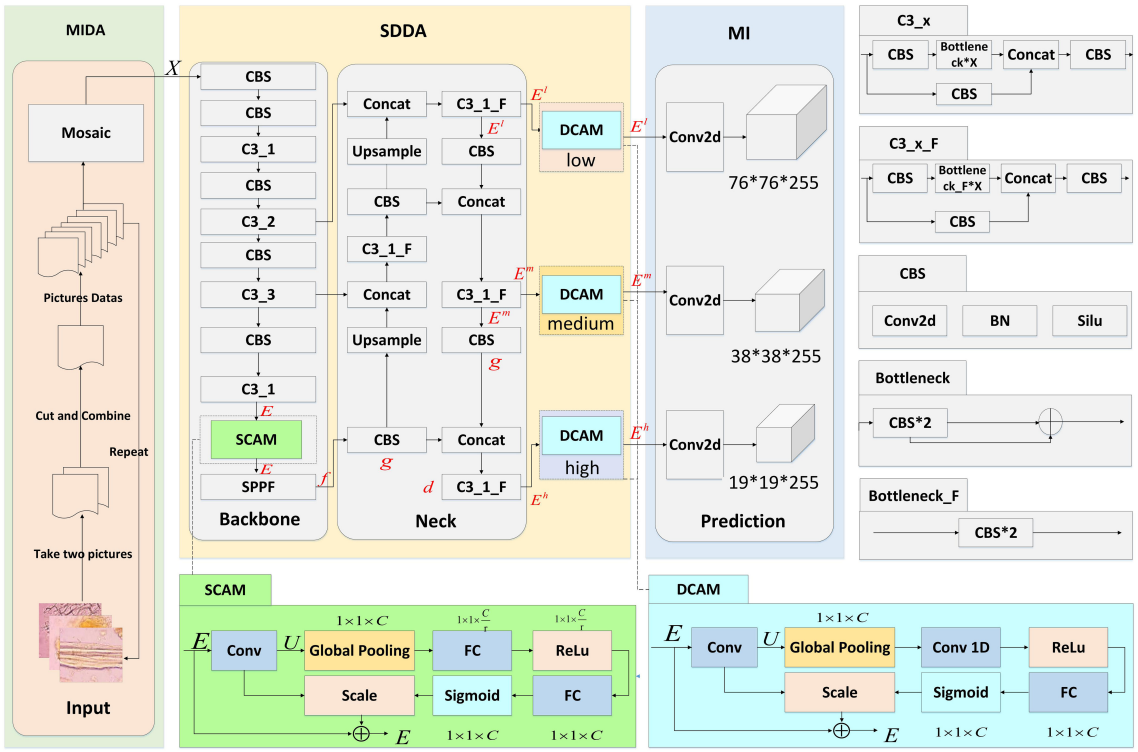


FIGURE 3
Network structure of CHMMI. MIDA is allowed to expand and balance the existing herbal microscopic image dataset. SDDA better captures cell features in the microscopic examination of CHM cells. MI integrates and analyzes features across multiple scales and dimensions intelligently to make final decisions.

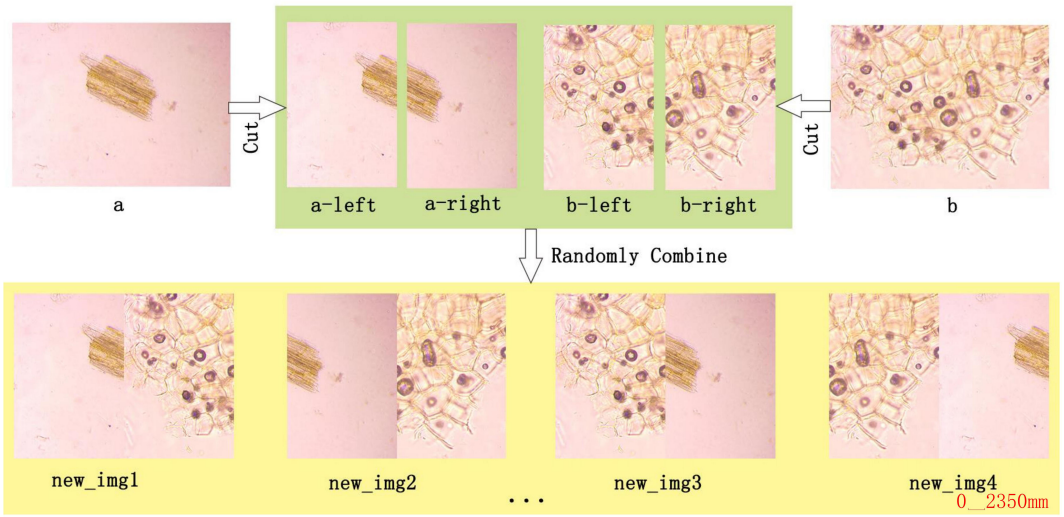


FIGURE 4
Example of MIDA processing. The MIDA module enhances the dataset through image segmentation **(A)** into a-left and a-right, **(B)** into b-left and b-right) and recombination (new_img1, new_img2, new_img3, new_img4, etc.), multiplying the number of images and introducing variability in the dataset.

Sigmoid form, which enables the network model to learn the importance of each channel automatically. Finally, the Scale operation outputs the resulting $1 \times 1 \times C$ real numbers with the original feature images, where C is the number of channels. The specific implementation of the SCAM module is given as follows:

Firstly, the input E is transformed through a series of convolution operations to obtain the features U. Use $V = [v_1, v_2, \ldots, v_C]$ to denote a series of convolution kernels, where v_C denotes the parameters of the cth layer convolution. That is, the output feature $U = [u_1, u_2, \ldots, u_C]$ can be expressed as follows:

$$u_c = v_C * E = \sum_{S=1}^{C'} V_C^S * E^S \tag{1}$$

where $*$ denotes the convolution operation V_C^S denotes the cth convolution kernel of the sth input, E^S denotes the sth input.

Secondly, a global average pooling Zaidi et al. (2022) is performed by the Squeeze operation in the SCAM module for the image features U, intending to compress the image features U. The compressed image feature becomes a one-dimensional real number z, and z is denoted as the residual channel statistic. Suppose the length of the output is set to c, $Z_c = [z_1, z_2 \ldots \ldots, z_c]$, (x, y) denotes the size is the feature of $W * H$, x is the horizontal coordinate and y is the vertical coordinate. That is, the cth element of z can be given by is expressed as:

$$z_c = \frac{1}{H \times W} u_c(x, y) \tag{2}$$

Immediately after, the importance of each channel is predicted by the Excitation operation in the SCAM module using a gating mechanism of the Sigmoid form to obtain the nonlinear relationship between the different channels. Assuming that $W_1 \in R^{\frac{r \times c}{c}}$, $W_2 \in R^{\frac{r}{c \times c}}$ are two different fully connected layers, r is the dimensionality reduction rate when r is small, the global information of the upper layer can be better preserved, but the computational cost will be relatively increased. To balance propagation speed and detection accuracy, refer to SENet Wang and Yoon (2021) and set r to 16. The final output parameter of the Excitation operation is the weight ω of each feature channel, and ω can be expressed as follows:

$$\omega = \sigma(W_2 \times \delta(W_1 \times z)) \tag{3}$$

where, σ is the Sigmoid function, δ is the ReLU activation function.

Finally, the resulting $1 \times 1 \times C$ real numbers are output with the original feature images by the Scale operation in the SCAM module. The formula is listed as follows:

$$\tilde{E}_C = \omega_c u_c \tag{4}$$

where $\tilde{E}_C = [\tilde{e}_1, \tilde{e}_2, \ldots, \tilde{e}_c]$ denotes the product of the corresponding pixel points in the channel between the image feature $u_c \in R^{W \times H}$ and the scalar ω_c. The Scale operation enables the network model to automatically learn the importance of each channel, thus enhancing the recognition of image features.

4.2.2 Deep channel attention mechanism

The DCAM module subtly enhances the feature representation extracted from the cells by adaptively recalibrating the channel feature response to address the CHM's incomplete and blurriness cell structure. The core of DCAM lies in the clever use of the ECA attention mechanism to function at deeper layers of the network, especially at the level where the semantic information is becoming progressively more abstract and where information localization is critical in accuracy. This is particularly beneficial in the context of the CHMMI network structure, where the fusion of features across different dimensions is critical for achieving high detection performance.

In the CHMMI network, the DCAM is strategically positioned within the 'Neck' layer, a critical juncture for feature fusion and refinement. This layer utilizes architectures like the Path Aggregation Network (PAN) and Feature Pyramid Network (FPN) to effectively amalgamate rich locational details from shallow layers with deeper, semantically strong features. The goal is to enhance the upward and lateral flow of information across the network, ensuring that each level receives a balanced mix of depth-specific features. The specific implementation of the DCAM module is given as follows:

The Neck layer has three different dimensional feature outputs towards the Prediction layer, namely low (E^l), medium (E^m), and high (E^h). Taking E^h as an example, E^h can be expressed as follows:

$$E^h = (\widetilde{E} + f + g) \oplus (E^m + g) + d \tag{5}$$

where + denotes the serial processing of features. $\oplus$ denotes tensor stitching, assigning weights to the input features at different levels. f denotes the processing of input features by the SPPF module, g denotes the processing of input features by the CBS module, and d denotes the processing of input features by the C3_1_F Zhu et al. (2021) module.

The DCAM module modifies the conventional channel attention by implementing a three-step process—Squeeze, Convolve, and Scale—tailored to handle multi-dimensional data more effectively:

Firstly, the input E^h is transformed through a series of convolution operations to obtain the feature U^h.

Secondly, the global average pooling of the feature U^h is performed using the Squeeze operation to compress the feature U^h. The feature U^h is compressed into a one-dimensional real number z. For the cth cell in z, the following is calculated:

$$z_c = \frac{1}{H \times W} u_c^h(x, y) \tag{6}$$

Next, to avoid dimensionality reduction, the DCAM module is implemented by a one-dimensional convolution with a convolution kernel size of k cross-channel information interaction. The equation is expressed as follows.

$$\omega = \sigma(C1D_k(z_c)) \tag{7}$$

where, $C1D$ is the one-dimensional convolution Wang et al. (2019). k is the size of the one-dimensional convolution kernel to represent the cross-channel range of interactions. k has a feature mapping relationship with the number of channels c, which can be calculated adaptively by the following equation.

$$k = \psi(C) = \| \log_2(C)/\gamma + b/\gamma \|_{odd} \tag{8}$$

where, $\| n \|_{odd}$ is the closest odd number to n. Referring to the experiments in the literature ECA Wang et al. (2019), γ and b are set to 2 and 1. By mapping ψ, high-dimensional channels have longer interactions, while low-dimensional channels have shorter interactions using nonlinear mappings.

Lastly, the obtained weights and the original feature image are output by the Scale operation in DCAM, and the final residual features are represented as follows.

$$\tilde{E}_C^h = \omega_c \cdot u_c^h \tag{9}$$

Similarly, the low-dimensional residual features $\tilde{E}_C^l$ and the medium-dimensional residual features $\tilde{E}_C^m$ can be obtained

4.3 Multi-scale inference module

The MI module is a crucial component of the CHMMI network and is responsible for effectively detecting and identifying herbal microscopic images. It intelligently integrates and analyzes features across multiple scales and dimensions, enabling the model to capture local and global information from the input images. The module consists of two main components: feature fusion and microscopic recognition.

The feature fusion module integrates features from different scales and channels using a feature pyramid network (FPN), allowing the model to capture local and global information from the input images. This is achieved by up-sampling the feature maps and fusing them with the shallow feature maps, resulting in a richer feature representation that facilitates accurate identification of cellular structures.

The microscopic recognition module is responsible for predicting the presence and location of cellular features in the input images. This is accomplished by applying a combination of convolutional and spatial attention mechanisms to focus on relevant regions of the images. The module outputs a set of bounding boxes and confidence scores for each predicted feature. The input herbal microscopic images are meshed, and if there is a center of the object in the mesh, the mesh is used to predict this object. The prediction of each grid cell includes information on the location of the three object-bounding boxes and a confidence level. An object box corresponds to four position information (x,y,w,h) and one confidence information. Where x and y denote the location of the object's center point, w and h denote the center point's width and height from the object's two sides. Confidence C represents the predicted object box contains two-fold information about the confidence of the object and the accuracy of the prediction of this object box, and the formula is expressed as follows:

$$\mathrm{C} = \mathrm{P_r}(obj) \times IOU_B^A \tag{10}$$

where $IOU = (A \cap B)/(A \cup B)$ A denotes the real box, B denotes the predicted box, IOU_B^A denotes the intersection ratio of A and B. when $\mathrm{P_r}(obj) = 1$, it indicates that there is an object in the image, when $\mathrm{P_r}(obj) = 0$, it indicates that there is no object in the image.

We use Non-maximum Suppression (NMS) Wu et al. (2020) to eliminate redundant prediction boxes and filter out high-quality detection results.

4.4 Training strategy

During the training phase, a three-part loss function is used: object loss, category loss, and confidence loss.

The object loss measures the difference between the predicted and ground-truth bounding boxes. It is calculated using the following equation:

$$l_{obj} = \sum_{i=0}^{S\times S}\sum_{j=0}^{N} I_{ij}^{obj}\left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\right] + \sum_{i=0}^{S\times S}\sum_{j=0}^{n} I_{ij}^{obj}\left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2\right] \quad (11)$$

where $S \times S$ denotes the partitioning of the input image into $S \times S$ mesh grids; N denotes a grid responsible for predicting number of boxes; (x_i, y_i, w_i, h_i) denotes the position information of the real box; $(\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i)$ denotes the position information of the predicted box; I_{ij}^{obj} denotes that the jth prediction box of each of the ith network is responsible for predicting object obj is 1, otherwise is 0.

The category loss measures the difference between the predicted class probabilities and the ground-truth class labels. It is calculated using the following equation:

$$l_{cls} = \sum_{i=0}^{S\times S} I_{ij}^{obj} \sum_{c\in classes}((p_i(c) - \hat{p}_i(c))^2 \quad (12)$$

where, c denotes the number of categories; $p_i(c)$ denotes the probability of the true category; $\hat{p}_i(c)$ denotes the probability of the predicted category.

The confidence loss was calculated using CIOU Zheng et al. (2020), and the equation was expressed as follows:

$$l_{ciou} = \sum_{i=0}^{S\times S}\sum_{j=0}^{n} I_{ij}^{obj}(C_i - \hat{C}_i)^2 + \lambda_{noobj}\sum_{i=0}^{S\times S}\sum_{j=0}^{n} I_{ij}^{noobj}(C_i - \hat{C}_i)^2 \quad (13)$$

where I_{ij}^{noobj} denotes 0 when the jth prediction box of the ith network is not responsible for predicting an object and 1 otherwise. λ_{noobj} is to reduce the confidence loss of the prediction box for the non-existent object obj. In this paper, reference paper Wang et al. (2021) sets λ_{noobj} to 0.5.

The total loss is the weighted sum of the three components of object loss, category loss, and confidence loss, expressed by the following equation.

$$L = \alpha l_{obj} + \beta l_{cls} + \gamma l_{ciou} \quad (14)$$

where, α, β, γ denote the weights of the three loss components respectively.

5 Experiments

To evaluate the performance of the proposed CHMMI method for microscopic image analysis of Chinese herbal medicines, we conducted a series of comprehensive experiments using our custom-built dataset. The experiments were designed to assess the effectiveness of CHMMI for accurately identifying and classifying different types of feature cells presented in the microscopic images of Scutellaria Baicalensis(SB) and Magnolia Officinalis(MO).

5.1 Experiment setup

5.1.1 Datasets

Due to the lack of publicly available datasets for microscopic images of Chinese herbal medicines, we constructed our dataset by preparing slides of powdered SB and MO. We used a Nikon E200 electron microscope with a 40/0.65 objective and the software Labeling to label the microscopic image of Chinese medicine feature cells. The resulting dataset consists of 11,060 microscopic images containing 12,840 labeled instances of nine distinct types of feature cells. The distribution of images and labeled instances for each feature cell type is shown in Table 1. These feature cells include Fibers, Stone cells, and Oil cells for MO, Phloem fibers, Stone cells, Corkcells, Vessels, Xylary fibers, and Starch granules for SB. Figure 5 presents sample images of the nine feature cell types.

To ensure a robust evaluation of the proposed CHMMI method, the dataset was partitioned into training and test sets following an 8:2 ratio. Furthermore, to rigorously assess the effectiveness of the CHMMI method and its individual components, we conducted a five-fold cross-validation experiment on the training dataset. This involved splitting the training data into five non-overlapping subsets. Each subset was then used in turn as a validation set while the remaining four subsets were combined for training. Applying each of the five trained models to the test set, generating five sets of prediction results for every test sample. Implementing a voting mechanism across the five predictions to determine the final predicted label for each test sample.

5.1.2 Implementation details

We implemented the CHMMI method based on the PyTorch deep learning framework YOLOv5, training the model on an NVIDIA GeForce RTX 3090 GPU with 24GB memory. The model has trained 100 epochs with the Adam optimizer, using a

TABLE 1 Statistics of Chinese medicine microscopic image annotation dataset.

Dateset	MO			SB					
	Fibers	Stonecells	Oilcells	Phloem	Stonecells	Corkcells	Vessels	Xylary	Starch
Images	7555	1662	576	304	156	550	229	13	15
Boxes	9080	1726	644	353	171	580	257	13	16
Images Total	9793			1267					
Boxes Total	11450			1390					

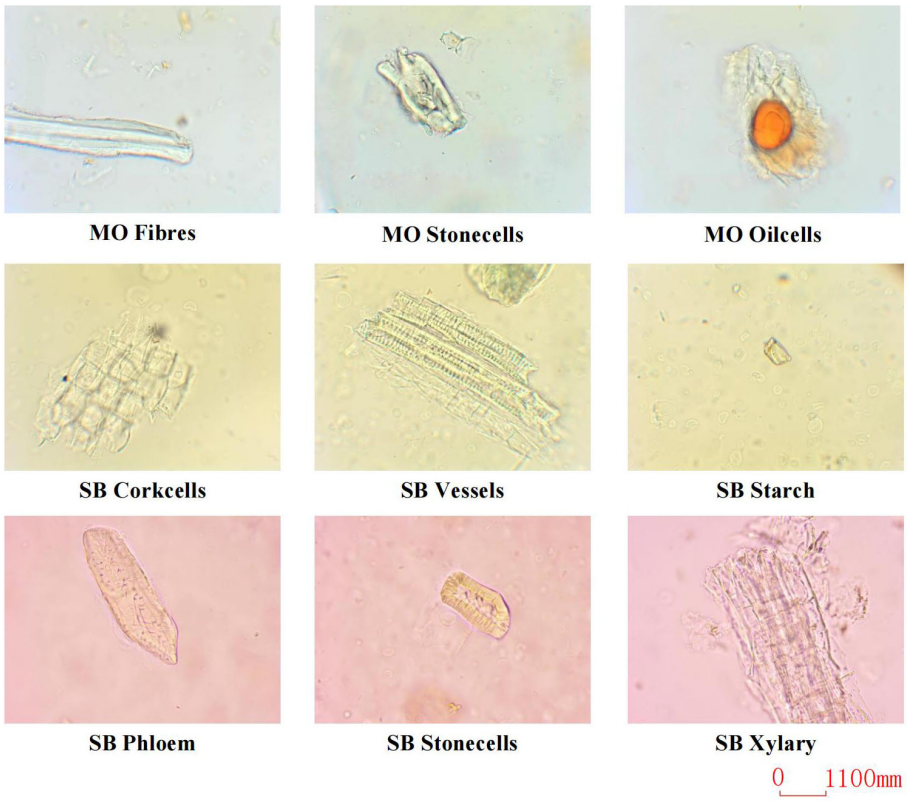


FIGURE 5
Sample images of the 9 cell types.

learning rate of 0.001 and a batch size of 16. In our implementation, we adopted a three-scale anchor system: P3/8, P4/16, and P5/32. Specifically, the P3/8 scale anchors are designed to detect small targets, the P4/16 anchors are geared towards medium-sized targets, and the P5/32 anchors aim to detect large targets. This hierarchical structure ensures comprehensive coverage of the target size spectrum within the microscopic images.

5.1.3 Evaluation metrics

To evaluate the CHMMI algorithm's performance comprehensively, we select four evaluation metrics: precision, Recall, Average Precision (AP) curve, Mean Average Precision (MAP), and Matthews Correlation Coefficient(MCC). These metrics evaluate the algorithm's ability to accurately identify and classify the feature cells present in microscopic images.

Precision denotes the ratio of true positive cases predicted to be true to all predicted positive cases Liu et al. (2018). It is calculated as:

$$precision = TP/(TP + FP) \tag{15}$$

where TP denotes that the predicted value is the same as the true value, and the predicted value is a positive sample; FP denotes that the predicted value is different from the true value, and the predicted value is a positive sample.

Recall denotes the ratio of true positive cases predicted to be true to all true positive cases. It is calculated as:

$$recall = TP/(TP + FN) \tag{16}$$

where FN denotes that the predicted value is not the same as the true value and the predicted value is a negative sample.

The AP curve is the area surrounded by the curve in two dimensions: Precision and Recall. Usually, Precision is higher when Recall is lower and lower when Recall is higher. That is, the larger the AP curve, the better the model's performance.

MAP is a comprehensive evaluation metric focusing on sequence weights. It has become one of the most important practical metrics for image recognition problems in recent years. mAP@.5 indicates that the average AP of all images under each category is calculated at IoU=0.5, and the higher the value of mAP, the better the model's performance.

MCC is an effective and comprehensive evaluation metric widely used in tasks with unbalanced sample categories, such as defect detection. It is particularly suitable for performance evaluation of binary classification models because it integrates the predictions of the model's TP, TN, FP, and FN and is thus more robust than other metrics in evaluating the model's ability to distinguish between positive and negative samples. It is calculated as:

$$MCC = \frac{TP \times TN - FP \times FN}{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)} \tag{17}$$

5.2 Comparisons with state-of-the-art methods

To assess the efficacy of our proposed CHMMI, we compared it with several widely adopted state-of-the-art image recognition algorithms. Specifically, we benchmarked our method against YOLOv5 Zhu et al. (2021), SSD Liu et al. (2016), Faster R-CNN Khan et al. (2022), ResNet He et al. (2016), FINet Zhang et al. (2022), YOLOT Liu et al. (2024), and an improved version of YOLOv5 (Improved_yolov5) Hu et al. (2024). These algorithms represent diverse architectural paradigms and have demonstrated exceptional performance across various computer vision tasks, providing a robust baseline for comparative analysis.

Table 2 presents the quantitative results of the comparative analysis. As the table shows, our proposed CHMMI approach outperformed all the state-of-the-art methods across all four evaluation metrics. Specifically, CHMMI achieved an impressive AP of 0.841, surpassing the second-best performer, YOLOT, by a significant margin of 0.013. Furthermore, CHMMI attained the highest mAP@.5 of 0.887, outperforming the closest competitor, Improved_yolov5, by 0.006. CHMMI demonstrated its superiority in the most challenging mAP@.5:.95 metric, achieving a remarkable score of 0.551, 0.016 higher than the second-best performer, Improved yolov5. CHMMI performs excellently on the comprehensive evaluation metric MCC, achieving an outstanding score of 0.898, surpassing the second-place YOLOT by 0.011.

To provide a visual representation of the performance difference, we plot the Receiver Operating Characteristic (ROC) curves for both YOLOv5 and CHMMI, using thresholds ranging from 0.1 to 1.0. Figure 6 illustrates these curves, revealing a higher Area Under the Curve (AUC) value for CHMMI (0.83) compared to YOLOv5 (0.74), further confirming CHMMI's superior performance.

In comparison to other CNN models, the CHMMI model has several advantages. For example, the YOLOv5 model uses a single-stage detection approach, which may not be suitable for handling the complexity of microscopic images. The SSD model uses a multi-scale feature fusion approach, but it may not be able to capture the contextual information of cells as effectively as the CHMMI model.

TABLE 2 Comparisons with state-of-the-art methods.

Method	AP	mAP@.5	mAP@.5:.95	MCC
YOLOv5 Zhu et al. (2021)	0.803	0.843	0.511	0.753
SSD Liu et al. (2016)	0.781	0.819	0.532	0.798
Faster R-CNN Khan et al. (2022)	0.629	0.757	0.521	0.647
ResNet He et al. (2016)	0.712	0.823	0.513	0.695
FINet Zhang et al. (2022)	0.637	0.869	0.524	0.823
YOLOT Liu et al. (2024)	0.828	0.877	0.531	0.887
Improved_yolov5 Hu et al. (2024)	0.807	0.881	0.535	0.873
CHMMI	**0.841**	**0.887**	**0.551**	**0.898**

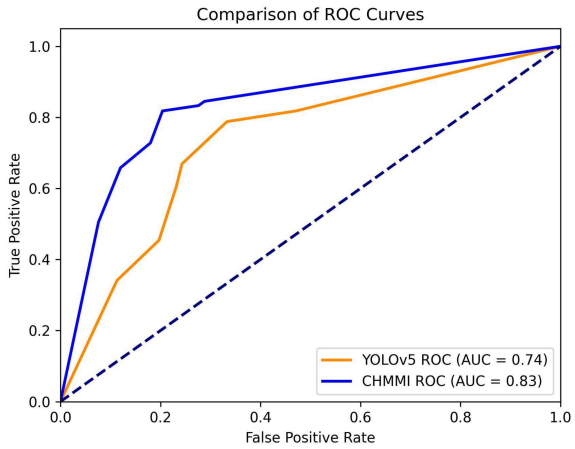


FIGURE 6
ROC Curves for models YOLOv5 and CHMMI.

The Faster R-CNN model uses a two-stage detection approach, but it may not be able to handle the issues of uneven cell distribution and incomplete and blurry cell structures as effectively as the CHMMI model. The ResNet model uses a residual learning approach, but it may not be able to capture the complex relationships between cells as effectively as the CHMMI model. These results underscore the efficacy of our proposed approach in accurately detecting and localizing objects under varying degrees of occlusion and overlap.

In addition, we show the detection results of our CHMMI model, as shown in Figure 7. As can be seen from the figure, CHMMI can not only identify different categories of Chinese medicine feature cells but also accurately detect incomplete and blurriness cell structures.

5.3 Ablation studies

5.3.1 Effectiveness of different modules

To assess the impact of each proposed module, we conducted a comprehensive set of ablation studies. Specifically, we systematically included or excluded the Microscopic Image Data Augmentation (MIDA), Shallow Channel Attention Module (SCAM), and Deep Channel Attention Module (DCAM) from our model and evaluated its performance. We employ a five-fold cross-validation strategy during the training phase to ensure a robust evaluation and mitigate the potential impact of data partitioning bias. The training dataset is divided into five non-overlapping subsets. For each fold, one subset is held out for validation, while the remaining four subsets are used for training. This process results in five distinct sets of model weights (M1, M2, M3, M4, and M5). During the testing phase, each of the five trained models (M1 to M5) is independently applied to the test set. This generates five sets of prediction results for each test sample. To combine these predictions, we implement a voting mechanism. The final predicted label for each test sample is determined by selecting the category that received the most votes across the five individual model predictions.

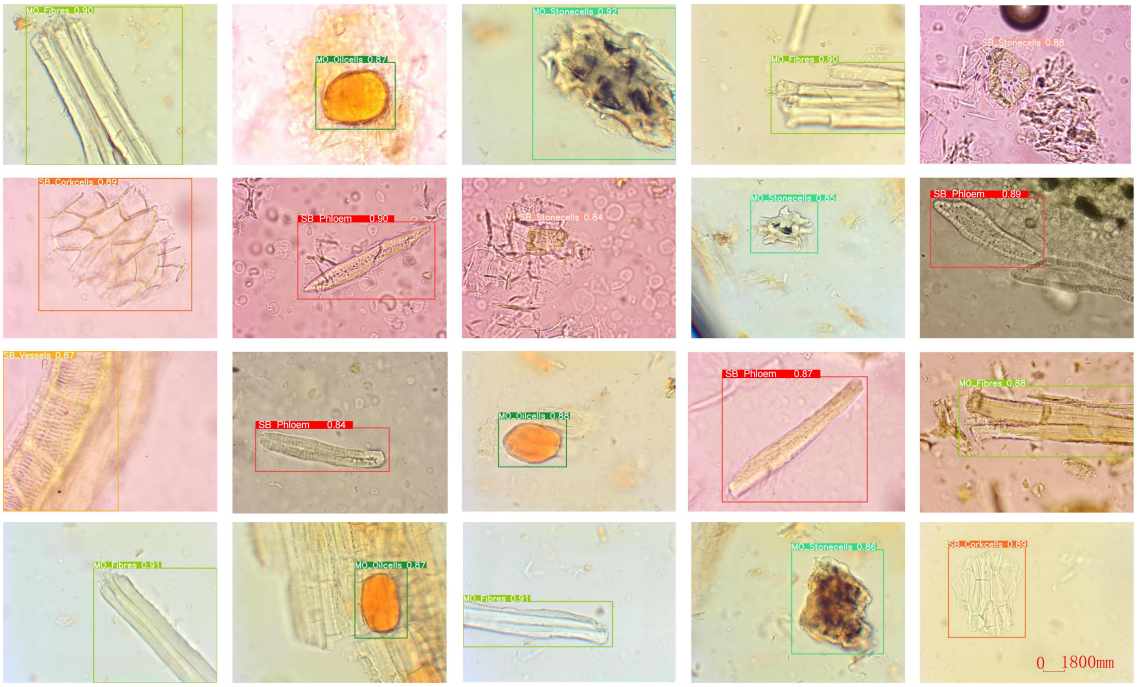


FIGURE 7
Visualization of the detection results using CHMMI.

The data in Table 3 demonstrates a clear trend of increasing model performance as more modules are incorporated. The inclusion of all three modules (MIDA, SCAM, and DCAM) results in the highest precision (P), recall(R), mAP@.5, and mAP@.5:.95. This suggests a synergistic effect between data augmentation, shallow feature attention, and deep feature attention mechanisms. The consistent improvement across all evaluation metrics indicates that the SDDA is vital in enhancing object detection accuracy. Furthermore, the results show that including MIDA alone significantly improves the model's performance compared to using SCAM or DCAM individually. This highlights the importance of data augmentation in improving the model's ability to detect objects in microscopic images. Integrating MIDA, SCAM, and DCAM leads to the most significant improvement in object detection accuracy, emphasizing the importance of combining data augmentation, shallow feature attention, and deep feature attention mechanisms.

5.3.2 Effectiveness of microscopic image data augmentation module

The MIDA module plays a crucial role in enhancing the performance of our model by addressing the challenges posed by limited and imbalanced datasets of herbal microscopic images. It is particularly effective when dealing with images that only partially demonstrate certain features or cell types. By enhancing the representation of these specific attributes, we improve our data's overall quality and diversity.

To evaluate the effectiveness of the MIDA module, we conducted extensive experiments by training our model with and without the augmented dataset generated by MIDA. The results, as shown in Table 4, demonstrate the significant impact of MIDA on the model's performance metrics. As evident from the table, including the MIDA module resulted in significant improvements across all performance metrics. The precision and recall values increased from 0.831 and 0.808, respectively, without MIDA to 0.854 and 0.835 with MIDA, indicating a substantial enhancement in the model's ability to accurately classify cell types and features while minimizing false positives and false negatives. Moreover, the mean Average Precision (mAP) values, which comprehensively evaluate the model's performance across different confidence thresholds, also exhibited notable improvements. The mAP@.5, which measures the average precision at an intersection-over-union (IoU) threshold of 0.5, increased from 0.843 without MIDA to 0.855 with MIDA. Similarly, the mAP@.5:.95, which averages the precision values across IoU thresholds ranging from 0.5 to 0.95, improved from 0.511 to 0.522 with MIDA.

5.3.3 Effectiveness of shallow-deep dual attention module

The SDDA module represents a significant advancement in addressing the complex challenges inherent in the microscopic examination of CHM cells. This module integrates the strengths of both shallow and deep feature representations within the model. The heatmaps in Figure 8 provide a visual representation of the impact of the SDDA module. When only the SCAM is used, the model tends to focus on less relevant areas, potentially discarding crucial feature information. Conversely, when only the DCAM is used, the attention becomes scattered, hindering the model's ability to focus on the

10.3389/fpls.2024.1442968

TABLE 3 Experimental results using SCAM only, DCAM only, and SCAM+DCAM.

MIDA	SCAM	DCAM	Model	P	R	mAP@.5	mAP@.5:.95
			M1	0.874	0.789	0.834	0.507
			M2	0.804	0.809	0.845	0.508
			M3	0.811	0.763	0.839	0.509
			M4	0.821	0.787	0.812	0.491
			M5	0.793	0.830	0.844	0.511
			vote	0.831	0.808	0.843	0.511
			M1	0.883	0.791	0.842	0.520
			M2	0.845	0.811	0.864	0.522
✓			M3	0.841	0.793	0.840	0.512
			M4	0.833	0.801	0.818	0.497
			M5	0.799	0.837	0.858	0.517
			vote	0.854	0.835	0.855	0.522
			M1	0.884	0.811	0.869	0.521
			M2	0.821	0.809	0.863	0.518
	✓		M3	0.828	0.849	0.859	0.516
			M4	0.825	0.823	0.853	0.497
			M5	0.805	0.838	0.850	0.514
			vote	0.851	0.831	0.861	0.522
			M1	0.875	0.811	0.841	0.509
			M2	0.812	0.831	0.852	0.520
		✓	M3	0.859	0.798	0.855	0.530
			M4	0.825	0.812	0.847	0.493
			M5	0.831	0.846	0.860	0.517
			vote	0.856	0.835	0.868	0.528
			M1	0.891	0.825	0.878	0.530
			M2	0.859	0.815	0.867	0.522
✓	✓		M3	0.849	0.849	0.868	0.519
			M4	0.842	0.835	0.858	0.504
			M5	0.824	0.841	0.861	0.519
			vote	0.876	0.844	0.881	0.532
			M1	0.902	0.814	0.875	0.529
			M2	0.865	0.838	0.877	0.525
✓		✓	M3	0.864	0.823	0.857	0.535
			M4	0.838	0.826	0.850	0.506
			M5	0.835	0.847	0.869	0.527
			vote	0.868	0.845	0.879	0.537
			M1	0.893	0.821	0.877	0.524
			M2	0.863	0.848	0.861	0.538
	✓	✓	M3	0.866	0.848	0.860	0.537

(Continued)

10.3389/fpls.2024.1442968

TABLE 3 Continued

MIDA	SCAM	DCAM	Model	P	R	mAP@.5	mAP@.5:.95
			M4	0.858	0.839	0.849	0.518
			M5	0.845	0.857	0.865	0.532
			vote	0.873	0.854	0.879	0.541
			M1	0.917	0.838	0.885	0.531
			M2	0.897	0.850	0.875	0.549
✓	✓	✓	M3	0.874	0.856	0.883	0.546
			M4	0.871	0.849	0.866	0.534
			M5	0.861	0.871	0.882	0.543
			vote	0.905	0.871	0.887	0.551

The symbol ✓ indicates that the module has been selected.

TABLE 4 Experimental results with and without microscopic image data augmentation module.

Module Name	P	R	mAP@.5	mAP@.5:.95
w/o MIDA	0.831	0.808	0.843	0.511
w MIDA	0.854	0.835	0.855	0.522

foreground regions of interest precisely. However, the simultaneous use of both SCAM and DCAM results in a focused and accurate attention map, highlighting the model's ability to detect cells with diverse morphological features, even incomplete or blurry.

Overall, the Shallow-Deep Dual Attention module effectively enhances the CHMMI model's ability to accurately detect and

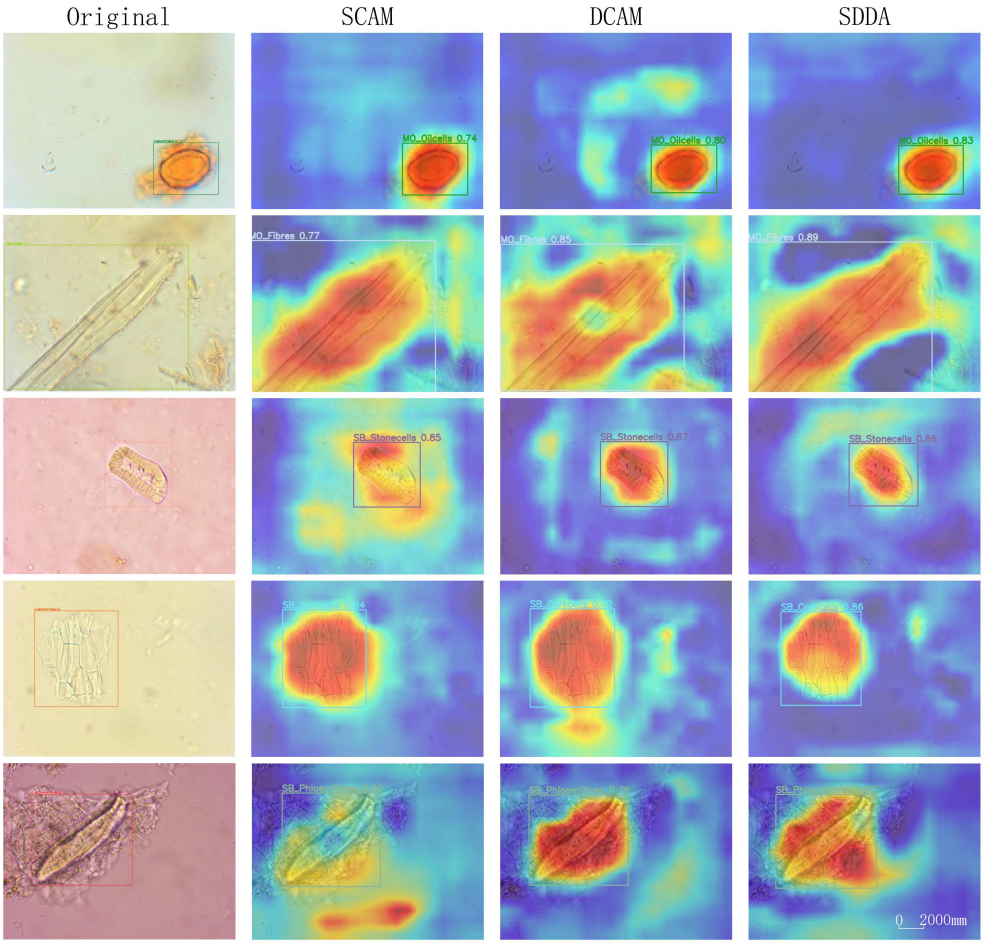


FIGURE 8
Heatmap examples using SCAM only, DCAM only, and SDDA.

analyze CHM cells by addressing the limitations of individual attention mechanisms. The combination of SCAM and DCAM allows the model to focus on relevant features and handle various challenges in microscopic cell examination, leading to improved performance and more accurate results.

6 Conclusion

Traditional Chinese Herbal Medicine (CHM) identification methodologies, such as original plant identification, character identification, microscopic identification, and physical and chemical identification, have long been relied upon but present significant challenges regarding labor intensity, subjectivity, and limitations in distinguishing similar substances. The rapid growth of the CHM market and the need for modernization call for more advanced and reliable identification techniques. Developing deep learning-based methods, particularly artificial neural networks, offers a promising solution to automate CHM microscopic identification. Our proposed methodology, CHMMI, addresses key challenges in automated CHM identification by combining segmentation methods with data augmentation and integrating attention mechanisms to enhance feature recognition and model accuracy. By effectively capturing small and uneven features and addressing issues with incomplete and blurry cell structures in CHM samples, CHMMI outperforms existing state-of-the-art approaches in experimental comparisons. CHMMI can be integrated into the quality control processes of CHM manufacturers. Automating the identification of herbal components can ensure consistency in raw material selection, detect adulterants or contaminants, and maintain the purity of herbal preparations. This application could significantly improve product quality and safety, potentially reducing the risk of adverse reactions due to misidentified or contaminated herbs. CHMMI can accelerate the discovery of new bioactive compounds from traditional herbal medicines in pharmaceutical research. By quickly and accurately identifying cellular structures, researchers can more efficiently screen large numbers of herbal samples, potentially leading to the development of novel drugs or therapies.

While CHMMI shows superior performance, understanding why certain features are prioritized over others could be beneficial. Future research will focus on developing or integrating explainable AI techniques to provide insights into the model's decision-making process, enhancing trust and acceptance in clinical and regulatory settings.

Data availability statement

The data analyzed in this study was obtained from Wuzhou University and Guangxi Wuzhou Zhongheng Group Co., Ltd. The following licenses/restrictions apply: Data is available for academic research use only and may not be redistributed or used for commercial purposes without prior approval. Requests to access these datasets should be directed to Dr Guangyao Pang via email at pangguangyao@gmail.com.

Ethics statement

Written informed consent was obtained from the individual(s) for the publication of any identifiable images or data included in this article.

Author contributions

XZ: Conceptualization, Data curation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. GP: Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Validation, Writing – review & editing. XH: Supervision, Validation, Writing – review & editing. YC: Data curation, Methodology, Validation, Writing – review & editing. ZY: Project administration, 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. This work was supported by the National Natural Science Foundation of China (Grant Nos. 62262059), Funding Scheme for Innovation and Technology Promotion of FDCT (Grant Nos. 0009/2024/ITP1), the Natural Science Foundation of Guangxi Province (Grant Nos.2021JJA170178), the Science and Technology plan project (Grant Nos. 2022B02030), the Guangxi Scholarship Fund of Department of Education of Guangxi Zhuang Autonomous Region the People's Republic of 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

References

Chen, X., Wang, X., and Qiu, Y. (2022). Recent advances and clinical applications of deep learning in medical image analysis. *Med. Image Anal.* 79, 102444. doi: 10.1016/j.media.2022.102444

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.

Hu, H., Li, Z., He, Z., Wang, L., Cao, S., and Du, W. (2024). Road surface crack detection method based on improved yolov5 and vehicle-mounted images. *Measurement* 229, 114443. doi: 10.1016/j.measurement.2024.114443

Ichim, M. C., Häser, A., and Nick, P. (2020). Microscopic authentication of commercial herbal products in the globalized market: Potential and limitations. *Front. Pharmacol.* 11, 876. doi: 10.3389/fphar.2020.00876

Jiang, P., Ergu, D., and Ma, B. (2022). A review of yolo algorithm developments. *Proc. Comput. Sci.* 199, 1066–1073. doi: 10.1016/j.procs.2022.01.135

Khan, S., Naseer, M., and Shah, M. (2022). Transformers in vision: A survey. *ACM computing surveys (CSUR)* 54, 1–41. doi: 10.1145/3505244

Kim, J.-H., Kim, N., Park, Y. W., and Won, C. S. (2022). Object detection and classification based on yolo-v5 with improved maritime dataset. *J. Mar. Sci. Eng.* 10, 377. doi: 10.3390/jmse10030377

Liu, W., Anguelov, D., and Berg, A. C. (2016). "Ssd: Single shot multibox detector," in *European conference on computer vision* (Amsterdam, The Netherlands: Springer International Publishing), 21–37.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., et al. (2018). Deep learning for generic object detection: A survey. *Int. J. Comput. Vis.* 128, 261–318.

Liu, D., Tian, Y., Xu, Y., Zhao, W., Pan, X., Ji, X., et al. (2024). Yolot: Multi-scale and diverse tire sidewall text region detection based on you-only-look-once (yolov5). *Cogn. Robot* 4, 74–87. doi: 10.1016/j.cogr.2024.03.001

Peng, W.-Y., and Tsa, T.-H. (2020). Scanning electron microscopy and liquid chromatography for physical and chemical inspection of industrial pharmaceutical traditional Chinese herbal medicine. *ACS omega* 5, 11563–11569. doi: 10.1021/acsomega.0c00809

Li, S. (1999). "Compendium of Materia Medica," in *Chinese materia medica* (Liaoning, China: Liaoning Nationalities Publishing House).

Sun, X., Wu, P., and Hoi, S. C. (2018). Face detection using deep learning: An improved faster rcnn approach. *Neurocomputing* 299, 42–50. doi: 10.1016/j.neucom.2018.03.030

Thongkhao, K., Pongkittiphan, V., Phadungcharoen, T., Tungphatthong, C., Urumarudappa, S. K. J., Pengsuparp, T., et al. (2020). Differentiation of cyanthillium cinereum, a smoking cessation herb, from its adulterant Emilia sonchifolia using macroscopic and microscopic examination, hptlc profiles and dna barcodes. *Sci. Rep.* 10, 1–11. doi: 10.1038/s41598-020-71702-7

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2021). "Scaled-yolov4: Scaling cross stage partial network," in *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*. (Nashville, TN, USA: IEEE), 13029–13038.

Wang, Y., Hao, C., and Yuan, Y. (2020a). Microscopic image identification for small-sample Chinese medicinal materials powder based on deep learning. *J. Comput. Appl.* 40, 1301–1308.

Wang, N., Lu, W., and Li, R. (2017). Feature extraction and image recognition of achyranthes bidentata and cyathula officinalis. *China Pharm.* 28, 1670–1673.

Wang, Y., Wang, C., and Wei, S. (2019). Automatic ship detection based on retinanet using multi-resolution gaofen-3 imagery. *Remote Sens.* 11, 531. doi: 10.3390/rs11050531

Wang, Q., Wu, B., 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, Washington, USA: IEEE), 11534–11542. doi: 10.1109/CVPR42600.2020

Wang, Y., Yao, Y., and Yuan, Y. (2020b). Research on microscopic image recognition of Chinese medicinal materials powder based on improved dynamic relu and attention mechanism model. *Appl. Res. Comput.* 38, 2861–2865. doi: 10.19734/j.issn.1001-3695.2020.11.042

Wang, L., and Yoon, K.-J. (2021). Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE Trans. Pattern Anal. Mach. Intel.* 44, 3048–3068. doi: 10.1109/TPAMI.2021.3055564

Wu, S., Li, X., and Wang, X. (2020). Iou-aware single-stage object detector for accurate localization. *Image Vision Comput.* 97, 103911. doi: 10.1016/j.imavis.2020.103911

Yan, B., Lei, X., Liu, Z., Fan, P., and Yang, F. (2021). A real-time apple targets detection method for picking robot based on improved yolov5. *Remote Sens.* 13, 1619. doi: 10.3390/rs13091619

Ye, L., Xue, Y., and Xiao, T. (2014). *In situ* investigation to three dimensional structures of chinese medicines seeds. *Zhongguo Zhong yao za zhi= Zhongguo Zhongyao Zazhi= China J. Chin. Materia Med.* 39, 2619–2623.

Yin, L., Zhou, J., and Shao, Q. (2019). A review of the application of near-infrared spectroscopy to rare traditional Chinese medicine. *Spectrochim. Acta Part A: Mol. Biomol. Spectrosc.* 221, 117208. doi: 10.1016/j.saa.2019.117208

Ying, C., Wang, D., and Hu, Q. (2012). Microscopic identification and comparison of the two crude medicines from buddleja(b. lindleyana and b. albiflora). *Lishizhen Med. Materia Med. Res.* 23, 706–708.

Yu, Z., Shen, Y., and Shen, C. (2021). A real-time detection approach for bridge cracks based on yolov4-fpm. *Autom. Construct.* 122, 103514. doi: 10.1016/j.autcon.2020.103514

Zaidi, S. S. A., Ansari, M. S., and Lee, B. (2022). A survey of modern deep learning based object detection models. *Digit. Signal Process.* 126, 103514. doi: 10.1016/j.dsp.2022.103514

Zhai, S., Shang, D., and Dong, S. (2020). Df-ssd: An improved ssd object detection algorithm based on densenet and feature fusion. *IEEE Access* 8, 24344–24357. doi: 10.1109/Access.6287639

Zhang, Z.-D., Zhang, B., Lan, Z.-C., Liu, H.-C., Li, D.-Y., Pei, L., et al. (2022). Finet: An insulator dataset and detection benchmark based on synthetic fog and improved yolov5. *IEEE Trans. Instrument. Measure.* 71, 1–8. doi: 10.1109/TIM.2022.3194909

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., and Ren, D. (2020). "Distance-iou loss: Faster and better learning for bounding box regression," in *Proceedings of the AAAI conference on artificial intelligence* (Hilton New York Midtown, New York, New York, USA: AAAI), Vol. 34. 12993–13000.

Zhu, X., Lyu, S., and Zhao, Q. (2021). "Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. (Nashville, TN, USA: IEEE), 2778–2788.

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 16 January 2025
DOI 10.3389/fpls.2024.1457360

OPEN ACCESS

EDITED BY
Mohsen Yoosefzadeh Najafabadi,
University of Guelph, Canada

REVIEWED BY
Senthil Ganesh R.,
Sri Krishna College of Engineering & Technology, India
Rima Tri Wahyuningrum,
Trunojoyo University, Indonesia

*CORRESPONDENCE
Lu Gao
02430@zjhu.edu.cn
Shanlin Ma
MSL205@126.com

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

RECEIVED 30 June 2024
ACCEPTED 16 December 2024
PUBLISHED 16 January 2025

CITATION
Jia L, Wang T, Li X, Gao L, Yu Q, Zhang X and Ma S (2025) DFMA: an improved DeepLabv3+ based on FasterNet, multi-receptive field, and attention mechanism for high-throughput phenotyping of seedlings.
Front. Plant Sci. 15:1457360.
doi: 10.3389/fpls.2024.1457360

DFMA: an improved DeepLabv3+ based on FasterNet, multi-receptive field, and attention mechanism for high-throughput phenotyping of seedlings

Liangquan Jia[1†], Tao Wang[1†], Xiangge Li[1], Lu Gao[1*], Qiangguo Yu[2], Xincheng Zhang[3] and Shanlin Ma[3*]

[1]School of Information Engineering, Huzhou University, Huzhou, China, [2]School of Electronic Information Engineering, Huzhou College, Huzhou, China, [3]Institute of Crop Science, Huzhou Academy of Agriculture Sciences, Huzhou, China

With the rapid advancement of plant phenotyping research, understanding plant genetic information and growth trends has become crucial. Measuring seedling length is a key criterion for assessing seed viability, but traditional ruler-based methods are time-consuming and labor-intensive. To address these limitations, we propose an efficient deep learning approach to enhance plant seedling phenotyping analysis. We improved the DeepLabv3+ model, naming it DFMA, and introduced a novel ASPP structure, PSPA-ASPP. On our self-constructed rice seedling dataset, the model achieved a mean Intersection over Union (mIoU) of 81.72%. On publicly available datasets, including Arabidopsis thaliana, Brachypodium distachyon, and Sinapis alba, detection scores reached 87.69%, 91.07%, and 66.44%, respectively, outperforming existing models. The model generates detailed segmentation masks, capturing structures such as the embryonic shoot, axis, and root, while a seedling length measurement algorithm provides precise parameters for component development. This approach offers a comprehensive, automated solution, improving phenotyping analysis efficiency and addressing the challenges of traditional methods.

KEYWORDS

plant seedlings, deep learning, plant seedling phenotyping analysis, DeepLabv3+, DFMA

1 Introduction

"High-Throughput Phenotyping" is a method for rapidly and automatically acquiring and analyzing large volumes of phenotypic data from plant or biological samples. This approach utilizes imaging technology, sensors, computer vision, and machine learning to collect extensive data without disrupting sample growth, thus revealing growth

characteristics, health status, and physiological changes of the organisms. This technique is particularly applicable in agriculture and plant sciences, enabling efficient evaluation of different genotypes under various environmental conditions, and providing essential data to support crop improvement and breeding programs. In recent years, plant phenotyping has emerged as a rapidly advancing, data-intensive field (Zhao et al., 2019; Yang et al., 2020). Studying plant phenotypes allows for a deeper understanding of genetic information (Richard et al., 2015; Holman et al., 2016) and the growth trends of plants. When it comes to monitoring the growth of plant seedlings, phenotypic analysis of seedlings becomes particularly crucial. Assessing various aspects of seedling development often requires the measurement of specific physical dimensions, with the length of the hypocotyl being a key phenotypic trait for monitoring and quantifying different responses (Dobos et al., 2019). Hypocotyl cells are formed during embryogenesis and undergo several rounds of cell division to develop. During seedling growth, the length of the hypocotyl is no longer determined by cell division but rather by the elongation of hypocotyl cells (Gendreau et al., 1997). Phenotypic analysis of the root system, known as Root System Architecture (RSA), is also a vital indicator for assessing seedling development. RSA refers to the spatial arrangement of the root system and its components (Lynch, 1995), and its functions include water and nutrient absorption, storage, as well as anchoring and facilitation of plant-microbe interactions, such as nodule formation in nitrogen-fixing crops. Although these features may not be readily apparent during plant growth, they have a crucial impact on overall plant performance, particularly for non-tuberous or rhizomatous crops (York et al., 2015). Root system architecture is closely related to a plant's competitive advantage in the environment, including nutrient acquisition (Lynch, 1995; MansChadi et al., 2014), drought tolerance (Ribaut, 2006; Comas et al., 2013; Fenta et al., 2014; Wade et al., 2015), waterlogging tolerance (VanToai et al., 2001), and lodging resistance (Guingo et al., 1998).

In the field of seedling phenotypic analysis, seed viability testing, and seed germination experiments, parameters such as germination rate, seedling length, and growth rate are frequently measured. For instance, Wang Binbin et al. (Wang and Wu, 2022) conducted a study on the impact of extracellular polysaccharides from lactic acid bacteria on the germination and stress tolerance of japonica rice seeds. They performed statistical analysis on parameters such as germination potential, germination rate, root length, and shoot length of japonica rice seeds incubated in different culture solutions at a constant temperature for 7 days. However, this process required a significant amount of manual measurements. Similarly, Jiang Yuting et al. (Jiang et al., 2022) investigated the effects of different particle sizes and concentrations of polystyrene microplastics (PS-MPs) on the germination and seedling growth of sorghum seeds to understand the material's impact on plants. These experiments also necessitated accurate measurements of germination, root length, and shoot length. Nevertheless, traditional manual measurement methods are no longer adequate to meet the demands of modern agriculture for efficient, precise, and automated measurements. Particularly in seed germination experiments, accurately measuring shoot length has become an urgent issue. Currently, there is a relatively limited body of research on methods for measuring shoot length during the seed germination stage, and there is no widely accepted automated detection method for measuring root or shoot length during seed germination.

In recent years, with the continuous progress of artificial intelligence, computer vision, and other technologies, more and more researchers have begun to explore how to utilize advanced technologies such as deep learning to solve problems in the field of agricultural detection. These studies have proposed a series of deep learning-based methods for image semantic segmentation and target detection to address the needs of modern agriculture. For example, Marset et al. (Marset et al., 2021) proposed a grape bud detection method based on the Fully Convolutional Network Mobile Network architecture (FCN-MN), which achieved improvements in segmentation, correspondence recognition, and localization, and realized the detection of the number of grape buds, bud area, and internode length. On the other hand, Yaying Shi et al. (Shi et al., 2022) achieved significant performance based on the YOLOv5 family of networks trained on a barley seed dataset, with the trained YOLOv5x6 model achieving a mean accuracy (mAP) of 97.5% in the recognition of barley seeds of different varieties. The development and application of these techniques provide new ideas and solutions to address automated seedling phenotyping, which is expected to play an important role in modern agriculture.

Considering the need for non-destructive, efficient, accurate, and consistent measurements for phenotyping rice seedlings, DeepLabv3+ (Chen et al., 2017) was used in this study as a baseline model for pixel-level segmentation of seedling images to extract the seedling's shoot, radicle, and seed parts. Subsequently, the shoot and root lengths of the seeds were analyzed in depth by further length measurement analysis methods. In the field of image segmentation, the DeepLab family is one of the widely used and excellent models. DeepLabv3+ has achieved 89.0% and 82.1% test performance on PASCAL VOC 2012 and Cityscapes datasets, respectively (Chen et al., 2017), which is accurate enough for high-precision image segmentation tasks. However, the main backbone network of this model, Xception, has a large number of parameters, which consumes a significant amount of GPU memory. Additionally, the model's memory footprint is substantial. As a result, it fails to meet the efficiency requirements for bud growth detection. To achieve fast and efficient detection, we optimized and improved the DeepLabv3+ model. We chose the FasterNet (Chen et al., 2023) network module with PConv as the backbone network to reduce the computational complexity. At the same time, we introduced the PSPA-ASPP structure and applied the EMA attention mechanism (Ouyang et al., 2023) to the network to improve the network operation speed and segmentation accuracy. This enables us to realize image segmentation in terms of efficiency and accuracy and significantly extends the applicability of the algorithm in practical applications. With this improvement, we can quickly and accurately recognize sprout root targets on the

germination plate. After obtaining the target contour, we used a length recognition algorithm and performed skeleton extraction based on the sprout-root contour, thus obtaining a high-precision skeleton of the seed germination and realizing the automated detection of sprout length and root length.

The goal of this study is to perform detailed phenotyping of rice seed germination and seedling stages based on deep learning techniques and high-throughput plant phenotyping methods. By deeply investigating the phenotypic changes in these critical growth stages, we can better understand the mechanisms of plant growth, development, and adaptation to the environment, and provide strong support for plant breeding and crop improvement. Meanwhile, this study is also expected to reveal the dynamic changes in root system structure during plant seed germination and seedling growth, thus providing new strategies and directions for improving crop yield and adapting to planting under different environmental conditions.

The contributions or innovations of this paper are mainly the following:

(1) A deep-learning-based high-throughput phenotyping tool for hypocotyls is presented, which is fully automated and achieves the accuracy of a human expert in length measurement tasks across various plant species.

(2) Using a germination plate to simulate the growth environment of rice seeds, images of rice seedlings were collected under the germination plate. Three common phenotypic targets—shoots, roots, and seeds—were selected to produce the dataset.

(3) An efficient plant phenotype segmentation method is provided, which can achieve efficient segmentation of crop images at the pixel level.

(4) The FasterNet-DeepLabv3+ (DFMA) semantic segmentation model is proposed, which reduces the computational complexity of the network and the impact of hollow convolutional meshing. It improves detection efficiency and accuracy, and addresses the problem of frequent memory accesses and inefficiency caused by using depth-separable convolution in the original network.

2 Materials and methods

2.1 Image acquisition and data preparation

The dataset is divided into two parts. The first part is a homemade rice seedling dataset for training and testing the model. The second part is the publicly available dataset used to validate the generalization of the proposed model.

The construction of the self-made rice seedling dataset involves two main stages, beginning with the setup of the growth environment. To simulate the natural growth environment of rice and ensure sample consistency, a custom-designed germination board was developed for this experiment. Seeds were laid flat on a black velvet cloth, then gently clamped between two acrylic sheets, which secured both the cloth and seeds without disrupting the normal growth process or disturbing their stable positions. The germination board was placed vertically in an incubator set to a temperature of 28°C, thus controlling the temperature to provide optimal conditions for germination. To maintain a moist environment, water was evenly sprayed onto the seed surface every 12 hours using a spray bottle. This controlled environment minimized external disturbances, creating consistent experimental conditions. The experiment spanned the critical 7- to 14-day growth period for rice seedlings, during which there are significant morphological changes, from germination to the preliminary formation of plant structure, capturing key characteristics of each growth stage. Consequently, the dataset contains images of seedlings from various growth stages, establishing a foundational resource for model development to recognize growth stage characteristics. A germination board seedling image is illustrated in Figure 1A.

During the germination and image capture phase, the experiment ensured stable seedling growth on the germination board under constant temperature and humidity conditions. Images were taken using various mobile devices to increase dataset diversity. All images were captured perpendicular to the germination board to minimize viewpoint deviation, while the well-lit laboratory environment ensured high-quality image sources. The use of different devices introduced natural device noise, attributed

(A) (B)

FIGURE 1
Homemade dataset germination plate pictures, **(A)** raw images, **(B)** mask images.

to sensor variations or light reflections, enhancing both the dataset's diversity and its robustness in real-world applications. To ensure data quality, all images were meticulously reviewed by botanical experts. A total of 115 healthy rice seedling samples were collected, spanning the 7- to 14-day growth period, thereby ensuring both representativeness and diversity in the dataset. In this study, Labelme open-source annotation software was used for manual image segmentation of images. The image was divided into four categories including shoots, roots, seeds and background. In the segmentation process, the parts of rice seedlings were separated from the background. For the fluff and secondary roots on the roots, they were treated as background. In this way, a homemade labeled dataset with the file suffix ".json" was obtained. Processed by the program, 115 sets of images were finally obtained. The sample image is shown in Figure 1B.

The public dataset was created using the Plant Segmentation Dataset, which was made public on the Kaggle platform by Orsolya Dobos et al. (https://www.kaggle.com/tivadardanka/plant-segmentation) in 2019. This dataset contains images of three seedlings, including *Arabidopsis thaliana*, *Brachypodium distachyon*, and *Sinapis alba*. The authors manually placed seedlings of these three plants on the surface of 1% agar plates and collected images using an EPSON PERFECTION V30 scanner. Images were saved in ".tif" or ".jpg" format using 800 dpi and 24-bit color settings. After collection, hypocotyls, cotyledons, seed coats, and roots were labeled using FIJI and used to create masks to train the segmentation algorithm. A sample of the dataset is shown in Figure 2.

2.2 Seedling phenotyping method

2.2.1 FasterNet network model

Some common network models, such as MobileNet (Howard et al., 2017), ShuffleNet (Zhang et al., 2017), and GhostNet (Han et al., 2020), widely utilize Depth-wise Separable Convolution (DWConv) and Group Convolution (GConv) to extract spatial features. Depth-wise Separable Convolution is favored for its advantage in reducing the number of parameters. However, replacing 2D convolution with Depth-wise Separable Convolution may result in a drop in model performance, yielding suboptimal models. Furthermore, Depth-wise Separable Convolution places higher demands on memory access, leading to slower computation speeds on GPUs, lower FLOPs, and higher latency. Similarly, Group Convolution can reduce the number of parameters, but the limited interaction between channels within the group may result in the loss of global channel information. During the process of reducing parameters and FLOPs, the computational operators often experience the side effect of increased memory access. These networks are often accompanied by additional data operations, such as concatenation, shuffling, and pooling, and the runtime latency of these operations is crucial for small-scale models. The formula for calculating latency is as follows:

$$\boldsymbol{Latency} = \frac{\boldsymbol{FLOPs}}{\boldsymbol{FLOPS}} \tag{1}$$

One of them, FLOPS (floating point operations per second), is widely used to evaluate the effectiveness of computational speed. Although there are many approaches aimed at reducing FLOPs, few of them also consider low-latency optimization. To address this issue, the authors (Chen et al., 2023) introduced PConv and proposed FasterNet. as a new family of net-works with lower latency, on a variety of devices, FasterNet not only provides state-of-the-art performance, but also enables lower latency and higher throughput.

The overall architecture of FasterNet has four layers, each containing respectively l1, and l2, l3, and l4 individual FasterNet blocks, which are preceded by an embedding or merging layer. The last layer is used for feature classification. In each FasterNet block, there is one PConv and two PWConv layers, corresponding to the two Conv 1×1 layers shown in the bottom-right corner of Figure 3. The resulting feature maps are convolved 1×1 after data normalization and ReLU activation function to preserve the complexity of the feature maps and to achieve lower latency. where PConv is a convolution operator that reduces computational redundancy and memory access. Figure 3, bottom left, illustrates how PConv works. It simply applies regular Conv to a portion of the input channel for spatial feature extraction while keeping the rest of the channel unchanged. For consecutive or regular memory accesses, the first or last consecutive channel is computed by considering the first or last consecutive channel as a representation of the entire feature map. The input

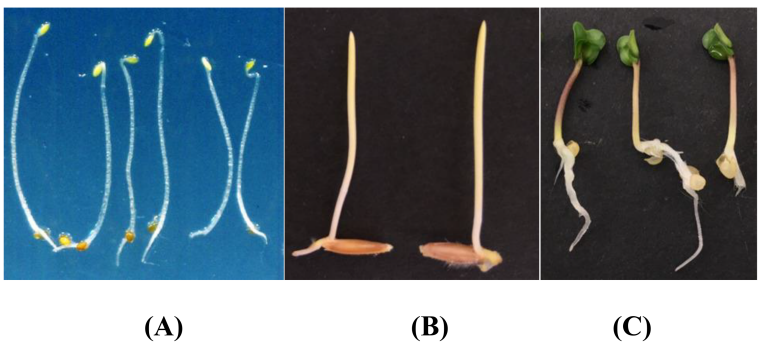


FIGURE 2
Plant segmentation public datasets. **(A)** Arabidopsis thaliana **(B)** Brachypodium dis-tachyon **(C)** Sinapis alba.

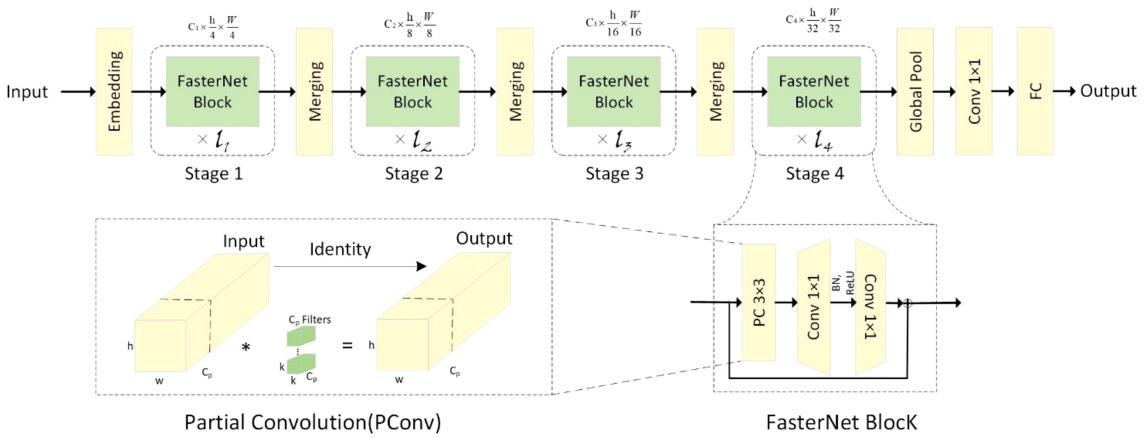


FIGURE 3
Overall architecture of FasterNet.

and output feature maps are considered to have the same number of channels without loss of generality. As a result, PConv reduces the FLOPs from $h \times w \times 2c' + k^2 \times c' \approx h \times w \times 2c'$ down to the number of channels in the $h \times w \times k^2 \times c_p^2$.

2.2.2 EMA attention mechanisms module

Attention mechanism modules are employed in neural networks to improve the selection and integration of information from image data, thereby enhancing model performance and accuracy. Examples include SE (Squeeze-and-Excitation) (Hu et al., 2020), CBAM (Convolutional Block Attention Module) (Woo et al., 2018), and CA (Channel Attention) (Hou et al., 2021). The SE attention mechanism focuses solely on channel-level attention and is suitable for scenarios with a higher number of channels but performs poorly when channels are limited. CBAM requires more computational resources, increasing computational complexity and FLOPs. CA also incurs additional computational overhead as it computes attention weights for the entire feature map, and it cannot capture long-range dependencies.

To further improve the performance of DeepLabv3+ network in extracting global information, we introduce a new efficient multi-scale attention module, EMA (Efficient Multiscale Attention) (Ouyang et al., 2023). EMA aims to preserve the information in each channel and reduce the computational overhead to achieve the goal of simultaneously preserving rich information and reducing the goal of computational cost. It achieves the effect of uniformly distributing spatial semantic features in each feature group by reconstructing some of the channels into batch dimensions and grouping the channel dimensions into multiple sub-features. The specific structure of EMA is shown in Figure 4.

A parallel substructure is used in the EMA module, which is applied in the attention mechanism to help the network avoid more parameters and greater depth, and the large local receptive fields of the neurons enable the neurons to collect multiscale spatial information. Therefore, EMA utilizes three parallel routes to extract the attention weight descriptors for the grouped feature maps. Two of the parallel routes are 1×1 branches and the third route is 3×3 branches. Cross-channel information interactions are also modeled in the channel direction. More specifically two 1D global average pooling operations are employed in the 1×1 branch to encode the channel along the two spatial directions respectively, while only one 3×3 kernel is stacked in the 3×3 branch for capturing multi-scale feature representations. Based on such a structure, EMA not only encodes the inter-channel information to adjust the importance of different channels, but also preserves the precise spatial structure information.

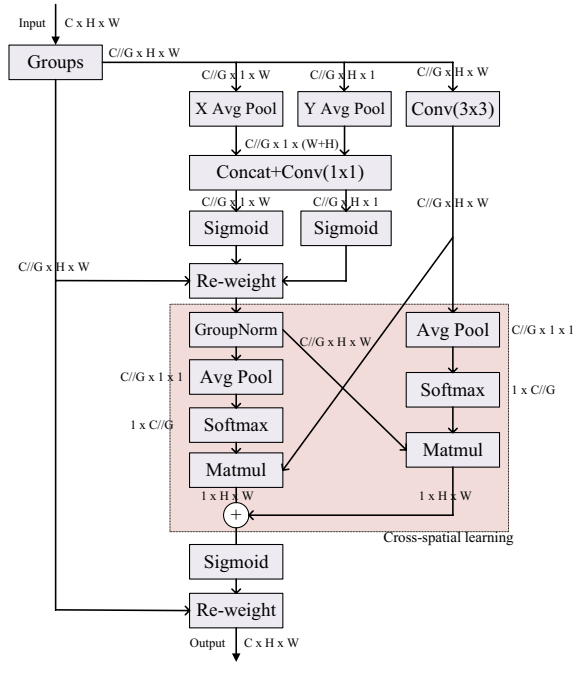


FIGURE 4
EMA self-attention module.

2.2.3 PSPA-ASPP spatial pooling pyramid layer

Inspired by Spatial Pyramid Pooling (SPP) (He et al., 2014), DeepLabv2 (Chen et al., 2017) introduced a novel module for semantic segmentation known as Atrous Spatial Pyramid Pooling (ASPP). The ASPP module's design is primarily based on the concept of dilated convolution. Traditional image segmentation algorithms often use pooling and convolution layers to increase the receptive field while simultaneously reducing the feature map size. However, when it becomes necessary to upsample or restore the size of feature maps from downsample and pooled layers, it can lead to a loss in the accuracy of image features and potential loss of semantic information from the original image. To address this issue, a method is needed that can increase the receptive field while keeping the feature map size unchanged, thus replacing upsampling and downsampling operations. Dilated convolution is precisely designed to meet this requirement. Dilated convolution extends the receptive field of convolutional operations by introducing holes (gaps) in the convolution kernel without changing the kernel's size. Specifically, dilated convolution introduces some virtual zero-value pixels in the convolution operation, allowing the expansion of the convolution kernel's receptive field without altering the feature map size. Figure 5A represents regular convolution, while (Figure 5B) represents dilated convolution with a dilation rate of 2, providing a comparison of the changes in receptive field between the two. ASPP's design represents a typical application of dilated convolution, achieving multiscale target information by parallelizing three dilated convolutions with different dilation rates, along with a standard convolution and a pooling operation.

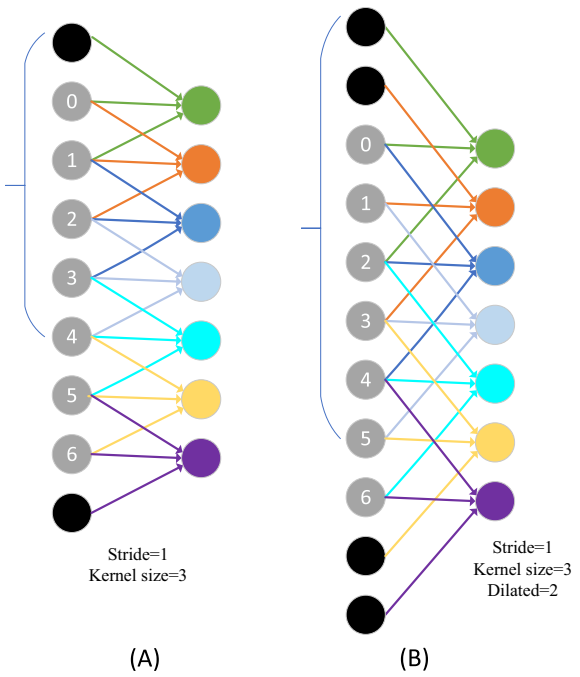


FIGURE 5
Visualization of the receptive field after the introduction of the null rate. **(A)** Represents regular convolution and **(B)** represents dilated convolution.

Although introducing dilated convolutions can increase the receptive field, it also suffers from two significant drawbacks. Firstly, dilated convolutions can lead to the problem of sparse sampling. While dilated convolutions excel in extracting global information, they may lack some semantic information when dealing with small targets. This is because larger dilation rates can result in excessive gaps between sampled points, making it challenging to capture fine details of small objects. Secondly, dilated convolutions exhibit the grid effect issue. When the same dilation rate is used or there exists a common divisor greater than 1, during the process of feature map stacking, it may lead to the loss of local detailed information in image features, resulting in a pixelated grid-like effect in the im-ages. This occurs because the same dilation rate or common divisor causes multiple sampled points to form a regular grid structure on the feature map, preventing the recovery of certain local information. Figure 6 illustrates the gridding effect of feature maps. When three consecutive convolution operations with a dilation rate of 2 and a kernel size of 3×3 are applied to a feature map, not all pixels on the feature map participate in the computation.

2.2.4 CARAFE up-sampling operator

The operator for feature upsampling is essential for increasing the resolution of low-resolution feature maps to match the size of high-resolution feature maps, and the design of an effective upsampling operator is of paramount importance (Mazzini, 2018; Chen et al., 2021; Dai et al., 2021). Among the widely used feature upsampling operators, nearest-neighbor interpolation and bilinear interpolation only consider sub-pixel neighborhoods, failing to capture the rich semantic information required for dense prediction tasks. The Transposed Convolution (Dumoulin and Visin, 2016), serving as the inverse operator of convolutional layers, employs convolution kernels of the same size throughout the entire image, thereby neglecting local information variations and leading to a significant increase in parameter count.

Wang et al. (Wang et al., 2019) introduced the CARAFE (Content-Aware ReAssembly of Features) feature re-sampling operator, which adaptively aggregates information within larger receptive fields, while maintaining remarkable computational efficiency. CARAFE generates weights in a content-aware manner by combining features within predefined regions near the central position. Multiple sets of such upsampling weights are computed for each central position, and the resulting features are rearranged into spatial blocks to complete the feature upsampling process. To validate the effectiveness of the CARAFE operator, the original authors conducted extensive experiments on Faster RCNN (Ren et al., 2015), employing various operators for upsampling within the Feature Pyramid Network (FPN). The results, as shown in Table 1, included cases denoted as "nearest neighbor + convolution" (N.C.) and "bilinear + convolution" (B.C.), where an additional 3×3 convolution layer was added after the corresponding upsampling. The comparative experiments also included three typical upsampling methods: deconvolution (Deconv), pixel shuffle (P.S.),

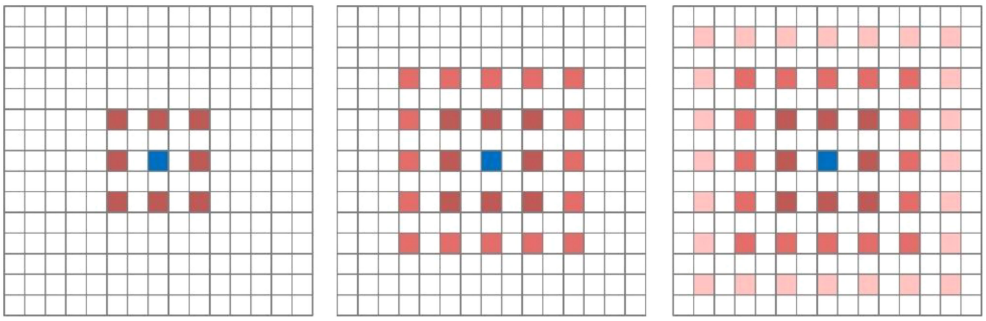


FIGURE 6
Mapping of gridding effects. From left to right, the dilation rates are 2, 2, and 2, respectively. Following the approach outlined by Shi et al (Shi and Bao, 2023), our research team devised a novel ASPP (Atrous Spatial Pyramid Pooling) structure known as PSPA-ASPP. Firstly, we replaced the original ASPP's first branch layer's 1×1 convolution with a 3×3 Pconv convolution to broaden the receptive field of the first layer while avoiding redundant learning. Secondly, we employed two 3×3 dilated convolutions with dilation rates of 2 and 3, each with 128 convolution kernels, which is half of the original ASPP's individual branch, and concatenated them in the channel dimension. Subsequently, we applied two additional 3×3 dilated convolutions with dilation rates of 5 and 7 in a similar concatenated manner. This design allows the network to capture features from different scales while substantially reducing the grid effect and making more effective use of feature layer information. The final layer still employs average pooling to capture global features of the feature map. Figure 7 illustrates the overall network architecture of PSPA-ASPP.

and guided upsampling (GUM), as well as spatial attention (S.A.). CARAFE exhibited the highest average precision (AP) among all upsampling operators while maintaining lower FLOPs and parameter counts, indicating its efficiency in enhancing detail recovery and excelling in model lightweighting. Results for N.C. and B.C. suggested that additional parameters did not yield significant gains, whereas Deconv, P.S., GUM, and S.A. all exhibited inferior performance compared to CARAFE.

As shown in Figure 8, CARAFE, as an upsampling operator with a content-aware kernel, consists of two steps. The first step is to predict the reassembly kernel for each target position based on its content (i.e., the Kernel Prediction Module in Figure 8). The second step is to use the predicted kernel to reassemble the features (i.e., the Content-aware Reassembly Module in Figure 8). In the first step, a feature map $\mathcal{X}$ of size $C \times W \times H$ is upsampled by a factor of σ, resulting in a new feature map of size C×σH×σW. Assuming an upsample kernel size of $k_{up} \times k_{up}$, if different upsample kernels are desired for each position in the output feature map, the predicted upsample kernel should have a shape of $\sigma H \times \sigma W \times k_{up} \times k_{up}$. To compress the input feature map, a convolution layer with a kernel size of $k_{encoder} \times k_{encoder}$ is used to predict the upsample kernel, with an input channel number of C_m and an output channel number of $\sigma^2 k_{up}^2$, resulting in an upsample kernel of shape $\sigma H \times \sigma W \times k_{up}^2$. In the second step, for each position in the output feature map, it is mapped back to the input feature map, and a $k_{up} \times k_{up}$ region centered on that point is extracted. The dot product is then computed between the extracted region and the predicted upsample kernel for that point to obtain the output value. Different channels at the same position share the same upsample kernel.

TABLE 1 Comparison of the performance of sampling operators on CARAFE.

Method	AP	FLOPs	Params
Nearest	36.5	0	0
Bilinear	36.7	8k	0
N.C	36.6	4.7M	590K
B.C	36.6	4.7M	590K
Deconv	36.4	1.2M	590K
P.S	36.5	4.7M	2.4M
GUM	36.9	1.1M	132K
S.A	36.9	28K	2.3K
CARAFE	37.8	199K	74K

In the improved Deeplab v3+ network, as illustrated in Equation 2, the kernel prediction module ψ predicts the position for each location based on the learned weights $\mathcal{W}_{l'}$ in the first step. Subsequently, as described in Equation 3, the content-aware recombination module ϕ recombines the features $\mathcal{X}_l$ with the kernel $\mathcal{W}_{l'}$ in the second step. To reduce the parameter count of upsampling operators and enhance efficiency, an 8-fold upsampling CARAFE module is introduced after the ASPP module, which restores the size of the feature maps from 256 × 16 × 16 to 256 × 128 × 128. Following feature fusion, a 4-fold upsampling operation is applied to restore the final feature map to 4 × 512 × 512 dimensions.

$$\mathcal{W}_{l'} = \boldsymbol{\psi}(\boldsymbol{N}(\mathcal{X}_l, \boldsymbol{k_{encoder}})) \tag{2}$$

$$\mathcal{X}_{l'} = \boldsymbol{\phi}(\boldsymbol{N}(\mathcal{X}_l, \boldsymbol{k_{up}}), \mathcal{W}_{l'}) \tag{3}$$

2.2.5 DFMA overall network structure

The DFMA model integrates the FasterNet backbone with the SPA-ASPP module enhanced by an EMA attention mechanism, aimed at improving feature extraction and segmentation accuracy for plant seedling images while being optimized for mobile deployment. Initially, the input RGB image undergoes feature extraction via the FasterNet backbone. FasterNet leverages a

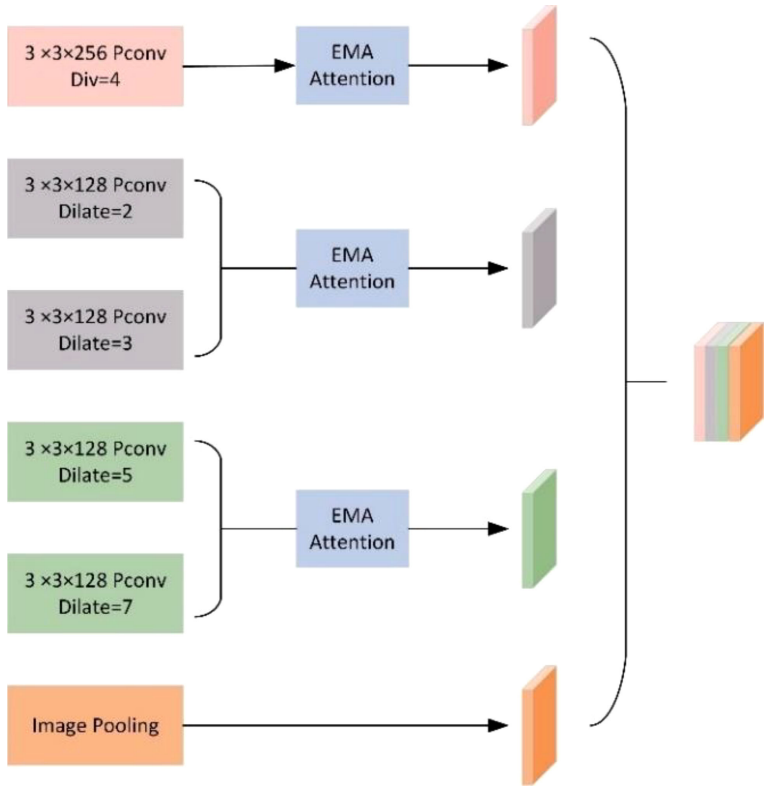


FIGURE 7
PSPA-ASPP series-parallel network structure diagram.

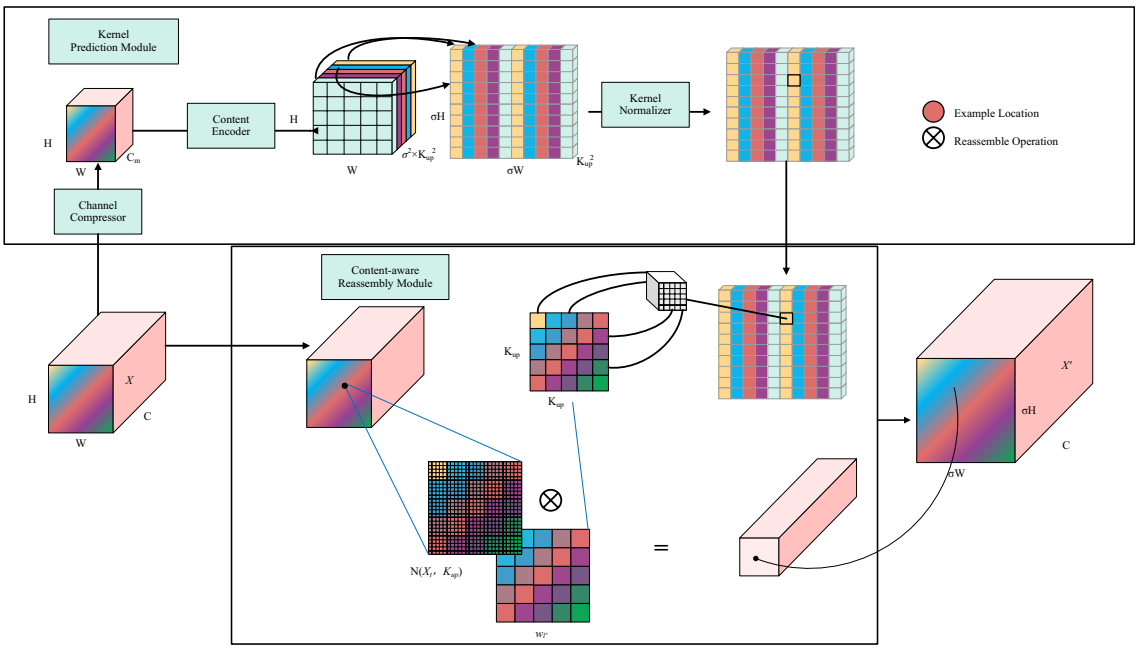


FIGURE 8
CARAFE up-sampling operator.

10.3389/fpls.2024.1457360

hybrid structure combining Pconv, PWconv, and standard convolution to efficiently extract both low-level and high-level features, overcoming the limitations of depthwise separable convolution. To ensure the participation of shallow features in subsequent processing, the model retains shallow feature maps downsampled four times within the backbone network. Following this, DFMA introduces an EMA (attention mechanism) module that enhances the fusion capability of high-level features. The EMA mechanism dynamically reweights features from different layers, enabling the network to focus on key parts of the image when extracting high-level features, thus boosting overall performance.

During the multi-scale feature extraction stage, DFMA employs the SPA-ASPP module with EMA attention. This module captures high-level semantic information across multiple scales through several branches, effectively avoiding grid effects common in traditional methods. The EMA attention mechanism further strengthens the representation capacity of these branches, allowing the model to concentrate on crucial features within plant seedling images.

In the decoding stage, the multi-scale feature information is merged and upsampled using the CARAFER operator, aligning the high-level feature map dimensions with the low-level feature map for subsequent fusion. DFMA applies a 1×1 convolution on the shallow feature map to match channel dimensions with the upsampled deep feature map, preparing it for concatenation. The concatenated feature map then undergoes partial convolution and additional upsampling, ultimately generating the model's prediction. This integrated design combines the strengths of FasterNet and the SPA-ASPP module, enhancing the model's feature extraction capacity while ensuring efficiency and accuracy for mobile deployment. The DFMA model structure is shown in Figure 9.

2.3 Seedling length detection method

Through training the DFMA network model, we can easily input seedling images for analysis and obtain corresponding masks. These masks accurately represent the different positions of seeds within the images, allowing researchers to observe the developmental details of seed germination, embryonic axis, and root structure clearly. In certain studies, it is not only necessary to conduct in-depth analysis of the development of various plant parts but also to acquire precise parameters for these developmental aspects. Therefore, we introduce a seedling length measurement algorithm, which not only provides accurate segmentation masks for the images but also enables us to obtain exact parameters for the development of different plant parts.

In this seedling length detection, we divided into two main steps. First, we skeletonize the image using the Hilditch algorithm to obtain the median length of the segmented image. Secondly, we utilize Hough Transform to obtain the transformation relationship between the true length of the seedling detection site and the pixels.

The Hough Transform is an early image processing algorithm that employs a voting-based approach for shape fitting. Its objective is to mathematically describe certain edges in an image to enhance information extraction. Unlike alternative techniques such as least squares, robust estimation, and RANSAC, the Hough Transform excels in simultaneously fitting multiple objects. The detection process in the Hough Transform involves iterating through all non-zero points, accumulating votes for each point's center, and assigning scores. For each point along a circle, its center lies on the vector perpendicular to the point and passing through the point's location. The intersection point of these center vectors corresponds to the desired circle center position. In this experiment, coins serve as a real-world scale for converting lengths to pixels, enabling the detection of coin diameters.

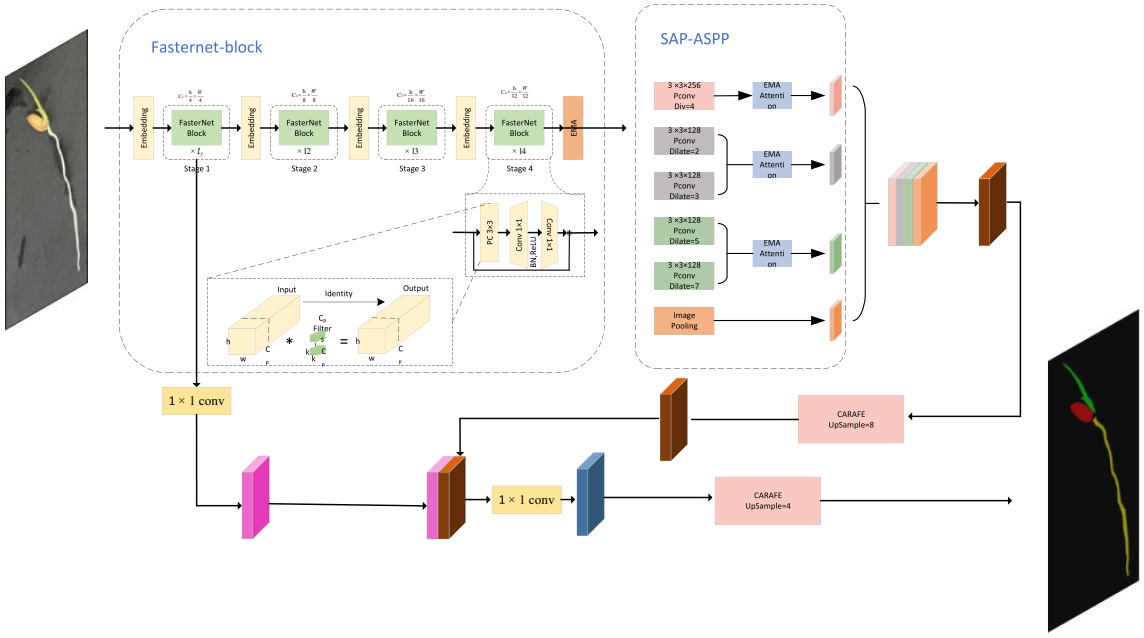


FIGURE 9
Improvement of the overall architecture of FasterNet-Deeplab v3+.

Within the Hough Transform, fitting circles requires three parameters - (x, y, r), where x and y denote coordinates, and r represents the circle's radius. These parameters are determined using the following formula:

$$(X - x^2) + (Y - y^2) = r^2 \quad (4)$$

The Hough Gradient method optimizes the standard Hough Circle Transform by eliminating the need to draw complete circles in parameter space for voting. Instead, it calculates the gradient vectors at contour points and casts votes along the gradient direction, at a distance of R in both directions from the contour point, effectively conducting one vote on each side. Ultimately, the circle center's position is determined based on the voting results as depicted in Figure 10.

As shown in the diagram, assuming that the gradient directions of the contour points ACDE all pass through point B, they will each cast a vote for point B. Within a search radius of R, votes are cast on both sides of the contour points at a distance of R based on the gradient direction. Ultimately, the center position is determined based on the voting results. Compared to the parameter space voting method for determining the center, this approach offers better resistance to interference. Even if other points also cast votes, their voting results are too dispersed, and their interference with the overall voting result can be almost negligible.

For this experiment we use coins as a scale between real and pixel values, and the actual value of the sprout length can be calculated based on the coin diameter. A dollar coin as a circle with a diameter of 25mm, get how many pixels it occupies in the figure, it can get the number of pixels per metric (pixel Per Metric), and then calculate the pixels occupied by other objects n, it can get the actual length (n × pixel Per Metric).

3 Experiments and results

3.1 Model evaluation criteria

In this network model of bud root region segmentation, the deep learning network mainly adopts Mean Intersection over Union (*mIoU*) as the evaluation index of the model, and mean intersection over union refers to the ratio of intersection and concatenation values between the true and predicted values of each classification, and then averages over multiple classifications.

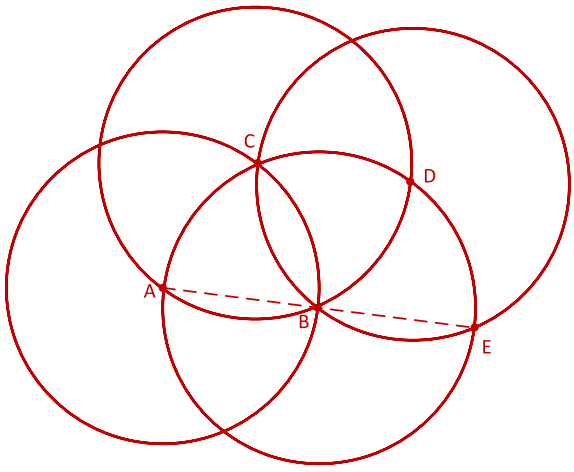


FIGURE 10
Hough gradient method.

In the field of scientific research and data analysis, True Positive (*TP*)is defined as the portion where both the actual value and the predicted value are true. True Negative (*TN*)corresponds to cases where both the actual value and the predicted value are false. False Positive (*FP*)refers to instances where the actual value is false, but the predicted value is true. False Negative (*FN*) denotes situations where the actual value is true, but the predicted value is false.

$$MIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{TP}{FN + FP + TP} \quad (5)$$

In addition to mIoU, precision (*Pre*), recall (*Rec*), and accuracy (*Acc*) are also used as evaluation metrics for the algorithm. Precision (*Pre*) is used to measure the proportion of predictions that are correct in the samples that the model predicts as positive examples, with the formula shown in Equation 6:

$$precision = \frac{TP}{TP+FP} \quad (6)$$

Recall (*Rec*) is the proportion of all positive cases that the model predicts correctly, as shown in Equation 7:

$$recall = \frac{TP}{TP+FN} \quad (7)$$

Accuracy (*Acc*) is the number of samples with all correct predictions as a percentage of all samples. The higher its value, the better the model. As shown in Equation 8:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

3.2 Data augmentation settings in the training phase

In this study, we employed online data augmentation techniques to enhance the robustness and generalization capability of the model. The data augmentation operations included random scaling (with a scale range of 0.25 to 2 times), aspect ratio distortion, horizontal flipping (with a probability of 50%), gray padding (pixel value of 128), random adjustments to hue, saturation, and brightness in the HSV color space, as well as random cropping and shifting. These augmentation methods were dynamically applied to the training data's images and labels during each training iteration, thereby expanding the original data distribution, simulating target variations under different scenarios and conditions, and significantly improving the model's adaptability to changes in lighting, orientation, and target shapes. Moreover, dynamic augmentation reduced the need for storing pre-augmented data while significantly increasing data diversity, thereby improving training effectiveness. It is important to note that data augmentation was only applied during the training phase and not during the validation phase to ensure that the validation results objectively reflect the true performance of the model. The

experimental results demonstrate the effectiveness of the proposed method in reducing overfitting and improving model performance.

3.3 Experimental platform and parameter design

The network is implemented based on the PyTorch library and trained on a single Nvidia RTX 3060 GPU, with a 12th Gen Intel(R) Core(TM) i5 - 12400F processor. The initial batch size is set to 10, and the initial learning rate is 0.05. Stochastic Gradient Descent (SGD) is adopted as the optimization method, and both Dice loss and cross - entropy loss are utilized as the objective functions. L2 regularization is applied for model regularization. We use online data augmentation techniques, such as rotation (by 90, 180, and 270 degrees), horizontal flipping, and random adjustments to hue, saturation, and brightness in the HSV color space. The original dataset contains 115 images, which are split into a training set of 92 images and a validation set of 23 images following an 8:2 ratio. Through these online augmentation operations, each original training image can generate multiple variants during each training iteration. To estimate the approximate quantity of the augmented training data, considering that each image has 7 different augmented forms on average (3 rotations + 1 horizontal flip + 3 color space adjustments), the total number of augmented training images is about 644. During training, the batch size is adjusted to 8. The training process will automatically stop when the loss function output of the validation set does not decrease for 20 consecutive epochs, with a maximum of 500 epochs permitted. The segmentation performance is evaluated on the validation set using the Mean Intersection over Union (mIoU) metric (Table 2).

TABLE 2 Training parameters.

Parameter	Value
Initial learning rate	0.005
End Lr	0.0001
Momentum	0.937
Batch size	8
Lr policy	Adam
Lr decay	cos
epoch	500

3.4 Evaluation of the results of the seedling phenotype segmentation experiment

According to the analysis results in Table 3, it is evident that FasterNet exhibits significant advantages in network backbone selection. Moreover, during the experimental phase, we observed that FasterNet's training process is notably faster, which may be attributed to the frequent memory access associated with depth-wise separable convolutions and pointwise convolutions used in Xception and MobileNet. In our proposed PSPA-ASPP structure, when the backbone networks are the same, the combination of FasterNet with ASPP achieves an mIoU of 79.84, whereas when combined with PSPA-ASPP, it reaches 81.36. It is noteworthy that FasterNet+PSPA-ASPP also boasts lower GFLOPs, indicating its competitiveness in terms of computational efficiency. The final experimental results demonstrate that the FasterNet+EMA +PSPA-ASPP+CARAFE combination exhibits the best performance, further substantiating its outstanding performance in image segmentation tasks.

The primary objective of this experiment is to achieve more precise phenotypic analysis; therefore, when differences in other metrics are minimal, this study prioritizes model accuracy. The improved FasterNet-DeepLab V3+ achieves the highest mIoU while significantly reducing GFLOPs. By simplifying the branches with the PSPA-ASPP module, the GFLOPs are reduced by approximately 2.161 G, effectively enhancing the model's learning capacity.

In accordance with Figure 11, we conducted a comparative experimental analysis of prediction results using the DeepLabv3+ semantic segmentation model with MobileNet and Xception as

TABLE 3 Results of ablation experiments.

Xception	MobileNetV2	FasterNet	CA	SP	EMA	PSPA_ASPP	CARAFE	MIoU/%	GFLOPs/G
✓								67.09	167.00
	✓							74.21	53.03
		✓						79.84	138.70
		✓	✓					78.79	138.71
		✓	✓	✓				81.32	141.52
		✓		✓	✓			81.35	139.45
		✓				✓		81.36	135.23
		✓			✓		✓	81.63	139.83
		✓			✓	✓		81.58	137.29
		✓			✓	✓	✓	**81.72**	137.67

Bold value represents the highest mIoU achieved by our model in the tests.

10.3389/fpls.2024.1457360

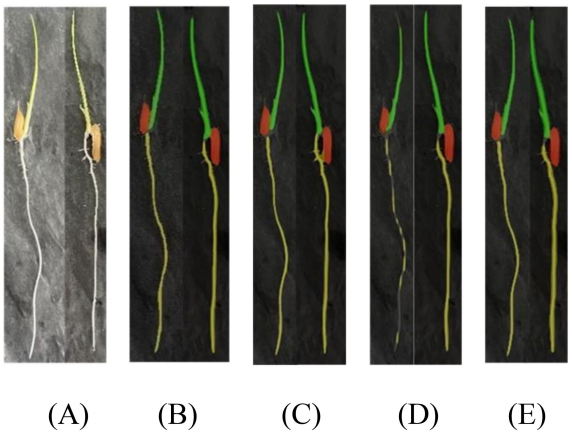


FIGURE 11
Phenotypic recognition results for homemade datasets. **(A)** Original figure. **(B)** DFMA. **(C)** deepLab v3+-MobileNet. **(D)** deepLab v3 +-Xception. **(E)** UNet-VGG.

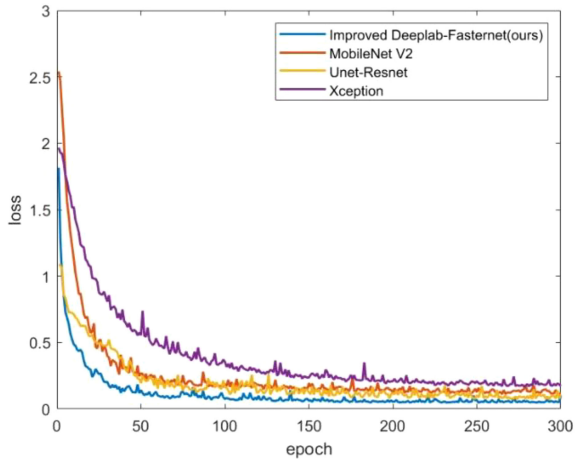


FIGURE 12
Loss curves of different models on two-spike phragmites picture dataset.

backbone networks, the Unet-VGG segmentation model, and our improved DFMA network in our research. As evident from the results in Figure 11D, the segmentation performance is the poorest in this case, with issues of coherence in the regions covered by masks for rice seedling shoots and root areas, resulting in suboptimal segmentation. In contrast, our proposed DFMA network model exhibits the best performance, accurately segmenting each region.

On the public dataset, the DFMA was compared with networks such as UNet (a network provided by the original authors of the public dataset), MobileNetV2, and Xception in terms of equalization and concurrency results, as shown in the Table 4.

Based on the analysis results presented in Table 4 and illustrated in Figure 12, it is evident that our proposed DFMA model demonstrates exceptional performance on publicly available datasets, outperforming other models. Across three distinct plant datasets, namely short-stalked grass, white Sinapis alba, and Arabidopsis thaliana, the DFMA model achieves average intersection over union (mIoU) ratios of 87.69%, 91.07%, and 66.44%, respectively, surpassing the other two models by at least 2 percentage points. Furthermore, as depicted in Figure 12, during the training process, it is apparent that the DFMA network model converges more swiftly and maintains a lower loss function value, providing additional evidence of its superior performance and efficiency.

TABLE 4 MIoU results (%) of different network trainings on public dataset.

Model	Brachypodium distachyon	Sinapis alba	Arabidopsis thaliana
DFMA	87.69	91.07	66.44
MobileNetV2	84.84	87.21	63.39
Xception	78.78	68.75	56.22
UNet-VGG	80.10	85.65	62.82

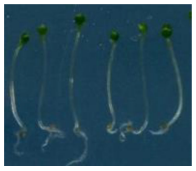
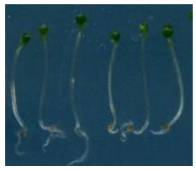
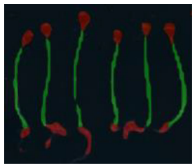
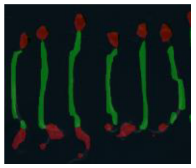
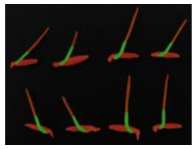
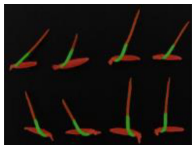
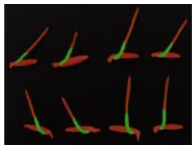
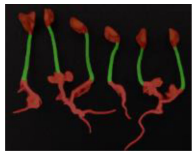
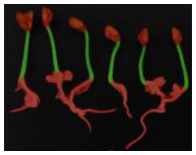
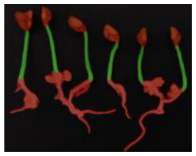
In accordance with Table 5, our proposed DFMA network achieves the best segmentation performance on publicly available datasets. Due to the limitations of depth-wise separable convolution, the MobileNet network exhibits poor mask recognition in the bud apex region. Conversely, due to its restricted network depth, UNet produces relatively coarse results in fine detail recognition.

The DFMA model outperforms other models in plant phenotyping analysis, largely due to its design tailored to address the unique challenges of seedling segmentation tasks. Plant phenotyping often requires accurate identification of intricate and complex structures. Seedling images commonly contain multi-scale, fine structural features, such as leaf edges and stems, which demand high segmentation precision. Additionally, the execution environment for seedling segmentation tasks is typically resource-limited, such as mobile devices or automated equipment, imposing strict requirements for model efficiency and lightweight design.

The DFMA model utilizes FasterNet as its backbone network, known for its efficient spatial feature extraction without relying on depthwise separable convolutions. While depthwise separable convolutions offer a lightweight solution, they may fall short in efficiently capturing details within complex structural images. FasterNet's design, incorporating a combination of Pconv, PWconv, and standard convolution, achieves a balance between lightweight operation and efficiency, making it well-suited for deployment in resource-constrained environments.

Furthermore, DFMA integrates an SPA-ASPP module with EMA (Attention Mechanism), enabling detailed feature capture across multi-scale branches and mitigating the grid effect commonly seen in traditional ASPP modules. The grid effect can lead to feature loss or blurred image boundaries, but the EMA attention mechanism allows the model to focus precisely on key areas of seedlings, such as leaves and stems, resulting in outstanding performance in detail-rich scenarios. This capability is critical for fine-grained segmentation in plant phenotyping, as capturing details aids researchers in better understanding plant growth conditions and morphological characteristics.

TABLE 5 Plant phenotype segmentation results for different networks of the open dataset.

	original figure	Improvement of FasterNet-Deeplab V3+	Deeplab V3+-MobileNet	UNet-VGG
Arabidopsis thaliana				
Brachypodium distachyon				
Sinapis alba				

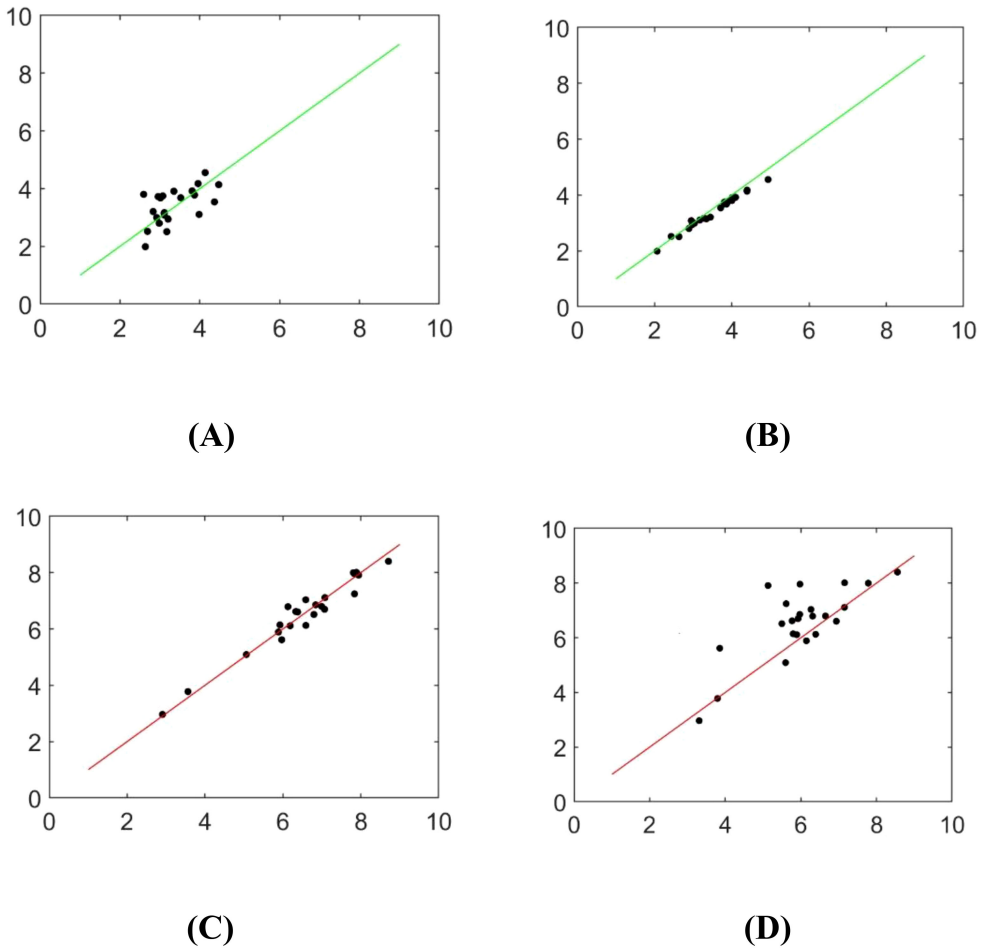


FIGURE 13
Scatterplot of identifying rice seedlings **(A)** Adam optimizer detects bud length **(B)** SGD optimizer detects bud length **(C)** Adam optimizer detects root length **(D)** The SGD optimizer detects root length.

3.5 Evaluation of the results of the seedling phenotype segmentation experiment

The skeleton extraction algorithm was employed to identify the central axis of the mask, enabling the computation of the seedling shoot and root length. Figure 13 and Table 6 depict the image analysis results obtained through both manual detection and the experimental method described in this paper. In these visualizations, the horizontal axis represents the manually measured values, while the vertical axis represents the corresponding measurements obtained from seedling images using the method outlined in this study. Statistical analysis in

TABLE 6 Relative errors of different algorithms for length recognition of rice seedling images.

Model	Serial number	Maximum absolute error (cm)	Minimum absolute error (cm)	Mean absolute error (cm)	Improvement (%)	
					Vs. DeepLabV3+	Vs. Unet-VGG
Original Deeplab V3+ model	bud	0.583	0.028	0.386	–	–
	radical	0.506	0.016	0.724		
UNet-VGG	bud	0.876	0.034	0.410	–	–
	radical	1.467	0.074	0.862		
DFMA	bud	0.384	0.007	0.146	+62.20	+64.44
	radical	0.393	0.006	0.231	+68.09	+73.20

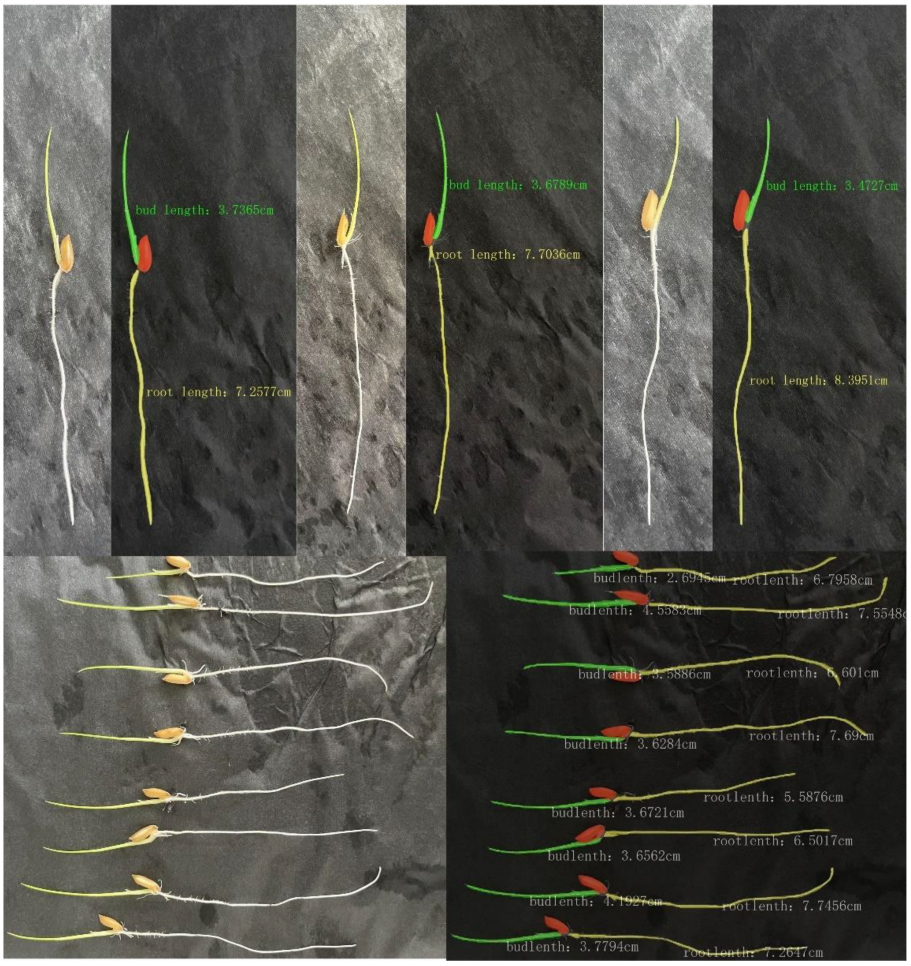


FIGURE 14
Results of batch testing of rice shoot root lengths.

Table 6 is conducted by grouping every 5 seedlings together for assessment.

The measured values obtained by the algorithm used in this paper and the manual measurement values are highly consistent, and the improved DeepLabv3+ network yields better results than the original DeepLabv3+ network. However, there is still a small error. Possible reasons for the error include the skeleton extraction step after filling the interior of the contour, which causes the algorithm to use the centerline instead of the main root. Additionally, there is an offset in the refinement process, resulting in inconsistent calculated lengths.

Compared to the original DeepLabv3+ model, the improved model reduced the mean absolute errors in measuring shoots and roots by 62.20% and 68.09%, respectively. Compared with the UNet-VGG model, it achieved improvements of 64.44% and 73.20%, respectively, and demonstrated a more significant detection advantage in terms of maximum and minimum absolute errors.

In this study, based on the improved DeepLabv3+ target segmentation network combined with the length detection algorithm, the sprout target is recognized and segmented, and the sprout length is ultimately obtained. The recognition results are shown in Figure 14 below. The model in this study demonstrates superior recognition of the target, accurately segments the outline and key parts of the target, and simultaneously avoids confusion between the target and the background. It provides more accurate length detection results and is capable of batch detection.

4 Discussion

This study proposes a high-throughput plant phenotyping method based on deep learning, highlighting its broad application potential and significance across multiple fields. Through a non-destructive, efficient, accurate, and consistent measurement approach, we achieved phenotypic analysis of rice seedlings at early growth stages, significantly improving research efficiency and broadening future applications. In line with specific experimental tasks, we selected datasets from four species, three from public Kaggle datasets and one collected independently. This choice allowed us to test the model's performance under relatively consistent environmental conditions, minimizing external factors and yielding clearer experimental results. However, we recognize that the current datasets are limited in species and environmental diversity, and expanding this diversity is necessary to further enhance the model's robustness and generalizability. Future research will therefore introduce more samples from diverse species and environmental conditions to improve the model's adaptability and applicability in complex, dynamic scenarios.

Although the improved DeepLabv3+ and the newly introduced DFMA semantic segmentation model perform excellently in segmentation efficiency and accuracy, they still face limitations in lighting adaptability, cross-crop transferability, and multi-species analysis. To enhance the model's broad applicability, future work will focus on further strengthening the model's robustness to varying lighting conditions and exploring ways to adjust feature extraction and attention modules to better accommodate plants with diverse morphological features. As research progresses, we also plan to expand this technology to other crops and plant species, further uncovering growth and developmental characteristics. This will provide scientific support for crop improvement and cultivation, and advance ecological research, helping scientists better understand plant responses to environmental changes.

The application prospects of this technology extend beyond plant phenotyping, with potential in fields such as medical image analysis and autonomous driving, demonstrating deep learning's immense potential for automation and precision in image processing. This technology holds significant value for research in biology and botany. In the future, we plan to open-source a WeChat-based plant phenotyping mini-program to promote practical applications of this research and facilitate further developments. This will provide innovative tools and directions for plant breeding and crop improvement.

5 Conclusion

In summary, our study addresses a critical need in the rapidly evolving field of plant phenotypic research. Accurate seedling length measurement is essential for evaluating seed viability and growth status. We have developed an efficient and versatile deep learning approach, named DFMA, which incorporates the innovative PSPA-ASPP structure. Our model consistently outperforms traditional methods and other models, achieving remarkable segmentation and detection results across various plant species. DFMA generates precise segmentation masks that highlight detailed developmental aspects of seedling components, such as cotyledons, hypocotyls, and roots. Furthermore, we introduce a novel seedling length measurement algorithm, providing precise parameters for a comprehensive plant phenotypic analysis. Our research holds great promise for offering more efficient tools and data support to advance the field of plant biology, enhancing our understanding of plant genetics and growth trends in the top-tier scientific community.

Data availability statement

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

Author contributions

LJ: Conceptualization, Methodology, Supervision, Visualization, Writing – original draft, Writing – review & editing. TW: Formal analysis, Methodology, Writing – original draft, Writing – review & editing. XL: Investigation, Methodology, Visualization, Writing – original draft, Writing – review & editing. LG: Funding acquisition, Visualization, Writing – original draft, Writing – review & editing. QY: Visualization,

Writing – original draft, Writing – review & editing. XZ: Resources, Writing – original draft, Writing – review & editing. SM: Project administration, Supervision, 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 Natural Science Foundation of Huzhou Public Welfare Projects (2021GZ23, 2021GZ30); Natural Science Foundation of Zhejiang Public Welfare Projects (LTGN23C130002, LGN22C190029); Postgraduate Research and Innovation Project of Huzhou University (No.2024KYCX47).

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

References

Chen, J., Kao, S. H., He, H., Zhuo, W., Wen, S., Lee, C. H., et al. (2023). "Run, Don't walk: Chasing higher FLOPS for faster neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, BC, Canada, 18–22 June. (Piscataway, NJ: IEEE) 12021–12031.

Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., et al. (2021). *Transunet: Transformers make strong encoders for medical image segmentation. arXiv*, arXiv:2102.04306.

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, 834–848. doi: 10.1109/TPAMI.2017.2699184

Comas, L. H., Becker, S. R., Cruz, V. M. V., Byrne, P. F., and Dierig, D. A. (2013). Root traits contributing to plant productivity under drought. *Front. Plant Sci.* 4, 442. doi: 10.3389/fpls.2013.00442

Dai, Y., Lu, H., and Shen, C. (2021). "Learning affinity-aware upsampling for deep image matting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual*, 19–25 June. 6841–6850. New York: IEEE.

Dobos, O., Horvath, P., Nagy, F., Danka, T., and Viczián, A. (2019). A deep learning-based approach for high-throughput hypocotyl phenotyping. *Plant Physiol.* 181, 1415–1424. doi: 10.1104/pp.19.00728

Dumoulin, V., and Visin, F. A. (2016). *guide to convolution arithmetic for deep learning. arXiv*, arXiv:1603.07285.

Fenta, B. A., Beebe, S. E., Kunert, K. J., Burridge, J. D., Barlow, K. M., Lynch, J. P., et al. (2014). Field phenotyping of soybean roots for drought stress tolerance. *Agronomy* 4, 418–435. doi: 10.3390/agronomy4030418

Gendreau, E., Traas, J., Desnos, T., Grandjean, O., Caboche, M., and Hofte, H. (1997). Cellular basis of hypocotyl growth in Arabidopsis thaliana. *Plant Physiol.* 114, 295–305. doi: 10.1104/pp.114.1.295

Guingo, E., Hébert, Y., and Charcosset, A. (1998). Genetic analysis of root traits in maize. *Agronomie* 18, 225–235. doi: 10.1051/agro:19980305

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*, Seattle, WA, USA, 14–19 June. 1577–1586. New York: IEEE.

He, K., Zhang, X., Ren, S., and Sun, J. (2014). "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Computer Vision—ECCV 2014*. Eds. D. Fleet, T. Pajdla, B. Schiele and T. Tuytelaars (Switzerland: Springer International Publishing, Cham), 346–361.

Holman, F. H., Riche, A. B., Michalski, A., Castle, M., Wooster, M. J., and Hawkesford, M. J. (2016). High throughput field phenotyping of wheat plant height and growth rate in field plot trials using UAV based remote sensing. *Remote Sens.* 8, 1031. doi: 10.3390/rs8121031

Hou, Q., Zhou, D., and Feng, J. (2021). *Coordinate attention for efficient Mobile Network Design. arXiv*, arXiv:2103.02907.

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*, arXiv:1704.04861.

Hu, J., Shen, L., Albanie, S., Sun, G., and Wu, E. (2020). Squeeze-and-excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 2011–2023. doi: 10.1109/TPAMI.34

Jiang, Y., Xu, H., Yan, S., and Wang, J. (2022). Effects of polystyrene microplastics on seed germination and seedling growth of wine sorghum variety "Red Cherry". *Seed* 41, 108–113. doi: 10.16590/j.cnki.1001-4705.2022.10.108

Lynch, J. (1995). Root architecture and plant productivity. *Plant Physiol.* 109, 7. doi: 10.1104/pp.109.1.7

MansChadi, A. M., Kaul, H. P., Vollmann, J., Eitzinger, J., and Wenzel, W. (2014). Developing phosphorus-efficient crop varieties—an interdisciplinary research framework. *Field Crops Res.* 162, 87–98. doi: 10.1016/j.fcr.2013.12.016

Marset, W. V., Pérez, D. S., Díaz, C. A., and Bromberg, F. (2021). Towards practical 2D grapevine bud detection with fully convolutional networks. *Comput. Electron. Agric.* 182, 105947. doi: 10.1016/j.compag.2020.105947

Mazzini, D. (2018). *Guided Up-sampling Network for Real-Time Semantic Segmentation. arXiv*, arXiv:1807.07466v1.

Ouyang, D., He, S., Zhang, G., Luo, M., Guo, H., Zhan, J., et al. (2023). "Efficient multi-scale attention module with cross-spatial learning," in *Proceedings of the ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 4–10 June. 1–5. New York: IEEE.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* 28, 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Ribaut, J. M. (2006). *Drought Adaptation in Cereals* Vol. p (Binghamton, NY, USA: The Haworth Press Inc), 682.

Richard, C. A., Hickey, L. T., Fletcher, S., Jennings, R., Chenu, K., and Christopher, J. T. (2015). High-throughput phenotyping of seminal root traits in wheat. *Plant Methods* 11, 13. doi: 10.1186/s13007-015-0055-9

Shi, Y., and Bao, G. (2023). Bamboo end face segmentation and branch position detection method fused with improved ASPP and CBAM. *J. For. Eng.* 05, 138–145. doi: 10.13360/j.issn.2096-1359.202211036

Shi, Y., Li, J., Yu, Z., Li, Y., Hu, Y., and Wu, L. (2022). Multi-barley seed detection using iPhone images and YOLOv5 model. *Foods* 11, 3531. doi: 10.3390/foods11213531

VanToai, T. T., St. Martin, S. K., Chase, K., Boru, G., Schnipke, V., Schmitthenner, A. F., et al. (2001). Identification of a QTL associated with tolerance of soybean to soil waterlogging. *Crop Sci.* 41, 1247–1252. doi: 10.2135/cropsci2001.4141247x

Wade, L. J., Bartolome, V., Mauleon, R., Vasant, V. D., Prabakar, S. M., Chelliah, M., et al. (2015). Environmental response and genomic regions correlated with rice root growth and yield under drought in the OryzaSNP panel across multiple study systems. *PloS One* 10, e0124127. doi: 10.1371/journal.pone.0124127

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 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Republic of Korea, 27 October–2 November. 3007–3016. New York: IEEE.

Wang, B. B., and Wu, B. (2022). Effect of exopolysaccharide from Leuconostoc pseudomesenteroides PC on seed germination of rice under different abiotic stresses. *Jiangsu. Agric. Sci.* 50, 59–64. doi: 10.15889/j.issn.1002-1302.2022.24.008

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, 8–14 September. 3–19. Switzerland: Springer, Cham.

Yang, W., Feng, H., Zhang, X., Zhang, J., Doonan, J. H., Batchelor, W. D., et al. (2020). Crop phenomics and high-throughput phenotyping: past decades, current challenges, and future perspectives. *Mol. Plant* 13, 187–214. doi: 10.1016/j.molp.2020.01.008

York, L. M., Galindo-Castañeda, T., Schussler, J. R., and Lynch, J. P. (2015). Evolution of US maize (Zea mays L.) root architectural and anatomical phenes over the past 100 years corresponds to increased tolerance of nitrogen stress. *J. Exp. Bot.* 66, 2347–2358. doi: 10.1093/jxb/erv074

Zhang, X., Zhou, X., Lin, M., and Sun, J. (2017). *ShuffleNet: an Extremely Efficient Convolutional Neural Network for Mobile Devices. arXiv*, arXiv:1707.01083.

Zhao, C., Zhang, Y., Du, J., Guo, X., Wen, W., Gu, S., et al. (2019). Crop Phenomics: Current status and perspectives. *Front. Plant Sci.* 10, 714. doi: 10.3389/fpls.2019.00714

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 16 January 2025
DOI 10.3389/fpls.2024.1458978

OPEN ACCESS

EDITED BY
Mohsen Yoosefzadeh Najafabadi,
University of Guelph, Canada

REVIEWED BY
Xin Wu,
Beijing University of Posts and Telecommunications (BUPT), China
Peter Yuen,
Cranfield University, United Kingdom

*CORRESPONDENCE
Ling Zhou
zhouling401618@163.com

RECEIVED 03 July 2024
ACCEPTED 13 December 2024
PUBLISHED 16 January 2025

CITATION
Wang B, Chen G, Wen J, Li L, Jin S, Li Y, Zhou L and Zhang W (2025) SSATNet: Spectral-spatial attention transformer for hyperspectral corn image classification. *Front. Plant Sci.* 15:1458978. doi: 10.3389/fpls.2024.1458978

SSATNet: Spectral-spatial attention transformer for hyperspectral corn image classification

Bin Wang[1], Gongchao Chen[2], Juan Wen[3], Linfang Li[2], Songlin Jin[2], Yan Li[4], Ling Zhou[2*] and Weidong Zhang[2]

[1]School of Life Sciences, Henan Institute of Science and Technology, Xinxiang, China, [2]School of Information Engineering, Henan Institute of Science and Technology, Xinxiang, China, [3]School of Art, Henan University of Economics and Law, Zhengzhou, China, [4]School of Software, Henan Institute of Science and Technology, Xinxiang, China

Hyperspectral images are rich in spectral and spatial information, providing a detailed and comprehensive description of objects, which makes hyperspectral image analysis technology essential in intelligent agriculture. With various corn seed varieties exhibiting significant internal structural differences, accurate classification is crucial for planting, monitoring, and consumption. However, due to the large volume and complex features of hyperspectral corn image data, existing methods often fall short in feature extraction and utilization, leading to low classification accuracy. To address these issues, this paper proposes a spectral-spatial attention transformer network (SSATNet) for hyperspectral corn image classification. Specifically, SSATNet utilizes 3D and 2D convolutions to effectively extract local spatial, spectral, and textural features from the data while incorporating spectral and spatial morphological structures to understand the internal structure of the data better. Additionally, a transformer encoder with cross-attention extracts and refines feature information from a global perspective. Finally, a classifier generates the prediction results. Compared to existing state-of-the-art classification methods, our model performs better on the hyperspectral corn image dataset, demonstrating its effectiveness.

KEYWORDS

corn identification, hyperspectral image classification, deep learning, morphology, image classification

1 Introduction

Hyperspectral imaging technology comprehensively measures an object's spectral properties by recording its absorption and reflection across various spectral bands (Li et al., 2024c; Zhang et al., 2024b; Li et al., 2024a). The resulting hyperspectral images, composed of multiple consecutive bands, are rich in feature information and can

thoroughly reveal the nature of the object. This technology advances intelligent agriculture by utilizing the detailed feature information in hyperspectral images, thereby avoiding the destructive methods of traditional seed identification. Hyperspectral imaging has gradually been applied to intelligent agriculture, geological exploration, and medical treatment, offering new development opportunities and technical capabilities.

The increasing variety of corn seeds available in the market presents a significant challenge to the cereal farming industry, making the accurate identification of corn varieties especially crucial. Recently, researchers have been investigating hyperspectral image classification techniques using machine learning and deep learning approaches (Zhang et al., 2023c; Wu et al., 2022). Ahmad et al. (Ahmad et al., 2019). utilized a self-encoder paired with a multilayer extreme learning machine to mitigate high computational overhead and the Thuesian phenomenon in hyperspectral images, which improved the accuracy of hyperspectral image classification. Okwuashi et al. (Okwuashi and Ndehedehe, 2020) introduced a deep support vector machine algorithm incorporating four kernel functions and demonstrated its effectiveness in hyperspectral image classification using publicly available datasets. Zhang et al. (Zhang et al., 2020) employed a deep forest model with hyperspectral imaging to classify rice seeds with different levels of frost damage in small sample datasets. Su et al. (Su et al., 2022) introduced a new semi-supervised method for hyperspectral image classification that integrates normalized spectral clustering with kernel learning, effectively addressing the issues of relevant features appearing in non-adjacent regions and the lack of non-Euclidean spatial correlation. Jin et al. (Jin et al., 2023) developed a cost-sensitive K-neighborhood algorithm to reduce noise interference, enhance spatial information utilization, and achieve robust performance in hyperspectral wheat image classification. Farmonov et al. (Farmonov et al., 2023) employed wavelet transform for feature extraction, combined with random forests and support vector machine algorithms, to localize crops in farmland and classify crop hyperspectral images, playing a significant role in crop growth monitoring and harvest prediction. Sim et al. (Sim et al., 2024) combined machine learning algorithms with hyperspectral imaging for fast, non-destructive detection of coffee origin without sample processing. Wang et al. (Wang et al., 2024b) proposed a cross-domain few-shot learning strategy utilizing a two-branch domain adaptation technique to mitigate distortion caused by enforcing different domain alignments, achieving effective cross-domain transfer learning for low/high spatial resolution data. Although machine learning methods have demonstrated exemplary performance in hyperspectral image classification, their reliance on manual or semi-automatic feature extraction limits their potential. The emergence of deep learning methods has enabled the automatic extraction of spectral, spatial and spatial-spectral features from hyperspectral images, leading to significant advancements in this field.

Zhang et al. (Zhang et al., 2019) created a straightforward 1D convolutional capsule network to tackle the high dimensionality and limited labeled samples in hyperspectral images, achieving effective feature extraction and classification. Wang et al. (Wang et al., 2020) developed an end-to-end cubic convolutional neural network that integrates Principal Component Analysis with 1D convolution for efficient extraction of spatial and spectral features. Roy et al. (Roy et al., 2020) proposed an improved residual network using an adaptive spatial-spectral kernel with attention mechanisms, utilizing 3D convolutional kernels to simultaneously extract spatial and spectral features, achieving excellent classification results. Cui et al. (Cui et al., 2021) introduced a lightweight deep network using 3D deep convolution to classify hyperspectral images with fewer parameters and lower computational costs. Ortac et al. (Ortac and Ozcan, 2021) evaluated the performance of 1D, 2D, and 3D convolutions in hyperspectral image classification, demonstrating that 3D convolution offers superior feature extraction capabilities. Ghaderizadeh et al. (Ghaderizadeh et al., 2021) employed depth-separable and fast convolutional blocks in combination with 2D convolutional neural networks to effectively tackle data noise and insufficient training samples. Paoletti et al. (Paoletti et al., 2023a) proposed a channel attention mechanism to automatically design and optimize convolutional neural networks, reducing the computational burden in feature extraction while obtaining effective classification outcomes. Sun et al. (Sun et al., 2023) introduced an extensive kernel spatial-spectral attention network designed to efficiently leverage 3D spatial-spectral features, maintaining the 3D structure of hyperspectral images. Jia et al. (Jia et al., 2023) developed a structure-adaptive CNN for hyperspectral image classification, which employs structure-adaptive convolution and mean pooling to extract deep spectral, spatial, and geometric features from a uniform hyperpixel region. Gao et al. (Gao et al., 2023) designed a lightweight 3D-2D multigroup feature extraction module for hyperspectral image classification, which mitigates the loss of crucial details in single-scale feature extraction and the high computational expense of multiscale extraction. Zhang et al. (Zhang et al., 2023b) introduced a method combining 3D and 2D convolution to fully utilize the spatial, texture and spectral features of hyperspectral data for the task of identifying wheat varieties. In conclusion, while 2D and 3D convolutions effectively capture spectral and spatial features from hyperspectral data, traditional convolutional neural networks are limited by high computational complexity and insufficient feature utilization, impacting their classification performance.

Inspired by (Vaswani et al., 2017), researchers have suggested a Transformer-based network model for image classification (Zhang et al., 2024a). Hong et al. (Hong et al., 2021) effectively classified hyperspectral remote sensing images by leveraging spectral local sequence information from neighboring bands, considering the temporal properties, and designing cross-layer skipping connections combined with the Transformer structure. Roy et al. (Roy et al., 2021) introduced an innovative end-to-end deep learning framework, using spectral and spatial morphological blocks for nonlinear transformations in feature extraction. Yang et al. (Yang et al., 2022) integrated convolutional operations into the Transformer structure to capture local spatial context and subtle spectral differences, fully utilizing the sequence attributes of spectral features. Sun et al. (Sun et al., 2022b) developed a spatial-spectral feature tokenization converter to capture both spectral-spatial and high-level semantic features, achieving hyperspectral image classification through a feature transformation module, a feature extraction module, and a sample label learning module. Kumar et al. (Kumar et al., 2022) developed a novel morphology-expanding convolutional neural network that connects the

morphological feature domain with the original hyperspectral data, reducing computational complexity and achieving good classification results. Peng et al. (Peng et al., 2022) developed a two-branch spectral-spatial converter with cross-attention, using spatial sequences to extract spectral features and capture deep spatial information to establish interrelationships among spectral sequences. Tang et al. (Tang et al., 2023) introduced a dual-attention Transformer encoder based on the Transformer backbone network for hyperspectral image classification, effectively extracting global dependencies and local spatial information between spectral bands. Qi et al. (Qi et al., 2023a) embedded 3D convolution in a two-branch Transformer structure to capture globally and locally correlated spectral-spatial domain features, demonstrating good performance for hyperspectral image classification through validation. Qiu et al. (Qiu et al., 2023) proposed a cross-channel dynamic spectral-spatial fusion Transformer capable of extracting multi-channel and multi-scale features, using multi-head self-attention to extract cross-channel global features and enhancing spatial-spectral joint features for hyperspectral image classification. Sun et al. (Sun et al., 2024) converted the spatial-spectral features into a memory marker storing *a priori* knowledge into an in-memory tagger, using a memory-enhanced Transformer encoder for the hyperspectral image classification task. Ahmad et al. (Ahmad et al., 2024) designed a Transformer-based network for hyperspectral image classification by combining wavelet transform with downsampling. The wavelet transform performs reversible downsampling, enabling attentional learning while preserving data integrity. Based on these studies, we propose utilizing a combination of 2D-3D convolution and Transformer, leveraging spectral-spatial morphological features to identify hyperspectral corn seed varieties. The contributions of this paper can be summarized as follows:

- We developed a 3D-2D convolutional cascade structure that autonomously extracts contextual features, reduces data complexity and efficiently captures high-level abstract features for integration into the Transformer architecture.
- We introduced a spectral-spatial morphology structure that employs expansion and erosion operations for spectral-spatial morphology convolution, enhancing the understanding of the data's intrinsic properties.
- We employed a Transformer Encoder with CrossAttention to comprehensively extract and refine feature information from hyperspectral corn images on a global scale using the attention mechanism.

2 Related works

Currently, researchers have proposed a variety of methods for classifying hyperspectral remote sensing images and hyperspectral seed images. We classify these approaches into deep learning methods, machine learning methods and traditional methods. The deep learning methods are further divided into hybrid CNN-Transformer methods, Transformer-based methods, and CNN-based methods. Next, we overview and summarize these research outcomes.

Traditional methods for hyperspectral image classification primarily rely on analyzing physical and statistical features. These methods typically include spectral feature extraction, pixel-based classification, and target-based classification. For example, Cui et al. (Cui et al., 2020) introduced a super-pixel and multi-classifier fusion approach to tackle the challenges of limited labeled samples and substantial spectral variations. Similarly, Chen et al. (Chen et al., 2021a) introduced a feature extraction means that combines PCA and LBP, optimized using the Gray Wolf optimization algorithm for hyperspectral image classification. While these methods perform well for simpler classification tasks, their effectiveness diminishes when faced with complex backgrounds and highly mixed pixels.

Machine learning methods effectively classify hyperspectral images by learning the features of sample data. With the advancement of machine learning technology, researchers increasingly utilize machine learning algorithms for hyperspectral image classification. For example, Pham et al. (Pham and Liou, 2022) developed a push-sweep hyperspectral system using a support vector machine to date surface defects, addressing the problem of insufficient accuracy and speed in detecting date skin defects with traditional methods. Sun et al. (Sun et al., 2022a) constructed a network integrating multi-feature and multi-scale extraction with a swift and efficient kernel-extreme learning machine for rapid classification, significantly enhancing hyperspectral image classification accuracy. Wang et al. (Wang et al., 2023b) proposed a capsule vector neural network that combines capsule representation of vector neurons with an underlying fully convolutional network, achieving good classification performance with insufficient labeled samples. Compared to traditional methods, machine learning approaches handle high-dimensional data more effectively and achieve higher classification accuracy. However, these methods still rely on human-designed feature extraction and selection, preventing them from fully utilizing all the information in hyperspectral data.

Deep learning methods excel in hyperspectral image classification due to their automatic feature extraction and end-to-end learning capability (Zhang et al., 2024c; Hong et al., 2023). These methods can be categorized into hybrid CNN-Transformer methods, Transformer-based methods, and CNN-based methods.

CNN-based methods are designed to capture spectral and spatial features through convolutional layers specifically tailored for hyperspectral data, significantly improving classification performance (Wu et al., 2021). Yang et al. (Yang et al., 2021) introduced a spatial-spectral cross-attention network that suppresses redundant data bands and achieves robust, accurate classification. Yu et al. (Yu et al., 2021) developed a spectral-spatial dense convolutional neural network framework with a feedback attention mechanism to tackle issues of high complexity, information redundancy, and inefficient description, thereby improving classification efficiency and accuracy. Zheng et al. (Zheng et al., 2022) developed a rotationally invariant attention network for pixel feature class recognition, leveraging spectral features and spatial information. Paoletti et al. (Paoletti et al., 2023b) created a channel attention mechanism to automatically design and

optimize a CNN, integrating 1D and spectral-spatial (3D) classifiers to process data from various perspectives while reducing computational overhead. Guo et al. (Guo et al., 2023) introduced a dual-view global spatial and spectral feature fusion network that efficiently extracts spectral-spatial features from hyperspectral images, accounting for global and local information.

Transformer-based methods excel at capturing long-range dependencies and complex features in hyperspectral images through a self-attention mechanism. Huang et al. (Huang et al., 2022) introduced a 3D swin transformer that captures rich spatial-spectral information, learns semantic representations from unlabeled data, and overcomes traditional methods' limitations regarding receptive fields and labeling requirements. Yu et al. (Yu et al., 2022) proposed a multilevel spatial-spectral transformer network that processes hyperspectral images into sequences, addressing issues faced by CNN-based methods such as limited receptive fields, information loss in downsampling layers, and high computational resource consumption. Zhang et al. (Zhang et al., 2023d) developed a location-lightweight multi-head self-attention module and a channel-lightweight multi-head self-attention module, allowing each channel or pixel to associate with global information while reducing memory and computational burdens. Zhao et al. (Zhao et al., 2023) proposed an active learning hyperspectral image classification framework using an adaptive super-pixel segmentation and multi-attention transformer, achieving good classification performance with small sample sizes. Wang et al. (Wang et al., 2023a) introduced a trispectral image generation channel that converts hyperspectral images into high-quality trispectral images, mitigating the spatial variability problem caused by complex imaging conditions. Compared to CNNs, transformers have significant advantages in processing global and multi-scale features, allowing for better handling of global information in hyperspectral images.

Methods that hybrid CNN and Transformer aim to utilize the strengths of both to enhance hyperspectral image classification performance. These hybrid methods typically employ Transformers to capture global dependencies and CNNs to extract local spatial features. Zhang et al. (Zhang et al., 2022a) designed a dual-branch structure combining Transformer and CNN branches, effectively extracting both global hyperspectral features and local spectral-spatial features, resulting in high classification accuracy. Zhang et al. (Zhang et al., 2023a) proposed a network that integrates Transformer and multiple attention mechanisms, utilizing spatial and channel attention to focus on salient information, thereby enhancing spatial-spectral feature extraction and semantic understanding. Qi et al. (Qi et al., 2023b) introduced a global-local 3D convolutional Transformer network, embedding a dual-branch Transformer in 3D convolution to simultaneously capture global-local correlations across spatial and spectral domains, addressing the restricted receptive field issue of traditional CNNs. Xu et al. (Xu et al., 2024) proposed a two-branch convolutional Transformer network based on 3D CNN and an improved Transformer encoder, integrating spatial and local-global spectral features with lower computational complexity. Chen et al. (Chen et al., 2024) developed the TCCU-Net, a two-stream collaborative network that learns spatial, spectral, local and global information end-to-end for effective hyperspectral unmixing. This integration enables the model to leverage both spectral and spatial information from hyperspectral images more comprehensively, enhancing classification robustness and accuracy.

3 Methodology

The network flowchart of our proposed Spectral-Spatial Attention Transformer for hyperspectral corn image classification is shown in Figure 1. It contains 3D-2D Convolutional Module, Spectral-Spatial Morphology, Transformer Encoder with CrossAttention, and Classifier.

3.1 Motivation

With the development of intelligent agriculture, the integration of hyperspectral imaging technology and deep learning has gained widespread application in crop research, particularly in seed classification and identification. As a globally important food crop, the classification of corn seeds is significant for improving agricultural productivity and preserving crop genetic resources. Hyperspectral images can capture reflectance features at different wavelengths, providing researchers with rich spectral information for more precise seed classification and quality assessment (Chang et al., 2024).

In recent years, transformer models have emerged as popular in computer vision due to their powerful feature extraction and representation capabilities (Han et al., 2023; Li et al., 2024b). Compared to traditional convolutional neural networks, transformers are better at handling high-dimensional data and capturing long-range dependencies, which are crucial for extracting complex features from hyperspectral images. Additionally, the self-attention mechanism of Transformers enables the model to flexibly focus on important areas within the image, thereby enhancing classification accuracy. Consequently, choosing Transformer-based methods allows for more effective utilization of hyperspectral data, providing more reliable support for corn seed classification.

3.2 3D-2D convolution module

In hyperspectral image classification, effective feature extraction is vital for improving accuracy. Both 3D and 2D convolutions are widely used in this domain due to their unique advantages. 3D convolution simultaneously operates in spectral and spatial dimensions, capturing their correlation. Unlike traditional 2D or 1D convolutions, 3D convolution provides richer feature descriptions and retains more original spectral and spatial information, thus enhancing classification accuracy. It fully leverages the three-dimensional data structure of hyperspectral images, avoiding information loss or oversimplification. However, as network depth and input data size increase, the computational complexity and memory requirements of 3D convolution rise significantly, demanding higher hardware resources and more

10.3389/fpls.2024.1458978

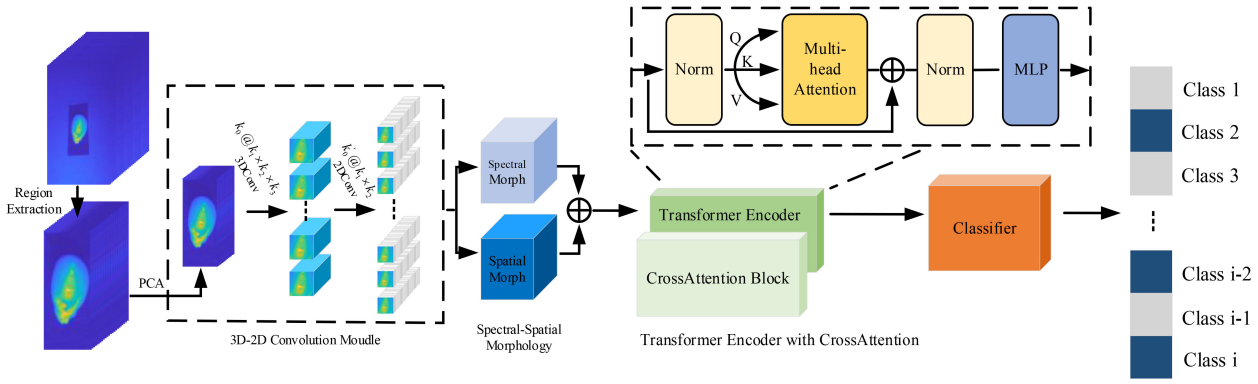


FIGURE 1
Flowchart of the spectral-spatial attention Transformer for hyperspectral corn image classification. Initially, the data are preprocessed with region of interest extraction and PCA dimensionality reduction. Subsequently, local spatial, spectral, and texture features are extracted using 2D and 3D convolutions. The spectral and spatial morphology modules further analyze the internal structure of the data. The Transformer encoder with cross-attention then extracts and refines the feature information from a global perspective. Finally, the classifier provides the prediction results.

training time. 2D convolution, on the other hand, has lower computational complexity and high efficiency, as it operates on two-dimensional space (width and height). It effectively utilizes spatial and texture information, making it suitable for handling local features and texture details in hyperspectral images. Combining 3D and 2D convolutions can efficiently leverage the strengths to extract features from hyperspectral corn images. 3D convolution captures complex spectral-spatial relationships, while 2D convolution extracts local spatial features and texture information, maintaining computational efficiency. This combination optimizes feature extraction, leading to improved classification performance.

3D convolution is mainly used for three-dimensional data processing, extracting features by sliding a convolution kernel across the three dimensions of the input data. Suppose the input data is $I^{D\times H\times W\times C}$, where C is the number of channels, W is the width, H is the height, and D is the depth (spectral dimension).The dimensions of the 3D convolution kernel are $K_d \times K_h \times K_w \times C \times N$, where N is the number of output channels (i.e., the number of convolution kernels), C is the number of input channels, K_w is the size in the width direction, K_h is the size in the height direction, and K_d is the size of the convolution kernel in the depth direction. For an input tensor I and a convolution kernel W, the output tensor Y of the 3D convolution can be expressed as

$$Y(n,d,h,w) = \sum_{c=0}^{C-1}\sum_{k_d=0}^{K_d-1}\sum_{k_h=0}^{K_h-1}\sum_{k_w=0}^{K_w-1} I(c,d+k_d,h+k_h,w+k_w) \times W(n,c,k_d,k_h,k_w) + b(n) \quad (1)$$

where $I(c,d+k_d,h+k_h,w+k_w)$ is the value of the input tensor I at channel c and position $(d+k_d,h+k_h,w+k_w)$. $W(n,c,k_d,k_h,k_w)$ represents the weight of the convolution kernel W at output channel n and input channel c, positioned at (k_d,k_h,k_w). $b(n)$ is the bias term for each output channel n in the convolutional layers. It is initialized with random values (typically small values close to zero) and then adjusted during training via backpropagation. The gradient of the loss with respect to the bias is computed and used to update $b(n)$, just like the weights of the convolutional filters. This adjustment allows the model to shift the activations of each channel, enabling the network to adapt to various patterns in the data and improve its representation of features.

2D convolution is applied to 2D data processing, extracting features by sliding a convolution kernel (filter) across the two dimensions of the input data. Assuming the input data is $I^{H\times W\times C}$, the 2D convolution kernel has dimensions $K_h \times K_w \times C \times N$, with the parameter presentation consistent with that of 3D convolution. For an input tensor I and a convolution kernel W, the output tensor Y of the 2D convolution can be expressed as

$$Y(n,i,j) = \sum_{c=0}^{C-1}\sum_{k_h=0}^{K_h-1}\sum_{k_w=0}^{K_w-1} I(c,i+k_h,j+k_w) \times W(n,c,k_h,k_w) + b(n) \quad (2)$$

where $I(c,i+k_h,j+k_w)$ is the value at position $(i+k_h,j+k_w)$ in the input tensor I at channel c. $W(n,c,k_h,k_w)$ represents the weight of the convolutional kernel W at position (k_h,k_w) for output channel n and input channel c.

3.3 Spectral-spatial morphology module

Hyperspectral images contain abundant textural, spatial, and spectral information. Morphology, a nonlinear image processing technique, is mainly used to analyze and manipulate the shape and structure of images. In hyperspectral image processing, morphological methods can effectively extract spatial and spectral features, enhancing the robustness and accuracy of image classification. Building on this, we integrate morphology with 2D convolution to locally manipulate images using structural elements, which can highlight or suppress specific shape features.

Spatial features can be extracted from each spectral band of a hyperspectral corn image through morphological operations like dilation and erosion. The dilation operation can emphasize the bright areas in the image and expand the edges of the target object,

10.3389/fpls.2024.1458978

making the morphological features of the corn seed more pronounced. The computational expression for dilation is as

$$D(I) = I \oplus B = \bigcup_{b \in B}(I + b) \tag{3}$$

where I denotes the input image, B is the structural element (a small template used to detect the morphological features of the image), $\oplus$ stands for the dilation operation, $\bigcup_{b \in B}()$ represents the union of all structural element positions to take the maximum value, and + denotes the pixel displacement operation. b influences the dilation and erosion operations. These operations involve shifting and adjusting the shape of features within the image, where b helps control the degree of expansion (dilation) or contraction (erosion). Like the convolutional biases, the values of b in these operations are also learned during training, refining the model's ability to capture spatial relationships and remove irrelevant details in the data. Conversely, the erosion operation removes noise and small bright spots, resulting in a smoother and more uniform target area. The computational expression for erosion is as

$$E(I) = I \ominus B = \bigcap_{b \in B}(I - b) \tag{4}$$

where $\ominus$ denotes the erosion operation, $\bigcap_{b \in B}()$ represents the intersection operation to take the minimum value for all structural element positions, and − indicates the negative displacement operation of pixels. Performing these operations on each spectral band extracts subtle spatial variations and enhances the representation of spatial features. Subsequently, these spatial features are combined with spectral features to fully utilize the spectral and spatial information in hyperspectral images. Specifically, we apply morphological operations to each spectral band to extract spatial features. These spatial features are merged with the original spectral information to construct high-dimensional feature vectors. This method preserves the spectral information of the hyperspectral image while enhancing the representation of spatial structure information. The feature extraction and classification effectiveness is further improved by integrating these morphological operations with 2D convolution. 2D convolution extracts local spatial features within each spectral band and enhances the representation of spatial information. These two convolutional operations complement each other, allowing the features, preprocessed through morphological operations, to be input into the convolutional neural network for more accurate classification.

The bias b in these equations plays a crucial role in adjusting the output activations, improving the feature extraction process. In the convolutional operations (Equations 1, 2), it allows the network to adapt to various activation patterns, enhancing the model's ability to learn more complex relationships in the data. In the morphological operations (Equations 3, 4), it enhances spatial feature representation by refining the shapes and structures in the image. This combination of accurate feature extraction and refinement leads to better corn seeds classification performance.

By integrating morphological and convolutional techniques, we substantially enhance hyperspectral corn image classification accuracy and robustness. This combined approach boosts classification performance and improves resilience against complex backgrounds and noise.

3.4 Transformer encoder with CrossAttention module

The Transformer encoder enhances input data representation through a sophisticated attention module that captures dependencies among different parts of the input sequence. Figure 2 depicts the detailed structure of this attention module, consisting of two primary components: multi-head self-attention and scaled dot-product attention.

Originally, the Transformer architecture was designed for natural language processing, particularly for handling sequence data, and it excels in this domain due to its multiple self-attention core blocks. Unlike conventional Convolutional Neural Networks and Recurrent Neural Networks, the Transformer exclusively utilizes the attention mechanism, enabling efficient capture of global dependencies in sequential data. The input sequence is initially converted into a fixed-dimensional vector representation via an embedding layer, with positional information preserved through positional encoding, which is generated by sine and cosine functions.

Each encoder layer includes multiple self-attention heads, each independently processing the input sequence to generate an attention representation, which is then concatenated and integrated through a linear transformation. The multi-head self-attention mechanism enables the model to attend to multiple parts of the input sequence simultaneously. Specifically, the input sequence is represented as a key (K), query (Q), and value (V). Multiple sets of Q, K, and V are created through the linear projection of a learned weight matrix. Each set of Q, K, and V is passed to the scaled dot-product attention mechanism, where attention scores are calculated and applied to the values. The Q is multiplied by the transposed key K^T to obtain the raw attention score, which is then divided by the square root of the key's dimension, $\sqrt{d_k}$, to maintain gradient stability. The computational process of self-attention can be summarized as

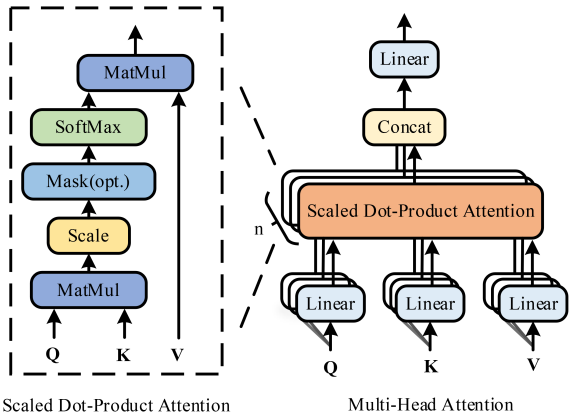


FIGURE 2
Diagram illustrating the structure of the multi-head attention mechanism and scaled dot-product attention.

$$\mathrm{SA} = \mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right)V \tag{5}$$

Through its unique multi-head self-attention mechanism and feed-forward neural network, the Transformer structure efficiently captures global dependencies and improves the classification accuracy of hyperspectral corn images.

3.5 Loss function

In this paper, we propose a method that combines spectral-spatial morphology with a 3D-2D convolutional Transformer network to classify hyperspectral corn images. This approach fully utilizes the spatial and spectral features of hyperspectral images. To optimize model performance, we employ the CrossEntropyLoss function.

The CrossEntropyLoss function is commonly used in classification tasks, especially for multi-class classification problems. It measures the discrepancy between the true category distribution and the predicted probability distribution by computing the negative log-likelihood between the actual labels and the predicted probabilities. This function ensures numerical stability by converting the output into a probability distribution using the Softmax function. Additionally, the gradient of the CrossEntropyLoss function is relatively easy to compute, facilitating the implementation of the back-propagation algorithm and model optimization. By directly quantifying the alignment between predicted probabilities and actual labels, it accurately reflects the performance of the classification model. Consequently, we apply the CrossEntropyLoss function to the hyperspectral corn image classification task. Its computational expression is as

$$\mathrm{CrossEntropyLoss} = -\sum_{i=0}^{N} y_i \log(\hat{y}_i) \tag{6}$$

where y_i represents the true label of the sample, N is the total number of samples, and $\hat{y}_i$ is the predicted probability from the model. The network model converts the output to a probability distribution using the Softmax function

$$\hat{y}_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{7}$$

where z_i represents the linear output of the model. For a given category c, the true label $y_c = 1$ while the labels for all other categories are 0. The predicted probability $\hat{y}_i$ corresponding to the true label y_i is substituted into Equation 6, and the loss value for each sample is

$$\mathrm{Loss} = -\sum_i y_i \log(\hat{y}_i) \tag{8}$$

By measuring the difference between actual and predicted labels and updating the model parameters through the backpropagation algorithm to minimize the loss, this approach effectively guides the model in learning to handle complex hyperspectral corn image features. Consequently, it improves both the classification accuracy and robustness.

4 Experiment and analysis

In this section, we will first discuss the dataset used, detail the specific implementation of SSATNet, and then present the evaluation metrics, multi-classification results, and ablation study.

4.1 Experimental dataset

To verify the effectiveness of the SSATNet, we utilized the hyperspectral corn image dataset from SSTNet (Zhang et al., 2022b). This dataset contains 10 corn varieties, each with 120 samples. The collected images cover a spectral range from 400 to 1000 nm, encompassing 128 bands. To reduce computational overhead and focus on retaining only the core area of the corn seeds, the collected raw data resolution of 696 × 520 was reduced to 210 × 200 for feature extraction. The corn seed images were sourced from planting areas in Henan Province, including varieties such as FengDa601, BaiYu9284, BaiYu8317, BaiYu918, BaiYu897, BaiYu879, BaiYu833, BaiYu818, BaiYu808, and BaiYu607. Figure 3 shows different spectral band maps of a sample randomly selected from FengDa601, BaiYu818, and BaiYu833. This corn image dataset was obtained by contacting the authors.

4.2 Implementation details

The hyperspectral corn image dataset includes 10 varieties, totaling 1200 samples, divide into training and test sets in a 4:1 ratio. We conducted our experiments on a Windows 10 PC with an Intel® Xeon® Gold 5218 CPU @ 2.30GHz x64, an NVIDIA GeForce RTX 3090*2 graphics card, and 256 GB RAM. The Batch size is set to 16 for the training and 8 for the testing. We used Adamax as the optimizer with a learning rate of 0.01, an exponential decay rate of 0.9, a gradient squared moving average rate of 0.999, and 250 iterations. Additionally, we implemented a Dropout mechanism that randomly deactivates 10% of nodes, effectively preventing overfitting.

4.3 Evaluation metrics

To thoroughly assess the performance of our SSATNet in classifying hyperspectral corn images, we employ four standard evaluation metrics: F1-Score, Recall, Precision, and the Kappa coefficient(K_A). Precision assesses the accuracy of the classification model by evaluating the proportion of instances predicted to be positive that are actually positive. There exists a trade-off between Precision and Recall; increasing Precision may lead to a decrease in Recall and vice versa. Therefore, the F1-Score, derived as the harmonic mean of Precision and Recall, is often used for a more balanced evaluation of model performance, and its calculation expression is shown in Equation 9. The K_A is a consistency test metric that evaluates the agreement between the classified image and the reference image in hyperspectral remote

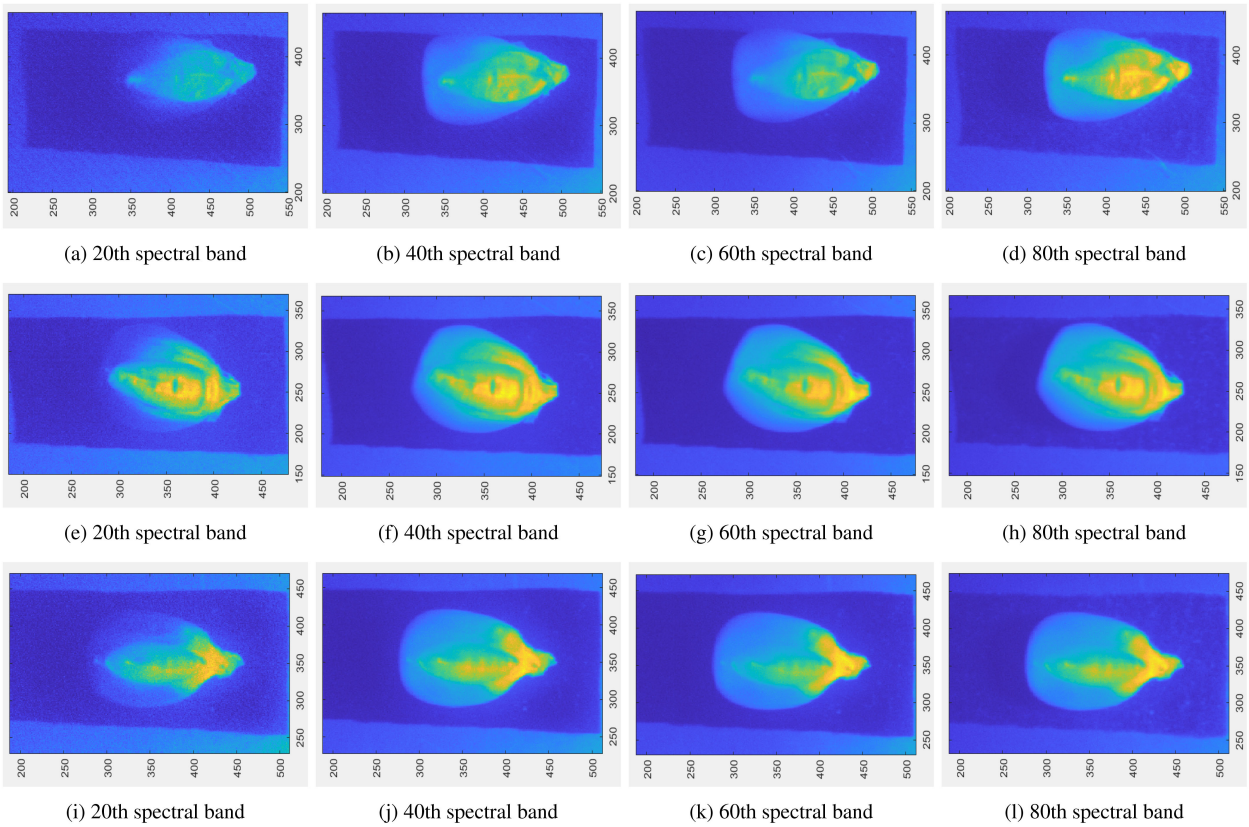


FIGURE 3
Randomly select a sample from three corn varieties, FengDa601 **(A–D)**, BaiYu818 **(E–H)**, and BaiYu833 **(I–L)**, and display their partial spectral bands.

sensing classification tasks, providing a more comprehensive reflection of the overall classification accuracy. Higher scores in these four evaluation metrics indicate better model performance. Figure 4 shows the confusion matrix of our model's classification results for hyperspectral corn images and the results of one of the training and testing sessions.

$$\text{F1} - \text{Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

4.4 Multi-classification results

Extensive experiments were performed to thoroughly test the generalization and effectiveness of our model for hyperspectral corn image classification. The comparison methods include KNN (Kumbure et al., 2020), SGD (Lei and Tang, 2021), RFA (Chen et al., 2021b), HybridNet (Roy et al., 2019), SSTNet (Zhang et al., 2022b), CTMixer (Zhang et al., 2022a), MSTNet (Yu et al., 2022), MATNet (Zhang et al., 2023a), and 3DCT (Wang et al., 2024a). The experimental results are presented in Table 1. The source code and parameters for the comparison methods were acquired from the original authors.

The results presented in Table 1 demonstrate the performance of various methods on the hyperspectral corn images dataset. Traditional machine learning models such as KNN (Kumbure et al., 2020), RFA (Chen et al., 2021b), and SGD (Lei and Tang, 2021) show subpar performance across all evaluation metrics, with RFA (Chen et al., 2021b) performing the worst across all metrics. These traditional models, lacking nonlinear activation mechanisms, struggle to extract deep spectral-spatial features effectively. In contrast, HybridNet (Roy et al., 2019), SSTNet (Zhang et al., 2022b), and 3DCT (Wang et al., 2024a), which integrate 3D convolution, demonstrate superior results due to their ability to capture spectral and spatial features simultaneously. Models like CTMixer (Zhang et al., 2022a), MSTNet (Yu et al., 2022), and MATNet (Zhang et al., 2023a) further leverage the Transformer architecture to address the complex relationships inherent in hyperspectral data. Our proposed model, which combines convolutional networks with Transformers and incorporates a novel spectral-spatial attention mechanism, achieves the best overall performance across all metrics. The integration of local and global feature extraction methods allows our model to substantially improve Precision, Recall, F1-Score, and K_A, surpassing existing state-of-the-art methods. These results validate the effectiveness of our design in capturing the complex spectral-spatial features of hyperspectral corn images and its superior ability to generalize to high-dimensional datasets.

4.5 Ablation study

To further evaluate the contribution of each module in SSATNet to the classification performance of hyperspectral corn seed images, we conducted ablation experiments on the dataset introduced by SSTNet

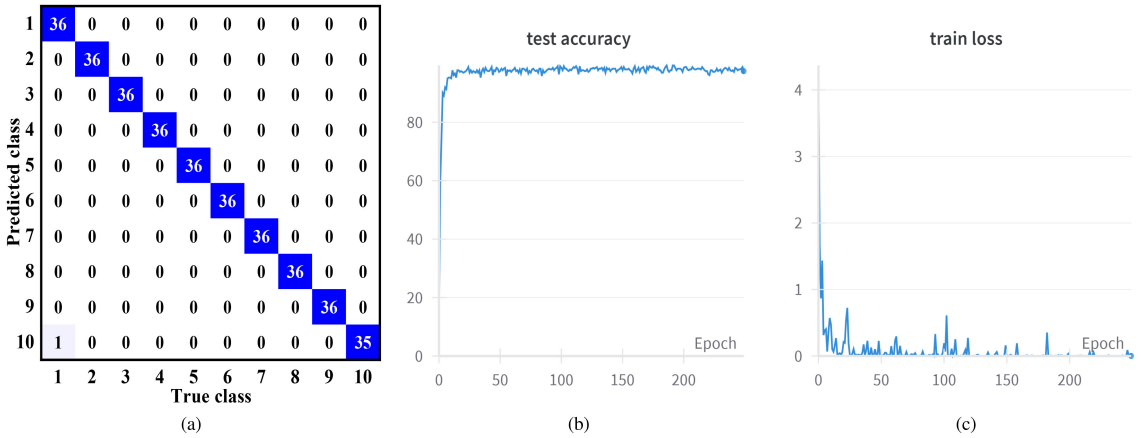


FIGURE 4
(A) The confusion matrix of our SSATNet classification results. **(B)** The results of one of the testing. **(C)** The results of one of the training.

TABLE 1 Test results of various methods on the hyperspectral corn images dataset.

Models	Hyperspectral Corn images			
	Precision	Recall	F1-Score	K_A
KNN (Kumbure et al., 2020)	96.12 ± 0.35	95.72 ± 0.32	95.90 ± 0.24	0.9675 ± 0.011
SGD (Lei and Tang, 2021)	96.98 ± 0.28	96.50 ± 0.18	96.70 ± 0.21	0.9721 ± 0.008
RFA (Chen et al., 2021b)	94.50 ± 0.40	94.10 ± 0.38	94.22 ± 0.39	0.9519 ± 0.009
HybridNet (Roy et al., 2019)	96.72 ± 0.30	96.44 ± 0.28	96.34 ± 0.21	0.9772 ± 0.007
SSTNet (Zhang et al., 2022b)	98.12 ± 0.18	97.78 ± 0.15	97.95 ± 0.17	0.9887 ± 0.005
CTMixer (Zhang et al., 2022a)	97.38 ± 0.33	97.75 ± 0.30	97.20 ± 0.32	0.9827 ± 0.008
MSTNet (Yu et al., 2022)	97.00 ± 0.38	96.95 ± 0.35	96.80 ± 0.36	0.9802 ± 0.009
MATNet (Zhang et al., 2023a)	98.27 ± 0.16	98.34 ± 0.14	98.25 ± 0.15	0.9930 ± 0.004
3DCT (Wang et al., 2024a)	98.30 ± 0.28	98.12 ± 0.25	98.19 ± 0.27	0.9928 ± 0.004
Our	**98.65 ± 0.18**	**98.57 ± 0.15**	**98.60 ± 0.17**	**0.9965 ± 0.003**

Optimal, bolded; Suboptimal, blue.

(Zhang et al., 2022b). In these experiments, we systematically removed individual components of the network while retaining the remaining modules unchanged. Specifically, we excluded the following components: 1) the 3D convolution module (-w/o 3DConv); 2) the 2D convolution module (-w/o 2DConv); 3) the spectral morphology structure (-w/o SpectralMorph); and 4) the spatial morphology structure (-w/o SpatialMorph). The Table 2 below illustrates the quantitative analysis metrics for each ablation experiment. The results demonstrate that the removal of the 3D convolution module leads to the most significant degradation in performance, underscoring its crucial role in capturing both spectral and spatial features in hyperspectral corn seed images. Without 3D convolution, the model's ability to integrate spatial-spectral correlations is substantially weakened. Similarly, the removal of the 2D

TABLE 2 Quantitative test results of ablation experiments.

Module	Precision	Recall	F1-Score	K_A
-w/o 3DConv	86.42 ± 0.31	87.33 ± 0.29	87.05 ± 0.36	0.8768 ± 0.006
-w/o 2DConv	89.51 ± 0.25	90.35 ± 0.25	90.52 ± 0.29	0.9117 ± 0.004
-w/o SpectralMorph	93.65 ± 0.22	93.27 ± 0.19	93.86 ± 0.25	0.9408 ± 0.004
-w/o SpatialMorph	92.59 ± 0.20	92.69 ± 0.21	92.31 ± 0.21	0.9332 ± 0.005
SSATNet (full model)	**98.65 ± 0.18**	**98.57 ± 0.15**	**98.60 ± 0.17**	**0.9965 ± 0.003**

Optimal, bolded.

convolution module also causes a noticeable decline in performance, although to a lesser extent compared to the absence of 3D convolution. This is because 2D convolution primarily focuses on extracting local spatial features and refining feature representations. The exclusion of the spectral morphology structure results in performance degradation, highlighting its importance in enhancing spectral feature representation and managing the complex spectral relationships inherent in hyperspectral data. Likewise, the spatial morphology structure significantly contributes to the model's performance by extracting and enhancing spatial features, enabling more accurate classification of corn seed images.

In summary, each module is crucial to the overall performance of SSATNet. The 3D convolution module provides the most significant enhancement to classification performance, followed by the spectral morphology structure and the spatial morphology structure. The 2D convolution module also provides substantial support in refining feature representation. Through the synergy of these modules, SSATNet excels in the hyperspectral corn seed classification task, demonstrating the effectiveness of its design.

5 Conclusion

In this paper, we propose the SSATNet method for non-destructive identification of hyperspectral corn varieties. First, we design a 3D-2D cascade structure to reduce image data complexity and effectively extract local feature information, facilitating the Transformer structure's processing. Additionally, we introduce a spectral-spatial morphology structure combined with 2D convolution to perform expansion and erosion operations on the data, providing a deeper understanding of the data's nature. Finally, we employ the Transformer structure to extract global feature information from hyperspectral corn images through the self-attention mechanism, achieving efficient capture of global dependencies between corn spectra. Ablation experiments highlight the effectiveness of each component of SSATNet in extracting features and classifying hyperspectral corn images. This method offers a new approach to non-destructive corn variety identification and significantly promotes the development of intelligent agriculture.

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: Investigation, Methodology, Resources, Writing – original draft. GC: Resources, Writing – original draft. JW: Validation, Visualization, Writing – review & editing. LL: Writing – review & editing, Formal analysis, Validation, Investigation. SJ: Supervision, Writing – review & editing. YL: Funding acquisition, Supervision, Writing – review & editing. LZ: Funding acquisition, Methodology, Supervision, Writing – original draft. WZ: Funding acquisition, Investigation, Supervision, Writing – original draft.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported in part by the China Postdoctoral Science Foundation project under Grant 2024M750747, in part by the Henan Provincial Science and Technology Research and Development Joint Foundation Project under Grant 235200810066, in part by the Teacher Education Curriculum Reform Research of Henan Province under Grant 2024-JSJYYB-099, and in part by the Key Specialized Research and Development Program of Science and Technology of Henan Province under Grants 242102210075, 232102210018, 242102211048, 242102211059, 242102211030, 242102210126.

Acknowledgments

This brief text acknowledges the contributions of specific colleagues, institutions, or agencies that assisted the authors' efforts.

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

References

Ahmad, M., Ghous, U., Usama, M., and Mazzara, M. (2024). Waveformer: Spectral–spatial wavelet transformer for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* 21, 1–5. doi: 10.1109/LGRS.2024.3353909

Ahmad, M., Khan, A. M., Mazzara, M., and Distefano, S. (2019). "Multi-layer extreme learning machine-based autoencoder for hyperspectral image classification." in *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)* - Volume 4: VISAPP. SciTePress, pp 75–82. doi: 10.5220/0007258000750082

Chang, H., Bi, H., Li, F., Xu, C., Chanussot, J., and Hong, D. (2024). Deep symmetric fusion transformer for multimodal remote sensing data classification. *IEEE Trans. Geosci. Remote Sens.* 62, 1–15. doi: 10.1109/TGRS.2024.3476975

Chen, H., Miao, F., Chen, Y., Xiong, Y., and Chen, T. (2021a). A hyperspectral image classification method using multifeature vectors and optimized kelm. *IEEE J. Selected Topics Appl. Earth Observations Remote Sens.* 14, 2781–2795. doi: 10.1109/JSTARS.4609443

Chen, J., Yang, C., Zhang, L., Yang, L., Bian, L., Luo, Z., et al. (2024). Tccu-net: Transformer and cnn collaborative unmixing network for hyperspectral image. *IEEE J. Selected Topics Appl. Earth Observations Remote Sens.* 17, 8073–8089. doi: 10.1109/JSTARS.2024.3352073

Chen, Y., Zheng, W., Li, W., and Huang, Y. (2021b). Large group activity security risk assessment and risk early warning based on random forest algorithm. *Pattern Recognition Lett.* 144, 1–5. doi: 10.1016/j.patrec.2021.01.008

Cui, B., Cui, J., Hao, S., Guo, N., and Lu, Y. (2020). Spectral-spatial hyperspectral image classification based on superpixel and multi-classifier fusion. *Int. J. Remote Sens.* 41, 6157–6182. doi: 10.1080/01431161.2020.1736730

Cui, B., Dong, X.-M., Zhan, Q., Peng, J., and Sun, W. (2021). Litedepthwisenet: A lightweight network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 60, 1–15. doi: 10.1109/TGRS.2021.3062372

Farmonov, N., Amankulova, K., Szatmári, J., Sharifi, A., Abbasi-Moghadam, D., Nejad, S. M. M., et al. (2023). Crop type classification by desis hyperspectral imagery and machine learning algorithms. *IEEE J. selected topics Appl. Earth observations Remote Sens.* 16, 1576–1588. doi: 10.1109/JSTARS.2023.3239756

Gao, H., Zhu, M., Wang, X., Li, C., and Xu, S. (2023). Lightweight spatial-spectral network based on 3d-2d multi-group feature extraction module for hyperspectral image classification. *Int. J. Remote Sens.* 44, 3607–3634. doi: 10.1080/01431161.2023.2224099

Ghaderizadeh, S., Abbasi-Moghadam, D., Sharifi, A., Zhao, N., and Tariq, A. (2021). Hyperspectral image classification using a hybrid 3d-2d convolutional neural networks. *IEEE J. Selected Topics Appl. Earth Observations Remote Sens.* 14, 7570–7588. doi: 10.1109/JSTARS.2021.3099118

Guo, T., Wang, R., Luo, F., Gong, X., Zhang, L., and Gao, X. (2023). Dual-view spectral and global spatial feature fusion network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 61, 1–13. doi: 10.1109/TGRS.2023.3277467

Han, D., Pan, X., Han, Y., Song, S., and Huang, G. (2023). Flatten transformer: Vision transformer using focused linear attention. In. *Proc. IEEE/CVF Int. Conf. Comput. vision.* 5961–5971. doi: 10.1109/ICCV51070.2023.00548

Hong, D., Han, Z., Yao, J., Gao, L., Zhang, B., Plaza, A., et al. (2021). Spectralformer: Rethinking hyperspectral image classification with transformers. *IEEE Trans. Geosci. Remote Sens.* 60, 1–15. doi: 10.1109/TGRS.2021.3130716

Hong, D., Yao, J., Li, C., Meng, D., Yokoya, N., and Chanussot, J. (2023). Decoupled-and-coupled networks: Self-supervised hyperspectral image super-resolution with subpixel fusion. *IEEE Trans. Geosci. Remote Sens.* 61, 1–12. doi: 10.1109/TGRS.2023.3324497

Huang, X., Dong, M., Li, J., and Guo, X. (2022). A 3-d-swin transformer-based hierarchical contrastive learning method for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 60, 1–15. doi: 10.1109/TGRS.2022.3202036

Jia, S., Bi, D., Liao, J., Jiang, S., Xu, M., and Zhang, S. (2023). Structure-adaptive convolutional neural network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 61, 1–16. doi: 10.1109/TGRS.2023.3326231

Jin, S., Zhang, F., Zheng, Y., Zhou, L., Zuo, X., Zhang, Z., et al. (2023). Csknn: Cost-sensitive k-nearest neighbor using hyperspectral imaging for identification of wheat varieties. *Comput. Electrical Eng.* 111, 108896. doi: 10.1016/j.compeleceng.2023.108896

Kumar, V., Singh, R. S., and Dua, Y. (2022). Morphologically dilated convolutional neural network for hyperspectral image classification. *Signal Processing: Image Communication* 101, 116549. doi: 10.1016/j.image.2021.116549

Kumbure, M. M., Luukka, P., and Collan, M. (2020). A new fuzzy k-nearest neighbor classifier based on the bonferroni mean. *Pattern Recognition Lett.* 140, 172–178. doi: 10.1016/j.patrec.2020.10.005

Lei, Y., and Tang, K. (2021). Learning rates for stochastic gradient descent with nonconvex objectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 4505–4511. doi: 10.1109/TPAMI.2021.3068154

Li, Z., Chen, G., Li, G., Zhou, L., Pan, X., Zhao, W., et al. (2024c). Dbanet: Dual-branch attention network for hyperspectral remote sensing image classification. *Comput. Electrical Eng.* 118, 109269. doi: 10.1016/j.compeleceng.2024.109269

Li, X., Ding, H., Yuan, H., Zhang, W., Pang, J., Cheng, G., et al. (2024b). Transformer-based visual segmentation: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 46, 12 doi: 10.1109/TPAMI.2024.3434373

Li, C., Zhang, B., Hong, D., Jia, X., Plaza, A., and Chanussot, J. (2024a). Learning disentangled priors for hyperspectral anomaly detection: A coupling model-driven and data-driven paradigm. *IEEE Trans. Neural Networks Learn. Syst.* doi: 10.1109/TNNLS.2024.3401589

Okwuashi, O., and Ndehedehe, C. E. (2020). Deep support vector machine for hyperspectral image classification. *Pattern Recognition* 103, 107298. doi: 10.1016/j.patcog.2020.107298

Ortac, G., and Ozcan, G. (2021). Comparative study of hyperspectral image classification by multidimensional convolutional neural network approaches to improve accuracy. *Expert Syst. Appl.* 182, 115280. doi: 10.1016/j.eswa.2021.115280

Paoletti, M. E., Moreno-Álvarez, S., Xue, Y., Haut, J. M., and Plaza, A. (2023a). Aatt-cnn: Automatical attention-based convolutional neural networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 61, 1–18. doi: 10.1109/TGRS.2023.3272639

Paoletti, M. E., Moreno-Álvarez, S., Xue, Y., Haut, J. M., and Plaza, A. (2023b). Aatt-cnn: Automatical attention-based convolutional neural networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 61, 1–18. doi: 10.1109/TGRS.2023.3272639

Peng, Y., Zhang, Y., Tu, B., Li, Q., and Li, W. (2022). Spatial–spectral transformer with cross-attention for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 60, 1–15. doi: 10.1109/TGRS.2022.3203476

Pham, Q. T., and Liou, N.-S. (2022). The development of on-line surface defect detection system for jujubes based on hyperspectral images. *Comput. Electron. Agric.* 194, 106743. doi: 10.1016/j.compag.2022.106743

Qi, W., Huang, C., Wang, Y., Zhang, X., Sun, W., and Zhang, L. (2023a). Global-local three-dimensional convolutional transformer network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 61, 1–20. doi: 10.1109/TGRS.2023.3272885

Qi, W., Huang, C., Wang, Y., Zhang, X., Sun, W., and Zhang, L. (2023b). Global-local three-dimensional convolutional transformer network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 61, 1–20. doi: 10.1109/TGRS.2023.3272885

Qiu, Z., Xu, J., Peng, J., and Sun, W. (2023). Cross-channel dynamic spatial-spectral fusion transformer for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 61, 1–12. doi: 10.1109/TGRS.2023.3324730

Roy, S. K., Krishna, G., Dubey, S. R., and Chaudhuri, B. B. (2019). Hybridsn: Exploring 3-d–2-d cnn feature hierarchy for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* 17, 277–281. doi: 10.1109/LGRS.2019.2918719

Roy, S. K., Manna, S., Song, T., and Bruzzone, L. (2020). Attention-based adaptive spectral–spatial kernel resnet for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 59, 7831–7843. doi: 10.1109/TGRS.2020.3043267

Roy, S. K., Mondal, R., Paoletti, M. E., Haut, J. M., and Plaza, A. (2021). Morphological convolutional neural networks for hyperspectral image classification. *IEEE J. Selected Topics Appl. Earth Observations Remote Sens.* 14, 8689–8702. doi: 10.1109/JSTARS.2021.3088228

Sim, J., Dixit, Y., Mcgoverin, C., Oey, I., Frew, R., Reis, M. M., et al. (2024). Machine learning-driven hyperspectral imaging for non-destructive origin verification of green coffee beans across continents, countries, and regions. *Food Control* 156, 110159. doi: 10.1016/j.foodcont.2023.110159

Su, Y., Gao, L., Jiang, M., Plaza, A., Sun, X., and Zhang, B. (2022). Nsckl: Normalized spectral clustering with kernel-based learning for semisupervised hyperspectral image classification. *IEEE Trans. Cybernetics*. 53 (10), 6649–6662. doi: 10.1109/TCYB.2022.3219855

Sun, L., Fang, Y., Chen, Y., Huang, W., Wu, Z., and Jeon, B. (2022a). Multi-structure kelm with attention fusion strategy for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 60, 1–17. doi: 10.1109/TGRS.2022.3208165

Sun, G., Pan, Z., Zhang, A., Jia, X., Ren, J., Fu, H., et al. (2023). Large kernel spectral and spatial attention networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 61, 1–15. doi: 10.1109/TGRS.2023.3292065

Sun, L., Zhang, H., Zheng, Y., Wu, Z., Ye, Z., and Zhao, H. (2024). Massformer: Memory-augmented spectral-spatial transformer for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 62, 1–15. doi: 10.1109/TGRS.2024.3392264

Sun, L., Zhao, G., Zheng, Y., and Wu, Z. (2022b). Spectral–spatial feature tokenization transformer for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 60, 1–14. doi: 10.1109/TGRS.2022.3144158

Tang, P., Zhang, M., Liu, Z., and Song, R. (2023). Double attention transformer for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* 20, 1–5. doi: 10.1109/LGRS.2023.3248582

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, 5998–6008. doi: 10.48550/ARXIV.1706.03762

Wang, J., Song, X., Sun, L., Huang, W., and Wang, J. (2020). A novel cubic convolutional neural network for hyperspectral image classification. *IEEE J. Selected Topics Appl. Earth Observations Remote Sens.* 13, 4133–4148. doi: 10.1109/JSTARS.4609443

Wang, X., Tan, K., Du, P., Han, B., and Ding, J. (2023b). A capsule-vectored neural network for hyperspectral image classification. *Knowledge-Based Syst.* 268, 110482. doi: 10.1016/j.knosys.2023.110482

Wang, Y., Yu, X., Wen, X., Li, X., Dong, H., and Zang, S. (2024a). Learning a 3d-cnn and convolution transformers for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* 21, 1–5. doi: 10.1109/LGRS.2024.3365615

Wang, D., Zhang, J., Du, B., Zhang, L., and Tao, D. (2023a). Dcn-t: Dual context network with transformer for hyperspectral image classification. *IEEE Trans. Image Process.* 32, 2536–2551. doi: 10.1109/TIP.2023.3270104

Wang, Z., Zhao, S., Zhao, G., and Song, X. (2024b). Dual-branch domain adaptation few-shot learning for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 62, 1–16. doi: 10.1109/TGRS.2024.3356199

Wu, X., Hong, D., and Chanussot, J. (2021). Convolutional neural networks for multimodal remote sensing data classification. *IEEE Trans. Geosci. Remote Sens.* 60, 1–10. doi: 10.1109/TGRS.2020.3040277

Wu, X., Hong, D., and Chanussot, J. (2022). Uiu-net: U-net in u-net for infrared small object detection. *IEEE Trans. Image Process.* 32, 364–376. doi: 10.1109/TIP.2022.3228497

Xu, R., Dong, X.-M., Li, W., Peng, J., Sun, W., and Xu, Y. (2024). Dbctnet: Double branch convolutiontransformer network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 62, 1–15. doi: 10.1109/TGRS.2024.3368141

Yang, X., Cao, W., Lu, Y., and Zhou, Y. (2022). Hyperspectral image transformer classification networks. *IEEE Trans. Geosci. Remote Sens.* 60, 1–15. doi: 10.1109/TGRS.2022.3171551

Yang, K., Sun, H., Zou, C., and Lu, X. (2021). Cross-attention spectral–spatial network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 60, 1–14. doi: 10.1109/TGRS.2021.3133582

Yu, C., Han, R., Song, M., Liu, C., and Chang, C.-I. (2021). Feedback attention-based dense cnn for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 60, 1–16. doi: 10.1109/TGRS.2020.3040273

Yu, H., Xu, Z., Zheng, K., Hong, D., Yang, H., and Song, M. (2022). Mstnet: A multilevel spectral–spatial transformer network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 60, 1–13. doi: 10.1109/TGRS.2022.3186400

Zhang, B., Chen, Y., Rong, Y., Xiong, S., and Lu, X. (2023a). Matnet: A combining multi-attention and transformer network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 61, 1–15. doi: 10.1109/TGRS.2023.3254523

Zhang, W., Chen, G., Zhuang, P., Zhao, W., and Zhou, L. (2024a). Catnet: Cascaded attention transformer network for marine species image classification. *Expert Syst. Appl.* 256, 124932. doi: 10.1016/j.eswa.2024.124932

Zhang, W., Li, Z., Li, G., Zhou, L., Zhao, W., and Pan, X. (2024b). Aganet: Attention-guided generative adversarial network for corn hyperspectral images augmentation. *IEEE Trans. Consumer Electron.* doi: 10.1109/TCE.2024.3470846

Zhang, W., Li, Z., Li, G., Zhuang, P., Hou, G., Zhang, Q., et al. (2023b). Gacnet: Generate adversarialdriven cross-aware network for hyperspectral wheat variety identification. *IEEE Trans. Geosci. Remote Sens.* 62, 1–14. doi: 10.1109/TGRS.2023.3347745

Zhang, W., Li, Z., Sun, H.-H., Zhang, Q., Zhuang, P., and Li, C. (2022b). Sstnet: Spatial, spectral, and texture aware attention network using hyperspectral image for corn variety identification. *IEEE Geosci. Remote Sens. Lett.* 19, 1–5. doi: 10.1109/LGRS.2022.3225215

Zhang, H., Meng, L., Wei, X., Tang, X., Tang, X., Wang, X., et al. (2019). 1d-convolutional capsule network for hyperspectral image classification. *arXiv preprint arXiv:1903.09834*. doi: 10.48550/arXiv.1903.09834

Zhang, J., Meng, Z., Zhao, F., Liu, H., and Chang, Z. (2022a). Convolution transformer mixer for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* 19, 1–5. doi: 10.1109/LGRS.2022.3208935

Zhang, X., Su, Y., Gao, L., Bruzzone, L., Gu, X., and Tian, Q. (2023d). A lightweight transformer network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 61, 1–17. doi: 10.1109/TGRS.2023.3297858

Zhang, L., Sun, H., Rao, Z., and Ji, H. (2020). Hyperspectral imaging technology combined with deep forest model to identify frost-damaged rice seeds. *Spectrochimica Acta Part A: Mol. biomolecular Spectrosc.* 229, 117973. doi: 10.1016/j.saa.2019.117973

Zhang, W., Sun, X., Zhou, L., Xie, X., Zhao, W., Liang, Z., et al. (2023c). Dual-branch collaborative learning network for crop disease identification. *Front. Plant Sci.* 14, 1117478. doi: 10.3389/fpls.2023.1117478

Zhang, W., Zhao, W., Li, J., Zhuang, P., Sun, H., Xu, Y., et al. (2024c). Cvanet: Cascaded visual attention network for single image super-resolution. *Neural Networks* 170, 622–634. doi: 10.1016/j.neunet.2023.11.049

Zhao, C., Qin, B., Feng, S., Zhu, W., Sun, W., Li, W., et al. (2023). Hyperspectral image classification with multi-attention transformer and adaptive superpixel segmentation-based active learning. *IEEE Trans. Image Process.* 32, 3606–3621. doi: 10.1109/TIP.2023.3287738

Zheng, X., Sun, H., Lu, X., and Xie, W. (2022). Rotation-invariant attention network for hyperspectral image classification. *IEEE Trans. Image Process.* 31, 4251–4265. doi: 10.1109/TIP.2022.3177322

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 06 February 2025
DOI 10.3389/fpls.2025.1489655

OPEN ACCESS

EDITED BY
Mohsen Yoosefzadeh Najafabadi,
University of Guelph, Canada

REVIEWED BY
Xiangfang Li,
Prairie View A&M University, United States
Mohammad Shameem Al Mamun,
Bangladesh Tea Research Institute,
Bangladesh

*CORRESPONDENCE
Lanting Li
1637020418@qq.com

RECEIVED 01 September 2024
ACCEPTED 07 January 2025
PUBLISHED 06 February 2025

CITATION
Li L and Zhao Y (2025) Tea disease identification based on ECA attention mechanism ResNet50 network.
Front. Plant Sci. 16:1489655.
doi: 10.3389/fpls.2025.1489655

Tea disease identification based on ECA attention mechanism ResNet50 network

Lanting Li* and Yingding Zhao

School of Software, Jiangxi Agricultural University, Nanchang, China

Addressing the challenge of identifying tea plant diseases against the complex background of tea gardens, this study proposes the ECA-ResNet50 model. By optimizing the ResNet50 architecture, adopting a multi-layer small convolution kernel strategy to enhance feature extraction capabilities, and introducing the ECA attention mechanism to focus on key features, the model achieves a 93.06% accuracy rate in tea disease identification, representing a 3.18% improvement over the original model, demonstrating industry-leading performance advantages. This model not only accurately identifies tea diseases in gardens but also possesses excellent generalization capabilities, performing outstandingly on datasets of other plant categories. These results indicate that ECA-ResNet50 can effectively mitigate the interference of complex backgrounds and precisely recognize tea disease targets.

KEYWORDS

tea plant diseases, ECA attention mechanism, ResNet50, deep learning, leave

1 Introduction

The tea industry in China has undergone years of development and continues to grow steadily, occupying an important position in the domestic market and enjoying a strong reputation internationally. However, throughout the cultivation process, tea plants inevitably face various diseases and pests, which not only severely affect tea yields but also pose a serious threat to the quality of the tea. To effectively address this challenge, it is essential to actively introduce and apply emerging technologies such as artificial intelligence, enabling precise and rapid detection and effective control of tea diseases, thereby ensuring the sustainable and healthy development of the tea industry.

Computer vision, as an important branch of artificial intelligence technology, aims to enable machines to possess visual perception capabilities similar to those of humans (Yu et al., 2023). Currently, many countries are actively exploring the practical applications of computer vision in the agricultural sector, achieving significant research results. Among these, employing deep learning technology for crop disease recognition, followed by the application of effective control strategies, has emerged as a pivotal trend shaping agricultural progress. The application of this technology not only allows computers to

provide rapid and accurate diagnostic results but also significantly enhances the quality and overall yield of crops while reducing additional labor costs and time consumption, thereby providing strong support for the sustainable development of agriculture (Dhanya et al., 2022).

In 2016, Li (2017) designed a tobacco disease diagnosis system based on a six-layer convolutional neural network model. This system utilizes deep learning techniques to identify tobacco diseases and provides convenient diagnosis and prevention services for growers through web design. The research further delved into the impacts of varying iteration numbers and resolutions on the training efficiency and classification capabilities of the network model. In 2017, Sun et al. (2017) and his team introduced a convolutional neural network framework incorporating batch normalization and global pooling methodologies. After adjusting the network structure and parameters, this model greatly improved the accuracy, efficiency, and stability of plant disease identification. Optimizations resulted in the best model significantly outpacing traditional convolutional neural networks in convergence speed, achieving an accuracy rate exceeding 90% after just three training iterations. Furthermore, this proposed model necessitates minimal computational demands, featuring a parameter memory of merely 2.6 MB, and attained a remarkable average testing recognition accuracy of 99.56%, with comprehensive performance for recall and precision reaching 99.41%. These improvements enable the model to deliver efficient and accurate performance in the field of plant disease identification. In 2018, Lu et al. (2018) and colleagues proposed a deep learning-based recognition method for rice leaf disease images. They constructed a rice disease image database and employed PCA (Principal Component Analysis) for dimensionality reduction. Utilizing the Caffe deep learning framework, they crafted a profound network architecture encompassing four convolutional tiers, three pooling stages, and a solitary fully connected layer. Training and simulation with 2,000 rice disease images, combined with ten-fold cross-validation testing, verified that the designed deep learning structure and learning algorithm achieved an average recognition rate of 96.9% for common diseases such as rice blast and sheath blight in northern cold region rice. The experimental results thoroughly demonstrated the effectiveness of this method in identifying major rice leaf diseases, providing strong technical support for accurate recognition and prevention of rice diseases. In 2019, Wu (2019) proposed a tomato leaf disease recognition technology based on a deep residual network. This technology automatically adjusts the key hyperparameters in the network using a Bayesian optimization algorithm, streamlining the training procedure for the deep learning network. By incorporating residual units into the traditional neural network structure, it mitigated potential concerns related to gradient vanishing and explosion phenomena within deep networks significantly enhancing the performance of the network model and allowing for precise extraction of high-dimensional features from tomato leaf images. These features were then used for accurate disease identification. Experiments showed that the deep residual network model in this study achieved recognition accuracy exceeding 95% for common tomato leaf diseases such as powdery mildew, early blight, late blight, and leaf mold on public datasets. This study offers a noteworthy reference for swiftly and precisely identifying tomato leaf diseases. In 2020, Ji et al. (2020) and colleagues adopted a convolutional neural network based on an improved residual network, using publicly available plant image datasets for training. Comparative experiments with the Xception and VGG-16 network models showed that the improved neural network model achieved an accuracy rate of 98.6%, significantly surpassing Xception's 93% and VGG-16's 95%, demonstrating its efficiency and accuracy. In 2021, Wang et al. (2021) and colleagues proposed an improved CenterNet-SPP model for potato leaf diseases. This model first precisely locates the central points of the targets using a feature extraction network, and then accurately obtains key image information such as center point offset and target size through center point regression techniques. The experiments demonstrated that the model attained a mean average precision (mAP) score of 90.03% on the validation dataset. In 2022, Sun and Lin (2022) and colleagues introduced a novel approach for detecting apple leaf diseases, leveraging ensemble learning techniques. This method integrates the YOLOv5 and EfficientDet models, achieving model integration through a non-maximum suppression algorithm. Testing showed that the new method effectively improved the detection performance of three common apple leaf diseases without sacrificing detection speed, with average precision rising to 73.4%. Compared to the individual use of YOLOv5 and EfficientDet, the new method improved accuracy by 3.0% and 4.8%, respectively. In 2023, Li et al. (2023) and colleagues constructed an alfalfa disease recognition model using an improved AlexNet deep learning convolutional neural network, trained on a dataset of 13 common alfalfa diseases. After comparing different image input resolutions, they found that the optimal model achieved the highest recognition accuracy with an input size of 512 pixels × 512 pixels, reaching an overall recognition accuracy of 72%. After further excluding low-accuracy samples, the recognition accuracy for five key alfalfa diseases significantly increased to 92%. In 2024, Qiu et al. (2024) and colleagues developed an algorithm called CBAM-YOLOv5l based on an improved YOLOv5. Through experiments, they confirmed that the method enhanced detection accuracy without compromising on the swiftness of the detection process. The algorithm achieved an overall average precision of 96.52% on the validation set, with an average detection time of 27.52 ms, demonstrating significant advantages in detection accuracy compared to other object detection algorithms like YOLOv4, YOLOv4-Tiny, and Faster R-CNN.

Currently, investigations into recognizing plant leaf diseases and pests with convolutional neural networks predominantly depend on conventional frameworks devoid of an attention weighting mechanism. This can lead to a misalignment of the model's focus, subsequently affecting recognition accuracy. Moreover, the aforementioned studies have not applied the improved models to the recognition of diseases and pests in other crop leaves, making it impossible to comprehensively validate their generalization capabilities. To tackle these challenges, this research introduces the ECA-ResNet50 model, which integrates the ECA attention mechanism with the ResNet50 network framework. This model focuses on various tea leaf diseases, such as algal leaf disease, anthracnose, and bird's eye spot disease, as well as healthy tea leaves. Through comparative experiments with traditional convolutional neural networks, the effectiveness of ResNet-ECA in tea disease recognition was validated. Additionally, to further

assess the generalization performance of the improved model, it was applied to train and validate datasets of disease and pest leaves from other crops, including corn, apples, and potatoes.

2 Research and implementation of algorithm

2.1 Dataset construction

2.1.1 Data acquisition for dataset

According to statistical data analysis of the system, China's tea plants suffer from a wide variety of diseases, totaling approximately over 140 types, which are widely distributed across various parts of the tea plants, including leaves, stems, roots, and flowers (Chen, 2022). Given the limitations of experimental conditions, this study collected a total of 885 images of tea diseases through search engines (https://www.kaggle.com/datasets/shashwatwork/identifying-disease-in-tea-leafs). After meticulous identification and classification by authoritative experts, these images were categorized into seven distinct types of leaf diseases, as well as healthy leaves. The seven disease types are algae leaf spot, anthracnose, bird's eye spot, cloud blotch, gray spot, red leaf spot, and white spot disease. Some images of tea disease leaves are shown in Figure 1.

2.1.2 Dataset processing

During the training phase of a Convolutional Neural Network (CNN) model, ensuring a large-scale and diverse dataset plays a decisive role in enhancing the model's performance. However, acquiring a sufficient number of images that cover various types of tea plant disease under current conditions is a formidable challenge. To address this issue, this research employs data augmentation strategies to efficiently augment the training dataset thereby improving the model's generalization capability and recognition accuracy for tea plant disease images. Firstly, the original dataset is expanded through a series of data augmentation techniques, including flipping, rotation, cropping, color transformation, and blurring, with each method expanding the dataset to 1000 images. Some examples of the augmented images are shown in Figure 2. Subsequently, the expanded dataset is divided into a training set and a test set at an 8:2 ratio. During the data preprocessing stage, to ensure data consistency and compatibility with the model's input requirements, all images are resized to a uniform dimension of 224×224 pixels. Furthermore, through padding and random shuffling, we aim to fully utilize the data information and enhance the model's training effectiveness.

2.2 ECA mechanism

The ECA (Wang et al., 2020) module is an optimized version of the SE (Hu et al., 2019) attention module that significantly enhances

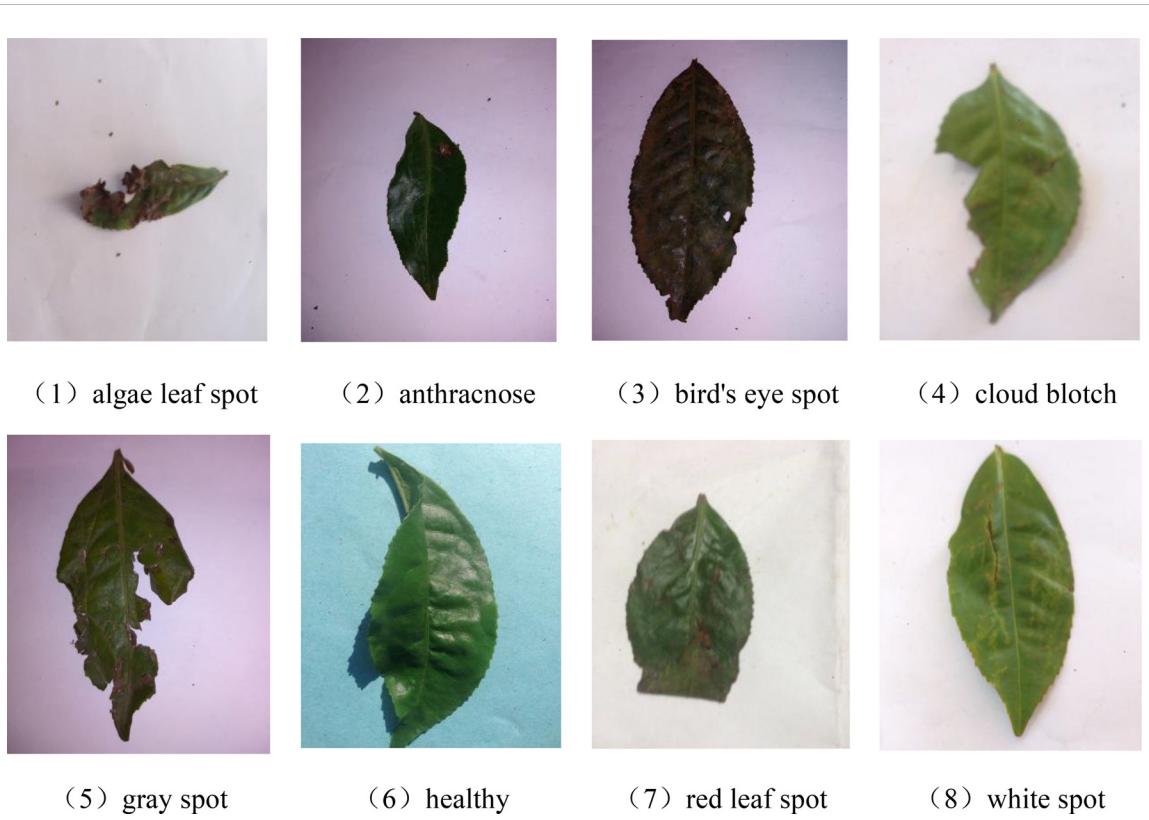


FIGURE 1
Images of tea diseases.

FIGURE 2
Image enhancement examples.

performance despite having fewer parameters. When performing global average pooling, it ingeniously avoids compressing the channels of the input feature map, aiming to mitigate the adverse effects of learning inter-channel dependencies. Within the ECA module, the extent of local cross-channel interaction is defined as k, meaning each channel and its adjacent k channels are considered. By utilizing a one-dimensional fast convolution tailored to the k value, the module efficiently accomplishes local cross-channel interaction, capturing the relationships among channels. Finally, the weights, post-processed via a Sigmoid function, are scaled with the corresponding entries in the input feature map to yield the output. Its structural diagram is shown in Figure 3. The distinctive architecture of the ECA module enables the model to prioritize the feature information pertaining to smaller objects, ensuring both efficiency and computational effectiveness. Since the k value is proportional to the number of channels, to avoid cross-validation, the k value can be obtained through Equation 1:

$$K = \left| \frac{1bC + b}{\gamma} \right|_{odd} \quad (1)$$

In the equation, C denotes the channel count in the input feature map, while b and γ are conventionally initialized as 1 and 2, respectively, respectively, to adjust the ratio between the dimensions of the convolutional kernel and the value of C. The notation odd indicates that K should be the odd number closest to the function's value.

2.3 ResNet50

ResNet50 (He et al., 2016) is a deep convolutional neural network-based algorithm designed for image classification tasks, proposed by Kaiming He and his colleagues at Microsoft Research in 2015. As an important member of the ResNet family, ResNet50 addresses the issue of gradient vanishing during the training of deep networks by introducing residual connections, effectively enhancing the model's performance.

The ResNet50 architecture comprises numerous residual blocks, which include additional layers such as pooling layers and fully connected layers. The overall structure of the network is very

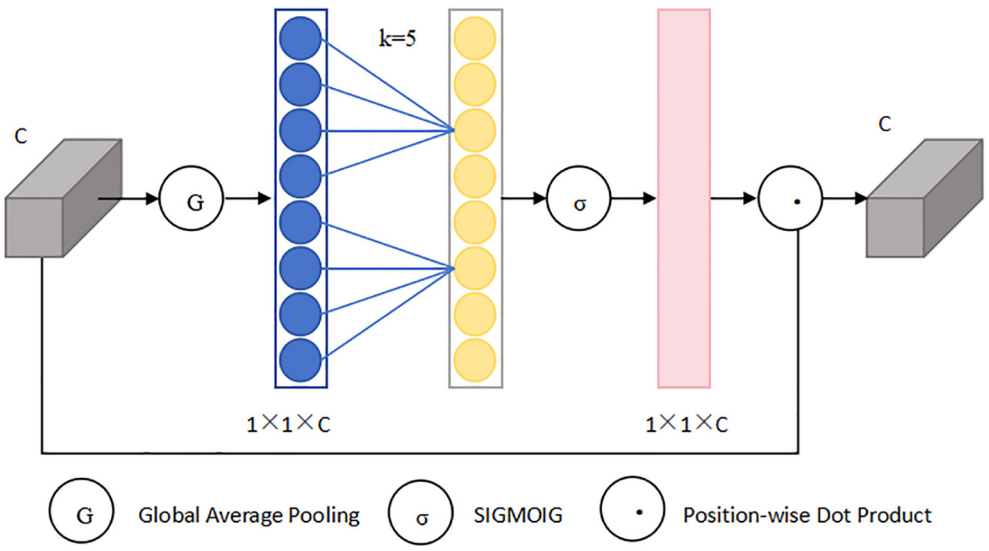


FIGURE 3
ECA architecture diagram.

deep, employing 50 convolutional layers, hence the name ResNet50. These convolutional layers extract features from images at different sizes and depths, enabling the model to capture features at various levels. The configuration of a residual block is depicted in Figure 4.

Each residual block is linked to each other by residual connections. This direct connection mitigates the issue of vanishing gradients by enabling the seamless flow of information across network layers. In ResNet50, each residual block consists of two convolutional layers, called the main path and the hop connection, respectively. By adding the input to the output of the main path, the residual learning of the information is realized. The formula for each residual element is as follows:

$$x_{j+1} = x_j + F(x_j, W_j) \tag{2}$$

where x_j、 x_{j+1} denotes both the input and output information of the layer network, respectively, and represents the learnable parameters within that layer. Perform a recursive operation on Equation 2 to obtain the relational expression of any deep J and shallow J:

$$x_J = x_j + \sum_{i=j}^{J-1} F(x_i, W_i) \tag{3}$$

According to the chain derivative used in the backpropagation algorithm, the gradient of backpropagation can be expressed as:

$$\frac{\partial \varepsilon}{\partial x_j} = \frac{\partial \varepsilon}{\partial x_J}\frac{\partial x_J}{\partial x_j} = \frac{\partial \varepsilon}{\partial x_J}\left[1 + \frac{\partial}{\partial x_j}\sum_{i=j} F(x_i, w_i)\right] \tag{4}$$

Because all $\frac{\partial}{\partial x_j}\sum_{i=j} F(x_i, W_i)$ in Equation 4 may be equal to −1, this unit effectively mitigates the issue of information loss during the learning phase.

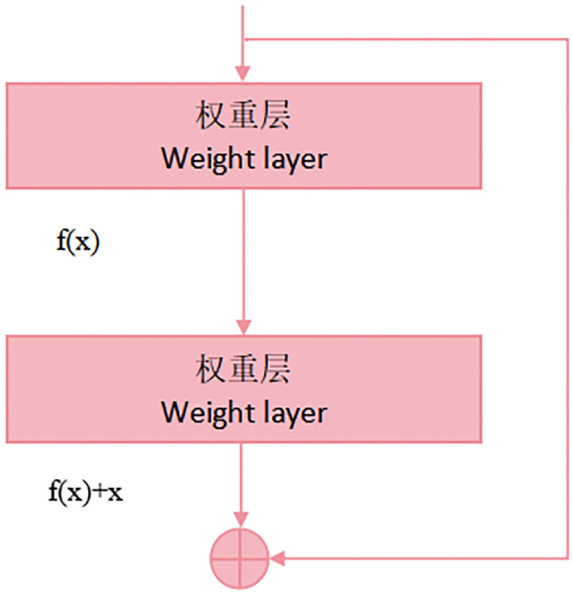


FIGURE 4
Residual block structure diagram.

2.4 Network architecture based on ECA attention mechanism and ResNet50

The ECA-ResNet50 model is optimized and improved on top of the ResNet-50 infrastructure. Firstly, the 7×7 convolution kernel of the first layer of ResNet-50 was replaced by three 3×3 convolution kernels. In the traditional ResNet50, the 7×7 convolution kernel is designed to capture a wider range of spatial context information in the input image, however, in the tea disease identification scenario, the disease characteristics are often complex and subtle, and the affected area is comparatively minute. In view of this, the strategy of using multi-layer small convolutional kernel not only refines the granularity of feature extraction and improves the accuracy of disease identification, but also enhances the learning ability and complexity of the model by reducing the total number of parameters and increasing the network depth, and significantly optimizes the performance. Moreover, to enhance the model's sensitivity and recognition efficiency towards tea disease characteristics even further, ECA-ResNet50 integrates the ECA attention mechanism into the first residual module of ResNet-50. Although ResNet-50 itself can effectively alleviate the gradient problem in deep network training, relying solely on numerical transfer may not be enough to accurately capture the key features when dealing with tea diseases with similar characteristics, which will affect the recognition accuracy and generalization ability. By introducing the ECA attention mechanism, the model can focus on more discriminative feature information in the image, which effectively enhances the learning and recognition ability of tea disease characteristics, which is a key measure to improve the overall performance of the model, Figure 5 is the structure diagram of ECA-ResNet50.

2.5 Experimental parameters and evaluation metrics

Precision, recall, accuracy, and F1-score were employed to assess the network model's performance in identifying tea diseases. The formulas for calculating these evaluation metrics are outlined below:

$$precision = \frac{TP}{TP + FP} \tag{5}$$

$$recall = \frac{TP}{TP + FN} \tag{6}$$

$$accuray = \frac{TP + TN}{TP + FN + FP + TN} \tag{7}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{8}$$

Here, TP denotes the count of samples accurately labeled as positive by the model, TN represents the count of samples correctly identified as negative. FP signifies the number of negative samples mistakenly predicted as positive, while FN represents the number of

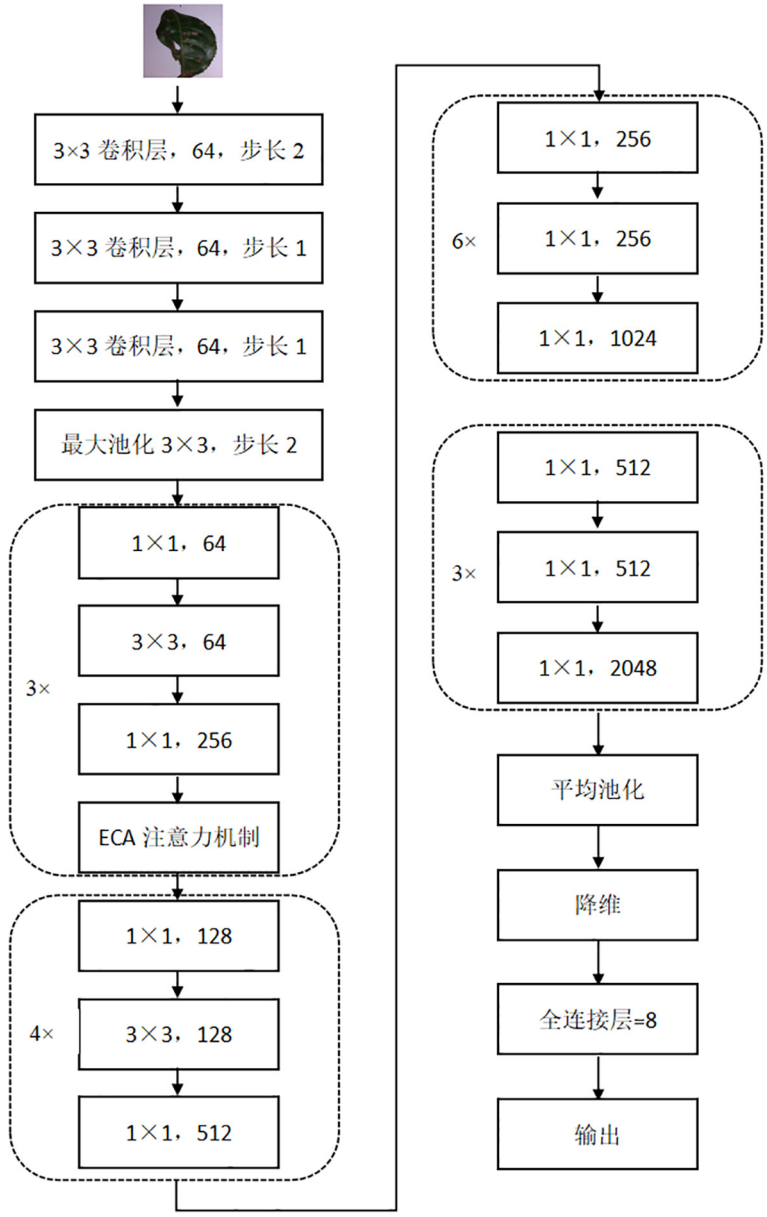


FIGURE 5
Structure diagram of ECA-ResNet50.

positive samples incorrectly labeled as negative. Accuracy gauges the fraction of samples correctly predicted by the model among all test samples, calculated as the total number of correctly predicted samples divided by the total test samples. Precision focuses on the ratio of samples predicted as positive by the model that are actually positive, computed as the number of correctly predicted positive samples divided by the total number of samples predicted as positive. Recall, also known as the true positive rate, assesses the proportion of actual positive samples accurately identified by the model, calculated as the number of correctly predicted positive samples divided by the total number of positive samples. F1 score serves as a comprehensive metric, balancing the significance of precision and recall by computing their harmonic mean. A higher F1 score signifies superior performance in both precision and recall, making it a frequently utilized evaluation metric for classification models.

3 Results and discussion

3.1 Experimental environment

This investigation is conducted utilizing the TensorFlow platform of the Python programming language, encompassing two distinct phases: model training and testing. In terms of hardware environment, IT uses an intel(R) Xeon(R) Silver 4112 processor with a frequency of 2.6 GHz. 16 GB of memory space;

NVIDIA Quadro RTX5000 graphics card. In terms of software environment configuration, CUDAToolkit 10.0, CUDNN 10.1 and TensorFlow 2.2 are selected as the deep learning framework, and the operating system is Windows 10.

3.2 Activation function comparison experiment

In neural network models, activation functions play a very important role, which greatly enhances the network's ability to process complex data and function mapping by giving the network nonlinear ability, adjusting the output range, and promoting sparse expression (Bahdanau et al., 2014). To enhance the efficiency and effectiveness of the model, optimization efforts are undertaken, three activation functions, ReLU (Xu et al., 2016), LeakyReLU (Technicolor T. et al., 2019) and ELU, were selected for training and comparison, so as to select the activation function strategy that is most consistent with the model. The experimental results are shown in Table 1, as evident from the tabular data, the ReLU activation function exhibits favorable performance in terms of accuracy, recall, and F1-score, and its accuracy is 1.68% and 7.5% higher than that of LeakyReLU and ELU, respectively. The above data show the superiority and applicability of the ReLU activation function in the ECA-ResNet50 model adopted in this study, and it can give full play to the potential of the model and achieve better performance than the other two activation functions.

3.3 Comparative experiments on attention mechanisms

Within the framework of neural network designs, the attention mechanism module plays a pivotal role, as an additional component of the neural network, can selectively focus on a specific part of the input, or effectively filter the information by assigning differentiated weights to different elements of the input. In recent times, due to its substantial contribution to enhancing model performance, this mechanism has garnered widespread adoption and implementation across diverse fields. In this study, three mainstream attention mechanisms, ECA, SE, and CABM (Fe et al., 2017), were selected to test and evaluate their respective effects in enhancing model performance. Table 2 shows the performance comparison results achieved after introducing these three attention mechanisms into the model. Based on an examination of the experimental data, under the same experimental environment settings and conditions, the ECA attention mechanism has the best effect among the three attention mechanisms, showing the best performance, with an accuracy of 93.06%, exhibiting a 3.5% increase in comparison to the SE attention mechanism within the model and 1.81% more accurate than the CBAM attention mechanism. These results show that the ECA attention mechanism can more effectively enhance the recognition ability and robustness of the model in this experimental model.

3.4 Ablation experiments

To validate the efficacy of the ECA attention mechanism module alongside three 3×3 convolutional kernel modules, ablation experiments were performed on the tea dataset, utilizing ResNet50 as the foundation network. The qualitative comparative outcomes are presented in Table 3. As can be seen from the data analysis in Table 3, the recognition accuracy of the model is significantly improved by 1.82% after the ECA attention mechanism is integrated into the ResNet50 model. The notable enhancement stems from the integration of the attention mechanism, empowering the model to precisely concentrate on the pivotal distinguishing characteristics within the image, thereby enhancing the recognition and learning efficiency of tea disease features, and improving the overall performance of the model. In addition, the replacement of three 3×3 convolution kernels with one 7×7 convolution kernel also brings a slight improvement in recognition accuracy. This improvement is due to the refinement of feature extraction brought about by the multi-layer small convolutional kernel design, which not only reduces the total number of model parameters, additionally, it enhances the intricacy and learning capacity of the model by augmenting the depth of the network, thereby promoting the improvement of

TABLE 1 Activation function comparison experiment.

Activate the function	The number of iterations	Precision%	Recall%	F1%	Accuracy%
ReLU	200	93.09	93.06	93.07	93.06
LeakyReLu	200	91.43	91.38	91.40	91.38
ELu	200	87.71	85.56	86.62	85.56

TABLE 2 Comparative experiments on attention mechanisms.

Attention mechanisms	Batch size	The number of iterations	Activate the function	Accuracy%	#P
Join ECA	64	200	ReLu	93.06	23,569,869
Join SE	64	200	ReLu	90.56	23,585,736
Join CBAM	64	200	ReLu	91.25	23,585,834

TABLE 3 Ablation experiments.

Join ECA	Replace the 3×3 convolution kernel	Accuracy%	Precision%	Recall%	F1%
×	×	89.88	89.90	89.88	88.89
√	×	92.5	92.61	92.5	92.55
×	√	90.68	90.69	90.69	90.69
√	√	93.06	93.09	93.06	93.07

× is not added, √ is added.

the accuracy of tea disease identification. Figures 6, 7 are ECA-ResNet50 and ResNet50 confusion matrices, respectively.

3.5 Comparative experiments with other datasets

To ascertain the versatility and generalizability of the model introduced in this research, extending beyond tea disease identification, we sourced disease image exemplars of apple and corn crops from the publicly accessible PlantVillage dataset (github.com/spMohanty/PlantVillage-Dataset), and each crop contained three different disease types, including 3000 apple disease images and 3192 maize disease images. The image data is divided into 80% training set and 20% test set. The ECA-ResNet50 model was then trained and tested with the original ResNet50 model, and the outcomes, presented in Table 4, indicate that the ECA-ResNet50 model demonstrates exceptional performance in the recognition of apple and maize diseases, and its accuracy is significantly improved compared with the unimproved ResNet50 model, which is 9.43% higher in apple disease identification and 4.17% higher in maize disease identification. This experimental endeavor conclusively establishes that the model presented in this research transcends the confines of solely tea disease identification, but also has a wide range of applicability, and can be effectively applied to the disease detection of other crops.

3.6 Other models than experiments

To assess the performance of the model introduced in this research in an unbiased manner, eight classical network models, including AlexNet (Huang et al., 2024), MobileNet (Sandler et al., 2018), and VGG16 (Zhao et al., 2024), were used to test and compare on the tea disease dataset, and the specific comparison results are shown in Table 5. The tabular data underscores the notable superiority of the ECA-ResNet50 model in the realm of tea disease identification, and its accuracy exceeds that of AlexNet

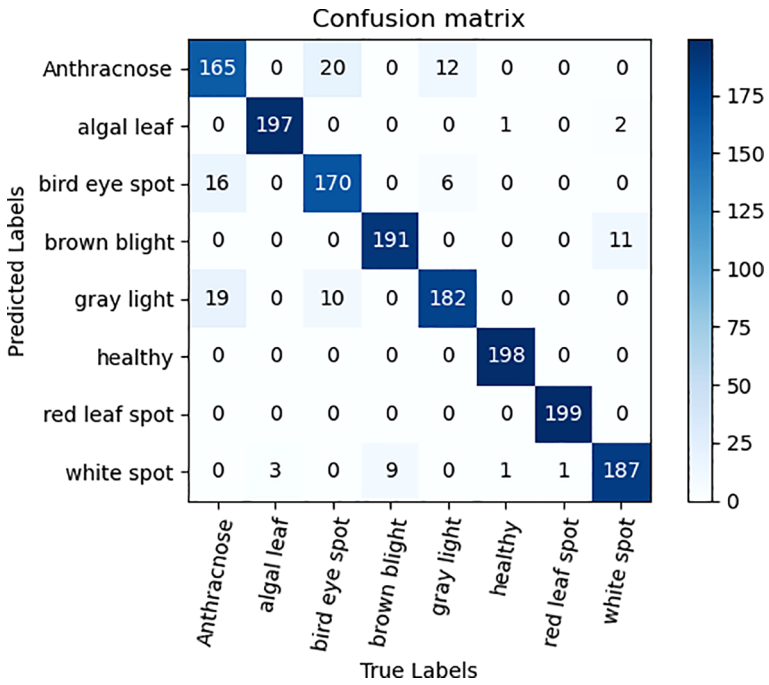


FIGURE 6
Confusion matrix diagram of ECA-ResNet50.

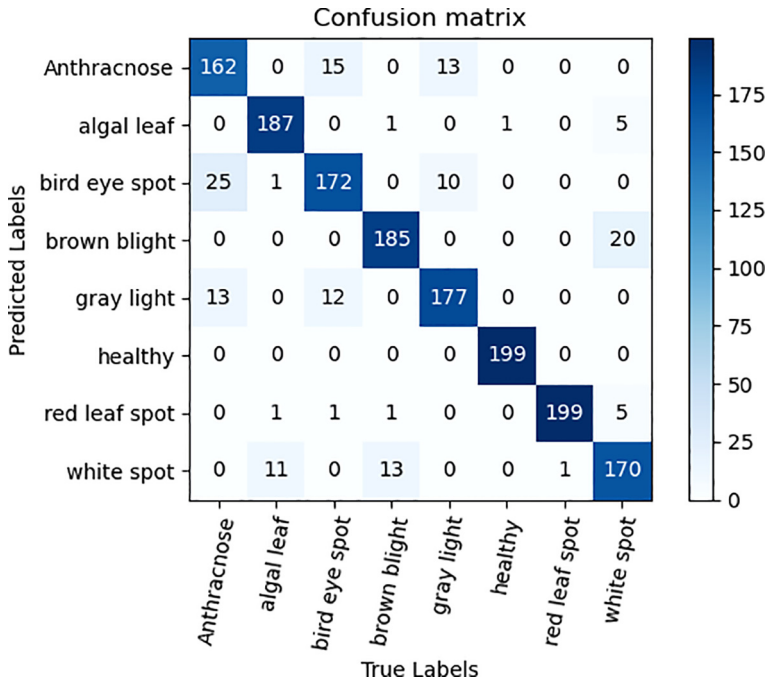


FIGURE 7
Confusion matrix diagram of ECA-ResNet50.

(2.68%), MobileNet (7.18%), VGG16 (1.81%), ResNet34 (2.43%), ResNet50 (3.18%) and ResNet101 (2.62%). Only slightly lower than InceptionResnetv2 model (0.57% lower) and lower than Transformer (1.43% lower). Nonetheless, it is pertinent to mention that the InceptionResnetv2 model and Transformer model exhibits a considerably higher level of complexity in comparison to ECA-ResNet50. In summary, the ECA-ResNet50 model not only performs well in tea disease identification, but also has high robustness, which is a relatively lightweight model with superior performance.

4 Conclusion

To address the challenge posed by the difficulty in identifying tea diseases amidst the intricate backdrop of tea gardens, a tea disease identification model based on ECA attention mechanism and ResNet50 network was proposed, namely ECA-ResNet50. In this study, utilizing ResNet50 as the fundamental network structure enhances the model's capability to discern tea disease traits within the intricate environment of tea gardens. Using three 3×3 convolutional kernels to replace the 7×7 convolutional kernels of

TABLE 4 Comparative experiments with other datasets.

Model	Plant species	Type of disease	Precision%	Recall%	F1%	Accuracy%
ResNet50	Apple	Scab	79.60	97.50	87.65	89.39
		Black rot	97.50	99.50	98.49	
		Red Star Disease	100	71.20	83.18	
	Corn	Gray spot disease	88.00	92.50	90.19	93.55
		rust	99.60	98.70	99.15	
		Big spot disease	92.10	88.40	90.21	
ECA-ResNet50	Apple	Scab	100	96.50	98.22	98.82
		Black rot	98.00	100	98.99	
		Red Star Disease	98.50	100	99.24	
	Corn	Gray spot disease	89.50	94.50	91.94	94.65
		rust	99.60	99.60	99.60	
		Big spot disease	94.10	88.90	91.43	

TABLE 5 Comparative experiments with other models.

Model	Accuracy%	Precision%	Recall%	F1%
AlexNet	90.38	90.75	90.38	90.56
MobileNet	85.88	85.30	85.88	85.59
VGG16	91.25	91.41	91.25	91.33
ResNet34	90.63	90.70	90.63	90.66
ResNet50	89.88	89.90	89.88	88.89
ResNet101	90.44	90.54	90.44	90.49
InceptionResnetv2	93.63	93.78	93.63	93.71
ECA-ResNet50	93.06	93.09	93.06	93.07
Transformer	94.49	94.38	94.10	94.20

the first layer of ResNet50, the strategy of using multi-layer small convolutional kernels can not only refine the granularity of feature extraction and improve the accuracy of disease identification, moreover, it augments the model's learning prowess and intricacy while optimizing performance through parameter reduction and network depth enhancement. The incorporation of the ECA attention mechanism fosters the model's ability to prioritize salient feature details within the imagery, which effectively enhanced the learning and recognition ability of tea disease characteristics and improved the overall performance of the model. Compared with the original ResNet50 model, the identification accuracy of ECA-ResNet50 on the tea disease dataset was improved by 3.18%. At the same time, its performance is also better than that of six other commonly used network models (such as AlexNet, MobileNet, VGG16, etc.). In addition, the ECA-ResNet50 model has also achieved good results in other plant datasets, which fully demonstrates the effectiveness and generalization of the model.

In this study, the tea disease identification model based on the ECA attention mechanism and ResNet50 network realized the accurate and efficient identification of seven tea diseases and one healthy leaf in the complex background of tea garden, which has certain significance for the prevention and control of tea garden diseases. However, the number of tea diseases in the dataset used in this study was relatively small, and some of the diseases were similar in color and characteristics, and may even appear in the same leaf, presenting a complex disease combination. In subsequent studies, the number of images of tea diseases will be expanded; The versatility and robustness of the model will be further improved, and the design will be lightweight to be embedded in different mobile equipment for tea gardening, thus, offering valuable insights for the intelligent oversight and management of the tea cultivation industry.

Data availability statement

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

Author contributions

LL: Writing – original draft, Writing – review & editing. YZ: Funding acquisition, 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 supported by the sub project fund of China's National Key Research and Development Program (2020YFD1100605-02).

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

References

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *Comput. Sci.*, 1–11.

Chen, X. (2022). Occurrence trend and green prevention and control of tea diseases in China. *Tea China* 44, 7–14.

Dhanya, V. G., Subeesh, A., and Kushwaha, N. L. (2022). Deep learning based computer vision approaches for smart agricultural applications. *Artif. Intell. Agric.* 6, 211–229. doi: 10.1016/j.aiia.2022.09.007

Fe, I. W., Jiang, M., Chen, Q., Yang, S., and Tang, X. (2017). Residual attention network for image classification. *IEEE*, 6450–6458.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *IEEE*, 770–778. doi: 10.1109/CVPR.2016.90

Hu, J., Shen, L., Sun, G., and Wu, E. (2019). Squeeze-and-excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 2011–2023. doi: 10.1109/TPAMI.34

Huang, M., Wang, S., and Zhou, Z. (2024). Tomato leaf disease identification based on improved AlexNet network. *J. Ningxia Normal Univ.* 45, 78–89.

Ji, X., Huo, X., and Xue, D. (2020). Identification method of crop diseases and insect pests based on deep learning. *South China Agric. Machin.* 51, 182–183.

Li, J. (2017). Research on automatic identification of tobacco diseases based on convolutional neural network. Tai'an City, Shandong Province, China: Shandong Agricultural University, 1–10.

Li, Y., Li, Z., and Fu, S. (2023). Disease identification of cultivated alfalfa based on AlexNet. *Acta Pratacult. Sin.* 32, 104–114.

Lu, Y., Guo, D., and Shen, H. (2018). Research on rice leaf disease identification method based on deep learning. *Inf. Record. Mater.* 19, 177–179.

Qiu, C., Tian, G., and Zhao, J. (2024). Strawberry disease identification based on improved YOLOv5. *Chin. J. Agric. Mechan.* 45, 198–204.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L. C. (2018). Inverted residuals and linear bottlenecks: mobile networks for classification, detection and segmentation 1–10.

Sun, C., and Lin, H. (2022). A method for detecting apple leaf diseases based on ensemble learning. *Jiangsu Agric. Sci.* 50, 41–47.

Sun, J., Tan, W., and Mao, H. (2017). Transactions of the CSAE 33, 209–215.

Technicolor T, Related S, Technicolor T and Related SOR (2019). ImageNet classification with deep convolutional neural networks [50 12–20.

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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 11531–11539 (IEEE).

Wang, L., Zhang, B., and Yao, J. (2021). Potato leaf disease identification and spot detection based on convolutional neural network. *Chin. J. Agric. Mechan.* 42, 122–129.

Wu, H. (2019). Identification method of tomato leaf disease based on deep residual network. *Smart Agric.* 1, 42–49.

Xu, L., Choy, C.-S., and Li, Y.-W. (2016). "Deep sparse rectifier neural networks for speech denoising," in *2016 IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*. 1–5.

Yu, X., Gong, Q., and Chen, C. (2023). Soybean pod morphology classification based on machine vision and convolutional neural network. *J. Biosyst. Eng.* 48, 26–35. doi: 10.1007/s42853-022-00174-6

Zhao, X., Wu, Z., and Chao, L. (2024). Research on building extraction based on CBAM VGG16-UNet semantic segmentation model. *J. Qiqihar University(Natural Sci. Edition)* 03), 1–7.

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 02 May 2025
DOI 10.3389/fpls.2025.1499909

OPEN ACCESS

EDITED BY
Mohsen Yoosefzadeh Najafabadi,
University of Guelph, Canada

REVIEWED BY
Hariharan Shanmugasundaram,
Vardhaman College of Engineering, India
Kelly Lais Wiggers,
Federal Technological University of Paraná, Brazil

*CORRESPONDENCE
Gopal Sangar
sangarrajagopal@gmail.com

RECEIVED 22 September 2024
ACCEPTED 27 March 2025
PUBLISHED 02 May 2025

CITATION
Sangar G and Rajasekar V (2025) Optimized classification of potato leaf disease using EfficientNet-LITE and KE-SVM in diverse environments. *Front. Plant Sci.* 16:1499909. doi: 10.3389/fpls.2025.1499909

Optimized classification of potato leaf disease using EfficientNet-LITE and KE-SVM in diverse environments

Gopal Sangar* and Velswamy Rajasekar

Department of Computer Science and Engineering,Faculty of Engineering and Technology, SRM Institute of Science and Technology, Vadapalani, Chennai, India

Introduction: Potatoes are a vital global product, and prompt identification of foliar diseases is imperative for sustaining healthy yields. Computer vision is essential in precision agriculture, facilitating automated disease diagnosis and decision-making through real-time data. Inconsistent data in uncontrolled contexts undermines classic image classification techniques, hindering precise illness detection.

Methods: We present a novel model that integrates EfficientNet-LITE for enhanced feature extraction with KE-SVM Optimization for effective classification. KE-SVM Optimization cross-references misclassified instances with correct classifications across kernels, iteratively refining the confusion matrix to improve accuracy across all classes. EfficientNet-LITE improves the model's emphasis on pertinent features through Channel Attention (CA) and 1-D Local Binary Pattern (LBP), while preserving computational economy with a reduced model size of 12.46 MB, fewer parameters at 3.11M, and a diminished FLOP count of 359.69 MFLOPs.

Results: Before optimization, the SVM classifier attained an accuracy of 79.38% on uncontrolled data and 99.07% on laboratory-controlled data. Following the implementation of KE-SVM Optimization, accuracy increased to 87.82% for uncontrolled data and 99.54% for laboratory-controlled data.

Discussion: The model's efficiency and improved accuracy render it especially appropriate for settings with constrained computational resources, such as mobile or edge devices, offering substantial practical advantages for precision agriculture.

KEYWORDS

EfficientNet-LITE, KE-SVM optimization, channel attention, 1-D local binary pattern, Sobel edge augmentation, uncontrolled environment data, potato leaf disease

1 Introduction

Crop and plant diseases lead to substantial revenue drops, incurring elevated disease management expenses and financial losses for farmers globally. Potatoes serve as a fundamental food source in India, which ranks as the second-largest producer globally, contributing over 15% to worldwide potato production. In India, potatoes are grown on around 2 million hectares, yielding 56 million tons (Mishra et al., 2024), thereby playing a crucial role in food security and the economy of agriculture. Potato crops experience yield losses of 5% to 15% owing to leaf diseases (Mishra et al., 2024), necessitating the implementation of effective disease management methods. Precisely diagnosing and categorizing diseases under diverse conditions is important for effective disease management. Conventional methods (Singla et al., 2024) necessitated manual field scouting, resulting in delayed disease diagnosis. These approaches are both inefficient and subjective, depending on visual evaluations conducted by trained plant pathologists. Computer vision-based image analysis (Gulame et al., 2023; Tholkapiyan et al., 2023) has been developed to address these constraints, enabling rapid and precise disease identification. However, initial solutions primarily focused on feature engineering to define particular attributes for each illness, which is unfeasible for the extensive variety of plant species and diseases. This has concluded in increased dependency on deep learning (DL) to provide more generalized and scalable options.

In recent years, deep learning has gained prominence because to developments in Graphics Processing Units (GPUs), increased storage space, and the availability of vast datasets. Convolutional Neural Networks (CNNs) (Huang et al., 2023) have become highly favored for the recognition and classification of plant diseases owing to their capacity to independently extract and learn optimal features from images. Although they perform well in controlled settings, numerous models fail to reproduce these outcomes with field data acquired under uncontrolled conditions (Shabrina et al., 2024). To mitigate this deficiency, the EfficientNet-LITE model, based on Convolutional Neural Networks (CNN) (Haque et al., 2022; Khamparia et al., 2020; Nagaraju and Chawla, 2022; Thakur et al., 2022), was utilized to extract pertinent and advanced features from images, facilitated by the incorporation of Channel Attention (CA) (Chen et al., 2021) and 1-D Local Binary Pattern (LBP) (Rachmad et al., 2022) features. The incorporation of 1-D LBP for texture analysis from feature maps is a distinctive method that markedly improved the model's capacity to identify complex patterns in uncontrolled settings. Additionally, Sobel edge-detected samples were incorporated into the improved dataset, providing an innovative method to improve edge information during training. Furthermore, KE-SVM Optimization (Deepti, 2023; Shrivastava et al., 2023) was employed to enhance classification by optimizing (Sorensen and Nielsen, 2018) SVM kernels and producing superior prediction data. This integrated methodology attained elevated precision in both regulated laboratory settings and demanding outdoor environments. The primary contributions of the paper are outlined below.

- The EfficientNet-LITE model, with the innovative incorporation of Channel Attention and the original utilization of 1-D Local Binary Pattern features, substantially enhanced the accuracy of plant disease classification, especially in severe uncontrolled situations. This distinctive integration enabled the model to concentrate more efficiently on pertinent image attributes.
- The incorporation of Sobel edge-detected samples into the supplemented dataset greatly enhanced the model's capacity to capture and leverage edge information, consequently raising classification performance.
- The KE-SVM Optimization utilized a kernel ensemble and presented an innovative method to enhance the confusion matrix by revisiting misclassified samples and accurately categorizing them with other kernels. This novel approach successfully reduced the constraints of conventional SVMs, resulting in enhanced classification efficiency across various datasets.
- The integration of EfficientNet-LITE with KE-SVM Optimization demonstrated a revolutionary methodology that attained higher accuracy and resilience. The model effectively generalized over both controlled and uncontrolled datasets.
- This research introduced an innovative, rapid, precise, and dependable approach for classifying plant diseases, thereby enhancing agricultural disease management, potentially reducing yield losses, and enabling informed decision-making for farmers.

Effective management of plant diseases requires timely and precise identification and classification. Development in artificial intelligence and machine learning has resulted in substantial enhancements in automated disease detection. This review examines contemporary methodologies and technologies, concentrating on image processing and deep learning models applied to various crops, with the objective of summarizing current achievements and pinpointing research opportunities.

Nabila Husna Shabrina et al. revealed shortcomings in the PlantVillage dataset for the diagnosis of potato leaf diseases in real-world scenarios. To resolve this, they presented a novel dataset of 3,076 pictures obtained in uncontrolled settings, encompassing seven disease varieties. This dataset offers a more precise depiction of potato leaf conditions. Testing EfficientNetV2B3 (Shabrina et al., 2024) resulted in 73.63% accuracy on the new dataset, in contrast to 98.15% on PlantVillage.

Aanis Ahmad et al. investigated (Ahmad et al., 2023) the generalization capacity of deep learning (DL) models for diagnosing corn diseases in field conditions using many datasets, including PlantVillage, PlantDoc, Digipathos, NLB, and a proprietary CD&S dataset. Five deep learning architectures—InceptionV3, ResNet50, VGG16, DenseNet169, and Xception—were trained utilizing diverse dataset pairings. DenseNet169 exhibited enhanced performance, achieving an accuracy of 81.60% using RGBA images from the CD&S dataset after

background removal. Furthermore, the amalgamation of field-acquired and laboratory data, encompassing sources from PlantVillage and CD&S, yielded an accuracy range of 77.50% to 80.33%, hence improving model generalization for field application.

Penghui Gui et al. tackled the issue of identifying plant diseases in uncontrolled field environments. They proposed an enhanced CNN model for field plant (Gui et al., 2021) disease identification (FPDR), incorporating strategies such as backdrop substitution and leaf resizing to optimize data augmentation. To improve feature differentiation, they employed a channel orthogonal constraint and utilized species categorization as a supplementary task. Utilizing the proprietary Field-PlantVillage (Field-PV) dataset, comprising 665 field photos, the model attained an accuracy of 72.03%, representing a substantial enhancement from 41.81%, despite being exclusively trained on the PlantVillage dataset.

A. Ubaidillah et al. sought to improve the categorizing of corn diseases using Random Forest, Neural Network, and Naive Bayes (Ubaidillah et al., 2022) techniques. The study utilized a compilation of corn leaf photographs obtained from agricultural regions in the Madura Region, concentrating on four classifications: healthy, gray leaf spot, blight, and common rust. The Neural Network technique outperformed the alternatives, with an AUC of 90.09%, a classification accuracy of 74.44%, an F1-score of 72.01%, precision of 74.14%, and recall of 74.43%, so establishing it as the most effective model for detecting maize diseases.

Priyanka Sahu and associates proposed a Deep-Dream (DD) architecture (Sahu et al., 2023) for Crop Leaf Disease Detection (CLDD), amalgamating deep learning (DL) with machine learning (ML) techniques. The study utilized the tomato crop dataset from PlantVillage and created 24 Hybrid Deep Neural (HDN) models, utilizing EfficientNet (B0-B7) as a feature extractor in conjunction with classifiers such as Random Forest (RF), AdaBoost (ADB), and Stochastic Gradient Boosting (SGB). The DD-EffiNet-B4-ADB model achieved optimal accuracy, ranging from 84% to 96%.

Hieu Phan et al. presented a deep learning approach utilizing Simple Linear Iterative Clustering (SLIC) segmentation (Phan et al., 2022) to identify diseased regions on corn leaves. The study employed five pre-trained models—VGG16, ResNet50, DenseNet121, Xception, and InceptionV3—on the PlantVillage and CD&S datasets, concentrating on super-pixel classes like northern leaf blight, gray leaf spot, and common rust. One hundred models were trained using diverse segments and split ratios. DenseNet121 achieved a peak accuracy of 97.77% on the CD&S dataset, employing five segments per image and an 80:20 split. Web and mobile applications were developed for disease identification, demonstrating the effectiveness of automated disease tracking relative to manual monitoring.

Mohit Agarwal et al. devised an efficient CNN model of 8 hidden layers (Agarwal et al., 2020) for the identification of tomato illnesses, therefore alleviating the computational demands linked to pre-trained models. Their approach, assessed with the PlantVillage dataset, achieved an accuracy of 98.4%, surpassing traditional machine learning methods (94.9% with k-NN) and pre-trained models like VGG16 (93.5%). The research employed image pre-processing techniques to enhance efficiency, achieving an accuracy of 98.7% on additional datasets. This study highlights the effectiveness and efficiency of lightweight (Zhu et al., 2023) CNN (Dai et al., 2023) models for disease detection in tomato crops.

Hasibul Islam Peyal and associates developed a lightweight 2D CNN model employing deep learning for the categorization of diseases in tomato and cotton plants. The algorithm, incorporated into an Android application named "Plant Disease Classifier," (Peyal et al., 2023) proficiently categorized 14 classifications, consisting of 12 diseased and 2 healthy categories. Despite having fewer variables than pre-trained models like VGG16, VGG19, and InceptionV3, it achieved an impressive average accuracy of 97.36%, with precision, recall, and F1-scores around at 97%, and an Area under Curve (AUC) score of 99.9%. The utilization of Grad-CAM for visual interpretations and the model's rapid classification time of around 4.84ms highlight its efficiency and effectiveness in disease detection.

Qiang Dai et al. created DATFGAN, a generative adversarial network that employs dual-attention and topology-fusion techniques to enhance the identification of agricultural disease photos. DATFGAN (Dai et al., 2020) improves image clarity and resolution, alleviating issues related to unclear images that hinder identification accuracy. The network's weight-sharing approach reduces the parameter count, and actual evidence demonstrates that DATFGAN produces visually superior results and significantly outperforms existing methods in practical identification tasks.

Junde Chen et al. developed the Crop Disease Recognition Model (CDRM), including the Location-wise Soft Attention mechanism (Ubaidillah et al., 2022) into a pre-trained MobileNet-V2 to enhance the detection of subtle lesion features. This model addresses challenges associated with chaotic backgrounds and variable lighting in crop disease images. The study's experimental results demonstrated an average accuracy of 99.71% on an open-source dataset, with a 99.13% accuracy in challenging conditions. The proposed method outperforms prior dominant techniques, showcasing its effectiveness and robustness in detecting agricultural illnesses.

Rabbia Mahum et al. proposed an enhanced deep learning technique for the diagnosis and categorization of potato leaf diseases. Unlike existing methods that categorize potato leaves into two groups utilizing the Plant Village dataset, their approach classifies leaves into five separate categories: Potato Late Blight (PLB), Potato Early Blight (Feng et al., 2023) (PEB), Potato Leaf Roll (PLR), Potato Verticillium Wilt (PVw), and Healthy (PH). Their model achieved an accuracy of 97.2% by utilizing a pre-trained Efficient DenseNet (Mahum et al., 2022) model, integrating an additional transition layer, and implementing a reweighted cross-entropy loss function. This method effectively tackles class imbalance and overfitting, offering a robust solution for comprehensive disease classification in potato leaves.

Zubair Saeed and associates developed a deep learning system focused on computer vision for the early detection and classification of potato leaf diseases. Utilizing deep convolutional neural networks (Saeed et al., 2021), specifically ResNet-152 and InceptionV3, trained on the Kaggle potato dataset, their methodology achieved accuracies of 98.34% and 95.24%, respectively, with a learning rate

of 0.0005. The method precisely classifies potato leaves into three categories: healthy, early blight, and late blight. This method aims to mitigate economic losses by enabling the prompt detection of disease outbreaks through accurate image-based categorization.

Kashif Shaheed et al. developed EfficientRMT-Net, a novel model that combines Vision Transformer (ViT) with ResNet-50 (Shaheed et al., 2023) for the automated detection and classification of potato leaf diseases. This technology addresses the limitations of traditional methods, such as labor-intensive procedures and inadequate illness detection. EfficientRMT-Net utilizes CNN for feature extraction, depth-wise convolution to reduce processing demands, and a stage block architecture to enhance scalability and sensitivity. The model, trained on bespoke datasets, achieved accuracies of 97.65% on a generic dataset and 99.12% on a tailored potato leaf dataset. EfficientRMT-Net offers a dependable approach for accurate disease classification, consequently improving crop yield and resource efficiency.

Mingjie Lv and associates devised a maize leaf disease recognition method to tackle challenges including variable lighting and complexities in feature extraction. Their methodology integrates a maize leaf enhancement framework and the DMS-Robust AlexNet, an advanced neural network (Lv et al., 2020) based on AlexNet. This network incorporates dilated and multi-scale convolutions to improve feature extraction. It utilizes batch normalization to reduce overfitting, with the PReLU activation function and Adabound optimizer to improve convergence and precision. Experimental results demonstrate that this technique significantly enhances disease identification in complex scenarios, providing a dependable alternative for advanced plant disease diagnostics.

Hatice Catal Reis and Veysel Turk developed the Multi-head Attention Mechanism Depthwise Separable Convolution Inception Reduction Network (MDSCIRNet) for the early identification of potato leaf diseases. This deep convolutional neural network utilizes depthwise separable convolutions and a multi-head attention mechanism to enhance classification accuracy. MDSCIRNet (Reis and Turk, 2024) achieved an accuracy of 99.33% by combining deep learning with SVM, outperforming contemporary algorithms such as Xception and MobileNet, as well as traditional methods like SVM and Random Forest. The study highlights the effectiveness of MDSCIRNet in improving early disease detection and reducing financial losses for agricultural producers.

Xiangpeng Fan and Zhibin Guan address critical challenges in maize disease identification with their proposed VGNet, a system that employs a pretrained VGG16 model. VGNet incorporates batch normalizing, global average pooling, and L2 normalization to enhance performance. Utilizing transfer learning and the Adam optimizer, the model achieves an accuracy of 98.3% with a learning rate of 0.001, exhibiting remarkable precision and recall for nine maize diseases. VGNet's small architecture (Fan and Guan, 2023), requiring only 79.5 MB, enables efficient processing, demonstrating effective disease recognition with a testing duration of 75.21 seconds for 230 images.

The reviewed literature demonstrates significant advancements in plant disease classification learning models, (Saritha and Thangaraja, 2023; Shahoveisi et al., 2023) using deep learning and machine learning models, yet several limitations persist. Many studies rely heavily on the PlantVillage dataset, which, while comprehensive, is collected in controlled environments and lacks diversity for real-world applications. For instance, Nabila Husna Shabrina et al. and Penghui Gui et al. highlighted the challenges of generalization in uncontrolled settings. Additionally, while methods such as DenseNet and EfficientNet have been explored, the absence of innovative feature extraction techniques, such as attention mechanisms and edge detection, limits their performance in detecting fine-grained features. Furthermore, traditional classifiers like SVMs, as used by A. Ubaidillah et al., often suffer from limitations in handling misclassified samples, reducing overall efficiency. Despite efforts to enhance accuracy, many studies fail to effectively combine lightweight models with robust optimization techniques for scalable and practical applications.

The proposed methods address these gaps by introducing EfficientNet-LITE with Channel Attention (Haider et al., 2024; Kumar et al., 2023; Navrozidis et al., 2018) and 1-D Local Binary Pattern (LBP) features, enabling precise focus on critical attributes even in uncontrolled environments. The inclusion of Sobel edge-detected samples enhances fine-detail recognition, while KE-SVM Optimization revisits and corrects misclassified samples, significantly improving classification efficiency. This integrated approach achieves superior generalization across diverse datasets, offering a fast, accurate, and reliable solution for real-world agricultural disease management, ultimately empowering farmers to reduce yield losses.

The remainder of the article is organized as follows: Section 2 outlines the structure of the feature extraction and classification model. Section 3 examines the experimental findings and analysis, while Section 4 presents the conclusions and future directions.

2 Materials and methods

The proposed approach initiates with image augmentation and Sobel edge identification to improve and diversity the dataset. Figure 1 illustrates the application of an attention-based EfficientNet-LITE model for feature extraction to identify essential leaf attributes, succeeded by KE-SVM optimization for precise classification of potato leaf diseases across diverse environments.

2.1 Dataset collection

This work utilized two datasets for the detection of potato leaf diseases: one from an uncontrolled environment (Shabrina et al., 2024) in Indonesia and the PlantVillage Dataset (Potato Species) (Shaheed et al., 2023) from a controlled laboratory setting. The first dataset, acquired from a Kaggle source, was compiled from multiple potato farms throughout Java Island by teams from Universitas Multimedia Nusantara and Universitas Gadjah Mada. It comprises 3,076 photos categorized into seven disease types: Figure 2 (a). virus, (b). phytophthora, (c). nematode, (d). fungal, (e). bacteria, (f). pest, and (g). healthy, taken under various settings. Figure 2 presents the

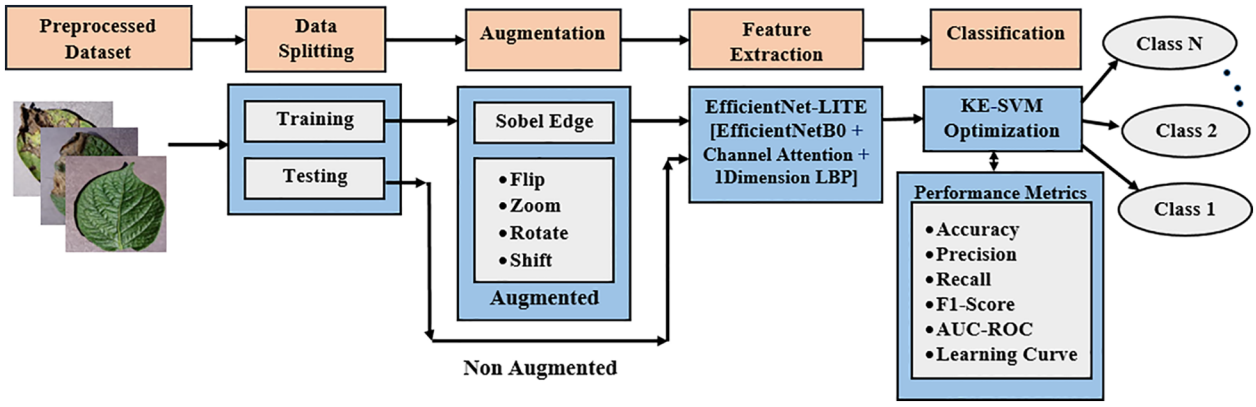


FIGURE 1
Proposed methodology for potato leaf disease classification.

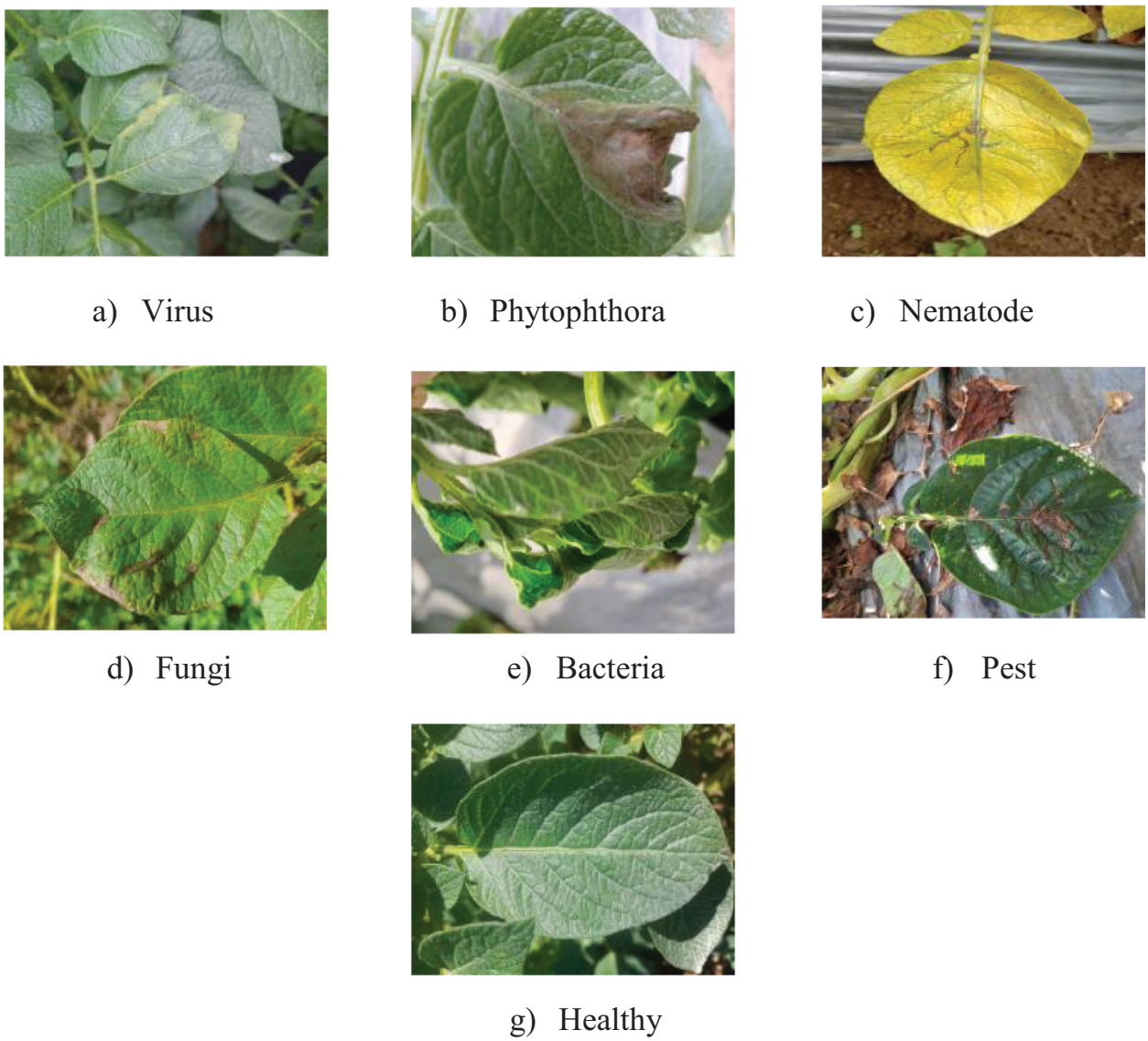


FIGURE 2
Samples of the seven categories in the potato leaf disease dataset: **(a)** Virus, **(b)** Phytophthora, **(c)** Nematode, **(d)** Fungal, **(e)** Bacteria, **(f)** Pest, and **(g)** Healthy.

sample photographs for each class. Each image possesses a resolution of 1500 × 1500 pixels and is stored in.jpg format for accessibility and compatibility with image-processing software.

The second dataset, PlantVillage (potato species), has 2,152 photos categorized into three classes: Healthy, Potato Late Blight, and Potato Early Blight, captured under uniform lighting circumstances with a resolution of 256 x 256 pixels. Both datasets provide a significant contrast between real-world and controlled settings for assessing model efficacy in disease diagnosis.

2.2 Preprocessing

Use bilinear interpolation (cv2.INTER_LINEAR) (Shabrina et al., 2024) to resize 1500x1500 potato leaf disease images to 224x224 pixels for machine learning models. This scaling was necessary to match image dimensions to models. We picked bilinear interpolation because it smoothed images while maintaining crucial characteristics and particulars from the high-resolution originals. Preprocessing the potato leaf disease images reduced computational effort and memory utilization, optimizing model performance and preparing the dataset for training and evaluation.

2.3 Data augmentation strategy

A complete data augmentation technique was applied to expand the training dataset of potato leaf disease image and improve the performance and resilience of the machine learning model. The initial dataset consisted of 3,076 pictures, with 2,460 allocated for training and 616 left aside for testing. Various augmentation strategies were employed to generate a more diverse and comprehensive training dataset, substantially enhancing the quantity of training samples.

Multiple fundamental augmentation methods were employed (Shabrina et al., 2024) to synthetically enlarge the training dataset. Rotation within a 20-degree range was implemented to imitate diverse viewing angles, enhancing the model's capacity to generalize across multiple orientations. Width and height adjustments of up to 20% of the image dimensions were executed to simulate differences in image positioning. Furthermore, shear transformations with a magnitude of 0.2 were implemented to produce tilting effects, facilitating the model's ability to manage images with perspective deviations. Zoom changes, with modifications of up to 20%, emulated various focal lengths and scales. Horizontal flips were utilized to mirror pictures and augment the model's resilience to variations in orientation.

Sobel edge detection was employed to enhance the edges and transitions in the potato leaf disease images. Employing the OpenCV library, Sobel filters calculated gradients in both the x and y directions, yielding edge-detected representations of the source images. This technique enhanced texture and boundary information, which was integrated into the training dataset. The edge-detected images were merged with the augmented versions generated through fundamental changes, enhancing the dataset with intricate edge information.

The enhancing method was efficiently performed by processing images of potato leaf disease in phases. Each image in a batch was

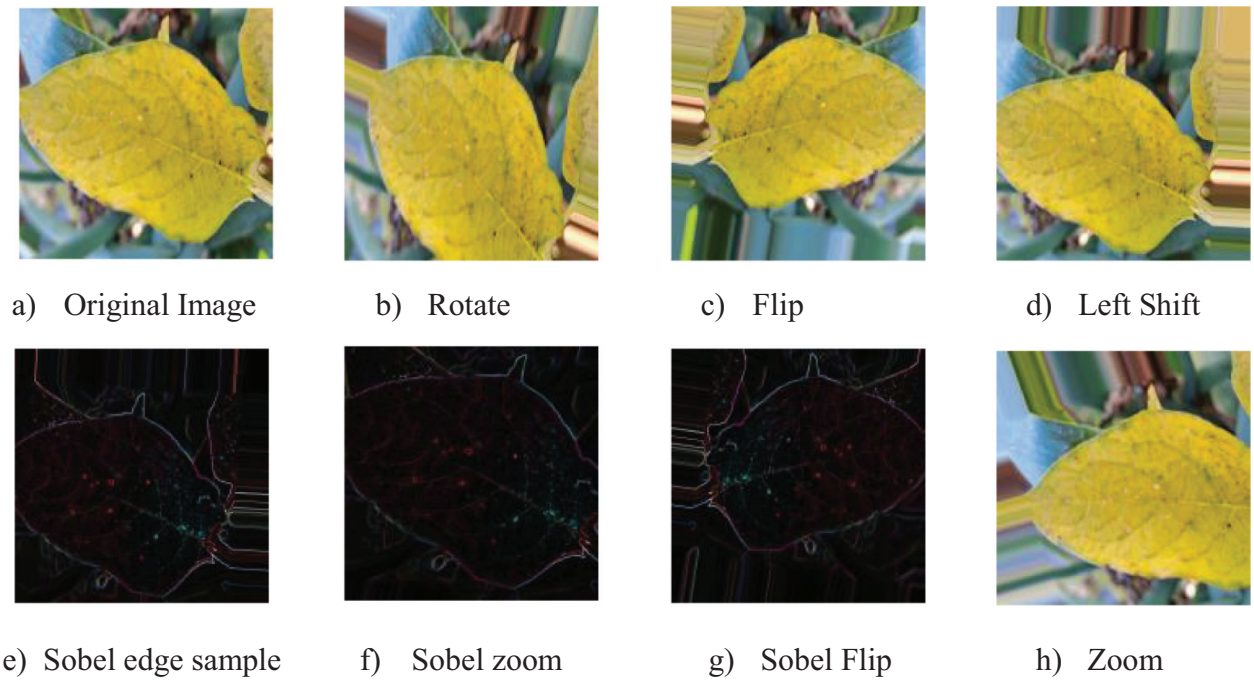


FIGURE 3
Sample images demonstrating original and augmented versions using various techniques: **(a)** Original Image, **(b)** Rotate, **(c)** Flip, **(d)** Left Shift, **(e)** Sobel Edge Sample, **(f)** Sobel Zoom, **(g)** Sobel Flip, and **(h)** Zoom.

initially converted to float32 format and augmented to incorporate a batch dimension. Six specific augmentations were done to each image with Keras's ImageDataGenerator class, enabling transformations including rotation, shifting, shearing, zooming, and flipping. Furthermore, Sobel edge detection was executed to produce further variations. Figures 3a–h illustrates the modified photos, accompanied by their respective labels, image names, and class names, which were subsequently gathered and preserved for model training.

This augmentation method led to a significant increase in the quantity of training samples. The initial training dataset of 2,460 photos was enlarged to 14,760 augmented samples (Xiong et al., 2020), incorporating those enhanced by Sobel edge detection. The quantity of original testing samples stayed at 616 and was not increased. The augmentation of the training dataset yielded a more varied collection of images, markedly improving the model's capacity to generalize and excel in multiple circumstances.

2.4 Feature extraction

EfficientNet-LITE is an enhanced version of the basic EfficientNetB0 (Upadhyay et al., 2024) design, specifically engineered to improve feature extraction through the strategic integration of a Channel Attention (CA) mechanism and 1-D Local Binary Pattern (LBP) for features. The improvements implemented post-Global Average Pooling layer are designed to augment the model's capacity to concentrate on pertinent features in images of diseased potato leaves, thus improving performance while preserving computational efficiency.

EfficientNet-LITE preserves the key principles of EfficientNetB0, which optimizes network depth, width, and resolution for enhanced accuracy with reduced parameters and FLOPs, while incorporating an attention mechanism for more targeted feature extraction. Figure 4 (Reproduced from (Upadhyay et al., 2024)) shows the combination of EfficientNetB0 with Channel Attention mechanism. In contrast to EfficientNetB0, which depends exclusively on convolutional processes and depthwise separable convolutions (Reis and Turk, 2024), EfficientNet-LITE's incorporation of Channel Attention and 1-D LBP enables the network to dynamically emphasize significant features. This produces a model that is both efficient and proficient at identifying nuanced patterns and details in potato leaf images, rendering it especially suitable for jobs demanding high accuracy with constrained computational resources.

The incorporation of the Channel Attention mechanism with 1-Dimensional LBP in EfficientNet-LITE tackles certain issues in feature extraction.

2.4.1 Channel Attention (CA)

Channel Attention operates by initially condensing the spatial dimensions of the input tensor into a channel descriptor by global average pooling. This description encapsulates the overall context for each channel, succinctly conveying its significance.

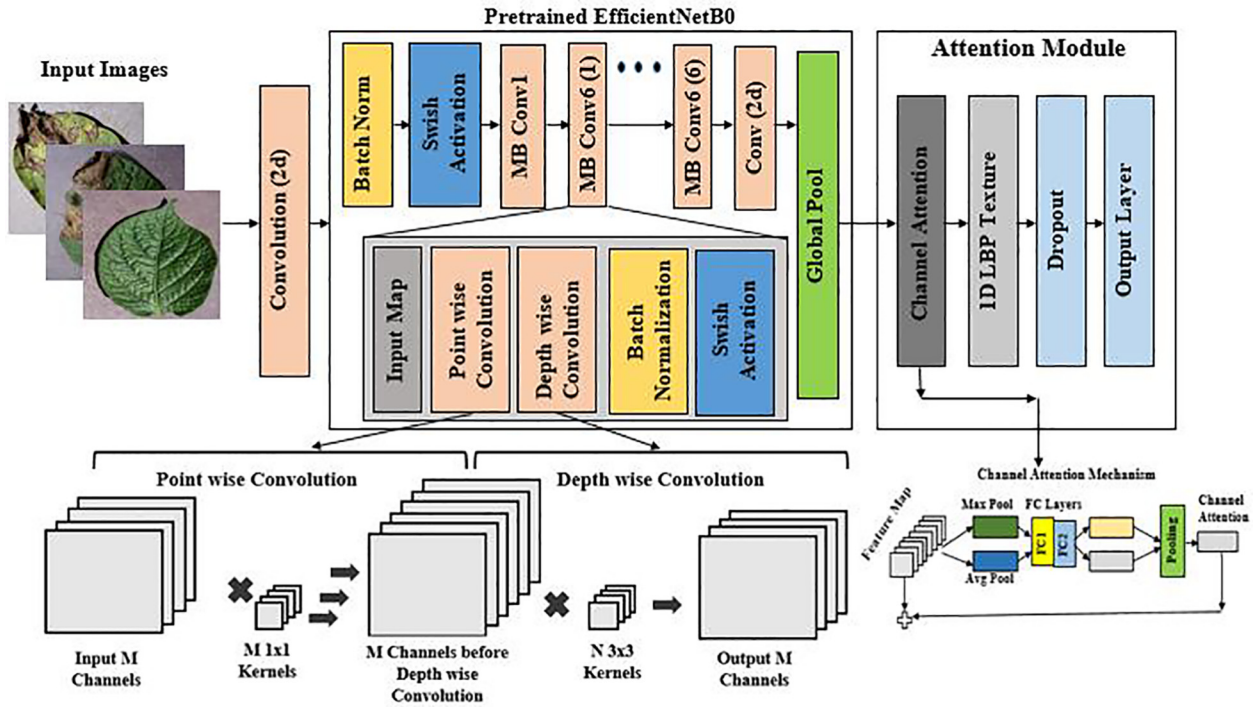


FIGURE 4
The architecture of EfficientNet-LITE Model with channel attention mechanism.

$$z_c = \frac{1}{H \star W} \sum_{h=1}^{H} \sum_{w=1}^{W} X_{b,c,h,w} \quad (1)$$

In Equation 1, z_c the global average is pooled value for channel c, $X_{b,c,h,w}$ is the value of the input tensor at batch b, channel c, height h, and width w.

The two completely connected layers subsequently convert this description into a series of attention weights. The initial fully connected layer diminishes the descriptor's dimensionality, whereas the subsequent fully connected layer reverts it to the original channel dimension. The ReLU activation introduces non-linearity, while the sigmoid activation guarantees that attention weights remain constrained between 0 and 1.

The vector z is then passed through two fully connected (FC) layers to generate channel attention weights:

$$\text{First FC Layer}: \quad y_1 = ReLU(W_1 z + b_1) \quad (2)$$

In Equation 2, W_1 is the weight matrix of the first fully connected layer, b_1 is the bias vector of the first fully connected layer, $ReLU$ is the Rectified Linear Unit activation function.

$$\text{Second FC Layer}: \quad y_2 = W_2 y_1 + b_2 \quad (3)$$

In Equation 3, W_2 is the weight matrix of the second fully connected layer, b_2 is the bias vector of the second fully connected layer.

Apply a sigmoid activation function to obtain the channel attention weights:

$$a_c = \sigma(y_2) \quad (4)$$

In Equation 4, σ is the sigmoid function, a_2 is the attention weight of channel c.

Ultimately, these attention weights are employed to scale the original input tensor, accentuating channels with greater weights and reducing the influence of channels with lesser weights. This approach allows the model to concentrate on the most pertinent aspects, enhancing its capacity to derive significant information from the incoming data.

2.4.2 1-D Local Binary Pattern (1D LBP):

1-D Local Binary Pattern (1-D LBP) is a method for identifying textural features from one-dimensional data, such sequential signals or feature vectors obtained from photographs. It operates by juxtaposing each data point with its adjacent counterparts to produce a binary pattern, subsequently transformed into a decimal code. The codes are compiled into a histogram that illustrates the distribution of local textures within the data points. This approach is resilient to periodic changes and effectively identifies critical local structures, including edges and peaks. The 1-D LBP (Algorithm 1) histogram offers a concise and distinctive feature descriptor that is efficient for signal classification and texture analysis tasks.

Input: 1D signal $X = \{X_1, X_2, \ldots, X_n\}$, Number of neighbors P
Output: LBP codes for each point in the signal
Step 1: Initialize Parameters
$P \leftarrow$ Number of neighbors;
Step 2: **Compute LBP Codes for Each Point in the Signal**;
for each point i from $P + 1$ to $N - P$ **do**
  $LBP_i \leftarrow 0$;
  for each neighbor j from 1 to $2P$ **do**
    if $j \leq P$ **then**
      $X_j \leftarrow X_{i-P+j-1}$;
    else
      $X_j \leftarrow X_{i+j-P}$;
    if $X_j \geq X_i$ **then**
      $S(X_i, X_j) \leftarrow 1$;
    else
      $S(X_i, X_j) \leftarrow 0$;
    $LBP_i \leftarrow LBP_i + S(X_i, X_j) \cdot 2^{j-1}$;
Step 3: **Return LBP Codes**;
Output: LBP_i for each i

Algorithm 1. 1–D Local Binary Pattern (1–D LBP).

2.4.3 Model Structure:

The Table 1 below summarizes the modified structure of EfficientNet-LITE, detailing the input and output shapes at each stage, along with the expansion factors, repeat times, and strides.

The proposed EfficientNet-LITE model was meticulously engineered with a systematic arrangement of layers to attain a compromise between computing efficiency and performance. The input layer received potato leaf pictures measuring 224×224×3, which were subsequently processed through a Conv2D layer that downsampled the input to 112×112×32 with a stride of 2, thus diminishing the spatial dimensions while augmenting the channel depth. Batch Normalization and Swish Activation are utilized to stabilize and non-linearly activate the refined feature maps, priming them for the ensuing MBConv blocks.

The Swish activation function is defined Equation 5 as:

$$Swish(x) = x \cdot \sigma(x) \quad (5)$$

where $\sigma(x)$ is the sigmoid function, given by Equation 6:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (6)$$

The MBConv layers facilitate effective feature extraction by gradually diminishing spatial dimensions while augmenting the amount of channels, culminating in a dense and compact feature representation. The model subsequently employed a 1x1 convolution to refine the features, followed by global pooling and a Channel Attention mechanism, which improved the model's capacity to concentrate on the most pertinent channels. This was succeeded by

TABLE 1 Structure of the proposed model.

Operators (modules)	Input shapes	Expansion factor	Output shapes	Repeat times	Strides
Input Layer	224 × 224 × 3	–	224 × 224 × 3	1	–
Conv2d	224 × 224 × 3	–	112 × 112 × 32	1	2
BatchNorm	112 × 112 × 32	–	112 × 112 × 32	1	–
Swish Activation	112 × 112 × 32	–	112 × 112 × 32	1	–
MBConv1	112 × 112 × 32	1	112 × 112 × 16	1	1
MBConv6	112 × 112 × 16	6	56 × 56 × 24	2	2
MBConv6	56 × 56 × 24	6	28 × 28 × 40	2	2
MBConv6	28 × 28 × 40	6	14 × 14 × 80	3	2
MBConv6	14 × 14 × 80	6	14 × 14 × 112	3	1
MBConv6	14 × 14 × 112	6	7 × 7 × 192	4	2
MBConv6	7 × 7 × 192	6	7 × 7 × 320	1	1
Conv2d 1 × 1	7 × 7 × 320	–	7 × 7 × 1280	1	1
Globalpool	7 × 7 × 1280	–	1 × 1280	1	–
Channel Attention	1 × 1280		1 × 1280	1	–
1-D LBP	1 × 1280		1 × 1290	1	
Dropout	1290	–	1290	1	–
Output Layer	1290	–	num_classes	1	–

a 1-D Local Binary Pattern (LBP) layer that expanded the feature vector to 1290 dimensions by integrating texture features.

2.4.4 Performance Comparison: EfficientNet-LITE vs EfficientNet-B0

In deep learning, determines like Floating Point Operations (FLOPs), parameter count, model size, and depth are essential for evaluating the performance and efficiency of neural network models. FLOPs measure a model's computational complexity, whereas the parameter count reflects its ability to learn and express intricate aspects. The model's size pertains to storage demands, whereas depth frequently associates with the model's capacity to discern complex patterns within the data.

EfficientNet-LITE had 359.69 MFLOPs, somewhat less than EfficientNet-B0's 390. EfficientNet-LITE required fewer computational resources due to its lower FLOPs, making it ideal for mobile or edge devices. Despite adding Channel Attention and 1-D LBP features, EfficientNet-LITE maintained a computational efficiency similar to EfficientNet-B0, demonstrating its design efficiency. There are 3.11 million parameters in EfficientNet-LITE, compared to 5.3 million in B0. EfficientNet-LITE's reduced parameters indicate a more streamlined architecture for memory-constrained applications. EfficientNet-LITE's 12.46 MB model size was lower than EfficientNet-B0's 20 MB due to fewer parameters. The compactness of EfficientNet-LITE accelerated model loading, memory usage, and inference times, making it better for real-time applications. Table 2 shows the size of pre-trained network model.

Also important is model depth, as deeper models can learn complex representations. EfficientNet-LITE had 27 layers, compared to 24 for EfficientNet-B0. This increased depth suggested that EfficientNet-LITE could capture more complex data characteristics, improving performance in sophisticated feature extraction tasks. The comparable FLOPs show that the extra depth did not reduce computing efficiency. EfficientNet-LITE balanced computational efficiency with model capacity. EfficientNet-LITE was ideal for mobile or embedded systems with limited computational resources because to its low FLOPs, parameter count, and model size. Despite being smaller, the model's depth let it accomplish complex tasks well.

Finally, EfficientNet-LITE has fewer parameters (3.11 million) and a smaller model (12.46 MB vs. 20 MB) (Ubaidillah et al., 2022) than EfficientNet-B0. It has more layers (27 vs. 24) but fewer FLOPs (359.69 vs. 390), requiring fewer computations. EfficientNet-LITE was more resource-efficient and performed well.

2.5 KE-SVM optimization (kernel ensemble SVM optimization)

SVMs were widely employed in image classification and machine learning to define class boundaries. By translating input information into high-dimensional spaces, SVM classifiers (Sorensen and Nielsen, 2018) accurately handled complex and non-linear patterns in many applications.

TABLE 2 The model size of the main networks.

Networks	Model size	Parameters	Depth
VGG16	528 MB	138 million	23
Inception V3	92 MB	23.8 million	159
ResNet50	98 MB	25.6 million	–
DenseNet121	33 MB	8.1 million	121
MobileNet-V1	16 MB	4.2 million	88
MobileNet-V2	14 MB	3.5 million	88
NASNetMobile	23 MB	5.2 million	–
EfficientNet-B0	20 MB	5.3 million	24
EfficientNet-LITE	**12.46 MB**	**3.11 million**	**27**

Bold values indicate the best performance.

However, datasets from uncontrolled environments with different backdrops, perspectives, and lighting conditions were difficult. Inconsistencies in image acquisition caused SVM kernels to struggle. Ensemble approaches (Sorensen and Nielsen, 2018) in machine learning improve performance by combining different models. This helped classify potato leaf diseases, where the dataset's unpredictability required a more robust technique.

Kernel-Ensemble SVM (KE-SVM) Optimization used Linear, Polynomial, Radial Basis Function (RBF), and Sigmoid SVM kernels to address these issues. KE-SVM Optimization enhanced classification accuracy and discussed dataset variability by capturing different data features and integrating their predictions. KE-SVM Optimization improves classification by combining SVM kernel strengths. Figures 5, 6 shows the work flow of KE-SVM method. This method compares misclassified instances in one kernel against proper classifications in others. The optimum confusion matrix is iteratively adjusted using this ensemble technique to optimize classification accuracy across all classes.

The novelty of this work lies in the application of Kernel-Ensemble SVM (KE-SVM) Optimization (Algorithm 2) to substantially improve classification efficacy by harnessing the advantages of several SVM kernels. Misclassified samples from the kernel exhibiting the highest accuracy were verified against predictions from alternative kernels, with those accurately classified by other kernels deemed as True Positives. The iterative modification process persisted until all classes were sufficiently addressed, resulting in significant enhancements in classification performance.

The potato leaf disease dataset, obtained from uncontrolled conditions, demonstrated that the SVM RBF kernel initially gave the highest performance among the kernels, attaining an accuracy of 79.38%. The Linear kernel achieved an accuracy of 72.89%, followed by the Polynomial kernel at 71.27%, and the Sigmoid kernel at 64.12%. The classification metrics and confusion matrix indicated a necessity for enhancement owing to the dataset's heterogeneity, including differing backdrops and lighting conditions.

```
Result: Optimized confusion matrix and evaluation metrics (accuracy, precision, recall, F1 score)
initialization;
confusion matrices ← [];
csv files ← [];
kernels ← {'linear','poly','rbf','sigmoid'};
while each kernel k ∈ kernels do
    svm classifier ← SVC (kernel = k, probability = True);
    svm classifier.fit (X_train resampled, Y_train resampled);
    ypred ← svm classifier.predict (X_test features);
    predictions df ← {X_test features, Class Name, True Label, Predicted Label};
    csv filename ← base path + 'predicted labels' + k + '.csv';
```

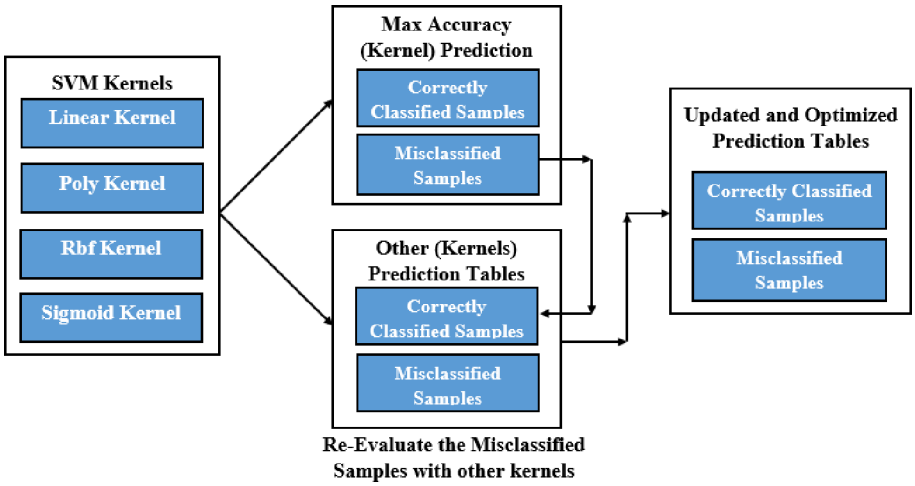


FIGURE 5
Block diagram of KE-SVM optimization.

```
    save (predictions df, csv filename);
    csv files.append (csv filename);
    Evaluate the Model;
    accuracy ← accuracy score (y_test, y_pred);
    precision ← precision score (y_test, y_pred,'weighted');
    recall ← recall score (y_test, y_pred, 'weighted');
    f1 ← f1 score (y_test, y_pred,'weighted');
    cm ← confusion matrix (y_test, y_pred);
    confusion matrices.append (cm);
end while
Determine the best kernel;
best index ← argmax ({accuracy (cm) for each cm ∈ confusion matrices});
best matrix ← confusion matrices [best index];
optimized matrix ← copy (best matrix);
while each sample with (true label ≠ best pred label) in csv files [best index]do
    for each i ≠ best index in csv files do
        if other preds [i] = true label for sample then
            optimized matrix [true label, best pred label]
            ← optimized matrix [true label, best pred label] − 1;
            optimized matrix [true label, true label]
            ← optimized matrix [true label, true label] + 1;
            break;
        end if
    end for
end while
Compute and print optimized metrics;
(accuracy, precision, recall, f1−score) ← calculate metrics (optimized matrix);
print (optimized matrix);
print (accuracy, precision, recall, f1−score);
```

Algorithm 2. KE–SVM Optimization.

The SVM classifiers with different kernels attained good accuracy on the lab-controlled dataset from PlantVillage (potato species). The Polynomial kernel attained the maximum accuracy of 99.07%, succeeded by the RBF kernel at 98.84%, the Linear kernel at 98.38%, and the Sigmoid kernel at 96.06%. The classification report indicated an exceptional performance, with an overall accuracy of 1.00. The precision, recall, and F1-scores were remarkably elevated across all categories, indicating the consistent conditions of the dataset. The confusion matrix revealed minimal misclassifications, illustrating the effectiveness of the SVM Polynomial kernel in controlled laboratory circumstances.

The EfficientNet-LITE + SVM model demonstrated higher performance on datasets from both controlled and uncontrolled settings. Following KE-SVM optimization, the accuracy on the PlantVillage dataset rises to 99.54%, while on the uncontrolled environment dataset, it dramatically climbs to 87.82%, showing the model's improved capacity to manage intricate, uncontrolled conditions.

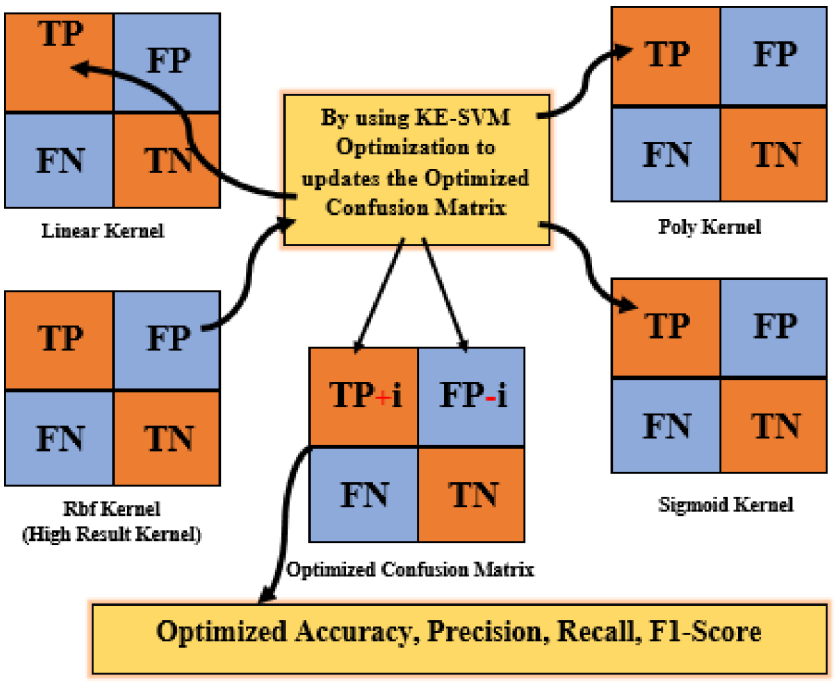


FIGURE 6
Misclassified samples re-evaluate with other kernels.

TABLE 3 Shows the results of both datasets before optimization.

Model	Dataset	Accuracy	Precision	Recall	F1-score
EfficientNet-LITE + SVM	Potato Leaf Disease in Uncontrolled Environment	79.38%	80%	79%	79%
EfficientNet-LITE + SVM	PlantVillage (Potato Species)	99.07%	99%	99%	99%

3 Result and discussions

This study's results are structured into three primary stages: (1) results before augmentation, (2) results before optimization, and (3) results after KE-SVM optimization. These stages comprehensively illustrate the progression in performance of the SVM classifiers when applied to controlled (PlantVillage) and uncontrolled environment datasets for diagnosing potato leaf diseases. The evaluation metrics employed include accuracy, precision, recall, F1-score, and other relevant measures to validate the model's effectiveness. Equation 7, Equation 8, Equation 9, Equation 10 employed to calculate these measures were included to clarify the evaluation procedure.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$F1\ \ Score = \frac{2{*}Precision{*}Recall}{Precision + Recall} \tag{10}$$

Accuracy evaluated overall correctness, whereas precision and recall examined the management of false positives and negatives. The F1-score offered a comprehensive assessment of the model's classification performance, as illustrated in Table 3 below.

The initial experiments were conducted using the raw dataset without applying Sobel edge filtering or augmentation techniques. The SVM classifier's performance in uncontrolled and controlled environments revealed significant room for improvement. In the uncontrolled environment dataset, the accuracy was 75.62%, while in the lab-controlled dataset, the accuracy was 98.62%. These results underscore the challenges posed by the inherent variability in the uncontrolled environment dataset.

The lab-controlled dataset demonstrated high accuracy due to reduced variability and noise. Following data augmentation with

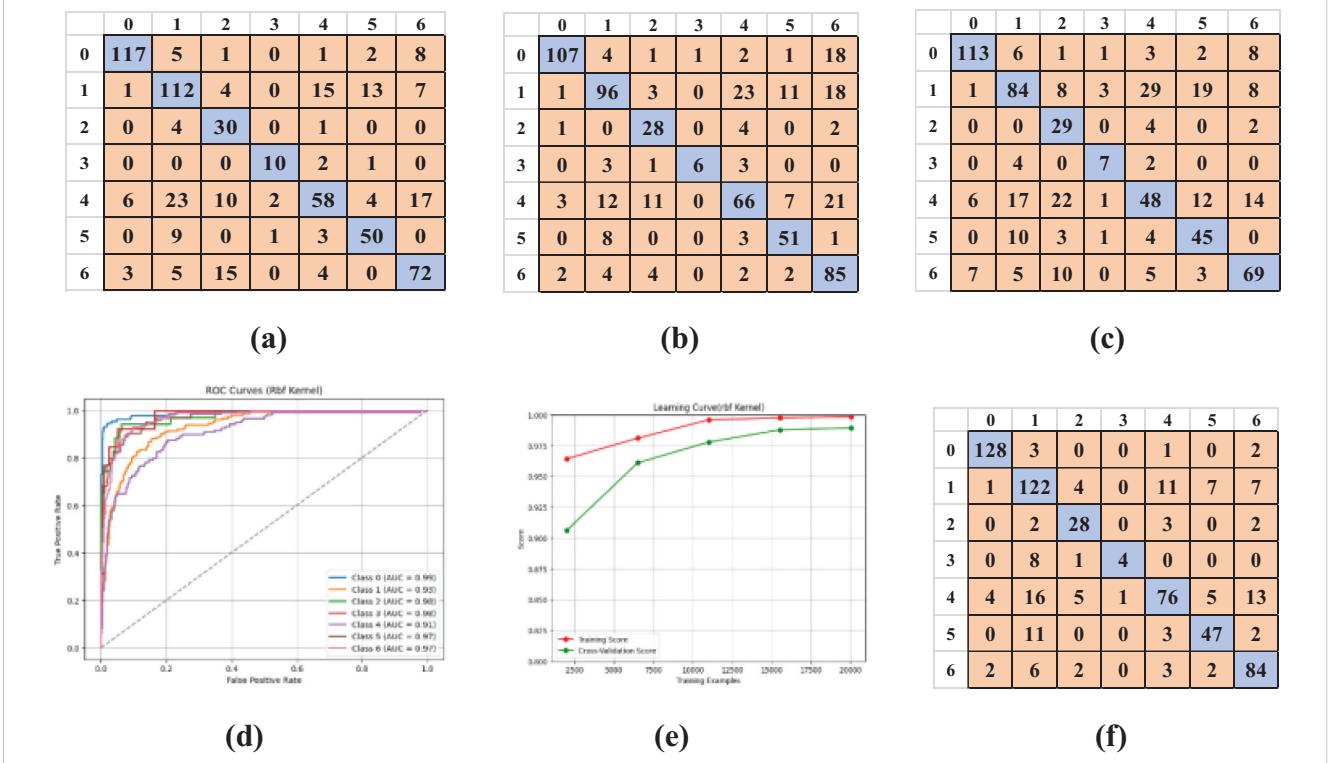


FIGURE 7
Confusion matrices of SVM kernels, AUC-ROC curve, and learning curve for the kernel with maximum accuracy (RBF). **(a)** Linear, **(b)** Polynomial, **(c)** Sigmoid, **(d)** AUC-ROC Curve, **(e)** Learning Curve, **(f)** RBF.

Sobel edge filtering to enhance feature extraction, the performance of the SVM classifiers was evaluated before applying the KE-SVM optimization technique. The augmented samples contributed to improved classification, particularly in uncontrolled environments.

A comprehensive examination of the SVM model was performed on the uncontrolled environment dataset utilizing four distinct kernels: Linear (Figure 7a), Polynomial (Figure 7b), RBF (Figure 7f), and Sigmoid (Figure 7c). Confusion matrices were produced for each kernel, offering insights into the model's classification proficiency across diverse categories: 0: Virus, 1: Phytophthora, 2: Nematode, 3: Fungi, 4: Bacteria, 5: Pest, 6: Healthy. Visual representations of these matrices are provided to illustrate the model's performance.

The overall effectiveness was evaluated by plotting the AUC-ROC curve (Figure 7d) and learning curves (Figure 7e) for the kernel exhibiting the highest accuracy. These visuals facilitated awareness of the model's capacity to generalize to unfamiliar data. To test the model's dependability, 5-fold cross-validation was employed. Table 4 results demonstrated constant performance across the folds, signifying the resilience of the SVM with RBF kernel, which attained the best accuracy.

The SVM model utilizing the RBF kernel exhibited robust performance, attaining an average training accuracy of 99.32 and a validation accuracy of 96.94. The minor discrepancy between these metrics signified effective generalization throughout the sample. The uniformity of results over the five folds further

TABLE 4 5-Fold cross validation for potato leaf disease in uncontrolled environment dataset.

Fold	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
Fold 1	0.993	0.9656	0.0796	0.4502
Fold 2	0.9932	0.969	0.0726	0.3942
Fold 3	0.9933	0.9736	0.0705	0.302
Fold 4	0.9936	0.9702	0.0626	0.3604
Fold 5	0.993	0.9686	0.0757	0.3837
Average	**0.9932**	**0.9694**	**0.0722**	**0.3781**

Bold values indicate the best performance.

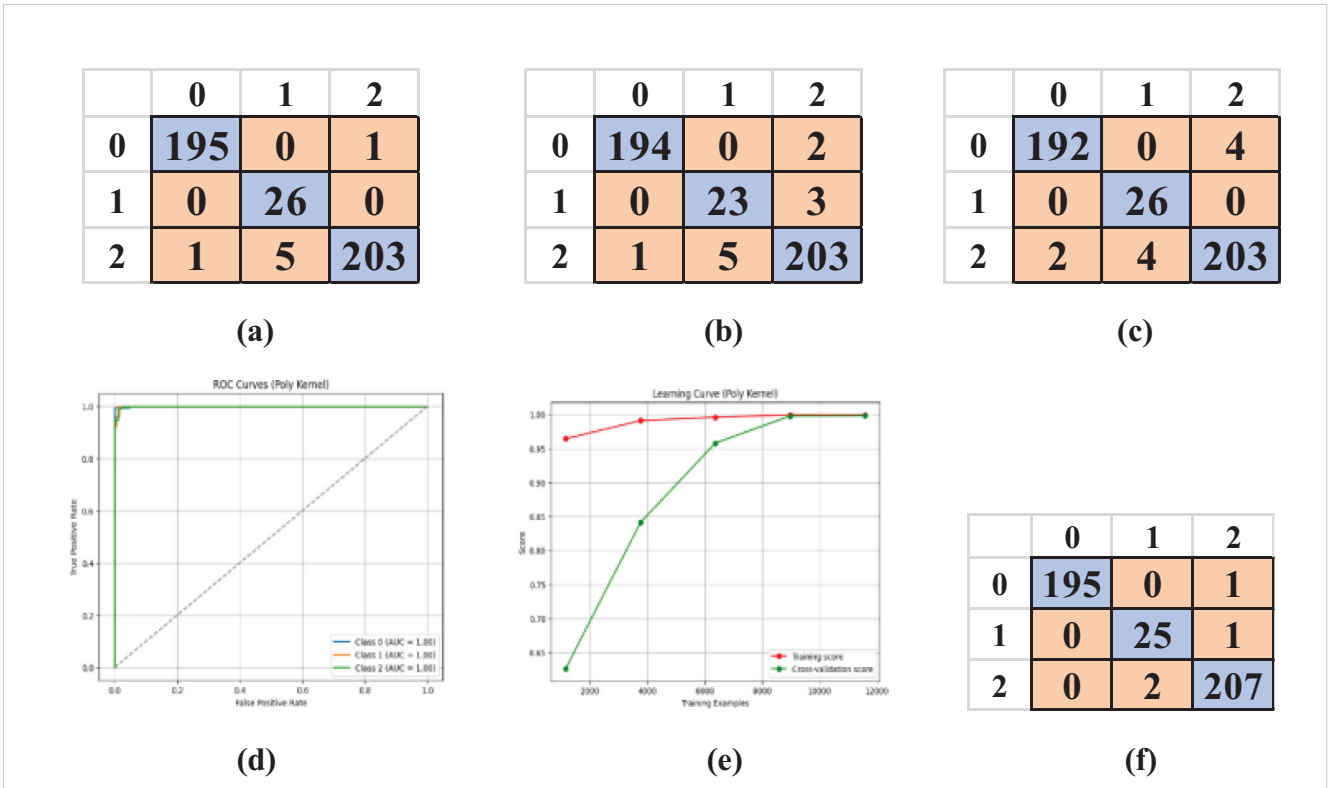


FIGURE 8
Confusion matrices of SVM kernels, AUC-ROC curve, and learning curve for the kernel with maximum accuracy (Polynomial). **(a)** Linear, **(b)** RBF, **(c)** Sigmoid, **(d)** AUC-ROC Curve, **(e)** Learning Curve, **(f)** Polynomial.

TABLE 5 5-Fold cross validation for PlantVillage dataset (potato species).

Fold	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
Fold 1	0.9996	1.0000	0.0004	0.000
Fold 2	0.9997	0.9972	0.0003	0.0028
Fold 3	0.9997	0.9986	0.0003	0.0024
Fold 4	0.9996	0.9979	0.0004	0.0031
Fold 5	0.9996	0.9983	0.0004	0.0038
Average	**0.9996**	**0.9984**	**0.0004**	**0.0024**

Bold values indicate the best performance.

emphasized the model's resilience, even in an uncontrolled setting. The RBF kernel effectively captured intricate correlations within the data, demonstrating its appropriateness for the dataset's inherent unpredictability. The model's excellent accuracy underscored its efficacy in classifying leaf diseases.

In the lab-controlled dataset, identical SVM kernels were utilized, and confusion matrices were produced for each kernel: Linear (Figure 8a), Polynomial (Figure 8f), RBF (Figure 8b), and Sigmoid (Figure 8c). It offers insights into the model's categorization proficiency across different categories: 0: Early Blight, 1: Healthy, 2: Late Blight. The findings from this dataset exhibited remarkably high accuracy owing to the controlled environment, which minimized data fluctuation.

The model's performance was additionally assessed by plotting the AUC-ROC curve (Figure 8d) and the learning curve (Figure 8e) for the optimal kernel. These curves demonstrated nearly flawless generalization. Consistent with the uncontrolled dataset, 5-fold cross-validation validated the model's reliability, with Table 5 indicating minimal variance among the folds.

The SVM model utilizing a polynomial kernel was assessed on laboratory-controlled data, demonstrating superior performance across all five folds. The model attained an average training accuracy of 99.96 and a validation accuracy of 99.84. The training loss of 0.0004 and validation loss of 0.0024 signify little error and robust generalization in a regulated environment. The results highlight the efficacy of the polynomial kernel in managing clean, organized data, exhibiting little variability relative to uncontrolled contexts.

3.1 After optimization

After implementing KE-SVM Optimization, the model's performance on the uncontrolled environment dataset shown significant enhancement. The optimal accuracy increased to 87.82%, accompanied by enhancements in precision to 86.77%,

TABLE 6 Optimized results for both lab and uncontrolled dataset.

Model	Dataset	No. of Classes	Accuracy
EfficientNet-LITE + SVM	Potato Leaf Disease in Uncontrolled Environment	07	87.82%
EfficientNet-LITE + SVM	PlantVillage(Potato Species)	03	99.54%

	0	1	2	3	4	5	6
0	130	3	0	0	0	0	1
1	1	131	2	0	8	5	5
2	0	0	33	0	2	0	0
3	0	1	1	11	0	0	0
4	4	5	5	1	88	4	13
5	0	6	0	0	1	56	0
6	0	1	2	0	2	2	92

(a)

	0	1	2
0	195	0	1
1	0	26	0
2	0	1	208

(b)

FIGURE 9
Optimized confusion matrices for both datasets: **(a)** Uncontrolled data and **(b)** Laboratory-controlled data.

TABLE 7 To compare the results with existing state-of-art-methods for uncontrolled dataset.

Author & Year	Model Name	Dataset	Accuracy
Penghui Gui, et al., 2021	CNN	Field-PV	72.03%
A Ubaidillah, et al., 2022	ANN	Cotton Disease (Field Data)	74.44%
AANIS AHMAD, et al., 2023	DenseNet169(RGBA)	Field-PV	77.50%
Shabrina, et al., 2024	EfficientNetV2B3	Potato Leaf Disease in Uncontrolled Environment	73.63%
	MobileNetV3-Large		72.03%
	VGG-16		59.81%
	ResNet50		68.17%
	DenseNet121		59.16%
Proposed Model	**EfficientNet-LITE (Before Optimization)**	Potato Leaf Disease in Uncontrolled Environment	**79.38%**
Proposed Model	**EfficientNet-LITE (After Optimization)**	Potato Leaf Disease in Uncontrolled Environment	**87.82%**

Bold values indicate the best performance.

TABLE 8 To compare the results with existing state-of-art-methods with PlantVillage(Potato) dataset.

Author & Year	Model Name	Dataset	Accuracy
Saeed Z et al., 2021	ResNet152	PlantVillage (Potato)	98.34%
	InceptionV3		95.24%
Rabia M et al., 2022	ResNet-202	PlantVillage (Potato)	97.2%
Shabrina et al., 2024	EfficientNetV2B3	PlantVillage (Potato)	98.15%
Jain J et al., 2024	EfficientNetB0	PlantVillage (Potato)	99.05
Proposed Model	**EfficientNet-LITE (Before Optimization)**	**PlantVillage (Potato)**	**99.07%**
Proposed Model	**EfficientNet-LITE (After Optimization)**	**PlantVillage (Potato)**	**99.54%**

Bold values indicate the best performance.

recall to 88.18%, and F1-score to 87.19%. The lab-controlled dataset has been somewhat enhanced to 99.54%. (Figure 9a) presents the optimized confusion matrix for uncontrolled data, while (Figure 9b) displays the matrix for the laboratory-controlled dataset.

The optimization approach improved classification by cross-validating misclassified examples from the most effective kernel with predictions from other kernels, resulting in a more precise and balanced confusion matrix. Table 6 presents the optimal outcomes of the KE-SVM optimization technique.

3.2 Comparative performance

The proposed model (EfficientNet-LITE + KE-SVM Optimization) exhibited substantial enhancements in accuracy relative to previous models utilized on comparable datasets. Prior to optimization, the model attained an accuracy of 79.38%, which rose to 87.82% post-optimization. This performance surpassed those of models like DenseNet121, ResNet50, and MobileNetV3-Large, which exhibited accuracies between 59.16% and 73.63%. This significant enhancement can be ascribed to the ensemble SVM kernel methodology and improved feature extraction with EfficientNet-LITE.

In the lab-controlled PlantVillage dataset, the suggested model attained nearly flawless accuracy both prior to and subsequent to KE-SVM Optimization. The model initially achieved an accuracy of 99.07%, which then increased to 99.54% during optimization. This performance surpassed other prominent models, including ResNet152, InceptionV3, and VGNet, which exhibited accuracies between 95.24% and 98.34%. The substantial enhancement upon optimization is attributable to the improved feature extraction and the strong classification capabilities of KE-SVM.

An optimized version of EfficientNetB0, EfficientNet-LITE, integrated Channel Attention (CA) and 1D Local Binary Pattern (LBP) features to increase feature extraction. This model prioritized potato leaf traits while being computationally efficient, making it ideal for resource-constrained mobile devices. KE-SVM Optimization used linear, polynomial, RBF, and sigmoid kernels to overcome typical SVM constraints. With SMOTE and confusion matrix optimization, classification accuracy improved, handling class imbalance and data variability.

The strengths of EfficientNet-LITE and KE-SVM Optimization were combined. EfficientNet-LITE's superior feature extraction and KE-SVM Optimization's classification created a model that could handle complex datasets. This collaboration produced high accuracy and reliable performance in all settings. The combined model exceeded expectations in early illness identification and uncontrolled environment management to satisfy research objectives. The results confirmed the model's efficacy and versatility in solving research problems.

Tables 7 and 8 highlight the superior performance of our proposed EfficientNet-LITE + KE-SVM Optimization model compared to existing methods. Notably, the model achieved an accuracy of 87.82% on uncontrolled datasets and 99.54% on the PlantVillage dataset, surpassing models such as DenseNet121 and ResNet50. These results underscore the robustness of our approach in handling variability and improving classification accuracy. The enhanced classification accuracy of our model has significant implications for agricultural diagnostics. By addressing challenges posed by uncontrolled environments, our model paves the way for reliable and resource-efficient solutions applicable in real-world farming scenarios. This contributes to the broader goal of precision agriculture and early disease detection.

4 Conclusion

In conclusion, our research revealed the effectiveness of combining EfficientNet-LITE with KE-SVM Optimization for the classification of potato leaf diseases. Initially, SVM classifiers demonstrated disparate performance, with the RBF kernel achieving 79.38% accuracy on uncontrolled data and the sigmoid kernel reaching 99.07% accuracy on laboratory-controlled data. Subsequent to KE-SVM Optimization, the accuracy on the uncontrolled dataset markedly increased to 0.8782, with precision at 86.77%, recall at 88.18%, and F1-score at 87.19%. Conversely, the accuracy on the lab-controlled dataset exhibited a minor enhancement to 99.54%. This integrated model adeptly tackles issues associated with early disease classification, dataset variability, and model robustness, demonstrating its versatility and dependability across many settings. Future work could explore integrating more comprehensive datasets that combine image data with clinical parameters such as plant height, size, irrigation schedules, and expert farmer insights. Additionally, leveraging generative AI techniques could provide holistic solutions for farmers, enhancing decision-making and improving crop management practices.

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 in the article/supplementary material.

Author contributions

GS: Conceptualization, Data curation, Investigation, Methodology, Resources, Software, Visualization, Writing – original draft, Writing – review & editing. VR: Conceptualization, Investigation, Project administration, Resources, Supervision, Validation, Writing – original draft, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

Conflict of interest

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

Publisher's note

References

Agarwal, M., Gupta, S. K., and Biswas, K. (2020). Development of Efficient CNN model for Tomato crop disease identification. *Sustain. Computing Inf. Syst.* 28, 100407. doi: 10.1016/j.suscom.2020.100407

Ahmad, A., Gamal, A. E., and Saraswat, D. (2023). Toward generalization of Deep Learning-Based plant disease identification under controlled and field conditions. *IEEE Access* 11, 9042–9057. doi: 10.1109/access.2023.3240100

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

Dai, Q., Cheng, X., Qiao, Y., and Zhang, Y. (2020). Crop leaf Disease Image Super-Resolution and Identification with dual attention and Topology Fusion Generative Adversarial Network. *IEEE Access* 8, 55724–55735. doi: 10.1109/access.2020.2982055

Dai, M., Sun, W., Wang, L., Dorjoy, M. M. H., Zhang, S., Miao, H., et al. (2023). Pepper leaf disease recognition based on enhanced lightweight convolutional neural networks. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1230886

Deepti, K. (2023). Comparative analysis of machine learning techniques for plant disease detection-data deployment. *J. Institution Engineers (India) Ser. B* 104, 837–849. doi: 10.1007/s40031-023-00897-w

Fan, X., and Guan, Z. (2023). VGNET: A lightweight intelligent learning method for corn diseases recognition. *Agriculture* 13, 1606. doi: 10.3390/agriculture13081606

Feng, J., Hou, B., Yu, C., Yang, H., Wang, C., Shi, X., et al. (2023). Research and validation of potato late blight detection method based on deep learning. *Agronomy* 13, 1659. doi: 10.3390/agronomy13061659

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

Gulame, M. B., Thite, T. G., and Patil, K. D. (2023). Plant disease prediction system using advance computational Technique. *J. Phys. Conf. Ser.* 2601, 12031. doi: 10.1088/1742-6596/2601/1/012031

Haider, I., Khan, M. A., Nazir, M., Hamza, A., Alqahtani, O., Alouane, M. T., et al. (2024). Crops leaf disease recognition from digital and RS imaging using fusion of multi Self-Attention RBNET deep architectures and modified dragonfly optimization. *IEEE J. Selected Topics Appl. Earth Observations Remote Sens.* 17, 7260–7277. doi: 10.1109/jstars.2024.3378298

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. doi: 10.1038/s41598-022-10140-z

Huang, Z., Jiang, X., Huang, S., Qin, S., and Yang, S. (2023). An efficient convolutional neural network-based diagnosis system for citrus fruit diseases. *Front. Genet.* 14. doi: 10.3389/fgene.2023.1253934

Khamparia, A., Singh, A., Luhach, A. K., Pandey, B., and Pandey, D. K. (2020). Classification and identification of primitive kharif crops using supervised deep convolutional networks. *Sustain. Computing Inf. Syst.* 28, 100340. doi: 10.1016/j.suscom.2019.07.003

Kumar, V. S., Jaganathan, M., Viswanathan, A., Umamaheswari, M., and Vignesh, J. (2023). Rice leaf disease detection based on bidirectional feature attention pyramid network with YOLO v5 model. *Environ. Res. Commun.* 5, 065014. doi: 10.1088/2515-7620/acdece

Lv, M., Zhou, G., He, M., Chen, A., Zhang, W., and Hu, Y. (2020). Maize leaf disease identification based on feature enhancement and DMS-Robust Alexnet. *IEEE Access* 8, 57952–57966. doi: 10.1109/access.2020.2982443

Mahum, R., Munir, H., Mughal, Z., Awais, M., Khan, F. S., Saqlain, M., et al. (2022). A novel framework for potato leaf disease detection using an efficient deep learning model. *Hum. Ecol. Risk Assess. an Int. J.* 29, 303–326. doi: 10.1080/10807039.2022.2064814

Mishra, P., Alhussan, A. A., Khafaga, D. S., Lal, P., Ray, S., Abotaleb, M., et al. (2024). Forecasting production of potato for a Sustainable Future: Global Market analysis. *Potato Res.* 67, 1671–1690. doi: 10.1007/s11540-024-09717-0

Nagaraju, M., and Chawla, P. (2022). Maize crop disease detection using NPNet-19 convolutional neural network. *Neural Computing Appl.* 35, 3075–3099. doi: 10.1007/s00521-022-07722-3

Navrozidis, I., Alexandridis, T., Dimitrakos, A., Lagopodi, A., Moshou, D., and Zalidis, G. (2018). Identification of purple spot disease on asparagus crops across spatial and spectral scales. *Comput. Electron. Agric.* 148, 322–329. doi: 10.1016/j.compag.2018.03.035

Peyal, H. I., Nahiduzzaman, M., Pramanik, A. H., Syfullah, M. K., Shahriar, S. M., Sultana, A., et al. (2023). Plant Disease Classifier: Detection of Dual-Crop diseases using lightweight 2D CNN architecture. *IEEE Access* 11, 110627–110643. doi: 10.1109/access.2023.3320686

Phan, H., Ahmad, A., and Saraswat, D. (2022). Identification of foliar disease regions on corn leaves using SLIC segmentation and deep learning under uniform background and field conditions. *IEEE Access* 10, 111985–111995. doi: 10.1109/access.2022.3215497

Rachmad, A., Syarief, M., Rifka, S., Sonata, F., Setiawan, W., and Rochman, E. M. S. (2022). Corn leaf disease classification using Local Binary Patterns (LBP) feature extraction. *J. Phys. Conf. Ser.* 2406, 12020. doi: 10.1088/1742-6596/2406/1/012020

Reis, H. C., and Turk, V. (2024). Potato leaf disease detection with a novel deep learning model based on depthwise separable convolution and transformer networks. *Eng. Appl. Artif. Intell.* 133, 108307. doi: 10.1016/j.engappai.2024.108307

Saeed, Z., Khan, M. U., Raza, A., Sajjad, N., Naz, S., and Salal, A. (2021). "Identification of leaf diseases in potato crop using deep convolutional neural networks (DCNNs)," in *2021 16th International Conference on Emerging Technologies (ICET)*. (Islamabad, Pakistan: IEEE). doi: 10.1109/icet54505.2021.9689807

Sahu, P., Chug, A., Singh, A. P., and Singh, D. (2023). Classification of crop leaf diseases using image to image translation with deep-dream. *Multimedia Tools Appl.* 82, 35585–35619. doi: 10.1007/s11042-023-14994-x

Saritha, S., and Thangaraja, G. A. (2023). Prediction of crop disease using Rank Regressive Learning and Proaftn Fuzzy Classification models. *Soft Computing*. doi: 10.1007/s00500-023-08357-9

Shabrina, N. H., Indarti, S., Maharani, R., Kristiyanti, D. A., Irmawati, N., Prastomo, N., et al. (2024). A novel dataset of potato leaf disease in uncontrolled environment. *Data Brief* 52, 109955. doi: 10.1016/j.dib.2023.109955

Shaheed, K., Qureshi, I., Abbas, F., Jabbar, S., Abbas, Q., Ahmad, H., et al. (2023). EfficientRMT-NeT—An efficient RESNET-50 and Vision Transformers approach for classifying potato plant leaf diseases. *Sensors* 23, 9516. doi: 10.3390/s23239516

Shahoveisi, F., Gorji, H. T., Shahabi, S., Hosseinirad, S., Markell, S., and Vasefi, F. (2023). Application of image processing and transfer learning for the detection of rust disease. *Sci. Rep.* 13. doi: 10.1038/s41598-023-31942-9

Shrivastava, V. K., Shelke, C. J., Shrivastava, A., Mohanty, S. N., and Sharma, N. (2023). Optimized deep learning model for disease prediction in potato leaves. *EAI Endorsed Trans. Pervasive Health Technol.* 9, 17. doi: 10.4108/eetpht.9.4001

Singla, A., Nehra, A., Joshi, K., Kumar, A., Tuteja, N., Varshney, R. K., et al. (2024). Exploration of machine learning approaches for automated crop disease detection. *Curr. Plant Biol.* 40, 100382. doi: 10.1016/j.cpb.2024.100382

Sorensen, L., and Nielsen, M. (2018). Ensemble support vector machine classification of dementia using structural MRI and mini-mental state examination. *J. Neurosci. Methods* 302, 66–74. doi: 10.1016/j.jneumeth.2018.01.003

Thakur, P. S., Sheorey, T., and Ojha, A. (2022). VGG-ICNN: A Lightweight CNN model for crop disease identification. *Multimedia Tools Appl.* 82, 497–520. doi: 10.1007/s11042-022-13144-z

Tholkapiyan, M., Devi, B. A., Bhatt, D., Kumar, E. S., Kirubakaran, S., and Kumar, R. (2023). Performance analysis of rice plant diseases identification and classification methodology. *Wireless Pers. Commun.* 130, 1317–1341. doi: 10.1007/s11277-023-10333-3

Ubaidillah, A., Rochman, E. M. S., Fatah, D. A., and Rachmad, A. (2022). Classification of corn diseases using random forest, neural network, and naive bayes methods. *J. Phys. Conf. Ser.* 2406, 12023. doi: 10.1088/1742-6596/2406/1/012023

Upadhyay, S. K., Jain, J., and Prasad, R. (2024). Early blight and late blight disease detection in potato using Efficientnetb0. *Int. J. Exp. Res. Rev.* 38, 15–25. doi: 10.52756/ijerr.2024.v38.002

Xiong, Y., Liang, L., Wang, L., She, J., and Wu, M. (2020). Identification of cash crop diseases using automatic image segmentation algorithm and deep learning with expanded dataset. *Comput. Electron. Agric.* 177, 105712. doi: 10.1016/j.compag.2020.105712

Zhu, X., Li, J., Jia, R., Liu, B., Yao, Z., Yuan, A., et al. (2023). LAD-Net: a novel light weight model for early Apple Leaf Pests and Diseases Classification. *IEEE/ACM Trans. Comput. Biol. Bioinf.* 20, 1156–1169. doi: 10.1109/tcbb.2022.3191854

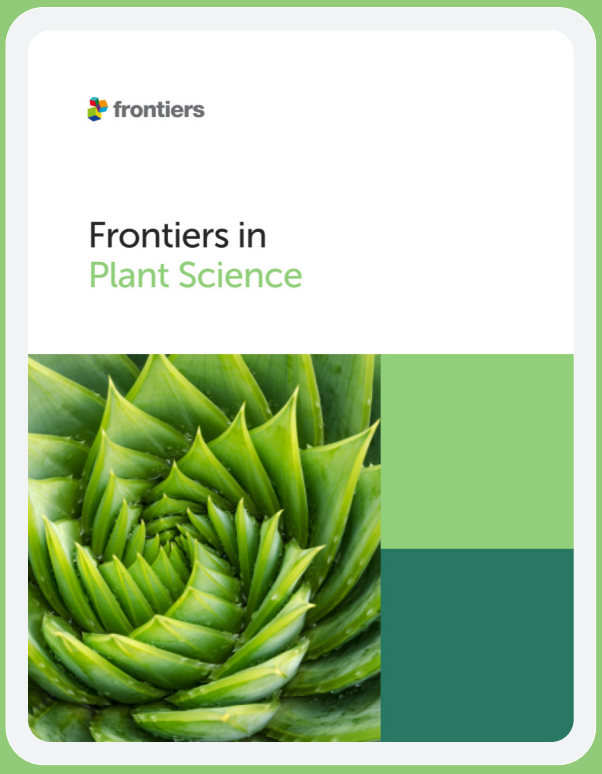
Frontiers in
Plant Science

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