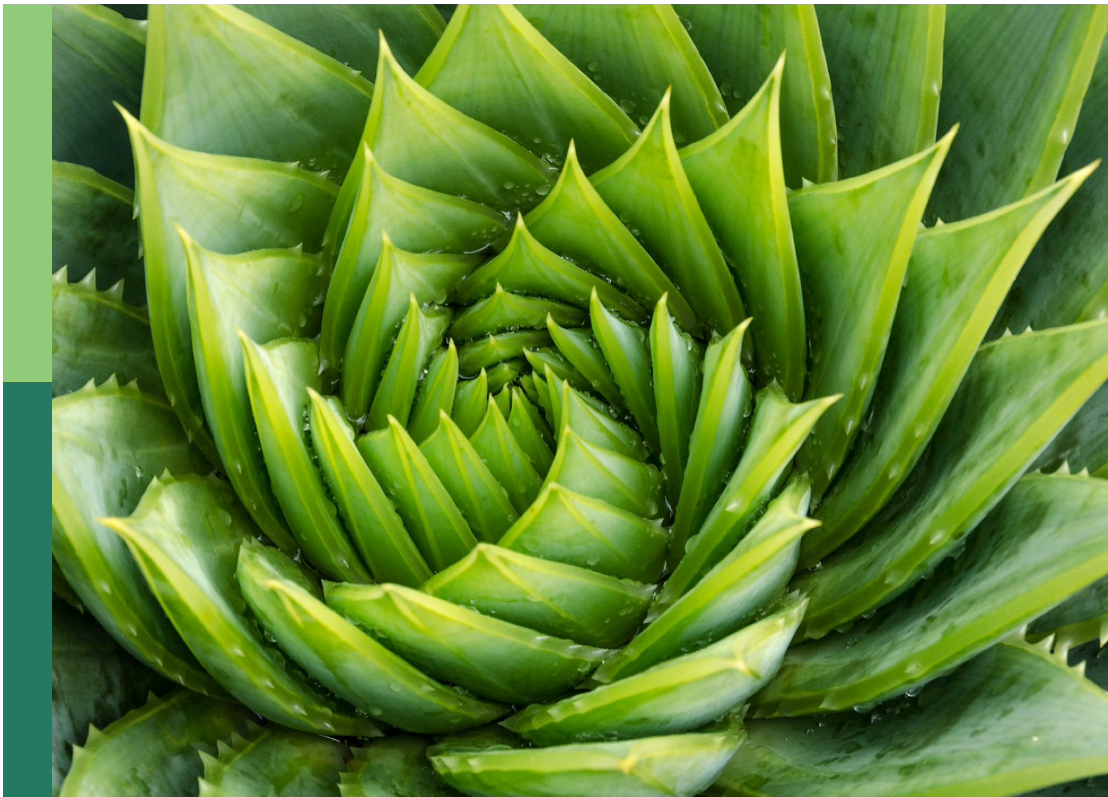
frontiers | Research Topics

# Computer vision in plant phenotyping and agriculture

**Edited by**
Valerio Giuffrida, Hanno Scharr and Ian Stavness

**Published in**
Frontiers in Plant Science
Frontiers in Artificial Intelligence



ISSN 1664-8714
ISBN 978-2-8325-1065-0
DOI 10.3389/978-2-8325-1065-0

## About Frontiers

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

## Frontiers journal series

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

## Dedication to quality

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

## What are Frontiers Research Topics?

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

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

# Computer vision in plant phenotyping and agriculture

**Topic editors**

Valerio Giuffrida — Edinburgh Napier University, United Kingdom

Hanno Scharr — Julich Research Center, Helmholtz Association of German Research Centres (HZ), Germany

Ian Stavness — University of Saskatchewan, Canada

**Citation**

Giuffrida, V., Scharr, H., Stavness, I., eds. (2023). *Computer vision in plant phenotyping and agriculture*. Lausanne: Frontiers Media SA.
doi: 10.3389/978-2-8325-1065-0

# Table of contents

frontiers | Frontiers in Artificial Intelligence

TYPE Editorial
PUBLISHED 16 May 2023
DOI 10.3389/frai.2023.1187301

Check for updates

OPEN ACCESS

EDITED AND REVIEWED BY
Matthew McCabe,
King Abdullah University of Science and Technology, Saudi Arabia

*CORRESPONDENCE
Ian Stavness
✉ ian.stavness@usask.ca

RECEIVED 15 March 2023
ACCEPTED 12 April 2023
PUBLISHED 16 May 2023

CITATION
Stavness I, Giuffrida V and Scharr H (2023) Editorial: Computer vision in plant phenotyping and agriculture. *Front. Artif. Intell.* 6:1187301. doi: 10.3389/frai.2023.1187301

# Editorial: Computer vision in plant phenotyping and agriculture

Ian Stavness[1]*, Valerio Giuffrida[2] and Hanno Scharr[3]

[1]Department of Computer Science, University of Saskatchewan, Saskatoon, SK, Canada, [2]School of Computing, Napier University, Edinburgh, United Kingdom, [3]Institute for Advanced Simulation, IAS-8: Data Analytics and Machine Learning, Forschungszentrum Jülich, Jülich, Germany

KEYWORDS

**computer vision, plant phenotyping, deep learning, plant disease, datasets**

Editorial on the Research Topic
Computer vision in plant phenotyping and agriculture

## 1. Introduction

Plant phenotyping is the process of identifying a plant's structural and functional characteristics. Plant phenotyping is used by plant scientists to uncover mechanisms of plant physiology, e.g., to characterize how plants respond to biotic and abiotic stress. Phenotyping is also used by plant breeders to evaluate cultivars in a plant population for beneficial characteristics in order to inform the selection of progeny to move forward within a multi-year breeding process. In an attempt to reduce the time and cost required to phenotype large plant populations, image-based phenotyping has become popular over the past 10 years. Extracting phenotypic information from images of plants and crops presents a number of challenging real-world computer vision problems, such as analyzing images with highly self-similar repeating patterns and analyzing densely packed and occluded plant organs.

This Research Topic is associated with the 7th Computer Vision in Plant Phenotyping and Agriculture (CVPPA) workshop, which was held at the International Conference on Computer Vision (ICCV) on 11 October 2021. During the workshop, 18 full-length papers and 14 extended abstracts were presented. This Research Topic includes three papers that are extended versions of abstracts presented at the workshop. The Research Topic also includes 11 new articles that fall under the general scope of CVPPA but were not previously presented at the workshop.

The papers in this Research Topic explore a number of high-priority challenges in image-based phenotyping, including curating new datasets, developing few- and zero-shot analysis approaches that do not require extensive labeled datasets, handling occlusion in plant images, and visualizing and selecting appropriate models for plant phenotyping tasks. The collection of papers largely focuses on three areas: (1) general plant phenotyping tasks, such as plant species classification; (2) detection and classification of plant disease symptoms; and (3) detection of other plant pests including insects and weeds. Overall, this collection of high-quality papers has accomplished the goals of the Research Topic, which demonstrated the state-of-the-art research in image-based plant phenotyping, identified key unsolved problems, and introduced computer scientists to the field of plant phenotyping.

## 2. Papers

### 2.1. Plant phenotyping

A number of papers within this Research Topic investigated novel deep-learning and computer-vision techniques to tackle general plant phenotyping tasks and related technical challenges associated with collecting crop images. Fujiwara et al. reported a comparison if different approaches to estimate plant height from UAV images of outdoor maize fields. Liu K.-H. et al. proposed an efficient convolutional neural network (CNN) architecture for plant species classification from hyperspectral images. Zhang et al. combined field images and genotypic information for a population of sorghum cultivars toward elucidating genotype-by-phenotype interactions. With an image dataset of isolated Chrysanthemum flowers, Wang et al. investigated cultivar classification in a plant population with large morphological variations. Moving down in scale to images of individual seeds, Fonseca de Oliveira et al. reported on phenotyping of peanut seed quality. The final two papers reported novel deep learning approaches to plant phenotyping. Mostafa et al. proposed a new metric, the SSIM cut curve, for model selection in plant species classification. Kierdorf et al. used deep generative adversarial networks to reveal the likely plant organs that are hidden behind leaves in images of grapevines.

### 2.2. Disease detection

Many papers in this Research Topic explored approaches for detecting and recognizing disease symptoms from images of plants and plant organs. This matches a trend of increased interest in plant pathology in image-based plant phenotyping research and highlights the importance of biotic and abiotic stress phenotyping in modern crop breeding and farming operations. Papers in the collection have proposed new deep learning approaches to detect diseases in images of strawberries (Liao et al.), grapes (Suo et al.), maize (Qian et al.), rubber trees (Zeng et al.), and citrus trees (Yang et al.). Bruno et al. investigated adaptive minimal ensembling to achieve state-of-the-art performance on the well-studied PlantVillage leaf disease dataset. Finally, Egusquiza et al. proposed a metric learning approach to extract features from a small number of sample images. They demonstrated that the learned features have better discriminative and clustering properties as compared to a traditional supervised learning approach using a novel challenging leaf disease dataset.

### 2.3. Pest detection

A few papers in this Research Topic analyzed plant images to identify and count crop pests, including insects and weeds. Liu B. et al. proposed a new dataset of images of a wide range of forestry pests. Dai et al. also introduced a new pest image dataset but specialized for the Citrus psyllid pest, which is associated with the huanglongbing disease that is affecting citrus production worldwide. The authors reported a novel CNN approach to detect the tiny Citrus psyllid insects from citrus leaf images. Finally, Sapkota et al. evaluated the accuracy of transferring CNN models trained for detecting weeds in cotton crops to similar environments, but with soybean and maize crops. The adaptation and generalization of image-based plant phenotyping approaches to novel domains, such as different crop species or different environmental conditions, remains important challenges for the field.

## 3. Conclusion

To conclude, this Research Topic on Computer Vision in Plant Phenotyping and Agriculture has assembled a collection of papers that showcase a range of computer vision approaches and application domains. The authors would like to thank all the authors for their contributions to the Research Topic, and look forward to future research activity through the CVPPA workshop series.

## Author contributions

IS wrote the first draft of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

## Acknowledgments

The authors would like to thank the CVPPA 2021 workshop organizing team, including Amy Tabb, Andrew French, Michael Pound, Sotirios Tsaftaris, Tony Pridmore, and the reviewers for their expert advice and constructive feedback on the articles published in this Research Topic.

## 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.

frontiers in Plant Science

ORIGINAL RESEARCH
published: 23 December 2021
doi: 10.3389/fpls.2021.809506

# Citrus Huanglongbing Detection Based on Multi-Modal Feature Fusion Learning

*Dongzi Yang*[1,2], *Fengcheng Wang*[1,2], *Yuqi Hu*[1,2], *Yubin Lan*[1,2,3,4]* *and Xiaoling Deng*[1,2,3,4]*

*[1] College of Electronic Engineering, College of Artificial Intelligence, South China Agricultural University, Guangzhou, China, [2] National Center for International Collaboration Research on Precision Agricultural Aviation Pesticide Spraying Technology, Guangzhou, China, [3] Guangdong Laboratory for Lingnan Modern Agriculture, Guangzhou, China, [4] Guangdong Engineering Technology Research Center of Smart Agriculture, Guangzhou, China*

Citrus Huanglongbing (HLB), also named citrus greening disease, occurs worldwide and is known as a citrus cancer without an effective treatment. The symptoms of HLB are similar to those of nutritional deficiency or other disease. The methods based on single-source information, such as RGB images or hyperspectral data, are not able to achieve great detection performance. In this study, a multi-modal feature fusion network, combining a RGB image network and hyperspectral band extraction network, was proposed to recognize HLB from four categories (HLB, suspected HLB, Zn-deficient, and healthy). Three contributions including a dimension-reduction scheme for hyperspectral data based on a soft attention mechanism, a feature fusion proposal based on a bilinear fusion method, and auxiliary classifiers to extract more useful information are introduced in this manuscript. The multi-modal feature fusion network can effectively classify the above four types of citrus leaves and is better than single-modal classifiers. In experiments, the highest accuracy of multi-modal network recognition was 97.89% when the amount of data was not very abundant (1,325 images of the four aforementioned types and 1,325 pieces of hyperspectral data), while the single-modal network with RGB images only achieved 87.98% recognition and the single-modal network using hyperspectral information only 89%. Results show that the proposed multi-modal network implementing the concept of multi-source information fusion provides a better way to detect citrus HLB and citrus deficiency.

**Keywords: convolutional neural network, citrus greening disease, machine learning, multi-modal feature fusion, hyperspectral images**

OPEN ACCESS

***Edited by:***
*Hanno Scharr,
Jülich Research Center, Helmholtz Association of German Research Centres (HZ), Germany*

***Reviewed by:***
*Dimitrios Fanourakis,
Technological Educational Institute of Crete, Greece
Yongqiang Zheng,
Southwest University, China*

***Correspondence:**
*Yubin Lan
ylan@scau.edu.cn
Xiaoling Deng
dengxl@scau.edu.cn*

***Specialty section:***
*This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science*

***Received:*** *05 November 2021*
***Accepted:*** *06 December 2021*
***Published:*** *23 December 2021*

***Citation:***
*Yang D, Wang F, Hu Y, Lan Y and Deng X (2021) Citrus Huanglongbing Detection Based on Multi-Modal Feature Fusion Learning. Front. Plant Sci. 12:809506. doi: 10.3389/fpls.2021.809506*

## INTRODUCTION

Citrus Huanglongbing (HLB), also called citrus greening, is commonly believed to be citrus cancer without effective treatment. The symptoms of HLB are mainly yellow shoots, yellow leaves, and red nose fruits, among others. The infected plants easily wither and die. HLB is found all over the World, and it also occurs in China, especially in the Guangdong Sihui and Guangdong Huizhou. HLB is infectious and can be spread through insect vectors or grafting. The three most effective

**Abbreviations:** HLB, Citrus Huanglongbing; PCA, principal component analysis; PCR, polymerase chain reaction; UVA, unmanned aerial drone.

methods to prevent HLB are planting non-toxic seedlings, preventing and controlling citrus psyllids, and removing diseased plants (Han et al., 2021). In traditional agriculture, the prevention and control of HLB relies on the observation of experts or experienced farmers to remove diseased plants as early as possible. For plants with mild symptoms, PCR (Polymerase Chain Reaction), and other biotechnological techniques can be used to accurately identify plants. This method has high accuracy and disease can be detected and eradicated in the early stages of plant infection. However, this approach relies on experts first identifying diseased plants, and then bringing the diseased plants back to the laboratory to have disease confirmed by genetic methods. This process is lengthy and dependent on those experts. If a machine is trained as an expert and replaces the expert for identification, the detection process will be significantly accelerated.

With the development of deep learning since 2015, many useful networks for special object extraction have emerged, such as CNNs,ResNet50 (He et al., 2016), VGG16 (Simonyan and Zisserman, 2014), GoogleNet (Szegedy et al., 2015), SeNet50 (Hu et al., 2020), ResNeXt101 (Szegedy et al., 2015), VGG (Simonyan and Zisserman, 2014), and Senet50 (Hu et al., 2020). They have been very successful in modeling complicated systems, owing to their ability of distinguishing patterns and extracting regularities from data. The above-mentioned networks have been effectively incorporated in plant phenotyping projects. For example, variety identification in seeds (Taheri-Garavand et al., 2021b; Plants 10, 1406) and in intact plants by using leaves (Nasiri et al., 2021; Plants 10, 1628), weed and crop classification and recognition is the frontier and trend of agricultural artificial intelligence (Deng et al., 2020; Jiang et al., 2020), detecting crop nutritional deficiencies (Baresel et al., 2017; Tao et al., 2020), and plant disease classification (Kaya et al., 2019; Karlekar and Seal, 2020). Mostly, studies learn single-source information, and classify or identify subsequent information. These kinds of networks mostly use visual image and have rather good accuracy in specific cases. However, agriculture is a complicated system in which the shooting conditions of visual images randomly change and the crops keep growing, which leads the networks reliant on visual imaging to lack universality. Several researchers have made some efforts to improve the accuracy by continuously supplementing datasets (Picon et al., 2019), yet data collecting is a very tough work in agriculture as it is restricted by the environment and the growth cycle of plants. Therefore, how to improve the precision rate under unabundant dataset is becoming increasingly more significant.

In recent years, with the rapid development of spectroscopy, some studies adopted multispectral and hyperspectral information to detect deeper information of objects, such as using infrared to evaluate the quality of strawberry by hyperspectral images (Su et al., 2021), using hyperspectral satellite remote sensing to estimate grassland yield (Ali et al., 2014), or using UVA-based hyperspectral imagery (Feng et al., 2020) for yield prediction. Compared with RGB images, hyperspectral images combined with neural network technology can more effectively identify plant diseases, even in the early stage of disease.

The internal information extracted from hyperspectral images can be used to compensate for the shortcomings of RGB images with only surface information. Hence, multi-source feature fusion can improve the predictive ability of the model. The purpose of the fusion model is to combine the strengths of different sub-models to compensate for any shortcomings (Zadeh et al., 2017). Deep multi-modal learning can reduce the design requirements for feature engineering and deep-learning architectures, and can achieve the required accuracy more simply and quickly (Atrey et al., 2010; Ramachandram and Taylor, 2017; Baltrusaitis et al., 2019). Yan et al. (2021) proposed a fusion scheme combining a multi-dimensional convolutional neural network with a visualization method for detection of aphis gossypii glover infection in cotton leaves using hyperspectral imaging, which achieved good development prospects in plant disease identification.

Numerous researchers have conducted laboratory investigations into the identification of HLB using different methods under different observation heights, such as using visual images in the laboratory with traditional machine-learning methods (Deng et al., 2016) and using UAV hyperspectral and multispectral images using deep-learning networks (Deng et al., 2019; Lan et al., 2020).

To increase the reliability and precision of HLB detection, in this study, a method is proposed that fuses two sources of information, namely, spectral and RGB images, by building a multi-modal deep-learning network to identify HLB leaves from four categories.

# MATERIALS AND METHODS

## Data Acquisition and Processing

The data used in this study were collected in the citrus test fruit field of South China Agricultural University, Tianhe, Guangdong Province (longitude 113.35875, latitude 23.15747). In early March, citrus trees are in the spring growth period and are grown in subtropical climate regions, shown in **Figure 1**. The variety of citrus is Shatangju (*Citrus reticulata Banco*). The selected tree samples were specially cultivated and PCR-tested, and Zinc deficiency was visually assessed by a field expert, and was confirmed by conducting mineral analysis. The data samples of this study include the leaves of HLB plants, of Zn-deficient plants, of healthy plants, and those with suspected HLB (in which case the surface of the leaf is uniformly yellow, which is different from the obvious symptoms of HLB). The four categories leaves are shown in **Figure 2**.

The collection environment is shown in **Figure 3**. The RGB images were taken with Sony cameras and under natural light, ensuring that the required foliage was clear, independent of the shooting location, and free of background interference. The distance between camera and leaves was controlled with 20–50 cm. The hyperspectral data of leaves were collected by a hyperspectral imager (Hypersis-VNIR-PFH, Zhuoli Hanguang, Beijing, China). The spectral range was 300 nm to 1070 nm and the exposure time for each collection was 30 ms. The running speed of the mobile platform was 5.0375 mm/s, the scanning

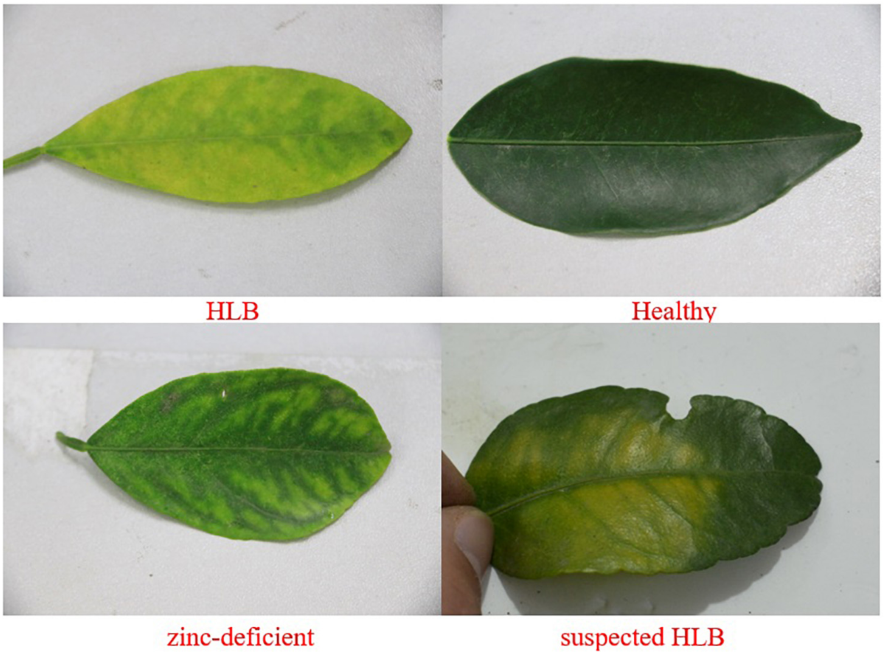
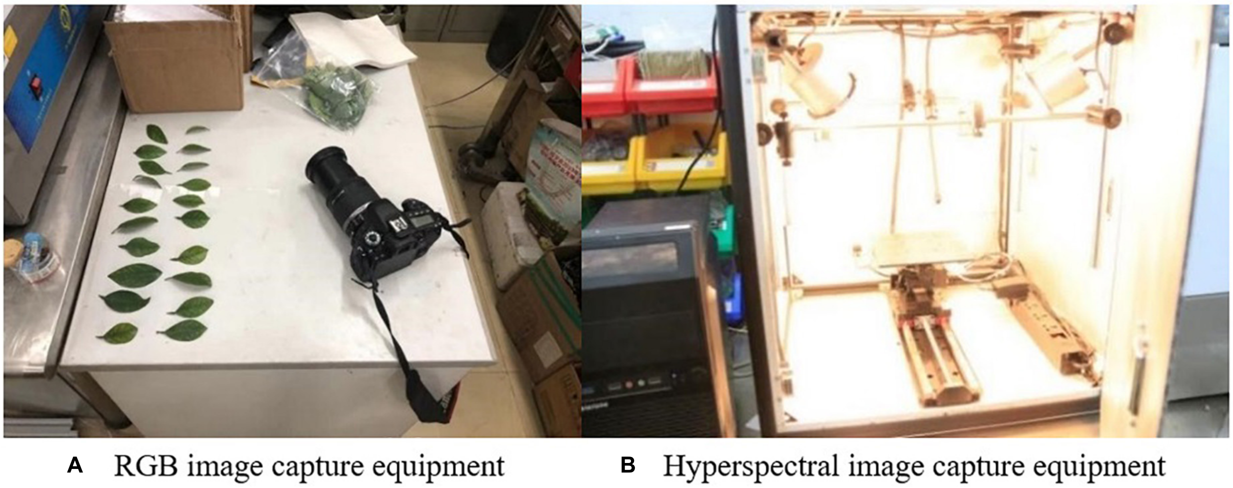


**FIGURE 1 |** Dataset collection location.

**FIGURE 2 |** Four categories leaves.

distance 120 mm, and the hyperspectral image size 100 × 200 pixels. Spectral data analysis and processing were implemented in ENVI 5.3 software (Harris Geospatial Solutions, Inc., Broomfield, CO, United States).

**Figure 4** shows the method of feature area selection during the data processing step. In the process of hyperspectral image analysis, the upper, middle, and lower regions of interest of the leaf blade were chosen as the feature region, the average reflectance in the region of interest calculated, and the average reflectance used to represent the area. Finally, the hyperspectral image was converted into a hyperspectral band, and the average reflectance used to reflect the area. The frequency band of each area ranged from 300 nm to 1070 nm, removing the incomplete information about the start and the tail, leaving 768 bands in the middle. Owing to the similarity of adjacent bands of hyperspectral images, to reduce similar repetitive features, every three adjacent bands in the 300–1070-nm range were extracted and combined into a new band. After final extraction, 256 composite bands remained.



**A** RGB image capture equipment

**B** Hyperspectral image capture equipment

**FIGURE 3 |** Data collection equipment. **(A)** RGB image capture equipment. **(B)** Hyperspectral image capture equipment.

**Table 1** shows the one-to-one correspondence dataset between images and spectral data. Each RGB image corresponds to a spectral sample and each piece of spectral data contains the spectral information of the upper, middle, and lower regions of the leaf.

## Multi-Modal Network Architecture

The multi-modal network proposed in this study consists of two backbone networks. The architecture was divided into four parts. The first is an image feature extraction network that extracts surface features of RGB images. The second is a hyperspectral band feature extraction network that extracts the HLB feature bands. The third is a feature fusion part that fuses the two features extracted from two different networks and performs classification with an auxiliary classifier. The fourth part is classification using auxiliary classifiers. The multi-modal network structure is shown in **Figure 5**.

## RGB Image Extraction Network

In the first part of RGB image feature extraction, ResNet50, VGG16, and ResNeXt101 were selected as the candidates for the backbone network. After experimental comparison, ResNet50 was adopted because it works well and in wide use. In terms of the network structure, ResNet50 has fewer parameters, but the effect achieved is similar to that of ResNeXt101. The image in this experiment is high definition, and the amount of calculation required for the extraction of the hyperspectral band is also large. To reduce the amount of calculation and not lose too much accuracy, ResNet50 was chosen. The results of the experiment are detailed further below at **Table 2**. To enrich the diversity of samples, a data enhancement module was added to the network. During the training process, there was a 10% probability that the RGB image would be randomly rotated forward or counterclockwise by 45°. The feature dimensionality extracted from the backbone network was 2048. To reduce the dimensionality obtained by feature fusion and reduce the amount of network calculation, the fully connected layer was used for feature dimensionality reduction, and the final image feature dimensionality obtained was 256.

## Feature Extraction Network for Hyperspectral Band

The second part of the multi-modal network is to extract feature band information of hyperspectral data. There are many common spectral feature band extraction methods, such as support vector machines and PCA (Principal Component Analysis), among others (Velasco-Forero and Angulo, 2013; Deng et al., 2014; Medjahed et al., 2015; Pérez et al., 2016). In this study, a simple neural network for feature extraction among the 300–1070-nm hyperspectral data is proposed, and an attention module was added in this hyperspectral feature band exaction network to increase the ability of extracting bands. After combining the three adjacent bands into one channel, the number of bands decreased from 758 to 256, which reduced the overall amount of calculation and number of parameters of the hyperspectral feature extraction network. Hyperspectral band information is one-dimensional (1D) information. Commonly used image neural networks are not suitable for 1D information extraction, and we only needed to extract the bands with large differences. Therefore, the designed neural network must be capable of 1D information extraction. Moreover, it must be able to find the bands with large differences and retain the characteristic of this large difference. As shown in **Figure 4**, the upper, middle, and lower parts of each hyperspectral image were selected, and the hyperspectral band of each hyperspectral image calculated by averaging each part of the sample. Thus, there were three pieces of hyperspectral 1D data for each channel of the hyperspectral image. Therefore, the input of the hyperspectral

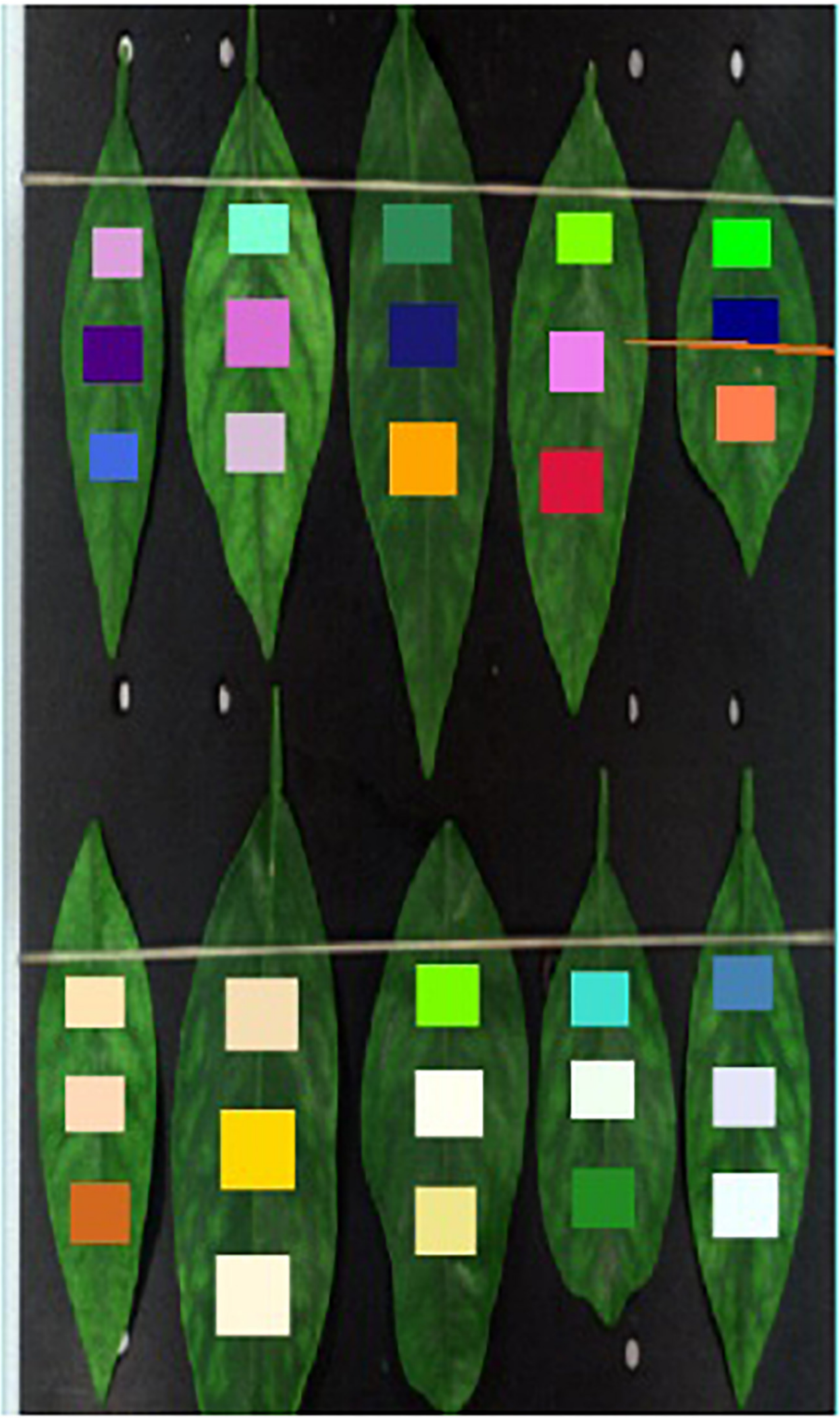


**FIGURE 4 |** Feature area selection during processing in Envi software.

band feature extraction network was 256 × 3. Even so, a significant amount of redundant information remains. To reduce the influence of this redundant information on the final classification results, a soft attention mechanism was adopted in the module to further extract the hyperspectral information of input data. Finally, the output size of the network was 1 × 256. The structure of the attention algorithm is shown in **Figure 6**.

## Multi-Modal Feature Fusion

Typical fusion methods mainly comprise early and late fusion. As the name suggests, early fusion is used to fuse features at feature levels, using operations such as concatenation and addition of different features (Chaib et al., 2017), and then inputting the fused features into a model for training. Late fusion refers to fusion on the score level. Methods such as a feature pyramid network (Pan et al., 2019) train multiple models, and each model will have a prediction score. The results of all models are fused to obtain the final prediction results. In this study, the 1D hyperspectral band information and 3D RGB picture information were fused before detection. ResNet50 and a hyperspectral band feature extraction network (spectrum) were used in the present work as the fusion network to carry out three different feature fusions, all of which are examples of early fusion. These three methods are feature addition, feature multiplication, and feature bilinear fusion. From **Figure 7** shows that the accuracy of addition is 94.58%, that of multiplication is 93.85%, and that of bilinear fusion is 95.1%. It can also be seen from **Figure 7** that the fitting speed of bilinear fusion was also faster than that of the other two methods.

The bilinear fusion method (Yu et al., 2018) was adopted to fuse the features between different networks. The original bilinear fusion is shown in Eqs. (1) and (2). The two input modes are **X** and **Y**, and the bilinear fusion can thus be expressed as:

$$\boldsymbol{Z_i} = \boldsymbol{X^T W_i Y}, \tag{1}$$

where, **W** is the projection matrix and **Z** the output of the bilinear model. **W** is decomposed into two low-rank **U** and **V** matrices, with ° indicating a matrix dot product:

$$\boldsymbol{Z_i} = \boldsymbol{X^T U_i V_i^T Y} = \boldsymbol{U_i^T X^\circ V_I^T Y}. \tag{2}$$

The specific fusion formula is shown in Equation (3), where Z $(Features_{Mix})$ represents the fusion features, I $(Features_{Image})$ the features extracted by the image network, and B $(Features_{Spectrum})$ the features extracted by the spectral network. A is an N × N matrix and Bias an N × 1 matrix; in the experiments detailed herein, $N = 256$.

$$\boldsymbol{Z\ (Features_{mix})} = \boldsymbol{I\ (Features_{Image})\ A\ B\ (Features_{band})\ +Bias} \tag{3}$$

## Auxiliary Classifier

After feature fusion, the samples were modeled using auxiliary classifier based on the fused feature values. The final classification effect of the network is affected by the two backbone feature extraction networks. To improve the feature extraction effects of the RGB image feature extraction network and hyperspectral band feature extraction network, the auxiliary classifiers were modified as shown in **Figure 8**, where the loss of the overall network consists of the loss of the fusion feature classifier and one of each backbone network classifier. The specific loss calculation formula is as in Equation (2), where *Total Loss* represents the overall loss value of the network, $\boldsymbol{Loss_{mix}}$ the loss value of the fusion feature classifier, $\boldsymbol{Aux\ Loss_1}$ the loss value of the image auxiliary classifier, and $\boldsymbol{Aux\ Loss_2}$ the spectral auxiliary classifier The loss values of $\boldsymbol{\mu_1}$ and $\boldsymbol{\mu_2}$ are the auxiliary classifier loss

**TABLE 1 |** Four different types of data and amounts of each.

| Species | Number of images | Number of spectral images |
|---|---|---|
| Healthy | 300 | 300 |
| HLB | 375 | 375 |
| Zn-deficient | 350 | 350 |
| HLB suspected | 300 | 300 |

*HLB, Citrus Huanglongbing.*

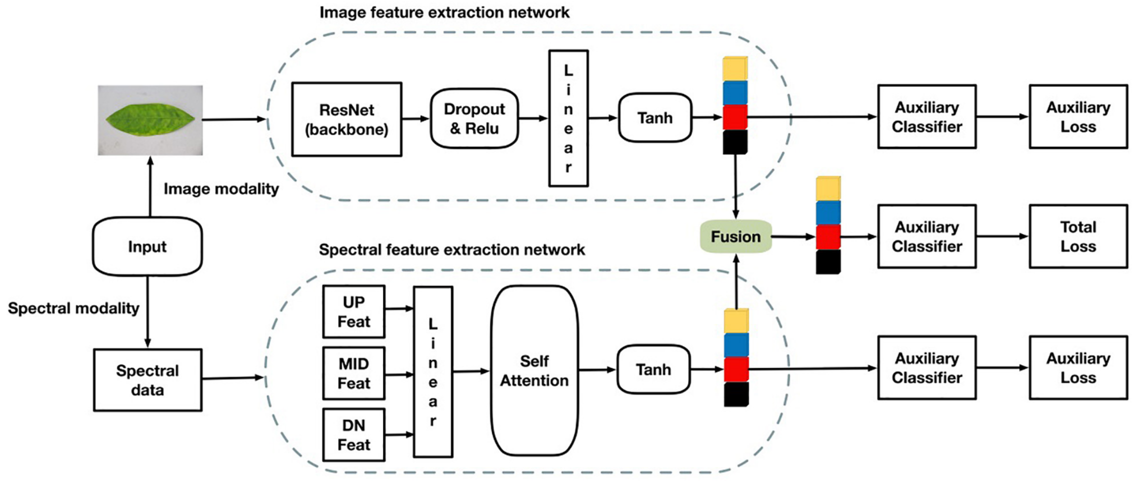


**FIGURE 5 |** Multi-modal network structure.

**TABLE 2 |** Single-network classification and multi-modal network classification accuracy.

| Sample | Model | Accuracy (%) |
|---|---|---|
| RGB image<br>hyperspectral data | ResNet50 | 85 |
| | VGG16 | 84.51 |
| | ResNeXt101 | 87.98 |
| | Hyperspectral feature extraction network | 89 |
| RGB image +<br>hyperspectral data | Multi-modal network M1 | 96 |
| | Multi-modal network M2 | 95.1 |
| | Multi-modal network M3 | 97.89 |

**M1, ResNet50+hyperspectral feature extraction network; M2, VGG16+hyperspectral feature extraction network; M3, ResNeXt101+hyperspectral feature extraction network.*

weight coefficients ($0 \le \mu_1 < 1, 0 \le \mu_2 < 1$). By testing different groups of weight coefficient values, it was found that the best classification effect is obtained when the coefficient $\mu_1 = 0.25$ and the coefficient $\mu_2 = 0.20$.

$$\textbf{\textit{Total Loss}} = \textbf{\textit{Loss}}_{mix} + \mu_1 \times \textbf{\textit{Aux Loss}}_1 + \mu_2 \times \textbf{\textit{Aux Loss}}_2 \quad (4)$$

# RESULTS

The experimental hardware environment of this study is listed in **Table 3**. The software environment was set as the following: python, Ubuntu 16.04, CUDA, CUDNN, and OpenCV. In this study, **F1** score and accuracy were used to evaluate the trained model. The formulas are given in Equations (3)–(6), where ***P*** is the precision rate, ***R*** the recall rate, ***TP*** the number of true positive samples, ***FP*** the number of false positive samples, ***FN*** the number of false negative samples, ***true_num*** the number of samples that are classified correctly, and ***total_num*** the total number of tests and total number of samples.

$$P = P = \frac{TP}{TP + FP} \quad (5)$$

$$R = \frac{TP}{TP + FN} \quad (6)$$

$$F1 = \frac{2P * R}{P + R} \quad (7)$$

$$Accuracy = \frac{true\_num}{total\_num} \quad (8)$$

## Experimental Results

The experimental comparison results between single-network and multi-modal network classification are shown in **Table 2**. The recognition accuracies of the single network using RGB images were 85, 84.51, and 87.98% based on ResNet50, VGG16, and ResNeXt101, respectively. The recognition accuracy of the hyperspectral data dimensionality reduction network based on the soft attention mechanism was 89%, while that of the multi-modal networks designated M1 (ResNet50+hyperspectral feature extraction network), that designated M2 (VGG16+hyperspectral feature extraction network), and that designated M3 (ResNext101 +hyperspectral feature extraction network) reached 96, 95.1, and 98% respectively, all significantly higher than that of a single network. Compared with the F1 score of 85% using only the image network and that of 89% using only the spectral network, increases of 13 and 9%, respectively, were found. It can be clearly seen that feature fusion based on the bilinear fusion method and the multi-modal network of the auxiliary classifier can extract more useful information, and can better classify items with similar features.

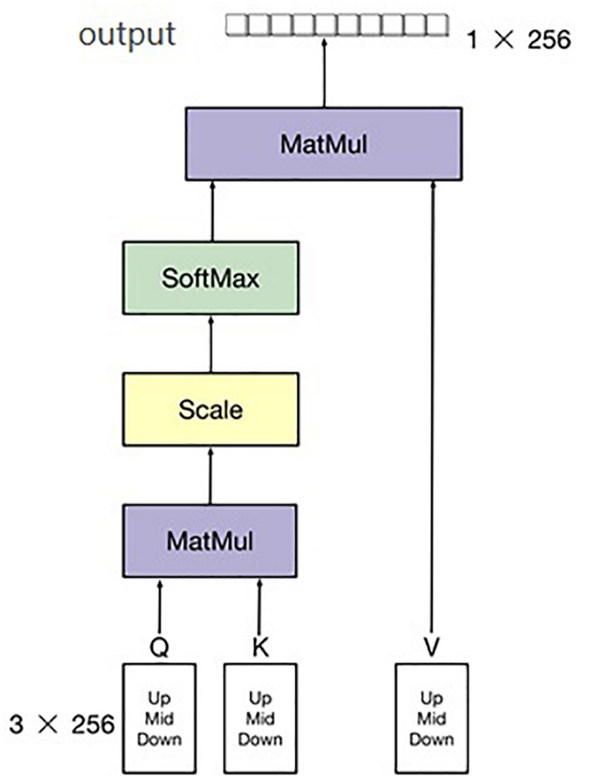
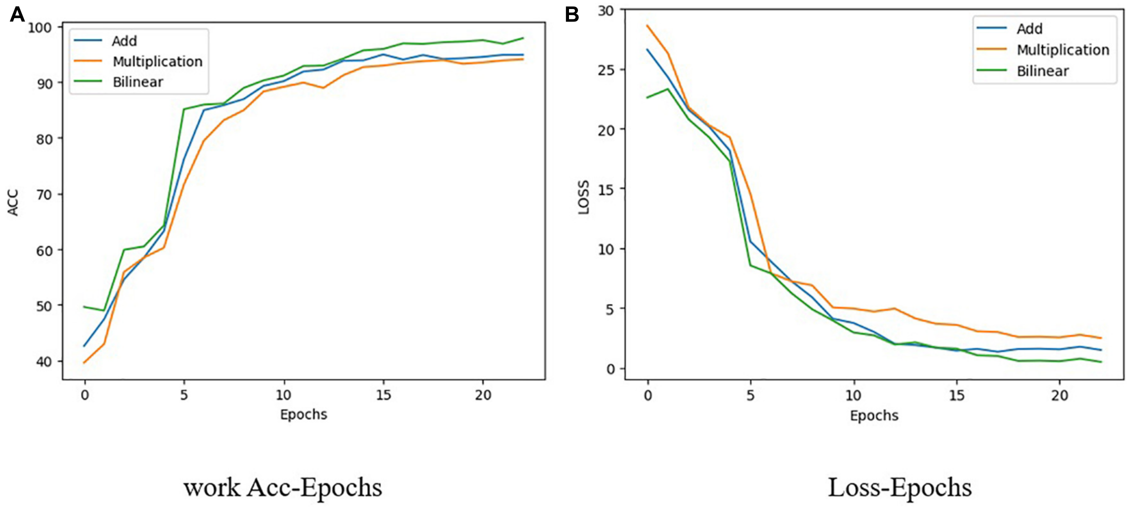


**FIGURE 6 |** Hyperspectral band feature extraction network.

To verify the performance of the multi-modal networks, ResNet50 with medium recognition accuracy (**Table 2**) was selected as the basic network to better reflect the improvement of recognition accuracy of multi-modal networks. **Table 4** shows the detection performance of each category based on multi-modal network M1, where the F1 scores of HLB, healthy, Zn-deficient, and suspected HLB-diseased leaves reached 95, 98, 96, and 94%, respectively, showing that average recognition accuracy reached over 95%.

## Visualization Analysis of Models

**Figure 9** shows the change of loss and accuracy with epoch during the training process of each network. It can be seen



work Acc-Epochs

Loss-Epochs

**FIGURE 7 |** Change with epoch of loss and accuracy of three feature fusion methods used in present. **(A)** Work Acc-Epochs. **(B)** Loss-Epochs.

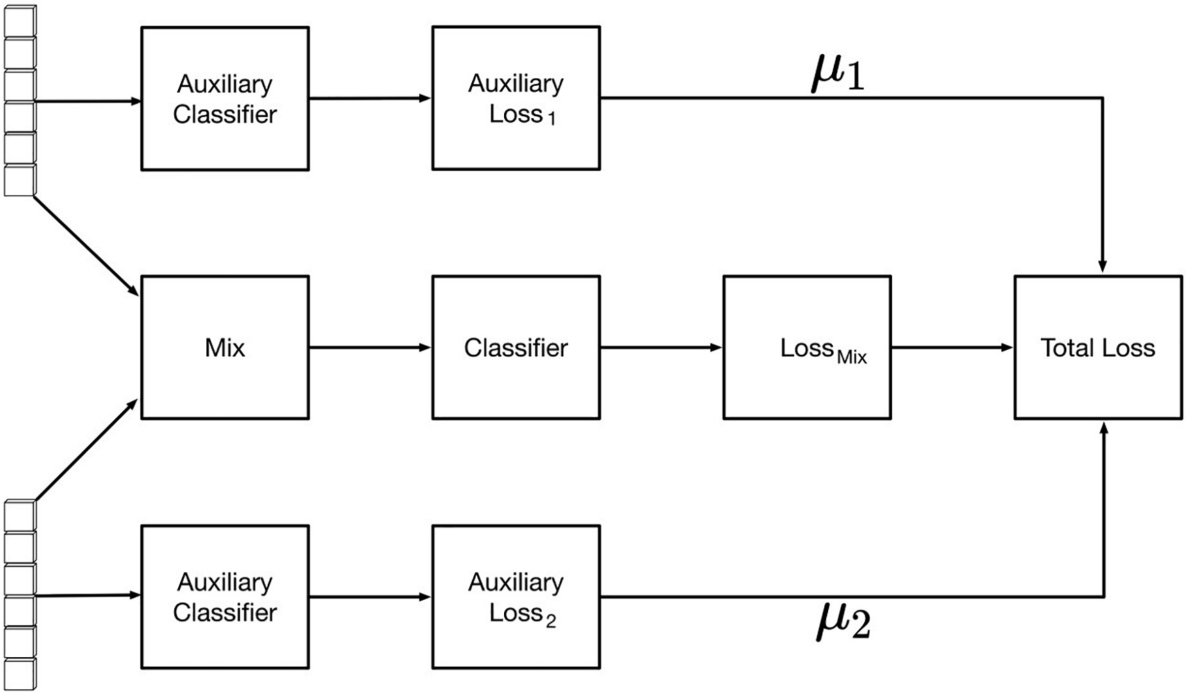


**FIGURE 8 |** Loss calculation method based on auxiliary and mixture loss.

from **Figure 9** that with increasing epoch loss, the fitting effect of the multi-modal model is obviously better than that of the RGB image network, and both tend to stabilize after 20 epochs. Compared with single-modal networks, including the spectrum network and RGB image networks using VGG16, ResNet50, and ResNeXt101, the three multi-modal networks achieved significantly better performance with faster convergence (as shown in **Figure 9A**) and higher accuracy (as shown in **Figure 9B**).

**Figure 10** shows the confusion matrix of the three models. **Figures 10B,C** is the confusion matrix of the RGB image network and the hyperspectral network. It can be seen that the classification effects of the RGB image network and the hyperspectral image network have complementary aspects, especially for zinc deficiency. Classification of symptoms and HLB symptoms. **Figure 10A** is the effect of the final multi-modal network. It can be seen that the final confusion matrix has achieved a good effect.

## DISCUSSION

Most existing networks can significantly improve the recognition accuracy by increasing the depth of the network, the dimensionality of the network, and the size of the data set. Such as ResNet, from ResNet50 to ResNet101, its recognition accuracy is improved, but the recognition speed and calculation amount are increased. When more than 101 layers are added, the recognition accuracy is not improved. This shows that although only increasing network depth can increase the accuracy, the cost is too high. The GoogleNet is to increase the width of each layer without increasing the depth of the network, but this improvement is also limited. Besides, dataset is difficult work in agriculture as it is restricted by the environment and the growth cycle of plants. Multi-modal networks can expand the data dimension through network fusion and fusion of features extracted from different data. Under the condition of insufficient data for a deep-learning network, it is relatively simple to combine other sources of information to improve the accuracy of the network from a horizontal perspective rather than a vertical perspective.

**TABLE 3 |** Experimental environment.

| Hardware | Brand | Number |
|---|---|---|
| CPU | I7–10700 | 1 |
| Storage | Kingston, 16 GB | 2 |
| Graphics card | GeForce GTX3070 | 1 |
| Hard disk | West Statistics, 1 TB | 1 |
| Main board | Dell Precision 3640 tower | 1 |

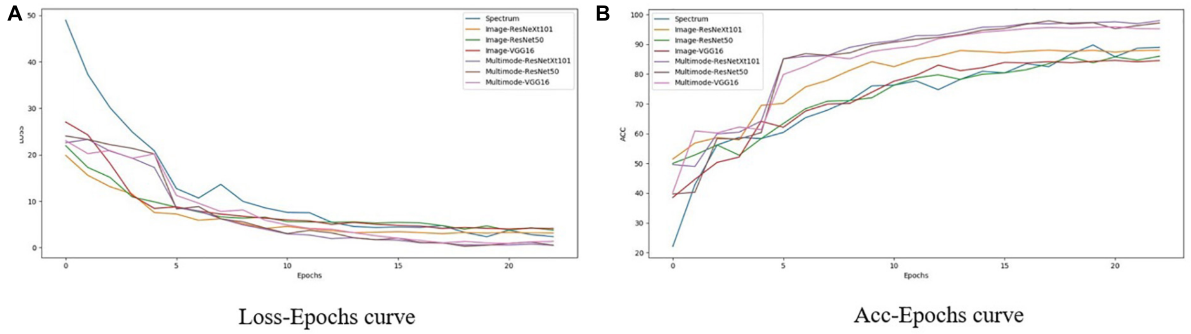
**TABLE 4 |** Four classification results of multi-modal network M1.

| Type | Precision (%) | Recall (%) | F1 score (%) |
|---|---|---|---|
| HLB | 96 | 94 | 95 |
| Health | 98 | 99 | 98 |
| Zn-deficient | 97 | 94 | 96 |
| HLB suspected | 92 | 96 | 94 |

**M1, ResNet50+ hyperspectral feature extraction network. HLB, Citrus Huanglongbing.*

In the present study, the four testing categories discussed have similar symptoms, and are difficult to discriminate only by visual imaging. Hyperspectral data can reflect the internal information of plants to a certain extent, such as chlorophyll or element content, and can make up for the lack of RGB imagery and solve the discrimination problem resulting from the similar appearance of leaves.

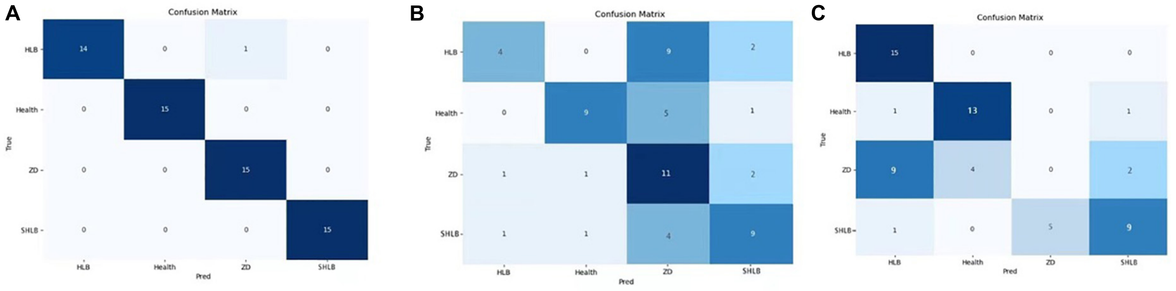
Regarding the multi-feature fusion part, fusion weight coefficients were introduced to the weigh the output result,



Loss-Epochs curve

Acc-Epochs curve

**FIGURE 9 |** Change with epoch of loss and accuracy of different networks in training process. **(A)** Loss-Epochs curve. **(B)** Acc-Epochs curve.



Multi-modal network

RGB image network

Hyperspectral band network

**FIGURE 10 |** Confusion matrix of the three models. **(A)** Multi-modal network. **(B)** RGB image network. **(C)** Hyperspectral band network.

thus improving the fitting effect of the proposed multi-modal network. The image recognition accuracy of the multi-modal model can be even improved by adding more dimensional information or improving the performance of the constituent network. The proposed method can also be applied to other agricultural applications, such as pest and disease detection with similar symptoms or appearances.

On a commercial scale, evidently, a capital investment is initially required for adopting the employed approach (Taheri-Garavand et al., 2021a Industrial Crop Prod 171, 113985). Nevertheless, the wide-ranging large-scale commercial applications can provide high returns through considerable improvements in process enhancement and cost reduction. Spectroscopy is a high-cost and high-tech imaging device, and its application areas are still being developed. However, through the research in this article, it can further expand its application fields and improve its technology. Through the neural network fusion method and the combination of RGB images, the recognition and classification of agricultural pests or agricultural diseases are enhanced.

## CONCLUSION

A multi-modal network for citrus HLB detection and a bilinear fusion method based on RGB images and hyperspectral information are proposed in this study. Four HLB types with similar symptoms of leaves (HLB, suspected HLB, Zn-deficient, and healthy) were tested experimentally to verify the effectiveness of the multi-modal network. Results show that the F1-score of HLB detection based on multi-modal network reached 95%, that of healthy leaves reached 98%, that of Zn-deficient leaves reached 96 %, and that of suspected HLB diseased leaves reached 94%. The image recognition accuracy of the multi-modal model can effectively improve the recognition accuracy of the model when the size of the dataset is limited.

## DATA AVAILABILITY STATEMENT

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

## AUTHOR CONTRIBUTIONS

DY conceptualized the experiment, selected the algorithms, collected and analyzed the data, and wrote the manuscript. FW and YH trained the algorithms, collected and analyzed data, and wrote the manuscript. XD and YL supervised the project. All authors discussed and revised the manuscript.

## FUNDING

This work was supported by Key-Area Research and Development Program of Guangdong Province (Grant No. 2019B020214003), Key-Area Research and Development Program of Guangzhou (Grant No. 202103000090), Key-Areas of Artificial Intelligence in General Colleges and Universities of Guangdong Province (Grant No. 2019KZDZX1012), Laboratory of Lingnan Modern Agriculture Project (Grant No. NT2021009), National Natural Science Foundation of China (Grant No. 61675003), National Natural Science Foundation of China (Grant No. 61906074), and Guangdong Basic and Applied Basic Research Foundation (Grant No. 2019A1515011276).

## ACKNOWLEDGMENTS

DY thank FW for supporting and helping, XD and YL for supervision, and YH for revising the article.

## REFERENCES

Ali, I. C. F., Green, S., and Dwyer, N. (2014). "Application of statistical and machine learning models for grassland yield estimation based on a hypertemporal satellite remote sensing time series," in *Proceedings of The 2014 IEEE Geoscience and Remote Sensing Symposium*, Quebec City, QC, 5060–5063.

Atrey, P. K., Hossain, M. A., El Saddik, A., and Kankanhalli, M. S. (2010). Multimodal fusion for multimedia analysis: a survey. *Multimed. Syst.* 16, 345–379. doi: 10.1007/s00530-010-0182-0

Baltrusaitis, T., Ahuja, C., and Morency, L.-P. (2019). Multimodal machine learning: a survey and taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 423–443. doi: 10.1109/tpami.2018.2798607

Baresel, J. P., Rischbeck, P., Hu, Y., Kipp, S., Hu, Y., Barmeier, G., et al. (2017). Use of a digital camera as alternative method for non-destructive detection of the leaf chlorophyll content and the nitrogen nutrition status in wheat. *Comput. Electron. Agric.* 140, 25–33. doi: 10.1016/j.compag.2017.05.032

Chaib, S., Liu, H., Gu, Y., and Yao, H. (2017). Deep feature fusion for VHR remote sensing scene classification. *IEEE Trans. Geosci. Remote Sens.* 55, 4775–4784. doi: 10.1109/tgrs.2017.2700322

Deng, R., Jiang, Y., Tao, M., Huang, X., Bangura, K., Liu, C., et al. (2020). Deep learning-based automatic detection of productive tillers in rice. *Comput. Electron. Agric.* 177:105703. doi: 10.1016/j.compag.2020.105703

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

Deng, X., Lan, Y., Hong, T., and Chen, J. (2016). Citrus greening detection using visible spectrum imaging and C-SVC. *Comput. Electron. Agric.* 130, 177–183. doi: 10.1016/j.compag.2016.09.005

Deng, X.-L., Li, Z., Deng, X.-L., and Hong, T.-S. (2014). Citrus disease recognition based on weighted scalable vocabulary tree. *Precis. Agric.* 15, 321–330. doi: 10.1007/s11119-013-9329-2

Feng, L., Zhang, Z., Ma, Y., Du, Q., Williams, P., Drewry, J., et al. (2020). Alfalfa yield prediction using UAV-based hyperspectral imagery and ensemble learning. *Remote Sens.* 12:2028. doi: 10.3390/rs12122028

Han, H.-Y., Cheng, S.-H., Song, Z.-Y., Ding, F., and Xu, Q. (2021). Citrus huanglongbing drug control strategy. *J. Huazhong Agr. Univ*. 40, 49–57. doi: 10.13300/j.cnki.hnlkxb.2021.01.006

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference Computing Vision Pattern Recognition (CVPR)*, Las Vegas, NV, doi: 10.1109/cvpr.2016.90

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.2019.2913372

Jiang, H., Zhang, C., Qiao, Y., Zhang, Z., Zhang, W., and Song, C. (2020). CNN feature based graph convolutional network for weed and crop recognition in smart farming. *Comput. Electron. Agric.* 174:105450. doi: 10.1016/j.compag.2020.105450

Karlekar, A., and Seal, A. (2020). SoyNet: soybean leaf diseases classification. *Comput. Electron. Agric.* 172:105342. doi: 10.1016/j.compag.2020.105342

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

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

Medjahed, S. A., Saadi, T. A., Benyettou, A., and Ouali, M. (2015). Binary cuckoo search algorithm for band selection in hyperspectral image classification. *IAENG Int. J. Comput. Sci.* 42, 183–191.

Nasiri, A., Taheri-Garavand, A., Fanourakis, D., Zhang, Y., and Nikoloudakis, N. (2021). Automated grapevine cultivar identification via leaf imaging and deep convolutional neural networks: a proof-of-concept study employing primary iranian varieties. *Plants* 10:1628. doi: 10.3390/plants10081628

Pan, H., Chen, G., and Jiang, J. (2019). Adaptively dense feature pyramid network for object detection. *IEEE Access* 7, 81132–81144. doi: 10.1109/access.2019.2922511

Pérez, M. R. V., Mendoza, M. G. G., Elías, M. G. R., González, F. J., Contreras, H. R. N., and Servín, C. C. (2016). Raman spectroscopy an option for the early detection of citrus Huanglongbing. *Appl. Spectrosc.* 70, 829–839. doi: 10.1177/0003702816638229

Picon, A., Seitz, M., Alvarez-Gila, A., Mohnke, P., Ortiz-Barredo, A., and Echazarra, J. (2019). Crop conditional Convolutional Neural Networks for massive multi-crop plant disease classification over cell phone acquired images taken on real field conditions. *Comput. Electron. Agric.* 167:105093. doi: 10.1016/j.compag.2019.105093

Ramachandram, D., and Taylor, G. W. (2017). Deep multimodal learning: a survey on recent advances and trends. *IEEE Signal Process. Mag.* 34, 96–108. doi: 10.1109/msp.2017.2738401

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv* [Preprint] arXiv: 1409.1556, doi: 10.3390/s21082852

Su, Z., Zhang, C., Yan, T., Zhu, J., Zeng, Y., Lu, X., et al. (2021). Application of hyperspectral imaging for maturity and soluble solids content determination of strawberry with deep learning approaches. *Front. Plant Sci.* 12:736334. doi: 10.3389/fpls.2021.736334

Szegedy, C., Wei, L., Yangqing, J., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). “Going deeper with convolutions,” in *Proceedings of the 2015 IEEE Conference Computing Vision Pattern Recognition (CVPR)*, Boston, MA, doi: 10.1109/cvpr.2015.7298594

Taheri-Garavand, A., Nasiri, A., Fanourakis, D., Fatahi, S., and Omid, M. (2021b). Automated in situ seed variety identification via deep learning: a case study in chickpea. *Plants* 10:1406. doi: 10.3390/plants10071406

Taheri-Garavand, A., Mumivand, H., Fanourakis, D., Fatahi, S., and Taghipour, S. (2021a). An artificial neural network approach for non-invasive estimation of essential oil content and composition through considering drying processing factors: a case study in *Mentha aquatica*. *Ind. Crops Prod.* 171:113985. doi: 10.1016/j.indcrop.2021.113985

Tao, M., Ma, X., Huang, X., Liu, C., Deng, R., Liang, K., et al. (2020). Smartphone-based detection of leaf color levels in rice plants. *Comput. Electron. Agric.* 173:105431. doi: 10.1016/j.compag.2020.105431

Velasco-Forero, S., and Angulo, J. (2013). Classification of hyperspectral images by tensor modeling and additive morphological decomposition. *Pattern Recognit.* 46, 566–577. doi: 10.1016/j.patcog.2012.08.011

Yan, T., Xu, W., Lin, J., Duan, L., Gao, P., Zhang, C., et al. (2021). Combining multi-dimensional convolutional neural network (CNN) with visualization method for detection of aphis *Gossypii* glover infection in cotton leaves using hyperspectral imaging. *Front. Plant Sci.* 12:604510. doi: 10.3389/fpls.2021.604510

Yu, C., Zhao, X., Zheng, Q., Zhang, P., and You, X. (2018). “Hierarchical bilinear pooling for fine-grained visual recognition,” in *Proceedings of the European Conference Computing Vision (ECCV)*, Cham, doi: 10.1007/978-3-030-01270-0_35

Zadeh, A., Chen, M., Poria, S., Cambria, E., and Morency, L.-P. (2017). “Tensor fusion network for multimodal sentiment analysis,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, doi: 10.18653/v1/d17-1115

**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.

frontiers in Plant Science

ORIGINAL RESEARCH
published: 24 January 2022
doi: 10.3389/fpls.2021.816272

# Detection Method of Citrus Psyllids With Field High-Definition Camera Based on Improved Cascade Region-Based Convolution Neural Networks

*Fen Dai[1,2,3,4], Fengcheng Wang[1,2], Dongzi Yang[1,2], Shaoming Lin[1,2], Xin Chen[1,2,3,4], Yubin Lan[1,2,3,4]* and Xiaoling Deng[1,2,3,4]**

[1] *College of Electronic Engineering, College of Artificial Intelligence, South China Agricultural University, Guangzhou, China,* [2] *National Center for International Collaboration Research on Precision Agricultural Aviation Pesticide Spraying Technology, Guangzhou, China,* [3] *Guangdong Laboratory for Lingnan Modern Agriculture, Guangzhou, China,* [4] *Guangdong Engineering Technology Research Center of Smart Agriculture, Guangzhou, China*

OPEN ACCESS

**Edited by:**
*Hanno Scharr,*
*Helmholtz Association of German Research Centres (HZ), Germany*

**Reviewed by:**
*Dimitrios Fanourakis,*
*Technological Educational Institute of Crete, Greece*
*Yongqiang Zheng,*
*Southwest University, China*

***Correspondence:**
*Yubin Lan*
*ylan@scau.edu.cn*
*Xiaoling Deng*
*dengxl@scau.edu.cn*

**Specialty section:**
*This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science*

**Received:** *16 November 2021*
**Accepted:** *06 December 2021*
**Published:** *24 January 2022*

**Citation:**
*Dai F, Wang F, Yang D, Lin S, Chen X, Lan Y and Deng X (2022) Detection Method of Citrus Psyllids With Field High-Definition Camera Based on Improved Cascade Region-Based Convolution Neural Networks. Front. Plant Sci. 12:816272. doi: 10.3389/fpls.2021.816272*

Citrus psyllid is the only insect vector of citrus Huanglongbing (HLB), which is the most destructive disease in the citrus industry. There is no effective treatment for HLB, so detecting citrus psyllids as soon as possible is the key prevention measure for citrus HLB. It is time-consuming and laborious to search for citrus psyllids through artificial patrol, which is inconvenient for the management of citrus orchards. With the development of artificial intelligence technology, a computer vision method instead of the artificial patrol can be adopted for orchard management to reduce the cost and time. The citrus psyllid is small in shape and gray in color, similar to the stem, stump, and withered part of the leaves, leading to difficulty for the traditional target detection algorithm to achieve a good recognition effect. In this work, in order to make the model have good generalization ability under outdoor light condition, a high-definition camera to collect data set of citrus psyllids and citrus fruit flies under natural light condition was used, a method to increase the number of small target pests in citrus based on semantic segmentation algorithm was proposed, and the cascade region-based convolution neural networks (R-CNN) (convolutional neural network) algorithm was improved to enhance the recognition effect of small target pests using multiscale training, combining CBAM attention mechanism with high-resolution feature retention network high-resoultion network (HRNet) as feature extraction network, adding sawtooth atrous spatial pyramid pooling (ASPP) structure to fully extract high-resolution features from different scales, and adding feature pyramid networks (FPN) structure for feature fusion at different scales. To mine difficult samples more deeply, an online hard sample mining strategy was adopted in the process of model sampling. The results show that the improved cascade R-CNN algorithm after training has an average recognition accuracy of 88.78% for citrus psyllids. Compared with VGG16, ResNet50, and other common networks, the improved small target recognition algorithm obtains the highest recognition performance. Experimental results also show that the improved cascade R-CNN algorithm not only performs well in citrus psylla identification but also in other small targets such as citrus fruit flies, which makes it possible and feasible to detect small target pests with a field high-definition camera.

**Keywords: citrus psyllids, deep learning, small target detection, cascade R-CNN, small target enhancement**

# INTRODUCTION

The prevention and control of agricultural pests and diseases is a very serious problem in agriculture. Farmers usually need to spray a lot of pesticides to prevent pests and diseases in advance. If the field pests can be detected as early as possible, the pesticides can be accurately controlled and reduced. Citrus Huanglongbing (HLB) is one of the most serious diseases that endanger the development of the world's citrus industry. It has caused a huge blow to the citrus industry in China, the United States, Brazil, Mexico, South Africa, and South Asia. The citrus psyllid is the only insect vector of citrus HLB, and it reproduces fast, has a strong ability to transmit the virus by sucking sap, and is difficult to identify because of its small size (average size of 2.5 mm), so early detecting of citrus psyllids and controlling their transmission are the key measures for prevention and control of HLB (Dala-Paula et al., 2019; Han et al., 2021; Ngugi et al., 2021). Citrus psyllids need an adapted host, mainly shoots, to survive (Gallinger and Gross, 2018). Traditional agricultural measures mainly kill citrus psyllids regularly with pesticides, which lead to the problems such as waste of agricultural materials and environmental and fruit pollution. There is 70–80% chance that citrus psyllids will transmit the HLB pathogen to healthy trees when they feed on the sap from the leaves of HLB trees and then fly to healthy trees. If farmers can detect citrus psyllids as soon as possible and spray pesticides accurately, the number of psyllids can be effectively reduced, the probability of psyllids sucking HLB diseased trees can be greatly reduced, and the transmission of HLB through psyllids can be effectively controlled. Therefore, through early detection and early control method, the population of citrus psyllids can be reduced, and the spread of HLB can be effectively prevented, thereby increasing the yield of citrus.

With the development of deep learning technology and the improvement of hardware equipment, the feasibility of image recognition of diseases and pests is constantly improving, more and more algorithms have been applied to the detection of plant diseases and pests (Ngugi et al., 2021). Accumulating evidence highlights the potential of employing CNNs in plant phenotyping settings. Their incorporation was proven to be very effective due to their capacity of distinguishing patterns and subtracting regularities from information under analysis. In plant sciences, there are many relevant and successful implementations including identification by examining seeds (chickpea; Taheri-Garavand et al., 2021a) or leaves (grapevine; Nasiri et al., 2021), tomato pest detection based on improved YOLOv3 (Liu and Wang, 2020, detection of mango anthracnose using neural networks (Singh et al., 2019), recognition of disease spots on soybean, citrus, and other plant leaves by deep learning (Arnal Barbedo, 2019), identification of rice-diseased leaves using transfer learning (Chen J. et al., 2020), classification and identification of agricultural pests in complex environments (Cheng et al., 2017), real-time detection of apple leaf diseases and insect pests (Jiang et al., 2019). These studies have shown that the neural network is successfully modeled under laboratory or field conditions with good recognition effect even under complex conditions, and transfer learning can also be performed according to different objects.

Convolutional neural network for image classification has become a standard structure to solve visual recognition problems, such as ResNet (He et al., 2016), VGGNet (Simonyan and Zisserman, 2014), GoogLeNet (Szegedy et al., 2014), and ResNetXt (Xie et al., 2016). The characteristic of these networks is that the learned representation gradually decreases in spatial resolution, which is not suitable for regional and pixel-level problems. The features learned through the above classification network essentially have low-resolution features. Therefore, the huge loss of resolution makes it difficult for the network to obtain accurate prediction results in tasks that are sensitive to spatial accuracy.

Target detection is constructed to solve the problems of classification and regression. At present, the target detection models based on deep learning are mainly divided into two categories: the two-stage method represented by faster region-based convolution neural networks (R-CNN) (Ren et al., 2017) and the one-stage method represented by Single Shot MultiBox Detector (SSD) (Liu et al., 2016). Although many different target detection algorithms have emerged, such as faster R-CNN (Ren et al., 2017), YOLO (Redmon et al., 2016), and other target detection algorithms, which achieve high recognition accuracy on conventional objects such as pedestrians and vehicles. However, the target of agricultural pests such as citrus psyllids is too small to be recognized by the above target detection algorithms. Chen et al. (2016) defined small targets with the characteristics of low-pixel occupancy in the whole picture, small candidate box, insufficient data sample, and so on. Because of these characteristics, the algorithms for small goals are still stuck in specific occasions, for example, building recognition in high altitude remote-sensing images (Xia et al., 2017), recognition of traffic lights in pictures (Behrendt et al., 2017), pedestrian recognition from the driver's perspective (Zhang et al., 2017).

The citrus psyllid studied in this experiment has the characteristics of small size, gray color, and easily being mistaken as branches, stems, and dead leaves. The deeper the layers of the neural network are, the more information will be lost. Therefore, it is difficult to extract useful feature information from the network for small target citrus psyllids. The similarity in color makes it difficult to identify the target, which makes citrus psyllids often be considered as branches or dead leaves.

**Abbreviations:** HLB, citrus Huanglongbing; CNN, convolutional neural network; CBAM, convolution block attention module.

Besides, the distribution of citrus psyllids is scattered, often concentrated in the bud, leaf back, and leaf veins, so each picture does not necessarily have a large number of psyllid samples. For images with fewer samples, the number of trainable positive sample boxes is greatly reduced, and it is not easy to train a model with superior performance. Cascade R-CNN (Cai and Vasconcelos, 2017) is a two-stage target detection model framework proposed in recent years, solving the IoU selection problem of the traditional target detection algorithm by cascading various detection models and having good detection performance for small targets. Therefore, to solve the problem of lack of citrus psyllids, a method of citrus psyllids enhancement based on semantic segmentation was explored, and the cascade R-CNN model for the small target recognition of citrus psyllids was improved in this study.

# MATERIALS AND METHODS

## Data Acquisition and Processing

The main location for collecting the data in this study is the Citrus HLB Test Base of South China Agricultural University (Longitude: 113.35875, Latitude: 23.15747), in Guangdong Province, China. The data collected in this experiment mainly used RGB images (visible spectrum 400–700 nm). The collected instruments include a mobile phone (Huawei Mate 40, China) with high-definition cameras and a Sony camera (Sony, ILCE-6400, made in China), with 4,000 × 5,000 pixels. The shooting distance was controlled within the range of 50–100 cm, and the shooting angle and orientation were not fixed. The shooting was performed in the morning, noon, and afternoon on a sunny day and under normal lighting condition. The targets in the picture are mainly citrus psyllids (average size of 2.5 mm) and fruit flies (average size of 5 mm) which are the main pests in citrus orchards. Although the citrus fruit fly is larger than the citrus psyllid, it still has the characteristics of being small and difficult to detect. In this study, the data of citrus psyllids and citrus fruit flies were used for model training. The model is proved to be transplantable to other small target pests by adding citrus fruit flies to the training. The data were collected in the spring of March, April, and May, from different Rutaceae plants (*Rutaceae Juss.*) including Shatangju (*Citrus reticulata Blanco*), kumquat potted plant [*Fortunella margarita* (Lour.) Swingle], and *Murraya exotica* (*Murraya exotica L.*) potted plant. Finally, a total of 500 high-definition sample images were obtained. The expensive price experimental data are shown in **Figure 1**.

## Target Sample and Data Set Enhancement

The relationship between the number of targets and images was analyzed through observation and mathematical statistics and is shown in **Figure 2**. Most of the pictures contain a small number of psyllid samples. The size of the citrus psyllid is much smaller than the size of the entire image, so this research belongs to the small target detection range.

The number of citrus psyllids in the picture can influence the training effect of the model. The more the number of psyllids in the picture, the more positive samples will be produced during the training, and the more the model will learn the characteristic information of psyllids. Therefore, to improve the identification effect of psyllids, the first step is to increase the sample number of citrus psyllids in the picture. For this kind of small target, there are many ways to enhance the small target, such as component stitching (Chen Y. et al., 2020), artificial augmentation by copy-pasting the small objects (Kisantal et al., 2019), AdaResampling (Ghiasi et al., 2020), and scale match (Yu et al., 2020). Due to the randomness of the target distribution, the number of targets distributed in each image is inconsistent. After cropping, a lot of pictures lack psyllid samples, which causes great difficulties in the recognition of the neural network.

To enhance the training effect of the model and improve the overall generalization ability of the model, the diversity of small target positions was increased by copying and pasting

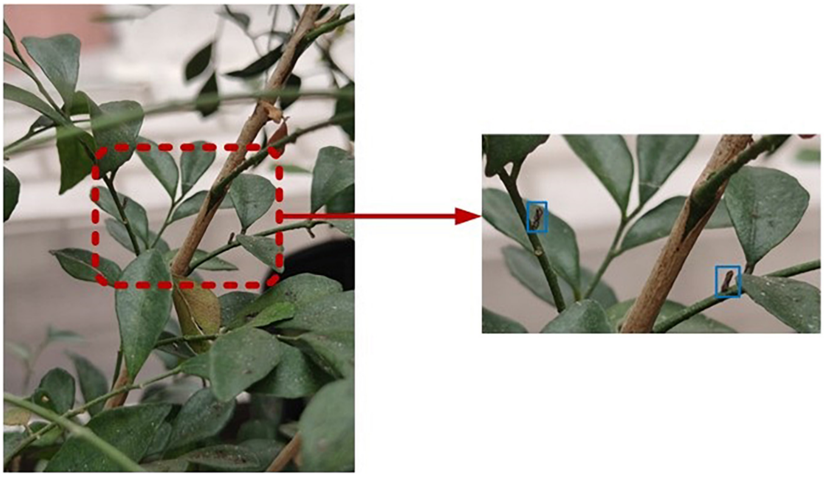


**FIGURE 1 |** Experimental data. The blue box is used to mark citrus psyllids.

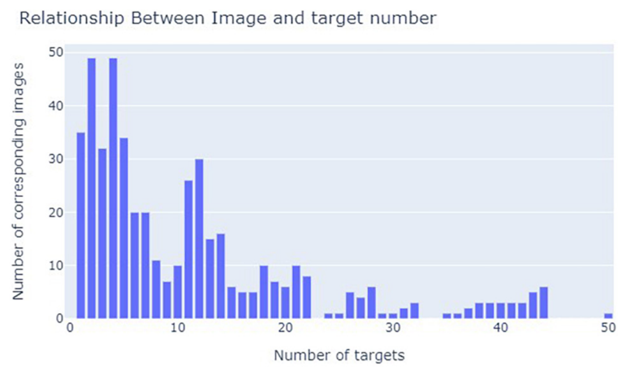


**FIGURE 2 |** Diagram of the relationship between the picture and the target quantity.

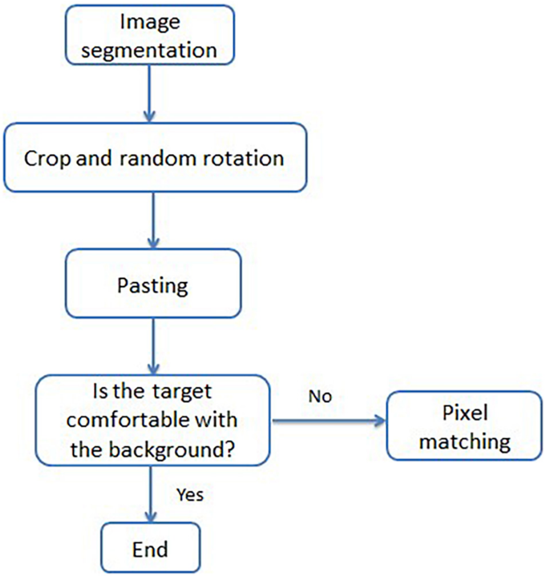


**FIGURE 3 |** Small sample number enhancement flowchart.

small targets multiple times and randomly pasting targets that do not overlap with existing targets. Based on the principle of replication, a method to increase the number of citrus psyllids was proposed, and the process is shown in **Figure 3**.

First, the pretrained semantic segmentation model U-Net (Ronneberger et al., 2015) was adopted to remove the redundant background of each image in the training set, only retaining the leaves and tree trunks. Then, each citrus psyllid sample was randomly copied and pasted onto the leaf or trunk position, ensuring that the pasted position does not coincide with the current position. Multiple copy and paste operations on different pictures with fewer targets were performed. The pasted target may not be in harmony with the background of the pasted position. When the pasted target is too bright or too dark compared with the surrounding background, the neural network will be very sensitive to the difference. The trained model can only have good generalization ability for the enhanced image, but a poor detection effect for the unenhanced natural lens image. Therefore, the color of the target that cannot be integrated into the background after pasting was modified manually, so that the target is as harmonious as possible with the background of the pasting place. The calculation formula for the overlap rate of the outer region (the background part removed by segmentation) and the sample overlap rate of the inner region (the leaves and trunk parts) is defined as Eqs. 1 and 2, where $U_{Outer\ region}$ is the overlap rate of the outer region and $U_{Samples}$ is the sample overlap rate of the inner region; $Area_{copy}$ represents the area where the sample is located after copying, $Area_{outer\ region}$ represents the outside area, and $U_{Outer\ region}$ represents the degree of overlap between the area where the sample is located and the outer area after copying. The larger the $U_{Outer\ region}$, the larger the area where the sample is located in the outer area after copying; n is the total number of original image samples; $Area_{Samples}$ represents the total area occupied by the original image samples, and $U_{Samples}$ represents the degree of overlap between the area occupied by the copied samples and the area occupied by the original image samples.

$$U_{Outer\ region} = \frac{Area_{copy}}{Area_{outer\ region}} \tag{1}$$

$$U_{Samples} = \frac{Area_{copy}}{Area_{Samples}}(Samples = Sample1 + Sample2 + \ldots + SampleN) \tag{2}$$

During the enhancement process, the $Area_{copy}$ of the sample area after being selected and pasted needs to meet the following conditions:

$$U_{Outer\ region} = 0,\ U_{Samples} = 0 \tag{3}$$

That is, the copied sample area needs to be in the area without overlapping with the original sample area. The example process of data enhancement is shown in **Figure 4**.

The white part in the segmentation diagram is the inner area, and the black part is the outer area.

Furthermore, the performance of the model is affected by the number of training sets. With the increase in training data, the recognition performance of the model will be improved to a certain extent. To increase the training data set and improve the general recognition ability of psyllids, offline resampling was used in the experiment. Two times the resampling rate was used to process the image set after the target samples were enhanced in the image. Since the image needs to be preprocessed before being input to the network, the data are not exactly the same after preprocessing, so there will be no overfitting of the training set.

## Image Preprocessing

The image set captured in this study was high-resolution (4,000 × 5,000 pixels). If these data are directly input into the network for training, it will lose a lot of useful citrus psyllid information due to compression during the training process. Therefore, the high-resolution images were cropped into nine

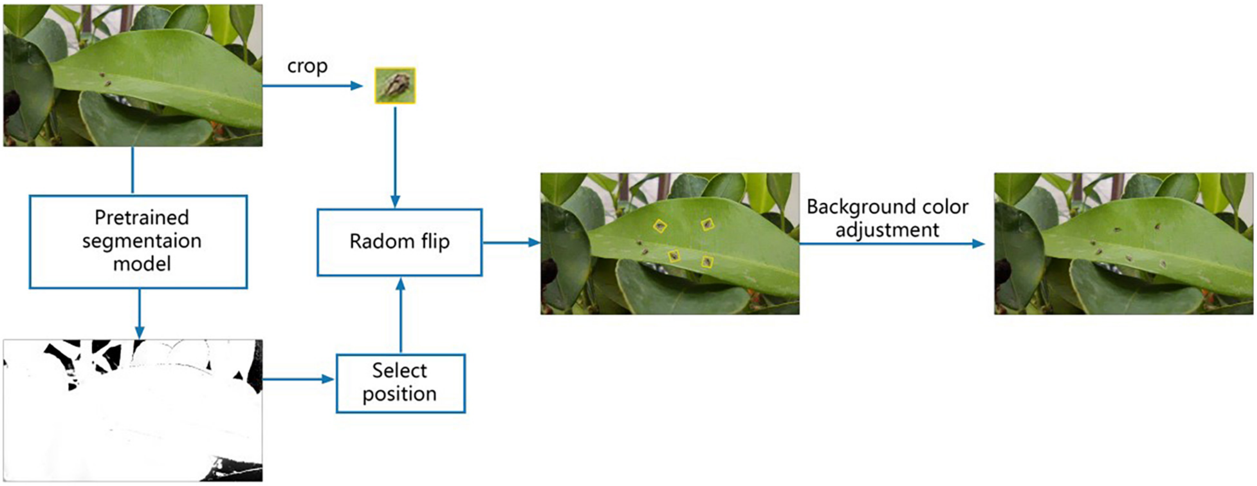


**FIGURE 4 |** Small sample number enhancement example diagram.

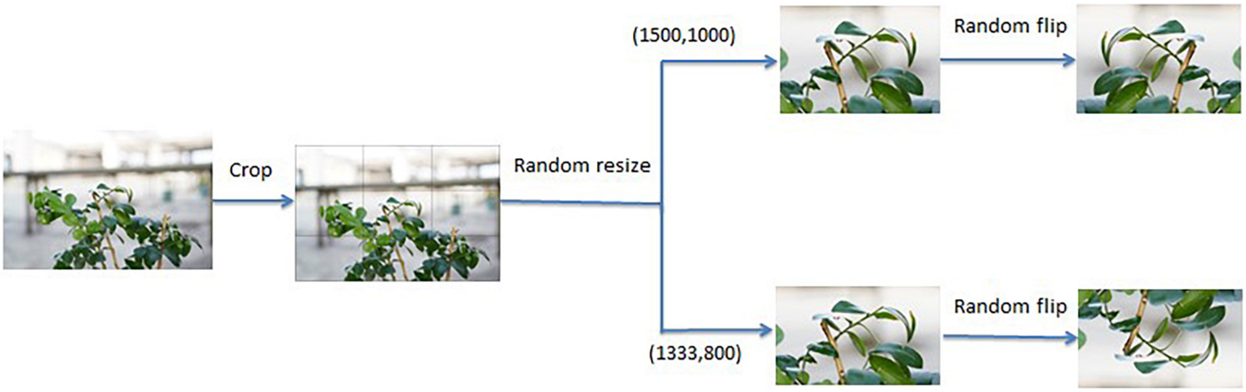


**FIGURE 5 |** Image cropping flowchart.

blocks before being input into the network in this study. The size of the input image affects the performance of the detection model, and the feature map generated by the feature extraction network is often dozens of times smaller than the original image, which will make it difficult for the detection network to capture the feature description of citrus psyllids. Therefore, this study uses multiscale training to improve the performance of the model. In view of the advantages of multiscale training, two scales (1500, 1000) and (1333, 800) were set, and each scale was randomly selected for training in each epoch. To increase the diversity of training samples, the input image was rotated randomly with 50% probability, and the image cropping flowchart is shown in **Figure 5**.

## Improved High-Resoultion Network for Feature Extraction

High-resoultion network (HRNet) (Sun et al., 2019) can learn enough high-resolution representations, which is different from the traditional classification network. In view of the small size of the citrus psyllids, if the traditional neural network is used for feature extraction, the key feature information of the shape and color of the citrus psyllids or the high-resolution feature information can be easily lost in the last layer of the network, whereas HRNet can maintain the high-resolution representation of citrus psyllid features by connecting high-resolution and low-resolution convolutions in parallel and enhance the high-resolution representation of citrus psyllid features by repeatedly performing multiscale fusion across parallel convolutions. It can achieve better results in small-area classification such as citrus psyllids. Therefore, HRNet was adopted as a feature extraction network to reduce the information loss of citrus psyllid features in this study.

To make the network pay more attention to the characteristics of citrus psyllids, a lightweight attention mechanism convolution block attention module (CBAM) (Woo et al., 2018) was added to the network, which is an attention mechanism module combining space and channel, where channel attention mechanism focuses on what features are meaningful from the perspective of channel, while space attention mechanism focuses on what features are meaningful from the space scale of image. In this study, the CBAM attention mechanism was added to the first-stage feature extraction and the second-, third-, and fourth-stage feature fusion of HRNet as shown in **Figure 6**. Adding CBAM blocks to the first stage enables the network to lock the target features that need attention in the initial stage of feature extraction. Adding the CBAM fusion module to the second stage, the third stage, and the fourth stage fully enable the extraction of the important citrus psyllid feature information of different resolutions when fusing the features of different resolutions.

## Feature Fusion Strategy

As shown in **Figure 6**, the final network layer of the HRNet outputs four feature maps with different resolutions. Because of the small size of the citrus psyllids, the information of the citrus psyllid characteristic map at each resolution may be lacking, and different resolutions' information needs to be combined to complement each other. To make full use of the feature maps at different resolutions, the atrous spatial pyramid pooling (ASPP) (Chen et al., 2017) (dilated space convolution pooled pyramid) structure was used for feature fusion at different resolutions in

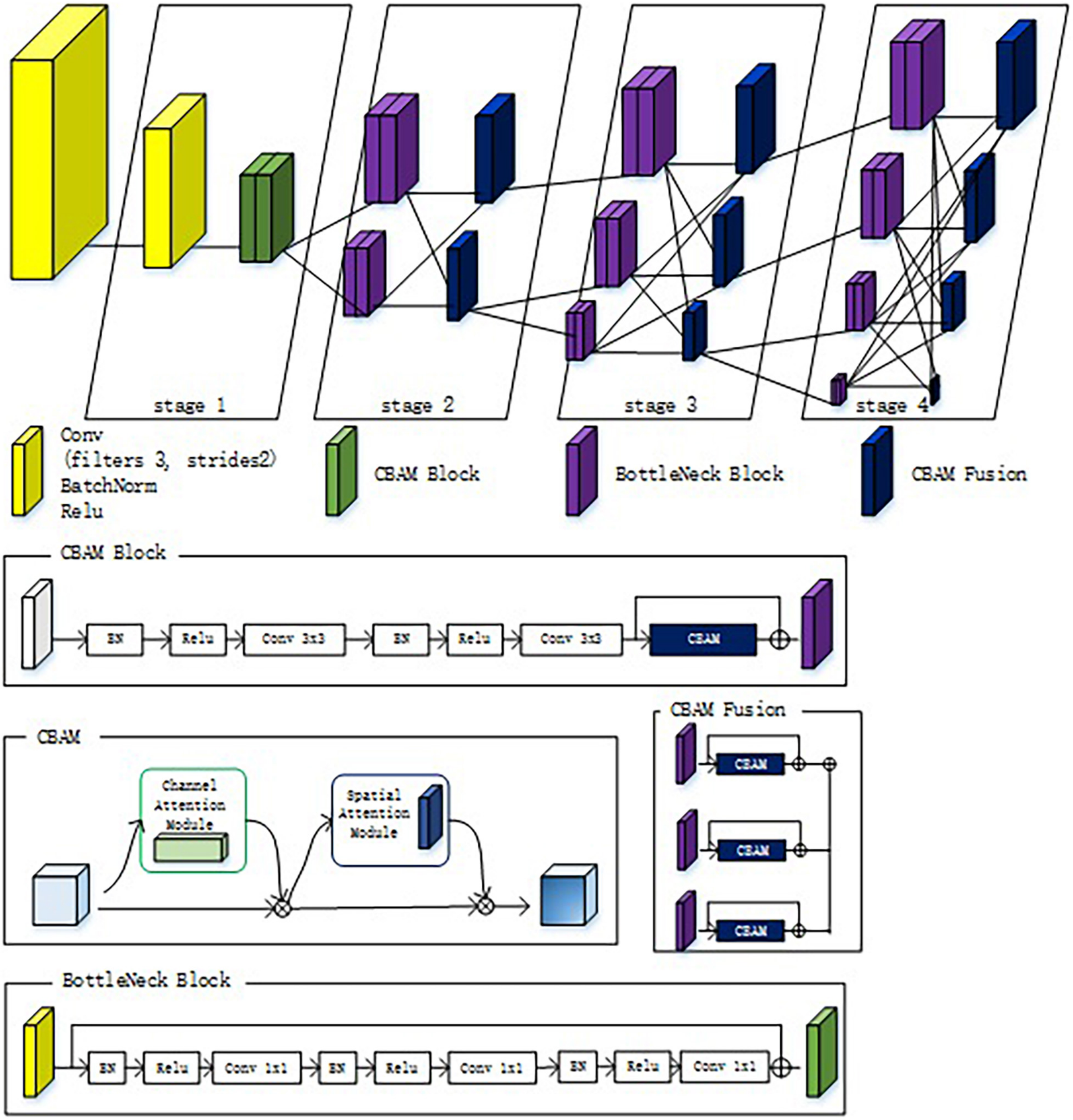


**FIGURE 6 |** High-resoultion network (HRNet) model with CBAM attention mechanism.

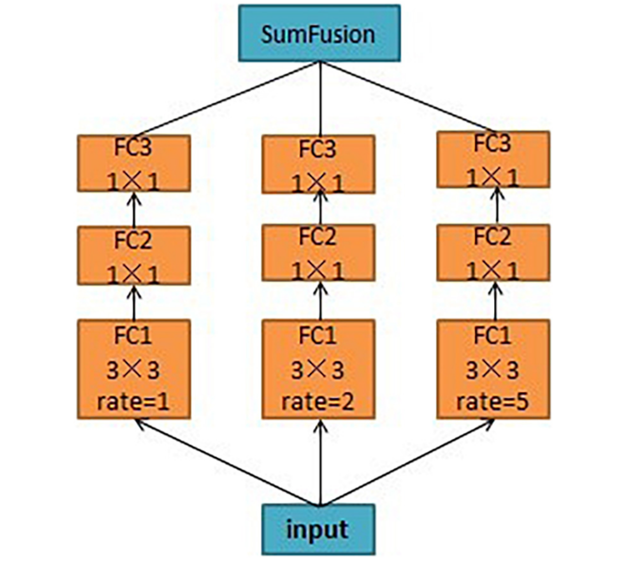


**FIGURE 7 |** Atrous spatial pyramid pooling (ASPP) structure.

this study. In ASPP, the extracted characteristics of citrus psyllids are input into the dilated convolutions at different sampling rates, which is equivalent to capturing the characteristic information of citrus psyllids at multiscale. The dilated convolutions in the ASPP structure can expand the field of view without losing the resolution, and features under different dilation rates are collected in parallel to obtain the multiscale information of citrus psyllids, and such these operations can improve the recognition effect of the entire network, which is shown in **Figure 7**. However, ASPP only uses a large dilation rate [such as (1, 3, 6, 12)] which is only effective for large object detection, but not suitable for citrus psyllid detection. To make full use of the advantages of ASPP and more suitable for the characteristics of small target detection in this study, the dilation rate was designed into a zigzag structure, the dilation rate was set to (1, 2, 5), and the feature pyramid networks (FPN) (Pan et al., 2019) which is shown in **Figure 8** was used to fuse the 4 citrus psyllids features at different resolutions processed by ASPP.

Besides, the shape and color of citrus psyllids are similar to branches, tree stems, and dead leaves, so the model will produce more positive and negative samples with higher loss during the training process. For example, the model predicts part of the trunk as citrus psyllids. Using the traditional random sampling method, a large number of difficult samples, such as branches, may be missed. The model trained using the random sampling method cannot distinguish citrus psyllids, tree branch, and tree stem. So, in this study, random sampling method in cascade R-CNN was replaced by an online hard sample mining strategy (Shrivastava et al., 2016), and the suggestion box with high-loss value was given high priority to be sampled. The strategy is as follows: in the training process, the ROI loss in each stage is sorted, and the first 64 samples of ROI loss in the positive samples and the first 192 samples of ROI loss in the negative samples as training samples are selected according to the sorting structure.

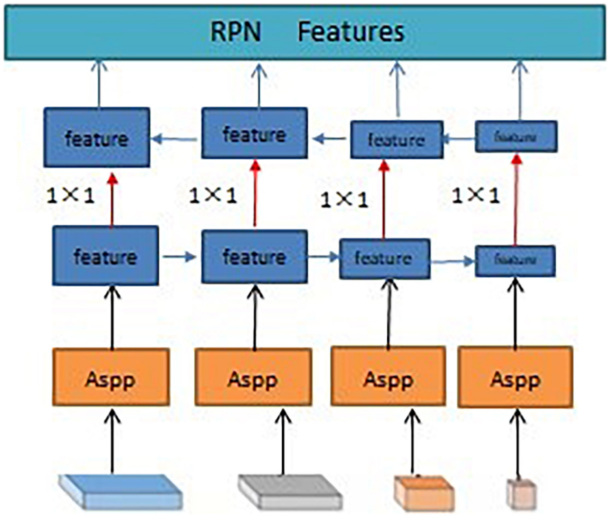


**FIGURE 8 |** Feature pyramid networks (FPN) structure.

# RESULTS

## Experiment Setting and Evaluation Index

The models for identifying citrus psyllids were trained and tested under the desktop computer with inter-i7-9800x CPU, GeForce GTX 1080ti GPU, Ubuntu 16.04 operating system, and PyTorch deep learning framework. The average detection time of high-resolution images is 10 ms per image. To evaluate the effectiveness of the citrus psyllid detection method proposed in this study, the average precision (*AP*) and mean average precision (*mAP)* were chosen as evaluation indicators, where *AP* is a measure of the average precision value of a category detection, using the precision rate to integrate the recall rate, as shown in eq. (4), *mAP* is a measure of the average value of all types of *AP*, as shown in formula (5).

$$AP = \int_{0}^{1} Precision\ rate\ dRecall\ rate \tag{4}$$

$$mAP = \frac{1}{C}\sum_{c\in C} AP(c) \tag{5}$$

Where ***c*** represents a certain category, and ***C*** represents the general category.

## Small Target Number Enhancement Results

**Figure 9** shows the comparison matrix of the number of small targets before and after enhancement. The top row of **Figure 9A** is the original image, and the bottom row is the enhanced image. **Figure 9B** is the contrast matrix of the number of objects before and after the enhancement. Through observation, it can be found that using the small sample number enhancement method based on the semantic segmentation model improves the number of small targets in each picture in the data set. As the number of

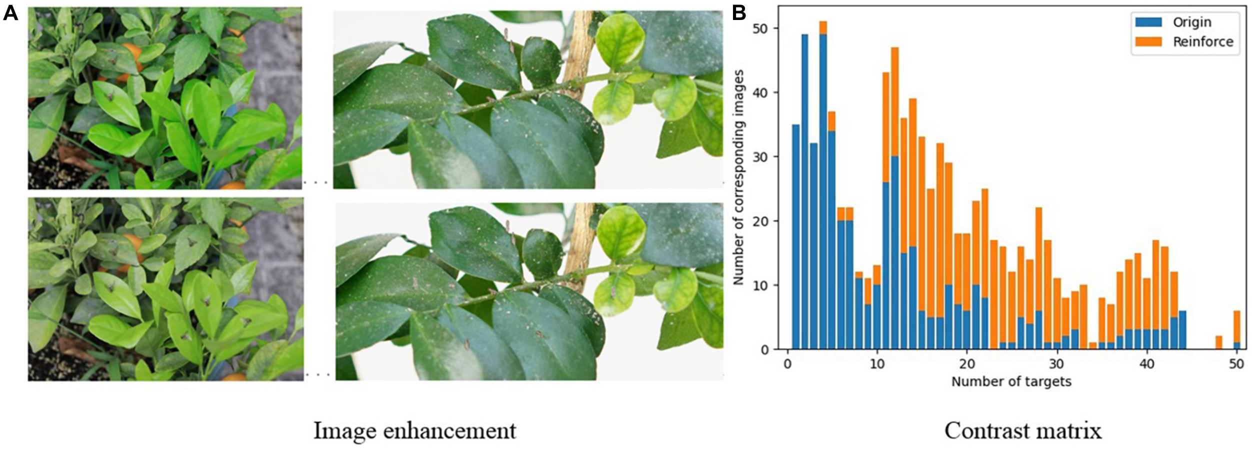


**FIGURE 9 |** Comparison of the number of small targets before and after enhancement **(A)** image enhancement **(B)** contrast matrix.

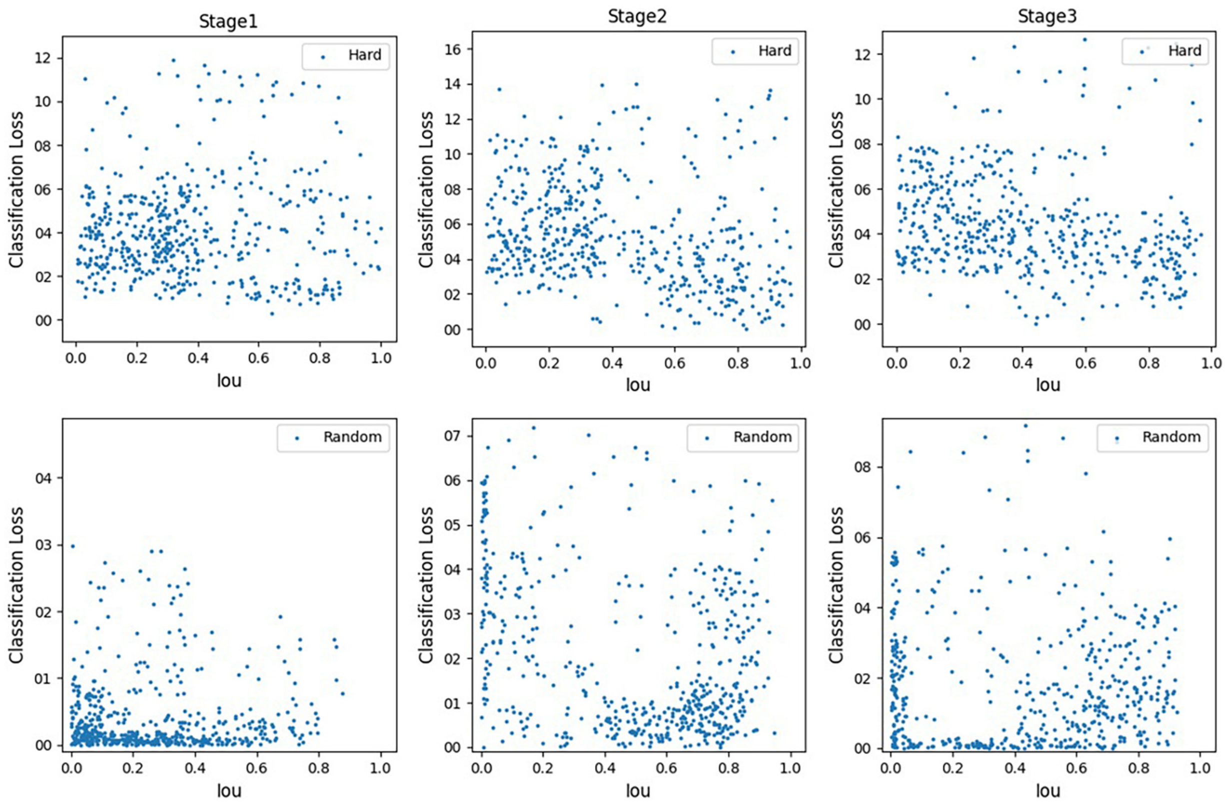


**FIGURE 10 |** Comparison of difficult sampling and random sampling.

small samples per picture increases, a large number of useful positive samples is increased, which can effectively increase the performance of the model.

## Online Hard Sample Mining

**Figure 10** is a comparison diagram of sampling three cascades using random sampling and difficult sample mining methods in cascade R-CNN. It can be found that most random samples are distributed in areas with low classification loss. This is because negative samples contain lots of low-loss samples, so random sampling has a high probability of collecting these low-loss samples. However, the generalization ability of the model trained with lots of low-loss samples is not strong, and it could not classify hard samples well. Also, it shows that the samples collected by the hard sampling method are concentrated in the high-loss area. This is because the hard sample mining method will give priority to the samples with high classification loss, even if the samples contain a large number of samples with low loss. Therefore, the online hard sample mining algorithm can improve the ability of the model to identify difficult samples, so as to improve the overall performance of the model.

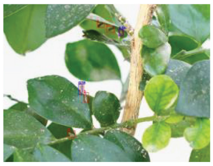
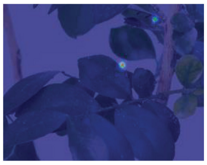
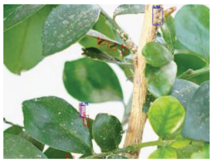
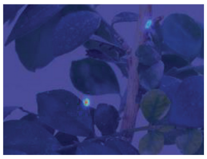
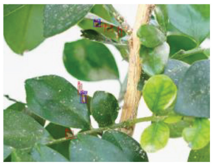
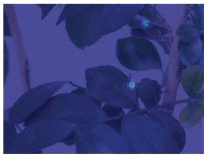
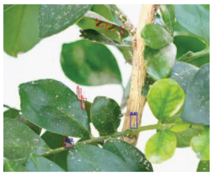
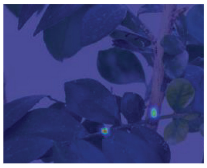
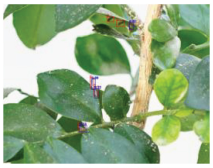
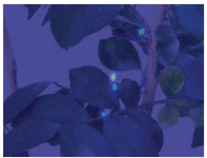
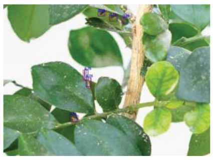
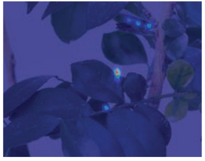
## The Performance of Modeling

Some common models that are ResNet, VGGNet, and ResNetXt were adopted for comparison with the proposed method in this study. The *AP* and *mAP* values of the test results of the models are shown in **Table 1**. Data enhancement represents small

**TABLE 1 |** Different model recognition effects.

| Models | Average precision | | Mean average precision |
|---|---|---|---|
| | Citrus psyllids | Fruit flies | |
| ResNet50 + Data enhancement | 67.4% | 78.48% | 72.94% |
| ResNet101 + Data enhancement | 66.45% | 75.33% | 70.89% |
| ResNetXt101 + Data enhancement | 67.78% | 77.82% | 73.3% |
| VGG16 + Data enhancement | 66.89% | 77.37% | 72.13% |
| HRNet + Data enhancement | 73.54% | 80.25% | 76.89% |
| Improved HRNet + Data enhancement | 81.89% | 84.73% | 76.89% |
| Improved HRNet + Offline resampling | 72.89% | 76.25% | 74.57% |
| Improved HRNet + ASPP + FPN + Online hard sample mining strategy + Data enhancement | 88.78% | 91.64% | 90.21% |

**TABLE 2 |** Visualization results of different models.

| Model | Data | Visualization of results | Heat map |
|---|---|---|---|
| ResNet50 + Data enhancement | |  |  |
| ResNet101 + Data enhancement | |  |  |
| ResNetXt101 + Data enhancement | |  |  |
| VGG16 + Data enhancement | |  |  |
| Improved HRNet + Data enhancement | |  |  |
| Improved HRNet + ASPP + FPN + Online hard sample mining strategy + Data enhancement | |  |  |

target number enhancement and offline resampling. In **Table 1**, it can find that the improved HRNet + Data enhancement model has 9.0% higher *AP* than the improved HRNet + Offline resampling model. It can also be found that using small target number enhancement can improve the recognition effect of citrus psyllids to some extent. From the comparison, the *AP* values of the HRNet model were 6.14, 7.09, 5.76, and 6.65% higher than those of ResNet50, ResNet101, next101, and VGG16, respectively. The reason is that citrus psyllid is small in size, and it is easy to lose information when using traditional CNN to extract the characteristics of citrus psyllids. The model cannot fully learn the shape, size, distribution, and color information of citrus psyllids, so it does not have good generalization ability. The background area in the picture is much larger than the total area of the citrus psyllids. The model needs to find out the characteristic information of the citrus psyllids from many characteristic information. Therefore, adding an attention mechanism can enhance the attention of the network to the characteristic of the citrus psyllids, extracting key information from the shape, distribution, color, and size of citrus psyllids. In **Table 1**, the improved HRNet + Data enhancement model has higher *AP* than the HRNet model, which proves that adding the attention mechanism can improve model performance. The comprehensive improvement scheme (improved HRNet + ASPP + FPN + Online hard sample mining strategy + Data enhancement) has the best performance in detecting citrus psyllids, which is more than 10% higher than that of other models. This shows that the addition of ASPP and FPN structures can fully complement the characteristics of citrus psyllids at different resolutions and solve the problem of lack of information on citrus psyllids at a single resolution. Adding an online hard sample mining strategy can allow the model to focus on learning features of objects similar to citrus psyllids and solve the problem of indistinguishable branches, stalks, and dead leaves from citrus psyllids.

The final proposed model not only performs well in the detection of citrus psyllids but also achieves good recognition performance on small targets such as citrus fruit flies. Citrus fruit flies are larger and have more obvious appearance characteristics than citrus psyllids, such as double wings and heads, so they are easier to identify than citrus psyllids. However, citrus fruit flies are also in the recognition range of small targets, and there are fewer pixels in the identifiable area. Therefore, it is difficult to accurately extract the characteristic information of fruit flies. Through **Table 1**, it can be found that the final model proposed can achieve 91.64% accuracy in citrus fruit fly recognition.

**Table 2** shows the visual prediction results of each model on the original data. The red box is the citrus psyllid label, and the blue box is the prediction result. From the results, it can be found that the model using the improved HRNet as the feature extraction network can achieve a better recognition effect. At the same time, comparing the model of ResNet50 and the model of ResNet101 shows that the deeper the network layer is, the more unfavorable it is to recognize the citrus psyllids. From the results of the heat map, it shows that the model proposed in this study based on improved HRNet + Data enhancement + SPP + FPN + online hard sample mining can better extract the feature information of citrus psyllids.

**Figure 11** shows the training loss diagram of the proposed comprehensive improvement scheme, where **Figure 11A** represents the total loss curve calculated with 1:0.5:0.25 weights for stage 1, stage 2, and stage 3, and **Figure 11B** represents the loss curve for RPN bbox on the training set.

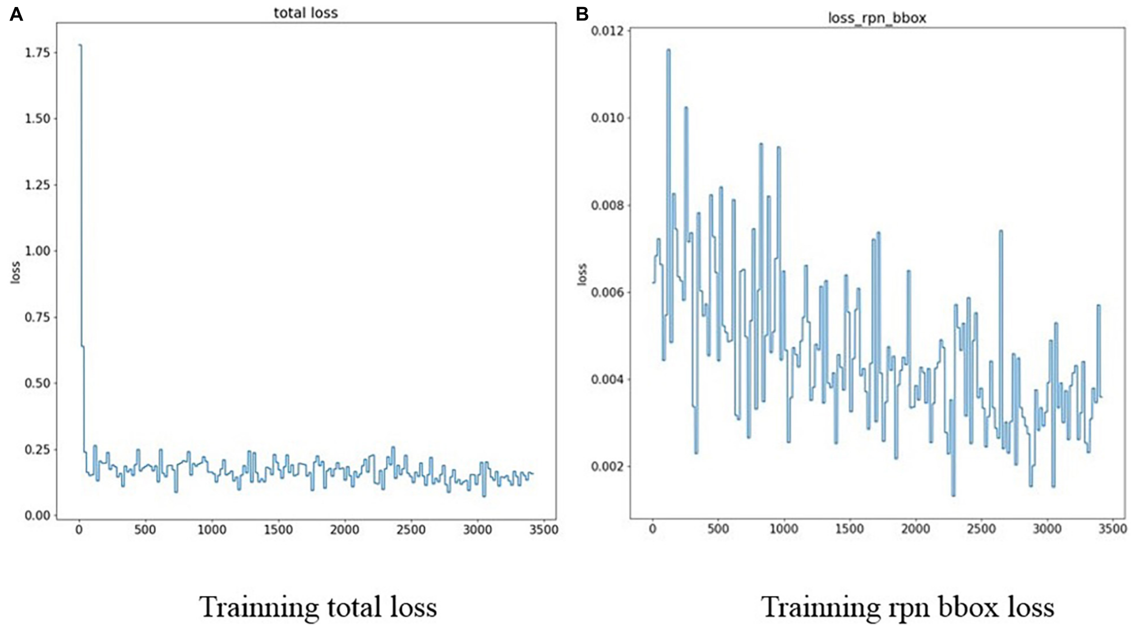


**FIGURE 11 |** Training loss diagram **(A)** training total loss **(B)** training RPN bbox loss.

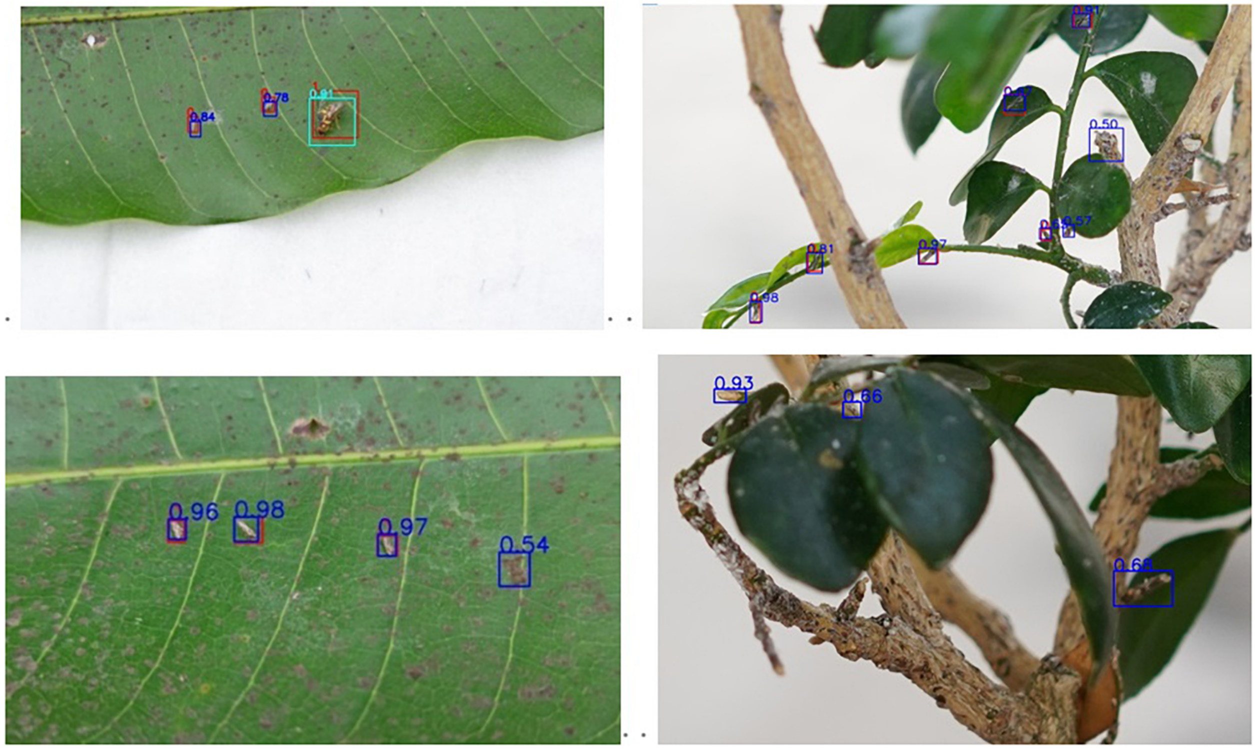


**FIGURE 12** | Prediction results.

The prediction results for testing data of the proposed model are shown in **Figure 12**, where the red box is the label box, the blue box is the citrus psyllids prediction box, and the mint green is the fruit fly prediction box; the label category 0 represents the citrus psyllids and the label category 1 represents the fruit flies. The result shows that the overlap between the label frame and the prediction frame is very high, which proves that the improved model has a good performance in detecting citrus psyllids.

## DISCUSSION

Agricultural pests are small and difficult to be found, and traditional neural networks cannot meet the recognition of small target agricultural pests such as citrus psyllids. This study explores a suitable network and solution for the identification of citrus psyllids. The model proposed in this study can achieve a good recognition effect when the target is larger or slightly smaller than the size of the citrus psyllids, and there are some or no objects similar to the target in the background. Unfortunately, if the detection target is too small, much smaller than the psyllids, such as the red spider citrus (average size of 0.39 mm), the detection effect is not satisfactory using the proposed method of this article. Also, if the detection target is difficult to capture, such as the rice stem borer whose larvae burrow into the rice stalk to eat, it is hard to identify without manual intervention. Besides, if the plant background has a large number of parts that are very similar to the characteristics of the detection target, the recognition accuracy of the method proposed in this article might be reduced.

The use of RGB camera shooting in this study has a good promotion ability without costly equipment and professionals. Outdoor acquisition of RGB images is affected by lighting conditions. When the weather is cloudy, the general brightness of the image will be low due to the lack of light, which will lead to a decrease in the accuracy of target recognition. Besides, the different angles between the camera and the object will affect the recognition accuracy to a certain extent, and these problems can be solved by adding training data sets. If light conditions during acquisition are of interest, there is always the possibility of using scanners (Taheri-Garavand et al., 2021b). In this experiment, the model is trained by inputting pictures with different illumination conditions, distances, and angles. The complexity of the data sources in this experiment shows that this model can be applied to various environments, including laboratory environments and complex external environments, so the model has good generalization ability.

The proposed citrus psyllid detection method based on machine vision can be applied to the actual field monitoring of orchards, and the detection model can be deployed on edge computing devices to help orchard managers more easily monitor the occurrence of orchard pests and summarize the changes. Also, the proposed model can be deployed on terminal devices such as RGB cameras, mobile phones, and cameras mounted on an insect trapper or mobile platform to monitor pests in real time, greatly reducing labor costs, time costs, and resources.

## CONCLUSION

The citrus psyllid has the characteristics of small size, similar color, and shape to branches, stems, and dead leaves, which causes difficulties to actual field monitoring based on machine vision. To detect citrus psyllids effectively, a comprehensive detection solution based on a high-definition camera for field detection is introduced in this paper. In view of uneven distribution of a small target in the image, a sample enhancement method to increase the number of target samples was first proposed. The detection model was built based on cascade R-CNN, which was improved by using HRNet as the feature extraction network, adding a lightweight attention mechanism CBAM in HRNet to make the network pay more attention to the citrus psyllid features, adding the ASPP structure to extract the high-resolution features from different scales, and integrating the features of different scales with FPN structure. In view of the similarity between citrus and tree branches, tree stems, and dead leaves, online hard sampling mining strategy was adopted. The results show that the improved cascade R-CNN detection model achieved 90.21% *mAP* on the test set, which is much higher than that of other models for the recognition of small targets. After deploying the detection model on edge computing devices, the proposed comprehensive solution can provide real-time detection of citrus psyllids in practical application, reducing the cost of artificial patrol and waste of resources. The solution proposed in this article provides a reference for field camera detection and identification of pests. In addition, early detection and treatment of citrus psyllids can reduce and prevent the occurrence of citrus HLB in citrus orchards.

## DATA AVAILABILITY STATEMENT

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

## AUTHOR CONTRIBUTIONS

FD conceptualized the experiments, selected the algorithms, collected and analyzed the data, and wrote the manuscript. FW and DY trained the algorithms and collected and analyzed the data. SL and XC wrote the manuscript. XD and YL supervised the project and revised the manuscript. All authors discussed and revised the manuscript.

## FUNDING

This work was supported by the Key-Area Research and Development Program of Guangzhou (grant number 202103000090), Key-Area Research and Development Program of Guangdong Province (grant number 2019B020214003), Key-Areas of Artificial Intelligence in General Colleges and Universities of Guangdong Province (grant number 2019KZDZX1012), Laboratory of Lingnan Modern Agriculture Project (grant number NT2021009), National Natural Science Foundation of China (grant number 61675003), National Natural Science Foundation of China (grant number 61906074), and Guangdong Basic and Applied Basic Research Foundation (grant number 2019A1515011276).

## REFERENCES

Arnal Barbedo, J. G. (2019). Plant disease identification from individual lesions and spots using deep learning. *Biosyst. Eng.* 180, 96–107. doi: 10.1016/j.biosystemseng.2019.02.002

Behrendt, K., Novak, L., and Botros, R. (2017). *A Deep Learning Approach to Traffic Lights: Detection, Tracking, and Classification*. Piscataway, NJ: IEEE. doi: 10.1109/ICRA.2017.7989163

Cai, Z., and Vasconcelos, N. (2017). *Cascade R-CNN: Delving into High Quality Object Detection*. Ithaca: Cornell University Press. doi: 10.1109/CVPR.2018.00644

Chen, C., Liu, M. Y., Tuzel, O., and Xiao, J. (2016). "R-CNN for small object detection," in *Proceedings of the Computer Vision – ACCV 2016. ACCV 2016. Lecture Notes in Computer Science*, eds S. H. Lai, V. Lepetit, K. Nishino, and Y. Sato (Cham: Springer). doi: 10.1016/j.neucom.2017.09.098

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

Chen, L., Papandreou, G., Schroff, F., and Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/1706.05587 (accessed September 24, 2020).

Chen, Y., Zhang, P., Li, Z. L. Y., Zhang, X., and Meng, G. (2020). Feedback-driven data provider for object detection. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/2004.12432 (accessed January 10, 2021).

Cheng, X., Zhang, Y., Chen, Y., Wu, Y., and Yue, Y. (2017). Pest identification via deep residual learning in complex background. *Comput. Electron. Agric.* 141, 351–356. doi: 10.1016/j.compag.2017.08.005

Dala-Paula, B. M., Plotto, A., Bai, J., Manthey, J. A., Baldwin, E. A., Ferrarezi, R. S., et al. (2019). Effect of huanglongbing or greening disease on orange juice quality, a review. *Front. Plant Sci.* 9:1976. doi: 10.3389/fpls.2018.01976

Gallinger, J., and Gross, J. (2018). Unraveling the host plant alternation of cacopsylla pruni – adults but not nymphs can survive on conifers due to Phloem/Xylem composition. *Front. Plant Sci.* 9:484. doi: 10.3389/fpls.2018.00484

Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.-Y., Cubuk, E. D., et al. (2020). Simple Copy-Paste is a strong data augmentation method for instance segmentation. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/2012.07177 doi: 10.1109/CVPR46437.2021.00294 (accessed January 25, 2021).

Han, H.-Y., Cheng, S.-H., Song, Z.-Y., Ding, F., and Xu, Q. (2021). Citrus huanglongbing drug control strategy. *J. Huazhong Agric. Univ.* 40, 49–57. doi: 10.13300/j.cnki.hnlkxb.2021.01.006

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE Computer Society), 770–778. doi: 10.1109/CVPR.2016.90

Jiang, P., Chen, Y., Liu, B., He, D., and Liang, C. (2019). Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. *IEEE Access* 7, 59069–59080. doi: 10.1109/ACCESS.2019.2914929

Kisantal, M., Wojna, Z., Murawski, J., Naruniec, J., and Cho, K. (2019). "Augmentation for small object detection," in *Proceedings of the 9th International Conference on Advances in Computing and Information* (Reston: AIAA). doi: 10.3390/s21103374

Liu, J., and Wang, X. (2020). Tomato diseases and pests detection based on improved yolo v3 convolutional neural network. *Front. Plant Sci.* 11:898. doi: 10.3389/fpls.2020.00898

Liu, W., Anguelov, D., Erhan, D. S. C., Reed, S., and Fu, C. Y. (2016). *SSD: Single shot MultiBox Detector*. Cham: Springer. doi: 10.1007/978-3-319-46448-0_2

Nasiri, A., Taheri-Garavand, A., Fanourakis, D., Zhang, Y., and Nikoloudakis, N. (2021). Automated grapevine cultivar identification via leaf imaging and deep convolutional neural networks: a Proof-of-Concept study employing primary iranian varieties. *Plants* 10:1628. doi: 10.3390/plants10081628

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. *Inform. Process. Agric.* 8, 27–51. doi: 10.1016/j.inpa.2020.04.004

Pan, H., Chen, G., and Jiang, J. (2019). Adaptively dense feature pyramid network for object detection. *IEEE Access* 7, 81132–81144. doi: 10.1109/access.2019.2922511

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 (CVPR)* (New Jersy: IEEE). 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. Pattern Anal. Mach Intell.* 39, 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Ronneberger, O., Fischer, P., and Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Berlin: Springer International Publishing. doi: 10.1007/978-3-319-24574-4_28

Shrivastava, A., Gupta, A., and Girshick, R. (2016). "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE Conference On Computer Vision & Pattern Recognition* (San Juan: IEEE), 761–769. doi: 10.1109/CVPR.2016.89

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for Large-Scale image recognition. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/1409.1556 doi: 10.3390/s21082852 (accessed September 20, 2020).

Singh, U. P., Chouhan, S. S., Jain, S., and Jain, S. (2019). *Multilayer Convolution Neural Network for the Classification of Mango Leaves Infected by Anthracnose Disease*. Piscataway, NJ: IEEE. doi: 10.1109/ACCESS.2019.2907383

Sun, K., Xiao, B., Liu, D., and Wang, J. (2019). Deep High-Resolution representation learning for human pose estimation. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/1902.09212 (accessed August 10, 2020).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2014). Going deeper with convolutions. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/1409.4842 (accessed August 20, 2020).

Taheri-Garavand, A., Nasiri, A., Fanourakis, D., and Fatahi, S. (2021a). Automated in situ seed variety identification via deep learning: a case study in chickpea. *Plants* 10:1406. doi: 10.3390/plants10071406

Taheri-Garavand, A., Rezaei Nejad, A., Fanourakis, D., Fatahi, S., and Ahmadi Majd, M. (2021b). Employment of artificial neural networks for non-invasive estimation of leaf water status using color features: a case study in *Spathiphyllum wallisii*. *Acta Physiol. Plant.* 43:78. doi: 10.1007/s11738-021-03244-y

Woo, S., Park, J., Lee, J. Y., and Kweon, I. S. (2018). "CBAM: convolutional block attention module," in *Proceedings of the Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science*, eds V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss (Cham: Springer). doi: 10.1007/978-3-030-01234-2_1

Xia, G., Bai, X., Ding, J., and Zhu, Z. (2017). DOTA: a large-scale dataset for object detection in aerial images. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/1711.10398 (accessed October 10, 2020).

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2016). Aggregated residual transformations for deep neural networks. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/1611.05431 (accessed October 1, 2020).

Yu, X., Gong, Y., Jiang, N., Ye, Q., and Han, Z. (2020). "Scale match for tiny person detection," in *Proceedings of the 2020 IEEE Winter Conference On Applications of Computer Vision (WACV)* (Piscataway, NJ: IEEE). doi: 10.1001/archneur.1994.00540150054016

Zhang, S., Benenson, R., and Schiele, B. (2017). CityPersons: a diverse dataset for pedestrian detection. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/1702.05693 doi: 10.1109/CVPR.2017.474 (accessed October 20, 2020).

**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.

ORIGINAL RESEARCH
published: 28 February 2022
doi: 10.3389/fpls.2022.829479

# Rubber Leaf Disease Recognition Based on Improved Deep Convolutional Neural Networks With a Cross-Scale Attention Mechanism

*Tiwei Zeng[1,2], Chengming Li[2], Bin Zhang[1,2], Rongrong Wang[2], Wei Fu[1,2]*, Juan Wang[2]* and Xirui Zhang[1,2]*

*[1] School of Information and Communication Engineering, Hainan University, Haikou, China, [2] Mechanical and Electrical Engineering College, Hainan University, Haikou, China*

Natural rubber is an essential raw material for industrial products and plays an important role in social development. A variety of diseases can affect the growth of rubber trees, reducing the production and quality of natural rubber. Therefore, it is of great significance to automatically identify rubber leaf disease. However, in practice, different diseases have complex morphological characteristics of spots and symptoms at different stages and scales, and there are subtle interclass differences and large intraclass variation between the symptoms of diseases. To tackle these challenges, a group multi-scale attention network (GMA-Net) was proposed for rubber leaf disease image recognition. The key idea of our method is to develop a group multi-scale dilated convolution (GMDC) module for multi-scale feature extraction as well as a cross-scale attention feature fusion (CAFF) module for multi-scale attention feature fusion. Specifically, the model uses a group convolution structure to reduce model parameters and provide multiple branches and then embeds multiple dilated convolutions to improve the model's adaptability to the scale variability of disease spots. Furthermore, the CAFF module is further designed to drive the network to learn the attentional features of multi-scale diseases and strengthen the disease features fusion at different scales. In this article, a dataset of rubber leaf diseases was constructed, including 2,788 images of four rubber leaf diseases and healthy leaves. Experimental results show that the accuracy of the model is 98.06%, which was better than other state-of-the-art approaches. Moreover, the model parameters of GMA-Net are only 0.65 M, and the model size is only 5.62 MB. Compared with MobileNetV1, V2, and ShuffleNetV1, V2 lightweight models, the model parameters and size are reduced by more than half, but the recognition accuracy is also improved by 3.86–6.1%. In addition, to verify the robustness of this model, we have also verified it on the PlantVillage public dataset. The experimental results show that the recognition accuracy of our proposed model is 99.43% on the PlantVillage dataset, which is also better than other state-of-the-art approaches. The effectiveness of the proposed method is verified, and it can be used for plant disease recognition.

**Keywords: rubber leaf disease recognition, lightweight neural network, attention mechanisms, GMA block, GMA-Net**

OPEN ACCESS

***Edited by:***
*Valerio Giuffrida,
Edinburgh Napier University,
United Kingdom*

***Reviewed by:***
*Chuanlei Zhang,
Tianjin University of Science
and Technology, China
Huaming Chen,
University of Adelaide, Australia*

****Correspondence:***
*Wei Fu
994026@hainanu.edu.cn
Juan Wang
wj-jdxy@hainanu.edu.cn*

***Specialty section:***
*This article was submitted to
Technical Advances in Plant Science,
a section of the journal
Frontiers in Plant Science*

***Received:*** *05 December 2021*
***Accepted:*** *24 January 2022*
***Published:*** *28 February 2022*

***Citation:***
*Zeng T, Li C, Zhang B, Wang R,
Fu W, Wang J and Zhang X (2022)
Rubber Leaf Disease Recognition
Based on Improved Deep
Convolutional Neural Networks With
a Cross-Scale Attention Mechanism.
Front. Plant Sci. 13:829479.
doi: 10.3389/fpls.2022.829479*

## INTRODUCTION

The rubber tree is one of the most important economic crops in the tropics, and the planting area of rubber trees in China is more than 1.16 million hectares, more than half of which are planted in Hainan Province (Ali et al., 2020; Li and Zhang, 2020). The milky latex extracted from the tree is the primary source of natural rubber, which is an essential raw material for industrial products. However, rubber leaf diseases cause annual losses of approximately 25% of the total yield of natural rubber and cause significant economic losses. A variety of diseases can affect the growth of rubber trees, reducing the production of natural rubber and seriously hindering the development of the natural rubber industry. Hence, the identification and diagnosis of rubber leaf diseases (e.g., powdery mildew disease, rubber tree anthracnose, periconla leaf spot disease, and Abnormal Leaf Fall Disease) are of great significance for increasing the yield of natural rubber and have received extensive attention from rubber planting workers and experts on disease and pest control. Unfortunately, manual identification and diagnosis are time-consuming and laborious in practice, and the recognition accuracy does not satisfy the requirement.

To solve the problems caused by the manual diagnosis, researchers have proposed some machine learning-based methods for plant disease recognition (Sladojevic et al., 2016; Hu et al., 2018). The plant disease recognition method based on traditional machine learning is mainly through the manual design of classification features, such as color features (Semary et al., 2015), shape features (Parikh et al., 2016), texture features (Mokhtar et al., 2016), or the fusion of two or more manual features (Shin et al., 2020). However, the manual features in these approaches are selected based on human experience, which limits the generalizability of the models.

Recently, deep convolutional neural networks (DCNNs) have been widely applied in image and video classification tasks (Ren et al., 2020). Compared with traditional machine vision algorithms, DCNN can complete feature extraction and classification tasks through the self-learning ability of the network without manual design features (Liu et al., 2017). Anagnostis et al. (2020) offered a Walnut disease classification system using CNN with an accuracy range from 92.4 to 98.7%. Zhu et al. (2019) investigated a two-way attention model for plant recognition and validated the method in four challenging datasets, and the recognition accuracy reaches 99.8, 99.9, 97.2, and 79.5%, respectively. Anwar and Anwar (2020) used DenseNet networks without transfer learning methods to identify four different citrus diseases, and experimental results show that the model can accurately treat citrus diseases, with an accuracy of 92% on the given test dataset. Suh et al. (2018) proposed a transfer learning classifier based on the VGG-19 CNN architecture for the classification of sugar beet and volunteer potato and reported a maximum of 98.7% accuracy for the classification. Maeda-Gutiérrez et al. (2020) classified nine different types of tomato diseases and a healthy class using AlexNet, GoogleNet, InceptionV3, and ResNet18, and the highest recognition rate reached 99.12%. According to these studies, DCNN has higher predictive value and reliability than well-trained humans.

To run the DCNN model on mobile and embedded devices, some scholars have also proposed lightweight networks, which have the advantages of fewer parameters and smaller model size, such as MobileNetV1 (Howard et al., 2017), MobileNetV2 (Sandler et al., 2018), ShuffleNetV1 (Zhang et al., 2018), and ShuffleNetV2 (Ma et al., 2018). Liu et al. (2020) proposed a robust CNN architecture for the classification of six different types of grape leaf disease. This method uses depth-separable convolution instead of standard convolutional layers to reduce model parameters, and the recognition accuracy reached 97.22%. Rahman et al. (2020) proposed a two-stage small CNN architecture named SimpleNet for rice diseases and pest identification with an accuracy of 93.3%. This method is fine-tuned based on VGG16 and InceptionV3 structure to reduce model parameters. The parameters of this network model are less than those of classical CNN models. Tang et al. (2020) identified grape disease image based on improving the ShuffleNet architecture, with an accuracy of 99.14%, similar to the existing CNN models, but the computational complexity is slightly lower. These studies have shown good results, but different diseases have complex morphological characteristics of disease spots at different stages and scales, and the same scale often has similar characteristics, which makes image disease recognition difficult. Therefore, how to fully extract the key information of the local area is the key to improve the performance of disease image recognition. To address these issues, many researchers have focused on attentional features of mechanism-based methods. Li et al. (2020) used the GoogleNet model and embedded SENet attention mechanism to enhance information expression of Solanaceae diseases, with an accuracy rate of 95.09%, and the model size is 14.68 MB, which can be applied to the mobile terminal to identify Solanaceae disease. Mi et al. (2020) proposed a novel deep learning network, namely, C-DenseNet, which embeds convolutional block attention module (CBAM) in the densely connected convolutional network with an accuracy rate of 97.99%. Wang et al. (2021) proposed a novel lightweight model (ECA-SNet) based on Shufflenet-V2 as the backbone network and introduced an effective channel attention strategy to enhance the model's ability to extract fine-grained lesion features with an accuracy rate of 98.86%. Chen et al. (2021) chose the MobileNet-V2 as the backbone network and added the attention mechanism to learn the importance of interchannel relationships and spatial points for input features, and the average accuracy reaches 98.48% for identifying rice plant diseases. In addition, to further improve the performance of feature extraction, some work improves the representation of feature information by integrating multiple-scale features (Liu et al., 2018; Zhang et al., 2020; Pan et al., 2021). Shen et al. (2021) proposed a feature fusion module named adaptive pyramid convolution, which aggregates the features of different depths and scales to suppress the messy information in the background and enhance the feature representation capability of local regions. Sagar (2021) proposed to enhance the dependence between local features and global features by extracting spatial and channel attention features in parallel. Although these methods achieve good results, they can easily increase computational complexity.

Inspired by the above research, we proposed a deep neural network model, namely, group multi-scale attention network (GMA-Net). The main innovations and contributions are summarized as follows:

(1) A rubber leaf disease dataset is established, and the image data augmentation scheme is used to synthesize new images to diversify the image dataset and enhance the anti-interference ability under complex conditions.

(2) The model uses group convolution structure to reduce model parameters and provide multiple branches for multi-scale feature extraction, then embeds dilated convolution to improve the model's adaptability to the scale variability of disease spots, and adds a cross-scale attention feature fusion (CAFF) module to suppress complex background information to strengthen the disease features fusion at different scales.

The rest of this article is organized as follows. The "Materials and methods" section presents the dataset and methods adopted in this study. The "Experimental results and analysis" section presents the experiments for evaluating the performance of the model and analyzes the results of the experiments. Finally, the "Conclusion and future work" section summarizes the main conclusions and future avenues.

# MATERIALS AND METHODS

## Dataset Preparation

### Data Acquisition

The spread of rubber leaf disease is closely related to season, temperature, light, and other factors. For example, powdery mildew disease mainly occurs in spring, and it is more likely to breed disease after rainy days. The rubber leaf disease dataset is created, which included 2,788 rubber leaf samples collected from the rubber tree cultivation farm of Rubber Research Institute, Chinese Academy of Tropical Agricultural Sciences in Danzhou City, Hainan Province, in April 15–20, 2021, and May 13–16, 2021. The types of rubber leaf diseases of these samples were known in advance and labeled according to the domain experts' knowledge. The classification and labeling of different rubber leaf diseases only consider different external visual symptoms, and then image data were captured in the laboratory. Red, green, and blue (RGB) leaf images were taken with the default parameters of the NIKON D90 camera (with a lens Tamron AF 18–200 mm f/3.5–6.3) and iPhone 11 mobile phone. A total of 5 types of image samples of rubber leaves were collected, including four kinds of diseases (i.e., powdery mildew disease, rubber tree anthracnose, periconla leaf spot disease, and abnormal leaf fall disease) and healthy leaves.

Examples of typical symptoms of these rubber leaf diseases are given in **Figure 1**. Healthy rubber leaves appear green, the surface is smooth without disease spots, and the veins are visible. Powdery mildew disease is considered one of the major diseases that threaten the stability of natural rubber production. It spreads rapidly because the pustules can be dispersed for miles on air currents. The lesions initially appear as small, radiating silver-white spots of cobweb-like hyphae scattered on the surface or back of the leaf and then develop to the entire leaf. As the lesion matures, the powdery mildew spots turn into white ringworm-like spots, the surface of the leaves becomes dried and yellow, and finally falls off. The powdery mildew disease can cause high yield losses when severe epidemics occur. Rubber tree anthracnose can appear on stalks, leaves, petioles, tender shoots, or fruits of the rubber tree. The symptoms of this disease begin at the tip and edge of the leaf and can be observed on the leaf as yellow or brown water-stained spots, while as the lesion matures, it becomes irregular, narrow, and gray-white. Periconla leaf spot disease appears as small, dark brown spots scattered on the leaf surface, the tissues at the center of the lesions later decay and become gray to white with black rings at the margin, and the lesions are oval to circular spots, with 0.2–4 cm in diameter. For abnormal leaf fall disease, the small dark brown water-stained spots on the leaf blade may have light brown halos; as the lesions mature, they expand to circular or nearly circular lesions with a diameter of 1–3 mm and turn dark brown near the stalk of the leaf when some of the lesions appeared perforated.

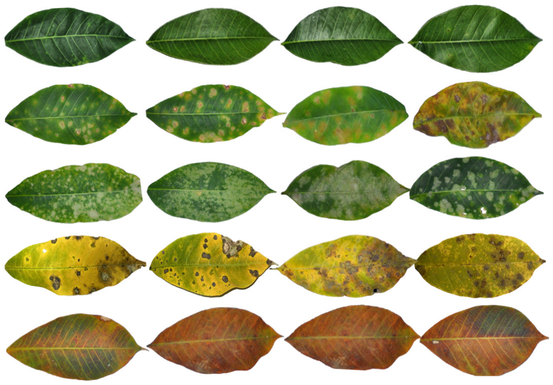


**FIGURE 1 |** Sample images of our constructed rubber dataset, from top to bottom, are healthy leaves, powdery mildew disease, rubber tree anthracnose, periconla leaf spot disease, and abnormal leaf fall disease.

### Data Augmentation

Image preprocessing was carried out on the RGB raw images before image data augmentation, including image scaling, image clipping, and image background removal. Then, the dimensions of the sample images were uniformly resized to 224 × 224 pixels as input to image analysis to reduce the computational cost and improve the image processing efficiency. Our constructed dataset contains 885 images of powdery mildew disease, 829 images of rubber tree anthracnose, 335 images of periconla leaf spot disease, 521 images of abnormal leaf fall disease, and 218 images of healthy leaves. By analyzing the distribution of the number of samples in each category, the dataset we construct is unbalanced. Therefore, the image data enhancement scheme is used to synthesize new images to diversify the image dataset, suppress the impact of unbalanced data, and enhance the anti-interference ability under complex conditions. In this article, based on the Keras' framework, the batch size is set to 32, and brightness adjustment, rotation, scaling, horizontal flip, vertical flip, and other methods are selected to synthesize new images to diversify the image dataset. The specific image augmentation operation is shown in **Table 1**. It should be noted that the data enhancement method adopted in this article will not reduce the size of the image, nor will it change the image's overall color. Finally, the enhanced dataset distribution contains 1,982 images of powdery mildew disease, 2,516 images of rubber tree anthracnose, 2,350 images of periconla leaf spot disease, 2,406 images of abnormal leaf fall disease, and 2,396 images of healthy leaves, and the detailed report of the dataset before and after applying the augmentation process is shown in **Table 2**.

## Architectures of Group Multi-Scale Attention Network Model

### Network Architecture

In this article, a GMA-NET model was proposed for rubber leaf disease image recognition. The architecture of the GMA-Net is illustrated in **Figure 2**. The GMA-Net model includes three parts. The first part is the "pre-network Module" which consists of 3 × 3 convolution layers and max-pooling layers to extract the features of the input image. The second part consists of five cascaded GMA blocks. The GMA block consists of a group multi-scale dilated convolution (GMDC) module and a CAFF module. By utilizing the GMDC module, the network can extract lesion characteristics at different scales and enhance the network's representation ability. After that, the CAFF module is used to fuse the multi-scale attention feature maps from the output of the GMDC module. The last part is composed of a convolution layer, an average pooling layer, a fully connected layer, and a 5-way Softmax layer. Moreover, the batch normalization layer and ReLu activation function are added after each convolution layer. Overall, the proposed method can effectively extract disease feature representation at different scales and aggregate the cross-scale attention feature, which is conducive to fine-grained disease image classification. We detail the different modules of the network, which are summarized in **Table 3**.

**TABLE 1** | Parameter set for data augmentation.

| Technology | Range |
|---|---|
| Rescale the image | 1./255 |
| Rotation_range | 40 |
| Width_shift_range | 0.2 |
| Height_shift_range | 0.2 |
| Fll_mode | "Nearest" |
| Horizontal_flip | True |
| Vertical_flip | True |
| Brightness_range | (0.6, 0.9) |
| Zoom_range | (0.5,0.9) |

**TABLE 2** | Detailed report of the constructed dataset before and after applying the augmentation process.

| Disease name | Class | Images (Raw) | Images (Augmentation) |
|---|---|---|---|
| Healthy leaves | 0 | 218 | 1982 |
| Powdery mildew disease | 1 | 885 | 2516 |
| Rubber tree anthracnose | 2 | 829 | 2350 |
| Periconla leaf spot disease | 3 | 335 | 2406 |
| Abnormal leaf fall disease | 4 | 521 | 2396 |
| Total number | | 2788 | 11650 |

### Group Multi-Scale Dilated Convolution Module

Different diseases have complex symptoms and morphological characteristics at different stages and scales, and the same scale often has similar characteristics. As shown in **Figure 1**, the powdery mildew disease has various symptoms, with some appearing scattered cobweb spots and some appearing mass spots. Identifying this disease needs to consider large-scale coarse-grained features (e.g., the size and texture of the lesion). The characterization information of rubber tree anthracnose is similar to periconla leaf spot disease, with relatively yellowish leaves and scattered spots. Small-scale fine-grained features (e.g., color and texture of the lesion) are the key to recognizing these diseases. Therefore, multi-scale information of rubber leaf disease features in the image plays an essential role in accurately identifying the types of rubber leaf disease.

To address these problems, we design a GMDC module, which consists of a group convolution operation and a multi-scale feature extraction operation. Specifically, the purpose of group convolution operation is to reduce parameters and prevent overfitting. The multi-scale feature extraction operation is used to extract multi-scale disease features.

As shown in **Figure 3**, the group convolution structure consists of four parallel 1 × 1 convolutional layers, followed by batch normalization and ReLU activation functions to accelerate network convergence. Multi-scale feature extraction structure extracts multi-scale information through multiple dilated convolutions with different dilation rates, and then skip connections were used to make full use of the relevant information in the feature map. Dilated convolution (Yu and Koltun, 2014) is defined as follows:

$$\left(F{*}_{l}k\right)\left(p\right)=\sum_{s+lt=p}F\left(s\right)k\left(t\right) \tag{1}$$

where $F$ is a discrete function and $k$ is a discrete filter of size $(2r\ 1)^2$, $\overset{*}{l}$ is called a dilated convolution or a $d$-dilated convolution, $k$ is a 3 × 3 filter, and the kernel dilation rates are 1–4, respectively.

### Cross-Scale Attention Feature Fusion Module

Recently, the attention mechanism has been widely used, including image processing (Li et al., 2020; Tang et al., 2020), speech recognition (Xingyan and Dan, 2018), and natural language processing (Bahdanau et al., 2015). The attention mechanism pays attention to the useful information of various channels of the network, inhibits the useless information, which can enhance the representation of disease features, and effectively improves the identification performance of the model. In this study, as shown in **Figure 4**, a CAFF module was designed to fuse attentional feature maps of different scales.

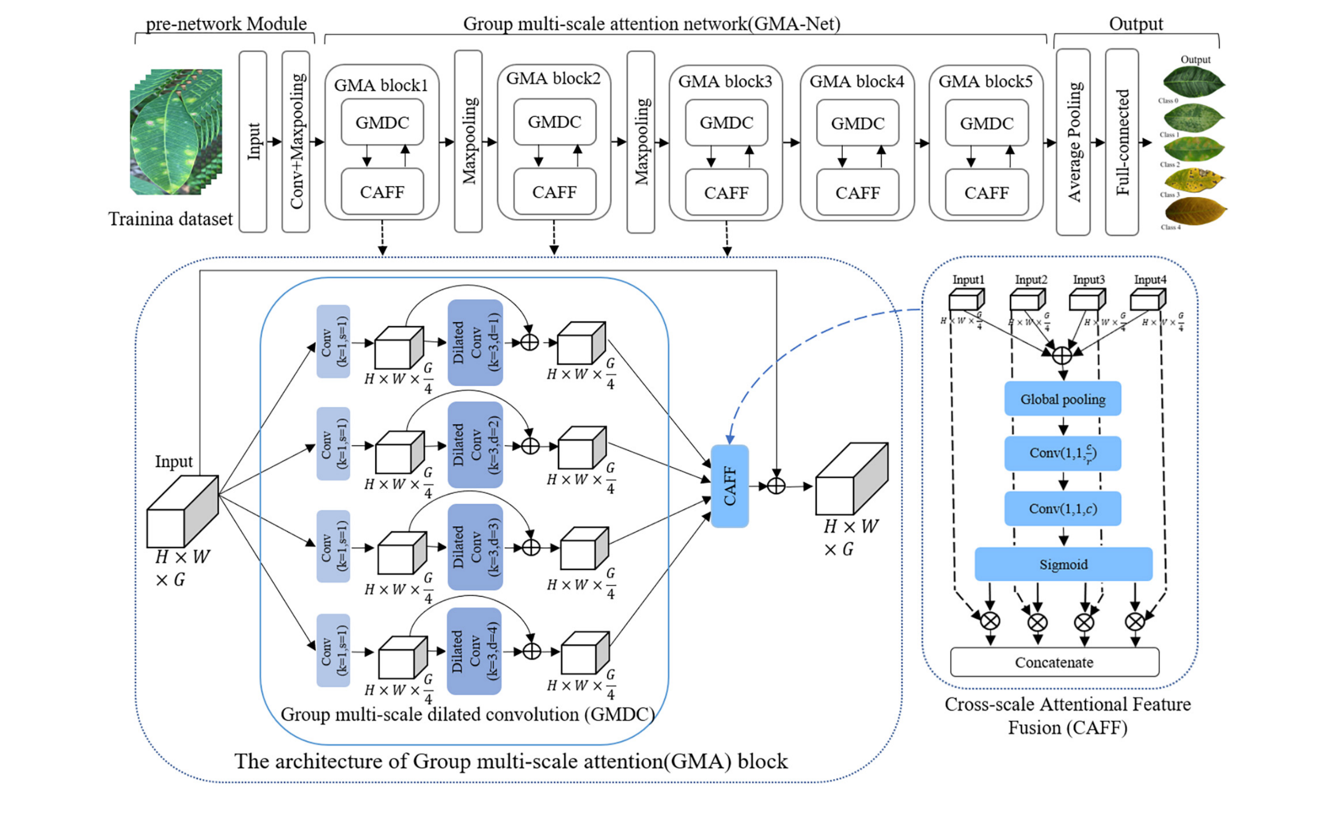


**FIGURE 2 |** The architecture of the proposed group multi-scale attention network (GMA-Net).

**TABLE 3 |** Detailed architectures of the proposed GMA-Net model in our experiments.

| Name | Input | Output | Kernel size | Filter number | Stride |
|---|---|---|---|---|---|
| Input | 224 × 224 × 3 | – | – | – | – |
| Conv | 224 × 224 × 3 | 112 × 112 × 96 | 3 × 3 | 96 | 2 |
| Map | 112 × 112 × 96 | 56 × 56 × 96 | 3 × 3 | – | 2 |
| GMAB 1 | 56 × 56 × 96 | 56 × 56 × 64 | – | 64 | – |
| Map | 56 × 56 × 64 | 28 × 28 × 64 | 3 × 3 | – | 2 |
| GMAB 2 | 28 × 28 × 64 | 28 × 28 × 128 | – | 128 | – |
| Map | 28 × 28 × 128 | 14 × 14 × 128 | 3 × 3 | – | 2 |
| GMAB 3 | 14 × 14 × 128 | 14 × 14 × 192 | – | 192 | – |
| GMAB 4 | 14 × 14 × 192 | 14 × 14 × 208 | – | 208 | – |
| GMAB 5 | 14 × 14 × 208 | 14 × 14 × 256 | – | 256 | – |
| Avg | 14 × 14 × 256 | 2 × 2 × 256 | 7 × 7 | – | 1 |
| Linear | 2 × 2 × 256 | 1 × 1 × 1024 | – | – | – |
| Softmax | 1 × 1 × 1024 | 5 | – | – | – |

First, local feature maps of different scales output by GMF module are added point by point to obtain $F_c$, and then the feature map $F_c$ is compressed into vector Z of 1 × 1 × $C$ by the global average pool layer, which can be expressed as follows:

$$F_c = Add\left[U_1 + U_2 + U_3 + U_4\right] \quad (2)$$

$$Z_c = \frac{1}{W \times H}\sum_{i=1}^{W}\sum_{j=1}^{H} U_c\left(i\ ,\ j\right) \quad (3)$$

Then, the global feature $S$ is obtained through two fully connected layers, one ReLU activation layer, one batch normalization layer, and one sigmoid layer, respectively. $S$ represents the weight coefficient information of different channel features. In this article, 1*1 convolution layer is used instead of fully connected layers to accelerate convergence. The specific formula can be described as:

$$S = \sigma\left(g\left(Z, W\right)\right) = \sigma\left(W_2 \partial\left(W_1 Z\right)\right) \quad (4)$$

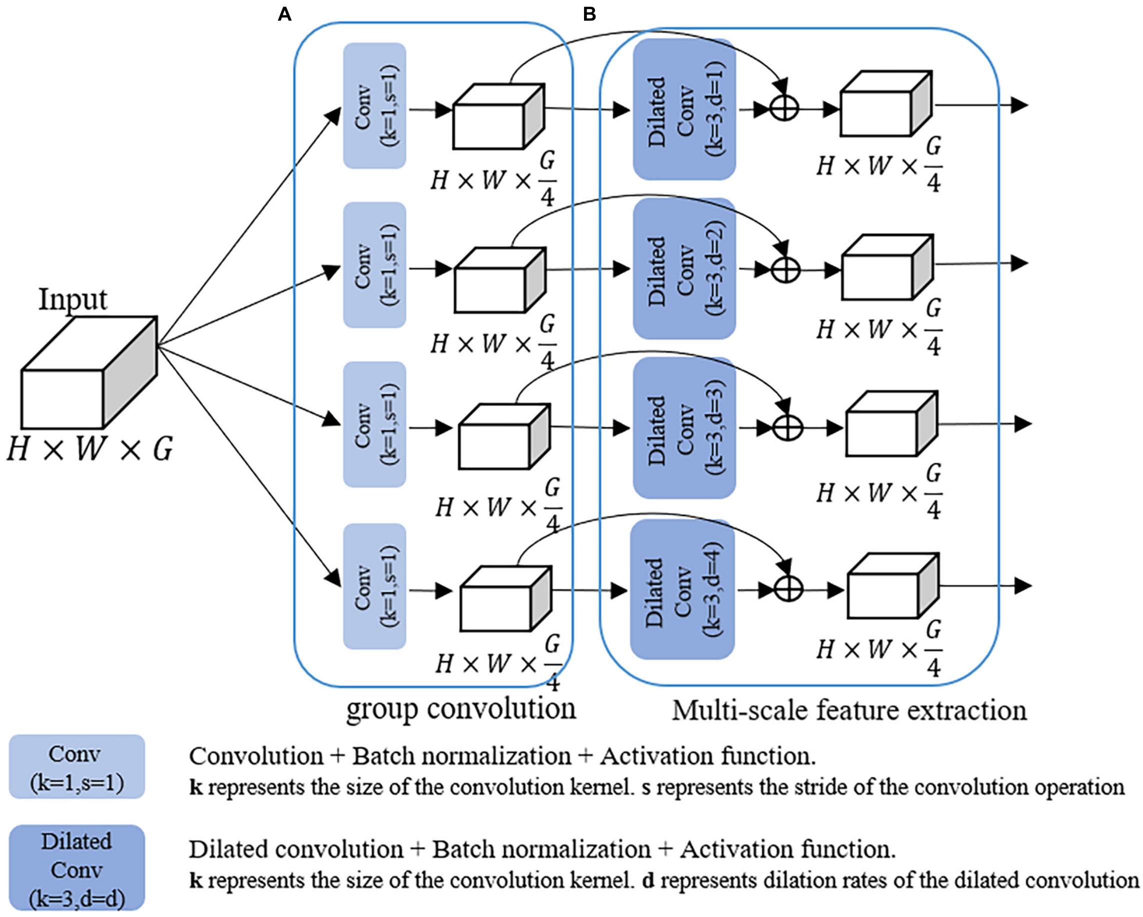

A
B
Input
$H \times W \times G$
Conv (k=1,s=1)
$H \times W \times \frac{G}{4}$
Dilated Conv (k=3,d=1)
Dilated Conv (k=3,d=2)
Dilated Conv (k=3,d=3)
Dilated Conv (k=3,d=4)
group convolution
Multi-scale feature extraction
Conv (k=1,s=1)
Convolution + Batch normalization + Activation function.
k represents the size of the convolution kernel. s represents the stride of the convolution operation
Dilated Conv (k=3,d=d)
Dilated convolution + Batch normalization + Activation function.
k represents the size of the convolution kernel. d represents dilation rates of the dilated convolution

**FIGURE 3 |** The structure of the GMDC module. **(A)** Multi-branch group convolution. **(B)** Multi-branched dilated convolution with different dilation rates.

where σ and ∂ are sigmoid activation function and ReLU activation function, respectively; $W_1 \in R^{\frac{C}{r} \times C}$ and $W_2 \in R^{C \times \frac{C}{r}}$ are dimension reduction and restoration parameters, respectively. $r$ is the reduction factor, which is set to 16 in this article.

The local feature image output by the GMF module is multiplied point by point with vector $S$, which enhances the feature representation information of diseases at different scales in the input feature map and obtain the local attention information $T(x)$ representing different scales. $T(x)$ can be expressed as

$$T\left(U_i\right) = Multiply\left(U_i, S\right) \tag{5}$$

Finally, the local attention information of different scales is connected to generate an effective multi-scale feature descriptor $Y$. $Y$ can be expressed as

$$Y = concat\left[T\left(U_1\right), T\left(U_2\right), T\left(U_3\right), T\left(U_4\right)\right] \tag{6}$$

The CAFF module can fuse attentional feature maps of different scales to enhance disease information, suppress useless information, and improve model performance.

# EXPERIMENTAL RESULTS AND ANALYSIS

## Experimental Configuration and Hyperparameter Setting

Data augmentation and deep learning algorithms are implemented in Keras' deep learning framework based on CNN using python language. The experimental hardware configurations include an Intel i5-10400F CPU (2.90 GHz), a memory of 16 GB, and an RTX 2060S graphics card.

The enhanced rubber disease dataset and PlantVillage (Hughes and Salathe, 2015) public dataset are divided into three groups, namely, the training set (60%), the validation set (20%), and the test set (20%). Comprehensively considering the performance of hardware devices and training effects, the batch size and the number of iterations for all network models are 16 and 40, respectively, and categorical cross-entropy is used to optimize the model. Stochastic gradient descent (SGD) was adopted for training. The initial learning rate is set to 0.1 for the first epoch, and the learning rate is dynamically adjusted by using the Keras' ReduceLROnPlatea function. If the accuracy of the validation set does not

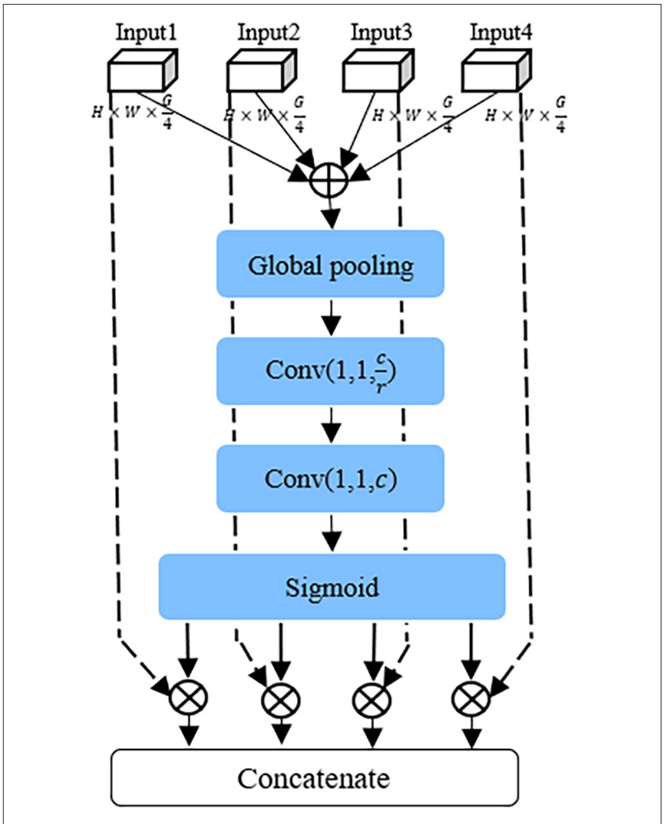

Input1
Input2
Input3
Input4
$H \times W \times \frac{G}{4}$
$H \times W \times \frac{G}{4}$
$H \times W \times \frac{G}{4}$
$H \times W \times \frac{G}{4}$
Global pooling
Conv(1,1,$\frac{c}{r}$)
Conv(1,1,c)
Sigmoid
Concatenate

**FIGURE 4 |** The structure of the cross-scale attention feature fusion module.

improve after three iterations, the learning rate will be reduced by half.

## Evaluation Indexes

In this study, precision, recall, F1-score, accuracy, model size, parameters, and floating-point of operations (FLOPs) are selected as evaluation indexes to evaluate the performance of deep learning algorithms comprehensively:

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1sore = \frac{2TP}{2TP + FP + FN} \tag{9}$$

$$Accuracy = \frac{TP + TN}{TN + TP + FP + FN} \tag{10}$$

where TP, TN, FP, and FN are the number of true positive samples, true negative samples, false-positive samples, and false-negative samples, respectively. Precision estimates how many of the predicted positive samples is positive. The recall is the assessment of how many of all positive samples can be correctly predicted as positive. F1-score is the synthesis of precision and recall. Accuracy measures global sample prediction. Model size, parameters, and FLOPs are commonly used to measure model complexity.

## Performance Comparison Between Different Models

To verify the validity of the GMA-Net model, based on our constructed disease dataset, a comparative experiment was carried out with VGG16, ResNet50, GoogleNet, InceptionV3, and DenseNet121 classical CNN models and MobileNetV1, MobileNetV2, and ShuffleNetv2 lightweight models. Moreover, we trained these models according to the training parameters in the "Experimental configuration and hyperparameter setting" section. **Figure 5** shows the accuracy curve and loss curve of the above eight networks and GMA-Net on the validation dataset. It can be seen from the accuracy curve and loss curve that GMA-Net has the highest

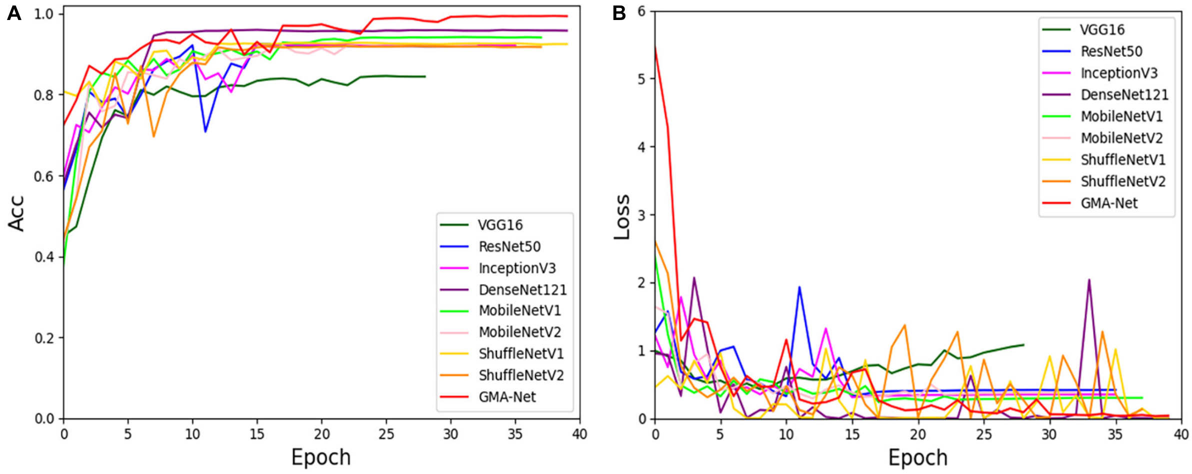


**FIGURE 5 |** Accuracy curve and loss curve of rubber leaf disease validation set. **(A)** Accuracy curve. **(B)** Loss curve.

**TABLE 4** | Comparison of the identification results of different CNN models.

| Models | Precision | Recall | F1_score | Accuracy | Size (MB) | Parameters (M) | FLOPs (M) |
|---|---|---|---|---|---|---|---|
| VGG16 | 85.45 | 85.44 | 85.09 | 84.53 | 1000 | 134 | 268.5 |
| ResNet50 | 93.26 | 93.39 | 93.20 | 92.61 | 180 | 23.6 | 47.1 |
| InceptionV3 | 93.15 | 93.23 | 93.03 | 92.31 | 167 | 21.8 | 43.6 |
| DenseNet121 | 96.61 | 96.67 | 96.55 | 96.01 | 54.6 | 7.04 | 13.9 |
| MobileNetV1 | 93.90 | 93.97 | 93.77 | 94.20 | 24.8 | 3.23 | 6.43 |
| MobileNetV2 | 91.73 | 91.69 | 91.38 | 92.13 | 17.8 | 2.28 | 4.48 |
| ShuffleNetV1 | 93.94 | 93.88 | 93.70 | 92.82 | 16.2 | 1.94 | 3.83 |
| ShuffleNetV2 | 92.22 | 92.14 | 91.84 | 91.96 | 10.5 | 1.28 | 2.52 |
| **GMA-Net** | **97.66** | **97.71** | **97.63** | **98.06** | **5.62** | **0.65** | **1.83** |

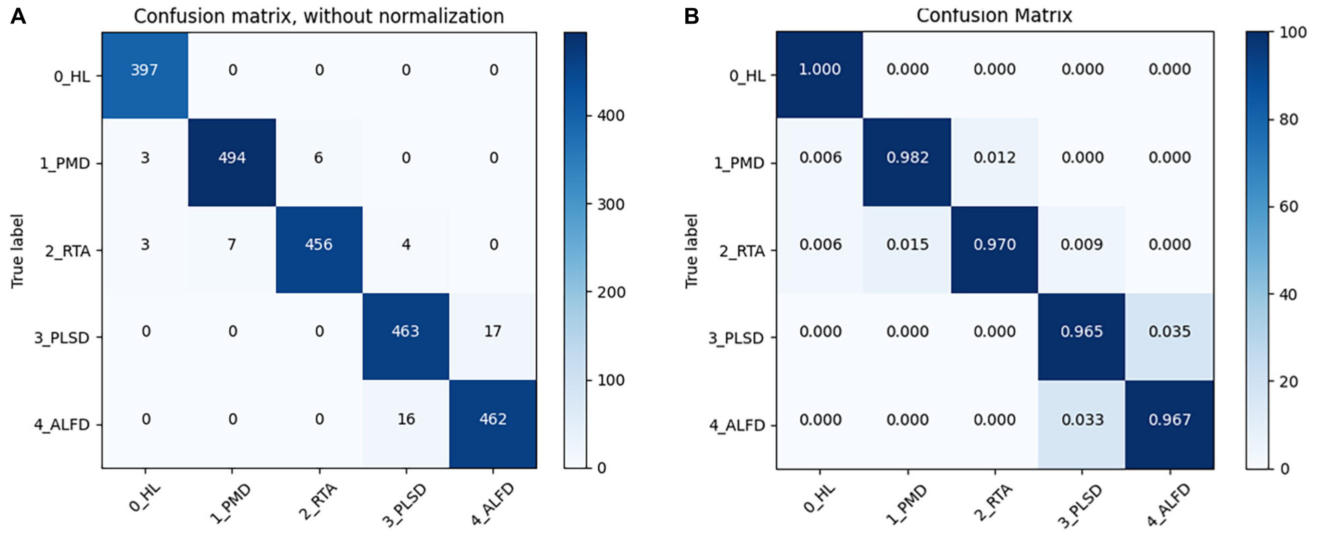


**FIGURE 6** | Confusion matrix of GMA-Net. **(A)** Without normalization. **(B)** Normalized (“Healthy Leaves”: 0, “Powdery Mildew Disease”: 1, “Rubber Tree Anthracnose”: 2, “Periconla Leaf Spot Disease”: 3, “Abnormal Leaf Fall Disease”: 4).

recognition accuracy and quickest convergence rate than other models on the rubber leaf disease dataset, and the model performance is better than the traditional CNN model and lightweight model.

**Table 4** compares the nine networks with the precision, recall, F1-score, accuracy, model size, parameters, and FLOP. The GMA-Net model has the best performance, with an accuracy of 98.06%. Model parameters, size, and FLOPs are 0.65, 5.62, and 1.83 M, respectively. The accuracy of VGG16, ResNet50, InceptionV3, and DenseNet121 models is 84.53, 92.61, 92.31, and 96.01%, respectively. Compared with the classical CNN model, the size and FLOPs of our constructed model are ten times smaller, and the accuracy of the proposed GMA-NET is increased by 13.53, 5.45, 5.75, and 2.05%, respectively. Meanwhile, compared with MobileNetV1, MobileNetV2, ShuffleNetV1, and ShuffleNetV2 lightweight networks. The size, parameters, and FLOPs of the GMA-NET model are not only smaller, but also the model accuracy is improved by 3.86, 5.93, 5.24, and 6.1, respectively.

In general, the GMA-Net model has a relatively small number of parameters and floating-point calculation to obtain better convergence and the highest accuracy of rubber leaf disease among the compared classical CNN model and lightweight model.

In addition, the confusion matrixes are used to summarize the performance of GMA-Net, as shown in **Figure 6**. The diagonals in the matrix are correctly classified, while all other entries are misclassified. It can be seen from the confusion matrix without normalization that 397 healthy leaves (0_HL), 494 Powdery Mildew Disease (1_PMD), 456 Rubber Tree Anthracnose (2_RTA), 463 Periconla Leaf Spot Disease (3_PLSD), and 462 Abnormal Leaf Fall Disease (4_ALFD) were correctly classified, and the number of misclassifications for 0_HL, 1_PMD, 2_RTA, 3_PLSD, and 4_ALFD is 0, 9, 14, 17, and 16, respectively. It can be seen from the confusion matrix that the accuracy of healthy leaves, rubber tree anthracnose, and powdery mildew disease was more than 97%, and the accuracy of periconla leaf spot disease and abnormal leaf fall disease reached 96.5 and 96.7%. Therefore, we can say that it is difficult to distinguish between periconla leaf spot disease and abnormal leaf fall disease classes.

## Ablation Experiment of Model Structure

To determine the final structure of the model, ablation experiments were carried out on the proposed model. We

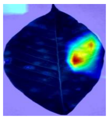
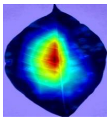
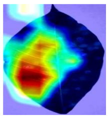
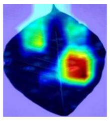
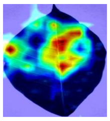
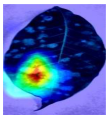
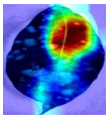
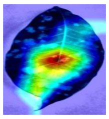
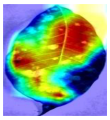
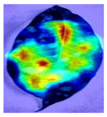
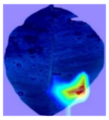
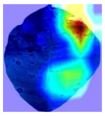
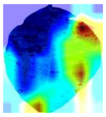
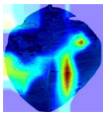
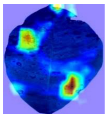
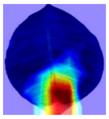
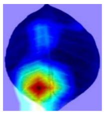
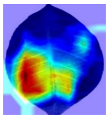
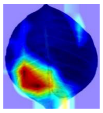
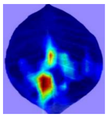
**TABLE 5 |** Classification results of different numbers of GMA blocks.

| Models | Precision | Recall | F1_score | Accuracy | Size (MB) | Parameters (M) | FLOPs (M) |
|---|---|---|---|---|---|---|---|
| GMA-Net-V1 | 96.95 | 96.95 | 96.86 | 97.03 | 2.04 | 0.22 | 0.68 |
| GMA-Net-V2 | 97.37 | 97.36 | 97.29 | 97.51 | 3.53 | 0.39 | 1.29 |
| **GMA-Net-V3** | **97.66** | **97.71** | **97.63** | **98.06** | **5.62** | **0.65** | **1.83** |
| GMA-Net-V4 | 97.50 | 97.49 | 97.42 | 97.46 | 12.3 | 1.52 | 4.67 |
| GMA-Net-V5 | 96.88 | 96.90 | 96.81 | 97.16 | 38.6 | 4.96 | 14.7 |

**TABLE 6 |** Effect of standard and dilated convolution.

| Models | Parameters (M) | F1_score | | | | | Accuracy |
|---|---|---|---|---|---|---|---|
| | | 0_HL | 1_PMD | 2_RTA | 3_PLSD | 4_ALFD | |
| Without dilated convolution | 0.658 | 0.99 | 0.97 | 0.97 | 0.95 | 0.96 | 96.91 |
| Without CAFF | 0.657 | 0.99 | 0.98 | 0.98 | 0.94 | 0.95 | 96.82 |
| Base (With dilated convolution, CAFF) | **0.658** | **0.99** | **0.98** | **0.98** | **0.96** | **0.97** | **98.06** |

**TABLE 7 |** Visualization results of different models.

| Class | Original image | ResNet50 | ShuffleNetV2 | DenseNet121 | MobileNetV2 | GMA-Net |
|---|---|---|---|---|---|---|
| Rubber tree anthracnose | |  |  |  |  |  |
| Powdery mildew disease | |  |  |  |  |  |
| Periconla leaf spot disease | |  |  |  |  |  |
| Abnormal leaf fall disease | |  |  |  |  |  |

only retained the GMA block1 and GMA block2 mentioned in the "Network architecture" section and used them as basic models. Based on the basic model, we designed the following five combinations: GMA-Net-V1 ($N = 1$), GMA-Net-V2 ($N = 2$), GMA-Net-V3 ($N = 3$), GMA-Net-V4 ($N = 4$), and GMA-Net-V5 ($N = 5$) to test the dataset we constructed, where $N$ represents the number of GMA blocks added to the basic model. The specific experimental results are shown in **Table 5**. In the beginning, as the number of cascaded GMAB blocks increases, the accuracy improves. For example, the recognition accuracy of GMA-Net-V1 is 97.03%. The recognition accuracy of GMA-Net V2 is 97.51%, and the GMA-Net V3 has a better effect of 98.06%, which is the highest among all comparison models. However, when the number of cascades of GMA blocks reaches 4 and 5, the accuracy of GMA-Net V4 and GMA-NET-V5 is 0.6 and 1.9% lower than that of GMA-NET-V3, and the model parameters are also improved by 0.87 and 4.31 M. The excessive number of cascaded GMA blocks may cause parameter redundancy, computational resource waste, and precision decline due to overfitting problems. If the number of cascaded GMA blocks is too small, the classification result will be unsatisfactory. In general, the appropriate number of GMAB blocks can effectively improve the accuracy of recognition but do not significantly increase the amount of computation.

## Effect of Dilated Convolution and Cross-Scale Attention Feature Fusion Module

Compared with other deep learning models, this study utilizes multiple dilated convolutions with different dilation rates to extract multi-scale receptive field features and increase the model's adaptability to the scale variability of disease spots. To verify the effect of dilated convolution on classification, all the dilated convolutions were replaced by standard convolutions, and the comparison results are shown in **Table 6**.

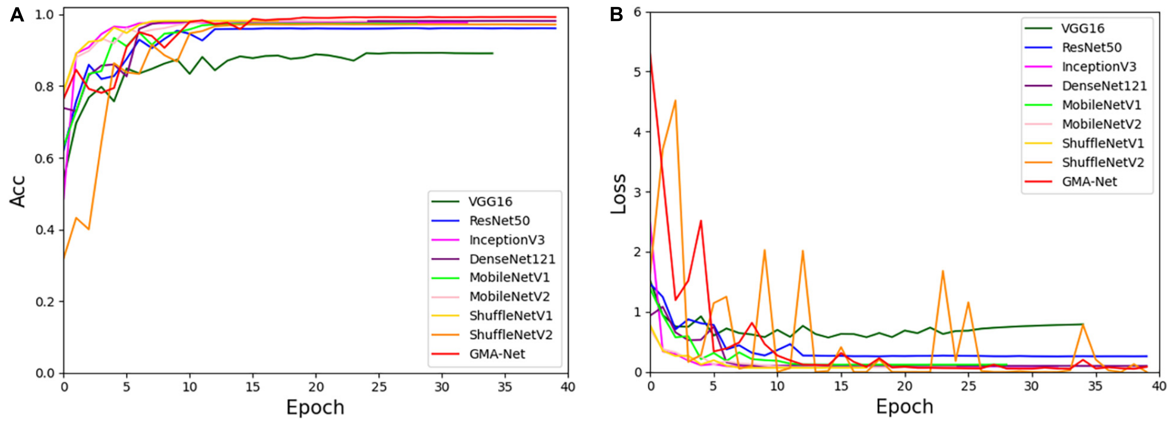


**FIGURE 7 |** Accuracy curve and loss curve of PlantVillage validation set. **(A)** Accuracy curve. **(B)** Loss curve.

**TABLE 8 |** Results of the PlantVillage test set.

| Models | Precision | Recall | F1_score | Top-1 | Top-5 | Parameters (M) | Size (MB) | FLOPs (M) |
|---|---|---|---|---|---|---|---|---|
| VGG16 | 86.53 | 85.43 | 89.18 | 89.25 | 98.99 | 134 | 1000 | 268.5 |
| ResNet50 | 95.35 | 95.21 | 95.87 | 96.14 | 99.23 | 23.6 | 180 | 47.1 |
| InceptionV3 | 96.84 | 96.79 | 97.68 | 97.75 | 99.88 | 21.8 | 167 | 43.6 |
| DenseNet121 | 96.80 | 97.17 | 97.67 | 97.62 | 99.76 | 7.07 | 54.9 | 13.9 |
| MobileNetV1 | 96.78 | 96.82 | 97.57 | 97.45 | 99.83 | 3.27 | 25.1 | 6.43 |
| MobileNetV2 | 97.55 | 97.53 | 98.17 | 98.19 | 99.87 | 2.32 | 18.1 | 4.48 |
| ShuffleNetV1 | 97.26 | 97.47 | 98.01 | 98.18 | 99.96 | 1.99 | 16.6 | 4.21 |
| ShuffleNetV2 | 96.34 | 96.54 | 97.33 | 97.25 | 99.88 | 1.31 | 10.8 | 2.58 |
| GMA-Net | **99.14** | **99.14** | **99.36** | **99.43** | **99.97** | **0.69** | **5.88** | **2.21** |

It can be seen that the recognition accuracy of standard convolution is 96.91%, but after replacing standard convolution with dilated convolution, the accuracy is improved from 96.91 to 98.06%, which improves the recognition accuracy of rubber leaf diseases. The reason why standard convolution shows an inferior performance is that it only samples at a fixed scale, which could not capture the scale variability of disease spots. Dilated convolution contributes to learn multi-scale useful information of disease spots and improves the recognition accuracy of the model.

In addition, we compare the classification accuracy of feature extraction with the CAFF module and without the CAFF module, respectively. It can be seen that the recognition accuracy of models without CAFF module is 96.82%, but when the CAFF module is added, the accuracy increases from 96.82 to 98.06%, which verifies the contribution of the CAFF module in classification. The CAFF module has the advantage of integrating multi-scale attention features, while reducing the influence of complex background in the image, and can provide more discriminative features.

## Visualization Results for Different Models

To better understand the learning capacity of the proposed GMA-Net model, Grad-cam (Selvaraju et al., 2016) was used to display the visualization results of different models, as shown in **Table 7**. The first column is disease class and the second column is the original image, followed by the visualization results of ResNet50, DenseNet121, MobileNetV2, ShuffleNetV2, and GMA-NET model in sequence. The visualization result is composed of the superposition of the rubber leaf disease image and their heatmaps. Heatmaps of ResNet50 and ShuffleNetV2 highlight the local leaf spot area, but the accuracy of heat maps was not high. Heatmaps of DenseNet121 and MobileNetV2 highlight the global leaf spot area but contain a lot of irrelevant background information. Compared with ResNet50 and DenseNet121 benchmark CNN model and MobileNetV2 and ShuffleNetV2 lightweight CNN model, the proposed GMA-NET model can accurately focus on the key areas of rubber leaf spots, with high heatmap accuracy and pays minimum attention to the irrelevant complex background, thus achieving higher disease recognition accuracy than other models.

## Experiment on the Open Dataset

To verify the effectiveness and robustness of the proposed GMA-Net, the PlantVillage public dataset was used for verification. The PlantVillage dataset consists of 54,303 images of healthy and unhealthy leaves, divided into 38 categories by species and disease. According to the training parameters in the "Experimental configuration and hyperparameter setting" section, we divided the PlantVillage dataset into the training set, the validation set, and the test set with 32,571, 10,852, and 10,852 pictures, respectively. Then, based on the PlantVillage dataset, a comparative experiment was carried out with VGG16, ResNet50, InceptionV3, DenseNet121, MobileNetV1, MobileNetV2, ShuffleNetv1, and ShuffleNetv2. **Figure 7** shows the accuracy curve and loss curve of the abovementioned eight networks and GMA-Net on the validation dataset. It can be seen from the accuracy curve that GMA-Net has the highest recognition accuracy than other models, and the loss curve shows that the loss value performed well. The test set accuracy, model size, FLOPs, parameters, top-1 accuracy, and top-5 accuracy of different models on the PlantVillage dataset are shown in **Table 8**.

**Table 8** reports that the top-1 accuracy of VGG16, ResNet50, InceptionV3, DenseNet121, MobileNetV1, MobileNetV2, ShuffleNetv1, and ShuffleNetv2 is 89.25, 96.14, 97.75, 97.62, 97.45, 98.19, 98.01, and 97.25%, respectively. The top-1 accuracy rates of the GMA-Net model are 99.43%, which is the highest of all the models. In addition, the parameters, size, and FLOPs of the GMA-Net model are 0.69, 5.88, and 2.21 M, respectively, which are lower than those of other classical CNN models and lightweight models. In general, the performance of the model on the PlantVillage public dataset shows that the GMA-Net model is efficient and robust, and it is an excellent lightweight CNN network with good performance in the field of crop disease identification.

## CONCLUSION AND FUTURE WORK

In this article, GMA-Net was proposed for rubber leaf disease image recognition. In our method, a GMDC module is responsible for multi-scale feature extraction, including small-scale fine-grained lesion features and large-scale coarse-grained lesion features. In the next phase, the CAFF module is used to fuse attention features of different scales by combining the GMDC module and the CAFF module to build the fine-grained GMA-Net model. To verify the effectiveness and robustness of the model, experiments were conducted on the constructed rubber leaf disease dataset and PlantVillage public dataset and compared with the lightweight and classical CNN models, such as ResNet50, DenseNet121, MobileNetV1, MobileNetV2, ShuffleNetV1, and ShuffleNetV2. The recognition accuracy of the model is 98.06 and 99.43%, which is the highest. In future, we collect more images of different types of rubber leaf diseases and deploy the proposed model on mobile devices.

## 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

TZ designed and performed the experiment, selected the algorithm, analyzed the data, trained the algorithms, and wrote the manuscript. TZ, CL, BZ, and RW collected data. JW monitored the data analysis. WF and XZ conceived the study and participated in its design. All authors contributed to this article and approved the submitted version.

## FUNDING

This work was supported by the Key R&D projects in Hainan Province (Grant No. ZDYF2020042), the National Natural Science Foundation of China (Grant No. 32160424), the Hainan Province Academician Innovation Platform (Grant No. YSPTZX202008), the Key R&D Projects in Hainan Province (Grant No. ZDYF2020195), and the Academician Lan Yubin Innovation Platform of Hainan Province.

## ACKNOWLEDGMENTS

We would like to thank the reviewers for their valuable suggestions on this manuscript.

## SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2022.829479/full#supplementary-material

**Supplementary Figure 1 |** The leaf samples were collected at the rubber tree cultivation farm in Danzhou City, Hainan Province, China. **(A)** Healthy leaves, **(B)** powdery mildew disease, **(C)** rubber tree anthracnose, **(D)** periconla leaf spot disease, and **(E)** abnormal leaf fall disease.

## REFERENCES

Ali, M. F., Aziz, A. A., and Sulong, S. H. (2020). The role of decision support systems in smallholder rubber production: applications, limitations and future directions. *Comput. Electron. Agric.* 173:105442. doi: 10.1016/j.compag.2020.105442

Anagnostis, A., Asiminari, G., Papageorgiou, E., and Bochtis, D. (2020). A convolutional neural networks based method for anthracnose infected walnut tree leaves identification. *Appl. Sci.* 10:469. doi: 10.3390/app10020469

Anwar, T., and Anwar, H. (2020). Citrus plant disease identification using deep learning with multiple transfer learning approaches. *Pakistan J. Eng. Technol.* 3, 34–38.

Bahdanau, D., Cho, K. H., and Bengio, Y. (2015). "Neural machine translation by jointly learning to align and translate," in *Proceedings of the 3rd International Conference Learning Representation ICLR 2015 Conference Track Proceedings*, San Diego, CA, 1–15.

Chen, J., Zhang, D., Zeb, A., and Nanehkaran, Y. A. (2021). Identification of rice plant diseases using lightweight attention networks. *Expert Syst. Appl.* 169:114514. doi: 10.1016/j.eswa.2020.114514

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*. Available online at: http://arxiv.org/abs/1704.04861 (accessed October 24, 2021).

Hu, J., Shen, L., and Sun, G. (2018). "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (New Jersey, NJ: IEEE), 7132–7141.

Hughes, D. P., and Salathe, M. (2015). *An open Access Repository Of Images On Plant Health To Enable The Development Of Mobile Disease Diagnostics*. Available online at: http://arxiv.org/abs/1511.08060 (accessed November 10, 2021).

Li, D., and Zhang, S. (2020). Natural rubber industry development policy analysis:borders and bonus. *Issues For. Econ.* 40, 208–215.

Li, Z., Yang, Y., Li, Y., Guo, R. H., Yang, J., and Yue, J. (2020). A solanaceae disease recognition model based on SE-Inception. *Comput. Electron. Agric.* 178:105792. doi: 10.1016/j.compag.2020.105792

Liu, B., Ding, Z., Tian, L., He, D., Li, S., and Wang, H. (2020). Grape leaf disease identification using improved deep convolutional neural networks. *Front. Plant Sci.* 11:1082. doi: 10.3389/fpls.2020.01082

Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018). *PANet: Path Aggregation Network for Instance Segmentation. (arXiv:1803.01534v3 [cs.CV] UPDATED). Cvpr, 8759–8768*. Available online at: http://arxiv.org/abs/1803.01534 (accessed March 3, 2021).

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., and Alsaadi, F. E. (2017). *NNs Archtectures Review. Elsevier, 1–31*. Available online at: https://www.sciencedirect.com/science/article/pii/S0925231216315533

Ma, N., Zhang, X., Zheng, H. T., and Sun, J. (2018). "Shufflenet V2: practical guidelines for efficient cnn architecture design," in *Proceedings of the Lecture Notes Computer Science (including Subser. Lecture Notes Artificial Intelligence Lecture Notes Bioinformatics*, Vol. 11218, (Cham: Springer), 122–138. doi: 10.1007/978-3-030-01264-9_8

Maeda-Gutiérrez, V., Galván-Tejada, C. E., Zanella-Calzada, L. A., Celaya-Padilla, J. M., Galván-Tejada, J. I., Gamboa-Rosales, H., et al. (2020). Comparison of convolutional neural network architectures for classification of tomato plant diseases. *Appl. Sci.* 10:1245. doi: 10.3390/app10041245

Mi, Z., Zhang, X., Su, J., Han, D., and Su, B. (2020). Wheat stripe rust grading by deep learning with attention mechanism and images from mobile devices. *Front. Plant Sci.* 11:558126. doi: 10.3389/fpls.2020.558126

Mokhtar, U., Ali, M. A. S., Hassenian, A. E., and Hefny, H. (2016). "Tomato leaves diseases detection approach based on support vector machines," in *Proceedings of the 2015 11th International Computer Engineering Conference Today Information Social What's Next?, ICENCO 2015*, Cairo, 246–250. doi: 10.1109/ICENCO.2015.7416356

Pan, X., Xu, J., Pan, Y., Wen, I., Lin, W., Bai, K., et al. (2021). *AFINet: Attentive Feature Integration Networks for Image Classification*. Available online at: http://arxiv.org/abs/2105.04354 (accessed July 20, 2021).

Parikh, A., Raval, M. S., Parmar, C., and Chaudhary, S. (2016). "Disease detection and severity estimation in cotton plant from unconstrained images," in *Proceedings of the 3rd IEEE International Conference Data Science Advance Analytics DSAA 2016*, Montreal, QC, 594–601. doi: 10.1109/DSAA.2016.81

Rahman, C. R., Arko, P. S., Ali, M. E., Iqbal, M. A., Apon, S. H., and Nowrin, F. (2020). Identification and recognition of rice diseases and pests using convolutional neural networks. *Biosyst. Eng.* 194, 112–120. doi: 10.1016/j.biosystemseng.2020.03.020

Ren, C., Kim, D. K., and Jeong, D. (2020). A survey of deep learning in agriculture: techniques and their applications. *J. Inf. Process. Syst.* 16, 1015–1033. doi: 10.3745/JIPS.04.0187

Sagar, A. (2021). *DMSANet: Dual Multi Scale Attention Network*. Available online at: http://arxiv.org/abs/2106.08382 (accessed October 5, 2021).

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L. C. (2018). "MobileNetV2: inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Computer Social Conference Computer Vision Pattern Recognition*, Salt Lake City, UT, 4510–4520. doi: 10.1109/CVPR.2018.00474

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2016). *Grad-Cam: Why Did You Say That? Visual Explanations From Deep Networks Via Gradient-Based Localization. Rev. Do Hosp. Das Cl??Nicas 17, 331–336*. Available online at: http://arxiv.org/abs/1610.02391 (accessed October 10, 2019).

Semary, N. A., Tharwat, A., Elhariri, E., and Hassanien, A. E. (2015). Fruit-based tomato grading system using features fusion and support vector machine. *Adv. Intell. Syst. Comput.* 323, 401–410. doi: 10.1007/978-3-319-11310-4_35

Shen, L., You, L., Peng, B., and Zhang, C. (2021). Group multi-scale attention pyramid network for traffic sign detection. *Neurocomputing* 452, 1–14. doi: 10.1016/j.neucom.2021.04.083

Shin, J., Chang, Y. K., Heung, B., Nguyen-Quang, T., Price, G. W., and Al-Mallahi, A. (2020). Effect of directional augmentation using supervised machine learning technologies: a case study of strawberry powdery mildew detection. *Biosyst. Eng.* 194, 49–60. doi: 10.1016/j.biosystemseng.2020.03.016

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., and Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Comput. Intell. Neurosci.* 2016, 1–11. doi: 10.1155/2016/3289801

Suh, H. K., Ijsselmuiden, J., Hofstee, J. W., and Van Henten, E. J. (2018). Transfer learning for the classification of sugar beet and volunteer potato under field conditions. *Biosyst. Eng.* 174, 50–65. doi: 10.1016/j.biosystemseng.2018.06.017

Tang, Z., Yang, J., Li, Z., and Qi, F. (2020). Grape disease image classification based on lightweight convolution neural networks and channelwise attention. *Comput. Electron. Agric.* 178:105735. doi: 10.1016/j.compag.2020.105735

Wang, P., Niu, T., Mao, Y., Liu, B., Yang, S., He, D., et al. (2021). Fine-grained grape leaf diseases recognition method based on improved lightweight attention network. *Front. Plant Sci.* 12:738042. doi: 10.3389/fpls.2021.738042

Xingyan, L., and Dan, Q. (2018). "Joint bottleneck feature and attention model for speech recognition," in *Proceedings of the ACM International Conference Series*, (New York, NY: Associationfor Computing Machinery), 46–50. doi: 10.1145/3208788.3208798

Yu, F., and Koltun, V. (2014). Multi-scale context aggregation by dilated convolutions. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/1511.07122

Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., et al. (2020). *ResNeSt: Split-Attention Networks*. Available online at: http://arxiv.org/abs/2004.08955 (accessed July 13, 2021).

Zhang, X., Zhou, X., Lin, M., and Sun, J. (2018). "Shufflenet: an extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (New Jersey, NJ: IEEE), 6848–6856.

Zhu, Y., Sun, W., Cao, X., Wang, C., Wu, D., Yang, Y., et al. (2019). TA-CNN: two-way attention models in deep convolutional neural network for plant recognition. *Neurocomputing* 365, 191–200. doi: 10.1016/j.neucom.2019.07.016

**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.

frontiers in Plant Science

ORIGINAL RESEARCH
published: 07 March 2022
doi: 10.3389/fpls.2022.813237

# Analysis of Few-Shot Techniques for Fungal Plant Disease Classification and Evaluation of Clustering Capabilities Over Real Datasets

*Itziar Egusquiza*[1,2]*, *Artzai Picon*[1,2], *Unai Irusta*[2], *Arantza Bereciartua-Perez*[1], *Till Eggers*[3], *Christian Klukas*[3], *Elisabete Aramendi*[2] *and Ramon Navarra-Mestre*[3]

[1] TECNALIA, Basque Research and Technology Alliance (BRTA), Parque Tecnológico de Bizkaia, Derio, Spain, [2] University of the Basque Country, Bilbao, Spain, [3] BASF SE, Limburgerhof, Germany

OPEN ACCESS

**Edited by:**
Hanno Scharr,
Helmholtz Association of German
Research Centres (HZ), Germany

**Reviewed by:**
Sambuddha Ghosal,
Massachusetts Institute of
Technology, United States
Daniel Morris,
Michigan State University,
United States

***Correspondence:**
Itziar Egusquiza
itziar.egusquiza@ehu.eus

**Specialty section:**
This article was submitted to
Technical Advances in Plant Science,
a section of the journal
Frontiers in Plant Science

**Received:** 11 November 2021
**Accepted:** 26 January 2022
**Published:** 07 March 2022

**Citation:**
Egusquiza I, Picon A, Irusta U,
Bereciartua-Perez A, Eggers T,
Klukas C, Aramendi E and
Navarra-Mestre R (2022) Analysis of
Few-Shot Techniques for Fungal Plant
Disease Classification and Evaluation
of Clustering Capabilities Over Real
Datasets. Front. Plant Sci. 13:813237.
doi: 10.3389/fpls.2022.813237

Plant fungal diseases are one of the most important causes of crop yield losses. Therefore, plant disease identification algorithms have been seen as a useful tool to detect them at early stages to mitigate their effects. Although deep-learning based algorithms can achieve high detection accuracies, they require large and manually annotated image datasets that is not always accessible, specially for rare and new diseases. This study focuses on the development of a plant disease detection algorithm and strategy requiring few plant images (Few-shot learning algorithm). We extend previous work by using a novel challenging dataset containing more than 100,000 images. This dataset includes images of leaves, panicles and stems of five different crops (barley, corn, rape seed, rice, and wheat) for a total of 17 different diseases, where each disease is shown at different disease stages. In this study, we propose a deep metric learning based method to extract latent space representations from plant diseases with just few images by means of a Siamese network and triplet loss function. This enhances previous methods that require a support dataset containing a high number of annotated images to perform metric learning and few-shot classification. The proposed method was compared over a traditional network that was trained with the cross-entropy loss function. Exhaustive experiments have been performed for validating and measuring the benefits of metric learning techniques over classical methods. Results show that the features extracted by the metric learning based approach present better discriminative and clustering properties. Davis-Bouldin index and Silhouette score values have shown that triplet loss network improves the clustering properties with respect to the categorical-cross entropy loss. Overall, triplet loss approach improves the DB index value by 22.7% and Silhouette score value by 166.7% compared to the categorical cross-entropy loss model. Moreover, the F-score parameter obtained from the Siamese network with the triplet loss performs better than classical approaches when there are few images for training, obtaining a 6% improvement in the F-score mean value. Siamese networks with triplet loss have improved the ability to learn different plant diseases using few images of each class. These networks based on metric learning techniques improve clustering and classification results over traditional categorical cross-entropy loss networks for plant disease identification.

**Keywords: plant disease, convolutional neural network, triplet loss, categorical cross-entropy loss, few-shot learning**

## 1. INTRODUCTION

Plants are vulnerable to attack by organisms that interrupt or modify their physiological processes, disrupting plant growth, their development or their vital functions, thus causing plant disease. Plant diseases have a significant impact in agriculture, producing crop yield losses, impairing product quality or limiting availability of food and raw materials. Estimations of global productivity losses are between 20 and 40% annually, and up to 16% of the losses are due to plant diseases (Oerke, 2006). Therefore, plant disease management is essential to reduce crop losses caused by pathogens. Diseases are mainly controlled using chemical fungicides, which in most cases are very efficient (Hirooka and Ishii, 2013). On the other hand, manual plant disease identification is expensive and time-consuming, as it involves human experts to ensure a correct diagnosis. Consequently, automatic plant disease classification algorithms have become a very important and active field of research in agriculture (Sandhu and Kaur, 2019; Shruthi et al., 2019).

Over the years classical computer vision techniques have been widely used for automatic plant disease classification. For instance, Kim et al. (2009) classified grapefruit peel disease using color texture feature analysis under laboratory conditions. Camargo and Smith (2009) developed an image-processing algorithm to identify visual symptoms of plant disease. Revathi and Hemalatha (2012) used edge detection technique to classify cotton leaf diseases. Sannakki et al. (2011) analyzed image color information to predict disease grade on plant leaves. Johannes et al. (2017) developed a early symptom wheat disease diagnosis algorithm for mobile capture devices. Several other algorithms have been developed for different crops, such as rice (Phadikar et al., 2012, 2013), corn (Kiratiratanapruk and Sinthupinyo, 2011), or potato (Dacal-Nieto et al., 2009).

Traditional computer vision approaches depend on the domain knowledge of experts who draft the relevant features for the classification task. This becomes overly complex as the number of crops and diseases increases, compromising the generalizability of the models. This is one of the reasons why Deep Learning (DL) models have replaced classical computer vision techniques in image classification (Picon et al., 2019a) or segmentation (Lin et al., 2019). DL models are frequently based on Convolutional Neural Networks (CNNs). CNNs automatically select most descriptive and salient features for the classification task, and have thousands of adjustable parameters to address complex classification tasks. Thus, CNNs have also been introduced for image based plant disease classification. For example, Sladojevic et al. (2016) created a leaf image classification algorithm to recognize 13 diseases which was also able to differentiate plant leaves from their surroundings. Ferentinos (2018) developed a classification algorithm to distinguish 58 plant species using an open database of more than 85,000 images. Picon et al. (2019a) and Johannes et al. (2017) extended by creating an early disease detection algorithm based on a DL model. Fuentes et al. (2017) presented a DL tomato plant disease and pest detector. There are various recent excellent reviews of DL for plant image classification (Saleem et al., 2019; Hasan et al., 2020; Li et al., 2021).

The importance of plant disease identification algorithms has led to the creation of open access agronomic datasets, for use by agronomists and artificial intelligence researchers as a benchmark to experiment and evaluate new techniques. A salient example is PlantVillage (Hughes and Salathé, 2015), an open access repository of over 50,000 expertly curated images of healthy and infected plant leaves acquired in laboratory conditions. The PlantVillage images include 14 crop types (e.g., cherry, corn, grape, tomato, and pepper) and 26 diseases. Many DL plant disease classification models have been developed and evaluated using the PlantVillage dataset (Mohanty et al., 2016; Rangarajan et al., 2018; Kamal et al., 2019; Too et al., 2019; Argüeso et al., 2020; Mohameth et al., 2020). However, images obtained under laboratory conditions have controlled lighting, smooth backgrounds, and diseases are at an advanced stage of infection. In the field, illumination conditions are uncontrolled, backgrounds are changeable and diseases appear at different stages, including early stages. Early stage disease detection in these challenging conditions is of outmost importance since proper treatment could cure the damage to the crop, but at an early infection stage healthy and diseased plant images are visually very similar. These reasons explain why algorithms developed using the PlantVillage dataset with accuracies over 99% (Mohanty et al., 2016), present accuracies as low as 31.4% when tested on real field images.

In Ghosal et al. (2018), an explainable deep CNN framework was developed to identify, classify and quantify biotic and abiotic stresses in soybean. This framework uses an unsupervised approach to accurately isolate visual symptoms without the need for detailed expert annotation. They identify and classify eight biotic and abiotic soybean stresses by learning from over 25,000 images. Thanks to the application of explainability techniques, they are able to understand the classification decisions made by extracting the visual features learned by the model based on their localized activation levels. These characteristics are then compared with the symptoms identified by humans to validate the results. Another study (Toda and Okura, 2019) also developed a variety of visualization methods using a CNN to understand the network mechanism for disease diagnosis. The attention maps generated by the network found the most significant regions of stressed lesions, matching human decision-making to determine disease. These two studies demonstrated the importance of understanding the mechanism of CNNs for plant stress phenotyping. DeChant et al. (2017) trained several CNNs to classify small regions of maize images as containing northern leaf light (NLB) lesions or not using a sliding window over the images. Predictions from all CNNs were combined into separate heat maps and then fed into a final CNN for stressed lesion detection. The generated heat maps were used as a visualization mechanism to explain classification decisions.

Another limitation of DL models is the need for large annotated datasets to adjust their millions of adjustable parameters. Compiling real field images with disease annotations is very resource consuming, so many efforts are focused on learning from few images, a set of techniques known as Few Shot Learning (FSL). FSL methods for image classification are divided into three main types: data augmentation, transfer-learning, and meta-learning. Data augmentation consists in generating new instances from previous images, for instance using generative adversarial networks (Hu et al., 2019). In transfer learning a baseline network is trained with a large number of images other than the target classes, and then the network is fine-tuned using few instances of the target classes. Typical learning architectures are based on siamese networks and metric learning (distances among classes), as proposed in Argüeso et al. (2020) for plant disease classification. Finally, in meta-learning the models are trained in a set of related prediction tasks, as described by Li and Yang (2021) for plant and pest image classification. Nazki et al. (2020) generated synthetic images to train CNN models for tomato plant disease classification based on generative adversarial networks (GANs). Their model, called AR-GAN, was based on Cycle-GAN (Zhu et al., 2017) and was developed to transform healthy tomato leaves into different types of diseases. They claimed that their technique could improve the performance of plant disease classification compared to other classical data augmentation techniques. AR-GAN was trained on images without complex backgrounds, so this approach might present difficulties in transforming images from datasets taken in the real field.

The objective of this study is to demonstrate a FSL approach based on siamese networks and metric learning trained and evaluated in the challenging conditions of real field images. For that purpose a dataset of real field images containing 5 crops and 17 diseases was used, and experiments were conducted to evaluate how small a dataset could be used to obtain acceptable classification results. Our results show that FSL methods based on siamese networks outperform classical CNN learning methods when trained with less than 200 images per class.

## 2. MATERIALS

The study dataset was compiled in the 2014–2019 period in three phases and at different farmlands in Germany and Spain, as described in Johannes et al. (2017), Picon et al. (2019a), and Picon et al. (2019b). Images were acquired using different electronic devices (e.g., iPhone4, iPhone5, Samsung Galaxy Note, and Windows Phone) throughout the growing season to capture different growth stages of infection.

The dataset is composed of 121,955 images of plant leaves, stems and panicles that have been taken by cell phone in real

**TABLE 1 |** Diseases and number of images from the annotated dataset.

| Crop | Disease | Images | Crop | Disease | Images |
|---|---|---|---|---|---|
| Wheat | Healthy | 6,704 | Rice | Healthy | 4,051 |
| Wheat | Septoria tritici | 18,841 | Rice | Various diseases | 206 |
| Wheat | Puccinia striiformis | 15,376 | Rice | Thanatephorus cucumeris | 2,438 |
| Wheat | Puccinia recondita | 16,413 | Rice | Pyricularia oryzae | 2,441 |
| Wheat | Septoria nodorum | 602 | | | |
| Wheat | Drechslera tritici-repentis | 9,550 | **Total rice:** | | **11,295** |
| Wheat | Oculimacula yallundae | 1,489 | | | |
| Wheat | Gibberella zeae | 1,207 | Corn | Healthy | 206 |
| Wheat | Blumeria graminis | 2,866 | Corn | Helminthosporium turcicum | 425 |
| **Total wheat:** | | **64,026** | **Total corn:** | | **631** |
| Barley | Healthy | 1,624 | Rape seed | Healthy | 6,850 |
| Barley | Pyrenophora teres | 15,352 | Rape seed | Phoma lingam | 6,924 |
| Barley | Ramularia collo-cygni | 3,441 | | | |
| Barley | Rhynchosporium secalis | 11,279 | **Total rape seed:** | | **13,774** |
| Barley | Puccinia hordei | 3,323 | | | |
| **Total barley:** | | **32,229** | | | |
| **TOTAL**: | | | | | **121,955** |

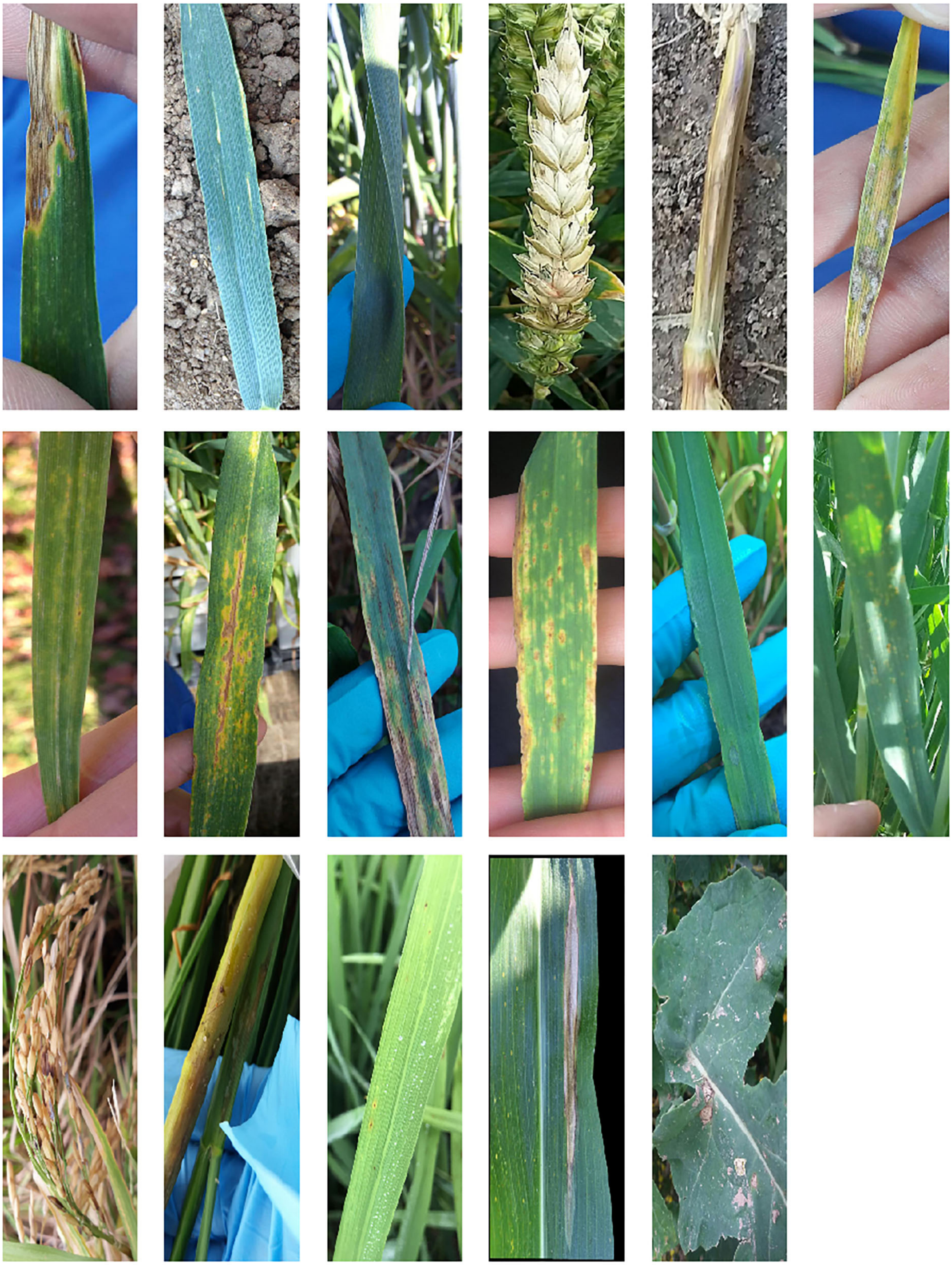


**FIGURE 1 |** Examples from the 17 diseases in the generated dataset, ordered from left to right by crops: wheat (*Septoria tritici, Puccinia striiformis, Puccinia recondita, Gibberella zeae, Oculimacula yallundae, Blumeria graminis, Septoria nodorum*, and *Drechslera tritici-repentis*), barley (*Pyrenophora teres, Ramularia collo-cygni, Rhynchosporium secalis*, and *Puccinia hordei*), rice (*Various diseases, Thanatephorus cucumeris*, and *Pyricularia oryzae*), corn (*Helminthosporium turcicum*), and rape seed (*Phoma lingam*).

field conditions (Picon et al., 2019b). It contains five types of crops: wheat, barley, rice, corn and rape seed. And in those crops there are 17 representative diseases, including: Rust, Septoria, Tan Spot, Eyespot, Scab, Powdry mildew, Net Blotch, Scald, Blast, Lef blight, or Blackleg. **Table 1** provides a detailed composition of the dataset in terms of number of images per crop and disease (causing fungi), and **Figure 1** shows an example of each disease.

The automatic classification of the images in the dataset is complex. Besides the differing illumination and acquisition conditions, the dataset presents several diseases at early stages which are very hard to differentiate, for example *Puccinia recondita* and *Puccinia striiformis* in wheat. Some other diseases present similar symptoms in both early and late stages of infection, like *Septoria tritici* and *Septoria nodorum* in wheat. **Figure 2** shows examples illustrating the similarities between those diseases. Moreover, in 9,923 images the crop was infected with various diseases.

This study focuses on single label classification, so images with multiple diseases were discarded. Then the dataset was split into a training (80%), a validation (10%), and a test set (10%). Experiments were conducted with a decreasing number of images per class during training, and once the models were trained the results were obtained for the complete test set. For the experiments, images of different resolution and size have been considered, as different devices have been used to acquire the images.

## 3. METHODS

In this section, the architecture used for the plant disease classification algorithm is presented. In order to compare the benefits of metric learning techniques over classical techniques, this architecture is composed of two parts as shown in **Figure 3**. First, images of leaves, stems or panicles of the specified plant species are used as input to the convolutional neural network (CNN), which is trained to extract features from the images by representing them with an embedding vector. A ResNet-50 (He et al., 2016) neural network has been selected as backbone. To analyze the quality of the generated latent spaces and to obtain the class predictions, a k-nearest neighbors classifier is then used as the shallow classifier, which is fed with the embedding vectors to learn to distinguish the different classes referring to plant disease and providing the final output value of the algorithm. The *k*-nn classifier is a non-parametric method that only depends on the quality of the features and does not interfere with any additional parameters, which has made it a common practice (Wu et al., 2018; Caron et al., 2021).

The idea of this work is to demonstrate that distance metric learning techniques achieve a better vector representation of the images than classical methods when a small dataset is used. We have focused on cases where the supporting dataset, which is often used to train the baseline models before applying the few-shot technique, is not available. Therefore, in this work, models capable of learning classes from a few samples have been developed using a metric learning loss function (triplet loss) and compared with a traditional loss function (cross-entropy loss). Several experiments have been developed using different numbers of training images per class (from $N = 4$ to $N = 2{,}000$) to evaluate how the network learns with few samples, where two approaches have been compared. On the one hand, a Siamese network with three sub-networks and the Triplet loss function was used to test the metric learning techniques. On the other hand, a traditional single network and the Categorical cross-entropy loss function were used. In both cases the networks worked as feature extractors and then a *k*-NN classifier was added to learn the feature map representations and convert them into class predictions.

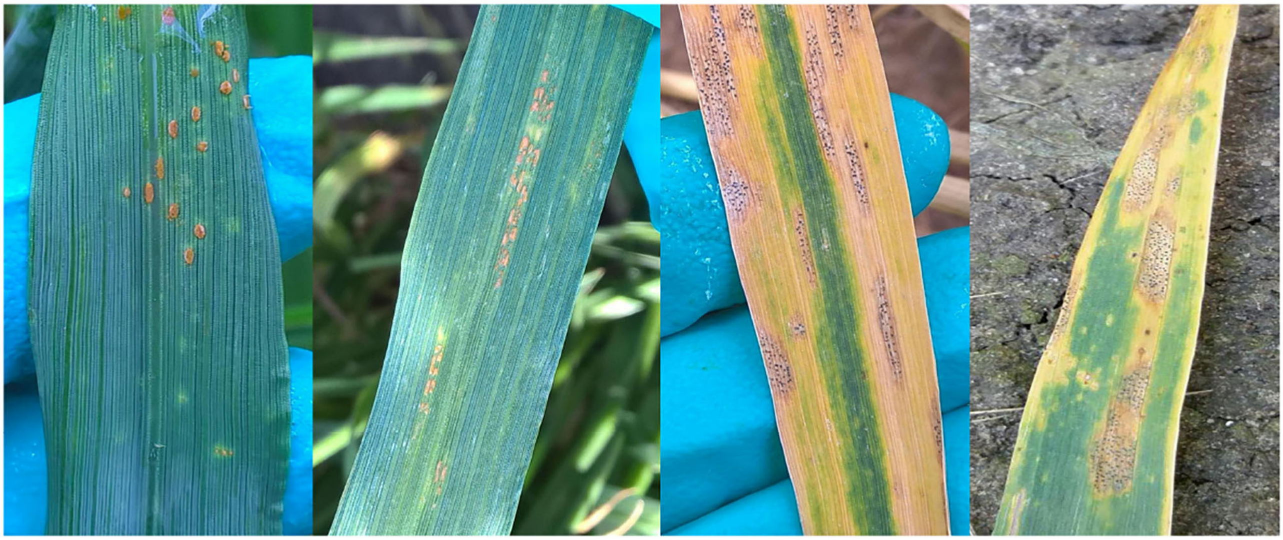


**FIGURE 2 |** Examples of similar diseases. On the one hand, from left to right: *Puccinia Recondita* and *Puccinia Striiformis*. On the other hand, *Septoria tritici* and *Septoria nodorum*.

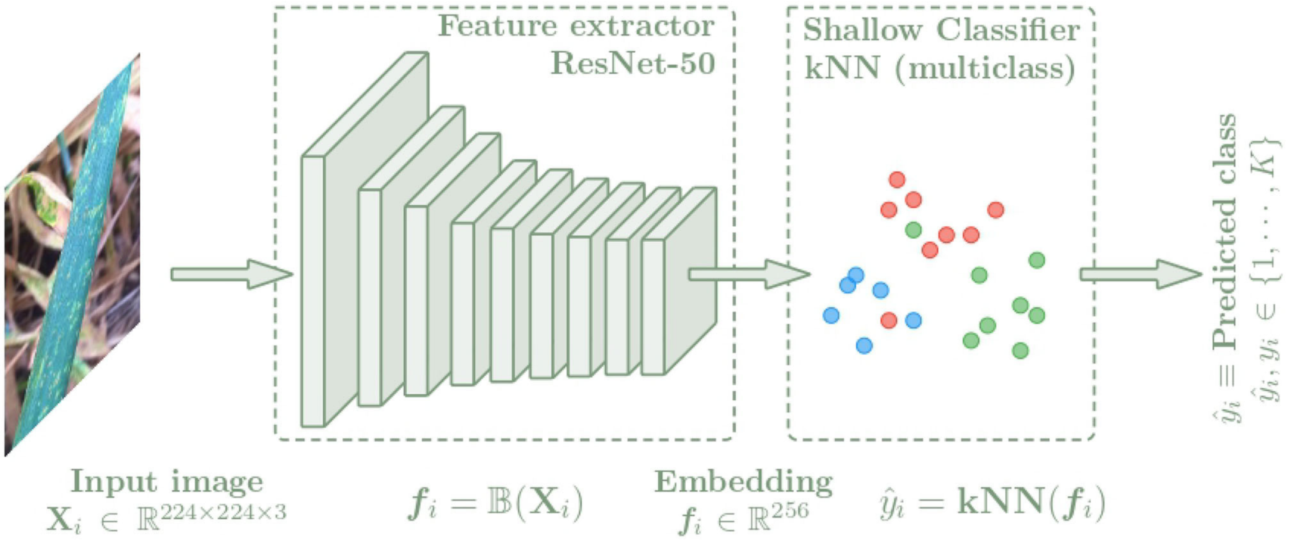

Feature extractor
ResNet-50
Shallow Classifier
kNN (multiclass)
Input image
$\mathbf{X}_i \in \mathbb{R}^{224\times224\times3}$
$\boldsymbol{f}_i = \mathbb{B}(\mathbf{X}_i)$
Embedding
$\boldsymbol{f}_i \in \mathbb{R}^{256}$
$\hat{y}_i = \mathbf{kNN}(\boldsymbol{f}_i)$
$\hat{y}_i \equiv$ Predicted class
$\hat{y}_i, y_i \in \{1, \cdots, K\}$

**FIGURE 3 |** Architecture of the plant disease classification algorithm separated into two blocks. First, a CNN is used to extract features from the input images $X_i$ getting an embedding vector $f_i = f(X_i)$. Then a $k$-NN classifier is trained with the $f_i$ embeddings to predict the class of each image.

## 3.1. Baseline CNN

In our method, the ResNet-50 convolution neural network has been used as the base model and adapted as described in Picon et al. (2019a) to identify diseases from a leaf centered image. This network falls into the subgroup of Residual Networks (*ResNets*) where the main idea is to skip convolutional layer blocks by using shortcut connections. ResNet implements, on the one hand identity blocks that have no convolution layer at shortcut, and do not change dimensions of the feature map, and on the other hand, convolution blocks, which add convolution layer at shortcut, thus increasing the output dimensions with respect to the input. In both cases, batch normalization is performed after each convolution, and then, ReLU activation is applied.

The ResNet-50 consists of 50 layers, with more than 23 million of tuning parameters. It is trained on more than a million images from the ImageNet database (Deng et al., 2009), from which meaningful feature representations have been learnt. In our experiments, the last 33 layers have been unfrozen to adjust the weights to our classification case. After the last layer, a global average pooling operation has been implemented to obtain an image representation of 2,048 features, which has then been reduced to 256 features by adding a neuron layer. Each experiment with the ResNet-50 backbone has been made with two different function losses: triplet loss and categorical cross-entropy loss. Finally, a $k$-NN classifier has been added to the baseline CNN to predict the final class values of the feature embeddings.

## 3.2. Loss Functions

Neural network models use loss functions to calculate the error of the model at each iteration. The algorithm then updates the weights so that the next iteration reduces the previous error, with the aim of minimizing it by means of Stochastic Gradient descent algorithm together with back-propagation. In this sense, the loss function is a fundamental part of a neural network training as it is the mathematical function that guides the training goal for the network.

Image classification neural networks such as ResNet50 are normally used with a cross-entropy loss function which leads to appropriate classification. However, when few samples are available, this loss tends to generate unreliable latent representations (Argüeso et al., 2020). However, metric learning losses aim to learn feature embeddings from images by applying distance metrics to ensure intra-class compactness and inter-class separability. These embeddings keep the most significant features related to the corresponding class of each image, and the loss function tries to increase the distance between samples of different classes while keeping samples of the same class close together.

In this paper, traditional categorical cross-entropy loss and distance metric-based triplet loss have been used and compared.

### 3.2.1. Categorical Cross-Entropy Loss

Categorical cross-entropy is a common loss function used to solve multi-class classification tasks. The block diagram used with the Categorical cross-entropy loss is shown in **Figure 4**. This loss function is designed to quantify the difference between two probability distributions. This loss function calculates the loss of an example by applying the following equation:

$$\mathcal{L}_c = -\sum_i y_i \cdot \log \widehat{y_i}. \tag{1}$$

where $y_i$ is the $i$-th real value and $\widehat{y_i}$ the predicted value by the algorithm. The minus sign ensures that the loss is reduced as the distributions approach each other.

This loss function, together with the architecture defined in Section 3.1 and adapted to identify diseases from a leaf centered

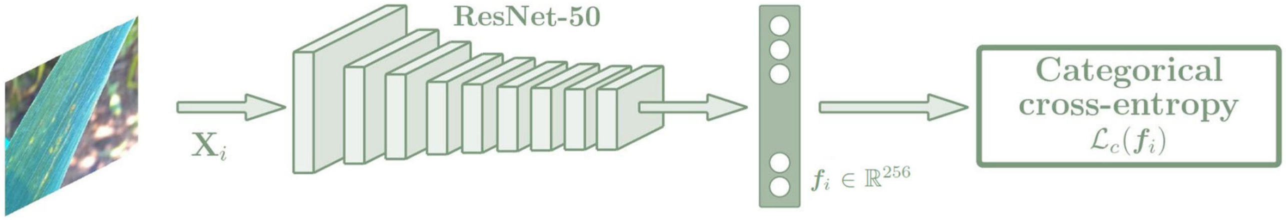


**FIGURE 4 |** Architecture of the CNN based on the Categorical cross-entropy loss. The network is trained with the input images $X_i$, and the output is an embedding vector $f_i$ of size 256. Then a *k*-NN classifier is trained with the $f_i$ embeddings to predict the class of each image, which is trained in the same way for the triple and categorical models.

image as it was described in Picon et al. (2019a) will serve as baseline model to compare with the metric learning approach described below.

### 3.2.2. Triplet Loss

Triplet loss function can be used to make the embedding representation more easily separable between classes in a Euclidean vector space. The triplet loss function is used to adjust the network parameters in order to minimize the distance between feature embeddings of the same class, and to maximize the distance between embeddings of different classes at the same time. For this purpose, a Siamese network with three sub-networks is used, where all sub-networks share the same weights and are joined by the triplet loss function (**Figure 5**). During training, three images of different plants are chosen, which are an anchor ($x_a$), a positive sample ($x_p$), and a negative sample ($x_n$), and each of them is introduced into one of the three sub-networks. In all cases, the anchor and the positive sample belong to the same class while the negative sample belongs to a different class. Image embedding vectors representing the most important features associated with the image class are created as output. The networks compute the distance between the three embedding vectors using the triplet loss function, which is calculated as a Euclidean distance function (Equation 2). Then the parameters of the networks are adjusted to minimize the distance between the embeddings of the anchor and the positive sample, while maximizing the distance between the anchor and the negative sample.

$$\mathcal{L}_t(x_a, x_p, x_n) = \max(\|f_a - f_p\|^2 - \|f_a - f_n\|^2 + \alpha, 0). \quad (2)$$

where $\| \cdot \|^2$ represents the Euclidean distance and $\alpha$ is a margin between positive and negative pairs, which is used to avoid wasting effort on extending the distance of a negative pair that is distant enough and to focus on more difficult pairs.

## 3.3. *k*-NN Classifier

In order to quantify the classification performance for both approaches, a *k*-nn classifier has been used as a shallow classifier after the feature extraction applied by the neural network. The classifier has been trained using the embedding vectors of the training set for each of the experiments with different neighbor values ($K$). The obtained knn models were used over the embedding vectors from the validation set to select the best $K$-value. This best $K$-value over the validation set was used to quantify the performance of the model in the testing set for class discrimination showing as output one of the 17 plant diseases being analyzed.

In the *k*-nn classification, the output value is selected by a plural vote of its neighbors. Thus, the output is assigned to the most common class among its *k* nearest neighbors, where *k* is a constant value defined by the user, and from which the prediction changes. Therefore, it is important to find the optimal *k*-value. The nearest neighbors can be found using different distance metrics; in this project the Euclidean distance has been applied.

## 3.4. Few-Shot Learning

To demonstrate that neural networks can achieve good results for image classification approaches when a large dataset is not available, several experiments were conducted using different numbers of training images per class, which were randomly selected and ranged from $N = 4$ to $N = 2,000$. The image features were obtained using a pre-trained network and fine-tuning some layers from the back of the backbone to adjust the weights to the dataset.

In addition, the Triplet and Categorical cross-entropy loss functions were applied separately to create the few shot learning models in order to compare the embedding vectors obtained with the application of each error function. The triplet loss, which is used for distance metric learning, is considered to better learn the embedding representation by keeping objects of the same class close and increasing the distance for objects of different classes.

In all experiments, data augmentation techniques were applied to the training images. Rotations (probability = 0.5, ±90° rotation range), translations (probability = 0.5, ±10%), scaling (probability = 0.7, scale ranges from 50 to 150%) and gamma transformation (probability = 0.5, gamma limits from 80 to 120) were selected. The experiments were conducted over 150 epochs and a learning rate of $\alpha = 10^{-4}$ was selected with the Adam optimizer. The number of training images per class was randomly selected and ranged from $N = 4$ to $N = 2,000$, and all the experiments were run identically for the triplet loss-based model as for the categorical cross-entropy-based model.

## 3.5. Evaluation

The models we have created refer to a multiclass classification problem. Three metrics widely adopted by the scientific

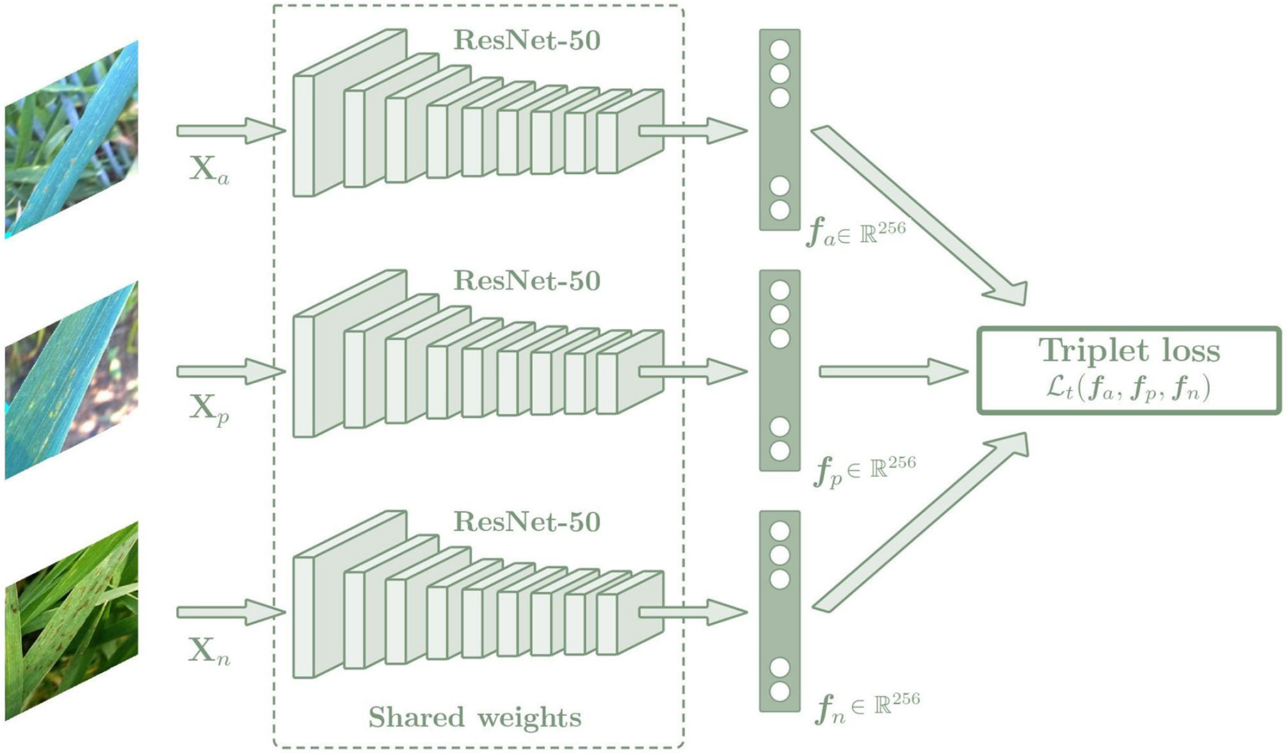

ResNet-50
ResNet-50
ResNet-50
$\mathbf{X}_a$
$\mathbf{X}_p$
$\mathbf{X}_n$
Shared weights
$f_a \in \mathbb{R}^{256}$
$f_p \in \mathbb{R}^{256}$
$f_n \in \mathbb{R}^{256}$
Triplet loss
$\mathcal{L}_t(f_a, f_p, f_n)$

**FIGURE 5 |** Architecture of the Siamese network based on the Triplet loss. The Siamese network is composed of three sub-networks that share the same weights. The images introduced by these networks must always maintain the same relationship: two of them must belong to the same class, which are the anchor and positive images, and the last one must belong to a different class, which is the negative image. In this way, the Siamese network is trained to minimize the distance between the embeddings of the same class (anchor and positive sample), while maximizing the distance between the embeddings of different classes (anchor and negative sample. The output of the network is an embedding vector $f_i$ of size 256. Then a *k*-NN classifier is trained with the $f_i$ embeddings to predict the class of each image, which is trained in the same way for the triple and categorical models.

community are used for multiclass classification problems such as the recall ($R_i$), precision ($P_i$), and F-score ($F_{1,i}$) will be employed. These metrics are calculated as follows:

$$\mathrm{R}_i = \frac{N_{ii}}{\sum_j N_{ij}},\ \mathrm{P}_i = \frac{N_{ii}}{\sum_i N_{ij}},\ F_{1,i} = 2\frac{\mathrm{P}_i \cdot \mathrm{R}_i}{\mathrm{P}_i + \mathrm{R}_i} \tag{3}$$

where $i$ refers to the real class (true label), $j$ to the predicted class from the algorithm and $N_{ii}$ and $N_{ij}$ correspond to the total number of images well-predicted or mixed between them respectively. These predictions are also used to represent the confusion matrix.

Besides that, the results obtained from the feature extractors will also be analyzed. First, a t-distributed Stochastic Neighbor Embedding (t-SNE) method (Van der Maaten and Hinton, 2008) is used to visualize the high-dimensional image features in a two-dimensional graph, which will allow the different clusters to be recognized. On the other hand, Davis-Bouldin index (Davies and Bouldin, 1979) and Silhouette score (Rousseeuw, 1987) clustering metrics are represented. DB index applies quantities and features inherent to the dataset to validate the clustering results, although a good value does not imply the best information retrieval. Lower values of the DB index mean better results. The Silhouette score measures the similarity of an object to its own cluster compared to other clusters. It ranges from −1 to +1, where a high value indicates a better result.

## 4. RESULTS

### 4.1. Training

All experiments were conducted on the training set, consisting of 80% of the full dataset, 10% was used for validation and all results were obtained from the test set, consisting of the other 10%. The distribution was done keeping the percentages of each class and considering the days of taking the images, so that all the images taken on the same day were included in the same set. The ResNet50 neural network pre-trained on the Imagenet dataset was used as the backbone, where the last layers were unfrozen allowing their weights to be modified (as described in Section 3.1). Different experiments were performed using two different loss functions and taking different number of images per class to create few shot learning models. The images were randomly selected and ranged from $N = 4$ images per class to $N = 2{,}000$. In all experiments, the same number of images was used for each of the disease classes as for the healthy class. In fact, although the collection of images of healthy plants is easier than that of diseased plants, the worst use case was defined and therefore an equal number of healthy images was selected as for the other classes. One experiment was run with each of the selected $N$-values and with each of the two explained loss functions to compare both networks in the few-shot experiments.

For the experiments the images were resized to 224 × 224 pixels, and data augmentation techniques were applied to the training images. Rotations (probability = 0.5), translations, scaling (probability = 0.7) and gamma transformation (probability = 0.5, gamma limits from 80 to 120) were selected. The experiments were conducted over 150 epochs and a learning rate of $10^{-4}$ was used with the Adam optimizer. The number of training images per class was systematically selected throughout the experiments and ranged from $N = 4$ to $N = 2{,}000$. All the experiments were run identically for the triplet loss-based model as for the categorical cross-entropy-based models. The output of the network was an embedding vector of 256 features. These vectors represent the ability of each loss function to cluster the different classes and hence the feature extraction capability of the network. In addition, a $k$-NN classifier is then applied to the feature extractor to predict the class related to each embedding so that classification results can also be evaluated.

## 4.2. Classification Results

F-score, Recall and Precision parameters have been measured to analyze plant disease classification results. **Figure 6** shows the average of the aforementioned metrics for each model created ranging from $N = 4$ images per class to $N = 2{,}000$. The embedding vectors obtained from the feature extractor have been fed into the $k$-NN classifier, which has been trained and then tested with the testing set. As shown in the first graph, the F-score parameter achieved with the triplet loss outperforms the categorical cross-entropy model up to $N = 200$ samples per class. This difference evens out for higher $N$-values, as the models use more images to train, but at the same time, the difference between them is larger when very few images are used to train (from $N = 4$ to $N = 30$). Thus, comparing the mean F-score values obtained from the F-score results for the different samples per image ($F1_{triplet} = 67.4\%$ vs. $F1_{cat} = 63.6\%$), we find that the F-score parameter increases by 6% when training with the proposed Siamese architecture and the triplet loss.

To show the most problematic classes, the F-score parameter has been calculated for each class using the model of $N = 30$ images per class. **Figure 7** shows that the F-score parameter is higher for almost all classes by using the triplet loss function. On the other hand, we can analyze that *Septoria nodorum* (LEPTNO) and *Septoria tritici* (SEPTTR) are the classes with the lowest value, below 50% for both Triplet and Categorical cross-entropy losses. For the case of Triplet loss, the confusion matrix has also been calculated to find the predictions of the algorithm for each class (**Figure 8**). As in the previous case, it can be observed that the class with the worst prediction is *Parastagonospora*

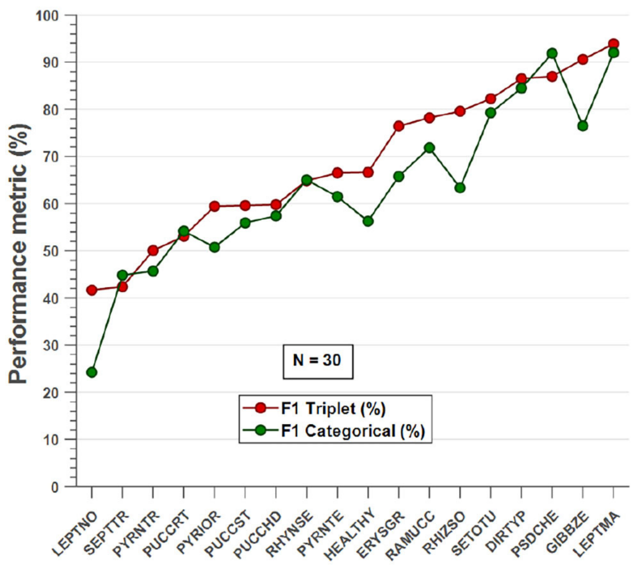


**FIGURE 7 |** F-score parameter calculated for each class for the case of $N$ = 30 images per class. This parameter shows a higher performance for almost all classes when using the triplet loss function.

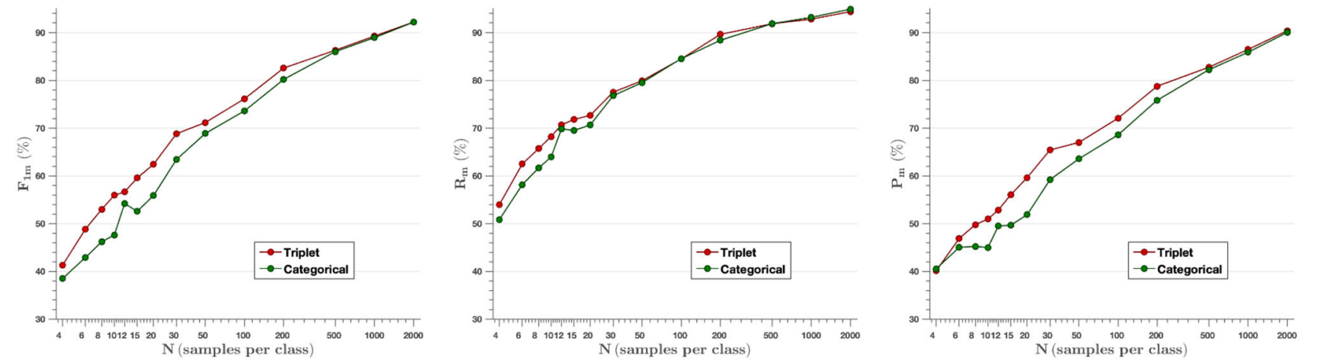
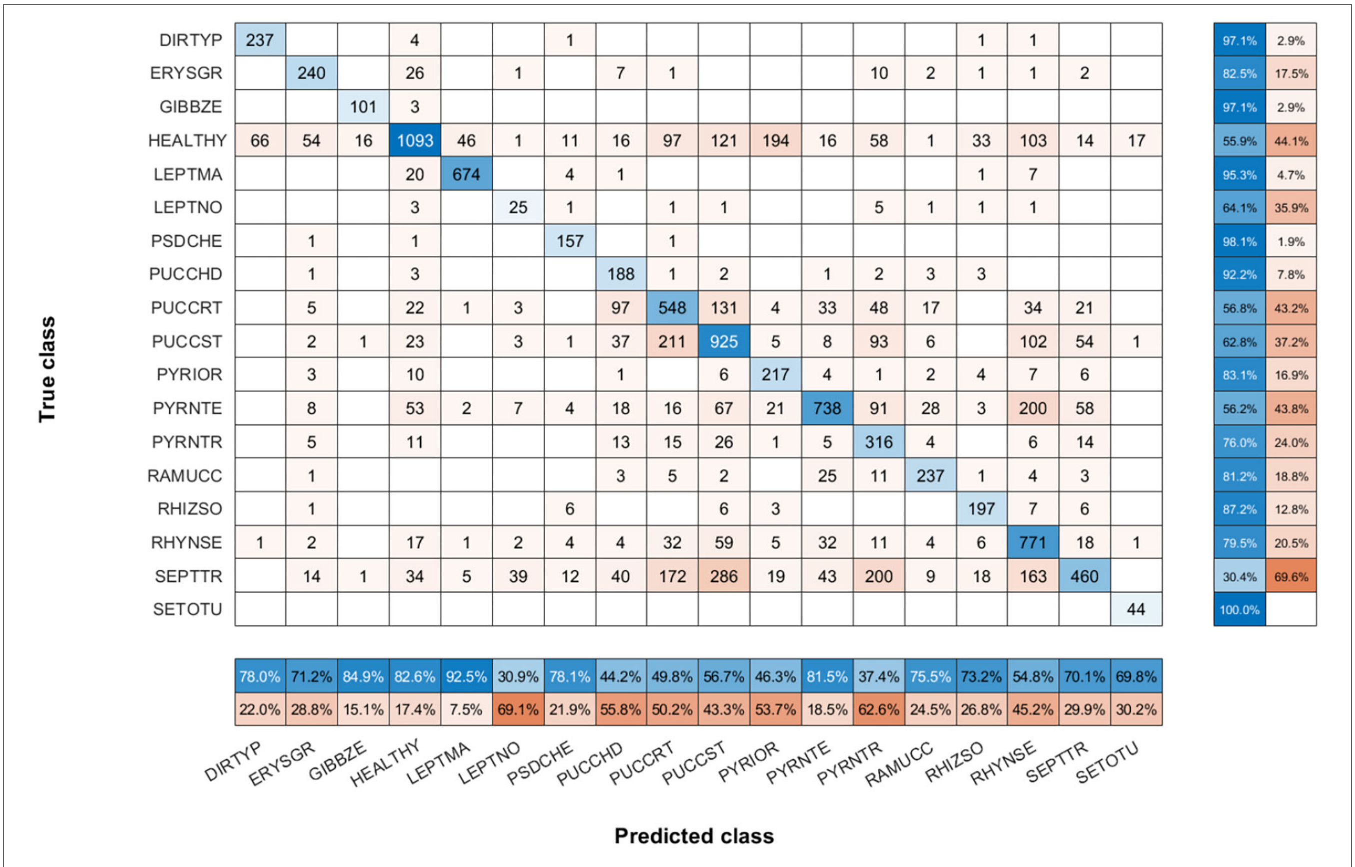


**FIGURE 6 |** Mean value of F-score (Left), Recall (Middle), and Precision (Right) parameters for all classes as a function of the number of images per class used in training. The results show that the model based on triplet loss outperforms the model based on categorical cross-entropy for experiments from $N = 4$ to $N = 200$ samples per class.



| | DIRTYP | ERYSGR | GIBBZE | HEALTHY | LEPTMA | LEPTNO | PSDCHE | PUCCHD | PUCCRT | PUCCST | PYRIOR | PYRNTE | PYRNTR | RAMUCC | RHIZSO | RHYNSE | SEPTTR | SETOTU | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIRTYP | 237 | | | 4 | | | 1 | | | | | | | | 1 | 1 | | | 97.1% | 2.9% |
| ERYSGR | | 240 | | 26 | | 1 | | 7 | 1 | | | | 10 | 2 | 1 | 1 | 2 | | 82.5% | 17.5% |
| GIBBZE | | | 101 | 3 | | | | | | | | | | | | | | | 97.1% | 2.9% |
| HEALTHY | 66 | 54 | 16 | 1093 | 46 | 1 | 11 | 16 | 97 | 121 | 194 | 16 | 58 | 1 | 33 | 103 | 14 | 17 | 55.9% | 44.1% |
| LEPTMA | | | | 20 | 674 | | 4 | 1 | | | | | | | 1 | 7 | | | 95.3% | 4.7% |
| LEPTNO | | | | 3 | | 25 | 1 | | 1 | 1 | | | 5 | 1 | 1 | 1 | | | 64.1% | 35.9% |
| PSDCHE | | 1 | | 1 | | | 157 | | 1 | | | | | | | | | | 98.1% | 1.9% |
| PUCCHD | | 1 | | 3 | | | | 188 | 1 | 2 | | 1 | 2 | 3 | 3 | | | | 92.2% | 7.8% |
| PUCCRT | | 5 | | 22 | 1 | 3 | | 97 | 548 | 131 | 4 | 33 | 48 | 17 | | 34 | 21 | | 56.8% | 43.2% |
| PUCCST | | 2 | 1 | 23 | | 3 | 1 | 37 | 211 | 925 | 5 | 8 | 93 | 6 | | 102 | 54 | 1 | 62.8% | 37.2% |
| PYRIOR | | 3 | | 10 | | | | 1 | | 6 | 217 | 4 | 1 | 2 | 4 | 7 | 6 | | 83.1% | 16.9% |
| PYRNTE | | 8 | | 53 | 2 | 7 | 4 | 18 | 16 | 67 | 21 | 738 | 91 | 28 | 3 | 200 | 58 | | 56.2% | 43.8% |
| PYRNTR | | 5 | | 11 | | | | 13 | 15 | 26 | 1 | 5 | 316 | 4 | | 6 | 14 | | 76.0% | 24.0% |
| RAMUCC | | 1 | | | | | | 3 | 5 | 2 | | 25 | 11 | 237 | 1 | 4 | 3 | | 81.2% | 18.8% |
| RHIZSO | | 1 | | | | | 6 | | | 6 | 3 | | | | 197 | 7 | 6 | | 87.2% | 12.8% |
| RHYNSE | 1 | 2 | | 17 | 1 | 2 | 4 | 4 | 32 | 59 | 5 | 32 | 11 | 4 | 6 | 771 | 18 | 1 | 79.5% | 20.5% |
| SEPTTR | | 14 | 1 | 34 | 5 | 39 | 12 | 40 | 172 | 286 | 19 | 43 | 200 | 9 | 18 | 163 | 460 | | 30.4% | 69.6% |
| SETOTU | | | | | | | | | | | | | | | | | | 44 | 100.0% | |
| | 78.0% | 71.2% | 84.9% | 82.6% | 92.5% | 30.9% | 78.1% | 44.2% | 49.8% | 56.7% | 46.3% | 81.5% | 37.4% | 75.5% | 73.2% | 54.8% | 70.1% | 69.8% | | |
| | 22.0% | 28.8% | 15.1% | 17.4% | 7.5% | 69.1% | 21.9% | 55.8% | 50.2% | 43.3% | 53.7% | 18.5% | 62.6% | 24.5% | 26.8% | 45.2% | 29.9% | 30.2% | | |

**FIGURE 8 |** Confusion matrix of $N = 30$ images per class model with triplet loss. The confusion matrix provides an accurate view of how correctly the model predicts the classes or how the classes are misclassified. The values of the diagonal represented in blue correspond to the number of correctly predicted images for each class. The values of the matrix outside the diagonal represented in orange correspond to incorrect predictions, where each cell relates the true class to the class predicted by the algorithm. In addition, below the confusion matrix, the precision values of each class are plotted horizontally in blue. Also, to the right of the confusion matrix, the recall values of each class are shown vertically in blue.

*nodorum* (LEPTNO), which is confused with *Zymoseptoria tritici* (SEPTTR). However, there is no other solution for that as there are not enough images available for LEPTNO and as mentioned above they are two very conflicting classes. From the point of view of plant physiology, both diseases are characterized by the presence of yellowish spots, which quickly turn into gray-brown lesions surrounded by a yellowish once the damage turns brown. In advances stages, the spots may contain small black dots (known as black pycnidia), which are the most characteristic sign of advanced septoria diseases, as shown in **Figure 2**. One difference is that LEPTNO the pycnidia are smaller, even very difficult to see without the aid of a magnifying glass, and in SEPTTR the pycnidia are visible to the naked eye. SEPTTR is also confused with some other classes, such as *Drechslera tritici-repentis* (PYRNTR), *Puccinia striiformis* (PUCCST), or Puccinia recondite (PUCCRT), which implies that all of them have poor outcomes. In addition, the latter two are also very confusing to each other, since as mentioned above, the different symptoms between them are very subtle. In the early stages of these diseases, individual yellow to orange-brown pustules appear on the leaves. In PUCCRT the pustules tend to be randomly scattered whereas in PUCCST they form in small pockets at the beginning or when the leaves are young, and as the disease progresses, they form in bands. The color of the pustules is usually orange-brown in PUCCRT and orange-yellow in PUCCST.

**Figure 9** analyzes the effect of the $K$-value in the $k$-NN classifier for three different experiments ($N = 6$, $N = 30$, $N = 200$), where the parameter F-score is calculated for different values of $K$ $(3, 5, ..., 21)$. It can be appreciated that Triplet approach surpasses the performance of categorical cross-entropy method. The only exception is observed on $N = 6$ where the choice of value of k higher than the number of image per class ($N = 6$) reduces the performance of the experiment, since in that case the classifier takes into consideration samples of different classes for each prediction. On the other hand, when the value of the classifier is lower than the number of images per class, there is little variability in the results, so the value of K does not influence them. That is, when designing the algorithm, it is not possible to set a value of K higher than the number of classes, since, as can be seen in the graph on the left, the F-score decreases in both models as the value of K increases with respect to N. However, for values

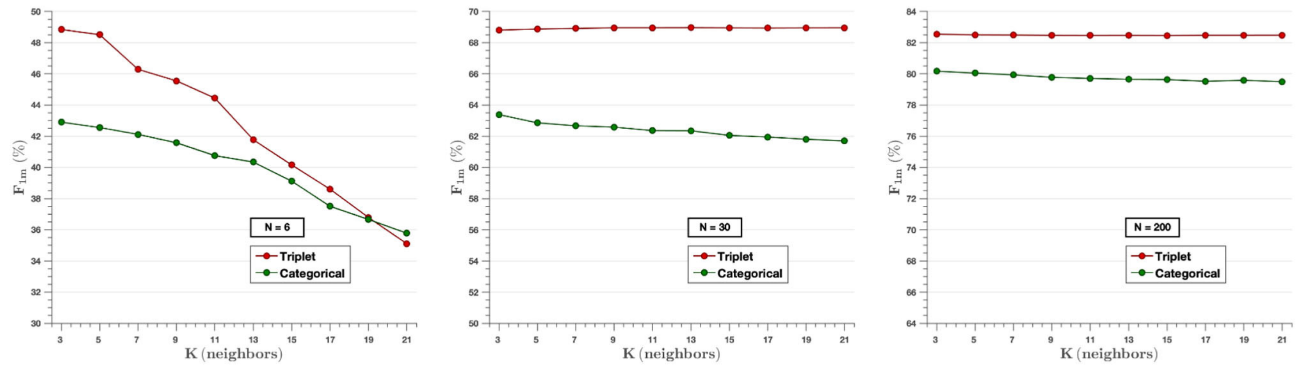


**FIGURE 9 |** The effect of the *K*-value of the classifier knn for the experiments of *N* = 6 (Left), *N* = 30 (Middle), and *N* = 200 (Right) images per class. In the cases of *N* = 30 and *N* = 200, very little variability is observed for all *K*-values selected. In contrast, in the case of *N* = 6, the value of the F-score decreases as higher *K*-values are chosen.

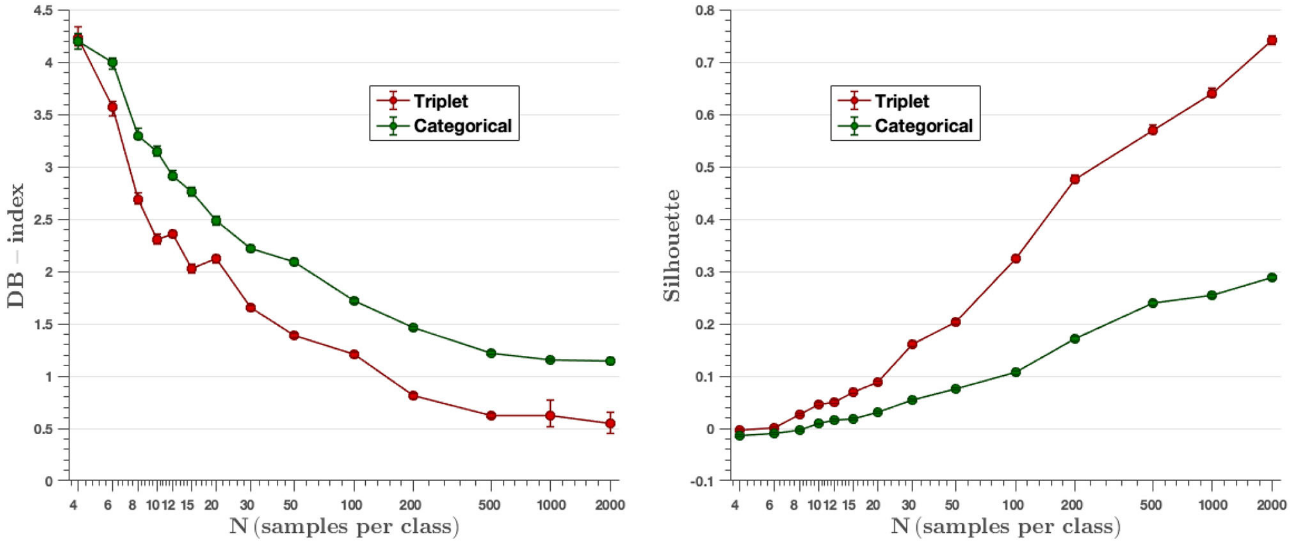


**FIGURE 10 |** DB-index and Silhouette score metrics. The triplet approach achieves a lower DB-index and a higher Silhouette value than the categorical cross-entropy method for all experiments, resulting in a better ability to group classes into different clusters.

of N higher than K, it is observed that the use of the triplet loss model achieves better performance.

## 4.3. Clustering Results

Clusters are generated by the output embedding vectors of the feature extractor. Davis-Bouldin and Silhouette metrics represent the capacity the network has to group the classes in different clusters. **Figure 10** displays the values of mentioned parameters for all the models created, where we can see that results improve while the number of images per class increment. Moreover, in all cases, triplet loss models achieve better results for both parameters. The average values of the DB-index and Silhouette parameters have been calculated considering the values obtained in all experiments (from $N = 4$ to $N = 2{,}000$). On the other hand, we observe that the triplet loss model improves the value by 22.7% with respect to the categorical cross-entropy loss model ($DBindex_{triplet} = 1.87$ vs. $DBindex_{cat} = 2.42$). Similarly, the Silhouette value also improves, now by 166.7% ($Silhouette_{triplet} = 0.24$ vs. $Silhouette_{cat} = 0.09$). These metrics are used to assess the quality of the clusters generated by the embedding vectors, and these results show that the triplet loss model achieves better cluster separability than the categorical-cross entropy loss model.

The t-SNE technique obtains the representation of the embedding vectors in a two-dimensional graph. The 256-dimensional embedding vectors are then reduced to two dimensions in order to visualize the clustering capabilities of the different losses to group the test embeddings into class clusters. **Figure 11** presents the t-SNE graph of the models created during training, showing the results obtained with the triplet loss models in the left column and the results with the categorical cross-entropy loss models in the right column. Four experiments have been plotted: on the top left, the model results using 2,000 images

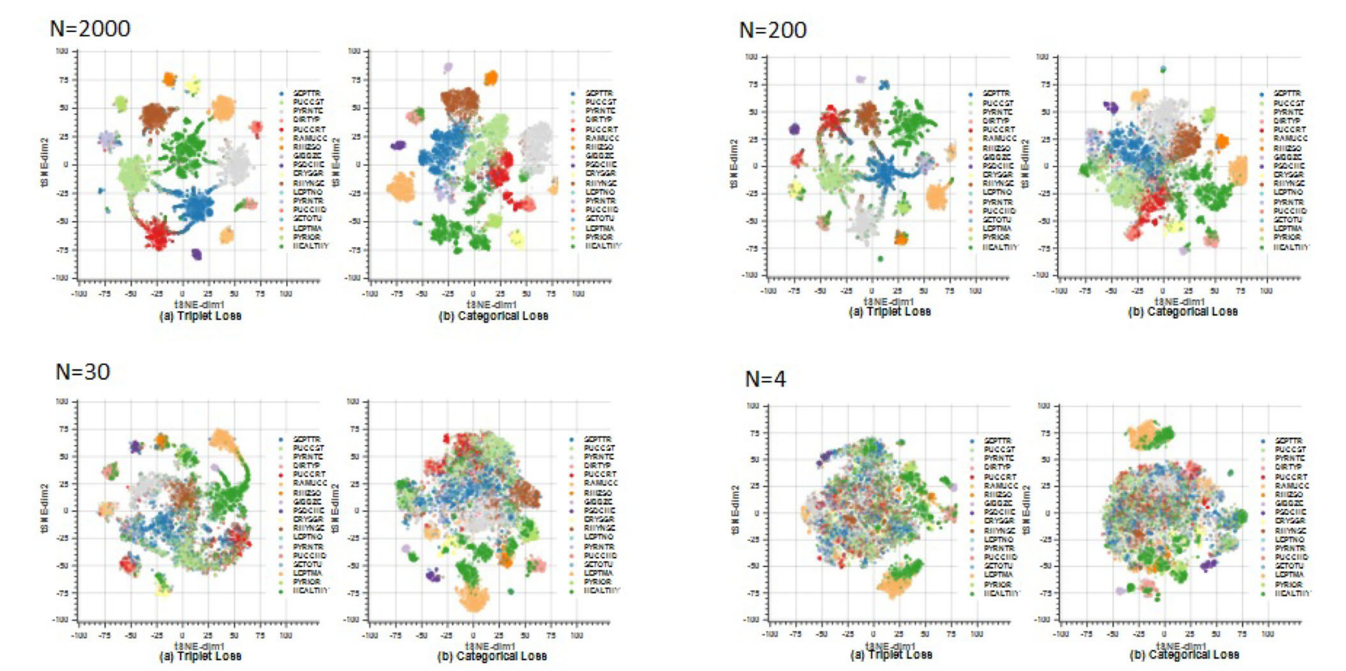


**FIGURE 11 |** Test embeddings of $N = 2{,}000$ (top left), $N = 200$ (top right), $N = 30$ (bottom left), $N = 4$ (bottom right) reduced to 2 dimensions using the t-SNE technique. For each experiment, the graphs on the left show the results obtained with the triplet loss, and those on the right show the results obtained with the categorical cross-entropy loss. In the case of $N = 2{,}000$ and $N = 200$, high class separability is observed with triplet loss. By reducing the number of images per class to $N = 30$, the triplet loss model loses separability in certain classes. Finally, in the case of $N = 4$, the class groups are not well-defined. In the case of categorical cross-entropy loss, similar results are shown in all graphs.

per class; on the top right, the model results using 200 images per class; on the bottom left, the model results using 30 images per class; and on the bottom right, the model results using 4 images per class. Each color represents a different class. We can see that in the case of $N = 2{,}000$, the model trained with the triplet loss maximizes the interclass distance achieving a high separability between classes, while minimizing the intraclass distance, creating a grouped cluster of each of the classes. It can also be observed how the SEPTTR class represented in dark blue is close at certain points to several clusters belonging to the PUCCST (light green) or PUCCRT (red) classes, which confuses the algorithm, as analyzed in the results of section 4.5. On the other hand, the t-SNE 2D projection of the model trained with categorical cross-entropy shows a larger overlap between classes which is consistent with the DB index and Silhouette values. By reducing the number of images to 200, a clear difference between the two models is observed. In the case of triplet loss, similar results are obtained with respect to the previous model analyzed ($N = 2{,}000$). In fact, the model already obtains compact clusters with $N = 200$ and improves slightly when training with more images. In the case of $N = 30$, as before, we obtain a better separability between classes by training the model with triplet loss, since in the case of categorical cross-entropy loss very few classes are well-defined. However, if we compare this with the previous cases of $N = 200$ and $N = 2{,}000$, we observe that for both models, the experiments trained with 200 and 2,000 images per class achieve a better clustering of the classes with respect to the experiments trained with 30 images per class. This is due to the fact that, by training with a larger number of images the models manage to extract the most representative features of each class which are reflected in the embedding vectors. Finally, in the case of $N = 4$, similar results are obtained with both triplet and categorical cross-entropy loss, where no class separability is shown in either case.

## 4.4. Statistical Analysis

To calculate the statistical significance of the performance of the two proposed algorithms, we follow the approach proposed by Dietterich (1998), where the use of the McNemar test is recommended in cases where multiple iterations of the test are not possible or are time-consuming. McNemar's test proposes a tabulation of the responses given by two proposed qualifiers (in our case the triplet loss and categorical cross-entropy algorithms) where their discrepancies are measured for marginal homogeneity.

Since McNemar's test is aimed at binary decision classifiers, we employ the Stuart-Maxwell test (Maxwell, 1970) which is an extension of McNemar's test for multiclass classification algorithms (Cano-Espinosa et al., 2020). **Table 2** details the results obtained. We obtain statistical significance ($p_{value}$<0.01) for all experiments involving less than 500 images for training, while there is no statistical significance for experiments with a number of images greater than 500. This demonstrates the benefit of using metric learning approaches for few-shot learning

**TABLE 2 |** This table shows the statistical significance of the differences among the two proposed classifiers by a different number of training images.

| No of images | 4 | 6 | 8 | 10 | 12 | 15 | 20 | 30 | 50 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| $p_{value}$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $\chi^2$ | 0963.26 | 1888.42 | 2101.88 | 2513.66 | 1044.49 | 1590.29 | 1552.43 | 1056.37 | 906.48 | 959.85 |

| No of images | 200 | 500 | 1k | 2k |
|---|---|---|---|---|
| $p_{value}$ | 0.00 | 0.00 | 0.38 | 1.00 |
| $\chi^2$ | 280.09 | 209.58 | 157.85 | 102.98 |

*It can be appreciated that the differences are significant ($p_{value}$ < 0.01) for training image numbers lower than 500 whereas classifier differences are not significant with larger number of training images.*

compared to classical metrics, as also seen in other fields (Medela and Picon, 2019; Argüeso et al., 2020).

## 4.5. Analysis of Explainability

The dataset used to develop the project contains images of plants taken in the field. Therefore, images containing human digits with or without blue gloves are included, which could affect the results of the experiments. The images were captured in different campaigns and at different locations, and in all cases the same protocols were followed for the acquisition of images of all crops so that the occurrence of artifacts was controlled. Additionally, along with the acquisition campaigns, the pictures taken on the same day were assigned to a unique dataset subset (train, validation, or test) to avoid data contamination.

To analyze the influence of these artifacts on the algorithm performance, the Grad-CAM technique (Selvaraju et al., 2017) has been selected, which produces visual explanations for the CNN-based model decisions. It uses the gradients leading to the final convolutional layer to produce a coarse localization map that highlights important regions of the image for prediction. The Grad-CAM technique has been applied to the trained models and test set images to find the most significant regions of the images that the model focused on to predict plant diseases. **Figure 12** shows the results obtained on different images for all the diseases, where it can be seen that the trained model correctly focuses on the most representative parts of the diseases, without being affected by the different artifacts such as gloves, human hands or specific backgrounds that might appear on the image.

## 5. DISCUSSION

In recent years, deep learning-based models for plant disease detection have become increasingly important. Thus, some datasets have been created and made publicly available for research. An example of an open access dataset is the PlantVillage dataset, which consists of over fifty thousand images of 26 different diseases that have been taken under controlled conditions. Images of individual leaves are photographed on a plain background where diseases are clearly visible in most cases, as only late stage diseases have been considered. Several experiments have been developed using the PlantVillage dataset and have achieved high performance (Mohanty et al., 2016; Rangarajan et al., 2018). In addition, few-shot learning models have also been applied to this dataset in order to address the problems of acquiring large datasets. Argüeso et al. (2020) reached a median accuracy of 80% for the 6 classes selected in the target dataset using only 15 images per class. However, this model was pre-trained with all class images from the source dataset which acted as a supporting dataset. Therefore, with our experiments we want to demonstrate that by using distance metric techniques good results can be achieved using few images without needing a large dataset of annotated images on which to train the base model. Moreover, the experiments have been developed using images from a dataset taken in real field conditions, which differs from a laboratory dataset by having varied and non-uniform backgrounds, different lighting conditions, different perspectives and distances, as well as by including different disease stages (early and late stages). Five crops with a total of 17 different diseases are included in the dataset.

Our experiments compare two different approaches: a Siamese network based on the distance metric with a triplet loss function, and a traditional network with the categorical cross-entropy loss function. Different models have been developed using from $N = 4$ to $N = 2{,}000$ images per class. The results show that the Siamese network with the triplet loss function achieves an average f-score above 55.0% from $N = 10$, while the values obtained with the categorical cross-entropy are below in most cases (**Figure 6**). By increasing the number of classes to $N = 30$, the triplet loss achieves an F-score of 69%, which, considering that there are still very few images, is a great improvement. The effect of the parameter k of the *k*-NN classifier has been analyzed, where it has been observed that the algorithm keeps the results constant for different values of k, except for the cases where K is larger than N for low values of N, in which the effect of the parameter k is large (**Figure 9**). On the other hand, the results of the feature extractor part of the model have been analyzed. We have analyzed the ability to create clusters of each class by means of the embedding vectors obtained from the CNN through the DB-index and Silhouette parameters, as well as by applying the t-SNE technique. In both cases it has been observed that the Siamese network with the loss of the triplet separates the different classes better, obtaining a clear cluster for each class (**Figure 11**).

We have performed a statistical analysis to find the most significant differences between the two approaches. In the

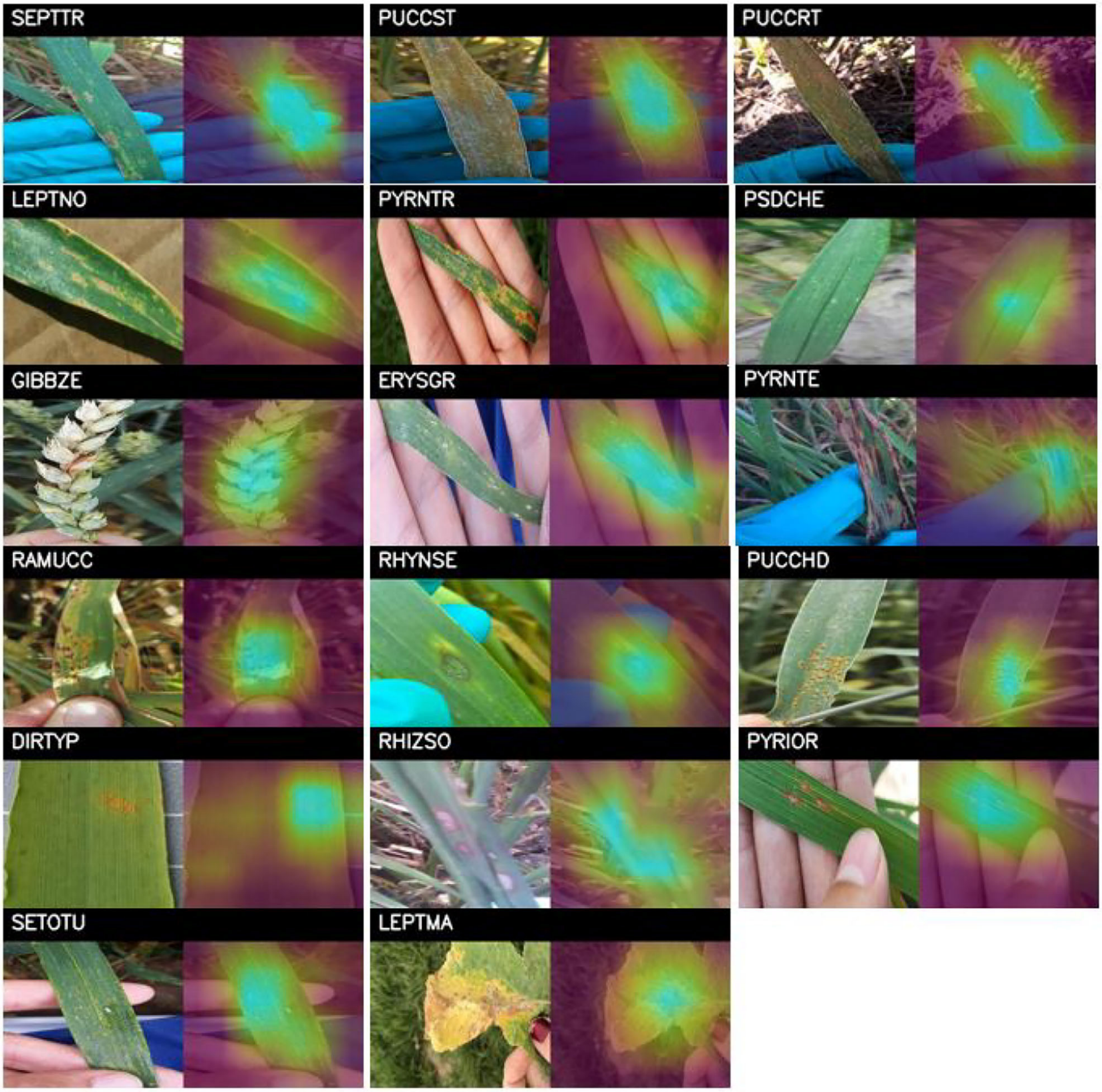


**FIGURE 12 |** Grad-CAM results applied to test images of all diseases. For each disease, the original image has been plotted on the left, and the most significant regions detected by the algorithm is represented on the right. Each disease is expressed by its EPPO code: SEPTTR (*Septoria tritici*), PUCCST (*Puccinia striiformis*), PUCCRT (*Puccinia recondita*), LEPTNO (*Septoria nodorum*), PYRNTR (Drechslera tritici-repentis), PSDCHE (*Oculimacula yallundae*), GIBBZE (*Gibberella zeae*), EYRSGR (*Blumeria graminis*), PRYNTE (*Pyrenophora teres*), RAMUCC (*Ramularia collo-cygni*), RHYNSE (*Rhynchosporium secalis*), PUCCHD (*Puccinia hordei*), DIRTYP (*Various diseases*), RHIZSO (*Thanatephorus cucumeris*), PYRIOR (*Pyricularia oryzae*), SETOTU (*Helminthosporium turcicum*), and LEPTMA (*Phoma lingam*).

range from $N = 4$ to $N = 500$, it has been observed that triplet loss-based method outperforms the results obtained with the categorical cross-entropy loss model. For example, in the intermediate value of $N = 30$, we can appreciate that we obtain better results both for classification performance (F1 = 0.69 vs. F1 = 0.63) as well as for clustering performance (DB-Index = 2.25 vs. DB-Index = 1.62) and Silhouette (0.17 vs. 0.05) for the triplet loss approach. For $N$-values above 500, differences are not statistically significant.

## 6. CONCLUSIONS

This study analyzes two different networks to develop a model based on deep learning techniques from a few images for plant disease classification: a Siamese network based on the distance metric with a triplet loss function, and a traditional network with the categorical cross-entropy loss function as defined by Picon et al. (2019a).

The experiments have been developed using few images per class. It is noteworthy that we stand for the most complicated case where there is no supporting dataset for creating the few-shot latent descriptor as it is performed in the classical few-shot approaches. For this reason, in this study we have sought to demonstrate that a distance metric-based Siamese network with a triplet loss function is able to learn image features from few images without the need for a supporting dataset which is a more realistic and demanding few-shot use case.

The triplet loss model improves the average F-score value by 6% with respect to the categorical cross-entropy loss. The triplet loss model achieves higher F-score values for all values of N, where the main difference between the two architectures appears at the lowest values of N. Furthermore, it has been analyzed that this difference is due to the fact that the triplet loss model is able to learn the features of classes with fewer available samples and similar symptoms, considered as the most difficult classes. Without loss of generality, in the particular case of $N = 30$, the proposed method outperformed the baseline method for disease classification (F1 = 0.69 vs. F1 = 0.63, $N = 30$). The classes that benefited most from these improvements were LEPTNO and PYRIOR among the most problematic classes, as well as ERYSGR, RHIZSO, or GIBBZE among the classes with the best predictions.

If we analyze the quality of the generated latent space descriptors, we can appreciate that the triplet loss model outperforms the cross-entropy categorical loss model by obtaining more compact and separated clusters (DB-Index = 2.25 vs. DB-Index = 1.62), Silhouette (0.17 vs. 0.05) which allows for easier feature extraction and image retrieval. The triplet loss model improves the mean value of the DB-index parameter by 22.7% over the categorical cross-entropy model, as well as the mean value of the Silhouette score by an improvement of 166.7%.

An important remark when using knn as shallow classifier after the feature extraction that must be taken into account is that the value selected for K must be greater than the number of images per class used for training. In addition, it has also been shown that the results obtained by the classifier are better in the case of triplet loss.

Our results show that triplet loss approach obtains better results than state of the art deep learning approaches for both discriminating plant diseases and generating better latent descriptors in the case of real and complex dataset taken in real field implying complex conditions (changing backgrounds, different lighting conditions, different distances, different disease stages...) where only few images per class are available.

This generates new research opportunities for the use of these techniques in the generation of large and openly available feature extraction models that could help structure plant disease representations allowing few-shot characterization of uncommon and rare diseases.

## DATA AVAILABILITY STATEMENT

The data analyzed in this study is subject to the following licenses/restrictions. The dataset used in this article has been generated by the BASF R&D field research community. It could be made available on reasonable request for non-commercial research purposes and under an agreement with BASF. Requests to access these datasets should be directed to ramon.navarra-mestre@basf.com.

## AUTHOR CONTRIBUTIONS

IE: conceptualization, investigation, software, formal analysis, methodology, and writing—original draft, review, and editing. AP: conceptualization, investigation, software, methodology, and writing—review and editing. UI: conceptualization, formal analysis, investigation, and writing—review and editing. AB-P: investigation and writing—review and editing. TE and EA: contextualization and writing—review and editing. CK: methodology, software, and writing—review and editing. RN-M: investigation, methodology, and writing—review and editing. All authors contributed to the article and approved the submitted version.

## FUNDING

This project was partially supported by the Spanish Government through CDTI Centro para el Desarrollo Tecnológico e Industrial project AI4ES (ref CER-20211030), by the University of the Basque Country (UPV/EHU) under grant COLAB20/01 and by the Basque Government through grant IT1229-19.

## ACKNOWLEDGMENTS

In Memoriam: We would like to thank UI for all his dedication, care, support, guiding, and affection during all time we have spent together. Eskerrik asko Unai.

## REFERENCES

Argüeso, D., Picon, A., Irusta, U., Medela, A., San-Emeterio, M. G., Bereciartua, A., et al. (2020). Few-shot learning approach for plant disease classification using images taken in the field. *Comput. Electron. Agric.* 175, 105542. doi: 10.1016/j.compag.2020.105542

Camargo, A., and Smith, J. (2009). An image-processing based algorithm to automatically identify plant disease visual symptoms. *Biosyst. Eng.* 102, 9–21. doi: 10.1016/j.biosystemseng.2008.09.030

Cano-Espinosa, C., González, G., Washko, G. R., Cazorla, M., and Estépar, R. S. J. (2020). Biomarker localization from deep learning regression networks. *IEEE Trans. Med. Imaging* 39, 2121–2132. doi: 10.1109/TMI.2020.2965486

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. (2021). Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*.

Dacal-Nieto, A., Vázquez-Fernández, E., Formella, A., Martin, F., Torres-Guijarro, S., and González-Jorge, H. (2009). "A genetic algorithm approach for feature selection in potatoes classification by computer vision," in *2009 35th Annual Conference of IEEE Industrial Electronics* (Porto: IEEE), 1955–1960.

Davies, D. L., and Bouldin, D. W. (1979). A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* 1, 224–227. doi: 10.1109/TPAMI.1979.4766909

DeChant, C., Wiesner-Hanks, T., Chen, S., Stewart, E. L., Yosinski, J., Gore, M. A., et al. (2017). Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. *Phytopathology* 107, 1426–1432. doi: 10.1094/PHYTO-11-16-0417-R

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). "Imagenet: a large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition* (Miami, FL: IEEE), 248–255.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput.* 10, 1895–1923. doi: 10.1162/089976698300017197

Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* 145, 311–318. doi: 10.1016/j.compag.2018.01.009

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

Ghosal, S., Blystone, D., Singh, A. K., Ganapathysubramanian, B., Singh, A., and Sarkar, S. (2018). An explainable deep machine vision framework for plant stress phenotyping. *Proc. Natl. Acad. Sci. U.S.A.* 115, 4613–4618. doi: 10.1073/pnas.1716999115

Hasan, R. I., Yusuf, S. M., and Alzubaidi, L. (2020). Review of the state of the art of deep learning for plant diseases: a broad analysis and discussion. *Plants* 9, 1302. doi: 10.3390/plants9101302

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 770–778.

Hirooka, T., and Ishii, H. (2013). Chemical control of plant diseases. *J. Gen. Plant Pathol.* 79, 390–401. doi: 10.1007/s10327-013-0470-6

Hu, G., Wu, H., Zhang, Y., and Wan, M. (2019). A low shot learning method for tea leaf's disease identification. *Comput. Electron. Agric.* 163, 104852. doi: 10.1016/j.compag.2019.104852

Hughes, D., and Salathé, M. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060*.

Johannes, A., Picon, A., Alvarez-Gila, A., Echazarra, J., Rodriguez-Vaamonde, S., Navajas, A. D., et al. (2017). Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case. *Comput. Electron. Agric.* 138, 200–209. doi: 10.1016/j.compag.2017.04.013

Kamal, K., Yin, Z., Wu, M., and Wu, Z. (2019). Depthwise separable convolution architectures for plant disease classification. *Comput. Electron. Agric.* 165, 104948. doi: 10.1016/j.compag.2019.104948

Kim, D. G., Burks, T. F., Qin, J., and Bulanon, D. M. (2009). Classification of grapefruit peel diseases using color texture feature analysis. *Int. J. Agric. Biol. Eng.* 2, 41–50. doi: 10.3965/j.issn.1934-6344.2009.03.041-050

Kiratiratanapruk, K., and Sinthupinyo, W. (2011). "Color and texture for corn seed classification by machine vision," in *2011 International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS)* (Chiang Mai: IEEE), 1–5.

Li, L., Zhang, S., and Wang, B. (2021). Plant disease detection and classification by deep learning–a review. *IEEE Access* 9, 56683–56698. doi: 10.1109/ACCESS.2021.3069646

Li, Y., and Yang, J. (2021). Meta-learning baselines and database for few-shot classification in agriculture. *Comput. Electron. Agric.* 182, 106055. doi: 10.1016/j.compag.2021.106055

Lin, K., Gong, L., Huang, Y., Liu, C., and Pan, J. (2019). Deep learning-based segmentation and quantification of cucumber powdery mildew using convolutional neural network. *Front. Plant Sci.* 10, 155. doi: 10.3389/fpls.2019.00155

Maxwell, A. E. (1970). Comparing the classification of subjects by two independent judges. *Brit. J. Psychiatry* 116, 651–655. doi: 10.1192/bjp.116.535.651

Medela, A., and Picon, A. (2019). Constellation loss: improving the efficiency of deep metric learning loss functions for optimal embedding. *arXiv preprint arXiv:1905.10675*. doi: 10.4103/jpi.jpi_41_20

Mohameth, F., Bingcai, C., and Sada, K. A. (2020). Plant disease detection with deep learning and feature extraction using plant village. *J. Comput. Commun.* 8, 10–22. doi: 10.4236/jcc.2020.86002

Mohanty, S. P., Hughes, D. P., and Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Front. Plant Sci.* 7, 1419. doi: 10.3389/fpls.2016.01419

Nazki, H., Yoon, S., Fuentes, A., and Park, D. S. (2020). Unsupervised image translation using adversarial networks for improved plant disease recognition. *Comput. Electron. Agric.* 168, 105117. doi: 10.1016/j.compag.2019.105117

Oerke, E.-C. (2006). Crop losses to pests. *J. Agric. Sci.* 144, 31–43. doi: 10.1017/S0021859605005708

Phadikar, S., Sil, J., and Das, A. K. (2012). Classification of rice leaf diseases based on morphological changes. *Int. J. Inform. Electron. Eng.* 2, 460–463. doi: 10.7763/IJIEE.2012.V2.137

Phadikar, S., Sil, J., and Das, A. K. (2013). Rice diseases classification using feature selection and rule generation techniques. *Comput. Electron. Agric.* 90, 76–85. doi: 10.1016/j.compag.2012.11.001

Picon, A., Alvarez-Gila, A., Seitz, M., Ortiz-Barredo, A., Echazarra, J., and Johannes, A. (2019a). Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild. *Comput. Electron. Agric.* 161, 280–290. doi: 10.1016/j.compag.2018.04.002

Picon, A., Seitz, M., Alvarez-Gila, A., Mohnke, P., Ortiz-Barredo, A., and Echazarra, J. (2019b). Crop conditional convolutional neural networks for massive multi-crop plant disease classification over cell phone acquired images taken on real field conditions. *Comput. Electron. Agric.* 167, 105093. doi: 10.1016/j.compag.2019.105093

Rangarajan, A. K., Purushothaman, R., and Ramesh, A. (2018). Tomato crop disease classification using pre-trained deep learning algorithm. *Proc. Comput. Sci.* 133, 1040–1047. doi: 10.1016/j.procs.2018.07.070

Revathi, P., and Hemalatha, M. (2012). "Classification of cotton leaf spot diseases using image processing edge detection techniques," in *2012 International Conference on Emerging Trends in Science, Engineering and Technology (INCOSET)* (Tiruchirappalli: IEEE), 169–173.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20, 53–65. doi: 10.1016/0377-0427(87)90125-7

Saleem, M. H., Potgieter, J., and Arif, K. M. (2019). Plant disease detection and classification by deep learning. *Plants* 8, 468. doi: 10.3390/plants8110468

Sandhu, G. K., and Kaur, R. (2019). "Plant disease detection techniques: a review," in *2019 International Conference on Automation, Computational and Technology Management (ICACTM)* (London: IEEE), 34–38.

Sannakki, S. S., Rajpurohit, V. S., Nargund, V., Kumar, A., and Yallur, P. S. (2011). Leaf disease grading by machine vision and fuzzy logic. *Int. J.* 2, 1709–1716. doi: 10.1109/SPIN.2015.7095350

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 (ICCV)*, Venice.

Shruthi, U., Nagaveni, V., and Raghavendra, B. (2019). "A review on machine learning classification techniques for plant disease detection," in *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)* (Coimbatore: IEEE), 281–284.

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., and Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Comput. Intell. Neurosci.* 2016, 3289801. doi: 10.1155/2016/3289801

Toda, Y., and Okura, F. (2019). How convolutional neural networks diagnose plant disease. *Plant Phenom.* 2019, 9237136. doi: 10.34133/2019/9237136

Too, E. C., Yujian, L., Njuki, S., and Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric.* 161, 272–279. doi: 10.1016/j.compag.2018.03.032

Van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9, 2579–2605.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. (2018). "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 3733–3742.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, 2223–2232.

**Conflict of Interest:** IE, AP, and AB-P were employed by TECNALIA. TE, CK, and RN-M were employed by BASF SE.

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

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

frontiers in Artificial Intelligence

ORIGINAL RESEARCH
published: 25 March 2022
doi: 10.3389/frai.2022.830026

# Behind the Leaves: Estimation of Occluded Grapevine Berries With Conditional Generative Adversarial Networks

*Jana Kierdorf*[1]*, *Immanuel Weber*[2], *Anna Kicherer*[3], *Laura Zabawa*[4], *Lukas Drees*[1] and *Ribana Roscher*[1]

[1] Remote Sensing Group, Institute of Geodesy and Geoinformation, University of Bonn, Bonn, Germany, [2] Application Center for Machine Learning and Sensor Technology, University of Applied Sciences Koblenz, Koblenz, Germany, [3] Julius Kühn-Institut (JKI), Federal Research Centre for Cultivated Plants, Institute for Grapevine Breeding Geilweilerhof, Siebeldingen, Germany, [4] Geodesy Group, Institute of Geodesy and Geoinformation, University of Bonn, Bonn, Germany

OPEN ACCESS

**Edited by:**
*Ian Stavness,*
*University of Saskatchewan, Canada*

**Reviewed by:**
*Andrew P. French,*
*University of Nottingham,*
*United Kingdom*
*Luigi Celona,*
*University of Milano-Bicocca, Italy*

***Correspondence:**
*Jana Kierdorf*
*jkierdorf@uni-bonn.de*

**Specialty section:**
*This article was submitted to*
*AI in Food, Agriculture and Water,*
*a section of the journal*
*Frontiers in Artificial Intelligence*

**Received:** *06 December 2021*
**Accepted:** *28 February 2022*
**Published:** *25 March 2022*

**Citation:**
*Kierdorf J, Weber I, Kicherer A, Zabawa L, Drees L and Roscher R (2022) Behind the Leaves: Estimation of Occluded Grapevine Berries With Conditional Generative Adversarial Networks.*
*Front. Artif. Intell. 5:830026.*
*doi: 10.3389/frai.2022.830026*

The need for accurate yield estimates for viticulture is becoming more important due to increasing competition in the wine market worldwide. One of the most promising methods to estimate the harvest is berry counting, as it can be approached non-destructively, and its process can be automated. In this article, we present a method that addresses the challenge of occluded berries with leaves to obtain a more accurate estimate of the number of berries that will enable a better estimate of the harvest. We use generative adversarial networks, a deep learning-based approach that generates a highly probable scenario behind the leaves exploiting learned patterns from images with non-occluded berries. Our experiments show that the estimate of the number of berries after applying our method is closer to the manually counted reference. In contrast to applying a factor to the berry count, our approach better adapts to local conditions by directly involving the appearance of the visible berries. Furthermore, we show that our approach can identify which areas in the image should be changed by adding new berries without explicitly requiring information about hidden areas.

**Keywords: deep learning, machine learning, Generative Adversarial Networks, domain-transfer, grape generation, occlusions, yield counting**

## 1. INTRODUCTION

With increasing competition on the wine market worldwide, the need for accurate yield estimations has been getting more and more important for viticulture. The variation of yield over the years is mainly caused by the berry number per vine (90%), while the remaining 10% are caused by the average berry weight (Clingeleffer et al., 2001), which is generally collected manually and averaged over many years. Traditionally, yield estimations in viticulture can be done at three phenological timepoints by (1) counting the number of bunches 4–6 weeks after budburst, (2) counting the number of berries after fruit set (May, 1972), or (3) destructively sampling vines or segments of vines close to harvest. Considering that yield estimation can be more accurately and reliably determined as harvest approaches, a berry count is a promising option that can be approached non-destructively and whose process can be automated.

Several papers show that machine learning-based methods for analyzing data from imaging sensors provide an objective and fast method for counting visible berries (Diago et al., 2012; Kicherer et al., 2014; Nuske et al., 2014; Roscher et al., 2014; Aquino et al., 2017; Coviello et al., 2020; Zabawa et al., 2020), and thus for automated yield predictions in the field. One of the main challenges in deriving berry counts from image data taken in the field is occlusions, which generally causes an underestimation of the number of berries and yield (Zabawa et al.[1]). First, occlusions of berries by other berries make it difficult to distinguish or count individual berries. Therefore, approaches that perform a segmentation of regions of berries and regions without berries is not sufficient, and more advanced methods that recognize individual instances of berries must be applied (Zabawa et al., 2020). Second, occlusions by leaves play a major role in underestimating the number of berries. Zabawa et al. (see text footnote 1, respectively) perform leaf occlusion experiments over two years and show that the yield estimation is highly dependent on the number of visible berries. With vines defoliated (i.e., with manually removed leaves) at pea size, they report an average error of total yield estimation of 27%, whereas Nuske et al. (2014) observed average errors between 3 and 11% using images of entirely defoliated fruit zones.

In order to overcome the challenge of leaf occlusions, defoliation can be performed in the grapefruit zone, but this is immensely time-consuming and labor-intensive. Partial defoliation is carried out in viticulture, for example, for ventilation and rapid drying of the grape zone to avoid fungal infections of the grapes or yield and quality regulation (Diago et al., 2009). However, complete defoliation is not feasible on a large scale or may lead to negative effects such as increased sunburn on the berries (Feng et al., 2015) or generally have an undesirable impact on yield results. Alternatively, machine-learning-based approaches can be used to obtain a more accurate estimation of the berry number. Numerous approaches rely on information where occlusions are present, which is generally provided as a manual input (Bertalmio et al., 2003; Barnes et al., 2009; Iizuka et al., 2017; Dekel et al., 2018; Liu et al., 2018). In contrast to this, two-step approaches first detect occlusions and then fill the corresponding regions with information according to the environment (Ostyakov et al., 2018; Yan et al., 2019).

This article addresses the challenge of occlusions caused by leaves by generating images that reveal a highly probable situation behind the leaves, exploiting learned patterns from a carefully designed dataset. The generated images can then be used to count berries in a post-processing step. Our approach generates potential berries behind leaves based on RGB information obtained by visible light imaging, as this is an efficient, cheap and non-harmful approach in contrast to data from material-penetrating sensors. In order to train our machine-learning method, we use aligned image pairs showing plants with leaves and the same plants after defoliation. In detail, we model this problem as a domain-transfer task and regard the aligned images containing occluded berries as one domain and images with revealed berries as a second domain. We resort to methods like Pix2Pix (Isola et al., 2017), that uses a conditional generative adversarial network (cGAN) (Mirza and Osindero, 2014) and can learn the described domain-transfer. In contrast to other works, we present a one-step approach that is end-to-end trainable, meaning the positions of the occlusions are identified, and patterns that need to be filled are learned simultaneously. Through the experience the model gains during training, it learns patterns such as grape instances with their appearing shapes, their environment, and where they occur in the image. This knowledge is exploited during the generation step, in which the learned domain-transfer model is applied to images of vines that have not been defoliated to obtain a high-probability and realistic impression of the scene behind the leaves. In order to obtain a berry count, the generated images are further processed with the berry counting algorithm of Zabawa et al. (2020). In this way, we provide a more accurate count of grape berries, since both visible berries and berries potentially occluded by leaves are taken into account.

A major challenge for training is that there is no large dataset of aligned natural images that includes both images with occluded berries and images with berries exposed by defoliation. In addition, in our case, the spatial alignment between the image pairs is not accurate enough since defoliation leads to a resulting movement of branches, grape bunches, and other objects in the non-occluded domain patches. As a result, the natural data is not sufficient to train a model that matches our requirements of a reliable model. Due to this, we propose the use of a synthetically generated dataset that contains paired data of both domains. Our main contributions of this article are:

- The true scenario behind the leaves without defoliation is unlikely to be identified. Therefore, our approach estimates a highly probable scenario behind the occlusions based on visible information in the image, especially of the surroundings of the occlusion, and learned patterns during the training process which include for example the berry shape and neighborhood of berries to obtain a distribution similar to the training data.
- We present a one-step approach, which can implicitly identify which image areas contain visible berries and which areas are occluded without supervision regarding occluded and non-occluded areas. This differs from approaches such as inpainting (Bertalmio et al., 2003; Barnes et al., 2009; Iizuka et al., 2017; Dekel et al., 2018; Liu et al., 2018), in which the occluded areas must be known *a priori*.
- In addition to the acquired images, we use so-called berry masks obtained by the approach presented in Zabawa et al. (2020), which uses semantic segmentation to indicate in the image which pixels belong to `berry`, `berry-edge`, and `background`. During training, this leads to a more stable and easier optimization process. During testing, the berry mask is only needed for the input image since our GAN-based method simultaneously generates the berry mask in which the berries are counted, in addition to the visually generated image.

**Abbreviations:** cGAN, conditional Generative Adversarial Network; SMPH, semi minimal pruned hedge; VSP, vertical shoot position.

[1] Zabawa, L., Kicherer, A., Klingbeil, L., Töpfer, R., Roscher, R., and Kuhlmann, H. (2021). Image-based analysis of yield parameters in viticulture. *Biosyst. Eng.* (under review).

- Since a direct comparison of the true scenario behind the leaves and our generated scenario is not appropriate using standard evaluation methods such as a pixel-by-pixel comparison, we perform a comprehensive evaluation using alternative evaluation metrics, such as generation maps and correlation, that assesses the performance of our approach.
- We show that the application of our approach minimizes the offset compared to the manual reference berry count and the variance, which is not achieved by applying a factor.
- We create various synthetic datasets and show that our approach trained on synthetic data also works on natural data.

The article is structured as follows: After surveying related works, we start by introducing our domain-transfer framework and describe the different components, such as the conditional generative adversarial network, that are used in our approach. We explain the data acquisition and post-processing of the natural and synthetic datasets we use in our work. We explain the evaluation metrics we use and then describe our experiments in which we analyze the generation quality of different synthetic input data, compare generated results with real results in the occluded as well as the non-occluded domain and analyze the berry counting based on the generated results. Finally, we investigate the application of the synthetically learned models to natural data. We end our article with the conclusion and future directions.

## 2. RELATED WORK

### 2.1. Yield Estimation and Counting

Since an accurate yield estimation is one of the major needs in viticulture, especially on a large scale, there is a strong demand for objective, fast, and non-destructive methods for yield forecasts in the field. For many plants, including grapevines, the derivation of phenotypic traits is essential for estimating future yields. Besides 3D-reconstruction (Schöler and Steinhage, 2015; Mack et al., 2017, 2018), 2D-image processing is also a widely used method (Hacking et al., 2019) for the derivation of such traits. For vine, one plant trait that strongly correlates with yield is the number of bearing fruits, that means the amount of berries. This correlation is underlined by the study of Clingeleffer et al. (2001), in which it is shown that the variation of grapevine yield over the years is mainly caused by the berry number per vine (90%).

The task of object counting can be divided into two main approaches: (1) regression (Lempitsky and Zisserman, 2010; Arteta et al., 2016; Paul Cohen et al., 2017; Xie et al., 2018) which directly quantifies the number of objects for a given input, and (2) detection and instance segmentation approaches which identify objects as an intermediate step for counting (Nuske et al., 2014; Nyarko et al., 2018). Detection approaches in viticulture are presented, for example, by Nuske et al. (2011), Roscher et al. (2014), and Nyarko et al. (2018), who define berries as geometric objects such as circles or convex surfaces and determine them by image analysis procedures such as Hough-transform. Recent state-of-the-art approaches, especially segmentation (He et al., 2017), are mostly based on neural networks. One of the earliest works combining grapevine data and neural network analysis was Aquino et al. (2017). They detect grapes using connected components and determine key features based on them, which are fed as annotations into a three-layer neural network to estimate yield. In another work, Aquino et al. (2018) deal with counting individual berries, which are first classified into berry candidates using pixel classification and morphological operators. Afterward, a neural network classifies the results again and filters out the false positives.

The two studies by Zabawa et al. (2019, 2020) serve as the basis for this article. Zabawa et al. (2019) use a neural network which performs a semantic segmentation with the classes `berry`, `berry-edge` and `background`, which enables the identification of single berry instances. The masks generated in that work serve as input for the proposed approach. The article by Zabawa et al. (2020) based on Zabawa et al. (2019) extends identification to counting berries by discarding the class edge and counting the berry components with a connected component algorithm. The counting procedure applied in that work is used for the analyses of the experiments.

### 2.2. Given Prior Information About Regions to Be Transferred

A significant problem in fruit yield estimation is the overlapping of the interesting fruit regions by other objects, like in the case of this work, the leaves. Several works are already addressing the issue of data with occluded objects or gaps within the data, where actual values are missing, which is typically indicated by special values like, e.g., not-a-number. The methodologies can be divided into two areas: (1) there is prior information available about where the covered positions are, and (2) there is no prior information. In actual data gaps, where the gap positions can be easily identified a priori, data imputation approaches can be used to complete data. This imputation is especially important in machine learning, since machine learning models generally require complete numerical data. The imputation can be performed using constant values like a fixed constant, mean, median, or k-nearest neighbor imputation (Batista and Monard, 2002) or calculated using a random number like the empirical distribution of the feature under consideration (Rubin, 1996, 2004; Enders, 2001; von Hippel and Bartlett, 2012). Also, possible are multivariate imputations, which additionally measures the uncertainty of the missing values (Van Buuren and Oudshoorn, 1999; Robins and Wang, 2000; Kim and Rao, 2009). Data imputation is also possible using deep learning. Lee et al. (2019), for example, introduce CollaGAN in which they convert the image imputation problem to a multi-domain image-to-image translation task.

In case there are no data gaps, but the image areas that are occluded or need to be changed are known, inpainting is a commonly used method. The main objective is to generate visually and semantically plausible appearances for the occluded regions to fit in the image. Conventional inpainting methods (Bertalmio et al., 2003; Barnes et al., 2009) work by filling occluded pixels with patches of the image based on low level features like SIFT descriptors (Lowe, 2004). The results of these methods do not look realistic if the areas to be filled are near foreground objects or the structure is too complex. An alternative is deep learning methods that learn a direct end-to-end mapping from masked images to filled output images. Particularly realistic results can be generated using Generative Adversarial Networks (GANs) (Iizuka et al., 2017; Dekel et al., 2018; Liu et al., 2018). For example, Yu et al. (2018) deal with generative image inpainting using contextual attention. They stack generative networks to ensure further the color and texture consistence of generated regions with surroundings. Their approach is based on rectangular masks, which do not generalize well to free-form masks. This task is solved by Yu et al. (2019) one year later by using guidance with gated convolution to complete images with free-form masks. Further work introduces mask-specific inpainting that fills in pixel values at image locations defined by masks. Xiong et al. (2019) learn a mask of the partially masked object from the unmasked region. Based on the mask, they learn the edge of the object, which they subsequently use to generate the non-occluded image in combination with the occluded input image.

## 2.3. No Prior Information About Regions to Be Transferred

Methods that do not involve any prior knowledge about gaps and occluded areas can be divided into two-step and one-step approaches. Two-step approaches first determine the occluded areas, which then are used, for example, as a mask to inpaint the occluded areas. Examples are provided by Yan et al. (2019), which visualize the occluded parts by determining a binary mask of the visible object using a segmentation model and then creating a reconstructed mask using a generator. The resulting mask is fed into coupled discriminators together with a 3D-model pool in order to decide if the generated mask is real or generated compared to the masks in the model pool. Ostyakov et al. (2018) train an adversarial architecture called SEIGAN to first segment a mask of the interesting object, then paste the segmented region into a new image and lastly fill the masked part of the original image by inpainting. Similar to the proposed approach, SeGAN introduced by Ehsani et al. (2018) uses a combination of a convolutional neural network and a cGAN (Mirza and Osindero, 2014; Isola et al., 2017) to first predict a mask of the occluded region and, based on this, generate a non-occluded output.

# 3. MATERIALS AND METHODS

## 3.1. Framework

In our work, we regard the revealing of the occluded berries as a transfer between two image domains. We first detail this and show how we model this transfer for our data. Then we will lay out the cGAN and the framework we use for this task. Finally, we show how we train this network.

### 3.1.1. Domain-Transfer Framework

On a high level, the task of revealing the occluded berries can be described as generating a new impression of an existing image. We model this generative task as a transfer of an existing image from one domain, the source domain, to another domain, the target domain. In our work, we regard images where berries are occluded by various objects as the source domain and call it *occluded domain*. Accordingly, our target domain contains images of defoliated plants, and we call it *non-occluded domain*. Therefore, by performing this domain-transfer, we aim to reveal hidden berries. Samples of both domains are shown at the top of **Figure 1**.

This task can typically be learned by a cGAN, like Pix2Pix in our case. We train this network using aligned pairs of images from the occluded domain and the non-occluded domain and indicate them with $\boldsymbol{x}_{\mathrm{occ}}$ and $\boldsymbol{x}_{\mathrm{non}}$, respectively. The first ones are used as the network input and the latter ones, being the desired output, as the training target. Due to computational limitations, we use cropped patches from the original data and convert them to grayscale to develop an efficient approach that is independent of the berry color. In practice, we accompany the images of each domain with a corresponding semantic mask, that indicates per image pixel the content based on the classes `berry`, `berry-edge`, and `background`. This mask supports the discriminability of relevant information like the berries from the surrounding information in the image and the generation of separated berries, supporting the later counting step. After training, we use the cGAN to generate images, $\widetilde{\boldsymbol{x}}_{\mathrm{non}}$, that we further process with a berry counting method.

Since we only have limited amounts of data available for training and testing, we resort to a dataset consisting of synthetic images for the occluded domain and natural images for the non-occluded domain that we describe in detail in section 3.2. In addition, we test our trained model on fully natural data to analyze the generalizability of the model. For the training set, the non-occluded domain contains natural images, whereas the images from the occluded domain are derived from the former domain, where berries are artificially occluded with leaf templates. To differentiate the different datasets of images, we further qualify the natural images with index N and the synthetic images with index S, which results in the two occluded domain groups: $\boldsymbol{x}^{\mathrm{N}}_{\mathrm{occ}}$ and $\boldsymbol{x}^{\mathrm{S}}_{\mathrm{occ}}$. The generated images are accordingly indicated by $\widetilde{\boldsymbol{x}}^{\mathrm{N}}_{\mathrm{non}}$ and $\widetilde{\boldsymbol{x}}^{\mathrm{S}}_{\mathrm{non}}$. We therefore train the model with input images $\boldsymbol{x}^{\mathrm{S}}_{\mathrm{occ}}$ and use $\boldsymbol{x}^{\mathrm{N}}_{\mathrm{non}}$ as target images. Finally, we apply the model on natural images $\boldsymbol{x}^{\mathrm{N}}_{\mathrm{occ}}$ and compute the berry counts of the generated output images, $\widetilde{\boldsymbol{x}}^{\mathrm{N}}_{\mathrm{non}}$.

### 3.1.2. Conditional Generative Adversarial Networks

The core of our framework is the cGAN that we use to generate images with berries being revealed. Specifically, we use the Pix2Pix (Isola et al., 2017) network and training method, which is illustrated in simplified form in **Figure 2**.

The model consists of two networks, the generator, and the discriminator. The generator network $\mathcal{G}$ takes images with occluded berries as an input and is intended to generate images with revealed berries $\mathcal{G}(\boldsymbol{x}_{\mathrm{occ}}) = \widetilde{\boldsymbol{x}}_{\mathrm{non}}$ that cannot be distinguished from real images $\boldsymbol{x}_{\mathrm{non}}$ of the non-occluded domain. The adversarially trained discriminator network $\mathcal{D}$, on the other side, tries to discriminate between generated images $\widetilde{\boldsymbol{x}}_{\mathrm{non}}$ and real images $\boldsymbol{x}_{\mathrm{non}}$. The generator used in Pix2Pix is based on a U-Net (Ronneberger et al., 2015), the discriminator $\mathcal{D}$ on a PatchGAN.

As described by Goodfellow et al. (2014), both parts of GANs are trained simultaneously using a min-max approach. The goal

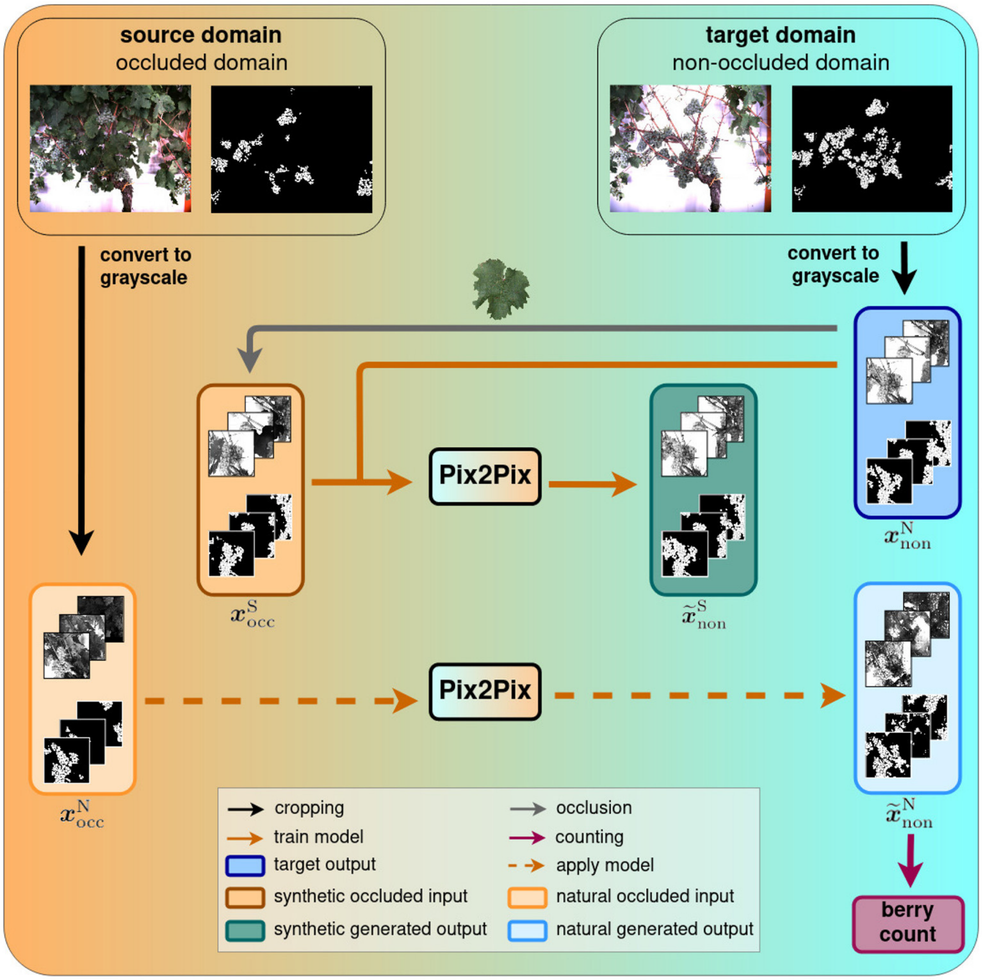


**FIGURE 1 |** Domain-transfer framework. We transfer images from the source domain with occluded berries to the target domain with revealed berries using the Pix2Pix cGAN. We train and test the model on synthetic data and subsequently apply it to natural data. Finally, a berry counting is performed on the generated outputs. Further evaluation steps will be performed in our experiments.

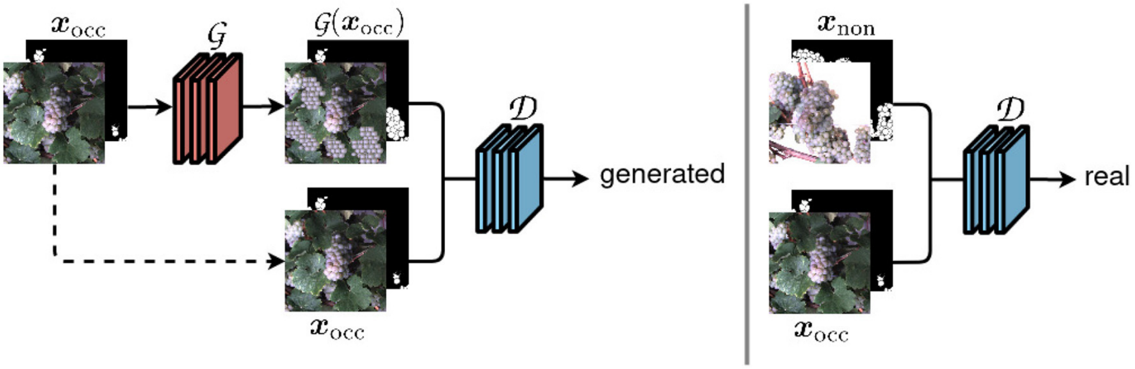


**FIGURE 2 |** Our network structure based on the Pix2Pix framework (Isola et al., 2017). An input image $\boldsymbol{x}_{\text{occ}}$ of the occluded domain is transferred to a non-occluded domain using a generator network $\mathcal{G}$. The discriminator network $\mathcal{D}$ distinguish whether the output of $\mathcal{G}$ looks real or generated.

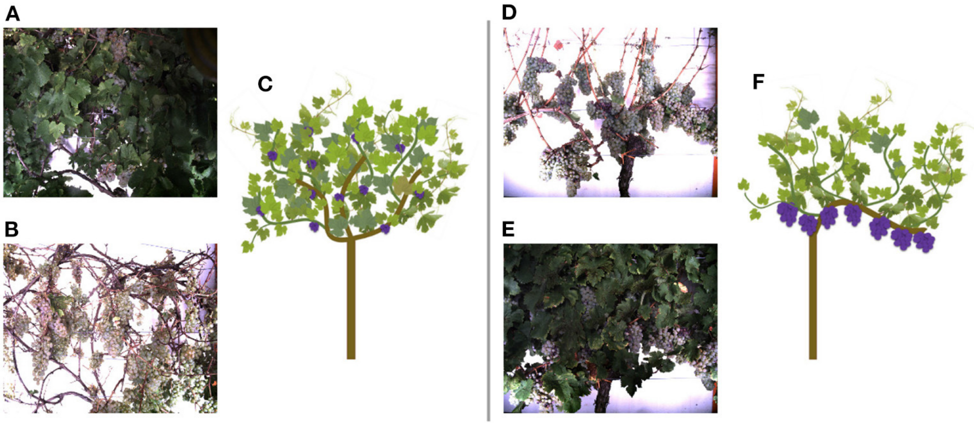


**FIGURE 3 |** Acquired images of the Phenoliner (Kicherer et al., 2017) for two different kind of cuttings: semi minimal pruned hedge SMPH **(A–C)** and vertical shoot positioned system VSP **(D–F)**. **(A,D)** Show example images before defoliation in the occluded domain. **(B,E)** Show example images after defoliation in the non-occluded domain.

of the discriminator during training is to be able to distinguish as good as possible between real and generated images. For this, the discriminator uses a mini-batch of input images $\boldsymbol{x}_{\text{non}}$ and computes the discriminator loss $\ell_{\mathcal{D}_{\text{real}}}$. Additionally, it uses generated images $\widetilde{\boldsymbol{x}}_{\text{non}}$ obtained from the generator $\mathcal{G}$ and computes the corresponding loss $\ell_{\mathcal{D}_{\text{gen}}}$. For both computations, the mean squared error (MSE) loss $\ell_{\text{MSE}}$ is used. The overall loss $\ell_{\mathcal{D}}$ of the discriminator is calculated as:

$$\ell_{\mathcal{D}} = \frac{1}{2} \cdot (\ell_{\mathcal{D}_{\text{fake}}} + \ell_{\mathcal{D}_{\text{real}}}) \qquad (1)$$

The objective is to maximize this loss, as this means that the discriminator can distinguish between generated and real images with ease. The weights of the discriminator network are then updates with respect to this loss.

When generating new images, the generator tries to trick the discriminator at the same time, which is the adversarial part of the network. Compared to the maximization of the discriminator loss, the objective of the generator is to minimize the generator loss $\ell_G$. This is calculated from a combination of MSE loss computed by $\mathcal{D}\left[\mathcal{G}\left(\boldsymbol{x}_{\text{occ}}\right)\right]$ referred to the reference label `generated` and a $\ell_1$ loss, which avoids blurring. The $\ell_1$ loss is computed using real and generated images, $\boldsymbol{x}_{\text{non}}$ and $\widetilde{\boldsymbol{x}}_{\text{non}}$, from the non-occluded domain. The generator loss $\ell_{\mathcal{G}}$ is then used to update the generator's weights.

$$\ell_{\mathcal{G}} = \ell_{\text{MSE}}(\mathcal{D}\left(\mathcal{G}\left(\boldsymbol{x}_{\text{occ}}\right)\right)) + \lambda \cdot \ell_1(\boldsymbol{x}_{\text{non}}, \widetilde{\boldsymbol{x}}_{\text{non}}) \qquad (2)$$

The weighting factor $\lambda$ adjusts the scale of the losses to each other and is, in our case, $\lambda = 100$.

The minimization of the generator loss $\ell_{\mathcal{G}}$ results in either a strong generator or a very weak discriminator. If the loss becomes maximal, the opposite possibilities can occur. The objective is to balance both adversarial goals at the end of the training in the best possible way by realizing both at the same time.

## 3.2. Data

### 3.2.1. Study Site

The data, we use in this work, were acquired at the experimental fields of JKI Geilweilerhof located in Siebeldingen, Germany. It was acquired using the Phenoliner (Kicherer et al., 2017), a reconstructed grape harvester that can be used as a phenotyping platform to acquire geo-referenced sensor data directly in the field. A description of the on-board camera setup can be found in Zabawa et al. (2020). The images were acquired in two different training systems of the cultivar Riesling (DEU098_VIVC10077_Riesling_Weiss_DEU098-2008-085): (1) Vertical shoot positioned (VSP) vines (**Figure 3C**) and (2) vines trained as semi minimal pruned hedges (SMPH) (**Figure 3F**) were chosen due to diverse difficulties in image analysis (Zabawa et al., 2020). The acquisition took place in September 2019 and 2020, before harvest at the plant growth stage BBCH89, and in each year the images were taken 1 day before (**Figures 3A,D**) and right after defoliation (**Figures 3B,E**). In 2019 50 cm and 2020, respectively, 100 cm of the grapevine canopy have been defoliated.

In our framework, we use three different types of inputs:

- **Natural data**: Images acquired in the vineyard before and after defoliation. For our studies, we use grayscale images. We denote this dataset with $X^{\text{N}}$.
- **Synthetic data**: Images acquired in the vineyard after defoliation. Images with occluded berries are synthetically generated. We denote this dataset with $X^{\text{S}}$.
- **Semantic segmentation masks (berry masks)**: So-called berry masks obtained by a semantic segmentation approach presented in Zabawa et al. (2019). Each pixel in these images is assigned to the class `berry`, `berry-edge`, or `background`. We denote this data as $X_{\text{B}}$.

The use of the mentioned grayscale images is indicated by the index G and with index B we denote the use of the berry masks. Moreover, we define $X_{\text{GB}}$ as the input where the grayscale image

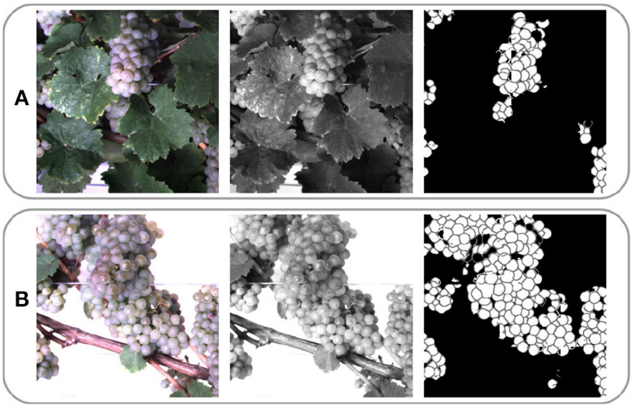


**FIGURE 4 |** Example patches extracted from images of **(A)** the occluded and **(B)** non-occluded domain. One row shows the same patch in RGB, grayscale, and berry mask format. One column represents a patch pair $\{\boldsymbol{x}_{\text{occ}}, \boldsymbol{x}_{\text{non}}\}$.

and the berry mask are stacked to form a multichannel 2D input. In the following, the used data is explained in more detail.

### 3.2.2. Natural Data

We convert the acquired RGB images into grayscale images in order to develop an efficient approach that is independent of the berry color. Covering the whole variability of possible berry colors is complex and not feasible in our case. For example, in the case of green berries, the color also does not serve to differentiate them from leaves.

Since the Phenoliner platform revisits the vine row for each data collection of the two domains, the images depicting the same scene are acquired at different times and from different positions, leading to differences in translation, rotation and scale. Moreover, the defoliation of vines causes a movement of the branches and grape bunches, and additional environmental changes between the two acquisition time points can result in different scenes in the aligned patches.

However, to obtain aligned image pairs for a qualitative evaluation, we manually align images from both domains. For this, we compute a four-parameter Helmert transformation (Helmert, 1880) between the two domains, where we manually define corresponding keypoints per image pair to calculate the parameters. We apply this transformation to images from the non-occluded domain to register them to the occluded domain.

Due to computational limitations, we use a sliding window of size 656 px × 656 px and stride 162 px to extract patches from the grayscale images. **Figure 4** illustrates one RGB patch, the grayscale patch, and the corresponding berry mask, which is explained in the following subsection, for both domains. We denote the aligned patch pair $\boldsymbol{x}^{\text{N}} = \{\boldsymbol{x}^{\text{N}}_{\text{occ}}, \boldsymbol{x}^{\text{N}}_{\text{non}}\}$, where $\boldsymbol{x}^{\text{N}} \in X^{\text{N}}$.

### 3.2.3. Semantic Segmentation Mask (Berry Mask)

Besides the acquired images, we use a berry mask, obtained with a semantic segmentation approach, presented by Zabawa et al. (2019). The identification of regions containing berries and the detection of single berry instances is performed with a convolutional neural network. The network uses a MobilenetV2 (Sandler et al., 2018) encoder and a DeepLabV3+ decoder (Chen et al., 2018). The network assigns each image pixel to one of the classes `background`, `berry-edge`, or `berry`, which corresponds to the grayscale values 0, 127, and 255. In contrast to a standard semantic segmentation without distinguishing between different instances, we use the additional class `berry-edge` to ensure the separation of single berries, which allows the counting of berries using a connected component approach.

For our task of generating a highly probable scenario behind leaves, the berry mask supports the discriminability of relevant information like the berries from the surrounding information in the image, and the generation of separated berries. In addition, since the berry masks contain a masking of existing berries, it provides further knowledge about which areas in the images do not show occlusions and should be preserved in the revelation process and where potentially occlusions might appear, which are areas that are unmasked.

Since we are interested in scenes in the image that depict berries, we only integrate patch pairs in training and testing, whose berry mask of the non-occluded domain contains more than 1/24 background pixels and mask of the occluded domain contains at least one pixel whose class differs from the background class.

### 3.2.4. Synthetic Data

One challenge for our application is that the amount of paired data from both domains containing both, occluded and non-occluded regions of berries, is limited for training a reliable model and for evaluation. We, therefore, resort to generate artificially modified images, where berries are artificially occluded, based on natural images of defoliated plants. This allows us to generate a large dataset to ease the described lack of natural images of both domains. We denote this synthetic dataset with $X^{\text{S}}$. The natural patches $\boldsymbol{x}^{\text{N}}_{\text{non}}$ of the non-occluded domain serve as a basis. We create paired patches $\{\boldsymbol{x}^{\text{S}}_{\text{non}}, \boldsymbol{x}^{\text{S}}_{\text{occ}}\}$ where $\boldsymbol{x}^{\text{S}}_{\text{non}} = \boldsymbol{x}^{\text{N}}_{\text{non}}$.

To generate $\boldsymbol{x}^{\text{S}}_{\text{occ}}$, we apply artificial data modification on both training and test data. We artificially occlude the patches using 24 different wine leaves (**Figure 5B**) with various shapes extracted from the natural dataset and use them as occluding objects in the patches. We use 18 leaves for augmenting the training set and six leaves to augment the test set. On the basis of one image patch $\boldsymbol{x}^{\text{S}}_{\text{non}}$, we create up to nine corresponding synthetically augmented versions of $\boldsymbol{x}^{\text{S}}_{\text{occ}}$ for the training set, resulting in nine aligned image patch pairs. During the procedure, a leaf is randomly selected from the set of leaves and rotated by a randomly chosen angle $\alpha \in \{-50, -30, -10, 0, 10, 30, 50, 70\}$. Converted to grayscale, it randomly overlays the grayscale patch and occludes parts of the visible berries. These steps are also performed for patches $\boldsymbol{x}^{\text{S}}_{\text{non}}$ of the test set. However, here only three new patch pairs are created. After applying artificial data modification, the proportion of test data amounts to ∼18–23% of the extracted patches depending on the type of defoliation (see **Figure 6B**). The split of the data into training and test data is illustrated visually and numerically in **Figure 6**.

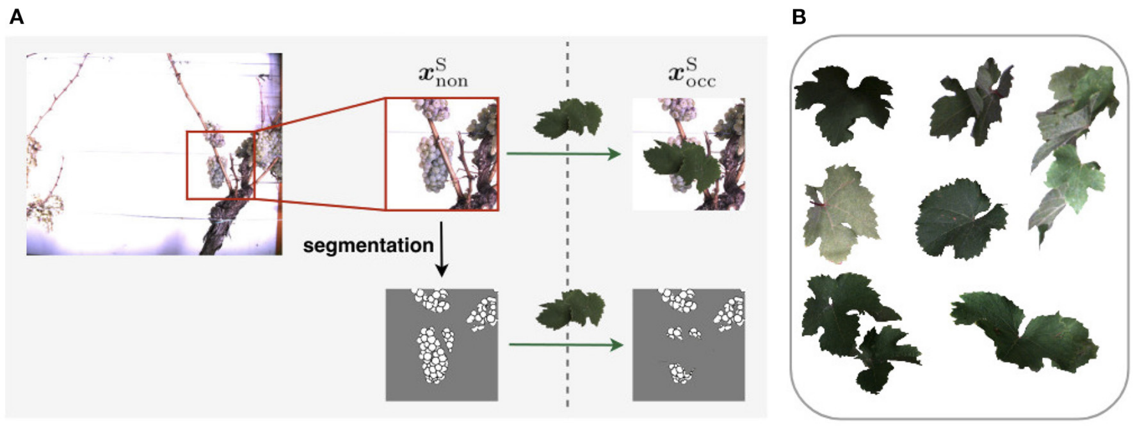
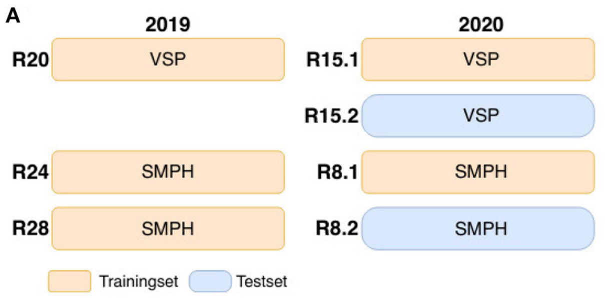

A
B
$x^{\mathrm{S}}_{\mathrm{non}}$
$x^{\mathrm{S}}_{\mathrm{occ}}$
segmentation

**FIGURE 5 | (A)** Visualization of synthetic data creation. The visualization indicates the use of an artificial leaf to calculate the corresponding mask of $x^{\mathrm{S}}_{\mathrm{occ}}$ of the occluded domain instead of using the segmentation mask prediction based on the occluded RGB-image.Exemplary leaves used for data augmentation. **(B)** Shows exemplary leaves used for data augmentation.



B

| | Training | Testing | Sum |
|---|---|---|---|
| SMPH | 25.209<br>81.39% | 5.763<br>18.61% | 30.972<br>100% |
| VSP | 15.996<br>77.82% | 4.560<br>22.18% | 20.556<br>100% |

**FIGURE 6 |** Dataset composition. **(A)** Shows the visual division of training and test data for the two data sets semi minimal pruned hedge (SMPH) (see **Figure 3C**) and vertical shoot positioned system (VSP) (see **Figure 3F**) which were collected in the years 2019 and 2020. The data sets are taken from different rows R. Each row corresponds to one of the two sets. In 2020 the same rows were run twice. Once left (R15.1 and R8.1) and once right (R15.2 and R8.2) of the row. Orange marked areas are used for training, blue marked ones for testing. The table shown in **(B)** indicates the corresponding numbers of training and test data patches.

The test data is taken from the dataset collected in 2020 (see **Figure 6A**).

For each synthetic grayscale image, we calculate a corresponding berry mask. However, depending on the used procedure, the appearance of the berry mask differs. In our work, we create the masks for the two domains, as illustrated in **Figure 5A**. The mask of the non-occluded patch $x^{\mathrm{S}}_{\mathrm{non}}$ is based on the segmentation step, described in Section 3.3.2, which needs RGB images as input. We compute the mask of $x^{\mathrm{S}}_{\mathrm{occ}}$ by overlaying the pixels of the non-occluded mask of $x^{\mathrm{S}}_{\mathrm{non}}$ that are covered with a leaf in the RGB, or respectively grayscale patch. These pixels in the berry mask are assigned to the class `background`. The leaf pixels adjacent to berry pixels are changed to `berry-edge` pixels. In this way, the overlapped berries have a closed contour. By adding these edges, the synthetic data thus has the same characteristics as berry masks derived from the natural data. With this step, we create two corresponding masks, $x^{\mathrm{S}}_{\mathrm{occ}}$ and $x^{\mathrm{S}}_{\mathrm{non}}$, which match exactly in the non-occluded pixel.

Another way to define the occluded mask is a direct computation as for $x^{\mathrm{S}}_{\mathrm{non}}$ using the segmentation step to create a predicted mask of the patch. Since the berry mask is an estimation, the class of individual non-occluded pixels may differ between $x^{\mathrm{S}}_{\mathrm{non}}$ and $x^{\mathrm{S}}_{\mathrm{occ}}$. For a simplified analysis, we have chosen the first option.

Overall, for dataset ${}^{VSP}X^{S}$, we obtain 20.556 synthetic patch pairs, and for dataset ${}^{SMPH}X^{S}$, we obtain 30.972 synthetic patch pairs **Figure 6B**.

### 3.2.5. Challenges

Various challenges occur in the data, which influence our training and thus our results. Since our reference masks are not manually derived but are estimations, uncertainties can occur. For example, not all visible berries are entirely shown in the images of the non-occluded domain. Therefore, it can happen that either berries are missed or only partly detected in the mask. Additionally, the estimated contour in the berry mask may not be closed and parts of the berry region may be classified as background. Thus, these errors in the reference could be learned in the model. Furthermore, there are images in the non-occluded domain, which contain leaves despite defoliation. In an ideal case, the model learns to ignore these faults in defoliation. Other challenges are the varying sharpness of the patches. This can be caused by resizing the data, shadows, or the varying distance of the berries to the camera. Furthermore, the illumination varies within the data, e.g., due to the coverage by surrounding objects like branches or leaves or the distance of the berries to the camera. Also, worth noting are the different growth stages of the grapes in 2019 and 2020, so the grapes have different sizes due to different berry sizes.

## 3.3. Model Evaluation

### 3.3.1. Data Post-processing

After the test phase, the generated masks do not only contain the values 0, 127, and 255. There are also mixed pixels that are not clearly assigned to one of the three classes. We use thresholding to ensure that only the values 0, 127, and 255 appear in the mask. We use the following class assignment.

- Pixel values in the interval [0, 50] are set to value 0 and assigned to class `background`.
- Pixel values in the interval [50, 180] are set to value 127 and assigned to class `berry-edge`.
- Pixel values in the interval [180, 255] are set to value 255 and assigned to class `berry`.

### 3.3.2. Evaluation Metrics

In the following, we describe several evaluation metrics used for our experiments. The first metric we use is the *area* $F_c$, that we define as the number of pixels within a mask that correspond to a class $c$ with $c \in \{\texttt{background}, \texttt{berry-edge}, \texttt{berry}\}$. With area $F_c$ and the generated area $\widetilde{F}_c$, which is based on the generated mask of the cGAN, we calculate the *intersection over union* IoU by dividing the area of overlap by the area of union.

$$\text{IoU}_c = \frac{F_c \cap \widetilde{F}_c}{F_c \cup \widetilde{F}_c} \tag{3}$$

The IoU compares the similarity between two arbitrary shapes.

The second metric we use is the *pearson product-moment correlation coefficient*. It gives a measure of the degree of linear relationship between two variables. The correlation coefficient is obtained by the correlation coefficient matrix Q, which is calculated by means of the covariance matrix C,

$$\text{Q}_{i,j} = \frac{\text{C}_{i,j}}{\sqrt{\text{C}_{i,i} \cdot \text{C}_{i,j}}} \tag{4}$$

where $i$ and $j$ indicate the row and column index, respectively. The values of Q are between $-1$ and 1, inclusive. The correlation coefficient $\rho$ between two variables can then be expressed by $\rho = \text{Q}_{0,1}$. A correlation coefficient $\rho$ equals 1 indicates that both input variables are equal. We use the correlation to compare the generated images $\widetilde{\boldsymbol{x}}_{\text{non}}$ from the model with the input $\boldsymbol{x}_{\text{occ}}$ as well as the target output $\boldsymbol{x}_{\text{non}}$ on pixel level.

The *coefficient of determination*, also denoted by $R^2$, indicates the relationship between a predicted value with respect to a reference value. It provides a measure of how well-observed references are replicated by the model. In our case, we use the $R^2$ value for the comparison between the predicted number of berries generated by the model and the reference number from the berries manually counted in the non-occluded domain. Plots, as illustrated in **Figure 10**, represent the generated distribution of the model compared to the reference. Please note, that the gray line represents the reference values. The optimal generated samples are distributed along this line, reflected in a $R^2$ value equal to 1.

The *counting* is based on the procedure described in the work of Zabawa et al. (2020). The counting is performed based on the masks, which are predicted with the convolutional neural network presented in their work. The classes `background` and `berry-edge` are discarded, and the counting is solely performed with pixels of the class `berry`. Before counting the number of connected components of the berry mask, we introduce geometrical and qualitative filter stages to improve the count. Filtering follows the observations of Zabawa et al. (2020) (Table 3) which show that when the filter is applied, the misclassifications for VSP cutting decrease by 9% and for SMPH cutting by 11%. For the first step of filtering, we discard elements that are smaller than 25 pixels, since these artifacts are too small to represent berries. Secondly, we exploit the knowledge that berries are roughly round by removing objects with a minor-major-axis ratio below 0.3 and an insufficient area. The actual area of each component is compared to the expected area based on a radius, which is computed as the mean of the minor and major axis of the component. Lastly, we check how well each object is surrounded by an edge, since most high confidence predictions are well surrounded by an edge. For further details, we refer the reader to Zabawa et al. (2019).

Another metric we use for a visual comparison is the *generation map*. Generation mapping is used to visualize the differences between two masks. In our case the distances are calculated between (1) the input mask $\boldsymbol{x}_{\text{occ}}$ and the generated mask $\widetilde{\boldsymbol{x}}_{\text{occ}}$ (**Figure 7A**), (2) the target output mask of $\boldsymbol{x}_{\text{non}}$ and the generated mask of $\widetilde{\boldsymbol{x}}_{\text{non}}$ (**Figure 7B**), and lastly (3) the target output mask of $\boldsymbol{x}_{\text{non}}$ and the generated mask of $\widetilde{\boldsymbol{x}}_{\text{non}}$ including only two classes, where `berry` and `berry-edge` are considered as one class (**Figure 7C**). We denote this mask as binary mask.

The different colors allow us to make a statement about the area in which, for example, berries are generated where none are present in the reference. The colors can be analyzed as follows: For **Figures 7A,B**, at pixel positions with a medium orange and medium blue discoloration, either the class berry is predicted to be an edge or the class edge is predicted to be a berry. These two cases are acceptable for our task, since we do not want to map the reference, but generate images, which provide highly probable results with a distribution that matches the input. The other pixel values are to be avoided, since at these positions for a light and dark orange discoloration the classes `berry` and `berry-edge` are generated, where in the reference `background` occurs. At the positions with a light and dark blue discoloration the class

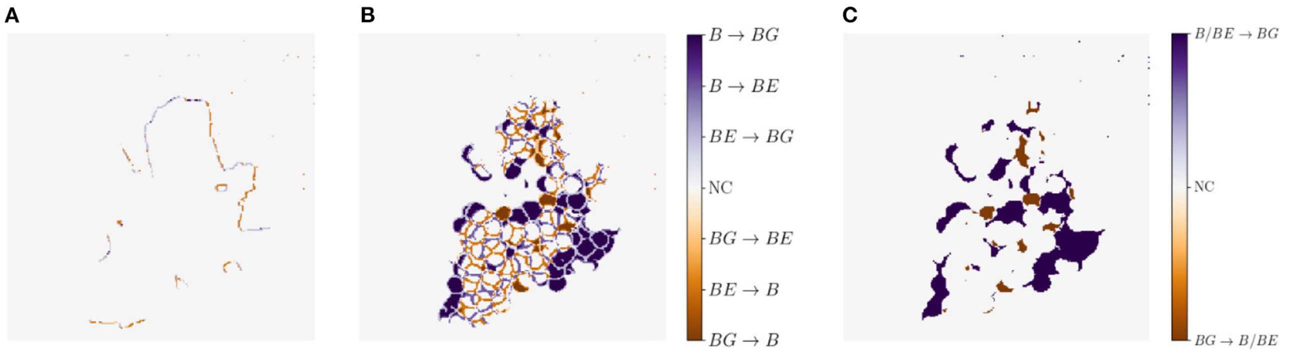


**FIGURE 7 |** Example representation of generation maps in **(A)** the occluded domain, and **(B,C)** the non-occluded domain for the berry mask **(B)** and a binary mask **(C)** where classes `berry` *B* and `berry-edge` *BE* are combined to one class. `Background` is indicated by *BG* and if there is no class change it is indicated by *NC* for no change.

`background` is generated, where in the reference the class `berry` or `berry-edge` is present. The generation map, where only two classes are included, highlights the non-acceptable pixel regions in the generated map.

# 4. RESULTS

## 4.1. Experimental Setup

Our experiments are designed to apply a domain-transfer using cGANs (section 3.1.2) to (1) learn a distribution by which we can generate a highly probable scenario of how occluded grapes could look like depending on the input, and (2) improve the counting of grapevine berries in images. To address the challenge of limited amount of natural data $X^{\mathrm{N}}$, we perform four experiments based on a synthetic dataset $X^{\mathrm{S}}$. In Experiment 5 (section 4.6), we show the applicability to natural data $X^{\mathrm{N}}$ based on the models and results learned in earlier experiments.

For our experiments, we define five different datasets, which are listed in **Table 1**. In addition to the natural data, described in section 3.3.1, we introduce a synthetic dataset in section 3.3.3. All five datasets, *Dataset 1-5*, will again be divided into the different types of defoliation SMPH and VSP. For our experiments, we also distinguish the set of input channels used. We claim that using a combination of grayscale image (G) and berry mask (B), denoted as GB, gives more accurate results both visually and in respect to berry counting than using the berry mask alone without grayscale information. We support this claim in Experiment 1. In the following experiments, the datasets are accordingly used with GB channels.

We resize all image patches to a uniform size of 286 × 286 px with nearest neighbor interpolation. During training, we follow the procedure of Isola et al. (2017) and add small variations to the data in each epoch by randomly cropping patches of size 256 × 256 px from the given patches. Additionally, patches are randomly flipped vertically, and the values within the patches are scaled and shifted to the range $[-1, 1]$. For testing, only scaling and shifting of the values to the range [-1,1] is carried out. The network output is scaled back to the value range [0, 255] for visualization.

**TABLE 1 |** Definitions of the used datasets.

| Definition | *N* | S | SMPH | VSP | GB | B | Experiment | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | 1 | 2 | 3 | 4 | 5 |
| Dataset 1 | | ✗ | | ✗ | ✗ | | ✗ | ✗ | ✗ | ✗ | |
| Dataset 2 | | ✗ | | ✗ | | ✗ | ✗ | | | | |
| Dataset 3 | | ✗ | ✗ | | ✗ | | | | | ✗ | |
| Dataset 4 | ✗ | | | ✗ | ✗ | | | | | | ✗ |
| Dataset 5 | ✗ | | ✗ | | ✗ | | | | | | ✗ |

*The table shows which kind of data is used for which experiment.*

To train the models, we use an Intel Core i7-6850 K 3.60 GHz processor and two GeForce GTX 1080Ti with 11 GB RAM. The models are trained over 600 epochs. We use the Adam optimizer, where the learning rate is constant at 0.0004 for the first 300 epochs and is reduced linearly toward 0 for the last 300 epochs.

## 4.2. Experiment 1—Comparison of Generation Quality Based on GB and B Data

With the first experiment, we analyze how the grayscale channel influences (i) the reproduction of hidden berries and (ii) the counting of berries per image. With the help of the grayscale channel G, it is possible to derive information about the presence of objects such as berries, leaves, and branches. Theoretically, this information helps to identify positions in the image where berries might be generated, for example, behind leaves or branches. In practice, however, in the non-occluded reference, a part of the berries is not present, since a proportion of berries is still occluded due to leaves or bigger branches not being cut away. This makes training more difficult, since it is generally learned that new berries should not be generated at the position of branches that have not been cut away. This implies, that we cannot expect to make new berries visible in the generated output $\widetilde{x}_{\text{non}}$ while testing, that are never present in the reference data $x_{\text{non}}$ of the training set.

To get further insights into this, we analyze whether ignoring the G channel leads to a generation of berries in areas such as

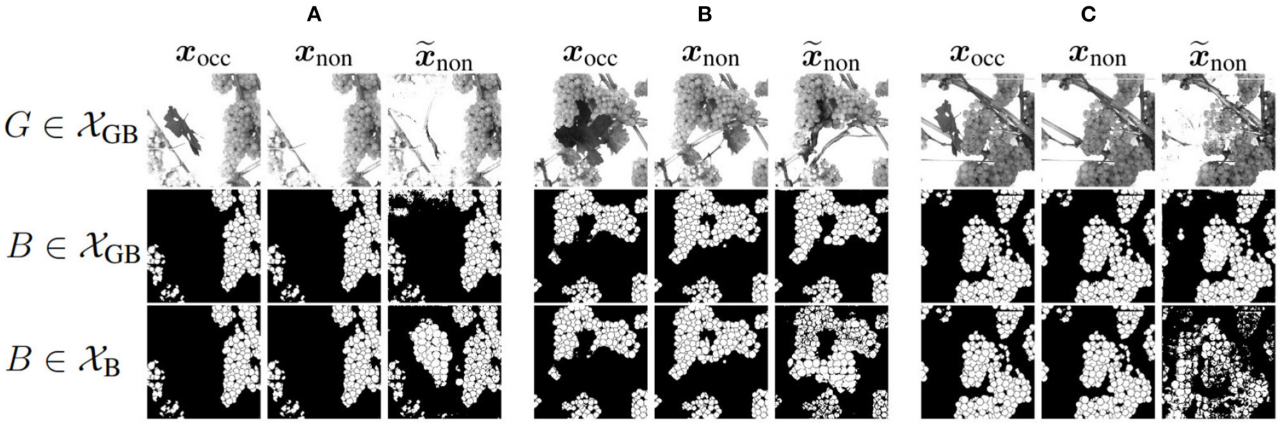

A
B
C
$x_{\text{occ}}$
$x_{\text{non}}$
$\widetilde{x}_{\text{non}}$
$G \in \mathcal{X}_{\text{GB}}$
$B \in \mathcal{X}_{\text{GB}}$
$B \in \mathcal{X}_{\text{B}}$

**FIGURE 8 |** Visual representation of generated test results $\widetilde{x}_{\text{non}}$ in the non-occluded domain of *Dataset 1* including GB channel and *Dataset 2* including only B channel in comparison to reference target output $x_{\text{non}}$ of the non-occluded domain and input $x_{\text{occ}}$ of the non-occluded domain. Three example **(A–C)** are shown. The first row shows the G channel of $\mathcal{X}_{\text{GB}}$. The second row illustrates the mask B of $\mathcal{X}_{\text{GB}}$. The last row represents the corresponding result of $\mathcal{X}_{\text{B}}$.

branches. Moreover, we investigate if using channel B only is better suited on natural data, because information such as color, exposure, and lighting conditions have no influence. Thus, this experiment determines that the G channel adds value to the experiments and shows what this added value looks like.

### 4.2.1. Used Data, Model, and Evaluation Metrics

For this experiment, we train a cGAN model on each of the training sets of *Dataset 1* and *Dataset 2*. The evaluation is based on the corresponding test sets. Since we want to determine the value of the G channel with this experiment, we limit the used data exclusively to defoliation type VSP. SMPH type shows proportionally similar outcomes to the VSP results.

To compare the two datasets $X_{\text{B}}$ and $X_{\text{GB}}$, we use the described metrics in Section 3.4.2. We compare the correlation and the IoU in the occluded domain between the input $x_{\text{occ}}$ and the generated input $\widetilde{x}_{\text{occ}}$, as well as in the non-occluded domain between the target output $x_{\text{non}}$ and the generated output $\widetilde{x}_{\text{non}}$ for both datasets. The generated input $\widetilde{x}_{\text{occ}}$ is computed by taking the generated output $\widetilde{x}_{\text{non}}$ and occlude the same pixels in the berry mask which are occluded in the input by a synthetic leaf.

### 4.2.2. Results

**Figure 8** shows three example results to visually compare $X_{\text{B}}$ and $X_{\text{GB}}$. The first two columns of an example show the reference of the two domains, where the third column represents the generated output $\widetilde{x}_{\text{non}}$. The first row shows the grayscale channel of GB, the second row shows the mask channel of GB, and the bottom row shows the mask channel of B.

Using data without the G channel leads to higher generalizability regarding different varieties such as color, lighting conditions, and occlusions. Remarkable for the mask of B (row 3) is that for input patches containing many berries, proportionally too large and therefore too few berries are generated in $\widetilde{x}_{\text{non}}$ of the test results. This applies to the entire dataset and is demonstrated by **Figures 8A,B**. Generated berries in $\widetilde{x}_{\text{non}}$ of $X_{\text{GB}}$ adapt better to existing berries in mask $x_{\text{occ}}$ than $\widetilde{x}_{\text{non}}$ of $X_{\text{B}}$. Furthermore, it turns out that the model trained on $X_{\text{B}}$ has problems in generating patches with many berries. The berries are not only too big, but also in general berries are difficult to represent in their shape, as seen in **Figures 8B,C**.

Another positive aspect of $X_{\text{GB}}$ is the already mentioned point that background information of the grayscale patch is included in the generation of new berries. The model learns to recognize where background is present in the patch and thus does not generate new berries in $\widetilde{x}_{\text{non}}$ in contrast to the model trained on $X_{\text{B}}$. This is particularly obvious in Example 1 (see **Figure 8A**), where a whole grape bunch is generated in the center of the mask. In the reference input and output of G, it is visible that on this position, background occurs.

In the following, we will take a look at the objective metrics described above. If we compare them regarding the $X_{\text{GB}}$ and $X_{\text{B}}$ input, we notice that the results for correlation between $x_{\text{occ}}$ and $\widetilde{x}_{\text{occ}}$ are similar (see **Figure 9A**). For $X_{\text{B}}$, there are more generated patches with a correlation smaller than 0.8 and, therefore, less with a higher correlation. The correlation histogram between $x_{\text{non}}$ and $\widetilde{x}_{\text{non}}$, shown in **Figure 9B**, shows different distributions for the datasets. While the correlation histogram of BG, presented in orange, shows a left-skewed distribution, the amount of test patches of B increases on average with increasing correlation. At a correlation in the interval of [0.99, 1], represented by the right bar, the distribution shows a striking peak. However, there is a larger proportion of values below a correlation of 0.85. Even in the interval [0.85, 0.99], the percentages of patches for GB are higher than for B.

**Figures 9B**, **10C** present a counting comparison of the different models in the non-occluded domain using a $R^2$-Plot. Additionally, **Figure 9A** shows the counting results without domain-transfer, i.e. no additional generated berries. Counting applied to the target $x_{\text{non}}$ in the non-occluded domain serves

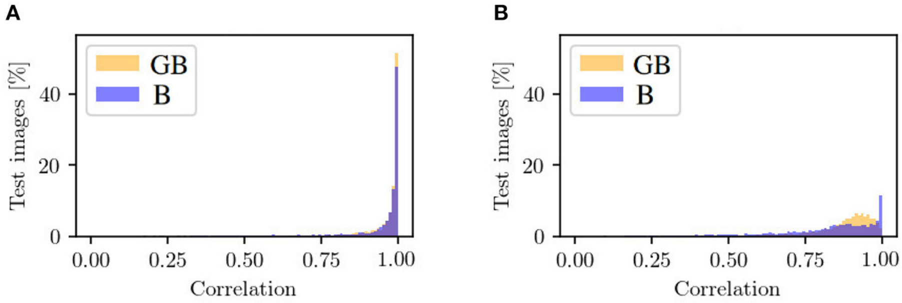


**FIGURE 9 |** Mask-based comparison of GB and B data with respect to **(A)** the correlation between input patch $\boldsymbol{x}_{\text{occ}}$ and generated input patch $\tilde{\boldsymbol{x}}_{\text{occ}}$ in the occluded domain, and **(B)** the correlation between target output $\boldsymbol{x}_{\text{non}}$ and generated output $\tilde{\boldsymbol{x}}_{\text{non}}$ in the non-occluded domain. Shown is the percentage of test images that are assigned to a specific range of correlation. One bar corresponds to the range of 0.01.

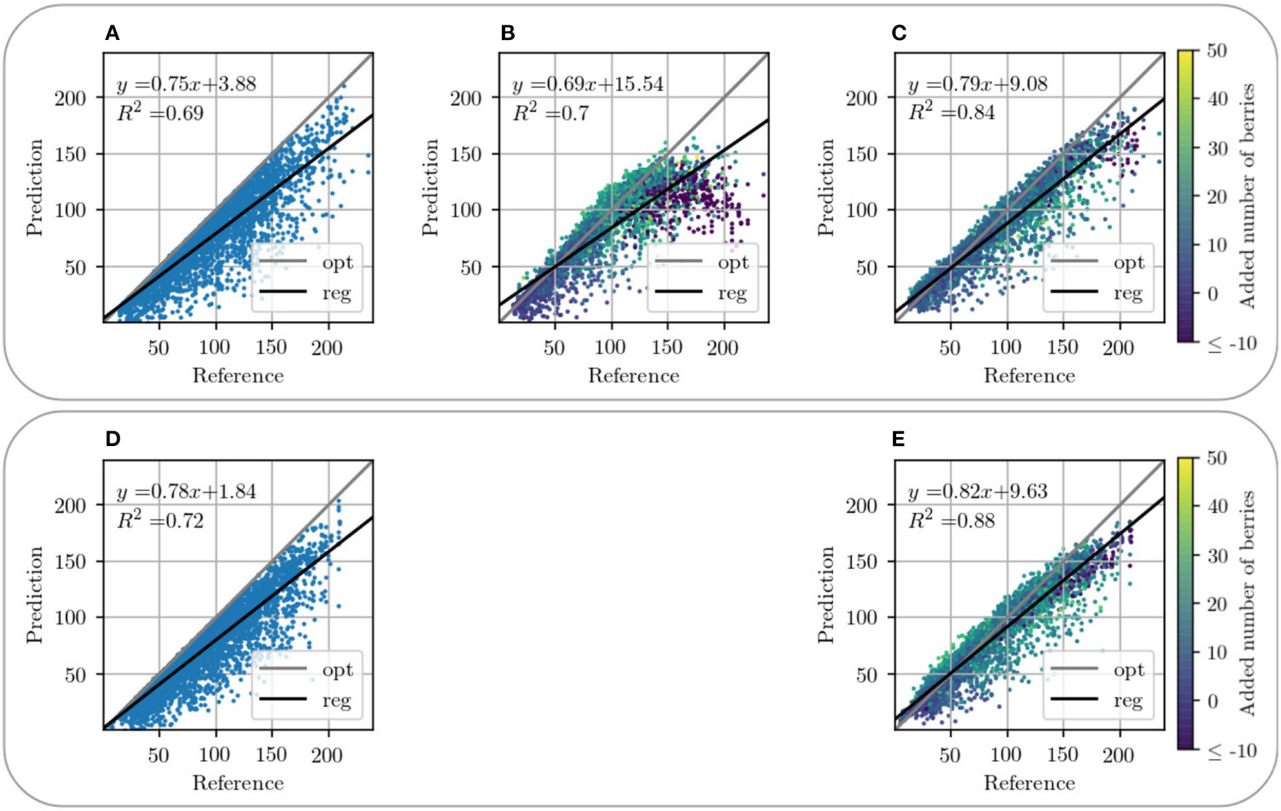


**FIGURE 10 |** Graphical visualization of the berry counting by a $R^2$-Plot. Row 1 illustrates the results for VSP defoliation while row 2 illustrates the results for SMPH defoliation. **(A,D)** Show the input of the occluded domain and **(B,C,E)** of generated berries in the non-occluded domain. Shown is the relation between the generated output $\tilde{\boldsymbol{x}}_{\text{non}}$ in relation to reference $\boldsymbol{x}_{\text{non}}$ for data with input channel B **(B)** and GB **(C,E)**. The coloration of the data points in the plots **(B–D)** indicates the added number of berries compared to the non-occluded domain.

as the counting reference and is represented by the diagonal gray line. We observe that the results with input GB give the best matched results with respect to the reference. This is indicated visually as well as by the $R^2$ value of the different models, which is the highest for our approach in the non-occluded domain with input GB. As in the visual evaluation, the counting plot for input B in **Figure 10B** shows that the model indicates problems generating berries with a larger number of berries per patch. Also in the GB results, we observe that, especially with a reference counting number of more than 150 berries, the model does not reach the reference.

## 4.3. Experiment 2—Real vs. Generated Results in the Occluded Domain

In this experiment, we investigate whether the regions showing berries in the occluded domain stay unchanged in the transferred non-occluded domain. Furthermore, we verify that new berries are generated exclusively in the occluded area, and thus, the model detects where the appearance of berries is very likely.

### 4.3.1. Used Data, Model, and Evaluation Metrics

For this experiment, we use synthetic *Dataset 1* of the VSP defoliation. For evaluation, we use different masks: The first mask is the so-called generated input mask $\widetilde{\boldsymbol{x}}_{\text{occ}}$, for which we take the generated output $\widetilde{\boldsymbol{x}}_{\text{non}}$ of the test set and overlay it with the leaf used for data augmentation of the synthetic input $\boldsymbol{x}_{\text{occ}}$. The other mask is the so-called baseline mask $\boldsymbol{x}_{\text{non,leaf}}$ of this experiment. For this purpose, we use the target output $\boldsymbol{x}_{\text{non}}$ and overlay it likewise with the leaf used for data augmentation of the synthetic input $\boldsymbol{x}_{\text{occ}}$. Thus, only the non-occluded pixels of $\boldsymbol{x}_{\text{occ}}$ will remain visible in $\widetilde{\boldsymbol{x}}_{\text{occ}}$ and $\boldsymbol{x}_{\text{non,leaf}}$. The evaluation is then performed on the pairs $\{\boldsymbol{x}_{\text{occ}}, \boldsymbol{x}_{\text{occ,leaf}}\}$ and $\{\boldsymbol{x}_{\text{occ}}, \widetilde{\boldsymbol{x}}_{\text{occ}}\}$.

We use IoU and correlation as comparative metrics for this experiment. Additionally, we create generation maps which show the differences between the masks within each of the pairs $\{\boldsymbol{x}_{\text{occ}}, \boldsymbol{x}_{\text{occ,leaf}}\}$ and $\{\boldsymbol{x}_{\text{occ}}, \widetilde{\boldsymbol{x}}_{\text{occ}}\}$, as illustrated in **Figure 11**. For this experiment, the first three rows are of interest to us. The first row shows the respective grayscale patch of the generation maps. The second row shows the differences within the pair $\{\boldsymbol{x}_{\text{occ}}, \boldsymbol{x}_{\text{occ,leaf}}\}$. Row three shows the differences within the pair $\{\boldsymbol{x}_{\text{occ}}, \widetilde{\boldsymbol{x}}_{\text{occ}}\}$. The columns indicate different patch examples.

### 4.3.2. Results

The reference correlation within the mask pair $\{\boldsymbol{x}_{\text{occ}}, \boldsymbol{x}_{\text{occ,leaf}}\}$ is above 0.98 for all test patches. With our method, we manage to achieve a correlation of over 0.98 within the pair $\{\boldsymbol{x}_{\text{occ}}, \widetilde{\boldsymbol{x}}_{\text{occ}}\}$ for about 65% of the test images (see **Figure 9A**, orange). The remaining 35% are largely distributed over a correlation within the interval [0.75, 0.98]. The correlation strongly correlates with the IoU calculation of the `berry` area. The low correlations are either due to artifacts in the generated masks or to test images with a high number of berries. In this case, the model does not transfer all non-occluded pixels one to one into the non-occluded domain. The effect of the amount of berries in the patch is shown in the generation maps in **Figure 11** {column 1, row 3} and {column 3, row 3}.

For the patch examples in columns 2, 4, and 5, the generation maps of the pairs $\{\boldsymbol{x}_{\text{occ}}, \boldsymbol{x}_{\text{occ,leaf}}\}$ are almost identical to the generation maps of the pairs $\{\boldsymbol{x}_{\text{occ}}, \widetilde{\boldsymbol{x}}_{\text{occ}}\}$. Such maps correspond to correlation values close to 1. It is noticeable that in all five examples, the border of the leaf used for data augmentation is highlighted in the generation maps. The coloring occurs at transitions between the leaf and the adjacent `berry-edge` and `berry` pixel.

## 4.4. Experiment 3—Real vs. Generated Results in the Non-occluded Domain

In this experiment, we investigate the similarity of our generated output $\widetilde{\boldsymbol{x}}_{\text{non}}$ compared to the target output $\boldsymbol{x}_{\text{non}}$ in the non-occluded domain.

### 4.4.1. Used Data, Model, and Evaluation Metrics

In this experiment, *Dataset 1* is used to train the model. Since we are aiming only for a highly probable result rather than the exact position and shape of specific berries, for our evaluation, we additionally create a binary mask based on the berry mask, which includes only the classes `berry` and `background`. For this, we merge the classes `berry` and `berry-edge`. We compare the mask pair $\{\boldsymbol{x}_{\text{non}}, \widetilde{\boldsymbol{x}}_{\text{non}}\}$ of the non-occluded domain in respect to the berry and binary mask. We evaluate the correlation and IoU within this pair. Furthermore, we create generation maps that illustrate the difference between this pair. Exclusively for the berry mask, we calculate the area and diameter of all individual berries in the entire test data set.

### 4.4.2. Results

The correlation (**Figure 12A**) shows a similar left-skewed distribution for berry mask and binary mask. The majority of the test images show a correlation of above 0.8. Although our approach does not aim to generate the exact position and shape of berries, the results indicate that the similarity of the generated results and the reference are high. The IoU in **Figure 12B** also supports this finding. The IoU of the binary mask has on average higher values and is closer to the possible maximum than the berry mask. The generation maps from **Figure 11** also show this property in the fourth and fifth row. The fourth row shows example results for the berry mask, where two cases can be seen. *Case 1*: The medium orange and medium blue colors in the fourth row illustrate pixels where the classes `berry` and `berry-edge` are confused. This incorrect generation is acceptable due to the desired property of highly probable results instead of exactly matching results. *Case 2*: Dark and light blue, and dark and light orange are incorrectly generated classes that need to be avoided. In the fifth row, these pixel regions are highlighted by dark blue and dark orange. These regions either represent berries where there are no berries in the reference, or *vice versa*. Such incorrect generations shift the position and size of the grape bunches. In the example maps, however, it can be seen that *Case 1* occurs predominantly. It is obvious that berries are predicted in the right areas, but their shape and position do not correspond exactly to the reference.

At the transition from image areas with berries to `background` pixels, the second case occurs where too small or too large grape bunches are produced, because either too few or too many berries are generated. This is illustrated by the second and fourth column. The generation maps of the binary masks only highlight the areas that contradict the property of highly probable results.

To further check the similarity between generated and reference data, we consider the distributions for area and

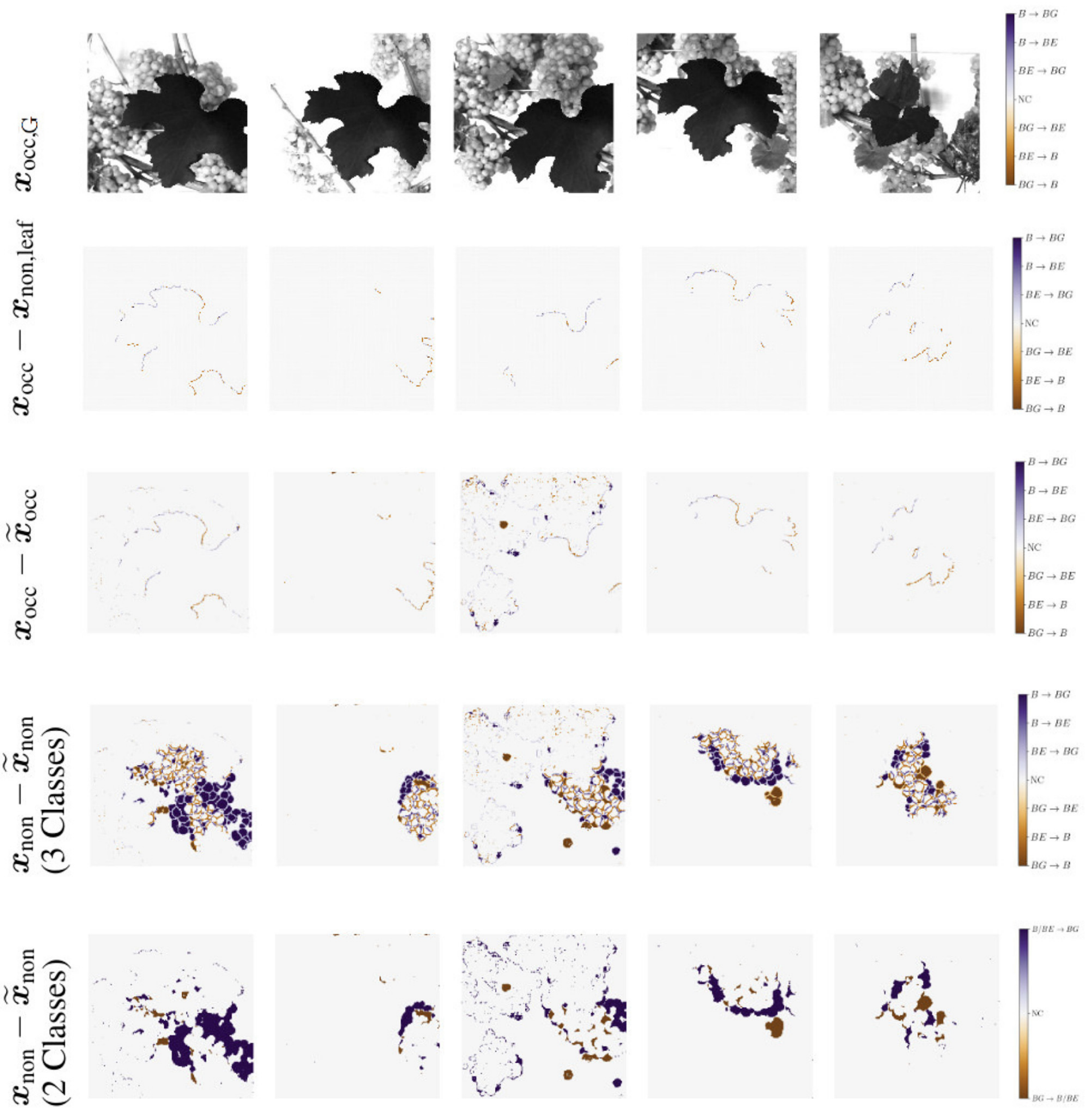


**FIGURE 11 |** Generation maps between target berry masks and generated output berry masks, as described in Section 3.4.2. The first row illustrates the respective grayscale input image. The second row shows the input $\boldsymbol{x}_{\text{occ}}$ compared to the target output occluded by the leaf used for creating the input $\boldsymbol{x}_{\text{non,leaf}}$. The third row shows the maps in the occluded domain. The input $\boldsymbol{x}_{\text{occ}}$ is compared to the generated output occluded by the leaf used for creating the input $\widetilde{\boldsymbol{x}}_{\text{occ}}$. Second and third row occur in the occluded domain. The fourth and fifth row show the maps of the non-occluded domain. The target output $\boldsymbol{x}_{\text{non}}$ is compared to the generated output $\widetilde{\boldsymbol{x}}_{\text{non}}$. In the fourth row all three classes are included. The last row illustrate the same, but including only two classes. Classes `berry-edge` *BE* and `berry` *B* are combined in one class. `Background` is indicated by *BG* and if there is no class change it is indicated by *NC* for no change.

diameter within the berry masks shown in **Figures 12C,D**. The distributions of the metrics are highly similar between generated result end reference. For both metrics, there is a slight tendency toward an increase in area and diameter for the generated berries.

## 4.5. Experiment 4—Counting in the Non-occluded Domain

Since the number of berries is of high importance for yield estimation, we investigate the estimation of this number in this experiment. We compare the counts based on the input

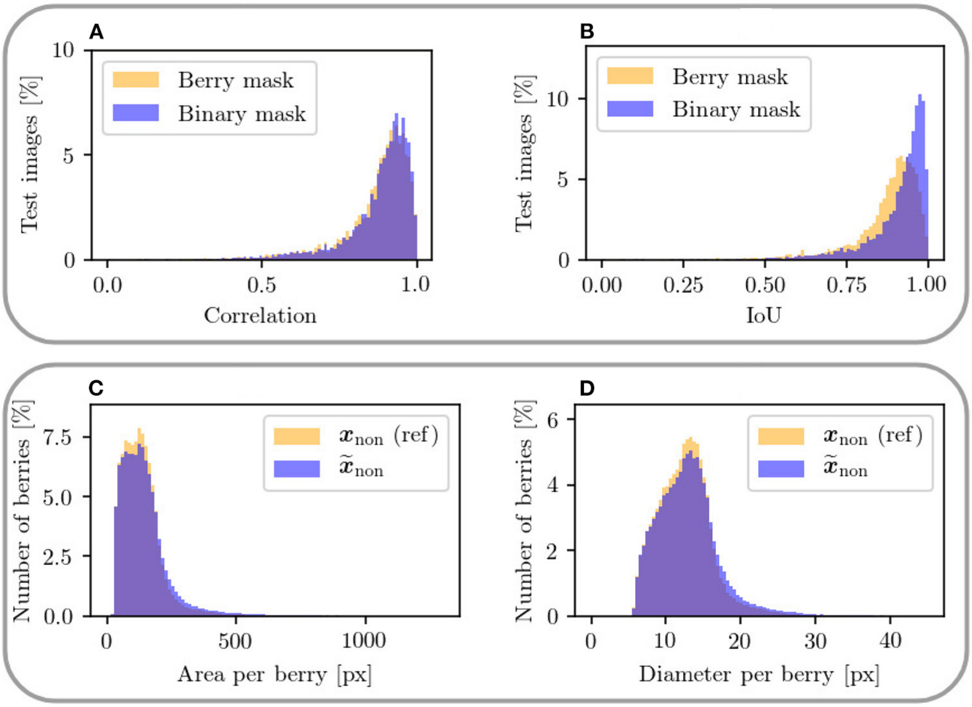


**FIGURE 12 |** The upper plots show a mask-based comparison within the non-occluded domain between berry mask and binary mask including only two classes for the metrics **(A)** correlation between $x_{non}$ and $\tilde{x}_{non}$ and **(B)** IoU of the `berry` pixel in $x_{non}$ and $\tilde{x}_{non}$. The lower plots **(C,D)** show a comparison of area and diameter per berry between target output $x_{non}$ and generated output $\tilde{x}_{non}$ in the non-occluded domain. Only areas up to 1,300 px and diameters up to 45 px are plotted.

patches in the occluded domain and the target patches in the non-occluded domain with the generated results of our approach.

### 4.5.1. Used Data, Model, and Evaluation Metrics

For this experiment, we use the synthetic datasets *Dataset 1* and *Dataset 3* based on VSP and SMPH defoliation. Our model is trained on both training sets and evaluated on the corresponding test sets. During testing, we consider only the mask of the data patches. For the evaluation, we use the $R^2$-Plot to plot the absolute count of the input (**Figures 10A,D**) and the absolute count of the generated output of our method (**Figures 10C,E**) with the reference count from the target mask, respectively. Furthermore, we examine the distribution of the relative deviations from the reference (see **Figure 13**).

### 4.5.2. Results

Counting in the occluded domain, presented in **Figures 10A,D**, shows that there is an underestimation of the number of berries compared to the reference. Our model shows a shift of the number of berries toward the reference for both types of defoliation. In both cases, the $R^2$ value increases compared to the $R^2$ value of the occluded domain, which corresponds to a better approximation of the data compared to the reference. It is important to mention that not only the sample distribution shifts, but also compresses and concentrates along the reference line.

**Figure 13** supports this observation. The plots show the relative difference of the counted berries in the occluded domain and our method in the non-occluded domain compared to the reference counting. Our method (blue) depicts a normal distribution with a mean near zero. If the values of the occluded distribution (orange) were increased by a factor, this would lead to a shift in the distribution, but it would still be more stretched than ours. The peaks at value 0 correspond mostly to synthetic images where the synthetic leaf does not cover any berries. This is the case, for example, with images that show few berries.

Both models exhibit problems in the generation of patches that depict more than 150 berries. This is the case for VSP (**Figure 10C**) and SMPH (**Figure 10E**). For both types of defoliation, a trend is nevertheless evident above the critical value of 150 berries. Even though an underestimation of berries tends to be counted above this value, the count fits the reference better than the count in the occluded domain.

In the occluded domain, there are data points that differ strongly from the reference. Our method reduces the amount of such points and also reduces the deviation of the highly deviating points.

## 4.6. Experiment 5—Application to Natural Data

One of the contributions of our work is to investigate the applicability of our approach to natural data. In detail, we evaluate whether our model generalizes to natural images when it is trained on synthetic data.

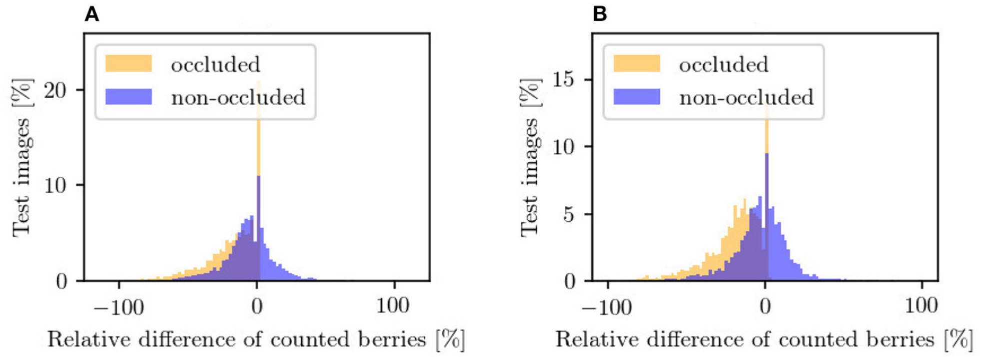


**FIGURE 13 |** Counting in the occluded domain (orange) and after applying our approach in the non-occluded domain (blue) relative to the reference counting in the non-occluded domain. The plots illustrate the results for **(A)** VSP defoliation and **(B)** SMPH defoliation. A negative value means that fewer berries are counted than in the reference and *vice versa*. Each bar corresponds to a width of 2%.

### 4.6.1. Used Data, Model, and Evaluation Metrics

We use the synthetic datasets *Dataset 1* and *Dataset 3* to train our model. For the test phase, we use the natural datasets *Dataset 4* and *Dataset 5*. One dataset each for VSP defoliation and one for SMPH defoliation.

The differences of the natural dataset to the synthetic dataset are the stronger coverage by a denser leaf canopy, the resulting deviating exposure ratios, and the lower contrast whereby the contours of the leaves are not easily distinguishable from berries. Other differences are found in the transformation applied to the natural dataset, since non-occluded areas are not identical in both domains, as already pointed out in the introduction. Depending on the patch position in the non-occluded domain in the original image, the transformation goes beyond the boundaries of the original image in the occluded domain. To achieve a patch size of 656 × 656 px which is equivalent to the cropped patch size of the dataset, the appropriate borders of the patch are filled with black spixels.

We perform our evaluation visually, which means we compare the input from the occluded domain with the generated output of our approach in the non-occluded domain. Due to the transformation issues, direct numerical comparison and evaluation between target and generated output are not useful for the majority of patches. However, we would like to give an impression of the results by means of the visual representation.

### 4.6.2. Results

In **Figure 14**, we provide example results of our approach applied to natural data. For each example, the first column shows the input $\boldsymbol{x}_{\text{occ}}$ of the occluded domain, the second column the reference $\boldsymbol{x}_{\text{non}}$ in the non-occluded domain, and the last column our generated output $\tilde{\boldsymbol{x}}_{\text{non}}$ in the non-occluded domain. The first row visualizes the G channel of a patch and the second row the corresponding mask. The results show that the canopy is reduced and important areas in the patch are reproduced. Generally, the observations from the previously described experiments can be repeated. Using our generative approach, `berry` and `berry-edge` pixel regions in the input mask are also transferred to the generated output for the natural data. For input patches of the occluded domain being similar to the synthetic data (**Figures 14A–C**), the results show an expansion of the existing berry region. Our approach is also able to deal with transformation problems, as in **Figure 14A** where the transformation goes beyond the original image boundaries. There are examples, like seen in **Figure 14C**, that look similar to the target, or examples that look real compared to the input but do not reflect the target output (**Figure 14B**).

For the majority of natural data, exact transformations are not available, so this is challenging to evaluate. In examples like the one in **Figure 14D** transformation, rotation and scale fit, but due to defoliation, the orientation of the grape bunch is different in input and output target. In the input, the grape bunch is more horizontal. In the target, it is vertical. The example in **Figure 14E** shows that grape bunches are also completely different in translation due to the different weights attached to the branches. In this example, the grape bunch that is visible in the input is only partially visible at the top of the patch in the target output. The generated output adapts to the input and is also expanded, but is not comparable to the target.

Furthermore, we observe checkerboard artifacts that appear in the generated G patches (see **Figures 14A,C**). The artifacts occur more in patches that present a dense canopy.

# 5. DISCUSSION

## 5.1. Experiment 1

Our results confirm that our model trained on GB data learns where background is present. This is an important factor for realistic generated images. We found that the model trained only with berry mask B has more problems with images containing many berries than the model trained on GB data, both visually and in the counting results. The deficits in counting are explained by the fact that there are relatively few patches in the dataset with a number greater than 150 compared to the number of patches containing <150 berries. This is also true for the underestimation of the count with the GB dataset. However,

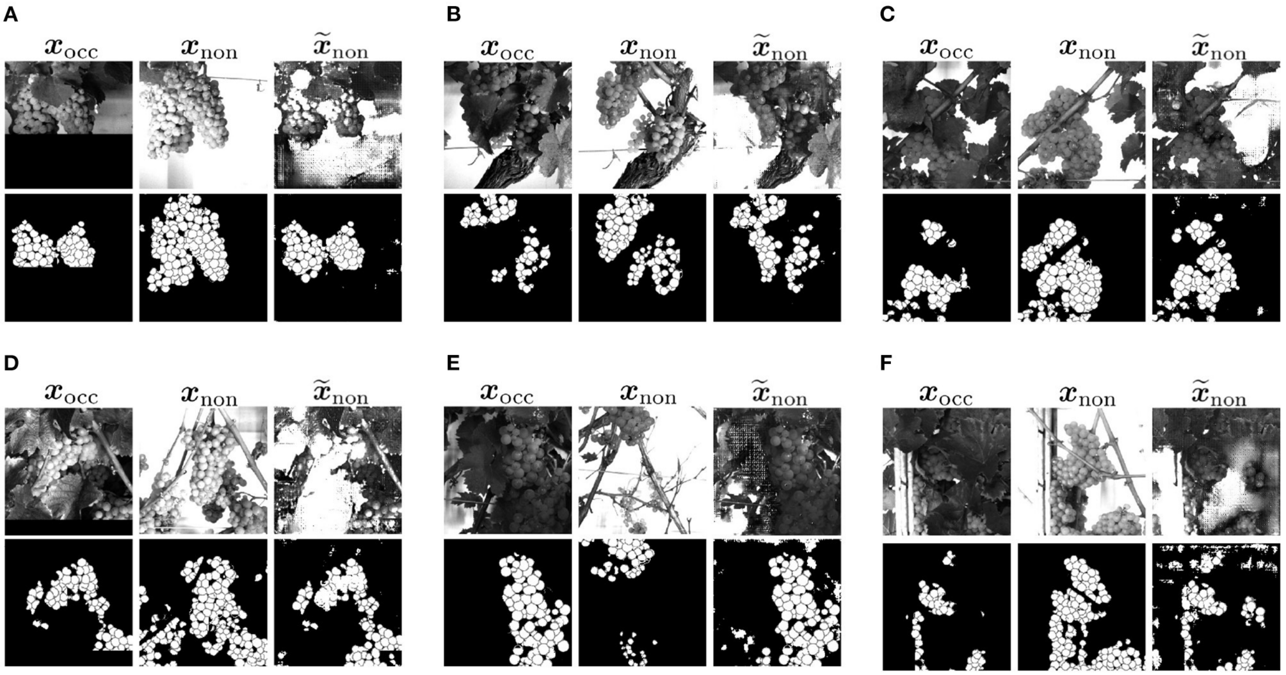

A
$\boldsymbol{x}_{\text{occ}}$
$\boldsymbol{x}_{\text{non}}$
$\widetilde{\boldsymbol{x}}_{\text{non}}$
B
$\boldsymbol{x}_{\text{occ}}$
$\boldsymbol{x}_{\text{non}}$
$\widetilde{\boldsymbol{x}}_{\text{non}}$
C
$\boldsymbol{x}_{\text{occ}}$
$\boldsymbol{x}_{\text{non}}$
$\widetilde{\boldsymbol{x}}_{\text{non}}$
D
$\boldsymbol{x}_{\text{occ}}$
$\boldsymbol{x}_{\text{non}}$
$\widetilde{\boldsymbol{x}}_{\text{non}}$
E
$\boldsymbol{x}_{\text{occ}}$
$\boldsymbol{x}_{\text{non}}$
$\widetilde{\boldsymbol{x}}_{\text{non}}$
F
$\boldsymbol{x}_{\text{occ}}$
$\boldsymbol{x}_{\text{non}}$
$\widetilde{\boldsymbol{x}}_{\text{non}}$

**FIGURE 14** | Visual representation of generated result based on natural data input. **(A–C)** Examples with a good transformation between the patches. They are present as a minority in the natural dataset. **(D,E)** Show examples with a insufficient transformation between the patches. **(F)** Shows an example with artifacts in the generated mask.

by using the additional G channel, the result images can be generated more precisely. More detailed analyses of the berry counting can be found in Experiment 4. Taking into account the correlations and with the goal to generate highly probable results with a distribution that matches the input, rather than the exact image content of each image, Dataset 1 leads to better results on average as claimed in the beginning of the results section.

## 5.2. Experiment 2

We found that a high percentage of the results is correctly transferred from occluded to non-occluded domain. The occurring deviations between $\{\boldsymbol{x}_{\text{occ}}, \widetilde{\boldsymbol{x}}_{\text{occ}}\}$ can be traced back to the test results, which not only show the class values 0, 127, and 255 within the mask, but also pixels with values in between. This means that the model does not clearly assign the respective pixel to a class. At this point, we apply data post-processing to our generated data, as described in Section 3.4.1. Pixel in areas of class boundaries are particularly affected here, which is why the differences arise in these areas.

The deviations at the edges of the leaf are due to an additional edge with a width of about three pixels, which was added during the creation of the synthetic occluded input mask. The masks $\widetilde{\boldsymbol{x}}_{\text{non}}$ and $\boldsymbol{x}_{\text{non}}$, on which the masks $\widetilde{\boldsymbol{x}}_{\text{occ}}$ and $\boldsymbol{x}_{\text{occ,leaf}}$ used in this experiment are based on, show a continuation of the depicted grape branches exactly at these transitions. This results in variations between the paired masks at this location.

The key findings from this experiment are that despite individual deviations, the visible part of the mask of the occluded domain is safely transferred to the non-occluded domain and stays unchanged. We assume that the model will make no result-altering changes.

## 5.3. Experiment 3

Although our approach does not aim to generate the exact position and shape of berries, the results indicate that the similarity of the generated results and the references are high. The observed high IoU indicates a similar position of the grape bunches independent of the berry objects in the generated result compared to the reference. Berries are predicted in the right areas, but their shape and position do not correspond exactly to the reference. An increasing area and diameter suggest, that if the area of the total `berry` pixel per patch remains the same, there is a possibility that too few berries are predicted.

## 5.4. Experiment 4

In the berry counting, the underestimation of the amount of berries per patch is clearly evident in the concealed area, which can be explained by the occlusion covering part of the berries. The results indicate that we obtain better results with our approach than when we apply only a factor to the counting. We explain the deteriorating results above a berry number of 150 by the fact that the proportion of training images with a count above the critical value is relatively small in contrast to the number of images with an amount below the critical value. Our method reduces the number of outliers and additionally reduces the variance of the highly deviant points. We achieve a shift of the distribution as well as a compression and concentration along the reference line, so that our results are more accurate than those in the occluded domain.

## 5.5. Experiment 5

Generally, our findings from the previously described experiments can be confirmed within this experiment. Although it is apparent that the model trained only on the synthetic data mentioned above is not yet strong enough to obtain similarly good results for the more complex natural data as for the synthetic data, we consider the results promising. We assume that mixing natural and synthetic data or using more complex synthetic training data can improve the results. The checkerboard artifacts that we observed could be reduced by improving the generator (Odena et al., 2016). This could also result in reduced artifacts, like they occur in the mask in **Figure 14F**. The artifacts occur more in patches that present a dense canopy.

## 5.6. Future Directions

To make the model more robust and generalizable to variations between natural and synthetic data, the synthetic data can be designed with more complex changes, for example, by increasing the synthetic occlusion through the use of more leaves per patch. In addition, brightness and contrast could be varied, for example, to reduce the dominant white background of the synthetic data and thus make it more difficult for the model to detect the occlusion. Interesting future work is the application of the model to other varieties and to see how it behaves. We assume that the model applied on varieties with a comparable or smaller grape bunch size and a similar data appearance will behave similarly to our presented results. With a larger grape bunch size and thus a larger number of berries, the model might have to be re-trained in order to achieve an accurate result for a large number of berries. Another promising future direction is to train the model from a combination of synthetic images and a limited amount of natural images. In this case, the transformation between the two required domains needs to be accurate enough and suitable data must be selected. Another possibility would involve extensive manual work on the transformation between the domains or more sophisticated techniques such as image warping. In the future, the checkerboard artifacts that occur in data could be reduced by replacing the transpose convolution layer of the decoder in the U-Net generator with bi-linear up-sampling operations, as described by Odena et al. (2016).

# 6. CONCLUSION

In this work, we have demonstrated the suitability of a conditional generative adversarial network like Pix2Pix to generate a scenario behind occlusions in grapevine images that is highly probable based on visible information in the images. Our experiments have shown that our approach has learned patterns that characterize typical berries and clusters without occlusions so that areas where berries are added and other areas where the image remains unchanged can be identified without having to provide prior knowledge about occlusions. Compared to counting with occluded areas, we show that our approach provides a count that is closer to the manual reference count. In contrast to applying a factor, our approach directly involves the appearance of the visible berries and thus better adapts to local conditions.

We have trained our conditional adversarial network-based model on synthetic data only in order to overcome the challenge of lacking aligned image pairs. We show that the model is also applicable to natural data, given that the canopy is not too dense and the variation between natural data and synthetic data is not too high.

# 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/s.

# AUTHOR CONTRIBUTIONS

JK initiated, designed, and conducted the analyses. RR helped to initiate the work and co-designed the experiments. JK, AK, and LZ contributed to the data preparation. All authors contributed to the writing of this manuscript. All authors contributed to the article and approved the submitted version.

# FUNDING

This project was funded by the European Agriculture Fund for Rural Development with contribution from North-Rhine Westphalia (17-02.12.01--10/16---EP-0004617925-19 -001). Furthermore, this work was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC 2070—390732324, and partially by the German Federal Ministry of Education and Research (BMBF, Bonn, Germany) in the framework of the project novisys (FKZ 031A349).

# ACKNOWLEDGMENTS

The content of the manuscript have previously appeared in a preprint (Kierdorf et al., 2021).

# REFERENCES

Aquino, A., Diago, M. P., Millán, B., and Tardáguila, J. (2017). A new methodology for estimating the grapevine-berry number per cluster using image analysis. *Biosyst. Eng.* 156, 80–95. doi: 10.1016/j.biosystemseng.2016.12.011

Aquino, A., Millan, B., Diago, M.-P., and Tardaguila, J. (2018). Automated early yield prediction in vineyards from on-the-go image acquisition.

*Comput. Electron. Agric.* 144, 26–36. doi: 10.1016/j.compag.2017.11.026

Arteta, C., Lempitsky, V., and Zisserman, A. (2016). "Counting in the wild," in *European Conference on Computer Vision* (Springer), 483–498. doi: 10.1007/978-3-319-46478-7_30

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. (2009). Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* 28, 24. doi: 10.1145/1531326.1531330

Batista, G. E., and Monard, M. C. (2002). A study of K-nearest neighbour as an imputation method. *His* 87, 48. doi: 10.1109/METRIC.2004.1357895

Bertalmio, M., Vese, L., Sapiro, G., and Osher, S. (2003). Simultaneous structure and texture image inpainting. *IEEE Trans. Image Process.* 12, 882–889. doi: 10.1109/TIP.2003.815261

Chen, L., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR, abs/1802.02611.* doi: 10.1007/978-3-030-01234-2_49

Clingeleffer, P. R., Martin, S., Dunn, G., and Krstic, M. (2001). *Crop Development, Crop Estimation and Crop Control to Secure Quality and Production of Major Wine Grape Varieties: A National Approach.* Final Report. Grape and Wine Research & Development Corporation.

Coviello, L., Cristoforetti, M., Jurman, G., and Furlanello, C. (2020). GBCNet: in-field grape berries counting for yield estimation by dilated CNNs. *Appl. Sci.* 10, 4870. doi: 10.3390/app10144870

Dekel, T., Gan, C., Krishnan, D., Liu, C., and Freeman, W. T. (2018). "Sparse, smart contours to represent and edit images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 3511–3520. doi: 10.1109/CVPR.2018.00370

Diago, M., Martinez De Toda, F., Poni, S., and Tardaguila, J. (2009). "Early leaf removal for optimizing yield components, grape and wine composition in tempradillo (*Vitis vinifera* L.)," in *Proceedings of the 16th International GiESCO Symposium*, ed J. A. Wolpert, Davis, CA, 113–118.

Diago, M.-P., Correa, C., Millán, B., Barreiro, P., Valero, C., and Tardaguila, J. (2012). Grapevine yield and leaf area estimation using supervised classification methodology on rgb images taken under field conditions. *Sensors* 12, 16988–17006. doi: 10.3390/s121216988

Ehsani, K., Mottaghi, R., and Farhadi, A. (2018). "Segan: segmenting and generating the invisible," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 6144–6153. doi: 10.1109/CVPR.2018.00643

Enders, C. K. (2001). A primer on maximum likelihood algorithms available for use with missing data. *Struct. Equat. Model.* 8, 128–141. doi: 10.1207/S15328007SEM0801_7

Feng, H., Yuan, F., Skinkis, P. A., and Qian, M. C. (2015). Influence of cluster zone leaf removal on pinot noir grape chemical and volatile composition. *Food Chem.* 173, 414–423. doi: 10.1016/j.foodchem.2014.09.149

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial networks. *arXiv preprint arXiv:1406.2661.*

Hacking, C., Poona, N., Manzan, N., and Poblete-Echeverría, C. (2019). Investigating 2-D and 3-D proximal remote sensing techniques for vineyard yield estimation. *Sensors* 19, 3652. doi: 10.3390/s19173652

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, 2961–2969. doi: 10.1109/ICCV.2017.322

Helmert, F. (1880). *Die Mathematischen Physicalischen Theorieen der höheren Geodäsie*. B. G. Teubner. Available online at: https://books.google.de/books?id=g0vkwQEACAAJ

Iizuka, S., Simo-Serra, E., and Ishikawa, H. (2017). Globally and locally consistent image completion. *ACM Trans. Graph.* 36, 1–14. doi: 10.1145/3072959.3073659

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, 1125–1134. doi: 10.1109/CVPR.2017.632

Kicherer, A., Herzog, K., Bendel, N., Klück, H.-C., Backhaus, A., Wieland, M., et al. (2017). Phenoliner: a new field phenotyping platform for grapevine research. *Sensors* 17, 1625. doi: 10.3390/s17071625

Kicherer, A., Roscher, R., Herzog, K., Förstner, W., and Töpfer, R. (2014). "Image based evaluation for the detection of cluster parameters in grapevine," in *XI International Conference on Grapevine Breeding and Genetics 1082*, Yanqing, 335–340. doi: 10.17660/ActaHortic.2015.1082.46

Kierdorf, J., Weber, I., Kicherer, A., Zabawa, L., Drees, L., and Roscher, R. (2021). Behind the leaves-estimation of occluded grapevine berries with conditional generative adversarial networks. *arXiv preprint arXiv:2105.10325.* doi: 10.48550/arXiv.2105.10325

Kim, J. K., and Rao, J. (2009). A unified approach to linearization variance estimation from survey data after imputation for item nonresponse. *Biometrika* 96, 917–932. doi: 10.1093/biomet/asp041

Lee, D., Kim, J., Moon, W.-J., and Ye, J. C. (2019). "Collagan: collaborative GAN for missing image data imputation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, 2487–2496. doi: 10.1109/CVPR.2019.00259

Lempitsky, V., and Zisserman, A. (2010). Learning to count objects in images. *Adv. Neural Inform. Process. Syst.* 23, 1324–1332.

Liu, G., Reda, F. A., Shih, K. J., Wang, T.-C., Tao, A., and Catanzaro, B. (2018). "Image inpainting for irregular holes using partial convolutions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, 85–100. doi: 10.1007/978-3-030-01252-6_6

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 60, 91–110. doi: 10.1023/B:VISI.0000029664.99615.94

Mack, J., Lenz, C., Teutrine, J., and Steinhage, V. (2017). High-precision 3d detection and reconstruction of grapes from laser range data for efficient phenotyping based on supervised learning. *Comput. Electron. Agric.* 135, 300–311. doi: 10.1016/j.compag.2017.02.017

Mack, J., Schindler, F., Rist, F., Herzog, K., Töpfer, R., and Steinhage, V. (2018). Semantic labeling and reconstruction of grape bunches from 3D range data using a new RGB-D feature descriptor. *Comput. Electron. Agric.* 155, 96–102. doi: 10.1016/j.compag.2018.10.011

May, P. (1972). Forecasting the grape crop. *Australien Wine, Brewing and Spirit Review.* 90, 46–48.

Mirza, M., and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784.* doi: 10.48550/arXiv.1411.1784

Nuske, S., Achar, S., Bates, T., Narasimhan, S., and Singh, S. (2011). "Yield estimation in vineyards by visual grape detection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, 2352–2358. doi: 10.1109/IROS.2011.6095069

Nuske, S., Wilshusen, K., Achar, S., Yoder, L., Narasimhan, S., and Singh, S. (2014). Automated visual yield estimation in vineyards. *J. Field Robot.* 31, 837–860. doi: 10.1002/rob.21541

Nyarko, E. K., Vidović, I., Radočaj, K., and Cupec, R. (2018). A nearest neighbor approach for fruit recognition in RGB-D images based on detection of convex surfaces. *Expert Syst. Appl.* 114, 454–466. doi: 10.1016/j.eswa.2018.07.048

Odena, A., Dumoulin, V., and Olah, C. (2016). Deconvolution and checkerboard artifacts. *Distill* 1, e3. doi: 10.23915/distill.00003

Ostyakov, P., Suvorov, R., Logacheva, E., Khomenko, O., and Nikolenko, S. I. (2018). Seigan: Towards compositional image generation by simultaneously learning to segment, enhance, and inpaint. *arXiv preprint arXiv:1811.07630.* doi: 10.48550/arXiv.1811.07630

Paul Cohen, J., Boucher, G., Glastonbury, C. A., Lo, H. Z., and Bengio, Y. (2017). "Count-ception: counting by fully convolutional redundant counting," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, Venice, 18–26. doi: 10.1109/ICCVW.2017.9

Robins, J. M., and Wang, N. (2000). Inference for imputation estimators. *Biometrika* 87, 113–124. doi: 10.1093/biomet/87.1.113

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-net: convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Munich: Springer), 234–241. doi: 10.1007/978-3-319-24574-4_28

Roscher, R., Herzog, K., Kunkel, A., Kicherer, A., Töpfer, R., and Förstner, W. (2014). Automated image analysis framework for high-throughput determination of grapevine berry sizes using conditional random fields. *Comput. Electron. Agric.* 100, 148–158. doi: 10.1016/j.compag.2013.11.008

Rubin, D. B. (1996). Multiple imputation after 18+ years. *J. Am. Stat. Assoc.* 91, 473–489. doi: 10.1080/01621459.1996.10476908

Rubin, D. B. (2004). *Multiple Imputation for Nonresponse in Surveys, Vol. 81*. John Wiley & Sons.

Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A., and Chen, L. (2018). Inverted residuals and linear bottlenecks: mobile networks for classification, detection and segmentation. *CoRR, abs/1801.04381*. doi: 10.1109/CVPR.2018.00474

Schöler, F., and Steinhage, V. (2015). Automated 3D reconstruction of grape cluster architecture from sensor data for efficient phenotyping. *Comput. Electron. Agric*. 114, 163–177. doi: 10.1016/j.compag.2015.04.001

Van Buuren, S., and Oudshoorn, K. (1999). *Flexible Multivariate Imputation by MICE*. Leiden: TNO.

von Hippel, P. T., and Bartlett, J. (2012). Maximum likelihood multiple imputation: Faster imputations and consistent standard errors without posterior draws. *Statistical Sci*. 36, 400–420. doi: 10.1214/20-STS793

Xie, W., Noble, J. A., and Zisserman, A. (2018). Microscopy cell counting and detection with fully convolutional regression networks. *Comput. Methods Biomech. Biomed. Eng*. 6, 283–292. doi: 10.1080/21681163.2016.1149104

Xiong, W., Yu, J., Lin, Z., Yang, J., Lu, X., Barnes, C., et al. (2019). "Foreground-aware image inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, 5840–5848. doi: 10.1109/CVPR.2019.00599

Yan, X., Wang, F., Liu, W., Yu, Y., He, S., and Pan, J. (2019). "Visualizing the invisible: occluded vehicle segmentation and recovery," in *Proceedings of the IEEE International Conference on Computer Vision*, Salt Lake City, UT, 7618–7627. doi: 10.1109/ICCV.2019.00771

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. (2018). "Generative image inpainting with contextual attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 5505–5514. doi: 10.1109/CVPR.2018.00577

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. (2019). "Free-form image inpainting with gated convolution," in *Proceedings of the IEEE International Conference on Computer Vision*, Long Beach, CA, 4471–4480. doi: 10.1109/ICCV.2019.00457

Zabawa, L., Kicherer, A., Klingbeil, L., Milioto, A., Topfer, R., Kuhlmann, H., et al. (2019). "Detection of single grapevine berries in images using fully convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Long Beach, CA. doi: 10.1109/CVPRW.2019.00313

Zabawa, L., Kicherer, A., Klingbeil, L., Töpfer, R., Kuhlmann, H., and Roscher, R. (2020). Counting of grapevine berries in images via semantic segmentation using convolutional neural networks. *ISPRS J. Photogr. Remote Sens*. 164, 73–83. doi: 10.1016/j.isprsjprs.2020.04.002

**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.

ORIGINAL RESEARCH
published: 13 April 2022
doi: 10.3389/fpls.2022.855660

# Plant Species Classification Based on Hyperspectral Imaging *via* a Lightweight Convolutional Neural Network Model

***Keng-Hao Liu[1], Meng-Hsien Yang[1], Sheng-Ting Huang[1] and Chinsu Lin[2]****

*[1] Department of Mechanical and Electro-Mechanical Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan,*
*[2] Department of Forestry and Natural Resources, National Chiayi University, Chiayi, Taiwan*

In recent years, many image-based approaches have been proposed to classify plant species. Most methods utilized red green blue (RGB) imaging materials and designed custom features to classify the plant images using machine learning algorithms. Those works primarily focused on analyzing single-leaf images instead of live-crown images. Without considering the additional features of the leaves' color and spatial pattern, they failed to handle cases that contained leaves similar in appearance due to the limited spectral information of RGB imaging. To tackle this dilemma, this study proposes a novel framework that combines hyperspectral imaging (HSI) and deep learning techniques for plant image classification. We built a plant image dataset containing 1,500 images of 30 different plant species taken by a 470–900 nm hyperspectral camera and designed a lightweight conventional neural network (CNN) model (LtCNN) to perform image classification. Several state-of-art CNN classifiers are chosen for comparison. The impact of using different band combinations as the network input is also investigated. Results show that using simulated RGB images achieves a kappa coefficient of nearly 0.90 while using the combination of 3-band RGB and 3-band near-infrared images can improve to 0.95. It is also found that the proposed LtCNN can obtain a satisfactory performance of plant classification (kappa = 0.95) using critical spectral features of the green edge (591 nm), red-edge (682 nm), and near-infrared (762 nm) bands. This study also demonstrates the excellent adaptability of the LtCNN model in recognizing leaf features of plant live-crown images while using a relatively smaller number of training samples than complex CNN models such as AlexNet, GoogLeNet, and VGGNet.

**Keywords: plant species classification, live-crown features, leaf feature recognition, plant stress detection, dimensionality reduction, convolutional neural network, hyperspectral imaging, deep learning**

***Edited by:***
*Ian Stavness,*
*University of Saskatchewan, Canada*

***Reviewed by:***
*Dimitrios Fanourakis,*
*Technological Educational Institute of Crete, Greece*
*Mohsen Yoosefzadeh Najafabadi,*
*University of Guelph, Canada*

****Correspondence:***
*Chinsu Lin*
*chinsu@mail.ncyu.edu.tw*

***Specialty section:***
*This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science*

***Received:*** *15 January 2022*
***Accepted:*** *01 March 2022*
***Published:*** *13 April 2022*

***Citation:***
*Liu K-H, Yang M-H, Huang S-T and Lin C (2022) Plant Species Classification Based on Hyperspectral Imaging via a Lightweight Convolutional Neural Network Model. Front. Plant Sci. 13:855660. doi: 10.3389/fpls.2022.855660*

## INTRODUCTION

Species composition provides basic individual biological features of a landscape and a forest ecosystem. The ability to identify species of individual plants or trees over an inventory plot as well as a forest stand is essential for the automatic mapping of plant distribution, biological diversity, stand structure, and even for diagnosing the dynamics of a forest stand (Lin et al., 2016; Lin, 2019; Santos et al., 2019). The development of plant mapping techniques has the benefit of identifying signals of climate change based on plant phenology (Lin C. et al., 2018) and advanced tree segmentation (Lin C.Y. et al., 2018; Jaskierniak et al., 2021) *via* remote sensing images. Remote sensing images have recently been used to map species distribution mainly according to spectral information with classification techniques. The gaps in/between tree crowns, which tend to be caused by lower crown density, greenness, and background materials, create a challenge for species classification using high-resolution satellite images (Lin et al., 2015a). Image fusion that integrates very high spatial resolution images with atmospherically corrected high spectral resolution can benefit tree crown delineation and improve the mapping (Lin et al., 2015b; Lin C.Y. et al., 2018). However, pixels of the inter- and intra-canopy gaps in a fused image became more significant and increased impact on species crown reflectance (Lin et al., 2015a). Consequently, plant species recognition with remote sensing images becomes a more complicated task involving not merely pixel-based but also object-based approaches. Recently, advanced sensor technology can acquire very high spatial resolution (VHSR) images from various platforms such as in-situ, drone, airborne, and spaceborne for environmental studies. With regard to plant studies, VHSR images are capable of sensing every subtle difference of reflectance in a scale from sub-centimetric to decimetric size allowing better opportunity to reveal detailed features of materials. This is particularly evident in in-situ hyperspectral imaging systems. Moreover, to address the impact of climate change on a vulnerable vegetation community or ecosystem, the dynamics of the community must be derived from the perspective of plant species composition. Therefore, more effort is needed to investigate the problem of developing suitable remote sensing algorithms for classifying a large number of plant species.

During the last decade, most research on plant classification with red green blue (RGB) images was primarily based on extracting leaf features and performing classification with machine learning (ML; Zhang, 2020) classifiers on single-leaf images. The ML classifiers used include support vector machine, K-nearest neighbor, probability neural network, and so on. The methods of leaf feature extraction include polar Fourier transform (Kadir et al., 2013), Canny edge detection (Salman et al., 2017), Fourier transforms (Hossain and Amin, 2010; Khmag et al., 2017), and wavelet decomposition method (Zhang H. et al., 2012), in which the most frequently used features were the color, shape, contour, and texture of leaves. Additional features such as leaf width factor and leaf edge were also used to develop multiscale-distance feature matrixes to improve classification by Beghin et al. (2010) and Hu et al. (2012). As noted, the issues raised in these image-based plant recognition methods are highly dependent on feature engineering and the lack of leaf composition information. In other words, much more effort should be made to achieve noise removal, leaf feature measurements, and texture divergence calculations. The classification seems very dependent on leaf preprocessing.

With the breakthrough of hardware technology, deep learning (DL; Bengio et al., 2017) became the mainstream data processing method in recent years. Among many DL approaches, the convolutional neural network (CNN) is the most popular and representative one in computer vision and imaging processing communities (Ioffe and Szegedy, 2015; Simonyan and Zisserman, 2015; Szegedy et al., 2015; He et al., 2016; Krizhevsky et al., 2017; Wang et al., 2021; Yang et al., 2021). Different from ML methods, CNN can integrate feature derivation, feature learning, and classifier into a single architecture. Many studies have reported that using CNN approaches can produce significantly higher accuracy than using conventional ML ones, as long as with sufficient training data. The reason is that CNN can automatically learn objective, multi-scale, and most discriminative features from raw data without human subjectivity. Following this trend, a few CNN-based plant recognition methods were proposed (Lee et al., 2015; Grinblat et al., 2016; Carranza-Rojas et al., 2017; Lee et al., 2017; Chen et al., 2018; Zhu et al., 2019; Chen et al., 2021). A two-dimensional (2D)-CNN model is adopted in each work to learn the discriminative features from the entire RGB plant images. The spatial relationship of leaf arrangement (phyllotaxy) and overlapping patterns can also be discovered. In other words, various spatial features of interest objects revealed in a VHSR image can be processed by suppressing background materials' signals and therefore recognized based on the spatial pattern in spectra. This allows us to identify plant species in a way very similar to phytologists with plant morphological features such as leaf color and size, contour, surface, venation, and even phyllotaxy.

Although the current DL approaches demonstrate a certain level of reliability, they still may fail to handle the cases that contain plant species that are similar in appearance, even with enough training data, due to the limited spectral information provided by RGB imaging. If two or more plants have similar outer appearance characteristics, the CNN-based methods may misclassify them. Under such circumstances, it is necessary to use the imaging system providing more delicate spectral information to improve the recognition performance. With the advancement of remote sensing imaging technology, hyperspectral imaging (HSI; Chang, 2013) was developed and widely applied to many topics such as agriculture (Nicolaï et al., 2006; Baiano et al., 2012; Teena et al., 2014; Jung et al., 2015; Marshall et al., 2015; Rapaport et al., 2015; Adão et al., 2017; Gao et al., 2018; Mirzaei et al., 2019; Sun et al., 2019; Sinha et al., 2020; Feng et al., 2021), military defense (Briottet et al., 2006), environment (Zhang B. et al., 2012; Schmitter et al., 2017; Harrison et al., 2018; Abbas et al., 2021), plant phenotyping (Ubbens and Stavness, 2017; Nasiri et al., 2021), and medical imaging (Liu et al., 2007; Fei, 2020). The familiar HSI image contains hundreds of spectral bands ranging from the visible spectrum to the near-infrared (NIR) spectrum so that it can capture the complete spectral characteristics of target objects. Due to its superior spectral resolution, many substances indistinguishable to the naked eye can be recognized. In recent years, hyperspectral cameras have been gradually commercialized. The use of micro hyperspectral cameras for research has become more and more popular. Therefore, using HSI technology to classify plant species has great potential.

Since the use of both HSI and DL techniques for plant species recognition has not been fully explored, in this paper, we conducted a study that adopts hyperspectral plant images as the sample materials and designed a lightweight CNN model to achieve accurate image classification. Firstly, we collect the hyperspectral images from 30 plants of different species with a hyperspectral camera and build a dedicated HSI plant dataset containing 1,500 images. Since the existing CNN-based classification models were designed for the datasets composed of tens of thousands of RGB images of specific objects with a large number of categories, they may not be suitable for the training of our HSI plant dataset. Therefore, this study proposed an improved lightweight convolutional neural network based on the architecture of GoogLeNet (Szegedy et al., 2015) to disclose the issues of species classification through hyperspectral images by deep learning technique. Hyperspectral images have hundreds of bands that are highly correlated, and spectral information of the bands is excessively redundant for vegetation application such as water content modeling (Lin et al., 2012), hyperspectral signal restoration (Lin, 2017; Lin, 2018), and chlorophyll concentration estimation (Lin et al., 2015c; Lin and Lin, 2019). Appropriate feature selection strategies in deriving critical bands for accurate species classification were also explored. Plant classification will be beneficial in diagnosing the stress of individual trees and, therefore the forest. The objectives of this study are:

(1) Applying hyperspectral imaging technique to build a plant species hypercube dataset consisting of 1500 images of plant species to support developing ML models for the plant species classification,
(2) Investigating the feasibility of applying published deep learning architectures to the species classification based on spectral-textural information of plant live-crown images,
(3) Proposing a lightweight CNN model to catch plant live-crown features in the hyperspectral images to achieve optimistic classification performance, and
(4) Exploring the appropriateness of feature selection for hyperspectral images in species classification and the influence of using a limited number of spectral bands on classification performance.

# HYPERSPECTRAL DATA COLLECTION

## Plant Preparation

As mentioned above, this study aims to recognize plant species based on plant morphology *via* features of leaf color, size, contour, surface, venation, and phyllotaxy. Thirty species of foliage plants from 27 genera and 18 families were collected to produce plant images for analysis. To increase the leaf features and plant geometry diversity in the images, at least 2 or 3 plant individuals were gathered for replications. As shown in **Figure 1**, leaf features of the plants appeared similar or dissimilar in color, size, venation, and leaf edge. Detailed taxonomy information of the species is shown in **Table 1**.

## Plant Image Acquisition

The IMEC Snapscan VNIR B150 imaging system[1] was used to capture the hyperspectral images of the species. This system composes of the spectral image sensor, HSI camera, optics, and some other components that can acquire hypercube datasets up to a full-image size of 3,600 × 2,048 pixels covering a spectrum range from 458 nm to 913 nm. The system's spectral and radiometric resolutions are 2.8 nm (equivalent to 161 bands) and 10 bits. In the image acquisition, the camera is mounted on a tripod facing downward to the plant at a distance of 40 and 60 cm. Two 50 w/12 V halogen lamps were deployed, one on each side of the plant at a 45-degree elevation angle from the horizontal plane. A black material was used to minimize the background/neighboring material reflectance effects on the target reflectance. The aperture of the camera was set to f5.6 for every single snapshot. Due to the vertical and horizontal variations of the leaves locations, changing the orientation of the plant led to changes in light intensity over the crown area and therefore helped to increase the diversity of the sample images. With the fixed positions of the two light sources, the plant was set to rotate 90 degrees to generate diverse hyperspectral images of the same plant. The image-acquisition scheme is shown in **Figure 2**. Accordingly, the snapshot acquisition produced a hypercube raw image with a dimension of 1,200 rows × 1,200 columns × 161 bands, and a dynamic range of 10 bits. With the combination of two camera-target distances and four plant orientations, eight HSI raw images of every individual plant of the 30 species were acquired. Due to the significant noise in the wavelengths at both ends of the sensor, the raw image was spectrally subset to 147 bands with a spectrum range of 468–898 nm for the analysis.

### Data Calibration

To eliminate the impact of inconsistent image quality caused by the environmental factors, such as different illuminations or sensor response, each acquired HSI raw image *Ro* was calibrated with the formula to derive the HSI reflectance image $R_f$:

$$R_f = \frac{R_o - I_B}{I_w - I_B} * 100\%, \tag{1}$$

where $I_B$ denotes the dark reference image with 0% reflectance recorded with the lens closed, and $I_w$ presents the white reference image with more than 95% reflectance recorded with white a Teflon panel.

### Hypercube Dataset Preparation

To increase the total number of images for DL and reduce the computational complexity of training a CNN model, we adopted the following steps to segment a large image into multiple smaller sub-images. First, each 1,200 × 1,200 HSI reflectance image is evenly segmented into nine non-overlapping 400 × 400 sub-images. Then, those sub-images with a noticeable shadow or insufficient leaves, e.g., the leaf/background ratio does not exceed 60%, were removed. As a result, 50 sub-images were inspected and retained for each species, and a total of 1,500 HSI reflectance images (hereafter hypercube images) were generated

[1] https://www.imec-int.com/en/

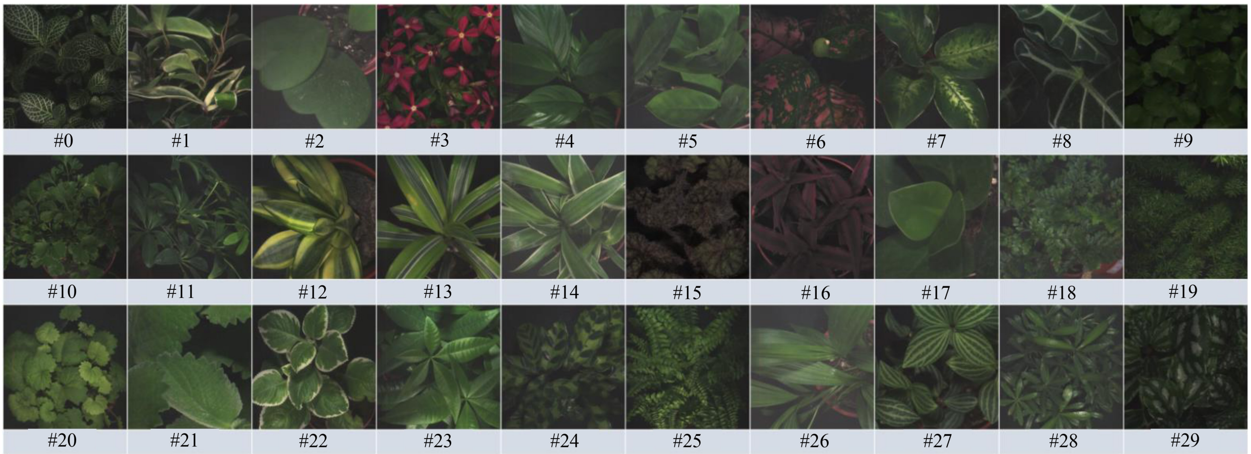


**FIGURE 1 |** The RGB sample images of the 30 plant species.

**TABLE 1 |** The taxonomy information and image samples list of the 30 plant species.

| ID | Family | Scientific name | Abbreviation | Number of individuals | Number of full-size images | Number of sub-images |
|---|---|---|---|---|---|---|
| 0 | Acanthaceae | *Fittonia albivenis* | F.a | 2 | 16 | 50 |
| 1 | Apocynaceae | *Hoya carnosa* | H.c | 2 | 16 | 50 |
| 2 | Apocynaceae | *Hoya kerrii* | H.k | 2 | 16 | 50 |
| 3 | Apocynaceae | *Ammocallis rosea* | A.r | 2 | 16 | 50 |
| 4 | Araceae | *Spathiphyllum kochii* | S.k | 2 | 16 | 50 |
| 5 | Araceae | *Zamioculcas zamiifolia* | Z.z | 2 | 16 | 50 |
| 6 | Araceae | *Aglaonema anyamanee* | A.an | 2 | 16 | 50 |
| 7 | Araceae | *Aglaonema commutatum* | A.c | 2 | 16 | 50 |
| 8 | Araceae | *Alocasia amazonica* | A.am | 2 | 16 | 50 |
| 9 | Araliaceae | *Hydrocotyle verticillata* | H.v | 2 | 16 | 50 |
| 10 | Araliaceae | *Polyscias guilfoylei* | P.g | 2 | 16 | 50 |
| 11 | Araliaceae | *Schefflera arboricola* | S.a | 2 | 16 | 50 |
| 12 | Asparagaceae | *Sansevieria trifasciata* | S.t | 3 | 24 | 50 |
| 13 | Asparagaceae | *Dracaena marginata* | D.m | 2 | 16 | 50 |
| 14 | Asparagaceae | *Chlorophytum comosum* | C.c | 2 | 16 | 50 |
| 15 | Begoniaceae | *Begonia cathayana* | B.c | 3 | 24 | 50 |
| 16 | Bromeliaceae | *Cryptanthus bivittatus* | C.b | 3 | 24 | 50 |
| 17 | Clusiaceae | *Clusia rosea* | C.r | 2 | 16 | 50 |
| 18 | Davalliaceae | *Davallia griffithiana* | D.g | 2 | 16 | 50 |
| 19 | Haloragaceae | *Myriophyllum aquaticum* | M.a | 2 | 16 | 50 |
| 20 | Lamiaceae | *Glechoma hederacea* | G.h | 2 | 16 | 50 |
| 21 | Lamiaceae | *Plectranthus amboinicus* | P.am | 2 | 16 | 50 |
| 22 | Lamiaceae | *Plectranthus amboinicus cv.* | P.a.cv | 2 | 16 | 50 |
| 23 | Malvaceae | *Pachira aquatica* | P.aq | 2 | 16 | 50 |
| 24 | Marantaceae | *Calathea lancifolia* | C.l | 3 | 24 | 50 |
| 25 | Nephrolepidaceae | *Nephrolepis exaltata* | N.e | 2 | 16 | 50 |
| 26 | Orchidaceae | *Spathoglottis plicata* | S.p | 2 | 16 | 50 |
| 27 | Piperaceae | *Peperomia puteolata* | P.p | 2 | 16 | 50 |
| 28 | Podocarpaceae | *Podocarpus macrophyllus* | P.m | 3 | 24 | 50 |
| 29 | Urticaceae | *Pilea cadierei* | P.c | 2 | 16 | 50 |

for the study. It is worth noting that each sub-image still retains sufficient spatial information of which type of plant it belongs to. The overall process is illustrated in **Figure 3**. After that, the images were randomly divided into training and test datasets. The former contains 1,200 hypercube images, and the latter has 300 ones. **Figure 4** shows some example images of the species at some selected wavelengths in the visible and NIR regions. The reflectance and the features of leaves are retained in each band of the hypercube image.

## Deriving Representative Spectra of the Plant Species

In remote sensing, a reflectance curve is typically representative of the spectral behavior of an object. In addition to the deep learning approach, this study also investigates the spectral reflectance of the plant species. The representative reflectance at each wavelength of the visible-NIR region is determined as the average of the plant leaves. To do this, it is necessary to eliminate background components to extract the region-of-interest (ROI), referred to as leaf regions. The steps to extract the representative reflectance curve of the species are described as follows.

First, two particular bands with the largest globally average reflectance, abbreviated maxBand, and the smallest globally average reflectance, abbreviated minBand, among all bands of a hypercube image were identified. Second, a different image of the two specific bands is determined as maxBand – minBand. Third, a thresholding method (Ma et al., 2015) is used to differentiate the different images into two parts, i.e., the region-of-interest vs. the background. The threshold value was set to be 0.3, determined based on the experience. Fourth, a regional-averaging model is adopted to the hypercube image to calculate the average reflectance of every pixel in the ROI. Fifth, the representative reflectance of a particular wavelength of a Plant Species is generalized as the mean of all the corresponding average values of the 50 hypercube images of the species. Finally, the full-wavelength reflectance Spectra of the species are restored by assembling the representative reflectance at every wavelength. **Figure 5** illustrates the overall procedures for deriving the generalized reflectance curve.

# METHODS

## The Lightweight Convolution Neural Network Architecture

Considering the limitation of gathering a large number of species and images, this study followed the concept of "compact" in ML to design the lightweight CNN (LtCNN) for better modeling fitting. The LtCNN model is developed by referring to GoogLeNet (Szegedy et al., 2015) and other networks (Ioffe and Szegedy, 2015; Simonyan and Zisserman, 2015; Szegedy et al., 2015). As shown in **Figure 6**, the architecture of LtCNN is only composed of three parts. The first two are responsible for feature extraction, and the last one is for prediction. The details of those parts are explained below, and the setting of network parameters of the LtCNN model is summarized in **Table 2**.

**Part I:** Part I aims to convert the input image into low-level (or shallow) features as the input of Part II. It comprises two convolutional layers (5 × 5 and 3 × 3) and one pooling layer.

**Part II:** The objective of Part II is to learn the high-level features in a multi-scale manner as the input of Part III. It adopts three "Inception modules" originating from GoogLeNet. Our first two inception modules adopt a 3-path structure and replace the 5 × 5 convolution in the original version with two 3 × 3 ones to reduce the number of parameters while maintaining the same receptive field. The third inception module only adopts a 2-path structure since the size of the feature map has been reduced.

**Part III:** Aims to perform classification *via* the features received from Part II. It uses global average pooling (GAP) to integrate all the features and then applies two fully-connected (FC) layers, one dropout layer, and a Softmax classifier to predict the species of the input image.

## Loss Function

The cross-entropy is selected as the loss function to measure the difference between two probability distributions of the target ground truth and the model's prediction. It is defined by

$$loss_{CE} = \sum_{c=1}^{C} \sum_{i=1}^{S} -y_{c,i} \log_2(p_{c,i}), \tag{2}$$

where $C$ stands for the number of classes, $S$ denotes the batch size, $y_{c,i}$ is a binary indicator, and $p_{c,i}$ is the predicted probability. In our experiment, we set $S = 12$ and $C = 30$.

## Experimental Setting

The experiments were implemented on the hardware environment with an Intel i7-7700k CPU, 32 GB RAM, and NVIDIA GTX-1080Ti GPU. Three well-known CNN models such as AlexNet (Krizhevsky et al., 2017), VGGNet (Simonyan and Zisserman, 2015), and GoogLeNet (Szegedy et al., 2015) were applied as a referring method for Performance comparison of the Species recognition/Classification. Since AlexNet was designed for the classification of a large number of categories with very deep neural networks, the number of neurons of FC layers was reduced. Specifically, the number of output classes was set to 30 in this study. This model is, therefore, named AlexNetr. Similarly, it is difficult to reach convergence when training the original VGGNet (16 layers) on our plant dataset. The original architecture of VGGNet (16 layers) is therefore simplified by preserving the first eight convolution layers and three FC layers and reducing the number of neurons in the FC layers. It is named VGGNetr in this study. Similar to the LtCNN, a ReLu activation function is applied to improve the nonlinearity.

All the models are trained from scratch without pre-trained parameters or transfer learning techniques. They are implemented on Tensorflow 1.8.0. The size of the input is set to 200 × 200 × $L$ for our lightweight model and 224 × 224 × $L$ for other CNN models, where $L$ denotes the number of selected bands. If we set $L = 6$, the number of parameters of AlexNetr, GoogLeNet, VGGNetr, and the proposed lightweight model are 10865310, 15901982, 10404938, and 1388950, respectively. For data augmentation, we used random crop, random flip in

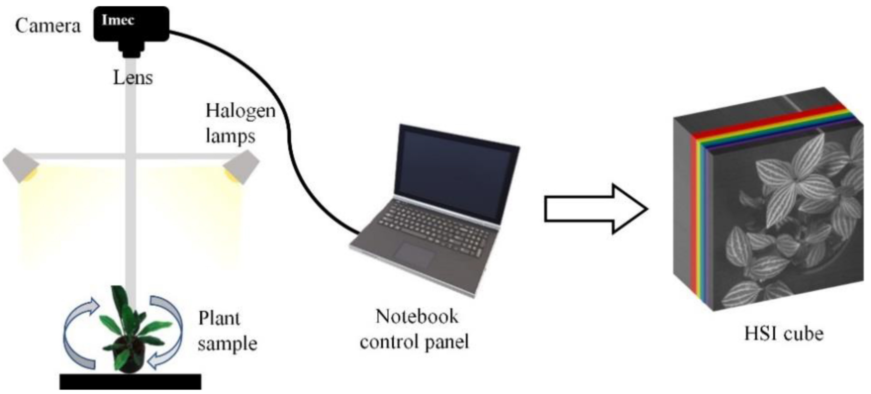


**FIGURE 2 |** Illustration of plant image acquisition.

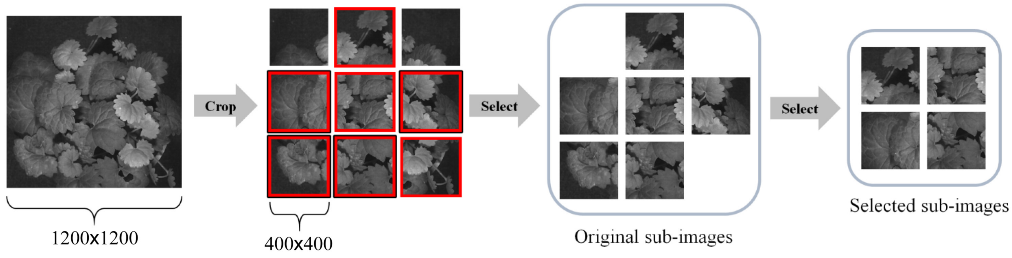


**FIGURE 3 |** An illustration of sub-image generation.

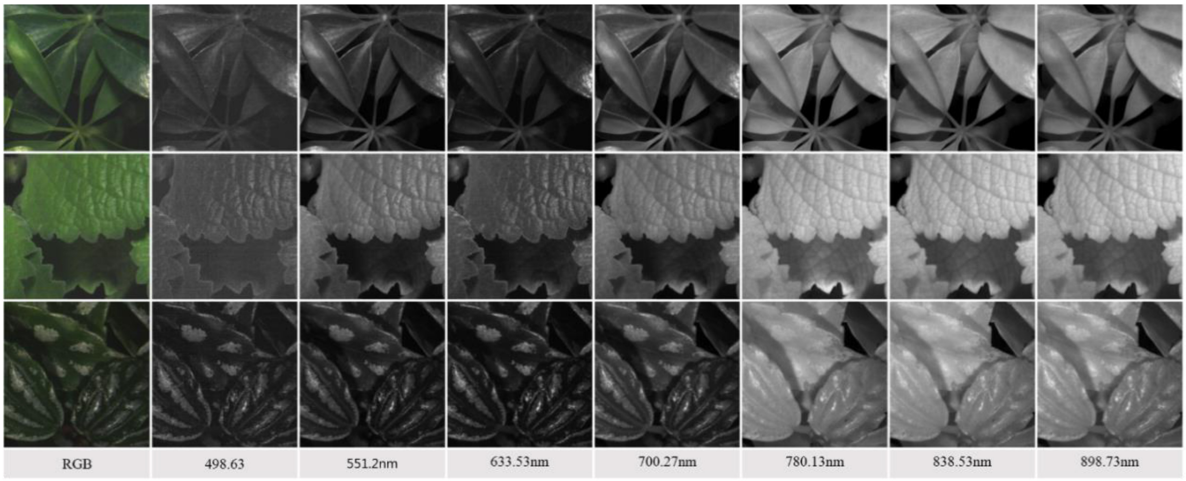


**FIGURE 4 |** Examples of the reflectance sub-images of plant species (Top: S.a, #11; Middle: P.am, #21; Bottom: P.c, #29) at some selected wavelengths in the visible-NIR region.

horizontal or vertical to expand the size of the training dataset. For parameter settings, the batch size was set to 12. The learning rate was set to be exponential decay with an initial rate of $10^{-3}$ and a decay rate of 0.9 for every 5 epochs. The training epoch was assigned as 200, and the optimizer was ADAM. To evaluate model performance, four quantitative metrics were used: overall accuracy (OA), precision, macro F1-Score, and kappa coefficient.

## Feature Selection Methods

The hyperspectral image bands are mostly correlated, particularly those in a similar spectral region. Using full spectral bands for data analysis may lead to the curse of Dimensionality (Hughes, 1968) and increase the computational burden. To achieve a better calculation efficiency while retaining classification accuracy, data dimensionality reduction (Chang, 2013) is required to

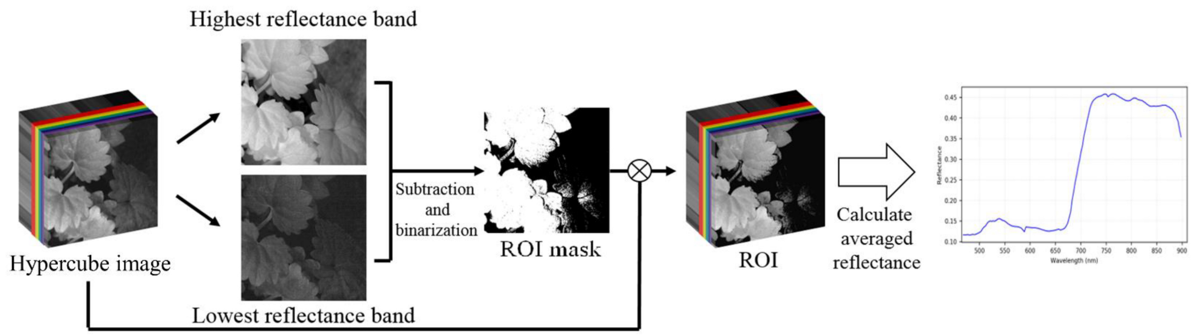


**FIGURE 5 |** An illustration of leaf region extraction and spectral reflectance generalization of plant species.

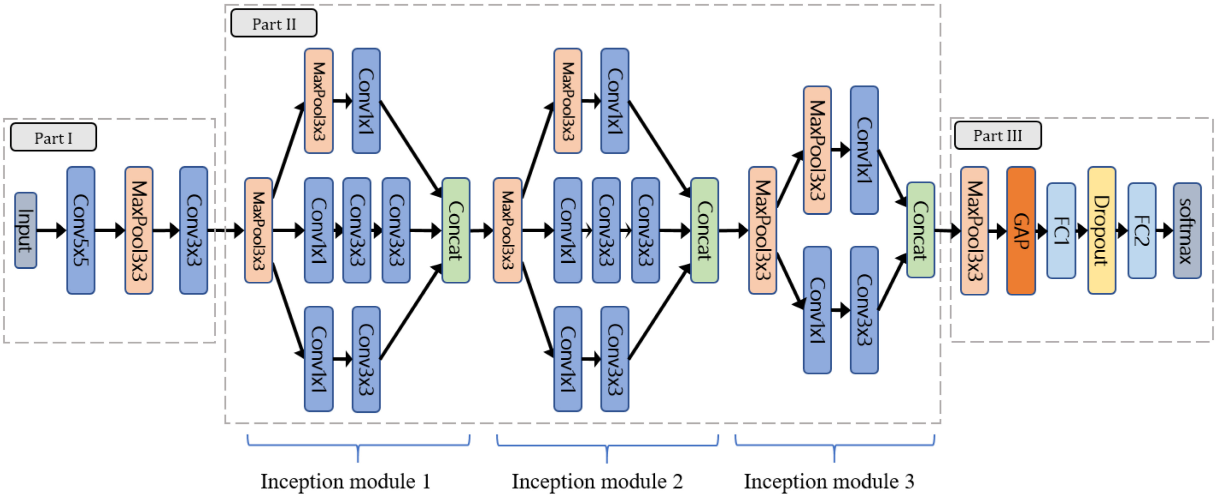


**FIGURE 6 |** The architecture of the proposed lightweight convolutional neural network (LtCNN). Conv is the convolution layer, Concat means concatenation operation, MaxPool denotes max-pooling layer, GAP stands for global average pooling layer, and FC means fully connected layer. A ReLu function follows every CNN process to increase the network's nonlinearity.

select critical bands or discriminative spectral features for the Species classification/recognition with DL techniques. The band selection was made *via* two approaches: manual inspection of the reflectance curves and automatic selection based on the spectral heterogeneity of bands. The methods are summarized in **Table 3**, and the suggested bands with the corresponding wavelength are listed in **Table 4**.

# RESULTS

## Use of the Visible-Infrared Spectra Variation of Plant Species for Classification

A generalized reflectance curve of species leaves (average spectra) is essential for differentiating and labeling pixels in a pixel-based classification. **Figure 7** shows the averaged spectra of the ROI regions of the 30 plant species, where the x-axis denotes wavelength and the y-axis indicates reflectance values. Each curve was drawn by averaging the spectral reflectance vectors of all the leaf pixels of that particular hypercube image of a species. As can be seen, the reflectance spectra of all species vary at each of the wavelengths. At the same time, the particular features of green peaks, blue and red valleys, and near-infrared plateau remain evident and visually differentiated. Due to the complicated light environment in leaf pixels and even a natural variety of leaf colors for the same species, the reflectance of the species changed dramatically and consequently showed a wide SD band along the visible-infrared regions. The high variation of reflectance of the same materials will lead to difficulty of species classification using pixel-based methods. For example, In **Figure 8**, the leaf of species D.m (#13) has a white line feature distributed from the bottom to the top of the leaf rib, but species C.c (#14) has two white stripes on the leaf edges. In contrast to the all-green-leaf image of species Z.z (#5), the red spots randomly distributed over the leaf mesophylls of species A.an (#6) make it more challenging for pixel-based species classification.

As noted in the subfigures on the right column of **Figure 8**, four hypercube images of species A.an highlight the difference

**TABLE 2 |** Detailed network parameters of the proposed lightweight CNN model.

| Parameter | Kernel size | Depth | Strides | Output |
|---|---|---|---|---|
| Input | | | | 200 × 200 × L |
| Convolution | 5 × 5 | 1 | 2 | 100 × 100 × 64 |
| Max pooling | 3 × 3 | 0 | 2 | 50 × 50 × 64 |
| Convolution | 3 × 3 | 1 | 1 | 50 × 50 × 96 |
| Max pooling | 3 × 3 | 0 | 2 | 25 × 25 × 96 |
| Inception module1 | | 2 | | 25 × 25 × 288 |
| Inception module2 | | 2 | | 13 × 13 × 296 |
| Inception module3 | | 2 | | 7 × 7 × 480 |
| Global average pooling | 7 × 7 | 0 | 1 | 1 × 1 × 480 |
| FC layer | | 1 | | 1 × 1 × 196 |
| Dropout (40%) | | 0 | | |
| Output | | 1 | | 1 × 1 × 30 |
| Softmax | | 0 | | 1 × 1 × 30 |

The number of parameters (Input dimension = 6): 1388950

**TABLE 3 |** A summary of the manual and automatic approaches for band selection.

| Approach | Description | Source |
|---|---|---|
| RGB | Three bands can be used to simulate the normal-color RGB image. The IMEC hyperspectral sensor recommends the bands. | IMEC Snapscan v1.1.2 |
| NIR | Three near-infrared bands are used to simulate a false-color RGB image and are used as a comparison of the normal-color RGB image. Bands are selected according to the reflectance curve, as shown in **Figure 7**. | Visually inspection |
| RGB+NIR | The combination of natural-color and false-color images is mentioned above. It is used to compensate for the spectral information absent in each of the two images. | |
| PCA | The principal component analysis (PCA) transforms the hyperspectral image to principal components (PCs). Only the first six PCs were selected for they retained over 99% energy of eigenvalues. | Gao et al., 2018; Ma et al., 2015. |
| UBS | The uniform band selection (UBS) is a typical band selection algorithm based on sampling with equal intervals in the whole spectrum. The full range of wavelengths of the IMEC VNIR sensor is divided into 3, 6, and 9 sub-regions. The datasets with 3, 6, and 9 bands are UBS-3, UBS-6, and UBS-9. | Li et al., 2019. |
| FNGBS | FNGBS stands for the Fast Neighborhood Grouping Band Selection algorithm that partitions the global wavelengths of an HSI cube into M groups based on a coarse-fine strategy and selects the band with the maximum product of local density and information entropy from each group to obtain a subset with M bands for application. In this study, the selected datasets with 3, 6, and 9 bands are abbreviated as FNGBS-3, FNGBS-6, and FNGBS-9. | Wang et al., 2020. |

between the global and local mean reflectance curves. The former is derived from every pixel of the whole image (the blue curve), and the latter is derived from those red leaf pixels (the red curve). In the visible region, the red-curve spectra spread far from the blue curve and locate almost close to the border of the standard deviation band of the global mean spectra. In contrast, the red curve in the infrared region distributes very close to that of the blue curve. These subfigures reveal that infrared reflectance of leaves is not correlated to the leaf color but to the chlorophyll contents and water contents in mesophyll tissues. In other words, the infrared reflectance of leaves behaves very similarly for the same species as the standard curve. According to Lin et al. (2012, 2015c), leaf reflectance over the VNIR-SWIR region may affected by leaf water and concentration contents. Their study first highlighted the effect of water stress on chlorophyll concentration estimation and further proposed effective chlorophyll indices to account for the influence of water content to achieve accurate estimation of leaf chlorophyll concentration. Therefore, the dissimilarity in the infrared reflectance among hypercube images indicates the possibility of species difference or physiological stress such as water content shortage. Including near-infrared spectra with visible spectra is beneficial to species classification because leaf pattern features and mesophyll structure are considered simultaneously.

## An Overall Assessment of Species Classification Accuracy for the Four Deep Learning Models

**Table 5** shows the accuracy measures of the CNN models performing on 6 different spectral features combinations of the

**TABLE 4 |** The selected bands and corresponding wavelengths of the hypercube image for species classification.

| Approach | Bands | The selected band no. | Representative wavelengths (nm) of the corresponding bands |
|---|---|---|---|
| RGB | 3 | 2/19/39 | 471.44/535.06/602.94 |
| NIR | 3 | 89/109/126 | 750.75/799.99/851.04 |
| RGB+NIR | 6 | 2/19/39/89/109/126 | 471.44/535.06/602.94/750.75/799.99/851.04 |
| PCA | 6 | PC1-PC6 | A component is a linear transformation of bands as the input |
| UBS | 3 | 1/74/147 | 468.63/700.27/898.72 |
| | 6 | 1/30/59/88/117/147 | 468.63/569.27/664.12/747.22/824.74/898.72 |
| | 9 | 1/19/37/55/74/92/110/128/147 | 468.63/535.06/594.66/651.37/700.27/761.18/802.71/856.68/898.72 |
| FNGBS | 3 | 36/68/92 | 591.46/681.80/761.18 |
| | 6 | 14/25/68/92/104/118 | 514.83/551.20/681.80/761.18/783.39/827.35 |
| | 9 | 15/32/33/69/80/92/103/118/137 | 520.08/557.84/571.04/685.36/721.58/761.18/780.53/827.35/871.66 |

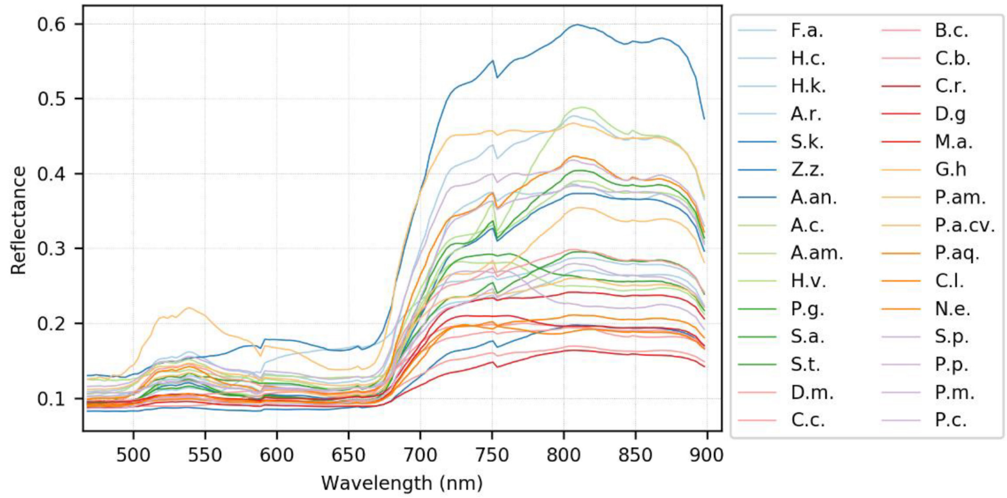


**FIGURE 7 |** The generalized reflectance curve of the plant species. Refer to **Table 1** for the abbreviation of the species.

plant hypercube dataset. In the classification with the natural-color RGB bands, the AlexNetr, GoogLeNet, VGGNetr achieved an OA of 76.3, 71.7, and 79.9%, a macro F1-score of 0.758, 0.699, and 0.782, and a kappa value of 0.755, 0.707, and 0.790, respectively. Each of the models had a precision of 0.832, 0.705, and 0.806, which indicates that the AlexNetr model and VGGNetr model showed better adaptability in retrieving information of species leaf features and leaf structure in RGB reflectance and therefore achieved a prediction with lower commission error or false positive than the GoogLeNet model. In contrast, the LtCNN model performed at the best accuracy with a value of OA = 89.7%, macro F1-score = 0.896, and kappa = 0.893 which is correspondingly higher than the previous models by 10–18%, 0.11–0.19, and 0.10–0.19. Using only RGB spectral features, the LtCNN model performed species classification with a commission error around 0.1.

The accuracy measures decreased significantly when checking with the classification results using three NIR bands or the simulated false-color RGB bands, as mentioned in **Table 5**. For example, the decrease of OA was 15, 5, 10, and 21% for the AlexNetr model, GoogLeNet model, VGGNetr model, and LtCNN model, respectively. These four DL architectures performed the species classification at an OA between 60 and 70% using only NIR-based false-color images. Obviously, the natural-color RGB images provide more diverse spectral information and inherently spatial information of the species than the false-color NIR images. Such cases are due to some species having a similar leaf mesophyll structure (Hopkins and Hüner, 2004; Lin et al., 2015c) and behaving similarly in the near-infrared bands. As shown in **Figure 7**, the reflectance in the visible-NIR region varied dramatically and overlapped significantly. This leads to a higher degree of omission and commission error in it. In contrast, the natural-color RGB is supposed to catch leaf color, shape, and surface texture changes and consequently contribute species classification accuracy.

In general, the reflectance of RGB bands is low correlated to the NIR bands. The leaf features derived simultaneously from the RGB natural-color bands (471.44, 535.06, and 602.94 nm) and the NIR false-color bands (750.75, 799.99, and 851.04 nm) are assumed to be of benefit to species classification. However, as noted in **Table 5**, the kappa coefficient achieved by the GoogLeNet model was 0.693, which is even slightly smaller than 0.707, the performance baseline achieved in the classification using only the RGB natural-color bands. In contrast, the AlexNetr, VGGNetr, and LtCNN models revealed a lively performance as the OA, F1-score, and kappa significantly increased by nearly 5%, 0.05, and 0.05, respectively. This verifies that additional NIR bands in respect to the RGB basic spectral information are beneficial to species classification.

Although the principal component analysis (PCA) method can transform the spectral information of bands in the hypercube image into several components, the classification using most informational details through the four CNN models did not perform better than the RGB bands' baseline. For example, the F1-score of PCA and RGB for the AlexNetr, GoogLeNet, VGGNetr, and LtCNN was 0.703/0.758, 0.700/0.699, 0.778/0.782, and 0.881/0.896, respectively. The result implies that the PCA is most likely inappropriate for use in the reduction of the dimensionality of hypercube images in the view of species classification *via* DL. Considering the performance improvement of RGB+NIR classification, the linear transformation of hyperspectral bands most likely destroyed the physical properties of the materials in each band, thereby weakening the spatial relationship between the features or different tissues on the leaf which decreases the classification ability of a CNN.

## A Comprehensive Examination of the Species Confusion in the Models

To illustrate the prediction results of the four CNN models more comprehensively, a confusion matrix is used to examine

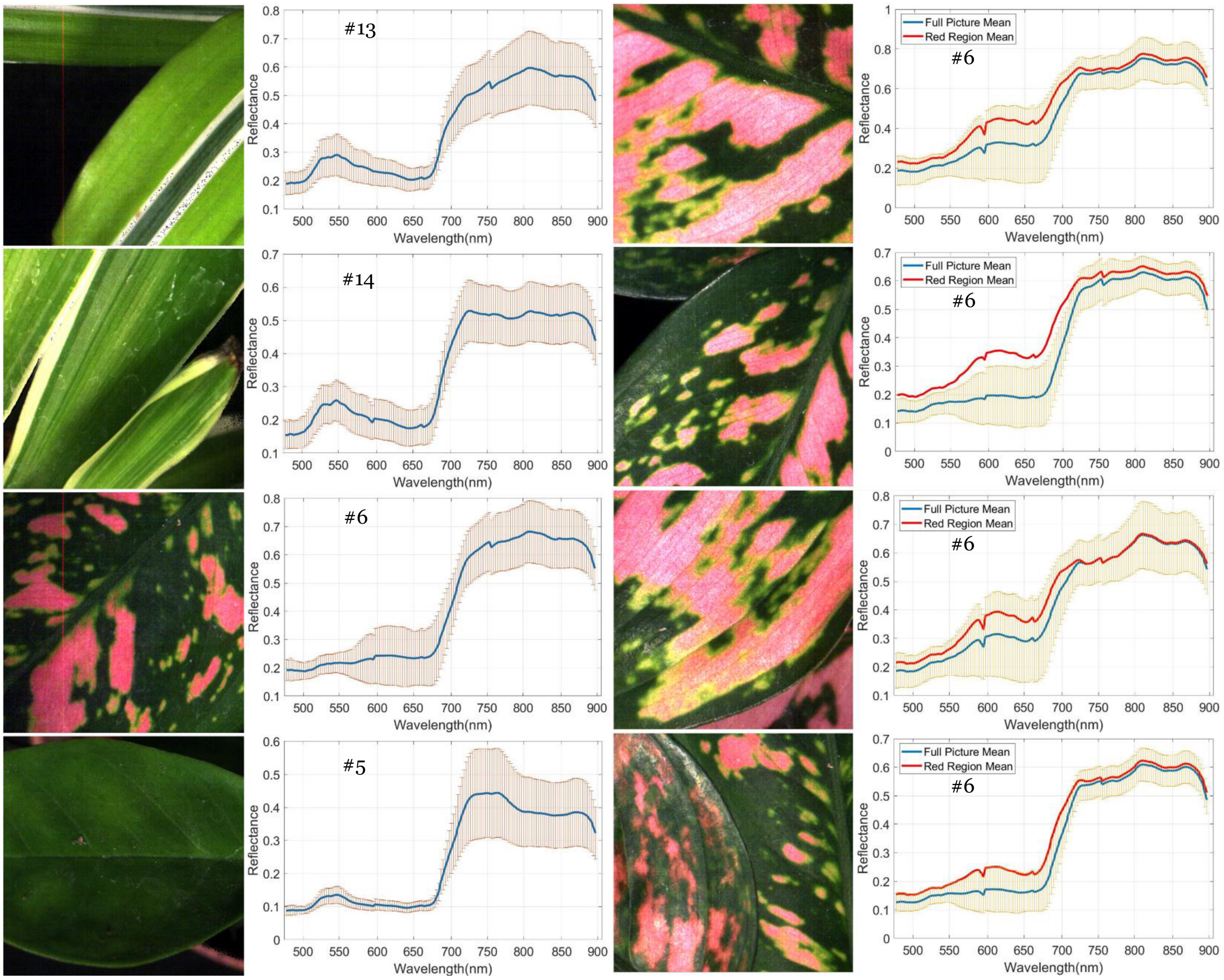


**FIGURE 8 |** Variation of spectral reflectance in hypercube image of plant leaves. Left: the generalized mean curve and standard deviation ring of the whole leaf pixels for species D.m (#13), C.c (#14), A.an (#6), and Z.z (#5). Right: the difference between the generalized mean curve of the red leaf pixels and that of the global leaf pixels for species A.an. The ring overlapped with the mean curves is the SD of the global leaf pixels.

the confusion among the species for the classification scenario using the RGB+NIR dataset. In **Figure 9**, the matrix entries contain the numbers of prediction rates of the four models, which are marked with different colored squares. The value of one in diagonal entries specifies the classification of 100% from the view of ground truth. The perfect true-positive rate indicates the superior excellence of a CNN model in describing the leaf features of the plant species. As can be seen, there were 10, 4, 13, and 22 species being classified with 100% of true positive rate for the AlexNetr, GoogLeNet, VGGNetr, and LtCNN models, respectively. As noted in **Figure 9**, the species *Ammocallis rosea* (A.r, #3) was completely misclassified by the GoogLeNet model. It is mostly recognized as *Sansevieria trifasciata* (S.t, #12) with a false-negative rate of 0.6 while as *Aglaonema commutatum* (A.c, #7), *Schefflera arboricola* (S.a, #11), and *Hoya carnosa* (H.c, #1) with false-negative rate 0.2, 0.1, and 0.1, respectively. Interestingly, this species was recognized accurately by the other three models. Comparing the appearance of species #3, #12, #7, and #1, the flowers of #3 in the hypercube images seem not to work like a feature but a noise in the GoogLeNet model.

Of the 30 plant species, the LtCNN model failed to completely and accurately recognize every image of 8 species, which are *Spathiphyllum kochii* (abbreviated S.a with the species identity #4), *Zamioculcas zamiifolia* (abbreviated Z.z, #5), *Alocasia amazonica* (A.am, #8), *Clusia rosea* (C.r, #17), *Glechoma hederacea* (G.h, #20), *Plectranthus amboinicus cv.* (P.am.cv, #22), *Spathoglottis plicata* (S.p, #26), and *Podocarpus macrophyllus* (P.m, #28) with a true-positive rate of 0.9, 0.5, 0.9, 0.7, 0.9, 0.8, 0.9, and 0.8, respectively. Poorer confidence of classification occurred in species #5 and #17. The false-negative in species #17 is mainly due to the lack of the full leaf shape in the sub-images randomly generated during the convolution, which resulted in a partial leaf and therefore increased the feature similarity of species #17, #5, and #2. Looking into the false-negative classification of species #5, whose images were misclassified as the species #4, #11, and #10 with a rate of 20, 20, and 10%, respectively. These species are

**TABLE 5 |** Classification performance of different CNN models applied to different feature selection settings.

| CNN model | Feature selection ¶ | OA (%) | Precision | Macro F1-score | Kappa coefficient |
|---|---|---|---|---|---|
| AlexNetr | RGB (3) | 76.30 | 0.832 | 0.758 | 0.755 |
| | NIR (3) | 60.70 | 0.637 | 0.601 | 0.593 |
| | RGB+NIR (3+3) | 82.30 | 0.836 | 0.824 | 0.817 |
| | PCA (6) | 70.70 | 0.716 | 0.703 | 0.697 |
| GoogLeNet | RGB (3) | 71.70 | 0.705 | 0.699 | 0.707 |
| | NIR (3) | 66.00 | 0.635 | 0.640 | 0.648 |
| | RGB+NIR (3+3) | 70.30 | 0.668 | 0.678 | 0.693 |
| | PCA (6) | 71.00 | 0.708 | 0.700 | 0.700 |
| VGGNetr | RGB (3) | 79.70 | 0.806 | 0.782 | 0.790 |
| | NIR (3) | **69.70** | **0.737** | **0.686** | **0.686** |
| | RGB+NIR (3+3) | 84.00 | 0.88 | 0.838 | 0.834 |
| | PCA (6) | 78.00 | 0.803 | 0.778 | 0.772 |
| LtCNN | RGB (3) | **89.70** | **0.903** | **0.896** | **0.893** |
| | NIR (3) | 68.00 | 0.728 | 0.674 | 0.669 |
| | RGB+NIR (3+3) | **94.70** | **0.950** | **0.945** | **0.945** |
| | PCA (6) | **88.00** | **0.898** | **0.881** | **0.876** |

*¶The numbers in parentheses represent the feature dimensionality used in the classification. The bold number in each column of the four accuracy measures indicates the best performance achieved by the corresponding CNN model with the selected features.*

visibly differentiated based on leaf margin, surface leathery, and petiole features, but the LtCNN model misclassified 50% of the species. The misclassification is also evident in the other models. This is highly probably due to the inability of retaining leaf margin (serrate and entire), surface leathery, and petiole features during the convolution and pooling processes as the features are too small to detect with respect to the leaf area. **Figure 10** illustrates some examples of the confusion in species #3, #5, #13, and #14, and the excellent recognition in species #6 and #19 for the AlexNetr, GoogLeNet, VGGNetr, and LtCNN models. It is also noted that the images of species #13 and #14 were partially misclassified by AlexNetr, GoogLeNet, and VGGNetr models mainly due to the leaf shape similarity; meanwhile, the models missed their heterogeneous features. In contrast, the LtCNN model showed excellence in successfully learning the key features, and therefore the images were classified as the species.

# DISCUSSION

## Band Selection Contributes to Improving the Performance of Species Classification

The accuracy figures for the species classification using the dataset with the predetermined bands of RGB, NIR, or RGB+NIR in **Table 5** shows the proposed LtCNN is more appropriate than the AlexNetr, GoogLeNet, and VGGNetr for dealing with classification when using a smaller number of species classes. This section examines the contribution of diverse bands in species classification. With regards to 3-band classification, the Fast Neighborhood Grouping Band Selection (FNGBS) method suggested the bands #36, #68, and #92, whose wavelengths are located at the green-edge (591.46 nm), red-edge (681.80 nm), and near-infrared (761.18 nm), while the uniform band selection (UBS) suggested bands #1 (468.83 nm), #74 (700.27 nm), and #147 (898.72 nm) at the regions of blue, red-edge, and near-infrared. As can be seen in **Figure 11**, the sensitivity of spectral features is evident in each of the four CNN models. With the diverse spectral bands, the kappa changes dramatically. For example, the value dropped by 0.11 for the UBS but raised by 0.042 for the FNGBS in the AlexNetr model. Accordingly, the change rate was equivalent to 15 and 6% of the RGB's kappa value. In contrast, the GoogLeNet and LtCNN models appeared to be more flexible at catching the spectral features from the three bands suggested by band selection methods. The kappa value was increased nearly by 11∼12% from the baseline of 0.707 for the GoogLeNet model and by 4∼6% concerning the baseline of 0.893 for the LtCNN model for UBS and FNGBS, respectively. Similarly, the VGGNetr model achieved a classification with an increase of kappa value by 10% through the FNGBS suggested bands but failed to improve the performance through the UBS suggested bands.

When the number of spectral bands in a species classification is raised to 6, for example, the bands #14, #25, #68, #92, #104, and #118 selected by FNGBS, three of the models failed to improve the classification performance the exception being the AlexNetr model with an increase of kappa by 0.049 or 6% of the baseline for the RGB+NIR case. Similarly, the classification with a rise in kappa occurred only in the GoogLeNet model when the six bands #1, #30, #59, #88, #117, and #147 recommended by UBS were used for classification. The kappa value was improved from 0.693 to 0.772, and the increase rate was around 11%. The kappa value achieved by the LtCNN model *via* the two band-selection methods is very close to the 3-band case (0.938 vs. 0.945 for FNGBS and 0.948 vs. 0.931 for UBS), this indicates that as long as the band is selected appropriately, using only three bands can achieve a satisfactory classification accuracy.

To summarize, the CNN models appeared to be sensitive to the spectral features of a hypercube image when the number of bands used for species classification is subject to only three spectral bands. For such cases, the FNGBS method works more efficiently and can adapt to AlexNetr, GoogLeNet, VGGNetr, and LtCNN models. And, the LtCNN is the most significant of the four models to achieve reliable and stable classification performance with a minimum number of bands and the most informative spectra. Specifically, the most appropriate spectral features for species classification *via* the LtCNN model are the green-edge (591.46 nm), red-edge (681.80 nm), and near-infrared (761.18 nm).

## Appropriate Dimensionality of Hyperspectral Imaging Images in Recognizing and Classifying Plant Species

As noted in **Figure 11**, the four CNN models revealed diverse sensitivity of spectral bands in species classification. An

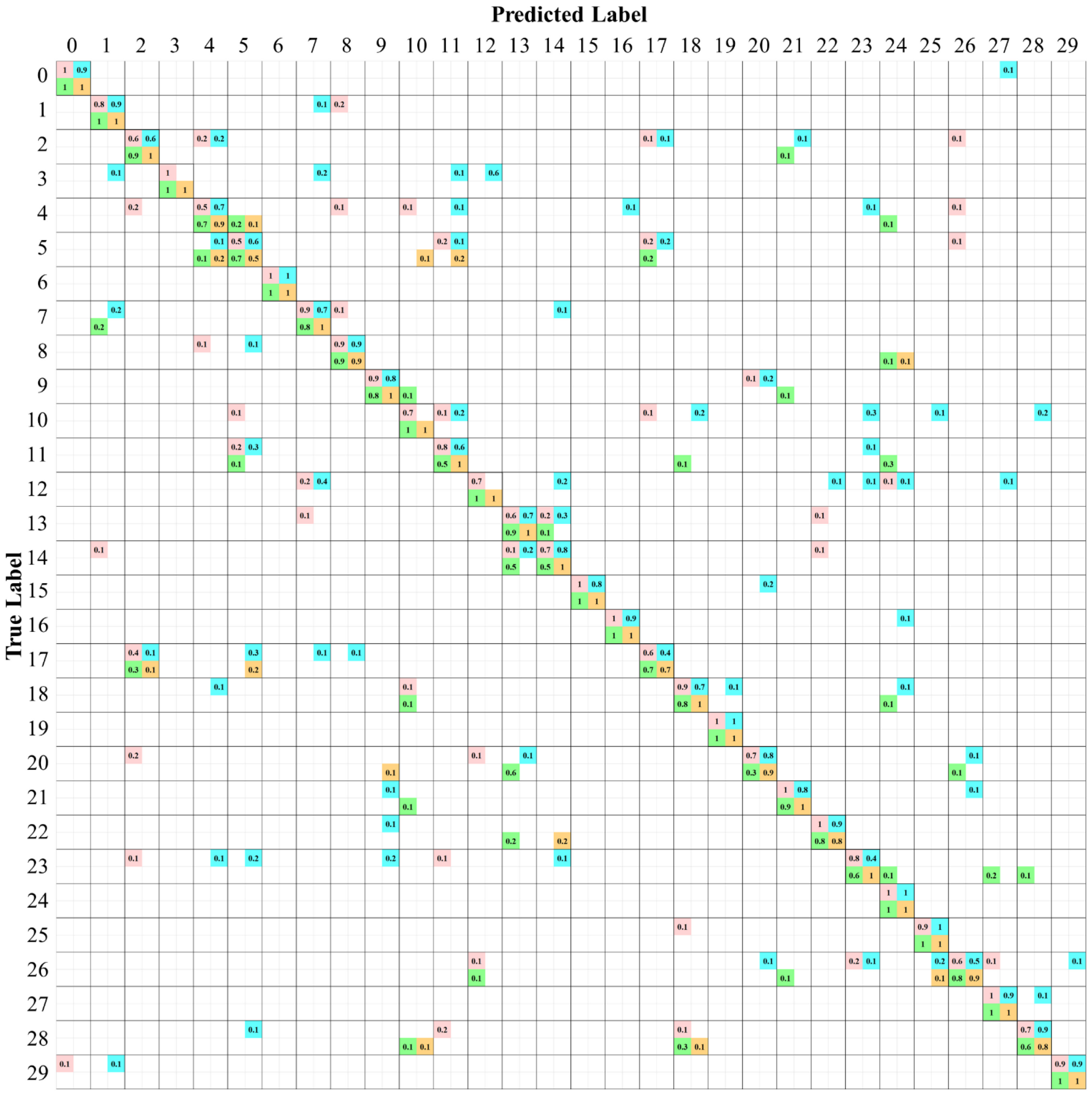


**FIGURE 9 |** Confusion matrices of species classification of the four CNN models using the six bands of the RGB+NIR dataset. The values in each diagonal entry are the probability of a species image being classified correctly. The numbers in upper/lower off-diagonal entries are the omission rate/commission rate. The numbers highlighted are for the models AlexNetr (pink), GoogLeNet (cyan), VGGNetr (green), and LtCNN (orange).

interesting question arises: Can using more bands help improve accuracy? Or what is the best accuracy achievable by the four models? To address this question this study conducted extended experiments by adding the number of bands progressively up to 60 with an interval of 3 as the input image of species classification for the CNN models. All the models are trained with the parameters mentioned in section "Experimental Setting", except for the VGGNetr model, because it was unable to handle higher dimensional data under our hardware environment. The adaptability of the AlexNetr, GoogLeNet, and LtCNN models to high-dimensional data is shown in **Figure 12**.

The x-axis presents the number of bands (M) in each subfigure, and the y-axis shows the corresponding accuracy measure. The yellow, blue, and red curves denote the accuracy trends of the three CNN models, respectively. From the point of view of the species classification, the main observation is that increasing M cannot help to improve accuracy and may even cause worse results. This phenomenon mainly occurred when using AlexNetr and GoogLeNet. The impacts of M on the CNN models are summarized below.

(1). For the cases of UBS, as shown in **Figure 12A**, the kappa coefficient of AlexNetr starts at 0.624 and increases to 0.779∼0.800 when $M$ = 6 to 18. As M increases, the accuracy is no longer improved but becomes unstable. The kappa of GoogLeNet starts at 0.797 and gradually decreases

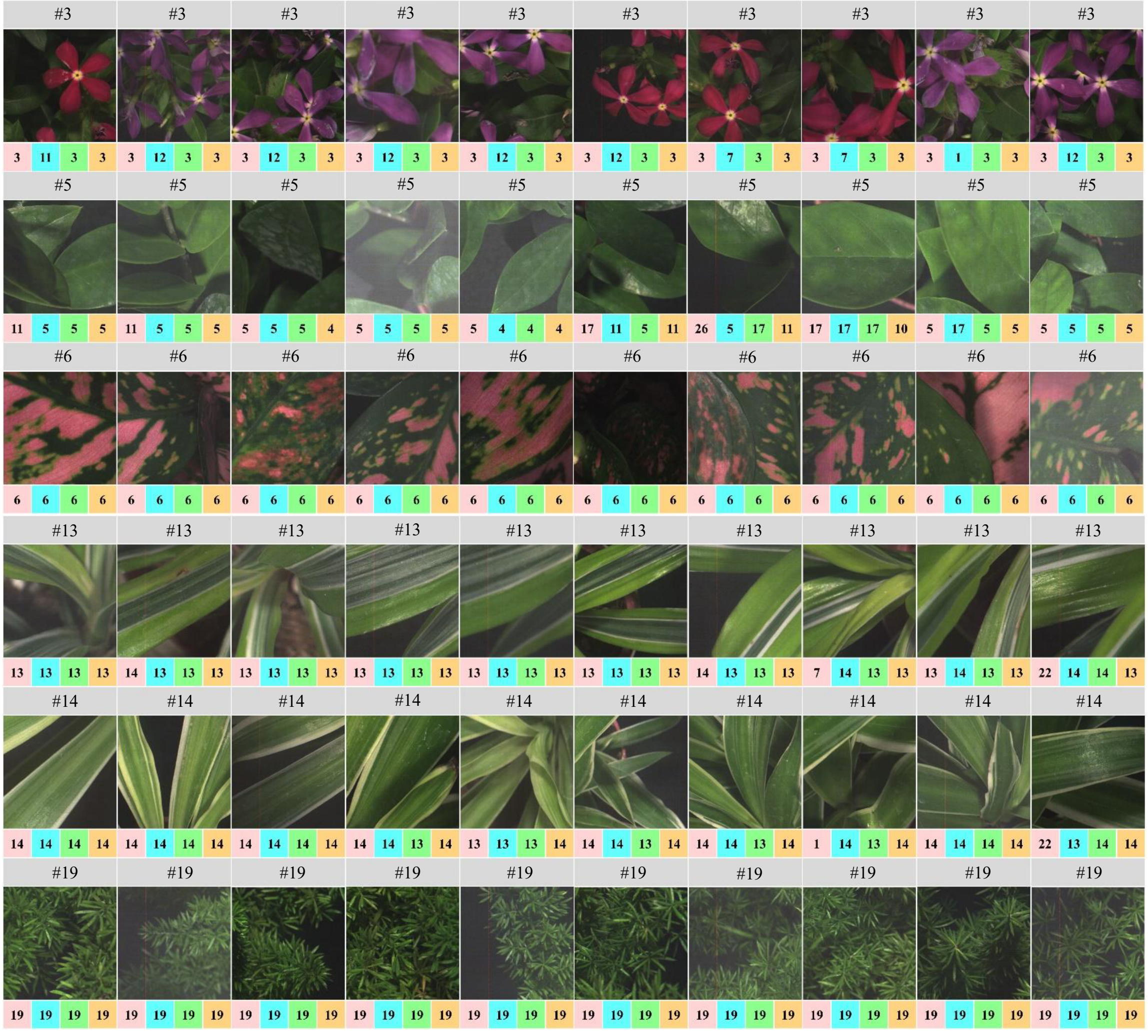


**FIGURE 10 |** An illustration of species classification by the four CNN models. Species #13 and #14 whose images are entirely recognized by the LtCNN model but eventually misclassified by the other models. The number above each image is the species identity, and the number highlighted by a color box indicates the classified label of the species by the models. Please refer to **Figure 8** for the color indication.

when M increases. For the proposed LtCNN model, the increase of M does not cause a significant change in the classification accuracy. The best value appears at $M = 15$, which is 0.976. After that, the accuracy curve maintains between 0.941 and 0.969. It implies that the amount of spectral information is saturated. On average, the kappa accuracy for the AlexNetr, GoogLeNet, and LtCNN models was 0.717 ± 0.079, 0.666 ± 0.084, and 0.951 ± 0.021, respectively.

(2). Similarly, as shown in **Figure 12B** for the cases of FNGBS, the trends generated by AlexNetr and GoogLeNet are gradually declining when M increases or fluctuates between 0.503 and 0.866 and 0.538 and 0.807, respectively. On the contrary, the LtCNN model can maintain accuracy between 0.92 and 0.973 and is not sensitive to M. Each of the three models was averaged 0.696 ± 0.096, 0.674 ± 0.081, and 0.940 ± 0.018.

(3). From the point of view of the amount of spectral information, it is evident that using a sufficient number of bands can achieve the highest accuracy. For example, the LtCNN model obtained 0.976 with $M = 15$ selected by UBS in **Figure 12A** and 0.972 with $M = 18$ recommended by FNGBS in **Figure 12B**. The other two CNNs models also follow the same fashion. This proves that using hyperspectral imaging for species classification can obtain good results without too many bands.

## Comparison of Conventional Neural Network Models

The AlexNetr and GoogLeNet models produced lower, unstable, and downward accuracy in the plant classification is most likely due to two reasons. Firstly, they were designed and specialized for handling large databases with a large number of categories

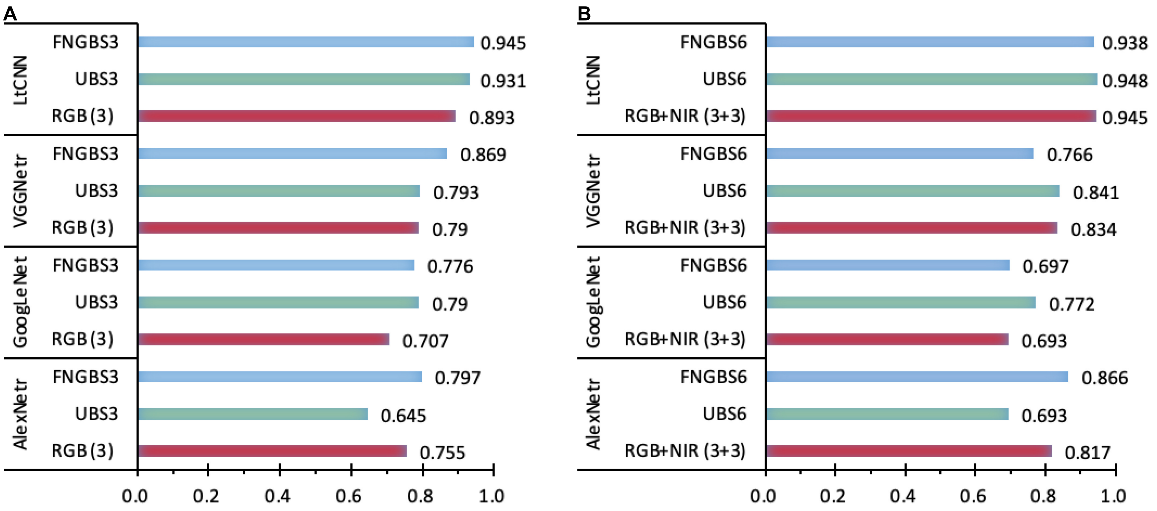
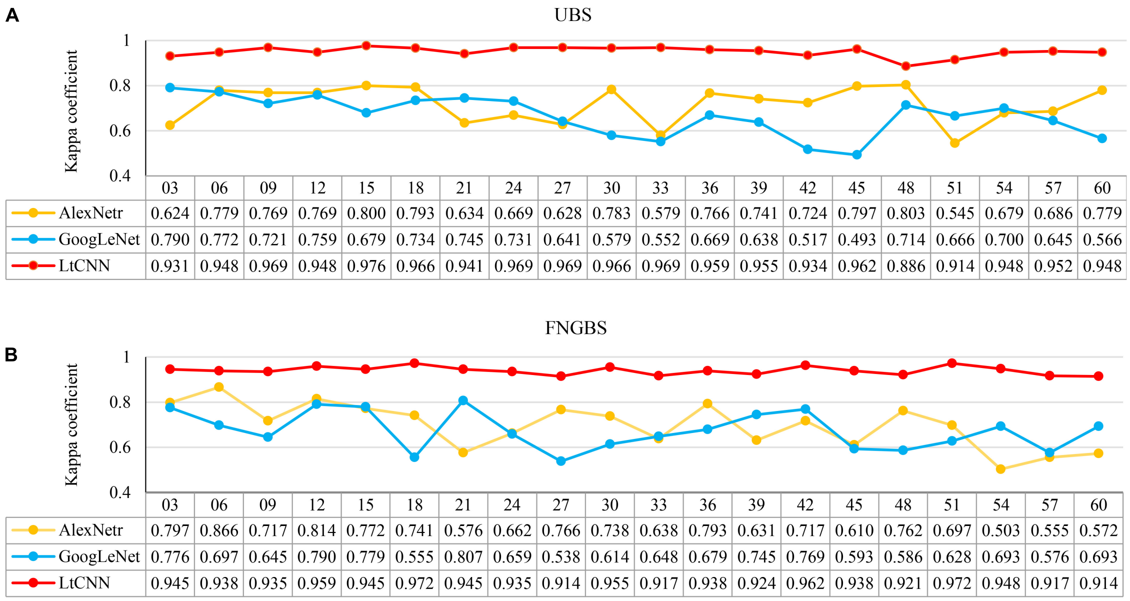


**FIGURE 11 |** A comparison of the sensitivity of spectral bands for species classification in the four CNN models. Charts **(A,B)** show the kappa coefficients with 3-band and 6-band classifications recommended by the band selection methods. A considerable difference of kappa values in any classifications with different datasets indicates higher sensitivity of the CNN models.



| | 03 | 06 | 09 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AlexNetr | 0.624 | 0.779 | 0.769 | 0.769 | 0.800 | 0.793 | 0.634 | 0.669 | 0.628 | 0.783 | 0.579 | 0.766 | 0.741 | 0.724 | 0.797 | 0.803 | 0.545 | 0.679 | 0.686 | 0.779 |
| GoogLeNet | 0.790 | 0.772 | 0.721 | 0.759 | 0.679 | 0.734 | 0.745 | 0.731 | 0.641 | 0.579 | 0.552 | 0.669 | 0.638 | 0.517 | 0.493 | 0.714 | 0.666 | 0.700 | 0.645 | 0.566 |
| LtCNN | 0.931 | 0.948 | 0.969 | 0.948 | 0.976 | 0.966 | 0.941 | 0.969 | 0.969 | 0.966 | 0.969 | 0.959 | 0.955 | 0.934 | 0.962 | 0.886 | 0.914 | 0.948 | 0.952 | 0.948 |

| | 03 | 06 | 09 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AlexNetr | 0.797 | 0.866 | 0.717 | 0.814 | 0.772 | 0.741 | 0.576 | 0.662 | 0.766 | 0.738 | 0.638 | 0.793 | 0.631 | 0.717 | 0.610 | 0.762 | 0.697 | 0.503 | 0.555 | 0.572 |
| GoogLeNet | 0.776 | 0.697 | 0.645 | 0.790 | 0.779 | 0.555 | 0.807 | 0.659 | 0.538 | 0.614 | 0.648 | 0.679 | 0.745 | 0.769 | 0.593 | 0.586 | 0.628 | 0.693 | 0.576 | 0.693 |
| LtCNN | 0.945 | 0.938 | 0.935 | 0.959 | 0.945 | 0.972 | 0.945 | 0.935 | 0.914 | 0.955 | 0.917 | 0.938 | 0.924 | 0.962 | 0.938 | 0.921 | 0.972 | 0.948 | 0.917 | 0.914 |

**FIGURE 12 |** The relationship between the number of bands (M) and produced accuracy when using a subset of hypercube images for classification. Line charts **(A,B)** present the kappa variation for the uniform band selection (UBS) and Fast Neighborhood Grouping Band Selection (FNGBS) method. The numbers below the M list the kappa values of the CNN models.

and training images. Since the tested plant database has only 30 classes and 1,500 images, the training data is relatively insufficient for them. Besides, the nature of the plant image is distinct from the objects' colors, shapes, and patterns for which the models were originated. This may explain why AlexNetr and GoogLeNet produce lower accuracy performance. Secondly, when M increases, the inter-band correlation of data increases. The input data with excessive redundant information may further interfere with the training process of the more extensive network under insufficient training data. This additionally imposes the difficulty of getting convergence in network training. This may explain why AlexNetr and GoogLeNet produce an unstable performance at different M values. In contrast, the proposed model LtCNN adopts a simplified architecture that is optimized for smaller datasets and significantly performs better than the other two in both accuracy and stability. This emphasizes the importance of designing a dedicated network for processing a particular dataset. And it suggests that as long as the network design is correct, it will not be too sensitive to data redundancy. Meanwhile, it can also be efficient with minimum bands to achieve satisfactory accuracy. Such a conclusion is significant for dealing with hyperspectral images.

### Limitation and Opportunity

The main strength of the proposed plant image classification method is that it uses the abundant spectral information provided by HSI, and uses the "deep features" learned by CNNs for plant species classification. However, this approach suffers from some drawbacks and limitations. Firstly, the cost of hyperspectral cameras is high so it is difficult for it to become widely adopted. Its long-shooting time also limits data acquisition and the possibilities for in-field investigations. Secondly, limited by the existing network architecture of CNN and memory size of GPU, it is hard to use high-resolution HSI images as the material to learn the more comprehensive features. Thirdly, our framework relies on band selection to reduce the data dimensionality. Ideally, we can feed all the bands into CNN and let it learn the discriminative bands automatically. However, limited by the network architecture, memory size, and the amount of training data, it is temporarily impossible to achieve. Finally, if we want to increase the species, except for adding image data to expand the database, it is also necessary to expand the network architecture and retrain the model. This is one of the crucial shortcomings of the current CNN approaches.

Even with these limitations, we are confident that this study will contribute to educational use as well as to the development of plant identification and forest remote sensing. One of the critical findings of this study is that applying only green-edge, red-edge, and near-infrared bands can substantially improve the species classification *via* the proposed model LtCNN. This finding provides an additional opportunity for sensor design specifically for plant applications and therefore benefits the imaging technology development for plant science research and education at a lower investment cost.

## CONCLUSION

From the point of view of individual tree recognition and mapping, this study applied hyperspectral imaging to build a plant image dataset *via* a VNIR imaging system with a spectral resolution of 2.8 nm and a radiometric resolution of 10 bits. The plant dataset contains 1500 images accounting for the crown and leaf features of 30 species. The plant images show dramatic reflectance values over the spectral range from the visible to the near-infrared region and therefore reveal the dilemma of pixel-based plant classification *via* remote sensing images. Although a pixel-based inspection of plant images reveals that diverse leaf colors increase the difficulty of plant classification using merely visible spectra, the near-infrared reflectance of colorful leaves of the same species remains very similar and behaves homogeneously and stably. In contrast to the variation of visible spectra, the species consistency of near-infrared spectral features provides an optimistic opportunity for plant classification.

According to the results, the complex deep learning architecture of AlexNetr, GoogLeNet, and VGGNetr models are not suitable for plant classification using a limited number of training samples and therefore failed to obtain satisfactory performance when integrating the features in 3-band RGB and 3-band NIR bands. Correspondingly, the best kappa accuracy for these models was 0.817, 0.693, and 0.834. The proposed lightweight conventional neural network, the LtCNN model, however, achieved an optimistic kappa accuracy of 0.945. Interestingly, this novel model has demonstrated its excellence in retrieving critical features from limited training samples through three bands suggested by the fast neighborhood grouping band selection method. The classification using the bands of green-edge (591.46 nm), red-edge (681.80 nm), and near-infrared (761.18 nm), the LtCNN model can achieve a kappa accuracy of 0.945, a value equal to the accuracy of a classification using 6 bands of RGB and NIR. Because the accuracy is very close to the maximum accuracy of 0.976, the best performance with 15 spectral bands of the hyperspectral images, the LtCNN model is concluded to be very efficient and reliable in classifying plant images. It is also concluded that a feature selection should be implemented before applying hyperspectral images to plant classification to reduce training cost and hardware loading significantly.

Many studies developed deep learning techniques for plant classification based on single-leaf images. This study is devoted to exploring an appropriate method for recognizing and classifying plant species according to live-crown and leaf features. Although the hyperspectral imaging technique can provide a hyperspectral dataset with critical spectral features for the application, some false-positive and false-negative errors still occurred in some species by the AlexNetr, GoogLeNet, VGGNetr, and LtCNN models simultaneously. These species are visual recognizably based on the features of leaf margin, surface leathery, and petiole. Developing a new network model with a 3D-CNN module should enhance feature learning in the spectral domain. The ability to retrieve tiny leaf features would also be an essential task for the future.

## DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be available at the following link: https://github.com/asufdhlkj456/Plant_Classification_with_HSI_and_DL.

## AUTHOR CONTRIBUTIONS

CL and K-HL: conceptualization, project administration, methodology, and writing – original draft. K-HL, M-HY, and S-TH: data curation and formal analysis. CL: writing – review and editing. All authors contributed to the article and approved the submitted version.

## FUNDING

This work was supported by the Ministry of Science and Technology, Taiwan, in Grant No.: MOST 109-2221-E-110-047-MY2 and 109-2221-E-415-003, and in part by the Council of Agriculture of the Executive Yuan, Taiwan, under Grant 110AS-8.3.2-ST-a7.

## REFERENCES

Abbas, S., Peng, Q., Wong, M. S., Li, Z., Wang, J., Ng, K. T. K., et al. (2021). Characterizing and classifying urban tree species using bi-monthly terrestrial hyperspectral images in Hong Kong. *ISPRS J. Photogramm. Remote Sens.* 177, 204–216. doi: 10.1016/j.isprsjprs.2021.05.003

Adão, T., Hruška, J., Pádua, L., Bessa, J., Peres, E., Morais, R., et al. (2017). Hyperspectral imaging: a review on UAV-based sensors, data processing and applications for agriculture and forestry. *Remote Sens.* 9:1110. doi: 10.3390/rs9111110

Baiano, A., Terracone, C., Peri, G., and Romaniello, R. (2012). Application of hyperspectral imaging for prediction of physico-chemical and sensory characteristics of table grapes. *Comput. Electron. Agric.* 87, 142–151. doi: 10.1016/j.compag.2012.06.002

Beghin, T., Cope, J. S., Remagnino, P., and Barman, S. (2010). "Shape and texture based plant leaf classification," in *Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems*, (Berlin: Springer), 345–353. doi: 10.1007/978-3-642-17691-3_32

Bengio, Y., Goodfellow, I., and Courville, A. (2017). *Deep Learning*, Vol. 1. Cambridge, MA: MIT press.

Briottet, X., Boucher, Y., Dimmeler, A., Malaplate, A., Cini, A., Diani, M., et al. (2006). "Military applications of hyperspectral imagery," in *Proceedings of the International Society for Optics and Photonics. Targets and Backgrounds XII: Characterization and Representation.*, Orlando, FL, Vol. 6239:62390B. doi: 10.1117/12.672030

Carranza-Rojas, J., Goeau, H., Bonnet, P., Mata-Montero, E., and Joly, A. (2017). Going deeper in the automated identification of Herbarium specimens. *BMC Evol. Biol.* 17:181. doi: 10.1186/s12862-017-1014-z

Chang, C. I. (2013). *Hyperspectral Data Processing: Algorithm Design and Analysis*. Hoboken, NJ: John Wiley & Sons.

Chen, S. Y., Lin, C., Li, G. J., Hsu, Y. C., and Liu, K. H. (2021). Hybrid deep learning models with sparse enhancement technique for detection of newly grown tree leaves. *Sensors* 21:2077. doi: 10.3390/s21062077

Chen, S. Y., Lin, C., Tai, C. H., and Chuang, S. J. (2018). Adaptive window-based constrained energy minimization for detection of newly grown tree leaves. *Remote Sens.* 10:96. doi: 10.3390/rs10010096

Fei, B. (2020). "Hyperspectral imaging in medical applications," in *Data Handling in Science and Technology*, Vol. 32, ed. J. M. Amigo (Amsterdam: Elsevier), 523–565. doi: 10.1016/B978-0-444-63977-6.00021-3

Feng, L., Wu, B., He, Y., and Zhang, C. (2021). Hyperspectral imaging combined with deep transfer learning for rice disease detection. *Front. Plant Sci.* 12:693521. doi: 10.3389/fpls.2021.693521

Gao, J., Nuyttens, D., Lootens, P., He, Y., and Pieters, J. G. (2018). Recognising weeds in a maize crop using a random forest machine-learning algorithm and near-infrared snapshot mosaic hyperspectral imagery. *Biosyst. Eng.* 170, 39–50. doi: 10.1016/j.biosystemseng.2018.03.006

Grinblat, G. L., Uzal, L. C., Larese, M. G., and Granitto, P. M. (2016). Deep learning for plant identification using vein morphological patterns. *Comput. Electron. Agric.* 127, 418–424. doi: 10.1016/j.compag.2016.07.003

Harrison, D., Rivard, B., and Sanchez-Azofeifa, A. (2018). Classification of tree species based on longwave hyperspectral data from leaves, a case study for a tropical dry forest. *Int. J. Appl. Earth Obs. Geoinf.* 66, 93–105. doi: 10.1016/j.jag.2017.11.009

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on CVPR*, (Piscataway, NJ: IEEE), 770–778. doi: 10.1109/CVPR.2016.90

Hopkins, W. G., and Hüner, N. P. A. (2004). *Introduction to Plat Physiology*, 3rd Edn. Hoboken, NJ: John Wiley & Sons.

Hossain, J., and Amin, M. A. (2010). "Leaf shape identification based plant biometrics," in *Proceedings of the 2010 International Conference on Computer and Information Technology. (ICCIT)*, (Piscataway, NJ: IEEE), 458–463. doi: 10.1109/ICCITECHN.2010.5723901

Hu, R., Jia, W., Ling, H., and Huang, D. (2012). Multiscale distance matrix for fast plant leaf recognition. *IEEE Trans. Image Process.* 21, 4667–4672. doi: 10.1109/TIP.2012.2207391

Hughes, G. F. (1968). On the mean accuracy of statistical pattern recognition. *IEEE Trans. Inform. Theory* 14, 55–63. doi: 10.1109/TIT.1968.1054102

Ioffe, S., and Szegedy, C. (2015). "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning, ICML*, (PMLR), 448–456.

Jaskierniak, D., Lucieer, A., Kuczera, G., Turner, D., Lane, P. N. J., Benyon, R. G., et al. (2021). Individual tree detection and crown delineation from Unmanned Aircraft System (UAS) LiDAR in structurally complex mixed species eucalypt forests. *ISPRS J. Photogramm. Remote Sens.* 171, 171–187. doi: 10.1016/j.isprsjprs.2020.10.016

Jung, A., Vohland, M., and Thiele-Bruhn, S. (2015). Use of a portable camera for proximal soil sensing with hyperspectral image data. *Remote Sens.* 7, 11434–11448. doi: 10.3390/rs70911434

Kadir, A., Nugroho, L. E., Susanto, A., and Santosa, P. I. (2013). Leaf classification using shape, color, and texture features. *arXiv* [preprint]. arXiv:1401.4447,

Khmag, A., Al-Haddad, S. R., and Kamarudin, N. (2017). "Recognition system for leaf images based on its leaf contour and centroid," in *Proceedings of the IEEE 15th Student Conference on Research and Development (SCOReD)*, (Piscataway, NJ: IEEE), 467–472. doi: 10.1109/SCORED.2017.8305438

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

Lee, S. H., Chan, C. S., Mayo, S. J., and Remagnino, P. (2017). How deep learning extracts and learns leaf features for plant classification. *Pattern Recognit.* 71, 1–13. doi: 10.1016/j.patcog.2017.05.015

Lee, S. H., Chan, C. S., Wilkin, P., and Remagnino, P. (2015). "Deep-plant: plant identification with convolutional neural networks," in *Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP)*, (Piscataway, NJ: IEEE), 452–456. doi: 10.1109/ICIP.2015.7350839

Li, F., Cao, H., Shang, X., Song, M., Yu, C., and Chang, C. I. (2019). "Uniform band interval divided band selection," in *Proceedings of the 2019 IEEE International Geoscience and Remote Sensing Symposium*, (Piscataway, NJ: IEEE), 3816–3819. doi: 10.1109/IGARSS.2019.8900363

Lin, C. (2017). "Applying a logistic-Gaussian complex signal model to restore surface hyperspectral reflectance of an old-growth tree species in cool temperate forest," in *Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium*, (Piscataway, NJ: IEEE), 3870–3873. doi: 10.1109/IGARSS.2017.8127847

Lin, C. (2018). A generalized Logistic-Gaussian-Complex Signal Model for the restoration of canopy SWIR hyperspectral reflectance. *Remote Sens.* 10:1062. doi: 10.3390/rs10071062

Lin, C. (2019). Improved derivation of forest stand canopy height structure using harmonized metrics of full-waveform data. *Remote Sens. Environ.* 235:111436. doi: 10.1016/j.rse.2019.111436

Lin, C. Y., and Lin, C. (2019). "Using ridge regression method to reduce estimation uncertainty in chlorophyll models based on worldview multispectral data," in *Proceedings of the 2019 IEEE International Geoscience and Remote Sensing Symposium*, (Piscataway, NJ: IEEE), 1777–1780. doi: 10.1109/IGARSS.2019.8900593

Lin, C. Y., Lin, C., and Chang, C. I. (2018). "A multilevel slicing based coding method for tree detection," in *Proceedings of the 2018 IEEE International Geoscience and Remote Sensing Symposium*, (Piscataway, NJ: IEEE), 7524–7527. doi: 10.1109/IGARSS.2018.8517654

Lin, C., Chen, S. Y., Chen, C. C., and Tai, C. H. (2018). Detecting newly grown tree leaves from unmanned-aerial-vehicle images using hyperspectral target detection techniques. *ISPRS J. Photogramm. Remote Sens.* 142, 174–189. doi: 10.1016/j.isprsjprs.2018.05.022

Lin, C., Wu, C. C., Tsogt, K., Ouyang, Y. C., and Chang, C. I. (2015a). Effects of atmospheric correction and pansharpening on LULC classification accuracy using WorldView-2 imagery. *Inf. Process. Agric.* 2, 25–36. doi: 10.1016/j.inpa.2015.01.003

Lin, C., Popescu, S. C., Thomson, G., Tsogt, K., and Chang, C. I. (2015b). Classification of tree species in overstorey canopy of subtropical forest using QuickBird images. *PLoS One* 10:e0125554. doi: 10.1371/journal.pone.0125554

Lin, C., Popescu, S. C., Huang, S. C., Chang, P. T., and Wen, H. L. (2015c). A novel reflectance-based model for evaluating chlorophyll concentrations of fresh and water-stressed leaves. *Biogeosciences* 12, 49–66. doi: 10.5194/bg-12-49-2015

Lin, C., Tsogt, K., and Chang, C. I. (2012). An empirical model-based method for signal restoration of SWIR in ASD field spectroradiometry. *Photogramm. Eng. Remote Sens.* 78, 119–127. doi: 10.14358/PERS.78.2.119

Lin, C., Tsogt, K., and Zandraabal, T. (2016). A decompositional stand structure analysis for exploring stand dynamics of multiple attributes of a mixed-species forest. *For. Ecol. Manag.* 378, 111–121. doi: 10.1016/j.foreco.2016.07.022

Liu, Z., Yan, J. Q., Zhang, D., and Li, Q. L. (2007). Automated tongue segmentation in hyperspectral images for medicine. *Appl. Opt.* 46, 8328–8334. doi: 10.1364/ao.46.008328

Ma, J., Pu, H., Sun, D.-W., Gao, W., Qu, J.-H., and Ma, K.-Y. (2015). Application of Vis–NIR hyperspectral imaging in classification between fresh and frozen-thawed pork Longissimus Dorsi muscles. *Int. J. Refrig.* 50, 10–18. doi: 10.1016/j.ijrefrig.2014.10.024

Marshall, S., Kelman, T., Qiao, T., Murray, P., and Zabalza, J. (2015). "Hyperspectral imaging for food applications," in *Proceedings of the 2015 23rd European Signal Processing Conference (EUSIPCO)*, (Piscataway, NJ: IEEE), 2854–2858. doi: 10.1109/EUSIPCO.2015.7362906

Mirzaei, M., Marofi, S., Abbasi, M., Solgi, E., Karimi, R., and Verrelst, J. (2019). Scenario-based discrimination of common grapevine varieties using in-field hyperspectral data in the western of Iran. *Int. J. Appl. Earth Obs. Geoinf.* 80, 26–37. doi: 10.1016/j.jag.2019.04.002

Nasiri, A., Taheri-Garavand, A., Fanourakis, D., Zhang, Y. D., and Nikoloudakis, N. (2021). Automated grapevine cultivar identification *via* leaf imaging and deep convolutional neural networks: a proof-of-concept study employing primary iranian varieties. *Plants* 10:1628. doi: 10.3390/plants10081628

Nicolaï, B. M., Lötze, E., Peirs, A., Scheerlinck, N., and Theron, K. I. (2006). Non-destructive measurement of bitter pit in apple fruit using NIR hyperspectral imaging. *Postharvest Biol. Technol.* 40, 1–6. doi: 10.1016/j.postharvbio.2005.12.006

Rapaport, T., Hochberg, U., Shoshany, M., Karnieli, A., and Rachmilevitch, S. (2015). Combining leaf physiology, hyperspectral imaging and partial least squares regression (PLS-R) for grapevine water status assessment. *ISPRS J. Photogramm. Remote Sens.* 109, 88–97. doi: 10.1016/j.isprsjprs.2015.09.003

Salman, A., Semwal, A., Bhatt, U., and Thakkar, V. M. (2017). "Leaf classification and identification using Canny Edge Detector and SVM classifier," in *Proceedings of the 2017 International Conference on Inventive Systems and Control (ICISC)*, (Piscataway, NJ: IEEE), 1–4. doi: 10.1109/ICISC.2017.8068597

Santos, F., Meneses, P., and Hostert, P. (2019). Monitoring long-term forest dynamics with scarce data: a multi-date classification implementation in the Ecuadorian Amazon. *Eur. J. Remote Sens.* 52(Suppl. 1), 62–78. doi: 10.1080/22797254.2018.1533793

Schmitter, P., Steinruecken, J., Roemer, C., Ballvora, A., Leon, J., Rascher, U., et al. (2017). Unsupervised domain adaptation for early detection of drought stress in hyperspectral images. *ISPRS J. Photogramm. Remote Sens.* 131, 65–76. doi: 10.1016/j.isprsjprs.2017.07.003

Simonyan, K., and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *arXiv* [preprint]. arXiv:1409.1556, doi: 10.3390/s21082852

Sinha, P., Robson, A., Schneider, D., Kilic, T., Mugera, H. K., Ilukor, J., et al. (2020). The potential of in-situ hyperspectral remote sensing for differentiating 12 banana genotypes grown in Uganda. *ISPRS J. Photogramm. Remote Sens.* 167, 85–103. doi: 10.1016/j.isprsjprs.2020.06.023

Sun, J., Zhou, X., Hu, Y., Wu, X., Zhang, X., and Wang, P. (2019). Visualizing distribution of moisture content in tea leaves using optimization algorithms and NIR hyperspectral imaging. *Comput. Electron. Agric.* 160, 153–159. doi: 10.1016/j.compag.2019.03.004

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). "Going deeper with convolutions," in *Proceedings of the IEEE conference on CVPR*, (Piscataway, NJ: IEEE), 1–9. doi: 10.1109/CVPR.2015.7298594

Teena, M. A., Manickavasagan, A., Ravikanth, L., and Jayas, D. S. (2014). Near infrared (NIR) hyperspectral imaging to classify fungal infected date fruits. *J. Stored Prod. Res.* 59, 306–313. doi: 10.1016/j.jspr.2014.09.005

Ubbens, J. R., and Stavness, I. (2017). Deep plant phenomics: a deep learning platform for complex plant phenotyping tasks. *Front. Plant Sci.* 8:1190. doi: 10.3389/fpls.2017.01190

Wang, Q., Li, Q., and Li, X. (2020). A fast neighborhood grouping method for hyperspectral band selection. *IEEE Trans. Geosci. Remote Sens.* 59, 5028–5039. doi: 10.1109/TGRS.2020.3011002

Wang, Y., Qin, Y., and Cui, J. (2021). Occlusion robust wheat ear counting algorithm based on deep learning. *Front. Plant Sci.* 12:645899. doi: 10.3389/fpls.2021.645899

Yang, G., Chen, G., Li, C., Fu, J., Guo, Y., and Liang, H. (2021). Convolutional rebalancing network for the classification of large imbalanced rice pest and disease datasets in the field. *Front. Plant Sci.* 12:671134. doi: 10.3389/fpls.2021.671134

Zhang, B., Wu, D., Zhang, L., Jiao, Q., and Li, Q. (2012). Application of hyperspectral remote sensing for environment monitoring in mining areas. *Environ. Earth Sci.* 65, 649–658. doi: 10.1007/s12665-011-1112-y

Zhang, H., Yanne, P., and Liang, S. (2012). "Plant species classification using leaf shape and texture," in *Proceedings of the 2012 International Conference on Industrial Control and Electronics Engineering*, (Piscataway, NJ: IEEE), 2025–2028. doi: 10.1109/ICICEE.2012.538

Zhang, X. D. (2020). *A Matrix Algebra Approach to Artificial Intelligence*, (Singapore: Springer), 223–440. doi: 10.1007/978-981-15-2770-8_6

Zhu, Y., Sun, W., Cao, X., Wang, C., Wu, D., Yang, Y., et al. (2019). TA-CNN: two-way attention models in deep convolutional neural network for plant recognition. *Neurocomputing* 365, 191–200. doi: 10.1016/j.neucom.2019.07.016

**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.

ORIGINAL RESEARCH
published: 14 April 2022
doi: 10.3389/fpls.2022.849986

# An Approach Using Emerging Optical Technologies and Artificial Intelligence Brings New Markers to Evaluate Peanut Seed Quality

*Gustavo Roberto Fonseca de Oliveira*[1]*, *Clíssia Barboza Mastrangelo*[2], *Welinton Yoshio Hirai*[3], *Thiago Barbosa Batista*[1], *Julia Marconato Sudki*[2], *Ana Carolina Picinini Petronilio*[1], *Carlos Alexandre Costa Crusciol*[1] and *Edvaldo Aparecido Amaral da Silva*[1]

[1] *Department of Crop Science, College of Agricultural Sciences, São Paulo State University, Botucatu, Brazil,* [2] *Laboratory of Radiobiology and Environment, Center for Nuclear Energy in Agriculture, University of São Paulo, Piracicaba, Brazil,* [3] *Department of Exacts Sciences, College of Agriculture "Luiz de Queiroz", University of São Paulo, Piracicaba, Brazil*

OPEN ACCESS

***Edited by:***
*Ian Stavness,*
*University of Saskatchewan, Canada*

***Reviewed by:***
*Roxana Vidican,*
*University of Agricultural Sciences and Veterinary Medicine of Cluj-Napoca, Romania*
*Dimitrios Fanourakis,*
*Technological Educational Institute of Crete, Greece*

***Correspondence:***
*Gustavo Roberto Fonseca de Oliveira*
*grfonseca.agro@gmail.com*

***Specialty section:***
*This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science*

***Received:*** *06 January 2022*
***Accepted:*** *21 March 2022*
***Published:*** *14 April 2022*

***Citation:***
*Fonseca de Oliveira GR, Mastrangelo CB, Hirai WY, Batista TB, Sudki JM, Petronilio ACP, Crusciol CAC and Amaral da Silva EA (2022) An Approach Using Emerging Optical Technologies and Artificial Intelligence Brings New Markers to Evaluate Peanut Seed Quality. Front. Plant Sci. 13:849986. doi: 10.3389/fpls.2022.849986*

Seeds of high physiological quality are defined by their superior germination capacity and uniform seedling establishment. Here, it was investigated whether multispectral images combined with machine learning models can efficiently categorize the quality of peanut seedlots. The seed quality from seven lots was assessed traditionally (seed weight, water content, germination, and vigor) and by multispectral images (area, length, width, brightness, chlorophyll fluorescence, anthocyanin, and reflectance: 365 to 970 nm). Seedlings from the seeds of each lot were evaluated for their photosynthetic capacity (fluorescence and chlorophyll index, $F_0$, $F_m$, and $F_v/F_m$) and stress indices (anthocyanin and NDVI). Artificial intelligence features (QDA method) applied to the data extracted from the seed images categorized lots with high and low quality. Higher levels of anthocyanin were found in the leaves of seedlings from low quality seeds. Therefore, this information is promising since the initial behavior of the seedlings reflected the quality of the seeds. The existence of new markers that effectively screen peanut seed quality was confirmed. The combination of physical properties (area, length, width, and coat brightness), pigments (chlorophyll fluorescence and anthocyanin), and light reflectance (660, 690, and 780 nm), is highly efficient to identify peanut seedlots with superior quality (98% accuracy).

**Keywords:** ***Arachis hypogaea*** **L., multispectral, images, machine-learning, fluorescence, reflectance, seed quality**

## INTRODUCTION

Peanut (*Arachis hypogaea* L.) is an oleaginous crop with considerable relevance in agriculture (Stalker and Wilson, 2016). Nations such as China, India, Nigeria and the United States produce most of the peanuts consumed in the world and contribute to global food security (Stalker and Wilson, 2016; USDA, 2020b). Peanut seeds are rich in oil and proteins (Arya et al., 2016), in addition to chemical properties that play an essential role in human health and in combating malnutrition (Temba et al., 2016; Bessada et al., 2019). Considering that the peanut production chain spans over six continents (USDA, 2020a), exploring factors that favor grain yield is part of a comprehensive global food security strategy. Taking this strategy into account, post-harvest technologies can increase seed quality which in turn would represent an increased grain yield.

Seeds of high physiological quality are the basic input for agriculture. They have high vigor which means better ability to promote rapid crop establishment under wide environmental conditions with a direct contribution to plant establishment and yield (Finch-Savage and Bassel, 2016; Ebone et al., 2020). Seeds with high quality have a prolonged lifespan, which ensures the retention of their vigor until sowing (Sano et al., 2016; Basso et al., 2018). Due to factors such as harvest immaturity (Okada et al., 2021), mechanical damage in processing (Barbosa et al., 2014), storage fungi (Ding et al., 2015) and inadequate transportation conditions (Groot et al., 2022), peanut seeds lose their quality in the production process. Few studies provide solutions to maximize peanut seed quality at post-harvest. For other species of agricultural interest, non-destructive technologies that generate data from multispectral images have been successfully used to assess seed quality (Elmasry et al., 2019a; Mortensen et al., 2021). Considering this possibility, the peanut seed may present unexplored spectral markers that allow the efficient evaluation of this quality.

The possibility of evaluating seed quality through multispectral images has been shown for legumes such as soybean (Baek et al., 2019), cowpea (Elmasry et al., 2019b) and six other species (Hu et al., 2020). In the case of crops such as tomatoes and carrots (Galletti et al., 2020), low seed reflectance at short wavelengths and reduced chlorophyll fluorescence were identified as markers of their quality. Reflectance makes it possible to investigate the spectral behavior of plant tissues through the pattern of reflected light at different wavelengths (Meireles et al., 2020). The light reflectance properties are also affected by the physiological state of the plants under unfavorable conditions, such as water stress (Caturegli et al., 2020). The application of reflectance in seed studies allows the evaluation of fungal incidence (França-Silva et al., 2020; Rego et al., 2020), color (Wang X. et al., 2021) and chemical composition variations (Barboza da Silva et al., 2021a; Bianchini et al., 2021). Under another principle, fluorescence is detected by the excitation of chlorophylls (a/b) in plant tissues in specific bands of the spectrum (Murchie and Lawson, 2013). The dynamics of chlorophyll fluorescence in the seed domain may be associated with its maturity (Galletti et al., 2020) or aging (Barboza da Silva et al., 2021b). In the seedling domain, on the other hand, chlorophyll fluorescence behavior has to do with photosynthetic functioning (Herritt et al., 2020; Oliveira et al., 2021). Thus, peanut seeds and seedlings may present characteristics that can be useful to the seed industry.

With the development of data processing capacity, machine-learning algorithms are promising tools to autonomously categorize seedlot quality. This approach has been explored to identify seed patterns associated with physical, physiological, and health characteristics with high accuracy (Medeiros et al., 2020b; Barboza da Silva et al., 2021b; Bianchini et al., 2021). This approach has also been employed for seed variety identification (Taheri-Garavand et al., 2021b). In different species, the combination of multispectral images and algorithms has been highly effective for seed evaluation (Elmasry et al., 2019b; Hu et al., 2020). The idea of this research is that peanut seeds have markers of their quality which are detectable by these technologies. Here, it was investigated whether multispectral images combined with machine learning models can efficiently categorize the quality of peanut seedlots.

## MATERIALS AND METHODS

### Plant Material

Seven lots of peanut (*Arachis hypogaea* L.; cv. IAC OL3; Virginia group) seeds produced in 2019/2020 in the western region of the State of São Paulo, Brazil by COPERCANA[1] and COPLANA[2] seed companies, were used for the research. The fruits were harvested and then dried in the shade. After this, the seeds were manually extracted. The seeds obtained from each lot were homogenized by manually removing broken or malformed seeds (sectioned or damaged cotyledons) and seeds without the tegument. The seedlots were stored in a dry chamber at 12°C/55% relative humidity (RH) until the beginning of the experiments, after approximately 90 days of storage.

### Trial Design

Initially, conventional tests were conducted to assess the quality of seedlots through water content, fresh weight, germination, and vigor. Then, from a study using multispectral images, it was found that certain spectral characteristics of the seeds correlated strongly with their quality. From the characteristics found through these images, the quality of seedlots was classified (principal component analysis) into groups of low vigor (lots 1, 2, and 3) and high vigor (lots 4, 5, 6, and 7). With this qualitative information (two groups), machine learning models (quadratic discriminant analysis method) were used to autonomously recognize these behaviors (high and low vigor). Finally, seedlings from the seeds in each lot were evaluated for their photosynthetic capacity and stress indicators using multispectral images. In addition, two other studies were conducted with seeds exposed to stress conditions (high temperature and high RH). Seedlings from these seeds were also evaluated for their photosynthetic capacity and stress indicators. Details regarding the variables measured, method, and number of seeds used in each research test are available for consultation in the supplementary files (**Supplementary Tables 1, 2**).

### Characterization of Physiological Quality of Seeds

The water content of the seeds was determined by the oven method at 105 ± 3°C for 24 h (ISTA, 2020), using four replicates of 10 seeds. For the determination of seed fresh weight, four replicates of 100 seeds were weighed on an analytical scale with a precision of 0.001 g. Subsequently, a part of the seeds of each lot (about 500 g) was treated with fungicides (Carbendazim and Thiram; 2 mL $kg^{-1}$). This procedure aimed to inhibit the

[1] https://copercana.com.br

[2] http://www.coplana.com

occurrence of fungi during the execution of the tests and to reduce any interference of pathogenic microorganisms in the seed quality results. The remaining seeds were not submitted to the treatment with fungicides. It was considered that any product applied to the surface of the seeds could change their spectral characteristics and compromise the quality of the data generated.

Germination was evaluated on rolled paper towel and sand substrates. Four replicates of 25 seeds were placed between the paper towels and moistened with deionized water at 2.5 times the mass of the dry paper. The rolled paper towels were kept at a constant 25°C in the dark. For the sand substrate, a sterile medium textured sand in plastic boxes was used (34.0 × 21.7 × 7.0 cm), and the substrate was wet to 60% of its holding capacity. Then, four replicates of 25 seeds from each lot were sown at a depth of 5.0 cm. The boxes with the seeds remained in a growth chamber at 25°C and 80% RH. The percentage of normal seedlings (with all their essential structures, such as aerial part, hypocotyl and well-developed radicle, complete, proportional and healthy) produced in the germination test using paper towels and sand was obtained on the 10th day (final score) after initial sowing (ISTA, 2020).

Vigor was initially determined by the time required for 50% germination (t50). Four replicates of 25 seeds from each lot were used according to the conditions described for the germination experiment between rolled paper towels. Twenty-four hours after the beginning of the experiment germination was assessed, with radicles with ≥2 mm in length used as the criteria. The measurements were performed every 4 h. The calculation of t50 was performed using the Germinator software (Joosen et al., 2010).

The seeds of each lot were also evaluated for seedling emergence capacity. Four replicates of 25 seeds each were used, with sand as substrate for the test. The seeds were sown at a depth of 5.0 cm in a suspended bed under uncontrolled environmental conditions. The substrate was wetted after sowing and throughout the experiment. Emerged seedlings (cotyledons and epicotyl apparent on the substrate surface) were counted daily and at the same time until stabilization of the number of emerged seedlings (Krzyzanowski et al., 2020). Seed vigor was expressed as percentage of emerged seedlings.

Another vigor test was carried out based on the seedling performance. For that, four replicates of 10 seeds were used, sown equidistantly from each other on the upper third of the surface of paper towels, using the same conditions described for germination between rolled paper towels. After 5 days, shoot and radicle length of normal seedlings was measured. Afterward, the aerial part and the radicles were segmented and placed in an oven at 60°C for 72 h to assess the dry weight (Krzyzanowski et al., 2020).

## Multispectral Image Acquisition of Seeds

Multispectral images were acquired from a total of 170 seeds for each lot. The seeds were placed in 9.0 cm glass Petri dishes. Multispectral images were captured at 19 wavelengths – 365 (UV), 405 (violet), 430 (indigo), 450 (blue), 470 (blue), 490 (cyan), 515 (green), 540 (green), 570 (yellow), 590 (amber), 630 (red), 645 (red), 660 (red), 690 (dark red), 780 (dark red), 850, 880, 940, and 970 nm (the last four wavelengths in the near infrared region), using a VideometerLab4$^{TM}$ instrument (Videometer A/S, Herlev, Denmark; software version 3.14.9) as described by Galletti et al. (2020). This system can capture and combine high-resolution multispectral images (2192 × 2192 pixels). Before acquiring the seed images, the light configuration was adjusted to optimize the intensity at each bandwidth, resulting in a better signal-to-noise ratio so that the captured images could be directly comparable. The light configuration was adjusted using a representative sample, and then the strobe time of each type of illumination was optimized in relation to this area. Seeds were segmented based on thresholding and the following variables were extracted from individual seeds: area, length, width and brightness measured by CIELab $L^*$ (Oliveira et al., 2021), fluorescence of chlorophyll *a* (630/700 nm excitation/emission) and chlorophyll *b* (405/600 nm excitation/emission). In addition, the reflectance values of the seeds of each lot from 365 to 970 nm were collected, and the chlorophyll *a*/*b* ratio was calculated. The seed images were transformed by a normalized canonical discriminant analysis (nCDA) algorithm, in which pixel values are calculated based on 10% trimmed mean to provide a more realistic image.

Multispectral images were also captured using a SeedReporter$^{TM}$ instrument (PhenoVation B.V., Wageningen, Netherlands) to calculate the anthocyanin index of the seeds. Prior to image acquisition, light intensity was adjusted to avoid overload. Reflectance images were acquired in a few seconds, generating multispectral images with a spatial dimension of 2448 × 2448 pixels (3.69 μm/pixel). A broad-band blank white light (3000 K) in a range of 450 to 780 nm was used to illuminate the seeds, and reflectance data was collected using three optical filters at 540, 710, and 770 nm (Gitelson et al., 2009). The anthocyanin index was calculated by SeedReporter$^{TM}$ software version 5.5.1. using the equation presented by Oliveira et al. (2021).

## Machine Learning – Quadratic Discriminant Analysis

The Quadratic Discriminant Analysis (QDA) method was used for the classification of high and low vigor seedlots. The choice of this method was based on the following aspects: (i) QDA is one of the most widely used methodologies for cases where the response variable is qualitative (Hastie et al., 2009; James et al., 2021) and (ii) it allows for effective analyses with data that do not have a normal distribution and have inhomogeneous variance and a covariance matrix structure (Clarke et al., 1979). Classification modeling was used based on the dataset extracted from the multispectral images of the seeds. Four QDA-based method machine learning models were generated for different datasets. In this way, the capacity of these models to infer the accuracy (sensitivity and specificity) of the spectral variables regarding the vigor of the seedlots ($n$ = 1190) was tested. The learning models obtained through the QDA method were adjusted and tested by cross-validation using data related to the physical optical descriptors of the seedlots (first model: area, length, width and CIELab $L^*$), pigments (second model:

chlorophyll fluorescence and anthocyanins), reflectance (third model - bands that best discriminated seedlots: 660, 690, 780, 850, and 970 nm) and the sum of all these variables (fourth model: physical optical descriptors, pigments and reflectance). In all, four prediction models were built, and the data were divided into 70% for training and 30% for testing. The details of the mathematical procedures used are described in a supplementary file (**Supplementary Methodology 1**).

## Anthocyanin and Chlorophyll in Seedlings

Four replicates of 10 seeds per lot were sown in 500 mL polystyrene pots (8 pots per lot), filled with a mixture of pine bark, peat moss and vermiculite. Each pot contained 5 seeds. The seedlings were cultivated under controlled conditions of temperature (25°C), RH (50–70%) and white light (900 mm, LED lamps, 13 W) (Condado de Ilum., São Paulo, Brazil) with a photoperiod of 16/8 h light/dark. The pots were irrigated as needed. When the seedlings were well established, 7 days after sowing, the number of seedlings per pot was reduced to two, reducing overlap. Measurements were taken considering the canopy formed by the two seedlings in each pot, which totaled eight seedlings canopies per lot, taken 14 days after sowing.

The chlorophyll *a* index (Chl *a* index), anthocyanin index and the normalized difference vegetation index (NDVI) were calculated by a SeedReporter$^{TM}$ instrument (PhenoVation B.V., Wageningen, Netherlands). The Chl *a* index was estimated based on the reflectance at 710 and 770 nm (Gitelson et al., 2003), and the anthocyanin index from the reflectance at 540, 710, and 770 nm (Gitelson et al., 2009). The NDVI was calculated based on reflectance at 640 and 770 nm (Yengoh et al., 2015).

The initial fluorescence ($F_0$), maximum fluorescence ($F_m$), average chlorophyll *a* fluorescence and maximum quantum efficiency of photosystem II ($F_v/F_m$) were measured using a SeedReporter$^{TM}$ instrument, which is also integrated with high intensity amber LEDs (620 nm peak), with a saturating light intensity of 6.320 $\mu$mol m$^{-2}$ s$^{-1}$, while an interference filter (730 nm) transmitted the fluorescence signals from the leaves to a CCD chip. All parameters were calculated by SeedReporter$^{TM}$ software version 5.5.1.

## Further Experiments

This additional study was conducted with 300 seeds from one of the lots characterized as high quality (IAC OL3, lot 7) exposed to an artificial aging procedure (ISTA, 2020). The seeds were placed on a wire mesh suspended inside a covered plastic box containing 40 mL of distilled water at the bottom, providing a RH of 100%. Subsequently, the boxes were added to a B.O.D chamber set at 42°C. The seeds remained in these stress conditions for 24 and 48 h. A control group consisted of seeds not artificially aged. The objective was to induce seed deterioration by high temperature and high RH. Subsequently, the responses of the applied stress on pigment dynamics and seed brightness were investigated through multispectral images. To this end, stress-exposed and control seeds were subjected to evaluation of fluorescence chlorophyll *a*, fluorescence chlorophyll *b*, brightness (CIELAB $L^*$) and anthocyanin index as described previously. These variables were also measured in seeds of another cultivar (IAC 503) exposed to the same stress conditions. Seeds belonging to the research lot and exposed to stress (IAC OL3; lot 7) were also used for seedling production following the same conditions previously described. At 14 days after sowing, chlorophyll *a* and anthocyanin indices, NDVI, $F_0$, $F_m$, chlorophyll *a* fluorescence and $F_v/F_m$ were calculated for each seedling using SeedReporter$^{TM}$ software.

## Statistical Design

The data obtained in the conventional tests performed for the seven seedlots were submitted to analysis of variance – ANOVA (*F* test; $p \leq 0.05$) with four repetitions ($n = 28$). Comparison of means was performed by Tukey test ($p \leq 0.05$). The data obtained from the multispectral images of 170 seeds of each lot were submitted to ANOVA and the Tukey test (each seed as a repetition; $n = 170$). The data obtained from multispectral images of the seedlings from the seeds of each lot were submitted to ANOVA and Tukey test with four replications ($n = 28$). The same analyses procedures were adopted for the data obtained in the further experiments. From the reflectance data (from 365 to 970 nm) observed for the seeds of each lot ($n = 170$), an interactive process analysis (*for loop*) was carried out in order to select the 20 combinations of 5 bands that best discriminated seedlots (660, 690, 780, 850, and 970 nm). The details of the computational procedures used are described in a supplementary file (**Supplementary Methodology 2**).

Principal component analysis (PCA) and correlation were performed with the data observed in conventional tests and multispectral images of the seeds. The Permanova test and the Bray-Curtis similarity index (Canoco 5 software) were used to identify the significance of the behavior observed in PCA between seedlots (*F* Test; $p \leq 0.05$). Correlation analysis was calculated using the Spearman method, due to the non-normality of the variables. Additionally, when the variables were a different number of repetitions, the average was calculated, so a balanced observation could be made. The "ExpDes.pt" package of the R software was used to perform the analysis of variance (completely randomized design) and the Tukey test (R Core Team, 2021). The QDA analysis was performed with the MASS library (Venables and Ripley, 2002) with the MASS:qda() function, and the results of the confusion matrix and accuracy measurement were collected by the library and caret: confusionMatrix() (Kuhn, 2017).

# RESULTS

## Physiological Quality and Physical Properties of Seeds

The germination test using paper substrate clearly separated the seedlots into two groups, i.e., lots 1, 2, and 3 (lower quality) *vs.* lots 4, 5, 6, and 7 (higher quality) (**Figure 1A**). In contrast, the germination test using sand as substrate did not show a clear quality difference among seedlots (**Figure 1B**). The average time for 50% germination (t50) classified lot 2 as lower vigor

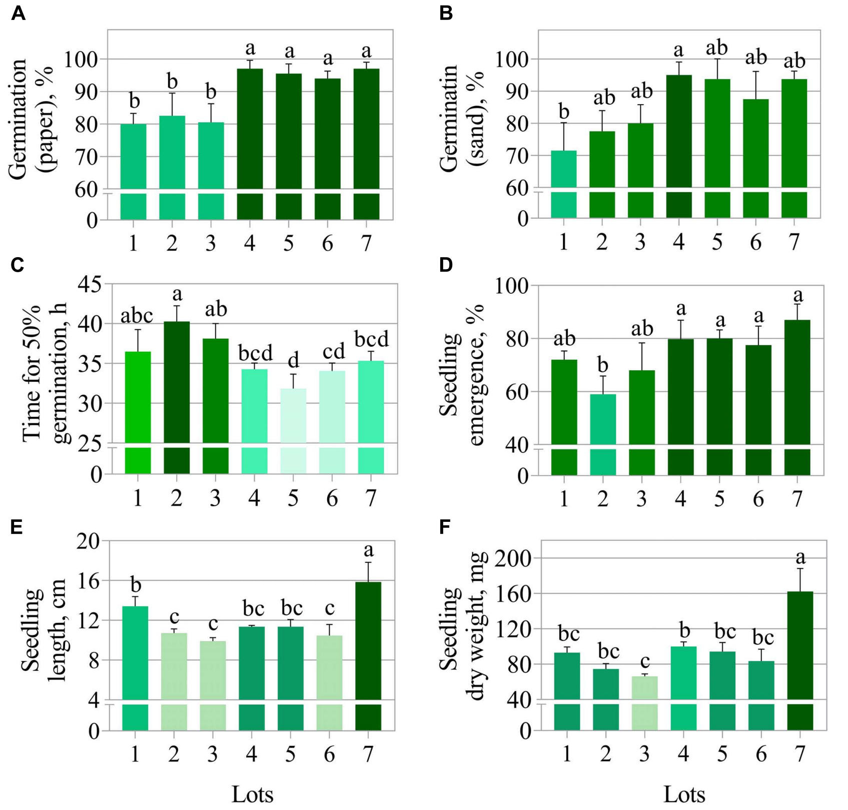


**FIGURE 1 |** Physiological quality of seven seedlots of peanut (*Arachis hypogaea* L.; cv. IAC OL3) based on germination on paper **(A)**, germination on sand **(B)**, time for 50% germination **(C)**, seedling emergence **(D)**, seedling length **(E)**, and seedling dry weight **(F)**. Means (± standard deviation) with different letters indicate a significant difference ($p \leq 0.05$).

(higher values of t50) (**Figure 1C**). In addition, seeds from lot 2 also presented the worst performance for seedling emergence and seedling length (**Figures 1D,E**). Nevertheless, lots 2 and 3 generated seedlings with very similar length as lot 6 (**Figure 1E**). The seedling length and dry weight measurements revealed lot 7 as having the best vigor (**Figures 1E,F**). Except for germination on paper (**Figure 1A**), conventional tests detected punctual and unclear differences in the quality of seedlots. Regarding the physical properties, seeds from lots 4, 5, 6, and 7 had higher fresh weight (**Figure 2A**) and this was associated with lower water content (≅ 7%) (**Figure 2B**). These seedlots in addition to the high quality indicated by the germination test (**Figure 1A**) also had superior area, length, width and brightness (CIELab $L^*$) (**Figures 2C–F**).

## Seed Pigments

The seedlots that exhibited the best performance in the germination test, i.e., lots 4, 5, 6, and 7 (**Figure 1A**) showed higher chlorophyll *a* and *b* fluorescence (**Figures 3A,B**), but a lower chlorophyll *a*/*b* ratio (**Figure 3C**) and anthocyanin index (**Figure 3D**). Therefore, the results indicated that there is a stronger difference in chlorophyll *b* between the two groups (lots 1, 2, 3 *vs.* lots 4, 5, 6, and 7), and this was also shown by comparing the chlorophyll *a* and *b* images (**Figures 4A,B**), in parallel with lower anthocyanins in the group with greater germination performance (**Figure 4C**).

Curiously, when lot 7 was artificially aged, chlorophyll *a* and *b* fluorescence was rapidly reduced (**Figures 5A,B**, **6A,B**). In addition, there was a reduction in the seed coat brightness (CIELab $L^*$) (**Figure 5C**) and an increase in the anthocyanin index (**Figures 5D**, **6C**). To verify whether this response can also occur in seeds of other genotypes, seeds obtained from IAC 503 cultivar were also artificially aged (**Supplementary Figure 1**). Likewise, there were lower chlorophyll *a* and *b* fluorescence signals, reduced seed coat brightness and increased anthocyanin index in aged seeds (**Supplementary Figure 1**).

## Seed Reflectance

The seeds with superior quality (lots 4, 5, 6, and 7) had the highest spectral signature in the visible region of the spectrum (405 to 540 nm; 630 to 780 nm) (**Figure 7A**). Seed reflectance was similar at longer wavelengths (850 and 970 nm), with the exception of lot 4 (**Figure 7A**). The combination of 660, 690, 780, 850, and 950 nm wavelengths showed superior accuracy to discriminate the spectral patterns of the seedlots (**Figure 7B**). When evaluating the bands individually, the results showed that the wavelengths of 660, 690, and 780 nm allow better separation of groups with lower and higher quality (lots 1, 2, and 3 *vs.* lots 4, 5, 6, and 7) (**Figure 8**).

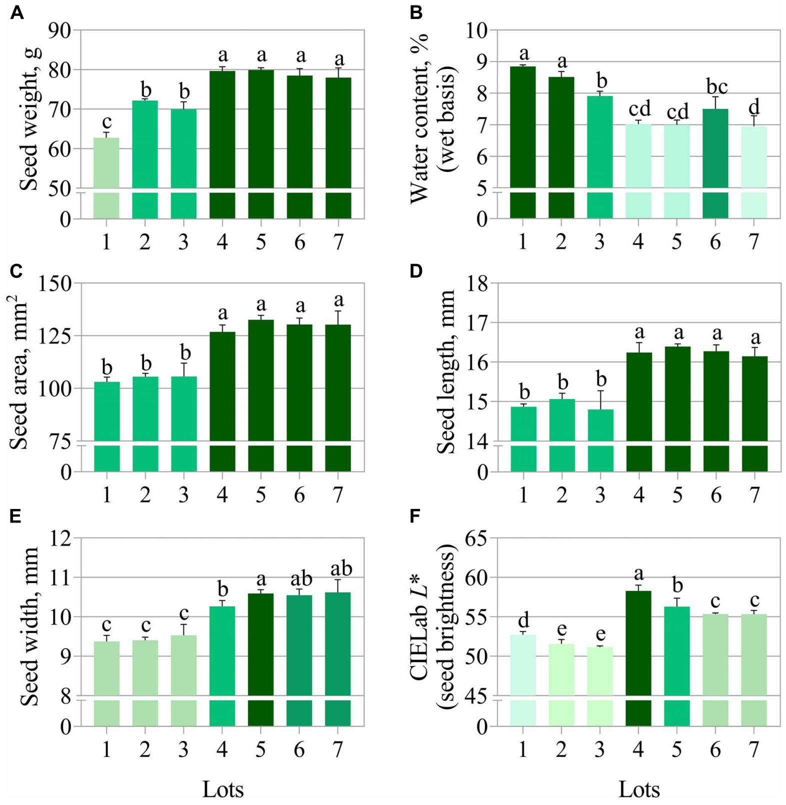


**FIGURE 2 |** Physical properties of seven seedlots of peanut (*Arachis hypogaea* L.; cv. IAC OL3) based on fresh weight **(A)**, water content **(B)**, area **(C)**, length **(D)**, width **(E)**, and CIELab $L^*$ **(F)**. The CIELab$L^*$ represents the perceived brightness ranging from 0.0 (black) to 100.0 (white). Means (± standard deviation) with different letters indicate a significant difference ($p \leq 0.05$).

## Correlation Between Physical, Physiological, Pigment and Reflectance Descriptors

The correlation coefficients showed a relationship between physical descriptors and germination (paper): 0.78 (seed weight), −0.77 (water content) 0.75 (area), 0.76 (length), 0.75 (width) and 0.76 (CIELab $L^*$ – seed brightness). Seed brightness was the only physical descriptor with a correlation coefficient greater than 0.7 vs. t50 (vigor test). Between seed pigments and germination (paper) the correlations were: 0.73 (chlorophyll *b*), −0.85 (chlorophyll *a*/chlorophyll *b*), and −0.75 (anthocyanin index). The germination (paper) vs. reflectance bands obtained the following correlations: 0.77 (660 nm), 0.78 (690 nm), and 0.76 (780 nm). The correlation coefficients obtained for seedling emergence vs. 690 and 780 nm were 0.71 and 0.72, respectively. The reflectance bands showed the following correlations with seed brightness: 0.98 (660 nm), 0.95 (690 nm), and 0.9 (780 nm). The correlation between seed brightness and the seed pigments were: 0.81 (Chl *a*), 0.95 (Cha *b*), −0.83 (Chl *a*/Chl*b*), and −0.78 (anthocyanin index; **Figure 9A**).

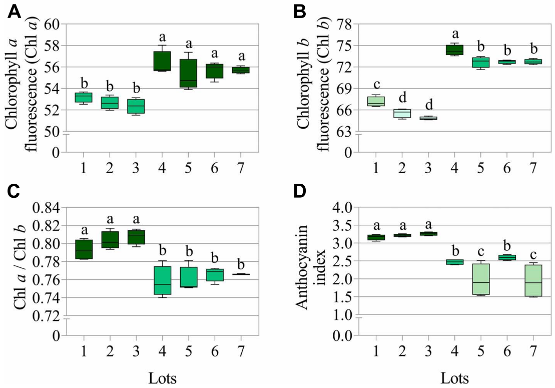


**FIGURE 3 |** Average chlorophyll *a* fluorescence (Chl *a*) at 630/700 nm excitation/emission combination **(A)**, chlorophyll *b* fluorescence (Chl *b*) at 405/600 nm excitation/emission combination **(B)**, chlorophyll a/b ratio (Chl *a*/Chl *b*) **(C)**, and anthocyanin index **(D)** measured in seven seedlots of peanut (*Arachis hypogaea* L.; cv. IAC OL3). Means (± standard deviation) with different letters indicate a significant difference ($p \leq 0.05$) ($n = 170$).

The PCA allowed the correlation of the groups of seeds with high and low vigor (lots 1, 2, 3 vs. lots 4, 5, 6, and 7), explaining 71.6% of the significant variation ($PCA_1$) found (PERMANOVA; $p < 0.001$). Most of the seeds with lower vigor were negatively correlated with the anthocyanin index, water content, chlorophyll *a*/*b* ratio, time for 50% germination and reflectance at 970 nm. Meanwhile, the group of seeds with higher vigor exhibited

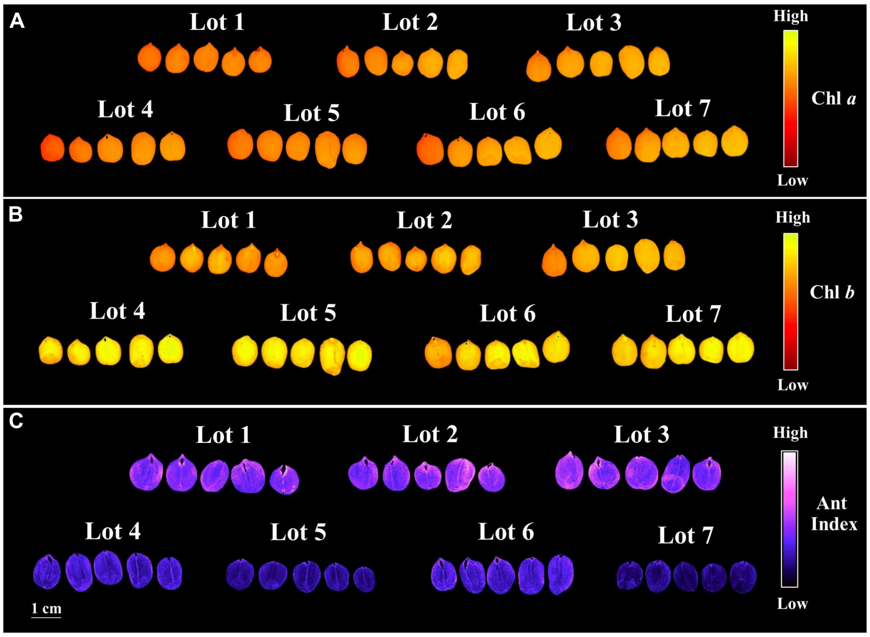


**FIGURE 4 |** Chlorophyll a fluorescence (Chl *a*) at excitation/emission combination of 630/700 nm **(A)**, chlorophyll *b* fluorescence (Chl *b*) at excitation/emission combination of 405/600 nm **(B)**, and anthocyanin index (Ant Index) **(C)** of seven seedlots of peanut seeds (*Arachis hypogaea* L.; cv. IAC OL3). Each pixel in the images is represented by a unique value that corresponds to chlorophyll *a* and *b* fluorescence intensity or anthocyanin level.

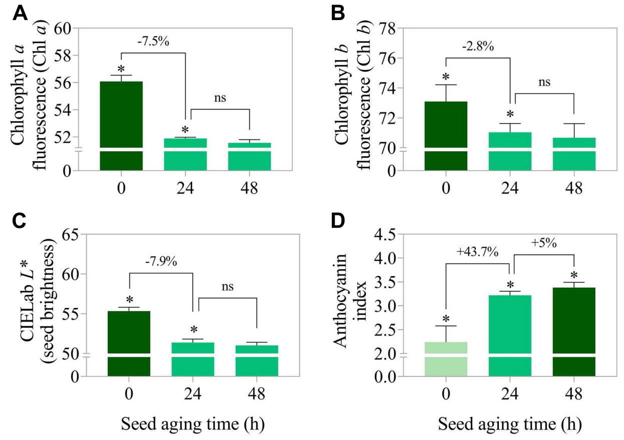


**FIGURE 5 |** Chlorophyll *a* fluorescence at excitation/emission combination of 630/700 nm **(A)**, chlorophyll *b* fluorescence at excitation/emission combination of 405/600 nm **(B)**, CIELab *L** representing the perceived brightness ranging from 0.0 (black) to 100.0 (white) **(C)**, and anthocyanin index **(D)** in peanut seeds (*Arachis hypogaea* L.; cv. IAC OL3) from lot 7 artificially aged for 0, 24, and 48 h. Means (± standard deviation); significant (*); not significant (ns); (p > 0.05) (*n* = 100).

positive correlation with all other variables as seed weight, area, length, width, brightness (CIELab *L**), chlorophyll *a*, chlorophyll *b*, and seed reflectance (660, 690, and 780 nm). These variables were expressed to a higher degree (vector modulus) in high vigor seedlots (lots 4, 5, 6, and 7) jointly with germination in paper, germination in sand, and seedling emergence (**Figure 9B**).

## Seed Quality Classification Based on Machine Learning Models Using Physical Properties, Pigments and Reflectance Descriptors

From the seed groups (**Figure 9B**) divided into high vigor (lots 1, 2, and 3) and low vigor (lots 4, 5, 6, and 7) quadratic discriminant analysis (QDA) models were constructed. Based on the data set ($n = 1190$), the first model generated using the physical optical descriptors (area, length, width, and CIELab *L**) was able to predict the behavior of the two seed groups (high and low vigor) with 89% accuracy. For the second model, using seed pigments (chlorophyll *a*, chlorophyll *b*, and the anthocyanin index), the accuracy was 94%. Using the most significant wavelengths of reflectance (660, 690, 780, 850, and 970 nm) the accuracy was 97%. From the union of the physical optical descriptors, pigments and reflectance of the seeds in a single model, the accuracy was 98% (**Table 1**).

## Pigments and Photosynthetic Efficiency of Seedlings

The low vigor seedlots (i.e., Lot 1) generated seedlings with higher values for the variables chlorophyll *a* index, initial fluorescence and maximum fluorescence, and $F_v/F_m$ ratio (**Figures 10A–D**). Seedlings from these seeds also had a high anthocyanin index (**Figure 10E**). Chlorophyll *a* fluorescence was similar among most of the seedlings from the analyzed seedlots (**Figure 10F**). Differences in the anthocyanin index and the chlorophyll *a* index of the seedlings were most evident between the high and low vigor seedlots 1 and 7 (**Figures 11A,B**). The $F_v/F_m$ ratio was very

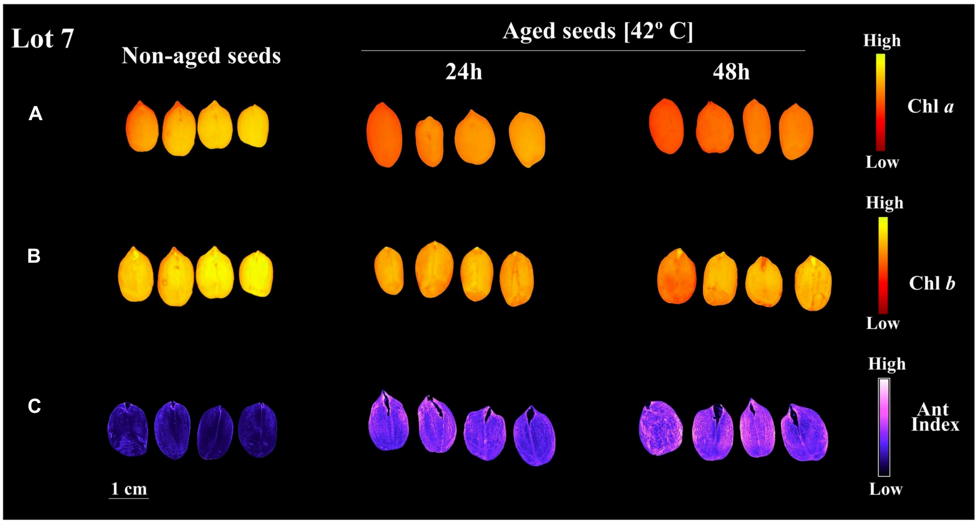


**FIGURE 6 |** Chlorophyll *a* fluorescence (Chl *a*) at 630/700 nm excitation/emission combination **(A)**, chlorophyll *b* fluorescence (Chl *b*) at 405/600 nm excitation/emission combination **(B)**, and anthocyanin index (Ant Index) **(C)** in peanut seeds (*Arachis hypogaea* L.; cv. IAC OL3) from lot 7 for classes on non-aged seeds and seeds aged for 24 h and 48 h. Each pixel in the images is represented by a unique value that corresponds to chlorophyll *a* and *b* fluorescence intensity or anthocyanin level.

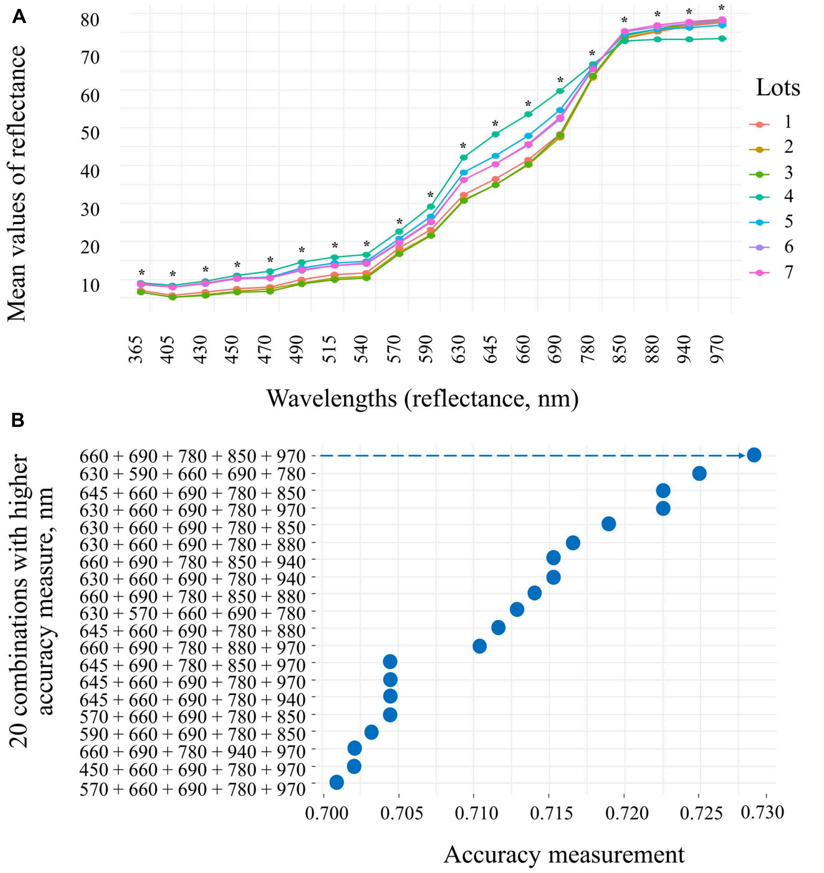


**FIGURE 7 |** Reflectance spectral signature at 19 wavelengths (365 to 970 nm) of seven peanut seedlots (*Arachis hypogaea* L.; cv. IAC OL3) **(A)** and 20 combinations of wavelengths with distribution of accuracy determined by interactive process analysis **(B)**. The arrow indicates the combination of bands (660, 690, 780, 850, and 970) that showed the highest accuracy (0.730) for the subsequent analyses. *significant at the 0.05 probability levels ($n$ = 170).

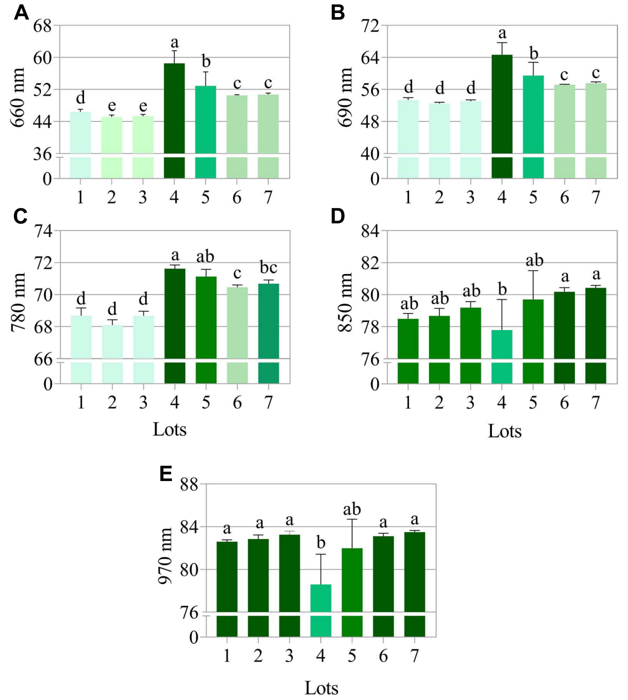


**FIGURE 8 |** Reflectance mean of seven peanut seedlots (*Arachis hypogaea* L.; cv. IAC OL3) at **(A)** 660, **(B)** 690, **(C)** 780, **(D)** 850, and **(E)** 970 nm (previously shown as the best wavelengths to discriminate seeds as high and low vigor) ($n$ = 170). Means (± standard deviation) with different letters indicate a significant difference ($p \leq 0.05$).

precise to show differences in photosynthetic activity by images of the evaluated seedlings (**Figure 11C**).

Seedlings from seedlot 7 that were submitted to artificial aging showed improvement in the main photosynthetic parameters. The chlorophyll *a* index, initial fluorescence and maximum fluorescence increased by 32, 4.8 and 5.6% after 24 h of stress, respectively (**Figures 12A–C**). The time of seed exposure to aging did not affect the quantum yield of the photosystem II system ($F_v/F_m$) of the seedlings (**Figure 12D**). However, it caused an increase in the anthocyanin index and the normalized vegetation index (**Figures 12E,F**). This behavior was clearly reflected in the images (**Figures 13A–C**).

# DISCUSSION

This study contains contributions that highlight the accuracy of technologies based on multispectral images and machine learning to identify peanut seeds with superior quality. New evidence reinforces the possibility of autonomous detection of physical parameters, chlorophyll fluorescence and light reflectance in peanut seeds to assess their physiological quality. Here, these and other original data address the use of post-harvest technologies to advance the peanut seed production sector in the world.

## Seed Quality

The seed industry performs the physiological quality control of lots every cultivation season. Among the conventional tests capable of assessing seed quality, germination performed within 10 days provides sufficiently satisfactory results (**Figure 1A**). In the case of t50 (vigor test), the distinction of seedlots with high and low vigor is also possible (**Figure 1B**). However, these are tests that require a lot of time and effort to be performed on a large scale. This makes the process of seed quality control inefficient. Regarding water content, the low moisture observed in certain lots (**Figure 2A**) is described as a state that slows the natural deterioration processes (Buitink and Leprince, 2004) in addition to prolonging the conservation of seeds in storage (Leprince et al., 2017). Therefore, evaluation is essential for obtaining seedlots with high quality. Still, it is a destructive methodology and, as with the other conventional tests, depends on human analytical ability. With this in mind, based on studies with seeds of other species (Mortensen et al., 2021) the potential of multispectral images technologies was investigated. New markers capable of efficiently determining peanut seed quality were found.

A first component of this approach comprises physical properties (shape and brightness). Characteristics such as area, length and width have been positively associated with seed vigor and adequate seedling establishment. In fact, peanut seeds with high quality showed additional dimensions (**Figures 2C–E**), and that possibly gave them a higher proportion of reserves to subsidize germination, such as lipids (Zhou et al., 2019). It has also been found that lower exposure of soybean seeds to stress situations, such as radiation (Oliveira et al., 2021), preserves their brightness characteristics. In alfalfa, it has been shown that the natural aging of seeds itself interferes with this aspect (Wang X. et al., 2021). It is worth noting that the reduction in tegument brightness is a common phenomenon in other species, such as beans (Piotrowicz-Cieślak et al., 2020), and may indicate the advancement of oxidative processes associated with seed deterioration (Erfatpour et al., 2021). In orthodox seeds, such as peanuts, seed deterioration occurs in progressive stages at the cellular level and results in loss of vigor (Ebone et al., 2019). Thus, the physical variables explored in this work through multispectral images demonstrated potential for quality control during the processing of peanut seedlots.

## Seed Pigments

In addition to the above physical properties, pigments in peanut seeds have also been found to add useful information for the seed industry. In an initial explanation, it can be pointed out that high quality seedlots may contain extra volume of both reserves and pigments (**Figures 3A,B**) due to their higher weight and area (**Figures 2B,C**). In fact, the low relation between chlorophyll *a/b* (**Figure 3C**) indicated a higher proportion of chlorophyll *b* in high quality lots (**Figure 4B**). In senescent plant tissues, the reduction in chlorophyll fluorescence is described as a deteriorating process (Donaldson and Williams, 2018; Donaldson, 2020). From this perspective, peanut seed quality may be directly associated with chlorophyll fluorescence dynamics. It may also be associated to the accumulation of

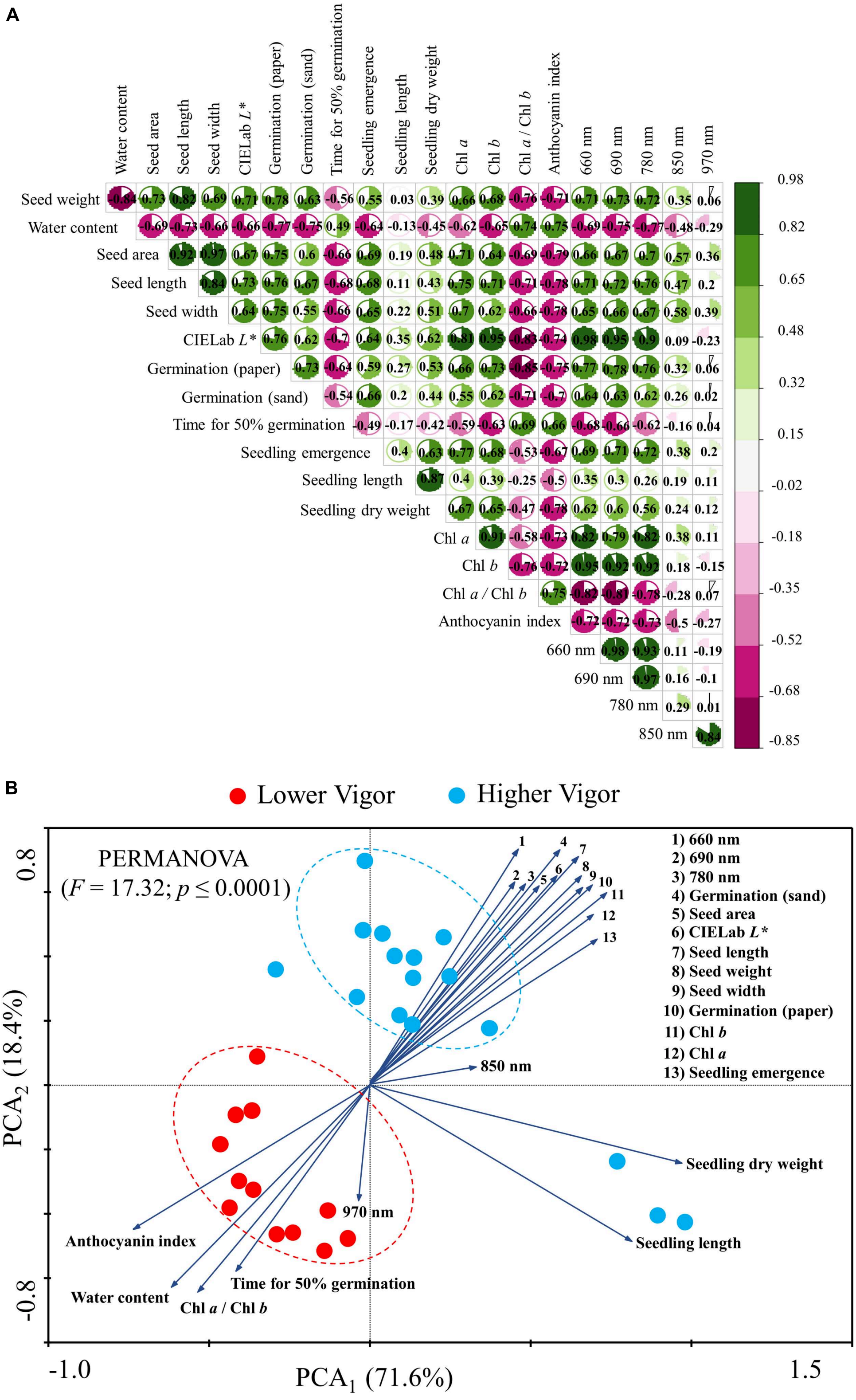


**FIGURE 9 |** Correlation matrix **(A)** and biplots of principal component analysis (PCA) **(B)** for physical optical descriptors, physiological, pigment, and reflectance of peanut seeds (*Arachis hypogaea* L.; cv. IAC OL3) with lower (lots 1, 2, and 3; red circles) and higher vigor (lots 4, 5, 6, and 7; blue circles). The PCA vectors indicate the correlation between the classes (lower and higher vigor) and the dimensions $PC_1$ and $PC_2$. We used the PERMANOVA test and the Bray-Curtis similarity index in the PCA to identify the difference between seed classes at a 1% significance.

**TABLE 1 |** Quadratic discriminant analysis (QDA) based on physical optical descriptors, pigments and reflectance of peanut seedlots (*Arachis hypogaea* L.; cv. IAC OL3) for groups of lower and higher vigor.

| **Predictor variable: area, length, width and CIELab *L** (physical optical descriptors)** | | | | | | |
|---|---|---|---|---|---|---|
| **Seedlot groups*** | **Training set (*n* = 833)$^{1}$** | | | **Validation set (*n* = 357)$^{1}$** | | |
| | **Lower Vigor** | **Higher Vigor** | **Accuracy** | **Lower Vigor** | **Higher Vigor** | **Accuracy** |
| Lower Vigor | 0.94 | 0.14 | 0.91 | 0.93 | 0.15 | 0.89 |
| Higher Vigor | 0.06 | 0.86 | | 0.07 | 0.85 | |
| **Predictor variable: Chlorophyll *a*, Chlorophyll *b* and anthocyanins (pigments)** | | | | | | |
| **Seedlot groups*** | **Training set (*n* = 833)$^{1}$** | | | **Validation set (*n* = 357)$^{1}$** | | |
| | **Lower Vigor** | **Higher Vigor** | **Accuracy** | **Lower Vigor** | **Higher Vigor** | **Accuracy** |
| Lower Vigor | 0.91 | 0.06 | 0.93 | 0.96 | 0.07 | 0.94 |
| Higher Vigor | 0.09 | 0.94 | | 0.04 | 0.93 | |
| **Predictor variable: 660, 690, 780, 850, and 970 nm (reflectance)** | | | | | | |
| **Seedlot groups*** | **Training set (*n* = 833) $^{1}$** | | | **Validation set (*n* = 357)$^{1}$** | | |
| | **Lower Vigor** | **Higher Vigor** | **Accuracy** | **Lower Vigor** | **Higher Vigor** | **Accuracy** |
| Lower Vigor | 0.98 | 0.04 | 0.97 | 0.99 | 0.05 | 0.97 |
| Higher Vigor | 0.02 | 0.96 | | 0.01 | 0.95 | |
| **Predictor variable: physical optical descriptors, pigments and reflectance** | | | | | | |
| **Seedlot groups** | **Training set (*n* = 833) $^{1}$** | | | **Validation set (*n* = 357) $^{1}$** | | |
| | **Lower Vigor** | **Higher Vigor** | **Accuracy** | **Lower Vigor** | **Higher Vigor** | **Accuracy** |
| Lower Vigor | 0.99 | 0 | 0.99 | 0.98 | 0.02 | 0.98 |
| Higher Vigor | 0.01 | 1 | | 0.02 | 0.98 | |

**Lower Vigor: lots 1, 2, and 3; Higher Vigor: lots 4, 5, 6, and 7.*

*[1] From the dataset observed in all seedlots (n = 1190), 70% (n = 833) were randomly sampled for training assessment and 30% for validation (n = 357).*

anthocyanins (**Figures 3D**, **4B,C**) since the biosynthesis of this flavonoid is part of the secondary metabolism of plants against stress (Liu et al., 2018). Further studies were conducted in order to understand whether pigment dynamics in peanut seeds interfere with their quality. For this purpose, seeds from one of the lots identified as high quality (high germination and vigor) were exposed to controlled stress (artificial aging).

Stress applied to peanut seeds (aged seeds) caused changes in pigment dynamics (chlorophylls fluorescence and anthocyanin index) and brightness (CIELab *L**). Considering mature, non-greenish soybean seeds, the fluorescence of chlorophylls (residual in the embryo) decreases as the artificial aging process under high temperature and high RH progresses (Barboza da Silva et al., 2021b). Furthermore, the increased exposure of seeds to this stress (high temperature and high RH) reduces their ability to form vigorous seedlings. It has been demonstrated that mature soybean seeds with reduced germination have lower chlorophyll fluorescence characteristics than seeds with higher viability (Li et al., 2019). Thus, the possibility exists that the loss of fluorescence occurs as seeds age. In plants, this has been documented for leaf tissues in advanced senescence (Donaldson, 2020). In this work, the reduction in chlorophyll fluorescence and brightness of seeds exposed to stress (IAC OL3 and IAC 503) reinforces the idea that the degree of deterioration or aging of peanut seeds alters their spectral properties. Taking these observations into consideration, pigment dynamics and seed brightness can be indicators of seed quality. Also, both reveal the degree of stress accumulated in seed tissues. In the peanut seed industry, technologies that detect these characteristics through multispectral images have a promising potential to improve lot quality control and making it more accurate.

## Seed Reflectance

Another promising possibility for assessing seed quality was found in this work through reflectance. Higher quality lots were formed by seeds with high reflectance at wavelengths between 660 and 780 nm (**Figures 8A,B**). The peculiarities of seeds, such as chemical composition, color and other attributes, are known to interfere in the absorbance and reflectance dynamics of incident light (Elmasry et al., 2019a). It is worth noting that high quality peanut seeds contained a naturally enhanced chlorophyll fluorescence, especially Chl *b*, and higher tegument brightness (**Figure 2F**). In this context, these characteristics can have contributed to the increased light reflected by the better-quality seeds, thus defining their high reflectance pattern

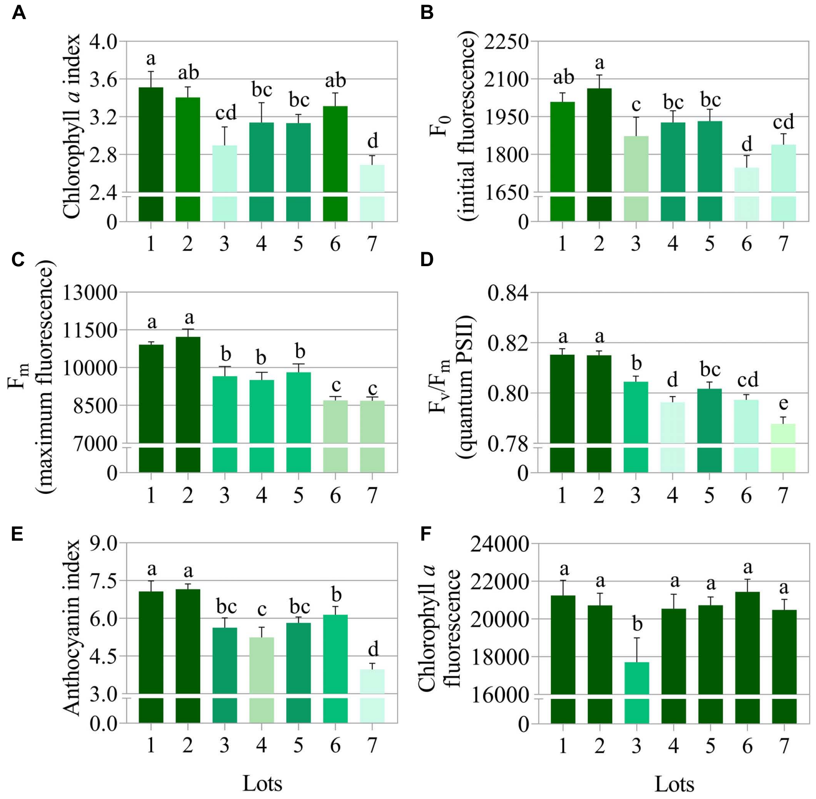


**FIGURE 10 |** Photosynthetic activity measured by chlorophyll *a* index **(A)**, initial fluorescence ($F_0$) **(B)**, maximum fluorescence ($F_m$) **(C)**, quantum yield of photosystem II measured by $F_v/F_m$ **(D)**, anthocyanin index **(E)**, and chlorophyll *a* fluorescence **(F)** in peanut seedlings (*Arachis hypogaea* L.; cv. IAC OL3) at 14 days after sowing: excitation of chlorophyll molecules were induced at 620 nm and emission at 700 nm. Means (± standard deviation) with different letters indicate a significant difference ($p \leq 0.05$). Peanut seedlings were obtained from seeds of lower (Lots 1, 2, and 3) and higher vigor (Lots 4, 5, 6, and 7).

in specific bands (**Figures 8A–C**). Apparently, this behavior is not a common and interspecific rule in nature. As an example, *Jatropha curcas* seeds have superior quality associated with enhancement in their lipid content, which results in low reflectance in the near infrared range (940 nm) (Bianchini et al., 2021). In tomato seeds, on the other hand, this high performance and low reflectance are linked to embryo maturity and protective pigments that absorb more light in the UV spectrum (365 nm) (Galletti et al., 2020). Here, the physiological quality attributes (germination and vigor) were associated with high reflectance at specific wavelengths (660 to 780 nm), so far not considered for peanut seeds. It is worth noting that in the plant domain (bermudagrass), higher reflectance values (900/970 nm) can be strongly associated with leaf water content under water stress conditions (Caturegli et al., 2020). Therefore, the reflectance patterns obtained in this work show a singular behavior with a unique competence to define the physiological quality of peanut seeds.

## Data Correlation and Seed Quality Classification Using Machine Learning

Summarizing our findings, it is worth highlighting the significant correlations between physical optical parameters (area, length, width and brightness – CIELab *L**), pigments (chlorophyll fluorescence and anthocyanin) and reflectances (660, 690, and 780 nm) with germination and seed vigor (**Figure 10A**). These results establish an unprecedented connection between tests performed to assess seed quality with multispectral images parameters, with the goal of categorizing seedlots with high quality. Furthermore, they demonstrated the robustness of potential markers of peanut seed physiological quality found through non-invasive technologies. The principal component analysis method proved to be an efficient technique for interpreting the behavior of seedlots (high and low vigor). The gain in the ability to manage datasets using PCA has been highlighted (Taheri-Garavand et al., 2021c). However, it should be considered that the manual management of the volume of data generated through multispectral seed images can hinder decision making in routine analyses in the seed industry. Separating the behavior of the seedlots into groups of low and high vigor (**Figure 10B**) brought up the following question: in practice, how can these differences in seed quality be quickly diagnosed using only the generated database containing all multispectral image parameters found? With this in mind, ways to automatically recognize seeds of high and low quality were tested using computational resources of high predictive accuracy.

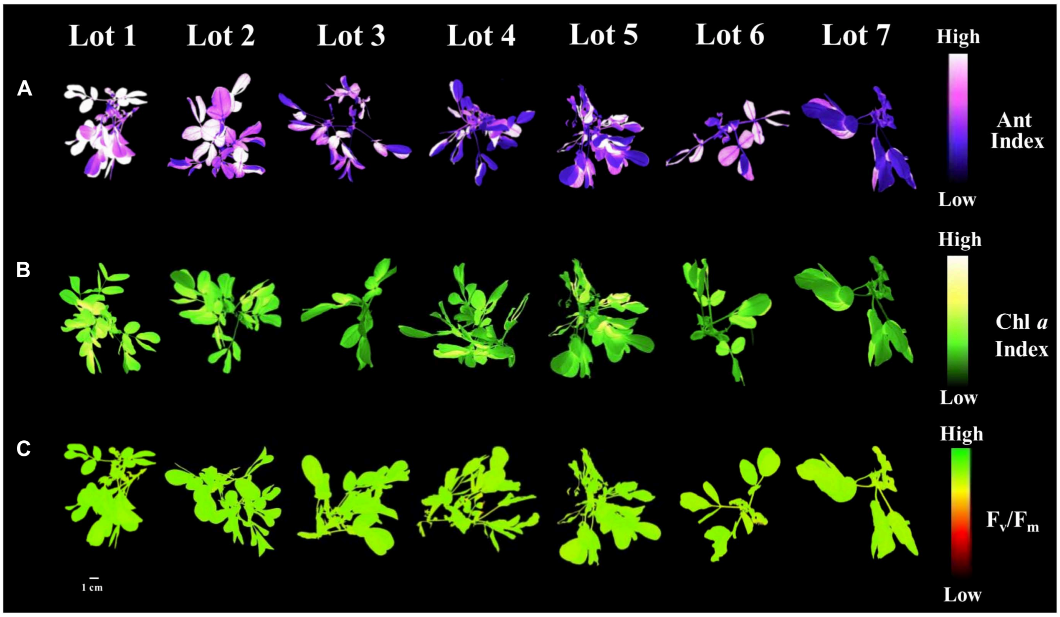


**FIGURE 11 |** Anthocyanin index **(A)**, chlorophyll *a* index **(B)**, and maximum quantum efficiency of photosystem II based on $F_V/F_m$ **(C)** in peanut seedlings (*Arachis hypogaea* L.; cv. IAC OL3) from seedlot 7. The pigments and photosynthetic efficiency of peanut seedlings were evaluated at 14 days after sowing of the seeds from the seven lots studied. Each pixel in the image is represented by a unique value that corresponds to fluorescence intensity; higher pixel values indicate higher anthocyanin, fluorescence and $F_V/F_m$ intensity.

From the multispectral seed dataset, the surprising sensitivity of machine learning algorithms based on the QDA method (**Table 1**) was verified for autonomous recognition of patterns identified in conventional seed quality analysis (**Figure 10B**). It is worth noting that the QDA method is quite robust to data non-normality (lower error probability), except when distributions are highly asymmetric (Clarke et al., 1979), different from what was observed here (**Supplementary Methodology 1**). Also, it is an efficient parametric method because it takes into account the low variability when different data sets are used to build prediction models (James et al., 2021). The QDA method has been used successfully in the field of Plant Science, with examples ranging from protein structure classification (Yuan et al., 2017) to phytosanitary diagnosis from plant oil dielectric properties (Khaled et al., 2018). The use of the QDA method as part of an artificial intelligence strategy applied to post-harvest proved to be a powerful tool for categorizing the quality of peanut seedlots.

This possibility of automation was successfully explored in previous studies for the analysis of image parameters of seeds from other crops (Elmasry et al., 2019a; Mortensen et al., 2021). In species such as soybean (Baek et al., 2019; Medeiros et al., 2020a), cowpea (Rego et al., 2020), oat (França-Silva et al., 2020), *U. brizantha* (Medeiros et al., 2020b) and corn (Wang Z. et al., 2021), the ability of algorithms to detect spectral features of seeds with high accuracy (above 90%) through images was proven. Taking this knowledge into consideration in addition to the findings of this work (**Table 1**), it is clear that part of the modernization process of the seed production sector in the world can be based on the use of multispectral image technologies. In the peanut production chain, these devices capable of capturing images in the UV, visible and near-infrared range have the potential to promote strategies to mitigate the incidence of seeds with low vigor in commercial lots. This problem, in addition to hampering the proper formation of a crop (Carter et al., 2019), can lead to a higher number of seeds needed to meet the intended plant stand. At this point, artificial intelligence resources have shown to be highly capable of improving seed quality management programs based on detailed and real-time diagnosis of the seedlots.

## Pigments and Photosynthetic Efficiency of Seedlings

In face of the primary technological aim of the seeds, which is the establishment of a seedling, its association with seed quality was investigated. Interestingly, seeds of low physiological quality gave rise to seedlings with superior photosynthesis parameters (**Figures 11A–D**). Even with the enhancement of the photosynthetic potential, there was an increase in the anthocyanin index in the leaves (**Figure 11E**), which indicates some degree of stress (Liu et al., 2018). The outcome of these results motivated us to think about whether there is an intrinsic protection mechanism in peanut seeds that helps the establishment of the seedling with low vigor. This can be a natural survival strategy in unfavorable situations (stress), which optimizes the chances of perpetuating the species in the cultivation environment, as discussed for other species (Marcos et al., 2018a,b). A similar proposal was explored in

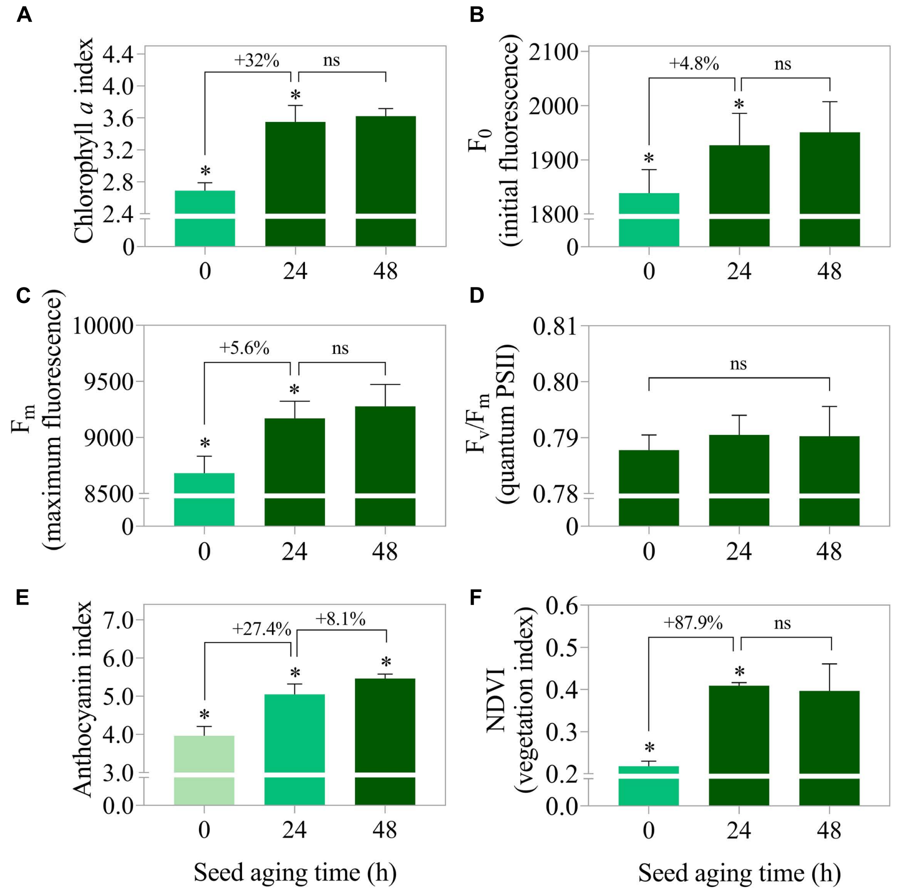


**FIGURE 12 |** Photosynthetic activity measured by chlorophyll *a* index **(A)**, initial fluorescence ($F_0$) **(B)**, maximum fluorescence ($F_m$) **(C)**, photosystem II quantum yield measured by $F_v/F_m$ **(D)** as well as stress indicators such as anthocyanin index **(E)**, and normalized vegetation index (NDVI) **(F)** in peanut seedlings (*Arachis hypogaea* L.; cv. IAC OL3) at 14 days after sowing: excitation of chlorophyll molecules were induced at 620 nm and emission at 700 nm. Means (± standard deviation). Asterisks (*) indicate significant differences ($p \leq 0.05$). Peanut seedlings were obtained from lot 7 seeds after aging times (0 h, 24 h, and 48 h at 42°C).

tomato (Nogueira et al., 2021), and it was found that seeds produced in a stressful environment gave rise to seedlings with adaptive enhancement in chlorophyll fluorescence. Thus, considering the notable connection of chlorophyll fluorescence found (**Figure 11A**) with photosynthesis in plant organisms (Valcke, 2021), the idea that peanut seedlings signaled compensatory adjustments in photosynthetic capacity in response to seed deterioration induced by artificial aging (high temperature and high RH) was proposed. In order to better understand these concepts, seedlings from seeds exposed to stress (24 and 48 h at 42°C/100% RH) were produced and assessed for their photosynthetic capacity as well as stress indicators (anthocyanin and normalized vegetation indices).

Surprisingly, after 24 h of artificial aging of high-quality seeds (lot 7), there was a proportional enhancement in the photosynthetic parameters of the seedlings (**Figures 12A–C**), besides an evident increase in leaf stress indices (**Figures 12D,E**). It is interesting to think that if the deteriorated seeds were really conditioned to access stress repair mechanisms, in practice the low quality of seedlots would naturally be compensated without harming the establishment of seedlings. On the other hand, seeds in this condition can lead to failures in the stand due to a higher incidence of abnormal seedlings and/or non-viable seeds (**Supplementary Figure 2**). Thus, seeds of low vigor should not be used for the installation of tillage, since the negative reflexes of the failures they cause in the plant stand extend to the harvest and reduce grain yield (Bagateli et al., 2019; Ebone et al., 2020). In this way, multispectral images of seedlings can provide information associated with their photosynthetic apparatus with the reverse logic of what happens in seeds (**Figure 13B**). For this reason, they need prior knowledge of the level of seed deterioration to effectively contribute as a marker of the physiological quality of seedlots. Still, stress indicators such as the levels of anthocyanins found (**Figures 12D, 13A**) connected more directly with what occurs in seeds (**Figure 5D**). Such results have the potential to anticipate the behavior of post-germination events and integrate robust quality control programs associated with seedling establishment. Also, allows the prediction of physiological dysfunctions associated

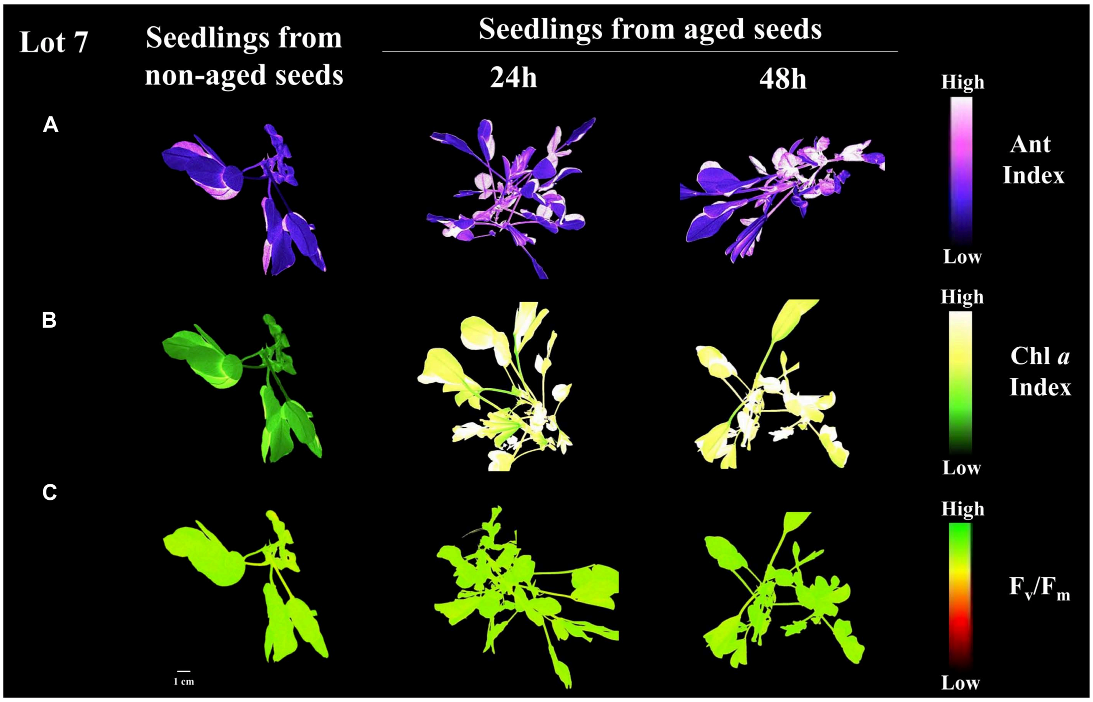


**FIGURE 13 |** Anthocyanin index **(A)**, chlorophyll α index **(B)**, and maximum quantum efficiency of photosystem II based on $F_v/F_m$ **(C)** in peanut seedlings (*Arachis hypogaea* L.; cv. IAC OL3) from seeds of lot 7 after artificial aging (24 h and 48 h at 42°C). Each pixel in the image is represented by a unique value that corresponds to fluorescence intensity; higher pixel values indicate higher anthocyanin, fluorescence and $F_v/F_m$ intensity.

with seed deterioration and the initial photosynthetic behavior of a crop in the field, which deserves to be explored in future investigations.

## Perspectives

These are innovative techniques to assess the quality of peanut seedlots in a non-destructive and accurate way. The possibility of providing farmers with seeds that are highly capable of generating productive plants makes the search for these innovations one of the technological priorities in agriculture. Multispectral images represent a sensory bridge that extends human vision to access information hitherto unexplored in peanut seeds. A practical example is that through images, seedlots of lower quality can be identified. They generate seedlings with higher levels of stress (anthocyanins). Therefore, these lots can be allocated to less stressful cultivation environments in order to take advantage of the seed stock, within a certain quality level, and mitigate possible losses in the future crop. From the quality markers found, improvement solutions can be thought along the peanut production chain, from classification in processing to seed quality control. There is also, the opportunity to carry out these steps autonomously through machine learning models (QDA method). On a commercial scale, a capital investment is initially required to adopt the approach employed (Taheri-Garavand et al., 2021a). However, the wide applications of these technologies in the seed industry can bring significant returns through two aspects: (i) increased efficiency of post-harvest processes and, consequently, (ii) cost reduction.

## CONCLUSION

New markers that effectively track peanut seed quality were found. The combination of physical properties (area, length, width, and coat brightness), pigments (chlorophyll fluorescence and anthocyanin), and light reflectance (660, 690, and 780 nm), is highly efficient to identify peanut seedlots with superior quality (98% accuracy). Regarding seedlings, stress indicators such as anthocyanins directly reflect the quality of the seedlots. The association of these markers with artificial intelligence highlights the potential for automation of post-harvest processes integrated with quality analysis logistics in the peanut seed industry. Overall, our findings provide valuable insights for managing the quality attributes of one of the most essential inputs to the world's agricultural activity: the seed.

## 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

GF, CM, and EA generated the research ideas. GF collected seed physiological quality data and wrote and formatted the manuscript. CM and JS collected multispectral image analysis data. WH and GF analyzed the data. CM, TB, AP, CC, and EA reviewed the manuscript, rewriting, discussing, and commenting. All authors read and approved the final manuscript.

## FUNDING

This work was supported by: National Council for Scientific and Technological Development (CNPq; Grant numbers 142236/2020-9, 309718/2018-0, and 303119/2016-0) and São Paulo Research Foundation (FAPESP; Grant numbers #2014/16712-2, #2017/50211-9, #2017/15220-7, #2018/01774-3, #2018/03802-4, #2018/03793-5, #2018/01774-3, #2020/12686-8, and #2020/14050-3).

## ACKNOWLEDGMENTS

We are thankful to Roger Hutchings for the English review of the manuscript. We also thank COPERCANA and COPLANA (peanut seed companies, São Paulo, Brazil) for their support throughout the experiments.

## SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2022.849986/full#supplementary-material

## REFERENCES

Arya, S. S., Salve, A. R., and Chauhan, S. (2016). Peanuts as functional food: a review. *J. Food Sci. Technol.* 53, 31–41. doi: 10.1007/s13197-015-2007-9

Baek, I., Kusumaningrum, D., Kandpal, L. M., Lohumi, S., Mo, C., Kim, M. S., et al. (2019). Rapid measurement of soybean seed viability using Kernel-based multispectral image analysis. *Sensors* 19:271. doi: 10.3390/s19020271

Bagateli, J. R., Dörr, C. S., Schuch, L. O. B., and Meneghello, G. E. (2019). Productive performance of soybean plants originated from seed lots with increasing vigor levels. *J. Seed Sci.* 41, 151–159. doi: 10.1590/2317-1545v41n2199320

Barbosa, R. M., Vieira, B. G. T. L., Martins, C. C., and Vieira, R. D. (2014). Qualidade fisiológica e sanitária de sementes de amendoim durante o processo de produção. *Pesqui. Agropecu. Bras.* 49, 977–985. doi: 10.1590/S0100-204X2014001200008

Barboza da Silva, C., Bianchini, V. D. J. M., de Medeiros, A. D., de Moraes, M. H. D., Marassi, A. G., and Tannús, A. (2021a). A novel approach for Jatropha curcas seed health analysis based on multispectral and resonance imaging techniques. *Ind. Crops Prod.* 161:113186. doi: 10.1016/j.indcrop.2020.113186

Barboza da Silva, C., Oliveira, N. M., de Carvalho, M. E. A., de Medeiros, A. D., de Lima Nogueira, M., and dos Reis, A. R. (2021b). Autofluorescence-spectral imaging as an innovative method for rapid, non-destructive and reliable assessing of soybean seed quality. *Sci. Rep.* 11:17834. doi: 10.1038/s41598-021-97223-5

Basso, D. P., Hoshino-Bezerra, A. A., Sartori, M. M. P., Buitink, J., Leprince, O., and da Silva, E. A. A. (2018). Late seed maturation improves the preservation of seedling emergence during storage in soybean. *J. Seed Sci.* 40, 185–192. doi: 10.1590/2317-1545v40n2191893

Bessada, S. M. F., Barreira, J. C. M., and Oliveira, M. B. P. P. (2019). Pulses and food security: dietary protein, digestibility, bioactive and functional properties. *Trends Food Sci. Technol.* 93, 53–68. doi: 10.1016/j.tifs.2019.08.022

Bianchini, V. D. J. M., Mascarin, G. M., Silva, L. C. A. S., Arthur, V., Carstensen, J. M., Boelt, B., et al. (2021). Multispectral and X-ray images for characterization of *Jatropha curcas* L. seed quality. *Plant Methods* 17:9. doi: 10.1186/s13007-021-00709-6

Buitink, J., and Leprince, O. (2004). Glass formation in plant anhydrobiotes: survival in the dry state. *Cryobiology* 48, 215–228. doi: 10.1016/j.cryobiol.2004.02.011

Carter, E. T., Rowland, D. L., Tillman, B. L., Erickson, J. E., Grey, T. L., Gillett-Kaufman, J. L., et al. (2019). An analysis of the physiological impacts on life history traits of peanut (*Arachis hypogaea* L.) related to seed maturity. *Peanut Sci.* 46, 148–161. doi: 10.3146/ps18-20.1

Caturegli, L., Matteoli, S., Gaetani, M., Grossi, N., Magni, S., Minelli, A., et al. (2020). Effects of water stress on spectral reflectance of bermudagrass. *Sci. Rep.* 10:15055. doi: 10.1038/s41598-020-72006-6

Clarke, W. R., Lachenbruch, P. A., and Broffitt, B. (1979). How non-normality affects the quadratic discriminant function. *Commun. Stat. Theory Methods* 8, 1285–1301. doi: 10.1080/03610927908827830

Ding, N., Xing, F., Liu, X., Selvaraj, J. N., Wang, L., Zhao, Y., et al. (2015). Variation in fungal microbiome (mycobiome) and aflatoxin in stored in-shell peanuts at four different areas of China. *Front. Microbiol.* 6:1055. doi: 10.3389/fmicb.2015.01055

Donaldson, L. (2020). Autofluorescence in plants. *Molecules* 25:2393. doi: 10.3390/molecules25102393

Donaldson, L., and Williams, N. (2018). Imaging and spectroscopy of natural fluorophores in pine needles. *Plants* 7:10. doi: 10.3390/plants7010010

Ebone, L. A., Caverzan, A., and Chavarria, G. (2019). Physiologic alterations in orthodox seeds due to deterioration processes. *Plant Physiol. Biochem.* 145, 34–42. doi: 10.1016/j.plaphy.2019.10.028

Ebone, L. A., Caverzan, A., Tagliari, A., Chiomento, J. L. T., Silveira, D. C., and Chavarria, G. (2020). Soybean seed vigor: uniformity and growth as key factors to improve yield. *Agronomy* 10:545. doi: 10.3390/agronomy10040545

Elmasry, G., Mandour, N., Al-Rejaie, S., Belin, E., and Rousseau, D. (2019a). Recent applications of multispectral imaging in seed phenotyping and quality monitoring - an overview. *Sensors* 19:1090. doi: 10.3390/s19051090

Elmasry, G., Mandour, N., Wagner, M. H., Demilly, D., Verdier, J., Belin, E., et al. (2019b). Utilization of computer vision and multispectral imaging techniques for classification of cowpea (*Vigna unguiculata*) seeds. *Plant Methods* 15:24. doi: 10.1186/s13007-019-0411-2

Erfatpour, M., Duizer, L., and Pauls, K. P. (2021). Investigations of the effects of the non-darkening seed coat trait coded by the recessive jj alleles on agronomic, sensory, and cooking characteristics in pinto beans. *Crop Sci.* 61, 1843–1863. doi: 10.1002/csc2.20477

Finch-Savage, W. E., and Bassel, G. W. (2016). Seed vigour and crop establishment: extending performance beyond adaptation. *J. Exp. Bot.* 67, 567–591. doi: 10.1093/jxb/erv490

França-Silva, F., Rego, C. H. Q., Gomes-Junior, F. G., de Moraes, M. H. D., de Medeiros, A. D., and da Silva, C. B. (2020). Detection of drechslera avenae (Eidam) sharif [*Helminthosporium avenae* (eidam)] in black oat seeds (*Avena strigosa* schreb) using multispectral imaging. *Sensors* 20:3343. doi: 10.3390/s20123343

Galletti, P. A., Carvalho, M. E. A., Hirai, W. Y., Brancaglioni, V. A., Arthur, V., and Barboza da Silva, C. (2020). Integrating optical imaging tools for rapid and non-invasive characterization of seed quality: tomato (*Solanum lycopersicum* L.) and Carrot (*Daucus carota* L.) as study cases. *Front. Plant Sci.* 11:577851. doi: 10.3389/fpls.2020.577851

Gitelson, A. A., Chivkunova, O. B., and Merzlyak, M. N. (2009). Nondestructive estimation of anthocyanins and chlorophylls in anthocyanic leaves. *Am. J. Bot.* 96, 1861–1868. doi: 10.3732/ajb.0800395

Gitelson, A. A., Gritz, Y., and Merzlyak, M. N. (2003). Relationships between leaf chlorophyll content and spectral reflectance and algorithms for non-destructive chlorophyll assessment in higher plant leaves. *J. Plant Physiol.* 160, 271–282. doi: 10.1078/0176-1617-00887

Groot, S. P. C., van Litsenburg, M. J., Kodde, J., Hall, R. D., de Vos, R. C. H., and Mumm, R. (2022). Analyses of metabolic activity in peanuts under hermetic storage at different relative humidity levels. *Food Chem.* 373:131020. doi: 10.1016/j.foodchem.2021.131020

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2 Edn. Germany: Springer.

Herritt, M. T., Pauli, D., Mockler, T. C., and Thompson, A. L. (2020). Chlorophyll fluorescence imaging captures photochemical efficiency of grain sorghum (*Sorghum bicolor*) in a field setting. *Plant Methods* 16:109. doi: 10.1186/s13007-020-00650-0

Hu, X., Yang, L., and Zhang, Z. (2020). Non-destructive identification of single hard seed via multispectral imaging analysis in six legume species. *Plant Methods* 16:116. doi: 10.1186/s13007-020-00659-5

ISTA (2020). *International Rules for Seed Analysis. International Rules for Seed Testing*. Bassersdorf: Zürischstr.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2021). *An Introduction to Statistical Learning with Applications in R*. Berlin: Springer.

Joosen, R. V. L., Kodde, J., Willems, L. A. J., Ligterink, W., Van Der Plas, L. H. W., and Hilhorst, H. W. M. (2010). Germinator: a software package for high-throughput scoring and curve fitting of Arabidopsis seed germination. *Plant J.* 62, 148–159. doi: 10.1111/j.1365-313X.2009.04116.x

Khaled, A. Y., Abd Aziz, S., Khairunniza Bejo, S., Mat Nawi, N., Abu Seman, I., and Izzuddin, M. A. (2018). Development of classification models for basal stem rot (BSR) disease in oil palm using dielectric spectroscopy. *Ind. Crops Prod.* 124, 99–107. doi: 10.1016/j.indcrop.2018.07.050

Krzyzanowski, F. C., França-Neto, J. B., Gomes-Junior, F. G., and Nakagawa, J. (2020). "Testes de vigor baseado em desempenho de plântulas," in *Vigor de Sementes: Conceitos e Testes*, 2 Edn, eds F. C. Krzyzanowski, R. D. Vieira, J. B. França-Neto, and J. Marcos-Filho (Londrina: ABRATES).

Kuhn, M. (2017). *Caret Package: Classification and Regression Training*. Available Online at: https://cran.r552project.org/web/packages/caret/index.html [accessed August 16, 2021].

Leprince, O., Pellizzaro, A., Berriri, S., and Buitink, J. (2017). Late seed maturation: drying without dying. *J. Exp. Bot.* 68, 827–841. doi: 10.1093/jxb/erw363

Li, Y., Sun, J., Wu, X., Chen, Q., Lu, B., and Dai, C. (2019). Detection of viability of soybean seed based on fluorescence hyperspectra and CARS-SVM-AdaBoost model. *J. Food Process. Preserv.* 43, 1–9. doi: 10.1111/jfpp.14238

Liu, Y., Tikunov, Y., Schouten, R. E., Marcelis, L. F. M., Visser, R. G. F., and Bovy, A. (2018). Anthocyanin biosynthesis and degradation mechanisms in *Solanaceous* vegetables: a review. *Front. Chem.* 6:52. doi: 10.3389/fchem.2018.00052

Marcos, F. C. C., Silveira, N. M., Marchiori, P. E. R., Machado, E. C., Souza, G. M., Landell, M. G. A., et al. (2018a). Drought tolerance of sugarcane propagules is improved when origin material faces water deficit. *PLoS One* 13:e0206716. doi: 10.1371/journal.pone.0206716

Marcos, F. C. C., Silveira, N. M., Mokochinski, J. B., Sawaya, A. C. H. F., Marchiori, P. E. R., Machado, E. C., et al. (2018b). Drought tolerance of sugarcane is improved by previous exposure to water deficit. *J. Plant Physiol.* 223, 9–18. doi: 10.1016/j.jplph.2018.02.001

Medeiros, A. D., da Silva, L. J., Ribeiro, J. P. O., Ferreira, K. C., Rosas, J. T. F., Santos, A. A., et al. (2020b). Machine learning for seed quality classification: an advanced approach using merger data from FT-NIR spectroscopy and x-ray imaging. *Sensors* 20:4319. doi: 10.3390/s20154319

Medeiros, A. D., Capobiango, N. P., da Silva, J. M., da Silva, L. J., da Silva, C. B., and dos Santos Dias, D. C. F. (2020a). Interactive machine learning for soybean seed and seedling quality classification. *Sci. Rep.* 10:11267. doi: 10.1038/s41598-020-68273-y

Meireles, J. E., Cavender-Bares, J., Townsend, P. A., Ustin, S., Gamon, J. A., Schweiger, A. K., et al. (2020). Leaf reflectance spectra capture the evolutionary history of seed plants. *New Phytol.* 228, 485–493. doi: 10.1111/nph.16771

Mortensen, A. K., Gislum, R., Jørgensen, J. R., and Boelt, B. (2021). The use of multispectral imaging and single seed and bulk near-infrared spectroscopy to characterize seed covering structures: methods and applications in seed testing and research. *Agriculture* 11:301. doi: 10.3390/agriculture11040301

Murchie, E. H., and Lawson, T. (2013). Chlorophyll fluorescence analysis: a guide to good practice and understanding some new applications. *J. Exp. Bot.* 64, 3983–3998. doi: 10.1093/jxb/ert208

Nogueira, M. L., Carvalho, M. E. A., Ferreira, J. M. M., Bressanin, L. A., Piotto, K. D. B., Piotto, F. A., et al. (2021). Cadmium-induced transgenerational effects on tomato plants: a gift from parents to progenies. *Sci. Total Environ.* 789:147885. doi: 10.1016/j.scitotenv.2021.147885

Okada, M. H., Fosenca de Oliveira, G. R., Sartori, M. M. P., Nakagawa, J., Crusciol, C. A. C., and Amaral da Silva, E. A. (2021). Acquisition of the physiological quality of peanut (*Arachis hypogaea* L.) seeds during maturation under the influence of the maternal environment. *PLoS One* 16:e0250293. doi: 10.1371/journal.pone.0250293

Oliveira, N. M., de Medeiros, A. D., Nogueira, M. D. L., Arthur, V., Mastrangelo, T. D. A., and Barboza da Silva, C. (2021). Hormetic effects of low-dose gamma rays in soybean seeds and seedlings: a detection technique using optical sensors. *Comput. Electron. Agric.* 187:106251. doi: 10.1016/j.compag.2021.106251

Piotrowicz-Cieślak, A. I., Krupka, M., Michalczyk, D. J., Smyk, B., Grajek, H., Podyma, W., et al. (2020). Physiological characteristics of field bean seeds (*Vicia faba* var. minor) subjected to 30 years of storage. *Agriculture* 10:545. doi: 10.3390/agriculture10110545

R Core Team (2021). *A Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing.

Rego, C. H. Q., França-Silva, F., Gomes-Junior, F. G., de Moraes, M. H. D., de Medeiros, A. D., and da Silva, C. B. (2020). Using multispectral imaging for detecting seed-borne fungi in cowpea. *Agriculture* 10:361. doi: 10.3390/agriculture10080361

Sano, N., Rajjou, L., North, H. M., Debeaujon, I., Marion-Poll, A., and Seo, M. (2016). Staying alive: molecular aspects of seed longevity. *Plant Cell Physiol.* 57, 660–674. doi: 10.1093/pcp/pcv186

Stalker, H. T., and Wilson, R. F. (2016). *Peanuts: Genetics, Processing, and Utilization*, 1st Edn. Cambridge, MA: Academic Press.

Taheri-Garavand, A., Nasiri, A., Fanourakis, D., Fatahi, S., Omid, M., and Nikoloudakis, N. (2021b). Automated in situ seed variety identification via deep learning: a case study in chickpea. *Plants* 10:1406. doi: 10.3390/plants10071406

Taheri-Garavand, A., Rezaei Nejad, A., Fanourakis, D., Fatahi, S., and Ahmadi Majd, M. (2021c). Employment of artificial neural networks for non-invasive estimation of leaf water status using color features: a case study in *Spathiphyllum wallisii*. *Acta Physiol. Plant.* 43:78. doi: 10.1007/s11738-021-03244-y

Taheri-Garavand, A., Mumivand, H., Fanourakis, D., Fatahi, S., and Taghipour, S. (2021a). An artificial neural network approach for non-invasive estimation of essential oil content and composition through considering drying processing factors: a case study in *Mentha aquatica*. *Ind. Crops Prod.* 171:113985. doi: 10.1016/j.indcrop.2021.113985

Temba, M. C., Njobeh, P. B., Adebo, O. A., Olugbile, A. O., and Kayitesi, E. (2016). The role of compositing cereals with legumes to alleviate protein energy malnutrition in Africa. *Int. J. Food Sci. Technol.* 51, 543–554. doi: 10.1111/ijfs.13035

USDA (2020b). *World Agricultural Production. Peanut Area, Yield Prod*. Available Online at: https://apps.fas.usda.gov/psdonline/circulars/production.pdf [accessed October 1, 2021].

USDA (2020a). *Oilseed, Peanut 2020. Peanut Explor. World Prod*. Available Online at: https://ipad.fas.usda.gov/cropexplorer/cropview/commodityView.aspx?cropid=2221000&sel_year=2020&startrow=11 [accessed October 1, 2021].

Valcke, R. (2021). Can chlorophyll fluorescence imaging make the invisible visible? *Photosynthetica* 59, 21–38. doi: 10.32615/ps.2021.017

Venables, W., and Ripley, B. (2002). *Modern Applied Statistics with S*. New York: Springer.

Wang, X., Zhang, H., Song, R., He, X., Mao, P., and Jia, S. (2021). Non—destructive identification of naturally aged alfalfa seeds via multispectral imaging analysis. *Sensors* 21:5804. doi: 10.3390/s21175804

Wang, Z., Tian, X., Fan, S., Zhang, C., and Li, J. (2021). Maturity determination of single maize seed by using near-infrared hyperspectral imaging coupled with comparative analysis of multiple classification models. *Infrared Phys. Technol.* 112:103596. doi: 10.1016/j.infrared.2020.103596

Yengoh, G. T., Dent, D., Olsson, L., Tengberg, A. E., and Tucker, C. J. III (2015). *Use of the Normalized Difference Vegetation Index (NDVI) to Assess Land Degradation at Multiple Scales: Current Status, Future Trends, and Practical Considerations*, 1st Edn. Berlin: Springer.

Yuan, L. Z., Yong, E. F., Wei, Z. G., and Shan, K. G. (2017). Using quadratic discriminant analysis to predict protein secondary structure based on chemical shifts. *Curr. Bioinform.* 12, 52–56. doi: 10.2174/1574893611666160628074537

Zhou, W., Branch, W. D., Gilliam, L., and Marshall, J. A. (2019). Phytosterol composition of *Arachis hypogaea* seeds from different maturity classes. *Molecules* 24:106. doi: 10.3390/molecules24010106

**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.

ORIGINAL RESEARCH
published: 28 April 2022
doi: 10.3389/fpls.2022.837726

# Evaluating Cross-Applicability of Weed Detection Models Across Different Crops in Similar Production Environments

*Bishwa B. Sapkota[1], Chengsong Hu[1,2] and Muthukumar V. Bagavathiannan[1]**

*[1]Department of Soil and Crop Sciences, Texas A&M University, College Station, TX, United States, [2]Department of Biological and Agricultural Engineering, College Station, TX, United States*

OPEN ACCESS

***Edited by:***
*Ian Stavness,*
*University of Saskatchewan,*
*Canada*

***Reviewed by:***
*Xin Zhang,*
*Mississippi State University,*
*United States*
*Thiago Teixeira Santos,*
*Brazilian Agricultural Research*
*Corporation (EMBRAPA), Brazil*

****Correspondence:***
*Muthukumar V. Bagavathiannan*
*muthu@tamu.edu*

***Specialty section:***
*This article was submitted to*
*Technical Advances in Plant Science,*
*a section of the journal*
*Frontiers in Plant Science*

***Received:*** *17 December 2021*
***Accepted:*** *30 March 2022*
***Published:*** *28 April 2022*

***Citation:***
*Sapkota BB, Hu C and*
*Bagavathiannan MV (2022)*
*Evaluating Cross-Applicability of*
*Weed Detection Models Across*
*Different Crops in Similar Production*
*Environments.*
*Front. Plant Sci. 13:837726.*
*doi: 10.3389/fpls.2022.837726*

Convolutional neural networks (CNNs) have revolutionized the weed detection process with tremendous improvements in precision and accuracy. However, training these models is time-consuming and computationally demanding; thus, training weed detection models for every crop-weed environment may not be feasible. It is imperative to evaluate how a CNN-based weed detection model trained for a specific crop may perform in other crops. In this study, a CNN model was trained to detect morningglories and grasses in cotton. Assessments were made to gauge the potential of the very model in detecting the same weed species in soybean and corn under two levels of detection complexity (levels 1 and 2). Two popular object detection frameworks, YOLOv4 and Faster R-CNN, were trained to detect weeds under two schemes: Detect_Weed (detecting at weed/crop level) and Detect_Species (detecting at weed species level). In addition, the main cotton dataset was supplemented with different amounts of non-cotton crop images to see if cross-crop applicability can be improved. Both frameworks achieved reasonably high accuracy levels for the cotton test datasets under both schemes (Average Precision-AP: 0.83–0.88 and Mean Average Precision-mAP: 0.65–0.79). The same models performed differently over other crops under both frameworks (AP: 0.33–0.83 and mAP: 0.40–0.85). In particular, relatively higher accuracies were observed for soybean than for corn, and also for complexity level 1 than for level 2. Significant improvements in cross-crop applicability were further observed when additional corn and soybean images were added to the model training. These findings provide valuable insights into improving global applicability of weed detection models.

**Keywords: deep learning, CNNs, digital technologies, precision weed control, site-specific weed management, precision agriculture**

## INTRODUCTION

Weeds are major pests in agricultural landscapes that can cause serious crop yield losses (Buchanan and Burns, 1970; Nave and Wax, 1971). A multi-tactic approach to weed management has become vital to thwart herbicide-resistant weed issues in global cropping systems (Bagavathiannan and Davis, 2018), and site-specificity is expected to improve control outcomes and conserve management inputs (Beckie et al., 2019). Injudicious use of agrochemicals has been linked to negative effects on non-target organisms and the broader environment (Liu and Bruch, 2020). Under the conventional broadcast approach, weed control tactics are applied without any regard to weed distribution and densities in the field. Weeds that escape the pre-emergent herbicides or mechanical tillage typically occur sparsely across the field. In such situations, weed control tactics can instead be strictly focused on areas of weed occurrence to save resources (Berge et al., 2012). In recent years, great efforts have been placed for developing and utilizing ground robots (Kargar and Shirzadifar, 2013; Aravind et al., 2015; Sujaritha et al., 2017; Lottes et al., 2019) and unmanned aerial systems (UAS) for site-specific weed control (Ahmad et al., 2020; Martin et al., 2020).

The precision weed control platforms ranging from ground robots to UAS-based selective spraying systems depend greatly on weed detection using computer vision techniques (Liu and Bruch, 2020; Machleb et al., 2020). The overall approach is to detect weeds in digital images and use the local or real world coordinates of the detected objects for site-specific control operations (López-Granados, 2011). In addition to weed control, these techniques offer tremendous opportunities for advancing weed ecology and biology research. Several image-based weed detection techniques have been proposed and implemented. Based on developments made so far, these techniques can be broadly categorized into two main groups: (1) traditional segmentation and machine learning-based techniques (Wu et al., 2011; Ahmed et al., 2012; Rumpf et al., 2012; García-Santillán and Pajares, 2018; Sabzi et al., 2018; Sapkota et al., 2020) and (2) advanced computer vision using convolution neural networks (CNNs; Adhikari et al., 2019; Ma et al., 2019; Sharpe et al., 2020; Hu et al., 2021; Xie et al., 2021).

The CNNs are a specialized type of neural networks that are designed to extract multi-scale features and merge semantically similar features for better prediction and/or detection (LeCun et al., 2015). The use of CNNs in weed detection tasks has gained great attention lately due to their ability to learn complex features through dense and rigorous feature representations (e.g., Xie et al., 2021). The attention has been fostered by the transfer learning concept in CNN that allows the sharing of common model weights from pre-trained models across different tasks (Abdalla et al., 2019; Fawakherji et al., 2019). The CNN-based object detection models have witnessed remarkable breakthroughs recently, and some of the detectors that have been widely used today for various detection tasks are Fast R-CNN (Girshick, 2015), Single-Shot Detector (Liu et al., 2016), Faster R-CNN (Ren et al., 2017), You Only Look Once (YOLO; Redmon et al., 2016), YOLOv3 (Redmon and Farhadi, 2018), YOLOv4 (Bochkovskiy et al., 2020), and more recently YOLOv5.

With respect to weed detection, different CNN-based detection frameworks have been successfully applied for various tasks. Gao et al. (2020) used YOLOv3 and Tiny YOLO models for detection of *Convolvulus sepium* (hedge bindweed) in *Beta vulgaris* (sugar beets) using field-collected and synthetic images. Using the same models, Jiang et al. (2020) also detected both grass and broadleaf weed species, including *Cirsium setosum*, *Descurainia sophia*, *Euphorbia helioscopia*, *Veronica didyma*, and *Avena fatua* in UAS-based Red-Green-Blue (RGB) imageries. Sharpe et al. (2020) detected goosegrass [*Eleusine indica* (L.) Gaertn.] in handheld digital camera-derived images obtained from two different horticultural crops, strawberry, and tomato, using YOLOv3-tiny model. Using YOLOv3, Partel et al. (2019) detected *Portulaca* spp. in pepper (*Capsicum annum*) for a precision spraying system. Yu et al. (2019) employed DetectNet to detect dandelion (*Taraxacum officinale*), ground ivy (*Glechoma hederacea*), and spotted spurge (*Euphorbia maculata*) in perennial ryegrass. Hu et al. (2021) tested Faster R-CNN, DeepLabv3, and Mask R-CNN for broadleaf and grass weed detection in cotton (*Gossypium hirsutum*) and soybean (*Glycine max*) using UAS-borne high-resolution images.

Cross-applicability of the deep learning models for weed detection across different crops is vital for two important reasons. First, several weed species continuously occur in the rotational crops in a given production field [e.g., *Amaranthus palmeri* (Palmer amaranth) occurring in both soybean and corn (*Zea mays*) grown in rotation], and computer vision models should be able to detect these weeds in all crops in the production system. Second, it is likely that the dominant weed species might be similar across production fields within a locality, and the ability to use these models across multiple production fields might be beneficial from efficiency and economic standpoint. This is because CNN models usually require a large set of annotated training images for better performance (Oquab et al., 2014; Gao et al., 2020), which can be difficult to obtain at times.

When only the weeds are annotated in the images and trained for detection, the model considers the crops in the same images as part of the background during the training process. Therefore, during inference, different crops may mimic different backgrounds for the same trained weeds in the images. It is therefore unclear how changes in the background (crop species in our case) may affect weed detection accuracies for different object detection frameworks under different detection scenarios. To the best of our knowledge, no study has looked at the cross-applicability of weed detection models across three of the most popular row crops in the United States: cotton, corn, and soybean. Such an investigation can further advance our understanding of weed detection models and help unleash their full potential.

The main goal of the study was to build a model for weed detection in cotton and investigate the use of the same model for detection of the same weed spectrum in corn and soybean. This study has two specific objectives: (1) build and evaluate models for weed detection in cotton under two weed detection schemes (detection of weeds at the meta-level, and detection at the individual weed species level), and (2) evaluate the performance of the cotton-based model on corn and soybean at different levels of detection complexity.

## MATERIALS AND METHODS

### Study Area and Experimental Setup

The study was conducted during the summers of 2020 and 2021 at the Texas A&M AgriLife Research farm (30°32′15″N, 96°25′35″W; elevation: 60 m). The location is characterized by a sub-tropical climate, with an average monthly maximum and minimum air temperatures during the study period (May–June) of 32.3 and 21.3°C, respectively. Glyphosate-resistant (Roundup Ready®) cotton and glufosinate-resistant (Liberty Link®) soybean were planted in two separate strips (**Figure 1**) adjacent to each other on May 1, 2020, and April 20, 2021, at the seeding rates of 100,000 and 312,500 per hectare, respectively. Each crop was planted using a 4-row seed drill (row spacing: 1 m), with strip sizes of 16 m × 30 m (2020) or 8 m × 40 m (2021). In 2021, corn (Roundup Ready®) was also planted (8 m × 40 m) adjacent to these crops at a seeding rate of 150,000 ha$^{-1}$. The fields were irrigated and fertilized as needed. The crops were grown following the recommended production practices for the region.

In this study, weeds that escaped preemergence and early postemergence herbicide applications were targeted for building and testing models. To this effect, postemergence applications of appropriate herbicides were made in all three crops following standard application procedures, resulting in random escapes at sufficient densities for imaging (**Table 1**). The dominant weed species in the study area were a mix of morningglories (*Ipomoea* spp.) that composed of tall morningglory (*Ipomoea purpurea*) and ivyleaf morningglory (*Ipomoea hederacea*), Texas millet (*Urochloa texana*), and johnsongrass (*Sorghum halepense*). Some other weed species occurred at low frequencies, including Palmer amaranth (*Amaranthus palmeri*), prostrate spurge (*Euphorbia humistrata*), and browntop panicum (*Panicum fasiculatum*). At the time of image collection, these weed species

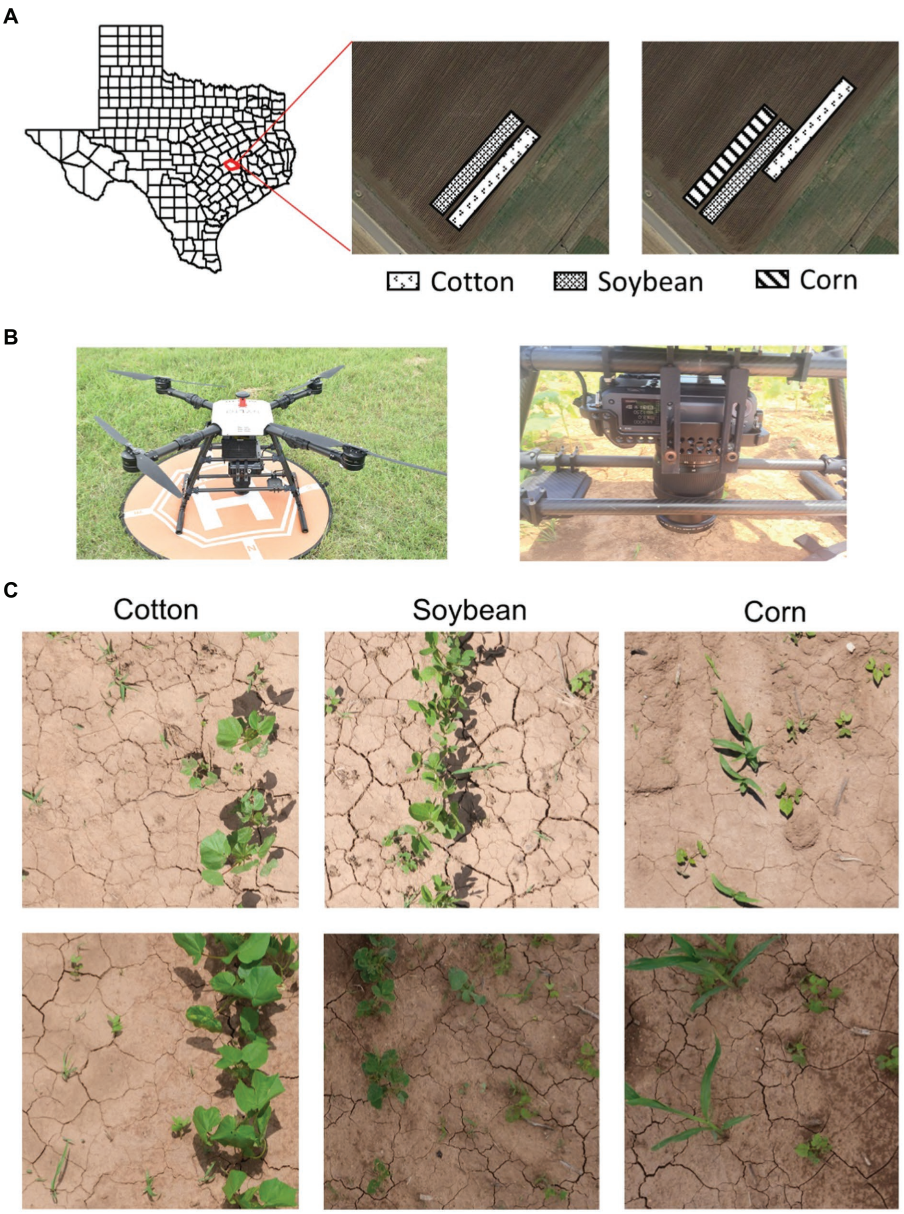


**FIGURE 1 | (A)** Study area (Texas A&M AgriLife Research Farm, Burleson County, TX, United States) and field setup for the 2 experimental years; **(B)** a multi-copter drone (Hylio Inc., Houston, TX, United States) attached with Fujifilm GFX100 (100 MP) camera; and **(C)** image datasets (top and bottom rows) collected under two different environmental conditions for cotton, soybean, and corn.

occurred at different growth stages, from cotyledon to about five true leaves.

## Workflow

The methodological workflow for this study involved three major steps: Data collection and management, model training, and model performance evaluation on different test datasets. See **Figure 2** for a schematic diagram showing the workflow followed in this research. The following sections describe these three steps in more detail.

### High-Resolution Digital Image Collection

A 100-megapixel FUJIFILM GFX100 medium format mirrorless RGB imaging camera was integrated with a multi-copter drone, Hylio AG-110 (Hylio Inc., TX, United States) to capture high-resolution aerial images of the crop fields (**Figure 1**). The images were captured by the drone operating at 4.9 m aboveground level and a speed of 0.61 m/s. The FUJIFILM GF 32–64 mm f/4 R LM WR lens was set at a focal length of 64 mm, shutter speed at 1/4,000 s, ISO at 1250, and f-stop at 8, which resulted in high-quality images with a spatial resolution of 0.274 mm/pixel at the given flying height. Under such configurations, image resolution and quality were sufficient for young grass seedlings to be recognized in the images. However, the wind thrust (i.e., downwash) from the drone operation impacted some plants, causing them to look unreal in the images. They were excluded from the dataset before further analysis. All the images were stored in standard PNG format at 16-bit depth. **Table 1** describes the details of the different image datasets collected in the study.

A total of three flights were made to capture images for all the crops in 2020 and 2021. Two image datasets for each crop (Cotton 1 & Cotton 2, Soybean 1 & Soybean 2, and Corn 1 & Corn 2) were acquired (**Table 1**). For each crop, the second image dataset (e.g., Cotton 2) differed from the first dataset with respect to crop growth stage, weed density, and image acquisition conditions. Cotton 1 was the prime dataset for this study as this consisted of cotton-weed images that were used for building the main model. This dataset was split into training (hereafter referred to as "Train100"), validation, and test datasets. Soybean 1 and Corn 1 datasets were also partitioned similarly to supplement training and validation images to Train100 during cross-applicability improvements later on. All images in Cotton 2, Soybean 2, and Corn 2 were used for testing purposes. Hereafter, these test datasets are referred to as "Cot1," "Cot2," "Soy1," "Soy2," "Corn1," and "Corn2" for respective crops.

### Weed Detection

#### *Image Annotations*

For this study, the images were annotated and recorded in COCO format as this format is inter-changeable to several formats quickly and easily. The VGG image annotator (Dutta and Zisserman, 2019) was used to annotate the weeds with bounding boxes in each image. The annotations were recorded for three categories: morningglories (MG), grasses (Grass), and other weed species (Other). Both Texas millet and johnsongrass seedlings were labeled as "Grass" during annotation as classifying them was not the scope of this study.

#### *Weed Detection in Cotton*

With respect to the first objective, i.e., develop and evaluate models for weed detection in cotton, the detection frameworks were trained with Train100. Train100 comprised of 8,580 annotations altogether, out of which MG, Grass, and Other represented 19.3, 79.5, and 1.2%, respectively (**Table 2**). Two popular object detection frameworks, YOLOv4 and Faster R-CNN, were used in this study. YOLOv4 is the 4th subsequent version

**TABLE 1** | Various datasets used in the study.

| | Image dataset name | Acquisition date | Crop/growth stage | Weed composition/ growth stage | Weed density (plants $m^{-2}$) | Image acquisition conditions | Train/Val/Test [images, annotations] | Annotation composition[a] [MG, Grass, and Other] |
|---|---|---|---|---|---|---|---|---|
| 1 | Cotton 1 (Test data referred to as Cot1) | May 06, 2020 | Cotton: 4–5 leaves | MG: cotyledon-4 leaves<br>JG: 2–3 leaves<br>TM: 2–3 leaves | 18 | Sunny | Train: [460, 8,580]<br>Val: [100, 721]<br>Test: [100, 848] | [19.3, 79.5, 1.2]<br>[22.4, 74.2, 3.4]<br>[51.8, 48.1, 1.1] |
| 2 | Cotton 2 (referred to as Cot2) | June 13, 2021 | Cotton: 2–4 leaves | MG: cotyledon-6 leaves<br>TM: 2–4 leaves | 21 | Partially cloudy | Test: [95, 600] | [36, 63.8, 0.2] |
| 3 | Soybean 1 (Test data referred to as Soy1) | May 06, 2020 | Soybean: 6–7 leaves | MG: cotyledon-4 leaves<br>JG: 2–3 leaves<br>TM: 2–3 leaves | 17 | Sunny | Train: [115, 990]<br>Val: [25, 200]<br>Test: [100, 848] | [46.4, 53.48, 0.07]<br>[48.4, 50.8, 0.8]<br>[54.22, 43.22, 2.56] |
| 4 | Soybean 2 (referred to as Soy2) | May 14, 2021 | Soybean: 1–3 leaves | MG: cotyledon-6 leaves<br>TM: 2–4 true leaves | 21 | Cloudy | Test: [97, 547] | [63.07, 35.4, 1.53] |
| 5 | Corn 1 (Test data referred to as Corn1) | May 07, 2021 | Corn: 2–3 leaves | MG: cotyledon-3 leaves<br>JG: 2–3 leaves<br>TM: 2–3 leaves | 18 | Sunny | Train: [115, 1,010]<br>Val: [25, 215]<br>Test: [100, 890] | [81.16, 16.75, 2.1]<br>[95.2, 4.1, 0.7]<br>[94.62, 4.9, 0.48] |
| 6 | Corn 2 (referred to as Corn2) | May 14, 2021 | Corn: 3–4 leaves | MG: cotyledon-6 leaves<br>TM: 2–4 true leaves | 23 | Cloudy | Test: [95, 559] | [80.5, 17.5, 2] |

*Train, training; Val, validation; MG, morningglories; TM, texas millet; and JG, johnsongrass.*

[a] *The annotations statistics shown within the brackets are given in %.*

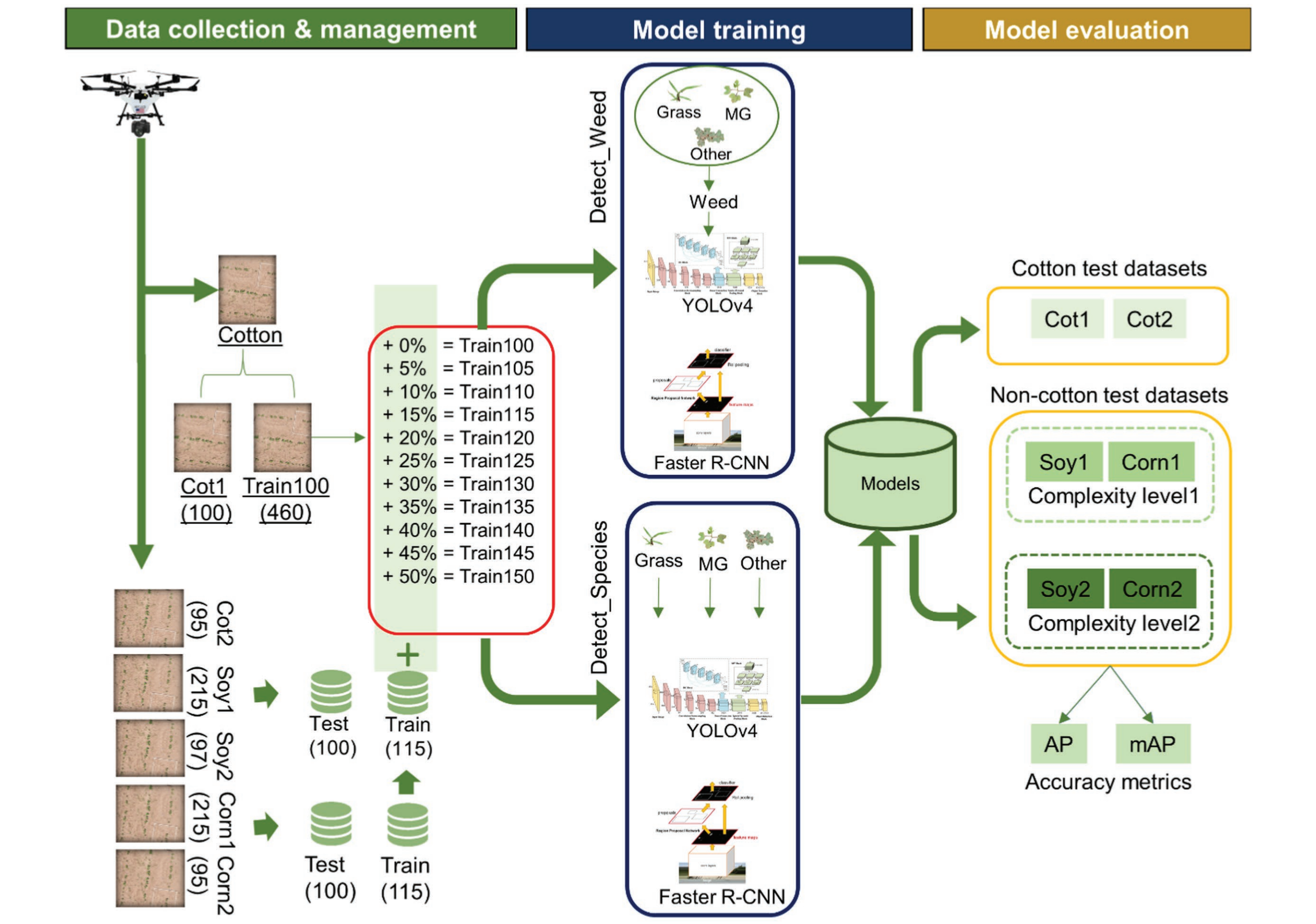


**FIGURE 2 |** Schematic showing the workflow used in the study. The study began with data collection using an UAV and the collected data were distributed for training and test datasets. Data management was followed by model training under two detection schemes: Detect_Weed (detecting at weed/crop level) and Detect_Species (detecting at weed species level). After the models were trained, they were evaluated on the test datasets (Other was excluded during the calculation of accuracy metrics). Average Precision (AP) and Mean Average Precision (mAP) was used as the metrics for performance evaluation.

of the YOLO (Redmon et al., 2016), developed recently by Bochkovskiy et al. (2020). This framework is a one-stage object detector that divides images into several grids and calculates the probabilities that the cell grids belong to a certain class by computing several feature maps. The bounding boxes are then predicted based on grids with the highest probability for the respective classes. The detector sees the entire image during training and inferences for encoding contextual information about classes. Faster R-CNN is the subsequent version of Fast R-CNN (Girshick, 2015) developed by Ren et al. (2017). In contrast to the YOLO frameworks, Faster R-CNN is a two-stage object detector composed of two modules working together. The first module is a Region Proposal Network (RPN) that proposes several candidate regions in the image. The second module is the detector that first extracts features from dense feature maps for the regions selected during RPN and then calculates the confidence score for each region that contains the object of interest (Girshick, 2015).

On-the-fly augmentation of data was carried out for both the frameworks. The "mosaic" augmentation (Bochkovskiy et al., 2020) was enabled for YOLOv4, whereas the "flip and resize" augmentation was performed with the default data loader when training Faster R-CNN. Pre-trained models as provided by the github sources (https://github.com/facebookresearch/detectron2 for Faster R-CNN and https://github.com/AlexeyAB/darknet for YOLOV4 for YOLOv4) were used for model initialization. A mini-batch Stochastic Gradient Descent method was used for model loss optimization for both frameworks. Faster R-CNN was trained for 50,000 iterations whereas YOLOv4 was trained for 6,000 epochs. The definition for *iterations* and *epochs* for these frameworks implies different meanings and are explained in their respective github documentation resource. The model weights were saved after every certain number of iterations or epochs so that the weight resulting in the highest validation accuracy can be chosen at the end for further analysis. Because of the differences in their detection mechanisms, these two frameworks could provide different results for the same detection problem. Hence, evaluation of these two frameworks can provide valuable insights into what level of accuracy can be expected for the given detection problem.

Hereafter, the model trained with Train100 is referred to as the "main cotton model." Two different schemes were designed

**TABLE 2** | Various training datasets evaluated in the study for training YOLOv4 and Faster R-CNN and annotations record for each training dataset.

| Training dataset | Non-cotton images (%)[a] | Annotations | | | |
|---|---|---|---|---|---|
| | | MG (%) | Grass (%) | Other (%) | Total |
| Train100[b] | 0 | 19.3 | 79.5 | 1.20 | 8,580 |
| Train105 | 5 | 20.0 | 78.8 | 1.25 | 8,775 |
| Train110 | 10 | 20.7 | 78.0 | 1.23 | 8,915 |
| Train115 | 15 | 21.7 | 77.1 | 1.21 | 9,072 |
| Train120 | 20 | 22.9 | 75.9 | 1.19 | 9,234 |
| Train125 | 25 | 23.0 | 75.7 | 1.33 | 9,480 |
| Train130 | 30 | 23.9 | 74.7 | 1.32 | 9,689 |
| Train135 | 35 | 24.3 | 74.4 | 1.34 | 9,827 |
| Train140 | 40 | 25.0 | 73.7 | 1.32 | 9,970 |
| Train145 | 45 | 25.5 | 73.2 | 1.31 | 10,113 |
| Train150 | 50 | 25.7 | 73.0 | 1.29 | 10,198 |

*MG-Morningglories; Grass-Grass weeds; and Other-Weeds other than MG and Grass.*
*[a]The numerical figures in this column indicate the percentage of images added to Train100 (i.e., 460 images).*
*[b]Train100 had a total of 460 cotton images and 0 non-cotton images.*
*"Train100" represents the dataset with cotton images only, i.e., no non-cotton images. The last two digits of training dataset names represent the percentage of non-cotton images added to Train100 randomly for building the respective training dataset. The percentage was with respect to Train100.*

for weed detection. In the first scheme, hereafter referred to as "Detect_Weed," frameworks were trained to detect weeds at the meta-level irrespective of the species. The label names for MG, Grass, and Other were merged and labeled as "Weed" while training under this scheme. However, in the second scheme, hereafter referred to as "Detect_Species," frameworks were trained to detect weeds at the species level. For training this scheme, the original annotation dataset that had separate labels for MG, Grass, and Other were used. These schemes have different significance depending on how they are utilized for management. Currently, most of the mechanical platforms for real-time weed control employ "Detect_Weed" scheme for precision control actions (Gai et al., 2020). In most of the existing commercial platforms, detectors are trained to only detect weeds, but not required to classify them at the species level, as the weeds are pulled, zapped, or clipped regardless of species in these platforms. However, selective herbicide spray systems would require detection and classification of individual weeds for species-specific herbicide input. Hence, it may be informative to investigate how these two frameworks behave under these weed detection schemes.

### *Cross-Crop Applicability Analysis*

With respect to the second objective, i.e., assess the scope and prospects for applying the main cotton models to corn and soybean, the performance of the main cotton models was evaluated for each test dataset. In addition, the four non-cotton test datasets (i.e., Soy1, Soy2, Corn1, and Corn2) were grouped into two complexity levels based on their similarity in weed pressure conditions and image acquisition environment. It was assumed that these factors would have more influence than the similarity between crops. Thus, Soy1 and Corn1 were grouped under complexity level 1, while Soy2, and Corn2 under level 2. Cot2 was not grouped under any complexity level, but was rather considered as a replicate of Cot1. In the complexity level 1, the Soy1 and Corn1 differed from the Cotton 1 dataset only for the background crop species, whereas the weed density, growth stages of weeds, and image acquisition conditions were similar. In the complexity level 2, the datasets differed not only for the background crop species, but also for weed density, growth stages of weeds, and light conditions; these differences constitute a higher level of complexity to the weed detection process. Evaluations with these two complexity levels advance our understanding of the model performances under various environments.

### *Cross-Crop Applicability Improvement With Training Size Expansion*

The third objective was to test if supplementing Train100 with additional training images from Soybean 1 and Corn 1 image datasets improves prediction for corn and soybean. As the frameworks were trained to recognize only the weeds and consider crops as part of the background, changes in crop species might confuse the frameworks as to what comprises the background. This confusion intensifies when the frameworks infer upon crop species that were never seen before. Due to this situation, it was assumed that exposing these unseen crops to the frameworks might help boost the confidence score for background. It was more desirable to achieve considerable improvement in the performance with a minimal number of Soybean 1 and Corn 1 images. For this purpose, 10 additional training datasets were prepared by randomly selecting an equal proportion of soybean and corn images and adding them to the main train dataset (i.e., Train100) such that the new dataset size did not exceed 150% of the Train100 size (**Table 2**). Both frameworks were trained independently using 10 different training datasets listed in **Table 2** under the two detection schemes and were validated against test datasets. The same pre-trained models provided by the github source were used for model initialization for each training dataset. Moreover, configurations were also kept the same across training datasets for these two frameworks.

## Accuracy Metrics for Performance Evaluation

The standard performance metric called Mean Average Precision (mAP) was calculated to assess the performance of weed detection under Detect_Species, whereas Average Precision (AP) was used as the performance metric for Detect_Weed. In recent years, these metrics have been frequently used to assess the accuracy of object detection tasks. mAP is a mean of AP calculated for each class to be detected/predicted by the model. AP for each class is calculated as the area under a precision-recall curve. The area is determined in two stages. First, the recall values are evenly segmented to 11 parts starting from 0 to 1. Second, the maximum precision value is measured at each level of recall and averaged to determine AP (Eq. 1).

$$\text{AP} = \frac{1}{11} \sum_{r \in \{0, 0.1, 0.2 \ldots 1\}} p_{\max}(r) \tag{1}$$

where $p_{(\max)}$ represents maximum precision measured at respective recall (r) level.

Precision and recall values are in turn calculated using the Eqs 2, 3, respectively.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

where *TP*, *FP*, and *FN* denote true positive, false positive, and false negative samples, respectively.

True positives, false positives, and false negatives are identified with the help of the Intersection over Union (IoU) ratio. This ratio is calculated by comparing the ground truth box with the model predicted box. If the ratio is above the user-defined threshold, the predicted box is labeled as TP. In this study, the threshold for IoU was set to 0.5. The mAP value ranges between 0 and 1, with 0 indicating null accuracy and 1 indicating perfect accuracy. Only the AP for MG and Grass were averaged to calculate mAP under Detect_Species. AP for Other were found to be very low due to a very small test sample size during the evaluation which led to non-representative mAP values; thus, the accuracy for Other category was excluded during the evaluation process for both frameworks and schemes.

# RESULTS AND DISCUSSION

## Performance of the Main Cotton Model Over Cotton Test Datasets

Two popular object detection frameworks, YOLOv4 and Faster R-CNN were trained to detect weeds in cotton and non-cotton crops. Train100 was used to build two cotton-weed detection models under different detection schemes for each framework. Both YOLOv4 and Faster R-CNN provided reasonably fair accuracy levels under both detection schemes for Cot1 (**Table 3**). Under Detect_Species, AP was higher for MG compared to Grass. Although grasses were visible to naked eyes and also discernible in the images, the model failed to detect a few grass instances. On the contrary, the model led to over-detection (i.e., more plants were predicted than what was present) when these grasses had multiple tillers spread out. Lottes et al. (2018) also observed lower AP for grasses compared to broadleaves when they tested their weed detection model on UAV imageries. However, the opposite was true when they tested on images collected using a ground robot.

When the same models were tested over the second cotton dataset (i.e., Cot2) collected in 2021, the AP & mAP values declined by 12.5 & 14.5% and 11.7 & 22.5% for YOLOv4 and Faster R-CNN, respectively. Unlike Cot1, AP was higher for Grass than for MG for both frameworks under Detect_Species. It should be noted that Cot2 differed from Cot1 in three aspects: (1) Cot2 had a relatively higher density of weeds and the median size of MG and Grass differed from that of Cot1, (2) some of the cotton plants in Cot2 had slightly different visual appearance due to herbicide drift, and (3) the illumination conditions for Cot2 was slightly darker than that of Cot1. Hu et al. (2021) suggested that illumination conditions can affect weed detection accuracy. With respect to herbicide drift impact, Suarez et al. (2017) found in cotton that drift can lead to a significant change in the spectral behavior of the crop. All these reports indicate that morphological, agronomical, and illumination differences can be attributed to the lower accuracy levels observed for Cot2.

Very few studies have looked at weed detection and mapping in cotton. Alchanatis et al. (2005) used rank order algorithms and neighborhood operations to detect broadleaves and grass weeds in cotton. With their approach, 86% of the true weed area was correctly identified, with only 14% misclassified as cotton. Lamm et al. (2002) developed an early growth stage weed control system for cotton. Using morphological analysis such as binarization and erosion, their system was able to correctly identify and spray 88.8% of the weeds. On a different note, both frameworks used in this study have been already used in other weed detection studies. For example, Gao et al. (2020) employed YOLOv4 and Tiny YOLO to detect field bindweed (*Convolvulus sepium*) in sugar beet (*Beta vulgaris*) fields. They used synthetic images in addition to real images to train the framework and obtained an $\text{mAP}_{50}$ value of 0.829 for field bindweed detection. Osorio et al. (2020) used YOLOv3 and other object detection frameworks for weed detection in commercial lettuce crops and obtained an overall accuracy of 89% with YOLOv3. Using the Faster R-CNN framework with the Inception_ResNet-V2 backbone, Le et al. (2020) achieved an $\text{mAP}_{0.50}$ value of 0.55 for detection of wild radish (*Raphanus raphanistrum*) and capeweed (*Arctotheca calendula*) in barley. The overall accuracy obtained in this study for weed detection compares well with reported accuracies by past studies.

**TABLE 3** | Accuracy obtained for various test datasets with YOLOv4 and Faster R-CNN under Detect_Weed and Detect_Species using the main cotton model.

| | Detect_Weed | | Detect_Species | | | | | |
|---|---|---|---|---|---|---|---|---|
| | YOLOv4 | Faster R-CNN | YOLOv4 | | | Faster R-CNN | | |
| | AP | AP | AP (MG) | AP (Grass) | mAP | AP (MG) | AP (Grass) | mAP |
| Cot1 | 0.88 | 0.87 | 0.88 | 0.83 | 0.85 | 0.86 | 0.79 | 0.83 |
| Cot2 | 0.79 | 0.74 | 0.71 | 0.79 | 0.75 | 0.60 | 0.70 | 0.65 |
| Soy1 | 0.83 | 0.76 | 0.83 | 0.75 | 0.79 | 0.72 | 0.70 | 0.71 |
| Soy2 | 0.35 | 0.60 | 0.63 | 0.64 | 0.64 | 0.72 | 0.49 | 0.61 |
| Corn1 | 0.72 | 0.62 | 0.88 | 0.15 | 0.52 | 0.78 | 0.15 | 0.47 |
| Corn2 | 0.33 | 0.39 | 0.65 | 0.15 | 0.40 | 0.54 | 0.03 | 0.29 |

*MG-Morningglories; Grass-Grass weeds; AP, average precision; and mAP, mean average precision. AP and mAP values were computed to assess the performance of the main cotton model over the test datasets. mAP was calculated by averaging AP for MG and Grass. AP was calculated as a function of precision and recall values obtained when Intersection Over Union (IoU) was set to 0.5.*

## Cross-Crop Applicability of Main Cotton Models

The main cotton models were also applied over non-cotton test datasets (i.e., Soy1, Soy2, Corn1, and Corn2) under both detection schemes. The main goal was to see if one crop-based weed detection model can be used to detect the same weeds in other crop species under similar or different agronomic and image acquisition conditions. The detection results by both frameworks for different test datasets under Detect_Weed and Detect_Species are shown in **Figures 3, 4**, respectively for qualitative evaluation. The Detect_Species cotton model performed satisfactorily for Soy1 and Soy2

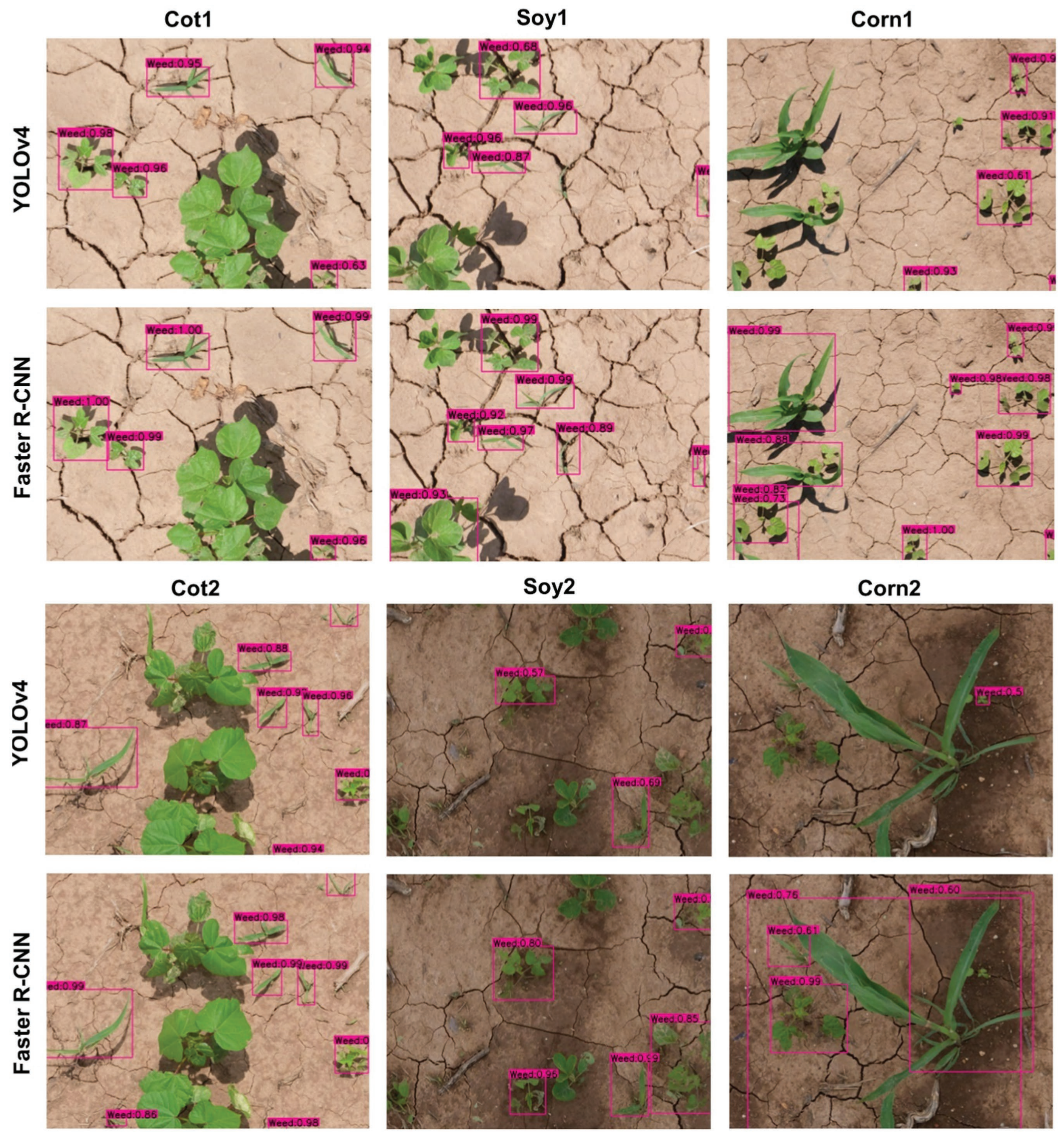


**FIGURE 3 |** Weed detection using bounding boxes by the main cotton models under "Detect_Weed" scheme for various test datasets used in the study. YOLOv4 and Faster R-convolutional neural network (CNN) were trained with the Train100 dataset (i.e., dataset containing cotton images only) to develop the main cotton models. Under this scheme, MG, Grass, and Other were combined into "Weed" category while training the model.

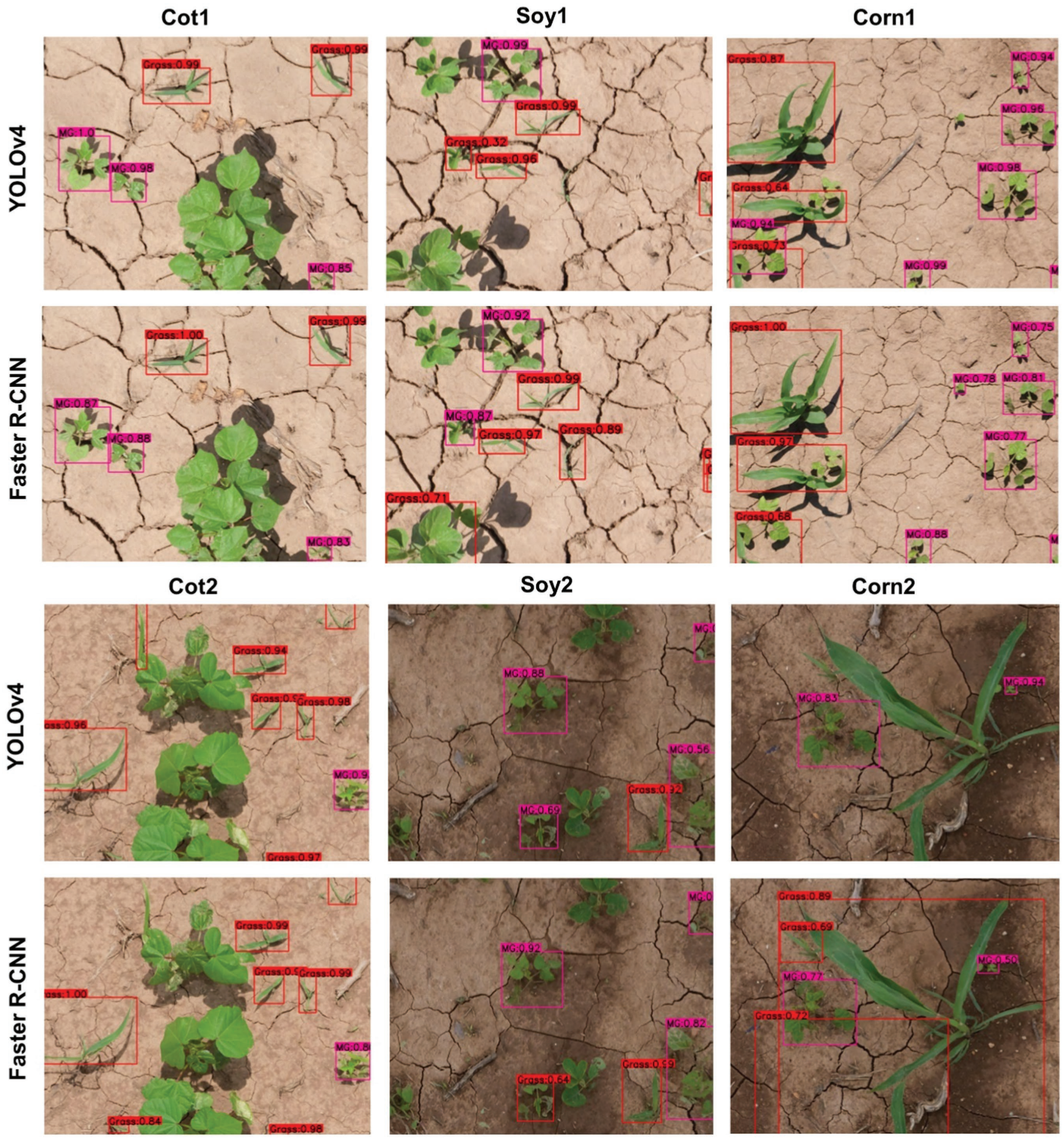


**FIGURE 4** | Bounding boxes generated for MG and Grass by the main cotton models under "Detect_Species" scheme for various test datasets used in the study. YOLOv4 and Faster R-CNN were trained with the Train100 dataset (i.e., dataset containing cotton images only) to develop the main cotton models. Under this scheme, MG, Grass, and Other were trained as separate categories.

datasets, while not so effectively for Corn1 and Corn 2 datasets. The Detect_Weed model performed the same way except that AP was higher for Corn1 but not for Soy2. The significant difference in performance between Faster R-CNN and YOLOv4 for Soy2 under Detect_Weed is notable. In this regard, YOLOv4 predictions on Soy2 images were further investigated. Several MG were not detected by the model, resulting in many false negatives. AP/mAP for non-cotton test datasets was not better than that of Cot1 for both frameworks. Among non-cotton test datasets, the highest

AP/mAP was obtained for Soy1 for both frameworks (**Table 3**). Further, in general, the model performed relatively better on complexity level 1 than level 2 (**Figure 5**). The difference in performance was more obvious under Detect_Weed for both frameworks.

It was notable that Soy1 yielded higher AP/mAP values than Cot2 for both frameworks under both schemes. The authors could think of two reasons for this outcome: Soy1 had similar weed density and sizes to that of Train100; further, Soy1 and Train100 datasets were acquired at the same time, and hence illumination conditions were exactly the same. Here, higher accuracy for Soy1 suggests that illumination conditions and weed density can impose more influence on the detection accuracy. In general, higher accuracies were obtained for soybean datasets compared to corn datasets. The main reason could be the confusion between Grass and corn plants. A few instances of corn plants were detected as Grass by the model as they looked similar during early growth stages. Such misclassification was also observed when corn was distinctively larger than grasses. This suggests that the model may have focused more on the canopy structure than canopy size. Further, the detection performances between complexity levels were in line with our expectations. The primary reason for higher accuracy with complexity level 1 was the similar illumination conditions and weed density to the training dataset, i.e., Train100 as compared to the level 2 test datasets.

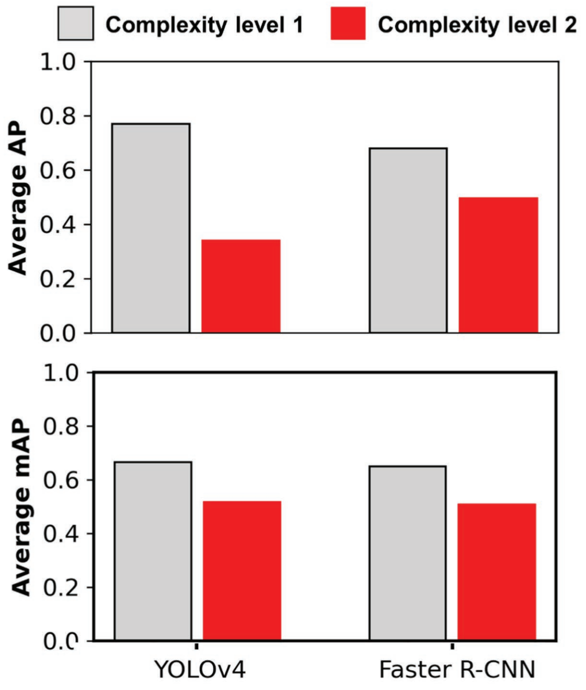


**FIGURE 5** | Average Precision and mAP achieved for different complexity level datasets with main cotton models. Complexity level 1 datasets include Soy1 and Corn1, whereas level 2 include Soy2 and Corn2. The main cotton models were derived by training the detection frameworks (YOLOv4 and Faster R-CNN) with Train100 (i.e., dataset containing cotton images only). The AP/mAP for datasets under each complexity level were averaged to derive average AP and mAP.

## Cross-Crop Applicability Improvement With Additional Non-cotton Image Datasets

Train100 was supplemented with different amounts of training images from Soy1 and Corn1 to generate various training datasets. These datasets were used to train new models under two detection schemes and finally, the built models were tested over cotton and non-cotton test datasets. Both frameworks showed general increments in accuracy with the addition of non-cotton crop images under both detection schemes (**Figure 6**). The rate of increment, however, varied across test datasets, frameworks, and detection schemes (**Table 4**). The trend was relatively smoother for Faster R-CNN compared to YOLOv4 for all test datasets. The increment was the highest for Corn2 and the lowest for either of the cotton test datasets for both frameworks and detection schemes. AP/mAP for test datasets under each complexity level were averaged along with Cot1 values to calculate average AP/mAP (**Figure 7**). The trend was smoother for Faster R-CNN compared to YOLOv4 for all complexity levels.

## Scope and Limitations of the Study

Cross-crop applicability assessments conducted in this study provides useful insights into how models can be generalized for broad application. Such generalization could save enormous efforts and resources and help make rapid progress toward effective site-specific weed management. Cross-applicability has become an absolute necessity owing to the huge data requirements by the CNN models for a given crop-weed environment. Often, a significant amount of data resources is used to train a weed detection model for a single crop environment. For example, Yu et al. (2019) used a total of 29,000 images to train a model that could detect multiple weeds in perennial ryegrass. Czymmek et al. (2019) trained a model to detect weeds in organic carrot farms using 2,500 images. It is increasingly important to focus on how these data resources can be exploited strategically for maximizing efficiency and productivity. By testing the approach of data supplementation, this study demonstrated that cross-crop applicability can be improved with such tactics.

It should be noted that this study evaluated CNN model cross-applicability for crops that had similar weed compositions. The cross-crop applicability findings from this study do not apply to crops differing in weed species composition. In other words, the models would fail to perform if applied over soybean and corn infested with other weed species. A single crop-based model may not be effectively applied at regional scales where weed composition differs. Furthermore, not all the hyperparameters for both frameworks used in the study were tuned, but rather used as defaults in the settings. The reported accuracies may change if parameters are tuned.

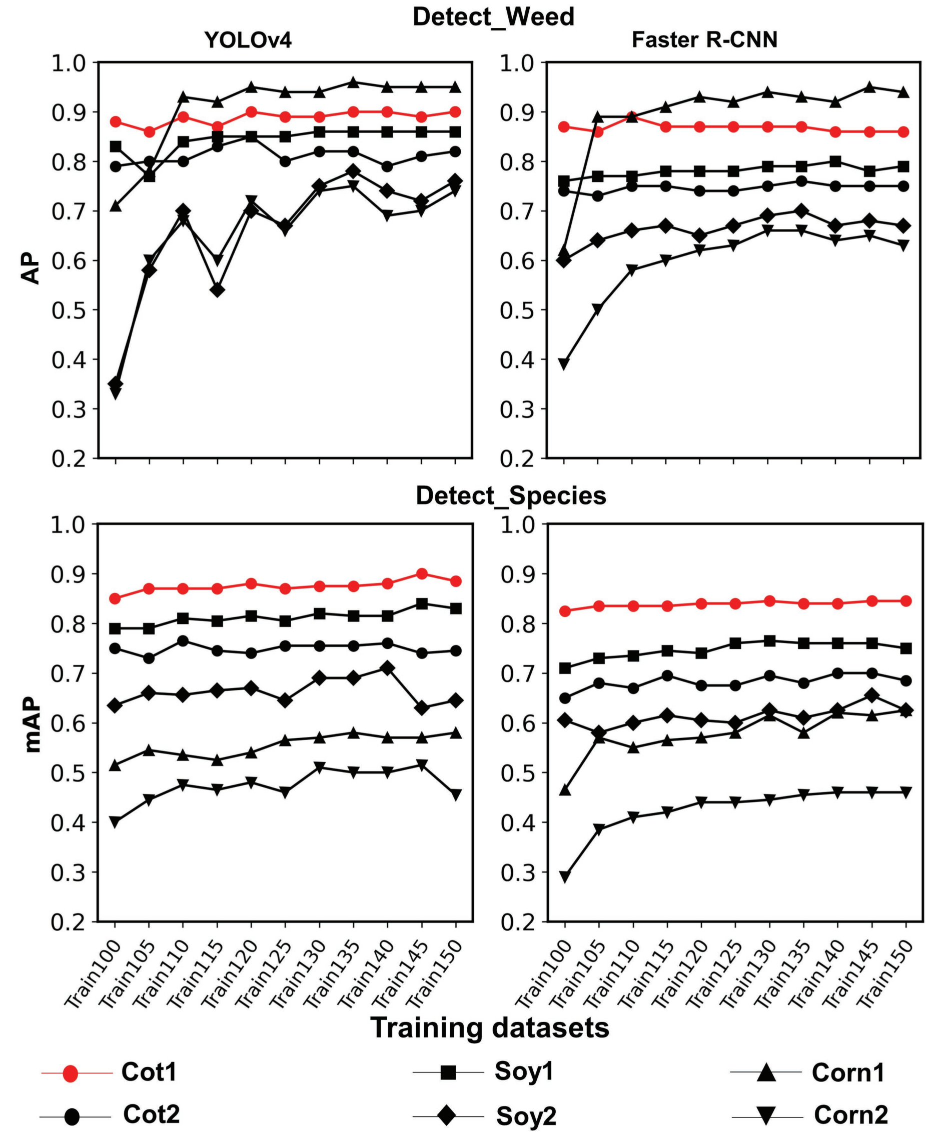


**FIGURE 6 |** Line plots showing AP and mAP achieved with various training datasets for each test dataset used in the study for both frameworks and detection schemes. Various training datasets were created by adding Soybean 1 and Corn 1 training images to the original dataset, i.e., Train100. These non-cotton crop images were added 5% at a time until they amounted to 50% of Train100. The last two digits in the training dataset name denote the % of images added to Train100.

**TABLE 4 |** The maximum rate of increment in accuracy for various test datasets with the addition of non-cotton images.

| | Detect_Weed (AP%) | | Detect_Species (mAP%) | |
|---|---|---|---|---|
| | YOLOv4 | Faster R-CNN | YOLOv4 | Faster R-CNN |
| Cot1 | 2.27 | 2.29 | 5.89 | 2.42 |
| Cot2 | 7.60 | 2.70 | 2.00 | 7.70 |
| Soy1 | 3.61 | 5.26 | 6.32 | 7.74 |
| Soy2 | 122.8 | 16.00 | 11.90 | 8.27 |
| Corn1 | 31.9 | 53.22 | 12.62 | 34.40 |
| Corn2 | 127.27 | 69.23 | 28.75 | 58.62 |

*AP, average precision; mAP, mean average precision. The rate was determined by subtracting the accuracy obtained with Train100 (no non-cotton images) from the highest accuracy obtained among all training datasets for the respective test dataset.*

## CONCLUSION

The study explored two popular object detection frameworks under two useful detection schemes for weed detection in cotton. The study also evaluated the feasibility of cross-crop applicability of the cotton model and experimented with several amounts of non-cotton images to improve cross-applicability. Based on the results, the following main conclusions could be derived:

a. The cotton model achieved reasonably high weed detection accuracy for cotton test datasets.
b. The cotton model achieved a fair level of accuracy for non-cotton crops infested with similar weed compositions.

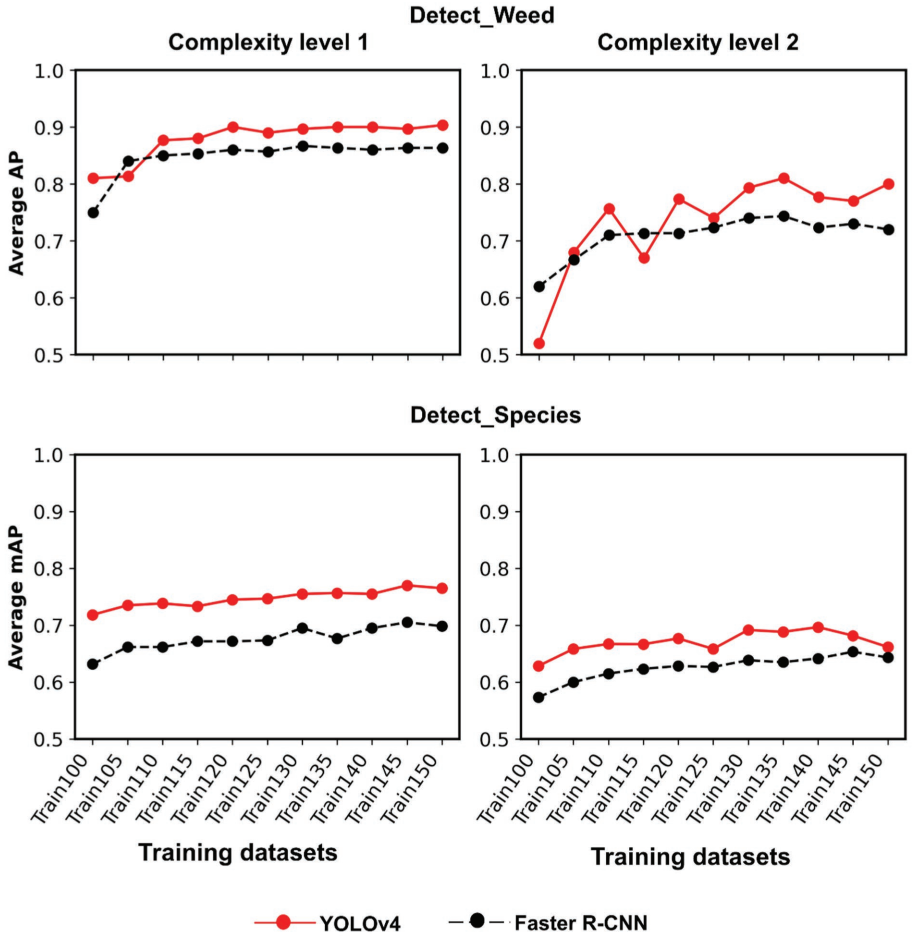


**FIGURE 7 |** Line plots showing AP and mAP achieved for each complexity level with YOLOv4 and Faster R-CNN. Complexity level 1 datasets include Soy1 and Corn1, whereas level 2 include Soy2 and Corn2. AP and mAP for Cot1 dataset were also included in the averaging process of each complexity level to understand how well the models perform with both cotton and non-cotton datasets.

On average, the performance was better for soybean than for corn.

c. The cross-crop applicability was improved (AP/mAP: +3.61 to 127.27%) when Train100 was supplemented with non-cotton images.

The outcomes of this study are expected to advance our understanding of cross-crop applicability of weed detection models. Such understanding will guide our efforts toward optimal use of data resources and accelerate weed detection, mapping, and site-specific management in agricultural systems. In the future, CNN model cross-applicability will be assessed for additional crops and different levels of complexities.

## DATA AVAILABILITY STATEMENT

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

## AUTHOR CONTRIBUTIONS

MB and BS: conceptualization and experimental design. MB: funding acquisition, supervision, and project management. BS and CH: field data acquisition and analysis. BS: writing the first draft of the paper. MB, BS, and CH: paper editing and revisions. All authors contributed to the article and approved the submitted version.

## FUNDING

This study was funded in part by the USDA-Natural Resources Conservation Service-Conservation Innovation Grant (NRCS-CIG) program (award #NR213A750013G017) and Cotton Incorporated (award #20-739).

## ACKNOWLEDGMENTS

The authors acknowledge field technical support by Daniel Hathcoat and Daniel Lavy.

## REFERENCES

Abdalla, A., Cen, H., Wan, L., Rashid, R., Weng, H., Zhou, W., et al. (2019). Fine-tuning convolutional neural network with transfer learning for semantic segmentation of ground-level oilseed rape images in a field with high weed pressure. *Comput. Electron. Agric.* 167:105091. doi: 10.1016/j.compag.2019.105091

Adhikari, S. P., Yang, H., and Kim, H. (2019). Learning semantic graphics using convolutional encoder–decoder network for autonomous weeding in paddy. *Front. Plant Sci.* 10:1404. doi: 10.3389/fpls.2019.01404

Ahmad, F., Qiu, B., Dong, X., Ma, J., Huang, X., Ahmed, S., et al. (2020). Effect of operational parameters of UAV sprayer on spray deposition pattern in target and off-target zones during outer field weed control application. *Comput. Electron. Agric.* 172:105350. doi: 10.1016/j.compag.2020.105350

Ahmed, F., Al-Mamun, H. A., Bari, A. S. M. H., Hossain, E., and Kwan, P. (2012). Classification of crops and weeds from digital images: a support vector machine approach. *Crop Prot.* 40, 98–104. doi: 10.1016/j.cropro.2012.04.024

Alchanatis, V., Ridel, L., Hetzroni, A., and Yaroslavsky, L. (2005). Weed detection in multi-spectral images of cotton fields. *Comput. Electron. Agric.* 47, 243–260. doi: 10.1016/j.compag.2004.11.019

Aravind, R., Daman, M., and Kariyappa, B. S. (2015). "Design and development of automatic weed detection and smart herbicide sprayer robot," in *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*. 257–261.

Bagavathiannan, M. V., and Davis, A. S. (2018). An ecological perspective on managing weeds during the great selection for herbicide resistance. *Pest Manag. Sci.* 74, 2277–2286. doi: 10.1002/ps.4920

Beckie, H. J., Ashworth, M. B., and Flower, K. C. (2019). Herbicide resistance management: recent developments and trends. *Plants* 8:161. doi: 10.3390/plants8060161

Berge, T. W., Goldberg, S., Kaspersen, K., and Netland, J. (2012). Towards machine vision based site-specific weed management in cereals. *Comput. Electron. Agric.* 81, 79–86. doi: 10.1016/j.compag.2011.11.004

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

Buchanan, G. A., and Burns, E. R. (1970). Influence of weed competition on cotton. *Weed Sci.* 18, 149–154. doi: 10.1017/S0043174500077560

Czymmek, V., Harders, L.O., Knoll, F.J., and Hussmann, S. (2019). "Vision-based deep learning approach for real-time detection of weeds in organic farming." in *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. 1–5.

Dutta, A., and Zisserman, A. (2019). "The VIA annotation software for images, audio and video." in *Proceedings of the 27th ACM International Conference on Multimedia* (MM '19); October 21-25 (New York, NY, USA), 2276–2279.

Fawakherji, M., Youssef, A., Bloisi, D., Pretto, A., and Nardi, D. (2019). "Crop and weeds classification for precision agriculture using context-independent pixel-wise segmentation." in *2019 Third IEEE International Conference on Robotic Computing (IRC)*. 146–152.

Gai, J., Tang, L., and Steward, B. L. (2020). Automated crop plant detection based on the fusion of color and depth images for robotic weed control. *J. Field Robot.* 37, 35–52. doi: 10.1002/rob.21897

Gao, J., French, A. P., Pound, M. P., He, Y., Pridmore, T. P., and Pieters, J. G. (2020). Deep convolutional neural networks for image-based *Convolvulus sepium* detection in sugar beet fields. *Plant Methods* 16:29. doi: 10.1186/s13007-020-00570-z

García-Santillán, I. D., and Pajares, G. (2018). On-line crop/weed discrimination through the Mahalanobis distance from images in maize fields. *Biosyst. Eng.* 166, 28–43. doi: 10.1016/j.biosystemseng.2017.11.003

Girshick, R. (2015). "Fast R-CNN." in *Proceedings of the IEEE International Conference on Computer Vision*. 1440–1448.

Hu, C., Sapkota, B. B., Thomasson, J. A., and Bagavathiannan, M. V. (2021). Influence of image quality and light consistency on the performance of convolutional neural networks for weed mapping. *Remote Sens.* 13:2140. doi: 10.3390/rs13112140

Jiang, H., Zhang, C., Qiao, Y., Zhang, Z., Zhang, W., and Song, C. (2020). CNN feature based graph convolutional network for weed and crop recognition in smart farming. *Comput. Electron. Agric.* 174:105450. doi: 10.1016/j.compag.2020.105450

Kargar, B. A. H., and Shirzadifar, A. M. (2013). "Automatic weed detection system and smart herbicide sprayer robot for corn fields." in *2013 First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*; February 13-15, 2003; Tehran, Iran, 468–473.

Lamm, R. D., Slaughter, D. C., and Giles, D. K. (2002). Precision weed control system for cotton. *Transact. ASAE* 45:231. doi: 10.13031/2013.7861

Le, V. N. T., Ahderom, S., and Alameh, K. (2020). Performances of the LBP based algorithm over CNN models for detecting crops and weeds with similar morphologies. *Sensors* 20:2193. doi: 10.3390/s20082193

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

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, Lecture Notes in Computer Science*. eds. B. Leibe, J. Matas, N. Sebe and M. Welling (Cham: Springer International Publishing), 21–37.

Liu, B., and Bruch, R. (2020). Weed detection for selective spraying: a review. *Curr. Robot. Rep.* 1, 19–26. doi: 10.1007/s43154-020-00001-w

López-Granados, F. (2011). Weed detection for site-specific weed management: mapping and real-time approaches. *Weed Res.* 51, 1–11. doi: 10.1111/j.1365-3180.2010.00829.x

Lottes, P., Behley, J., Chebrolu, N., Milioto, A., and Stachniss, C. (2018). "Joint stem detection and crop-weed classification for plant-specific treatment in precision farming." in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 8233–8238.

Lottes, P., Behley, J., Chebrolu, N., Milioto, A., and Stachniss, C. (2019). Robust joint stem detection and crop-weed classification using image sequences for plant-specific treatment in precision farming. *J. Field Robot.* 37, 20–34. doi: 10.1002/rob.21901

Ma, X., Deng, X., Qi, L., Jiang, Y., Li, H., Wang, Y., et al. (2019). Fully convolutional network for rice seedling and weed image segmentation at the seedling stage in paddy fields. *PLoS One* 14:e0215676. doi: 10.1371/journal.pone.0215676

Machleb, J., Peteinatos, G. G., Kollenda, B. L., Andújar, D., and Gerhards, R. (2020). Sensor-based mechanical weed control: present state and prospects. *Comput. Electron. Agric.* 176:105638. doi: 10.1016/j.compag.2020.105638

Martin, D., Singh, V., Latheef, M. A., and Bagavathiannan, M. (2020). Spray deposition on weeds (Palmer amaranth and Morningglory) from a remotely piloted aerial application system and packpack sprayer. *Drones* 4:59. doi: 10.3390/drones4030059

Nave, W. R., and Wax, L. M. (1971). Effect of weeds on soybean yield and harvesting efficiency. *Weed Sci.* 19, 533–535. doi: 10.1017/S0043174500050608

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). "Learning and transferring mid-level image representations using convolutional neural networks." in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1717–1724.

Osorio, K., Puerto, A., Pedraza, C., Jamaica, D., and Rodríguez, L. (2020). A deep learning approach for weed detection in lettuce crops using multispectral images. *AgriEngineering* 2, 471–488. doi: 10.3390/agriengineering2030032

Partel, V., Kakarla, S. C., and Ampatzidis, Y. (2019). Development and evaluation of a low-cost and smart technology for precision weed management utilizing artificial intelligence. *Comput. Electron. Agric.* 157, 339–350. doi: 10.1016/j.compag.2018.12.048

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*. 779–788.

Redmon, J., and Farhadi, A. (2018). YOLOv3: an incremental improvement. arXiv [Preprint]. doi: 10.48550/arXiv.1804.02767

Ren, S., He, K., Girshick, R., and Sun, J. (2017). 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

Rumpf, T., Römer, C., Weis, M., Sökefeld, M., Gerhards, R., and Plümer, L. (2012). Sequential support vector machine classification for small-grain weed species discrimination with special regard to *Cirsium arvense* and *Galium aparine*. *Comput. Electron. Agric.* 80, 89–96. doi: 10.1016/j.compag.2011.10.018

Sabzi, S., Abbaspour-Gilandeh, Y., and García-Mateos, G. (2018). A fast and accurate expert system for weed identification in potato crops using metaheuristic algorithms. *Comput. Ind.* 98, 80–89. doi: 10.1016/j.compind.2018.03.001

Sapkota, B., Singh, V., Neely, C., Rajan, N., and Bagavathiannan, M. (2020). Detection of Italian ryegrass in wheat and prediction of competitive interactions using remote-sensing and machine-learning techniques. *Remote Sens.* 12:2977. doi: 10.3390/rs12182977

Sharpe, S. M., Schumann, A. W., and Boyd, N. S. (2020). Goosegrass detection in strawberry and tomato using a convolutional neural network. *Sci. Rep.* 10:9548. doi: 10.1038/s41598-020-66505-9

Suarez, L. A., Apan, A., and Werth, J. (2017). Detection of phenoxy herbicide dosage in cotton crops through the analysis of hyperspectral data. *Int. J. Remote Sens.* 38, 6528–6553. doi: 10.1080/01431161.2017.1362128

Sujaritha, M., Annadurai, S., Satheeshkumar, J., Kowshik Sharan, S., and Mahesh, L. (2017). Weed detecting robot in sugarcane fields using fuzzy real time classifier. *Comput. Electron. Agric.* 134, 160–171. doi: 10.1016/j.compag.2017.01.008

Wu, X., Xu, W., Song, Y., and Cai, M. (2011). A detection method of weed in wheat field on machine vision. *Procedia Engin.* 15, 1998–2003. doi: 10.1016/j.proeng.2011.08.373

Xie, S., Hu, C., Bagavathiannan, M., and Song, D. (2021). Toward robotic weed control: detection of nutsedge weed in bermudagrass turf using inaccurate and insufficient training data. *IEEE Robot. Automat. Lett.* 6, 7365–7372. doi: 10.1109/LRA.2021.3098012

Yu, J., Schumann, A. W., Cao, Z., Sharpe, S. M., and Boyd, N. S. (2019). Weed detection in perennial ryegrass with deep learning convolutional neural network. *Front. Plant Sci.* 10:1422. doi: 10.3389/fpls.2019.01422

**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.

frontiers | Frontiers in Plant Science

ORIGINAL RESEARCH
published: 28 April 2022
doi: 10.3389/fpls.2022.864486

# Deep Learning-Based Identification of Maize Leaf Diseases Is Improved by an Attention Mechanism: Self-Attention

*Xiufeng Qian*[1,2,3], *Chengqi Zhang*[4], *Li Chen*[4] *and Ke Li*[1,2,3]*

*[1] School of Information and Computer, Anhui Agricultural University, Hefei, China, [2] Anhui Provincial Engineering Laboratory for Beidou Precision Agriculture Information, Anhui Agricultural University, Hefei, China, [3] Information Materials and Intelligent Sensing Laboratory of Anhui Province, Anhui University, Hefei, China, [4] School of Plant Protection, Anhui Agricultural University, Hefei, China*

Maize leaf diseases significantly reduce maize yield; therefore, monitoring and identifying the diseases during the growing season are crucial. Some of the current studies are based on images with simple backgrounds, and the realistic field settings are full of background noise, making this task challenging. We collected low-cost red, green, and blue (RGB) images from our experimental fields and public dataset, and they contain a total of four categories, namely, southern corn leaf blight (SCLB), gray leaf spot (GLS), southern corn rust (SR), and healthy (H). This article proposes a model different from convolutional neural networks (CNNs) based on transformer and self-attention. It represents visual information of local regions of images by tokens, calculates the correlation (called attention) of information between local regions with an attention mechanism, and finally integrates global information to make the classification. The results show that our model achieves the best performance compared to five mainstream CNNs at a meager computational cost, and the attention mechanism plays an extremely important role. The disease lesions information was effectively emphasized, and the background noise was suppressed. The proposed model is more suitable for fine-grained maize leaf disease identification in a complex background, and we demonstrated this idea from three perspectives, namely, theoretical, experimental, and visualization.

**Keywords: crop disease, machine learning, deep learning, attention mechanism, neural network**

OPEN ACCESS

***Edited by:***
*Valerio Giuffrida,*
*Edinburgh Napier University,*
*United Kingdom*

***Reviewed by:***
*Marcin Wozniak,*
*Silesian University of Technology,*
*Poland*
*Huaming Chen,*
*The University of Adelaide, Australia*

****Correspondence:***
*Ke Li*
*like@ahau.edu.cn*

***Specialty section:***
*This article was submitted to*
*Technical Advances in Plant Science,*
*a section of the journal*
*Frontiers in Plant Science*

***Received:*** *28 January 2022*
***Accepted:*** *28 March 2022*
***Published:*** *28 April 2022*

***Citation:***
*Qian X, Zhang C, Chen L and Li K*
*(2022) Deep Learning-Based*
*Identification of Maize Leaf Diseases*
*Is Improved by an Attention*
*Mechanism: Self-Attention.*
*Front. Plant Sci. 13:864486.*
*doi: 10.3389/fpls.2022.864486*

## INTRODUCTION

Maize is one of the most important crops for humanity, with the highest yield globally (Ranum et al., 2014). Maize diseases can cause severe yield reductions, a critical problem (Savary et al., 2012). Therefore, it is vital to promptly identify and monitor maize diseases during the growing period. Accurate identification of diseases in maize is difficult for crop growers who may not be professional in plant pathology, and expert identification is expensive and time-consuming (Ouppaphan, 2017). Traditional image recognition methods and deep learning are gradually entering the field of plant disease recognition (Saleem et al., 2019).

Mobile terminals based on web services and support vector machine (SVM) as back-end algorithms can automatically identify maize diseases (Zhang and Yang, 2014). Zhang et al. (2015) proposed an improved genetic algorithm-SVM (GA-SVM) algorithm to improve the accuracy. A recent study on maize disease identification compared five standard machine learning methods (Panigrahi et al., 2020), namely, Naive Bayes (NB), Decision Tree (DT), K-Nearest Neighbor (KNN), SVM, and Random Forest (RF), with RF achieving the highest accuracy of 79.23%.

However, traditional machine learning is mainly limited by feature extraction and feature representation. Deep learning has made significant progress in plant disease identification (Liu and Wang, 2021). Since AlexNet was proposed in 2012 (Krizhevsky et al., 2012), convolutional neural networks (CNNs) have been widely used in academia and industry, e.g., face detection in dangerous situations (Wieczorek et al., 2021) and combination of Internet of things (IoT) and pearl millet disease prediction (Kundu et al., 2021). In the field of plant disease identification, Dhaka et al. (2021) provided a systematic review of relevant deep learning techniques. Due to its low complexity, a lightweight CNN for mobile terminals has achieved satisfactory performance in maize disease identification (Ouppaphan, 2017). A CNN-based system (DeChant et al., 2017) was implemented to automatically identify northern leaf blight, addressing the challenges of limited data and various irregularities appearing in field-grown images. Ahila Priyadharshini et al. (2019) proposed a CNN modified from LeNet for identifying four maize categories (three diseases classes and one health class) with an accuracy of 97.89%.

However, most of the current studies are based on simple background maize leaf or other crop disease recognition, and the recognition effect of the trained models deteriorates in real field settings, because background noise information causes serious obstruction (Lv et al., 2020). Current research on popular or novel deep learning image recognition algorithms (CNNs) is mainly tested on the public dataset ImageNet, and its images are different from fine-grained images of crop disease. Those designed CNNs mostly focus on patterns of objects in images (e.g., profile features of dogs or cats), and these pattern features are reflected in feature maps of convolutional output, as can be demonstrated by numerous neural network visualization studies (Chattopadhay et al., 2018; Chen et al., 2020; Jiang et al., 2021). In contrast, fine-grained crop disease lesions are usually similar and discrete on the leaf surface; thus, CNNs may not be fully adapted to fine-grained maize leaf disease image classification tasks, which will result in no increase in model performance even by stacking the network layers and increasing model parameters. Rational model design for specific tasks is important and necessary, and the following analysis and experiments in this article also prove this perspective. In addition, many visual disturbances (e.g., reflection, dispersion, and blur) seriously affect fine-grained image classification (Lu Y. et al., 2017; Yang et al., 2020). Therefore, fine-grained maize disease identification in complex background field settings requires more rational models and computerized mechanisms.

Mutual attention between words is highly essential for machine translation tasks, which determines whether a sentence can be translated accurately. The transformer architecture (Vaswani et al., 2017) with the attention mechanism has achieved significant success in natural language processing (NLP). Although previous attention mechanisms have been applied to some specific tasks, e.g., image caption generation technology (Lu J. et al., 2017), text classification (Li et al., 2019), and human action recognition (Song et al., 2017), the form and principle of their attention mechanisms are too different and specialized. However, the transformer's attention mechanism (self-attention) has a universal form.

To explore whether the attention mechanism will bring enhancements to the field of computer vision, vision transformer (ViT, **Figure 1** depicts it) (Dosovitskiy et al., 2020) applies the transformer architecture directly to image classification tasks for the first time, outperforming the state of the art on large-scale datasets. Subsequently, researchers gradually began to study ViT and its attention mechanism. Transformer in transformer (TNT) (Han et al., 2021) embeds the inner transformer into the outer transformer to improve the feature extraction capability lacking in the patch embedding method (refer to **Figure 1** for the patch embedding method). Compact convolutional transformer (CCT) (Hassani et al., 2021) demonstrates that convolution can be used to extract local information better, thus making it possible to apply transformer to more tasks with small datasets.

In this study, we found that transformer and self-attention computer mechanisms are more suitable for maize leaf disease identification in complex backgrounds. This article will demonstrate their efficiency and why they work from three perspectives, namely, theoretical derivation, experiment, and visualization. We collected maize leaf diseases datasets with complex backgrounds in our experimental field and proposed an improved model based on ViT and CCT to classify maize into four categories (**Figure 2**), namely, healthy (H), southern corn leaf blight (SCLB) (Aregbesola et al., 2020), gray leaf spot (GLS) (Saito et al., 2018), and southern corn rust (SR) (Wang S. et al., 2019). The model outperforms some mainstream CNNs compared with it in all metrics, with a smaller number of parameters. In addition, we also conducted experiments on the necessity of the self-attention for the model, demonstrating that it is essential. This article also conducts experiments to observe the effect of the ratio of train set to validation set on the accuracy of the model.

The rest of this article is organized as follows. The section "Materials and Methods" introduces the details of our experimental field and experimental sample cultivation, describing our datasets and methods used to collect them. In that section, we focused on describing our algorithm and the detailed theoretical derivation and proving its effectiveness, as well as the experimental visualization schemes (three schemes). All experimental results are described in section "Results." We discussed the reasons for the efficiency of this model and some possible future extensions in section "Discussion."

## MATERIALS AND METHODS

### Data Collection and Preparation

The dataset of the images, which included 7,701 images, consists of two parts, namely, one part is collected from the public dataset Plant Village and the other part is taken by mobile phones in the natural environment of our experimental field. The maize plants grown in the experimental field are used to select suitable

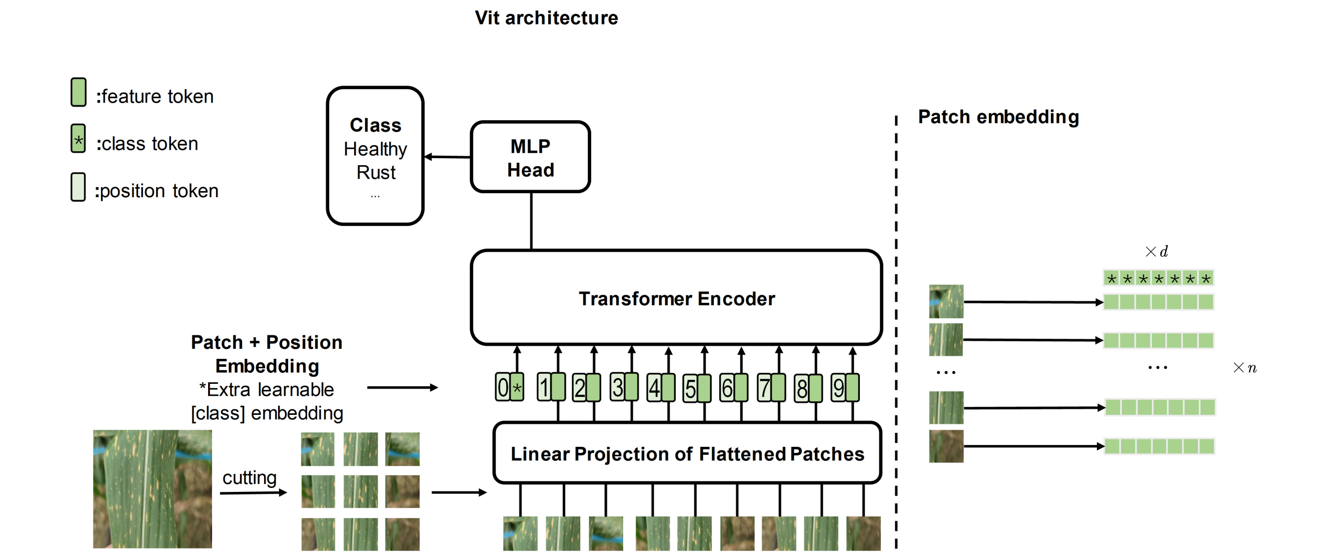


**FIGURE 1 |** The left side of the figure is the original vision transformer architecture, and the illustration is inspired by Dosovitskiy et al. (2020). The right side of the figure is the patch embedding method which cuts the image into several patches.

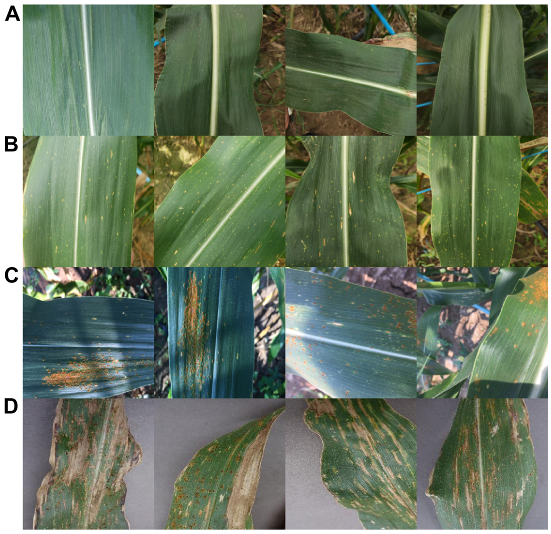


**FIGURE 2 |** Showing four categories of samples of maize. **(A)** H. **(B)** SCLB. **(C)** SR. **(D)** GLS.

disease-resistant varieties, so there are numerous maize varieties. However, as with other studies on maize disease identification, the variety of maize is not the focus and has no impact on the study of this article because the images of maize leaves in our dataset do not reflect their genetic variety. An area of the experimental field covered 3 acres was chosen for this study, planting a total of 80 rows of maize with 26 maize plants per row, 65 cm between rows, and 13 m length of each row. Half of this area was planted with maize inoculated with SCLB, and the other half with maize inoculated with SR. The conidia with a concentration of $10^6$/ml were sprayed at this maize in the sixth-leaf stage, namely, 40–50 days after sowing, to inoculate maize with the abovementioned diseases. After inoculation, the maize is allowed to develop naturally. One day of the milk stage of the maize is chosen to take all the images needed for our dataset. Every maize plant is sampling points. We walked along the rows and remained for several minutes to take images, and the same leaf will be photographed more than once to get 1–6 images. Furthermore, the leaves were manually moved to find a better angle to photograph a good image while adjusting the position of the phones to aid this operation. Despite the fact that a leaf may be photographed more than once, every image is different and contains complicated background visual information because the content of interest is different for each shot. The manual focus is chosen to solve the issue that phones cannot focus on the leaf lesion areas of interest, therefore, guaranteeing every image is clear and focused. The H maize images were obtained from another area of the experimental field where eight rows of maize plants were planted, and the planting pattern and the photographing mode are identical to the above. All the images photographed are under normal uncontrolled lighting conditions with mobile phones' low-cost red, green, and blue (RGB) sensor. The GLS maize images are downloaded from the Plant Village. This article divided the dataset into a training dataset and a validation dataset according to the principles of 3 to 1 due to the sample balance. **Table 1** shows the distribution of images and the division of the dataset.

## Data Processing

The images' size must be unified to a standard 224 × 224-pixel square offline to reduce the computational effort before the model training. Furthermore, some data augmentation techniques are separately applied to each image, with a certain probability

**TABLE 1 |** Distribution of data sources and division of training set and validation set.

| Categories | Shooting by us | Plant village | Train set | Validation set |
|---|---|---|---|---|
| SCLB | 2,243 | 0 | 1,743 | 500 |
| H | 1,273 | 1,162 | 1,953 | 500 |
| SR | 2,023 | 0 | 1,523 | 500 |
| GLS | 0 | 1,000 | 750 | 250 |

during model training online, thus enhancing the generalization ability of the model and preventing its overfitting. This article selects four data augmentation techniques suitable for maize leaf disease identification, namely, RandomFlip, ColorJitter, Cutmix (Yun et al., 2019), and Mixup (Zhang et al., 2018). Before an image is imported into the model, RandomFlip randomly rotates it horizontally or vertically, expanding the dataset, as this is equivalent to the images in the dataset having different shooting angles than their raw form. ColorJitter randomly changes the image's brightness, contrast, saturation, and hue. As a result, ColorJitter can improve the model's ability to adapt to different lights in field settings. The lesions of the three diseases chosen for the study are scattered on the surface of the leaves, which means that the model should not focus only on the lesions of one area but also on the entire leaf. Cutmix randomly crops a patch of the image and fills the area with a small and same size patch from another image. The size of the patches is a hyperparameter, and the position of the patches on the images is random. Mixup is widely used in image classification tasks, and it mainly constructs a virtual sample $\left(\widetilde{\mathrm{x}}, \widetilde{\mathrm{y}}\right)$ by the following methods:

$$\widetilde{\mathrm{x}} = \lambda \mathrm{x}_{\mathrm{i}} + (1-\lambda)\, \mathrm{x}_{\mathrm{j}} \tag{1}$$

$$\widetilde{\mathrm{y}} = \lambda \mathrm{y}_{\mathrm{i}} + (1-\lambda)\, \mathrm{y}_{\mathrm{j}} \tag{2}$$

where $\mathrm{x}_{\mathrm{i}}$ and $\mathrm{x}_{\mathrm{j}}$ are two different images, $\mathrm{y}_{\mathrm{i}}$ and $\mathrm{y}_{\mathrm{j}}$ are the unique one-hot labels corresponding to these two images, and $\lambda \in (0, 1)$. Mixup extends the distribution of samples by linear interpolation, making it popular for various image classification tasks. Both Cutmix and Mixup make models confusing, forcing them to focus on global information rather than local information. This article's data augmentation techniques are used only in training, not testing.

## Algorithm

At the beginning of this section, we have done some specifications of mathematical notation and some pre-paving for our model. The upper case non-bolded symbols in this article refer to matrices, the lower case bolded symbols refer to row vectors, and the lower case non-bolded symbols refer to constants or scalar variables. A complete image can be divided into several local regions. The critical feature information of maize leaf disease is located in some local regions where the lesions are located. From the visual point of view, the texture and color of these local areas are the feature information. From the algorithmic point of view, the RGB values of the pixels in these local areas are the feature information. Background information that interferes with the classification is useless information. CNNs usually downsample the image and use the generated feature maps to represent the information of the image. Our model encodes the feature information of local regions into vectors (called tokens) to represent the information of the whole image. The attention mechanism of this article will be based on these tokens to identify those critical regions to make the classification.

Our standard model has three stages, namely, Stage 1, Stage 2, and Stage 3 (**Figure 3**). Stage 1 extracts the image features and encodes them into a feature tokens matrix. Each row vector in the tokens matrix is a token, and a token is a vector used to represent the local visual features within a receptive field (convolution or max-pooling kernel). Passing the input image $\mathrm{I} \in \mathrm{R}^{\mathrm{h} \times \mathrm{w} \times \mathrm{c}}$ through a convolutional layer and a max-pooling layer generates feature maps $Fm \in \mathbb{R}^{l \times l \times d}$ with channels of d. The width of feature maps output from both convolution layer and max-pooling layer is expressed by the following equation:

$$l = \frac{i+2p-k}{s}+1 \tag{3}$$

where $i$ denotes the width of the original image or input feature maps, $k$ is the size of the kernel (convolution or max-pooling), $s$ is the stride of kernel movement, and $p$ is padding. We listed those hyperparameters at the end of the section algorithm. At the end of Stage 1, after extracting vectors along the channel dimension for the feature maps $Fm$, the vectors are arranged to obtain the feature tokens matrix, which can be described by the following equation:

$$X = Flatten\,(Fm) \tag{4}$$

where $X \in \mathbb{R}^{n \times d}$ is the tokens matrix, and $n = l^2$. Each row vector of dimension d in $X$ is a feature token.

At the beginning of Stage 2, a learnable vector "classification token" of dimension d is appended to the top of $X$; hence, $X \in \mathbb{R}^{n \times d}$ was transformed to $X \in \mathbb{R}^{n_t \times d}$, where $n_t = n + 1$. The "classification token" is derived from NLP and is similar to BERT's (Devlin et al., 2018) "class token." The classification token will be output at the end of Stage 2 as input to Stage 3 to complete the final classification. Therefore, the transformer encoder of Stage 2 is the core computational module of the whole network, and the essential part of it is multi-head self-attention (MSA) that is used to perform self-attention. The rest of the section algorithm will introduce how it works, demonstrating why it is effective. To better explain MSA, we first described the computational process of single-head self-attention (SSA). Tokens matrix $X$ is linearly transformed into queries Q, keys K, and values V by three matrices, $W_Q$, $W_K$, and $W_V$, respectively, and the linear transforms can be seen in the following equations:

$$Q = XW_Q \tag{5}$$

$$K = XW_K \tag{6}$$

$$V = XW_V \tag{7}$$

where $W_Q \in \mathbb{R}^{d \times d}$, $W_K \in \mathbb{R}^{d \times d}$, and $W_V \in \mathbb{R}^{d \times d}$ are parametric learnable matrices. In fact, each row vector in Q, K,

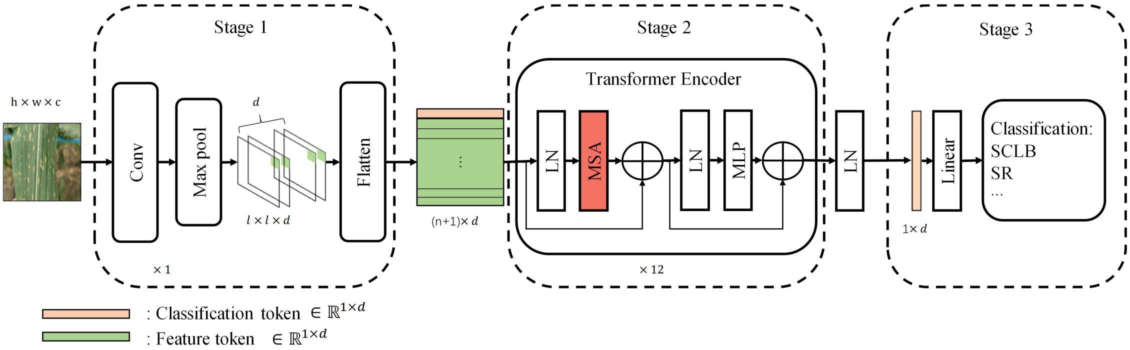


**FIGURE 3 |** The standard model architecture. Stage 1 extracts information from local regions of the image and encodes them into tokens. Stage 2 is the core computational network that performs the self-attention. Stage 3 maps the classification token into four classes to complete the final classification. Linear, linear layer; LN, layer normalization; MLP, multilayer perceptron.

and V is still a token used to represent feature information of the corresponding local region. Assume that $\mathbf{q_i}$, $\mathbf{k_i}$, and $\mathbf{v_i}$ denote the *i*-th token of *Q*, K, and V, respectively; thus, they all represent the feature information of the *i*-th receptive field of the original image. The correlation between tokens is obtained by calculating the inner product of all row vectors in *Q* and all row vectors in K. For example, $\langle \mathbf{q_i}, \mathbf{k_j} \rangle = \mathbf{q_i}\mathbf{k}_j^T$ represents the correlation between the *i*-th token and the *j*-th token or the degree of attention of the *i*-th token to the *j*-th token. However, it is usually not equal to $\langle \mathbf{q_j}, \mathbf{k_i} \rangle = \mathbf{q_j}\mathbf{k_i}^T$, which is due to two factors. On the one hand, *Q* and K are obtained by a linear transformation of two different learnable matrices, $W_Q$ and $W_K$. Although both $\mathbf{q_i}$ and $\mathbf{k_i}$ represent the visual information of the *i*-th receptive field, the elements in $W_Q$ and $W_K$ change in the direction favorable to the final classification as the model weights are updated. On the other hand, the self-attention mechanism is derived from NLP, where words are encoded as vectors (tokens) in a machine translation task. The correct translation of a sentence requires finding the relevance of each word, and two words have different attention to each other, which requires the correlation calculation method between tokens as described earlier. Therefore, the correlation between tokens can be calculated by the following equations:

$$A = QK^T = \begin{bmatrix} \mathbf{q}_1\mathbf{k}_1^T & \mathbf{q}_1\mathbf{k}_2^T & \cdots & \mathbf{q}_1\mathbf{k}_{n_t}^T \\ \mathbf{q}_2\mathbf{k}_1^T & \mathbf{q}_2\mathbf{k}_2^T & \cdots & \mathbf{q}_2\mathbf{k}_{n_t}^T \\ \vdots & \vdots & & \vdots \\ \mathbf{q}_{n_t}\mathbf{k}_1^T & \mathbf{q}_{n_t}\mathbf{k}_2^T & \cdots & \mathbf{q}_{n_t}\mathbf{k}_{n_t}^T \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n_t} \\ a_{21} & a_{22} & \cdots & a_{2n_t} \\ \vdots & \vdots & & \vdots \\ a_{n_t1} & a_{n_t2} & \cdots & a_{n_tn_t} \end{bmatrix} \tag{8}$$

*A* is the preliminary tokens correlation matrix; in other words, it represents the attention between tokens, e.g., $a_{ij}$ denotes the attention of the *i*-th token to the *j*-th token or the attention of the visual information of the *i*-th receptive field to the visual information of the *j*-th receptive field. The following equations normalize the attention matrix *A*:

$$A^{'} = softmax\left(\frac{A}{\sqrt{d_k}}\right) = \begin{bmatrix} \sigma(\mathbf{a}_1)/\sqrt{d_k} \\ \sigma(\mathbf{a}_2)/\sqrt{d_k} \\ \vdots \\ \sigma(\mathbf{a}_{n_t})/\sqrt{d_k} \end{bmatrix} \tag{9}$$

$$\sigma(\mathbf{a}_i) = \left[ \frac{e^{a_{i1}}}{\sum_{j=1}^{n_t} e^{a_{ij}}} \; \frac{e^{a_{i2}}}{\sum_{j=1}^{n_t} e^{a_{ij}}} \cdots \frac{e^{a_{in_t}}}{\sum_{j=1}^{n_t} e^{a_{ij}}} \right] \tag{10}$$

where $d_k$ is a normalization factor and a hyperparameter. Assume that $\alpha_{ij}$ is the element in row i and column j of $A^{'}$. Subsequently, elements in attention matrix $A^{'}$ are used as weights to linearly combine the tokens of V, which will integrate the information of the tokens they are focused on for each token. The following equation describes this process:

$$V^{'} = A^{'}V = \begin{bmatrix} \alpha_{11}\mathbf{v}_1 + \alpha_{12}\mathbf{v}_2 + \cdots + \alpha_{1n_t}\mathbf{v_{n_t}} \\ \alpha_{21}\mathbf{v}_1 + \alpha_{22}\mathbf{v}_2 + \cdots + \alpha_{2n_t}\mathbf{v_{n_t}} \\ \vdots \\ \alpha_{n_t1}\mathbf{v}_1 + \alpha_{n_t2}\mathbf{v}_2 + \cdots + \alpha_{n_tn_t}\mathbf{v_{n_t}} \end{bmatrix} \tag{11}$$

Thus, the new tokens of $V^{'}$ are integrated with the information they pay attention to. The above describes the computation of the attention mechanism. In this process, the classification token is fully involved in the computation of the self-attention mechanism, continuously integrating information about receptive fields in a different-attention way, and finally being output for final classification. The mode using tokens to represent receptive field information and integrating tokens information is more suitable for maize leaf disease identification, because the main characteristic of maize leaf diseases is lesions, which are usually small and widely distributed on the leaf surface. Hence, similarity exists between lesions in terms of texture and color, which is reflected in the RGB values of images. The visual information in receptive fields where lesions exist is similar, and vectors encoded are also similar, so critical information of images can be highlighted by the computational model presented earlier. Subsequent experiments and visualizations in this article will also demonstrate that the model will focus on lesions rather than background noise information. MSA is a simple extension of SSA, performing *head* SSA calculations independently of each other in parallel (**Figure 4**), and *head* is a hyperparameter. Based on the SSA presented earlier, the MSA is briefly described by the following equations:

$$Q_i = XW_i^Q \tag{12}$$

$$K_i = XW_i^K \tag{13}$$

$$V_i = XW_i^V \tag{14}$$

$$A_i^{'} = softmax\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) \tag{15}$$

$$V_i^{'} = A_i^{'} V_i \tag{16}$$

$$V^{'} = Concat\left(V_1^{'}, V_2^{'}, \cdots, V_{head}^{'}\right) = V_1^{'} \oplus V_2^{'} \oplus \cdots \oplus V_{head}^{'} \tag{17}$$

where $i = 1, 2, \cdots, head$, $W_i^Q \in \mathbb{R}^{d \times \frac{d}{head}}$, $W_i^K \in \mathbb{R}^{d \times \frac{d}{head}}$, $W_i^V \in \mathbb{R}^{d \times \frac{d}{head}}$, and $\bigoplus$ is the concatenated operation to matrices. Therefore, tokens matrix $X \in \mathbb{R}^{n_t \times d}$ is calculated by the MSA and outputs $V^{'} \in R^{n_t \times d}$.

Layer normalization (LN) (Ba et al., 2016) normalizes input tokens to speed up the convergence by the following equations:

$$LN\left(\mathbf{y_i}, \alpha, \beta\right) = \frac{\mathbf{y_i} - \mu}{\sigma} \odot \alpha + \beta \in R^{n \times d} \tag{18}$$

$$\mu = \frac{1}{d} \sum_{j=1}^{d} y_i^j \tag{19}$$

$$\sigma = \sqrt{\frac{1}{d} \sum_{j=1}^{d} \left(y_i^j - u\right)^2} \tag{20}$$

where $\mathrm{y_i}$ is the *i*-th token, and $\mathrm{y_i^j}$ refers to the *j*-th element of the *i*-th token. α and β are learnable gains and bias, respectively.

Linear layer can perform a linear transformation of the input matrix, which is described by the following equation:

$$M_o = MW + \mathbf{b} \tag{21}$$

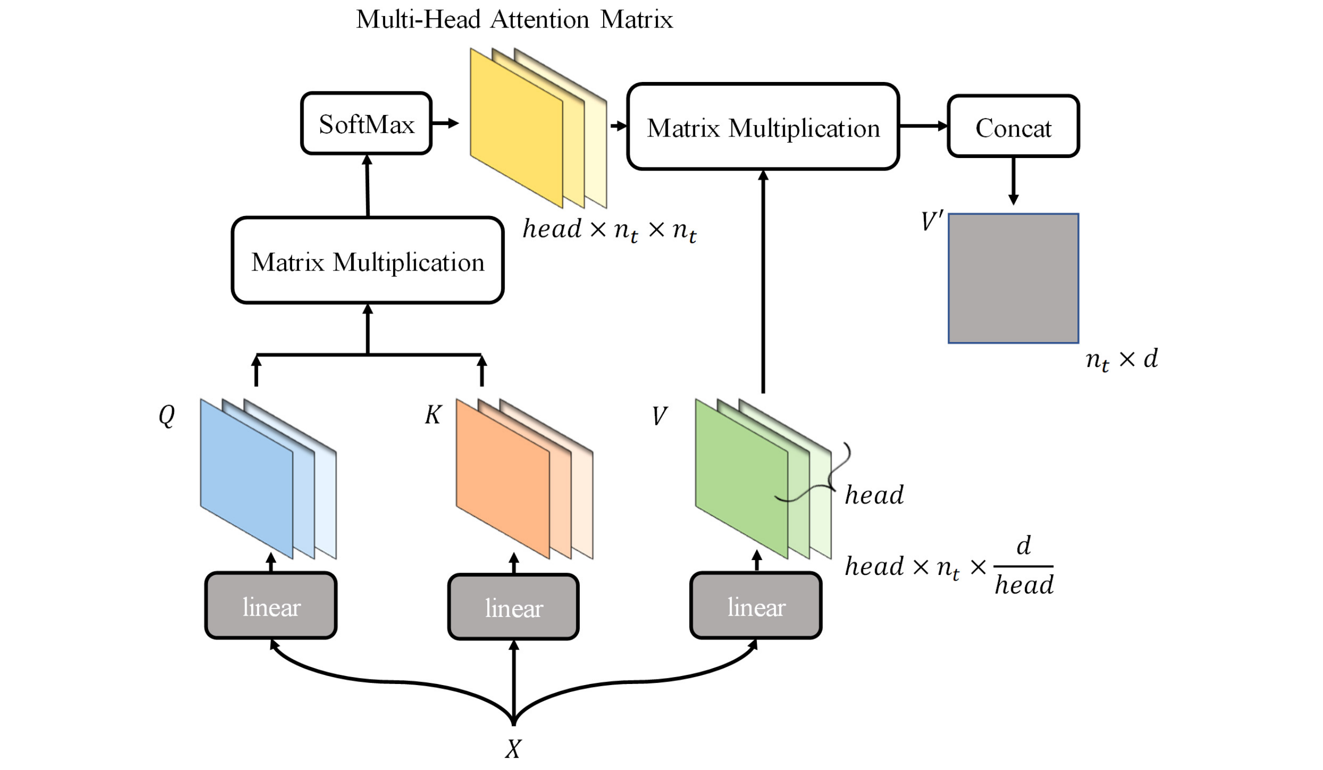


**FIGURE 4 |** The schematic of implementing multi-head self-attention.

where $M \in \mathbb{R}^{m \times n}$ is the input matrix, $W \in \mathbb{R}^{n \times o}$ refers to the learnable matrix, $\mathbf{b} \in \mathbb{R}^{1 \times o}$ refers to the learnable bias vector, and $M_o \in \mathbb{R}^{m \times o}$ refers to the output matrix.

Multilayer perceptron (MLP) obtains nonlinearity and transformation (Han et al., 2021), benefiting from the linear layer and the activation function Gaussian error linear units (GELU) (Hendrycks and Gimpel, 2016). This nonlinear transformation can be described as follows:

$$M_o = GELU(MW_1 + \mathbf{b}_1)W_2 + \mathbf{b}_2 \tag{22}$$

$$GELU(x) = 0.5x\left(1 + \tanh\left[\sqrt{2/\pi}(x + 0.044715x^3)\right]\right) \tag{23}$$

where $M \in \mathbb{R}^{m \times n}$ and $M \in \mathbb{R}^{m \times o}$ refer to input matrix and output matrix, respectively, $W_1 \in \mathbb{R}^{n \times h}$ and $W_2 \in \mathbb{R}^{h \times o}$ are learnable matrices, and $\mathbf{b}_1 \in \mathbb{R}^{1 \times h}$ and $\mathbf{b}_2 \in \mathbb{R}^{1 \times o}$ are learnable bias vectors. The GELU function, when applied to a matrix, will perform a nonlinear transformation on all elements of that matrix.

Stage 3 maps classification token ($\mathbf{v}_0^{'}$, the first row vector of the matrix $V^{'}$ of the last transformer encoder block) of the output of Stage 2 to four categories by a linear layer.

We conducted an ablation experiment on MSA to investigate the necessity of self-attention. The experiment needs to remove the MSA from transformer encoder. However, without the MSA, the classification tokens cannot participate in the computation of the integrated tokens information. Therefore, we designed Model-1 and Model-2 based on the standard model. Model-1 does not use classification tokens but integrates feature tokens to classify, and Model-2 removes MSA from Model-1 (refer to **Figure 5** for Model-1 and Model-2).

## Hyperparameters and Training Facilities

The hyperparameters of our standard model are as follows:

- Max pool layer: number = 1, kernel size = 3, stride = 2, padding = 1,
- Convolutional layer: number = 1, kernel size = 7, stride = 4, padding = 1,
- Transformer encoder blocks: 12,
- Heads of MSA: 4,
- Dimension of token: d 64,
- Normalization factor: $d_k = 16$,
- Batch size: 64,
- Learning rate: 0.004,
- Weight decay: 0.05.

We have made our dataset and code, as well as all the trained models of this article, publicly available in the site: https://github.com/haiyang-qian/code-and-dataset. Our model is trained on the open-source deep learning framework Pytorch 1.9, and the programming language is Python 3.7.10. Our experimental facilities are as follows:

- CPU: Xeon Gold 6142
- GPU: RTX 3090
- CUDA: V11.2
- OS: Ubuntu 20.04
- Memory: 60.9 GB
- SSD: 429.5 GB

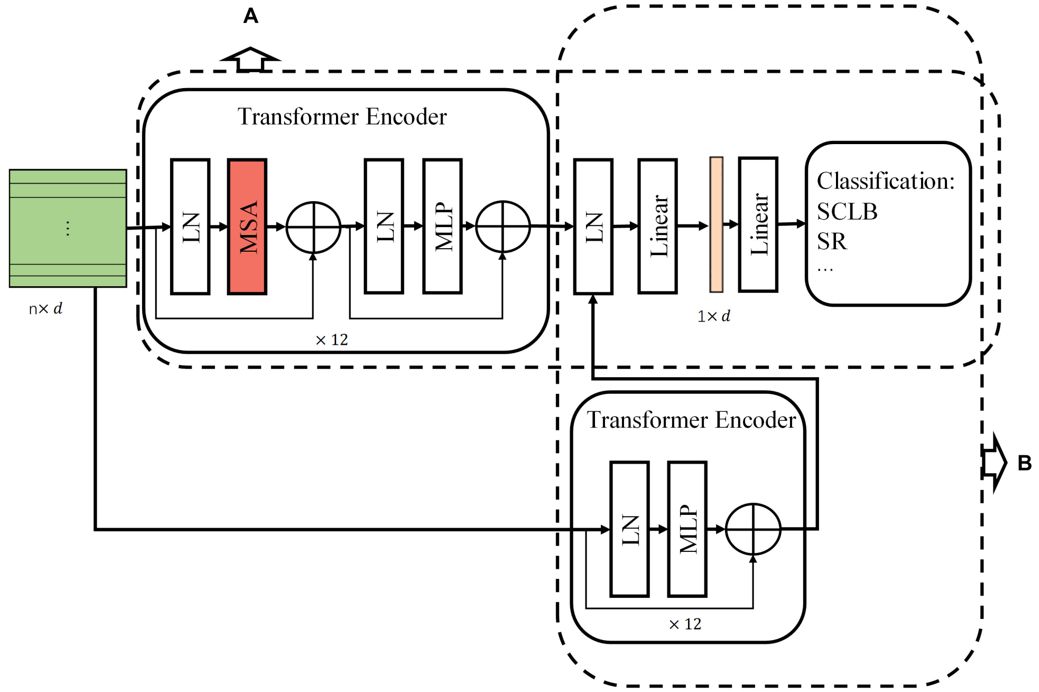


**FIGURE 5 |** Two models changed from the standard model. **(A)** Model-1. **(B)** Model-2.

## Visualization Methods

In this article, three visualization schemes are designed, targeting three network outputs, namely, convolutional or pooling layers, tokens matrix, and classification token for feature tokens' attention. First, for the feature maps of the output of the convolution or pooling layer, we have applied the Grad-Cam method (Selvaraju et al., 2017). The method first computes gradients for class $c$ regarding feature maps $Fm$ of a convolutional layer (assume that $Fm^k$ is the k-th channel of the feature maps). These gradients are globally averaged over the corresponding channels of $Fm$ to obtain the weights of that channel $\alpha_k^c$. $\alpha_k^c$ is the importance of feature map $Fm^k$ for class c and is used to weigh the feature map $Fm^k$. Then, the class discriminative localization map (CDLM) (a map of the importance of different regions of input image for class c) can be obtained by completing this operation for all the feature maps. The above computation can be described by the following equations:

$$\alpha_k^c = \frac{1}{z}\sum_i\sum_j \frac{y^c}{Fm_{ij}^k} \tag{24}$$

$$L_{Grad-CAM}^c = \mathrm{Re}LU\left(\sum_k \alpha_k^c Fm^k\right) \tag{25}$$

$$\mathrm{Re}LU(x) = \begin{cases} x \; if \; x > 0 \\ 0 \; if \; x < 0 \end{cases} \tag{26}$$

where $L_{Grad-CAM}^c$ is a CDLM calculated by the Grad-Cam method, and it will be mapped back to the input image to obtain the visualization result. The Grad-Cam method is usually used for feature maps of the convolution or pooling layer output. Therefore, in our second visualization scheme based on the tokens matrix, we have reshaped the two-dimensional feature tokens matrix $Y$ into a three-dimensional feature map matrix, expressed as the following mapping:

$$Y \in \mathbb{R}^{n \times d} \rightarrow Fm \in \mathbb{R}^{l \times l \times d} \tag{27}$$

where $n = l^2$. We applied the Grad-Cam method to $Fm$ to obtain the results of the second visualization scheme in this article. The third visualization scheme is used to directly map the attention of the classification token to the feature tokens back to the input image. Our standard model has 12 transformer encoder blocks, and each MSA has four heads. The attention of each MSA is combined by the following equations:

$$A^{(i)} = \sum_{j=1}^{4} A^{ij}, i = 1, 2, \cdots, 12 \tag{28}$$

$$A = \sum_{i=1}^{12} \frac{A^{(i)} - \min\left(A^{(i)}\right)}{\max\left(A^{(i)}\right) - \min\left(A^{(i)}\right)} \tag{29}$$

where $A^{ij}$ denotes the attention map of classification token to feature tokens in $j$-th head of $i$-th transformer encoder, and $A^{(i)}$ is the attention map that fuses the attention maps of all the heads in $i$-th transformer encoder. $A$ will be mapped directly to the input image. This visualization scheme does not involve any gradient calculation. It will reflect the attention of the classification token to feature tokens and demonstrate whether the calculation of MSA without increasing parameters is effective for identifying diseased maize leaves.

## Evaluation of Model Performance

We chose accuracy, precision, recall, F1 score, parameters, and floating-point operations per second (FLOPs) to evaluate our classification model. Among them, precision, recall, and F1 score can be calculated by the following equations:

$$Precision = \frac{TP}{TP + FP} \tag{30}$$

$$\mathrm{Re}call = \frac{TP}{TP + FN} \tag{31}$$

$$F1 = \frac{2Precision \times \mathrm{Re}call}{Precision + \mathrm{Re}call} \tag{32}$$

where $TP$ refers to the number of true positives, $FP$ refers to the number of false positives, and $FN$ refers to the number of false negatives.

# RESULTS

All models in this article were trained with 110 epochs. **Figure 6** shows the accuracy and loss of all models as a function of epochs. As can be seen, the performance dramatically improves within the first 20 epochs, but improvement is minor beyond 20 epochs. We compared five mainstream CNNs with our standard model. These CNNs have achieved excellent performance on some specific tasks. For example, MobileNet (Sandler et al., 2018) can be applied to mobile terminals due to its lightweight architecture. ResNet (He et al., 2016) as a baseline is widely used in the industry. EfficientNet (Tan and Le, 2019) has a relatively significant advantage in terms of speed and accuracy. **Table 2** compares the standard model with these CNNs in terms of six metrics (i.e., accuracy, precision, recall, F1 score, parameters, and FLOPs). The accuracies of CNNs reached VGG11 (Simonyan and Zisserman, 2014) 97.9%, ResNet50 96.6%, EfficientNet-b3 91.6%, Inception-v3 (Szegedy et al., 2016) 97.2%, and MobileNet-v2-140 90.7%, whereas the standard model reached 98.7% and surpassed these CNNs. **Figure 7** shows that comparison of the accuracy trends of the standard model with the mainstream CNNs and ViT-base during training. The recall of the standard model for class H is 1% lower than that of Vgg11, but it surpasses Vgg11 in all other metrics. Except for VGG11, the standard model surpasses or ties the rest of these CNNs in accuracy, precision, recall, and F1 score. For the FLOPs metric, MobileNet-v2-140 has lower FLOPs than the standard model and requires less computing power. Since MobileNet-v2-140 is designed for mobile terminals, its FLOPs must be lower than common models. Nevertheless, MobileNet-v2-140 has 6.6 times the number of the standard model parameters. The number of parameters and FLOPs of other models are significantly higher

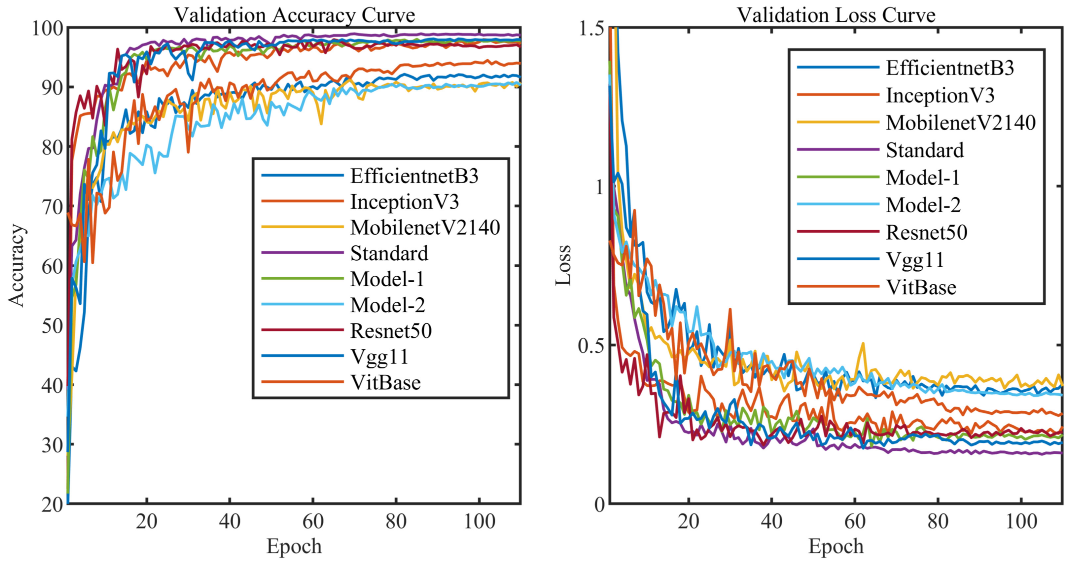


**FIGURE 6 |** Accuracy curve and loss curve of validation set of all models in this article (i.e., Model-1 and Model-2).

**TABLE 2 |** Comparison between the standard model and the other mainstream models.

| | Standard | VGG11 | EfficientNet-b3 | Inception-v3 | MobileNet-v2-140 | ResNet50 | Vit-base |
|---|---|---|---|---|---|---|---|
| Accuracy (%) | 98.7 | 97.9 | 91.6 | 97.2 | 90.2 | 96.6 | 93.9 |
| **Precision (%)** | | | | | | | |
| H | 97 | 96 | 88 | 96 | 88 | 94 | 91 |
| SCLB | 99 | 99 | 90 | 97 | 88 | 99 | 92 |
| SR | 99 | 98 | 94 | 98 | 92 | 96 | 98 |
| GLS | 100 | 100 | 97 | 99 | 99 | 100 | 96 |
| **Recall (%)** | | | | | | | |
| H | 99 | 100 | 92 | 98 | 93 | 99 | 95 |
| SCLB | 97 | 96 | 86 | 94 | 86 | 91 | 90 |
| SR | 100 | 99 | 98 | 100 | 96 | 99 | 97 |
| GLS | 99 | 97 | 89 | 98 | 85 | 97 | 92 |
| **F1 (%)** | | | | | | | |
| H | 98 | 98 | 90 | 97 | 91 | 97 | 93 |
| SCLB | 98 | 97 | 88 | 95 | 87 | 95 | 91 |
| SR | 100 | 98 | 96 | 99 | 94 | 98 | 98 |
| GLS | 100 | 98 | 93 | 99 | 91 | 98 | 94 |
| Parameter (M) | 0.65 | 128.78 | 10.70 | 21.79 | 4.32 | 23.52 | 82.80 |
| FLOPs (G) | 1.47 | 7.61 | 1.62 | 5.72 | 0.59 | 4.10 | 17.58 |

than the standard model (**Figure 8** clearly shows the comparison of parameters and FLOPs of the models), which means that these CNNs are designed to be bloated for maize leaf disease identification in a complex background. As can be seen, for the specific task of this article, stacking the number of layers of the network and increasing the number of parameters of the model are not effective in improving the performance of the model. Our model has only one convolutional layer and one pooling layer to encode local regions of images into tokens, and transformer encoder as the core computational module, which not only significantly reduces the number of parameters and FLOPs of the model but also achieves the best performance. From another perspective, although the number of parameters of the standard model is on average three orders of magnitude lower than the other models in **Table 2**, its FLOPs are in the same order of magnitude as theirs. Since MSA involves large-scale matrix computation when computing the attention matrix between tokens, this operation does not involve the model's parameters but increases the model computation. A comparison in **Table 2** between the standard and ViT (accuracy 93.9%) was created to compare the patch embedding method with the convolution method, showing that the convolution method is superior to the patch embedding method from the perspective of results, which indicates that convolutional layer and max-pooling

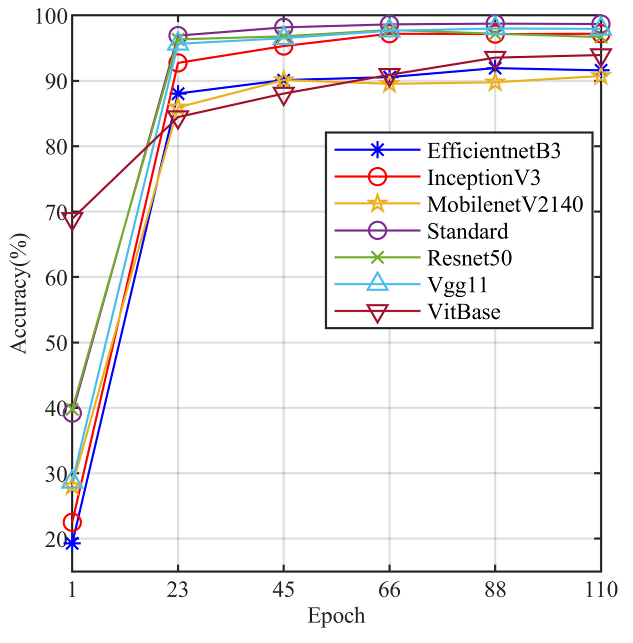


**FIGURE 7 |** The accuracy line chart of the standard model and the other models.

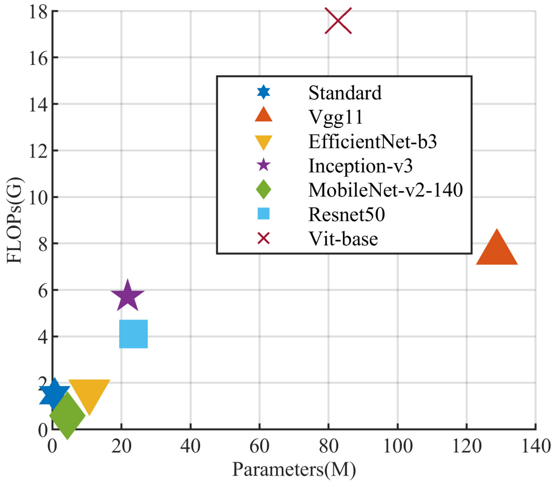


**FIGURE 8 |** Comparison of the standard model and the other models in parameters and FLOPs.

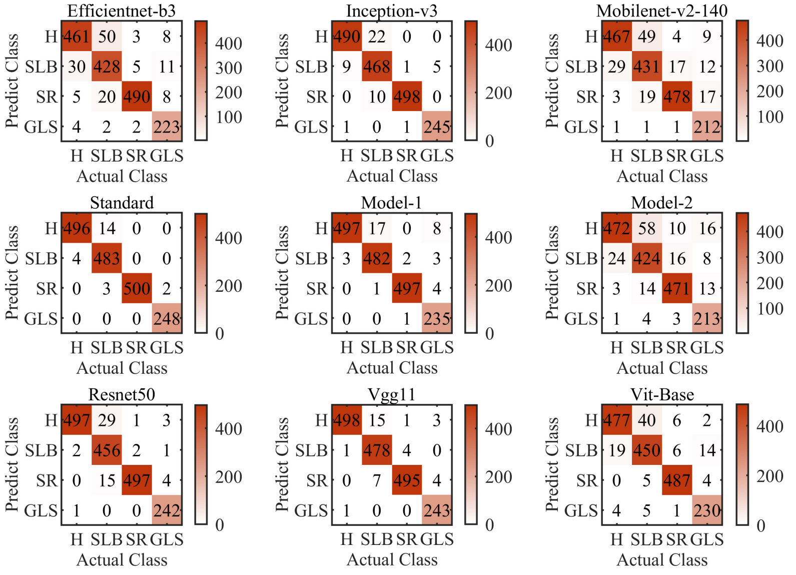
layer can sufficiently encode information of maize leaf disease lesions into tokens and reduce model's parameters.

**Figure 9** shows the confusion matrices of all models of this article (i.e., Model-1 and Model-2). The confusion matrix's abscissa axis represents actual class and ordinate axis represents predicted class. As can be seen, for the nine models, they always tend to identify the SCLB class as the H class. SCLB lesions on maize leaves are minor and scattered, which results in some samples infected similar to H class. In contrast, considering computing power limitation, the size of images can be shrunk small, which leads to SLCB lesions-pixels disappearing and classification error. SR and GLS are rarely misclassified, because their symptoms are markedly distinct from other categories of this article. SR lesions on the leaf tissue's aboveground surface resemble flecks that develop into small golden-brown pustules or bumps. Tan lesions of SR can be distinguished readily from yellow lesions on the surface of maize leaf infected SCLB or GLS.

**Table 3** compares the three models to explore the necessity of the self-attention. Model-1 and Model-2 (**Figure 5**) are modified from the standard model to conduct this study. Model-1 fuses feature tokens into a classification token in Stage 3 by a linear layer instead of adding a classification token at the end of Stage 1, and Model-2 removes the MSA based on Model-1. **Figure 10** clearly shows the increased curve of accuracy of the three models. The accuracy of the standard model exceeds Model-1 by 1%. They have almost the same number of parameters, which indicates that the classification token participating in MSA computation is better than fusing feature tokens into classification tokens. The accuracy of Model-2 is substantially lower than Model-1 by 7.5%. Among other metrics (e.g., precision, recall, and F1 score), Model-2 is also substantially lower than Model-1. The expected results indicate that the self-attention dramatically improves the performance of the model. Model-1 and Model-2 have the same number of parameters, but the FLOPs of Model-2 are much lower than those of Model-1. As mentioned above, the large-scale matrix operations involved in MSA do not increase the number of parameters in the model but do increase the computational complexity of the model. This little computational cost is worth the significant improvement it brings to the model, which also shows that self-attention, a computation that involves almost no parameters of the model, can dramatically improve the identification of maize leaf diseases in complex backgrounds.

In addition, we compared the effect of different train and validation set ratios on the accuracy of the standard model (**Table 4**). As can be seen, the model's accuracy gradually increases as the ratio increases. When the ratio reaches 20–80%, the accuracy reaches 94.0%, while when the ratio reaches 50–50%, the accuracy almost stops increasing. **Figure 11** shows the validation accuracy curve of the standard model over 9 ratios in the training process. The experiment indicates that the standard model can achieve satisfactory performance even when the number of training samples is small.

**Figure 12** provides the results of the visualization of the regions of interest to the model during the classification process. We chose ResNet50 to compare with the standard model and three visualization schemes. For the convolutional or pooling layer-based scheme, we chose the output of the last convolutional layer of layer2 of ResNet50 and the output of the first pooling layer of the standard model because they both output feature maps with a width of 28. In the tokens-based visualization scheme, we selected the output of the first LN layer in the last transformer encoder of the standard model. In the attention matrix-based visualization scheme, we combined the attention matrix of all transformer encoders in the entire model. By comparing **Figures 12A,B**, as can be seen, in field settings with complex backgrounds, ResNet50 has a large amount of attention scattered in the background. In contrast, the attention



Efficientnet-b3

| | H | SLB | SR | GLS |
|---|---|---|---|---|
| H | 461 | 50 | 3 | 8 |
| SLB | 30 | 428 | 5 | 11 |
| SR | 5 | 20 | 490 | 8 |
| GLS | 4 | 2 | 2 | 223 |

Inception-v3

| | H | SLB | SR | GLS |
|---|---|---|---|---|
| H | 490 | 22 | 0 | 0 |
| SLB | 9 | 468 | 1 | 5 |
| SR | 0 | 10 | 498 | 0 |
| GLS | 1 | 0 | 1 | 245 |

Mobilenet-v2-140

| | H | SLB | SR | GLS |
|---|---|---|---|---|
| H | 467 | 49 | 4 | 9 |
| SLB | 29 | 431 | 17 | 12 |
| SR | 3 | 19 | 478 | 17 |
| GLS | 1 | 1 | 1 | 212 |

Standard

| | H | SLB | SR | GLS |
|---|---|---|---|---|
| H | 496 | 14 | 0 | 0 |
| SLB | 4 | 483 | 0 | 0 |
| SR | 0 | 3 | 500 | 2 |
| GLS | 0 | 0 | 0 | 248 |

Model-1

| | H | SLB | SR | GLS |
|---|---|---|---|---|
| H | 497 | 17 | 0 | 8 |
| SLB | 3 | 482 | 2 | 3 |
| SR | 0 | 1 | 497 | 4 |
| GLS | 0 | 0 | 1 | 235 |

Model-2

| | H | SLB | SR | GLS |
|---|---|---|---|---|
| H | 472 | 58 | 10 | 16 |
| SLB | 24 | 424 | 16 | 8 |
| SR | 3 | 14 | 471 | 13 |
| GLS | 1 | 4 | 3 | 213 |

Resnet50

| | H | SLB | SR | GLS |
|---|---|---|---|---|
| H | 497 | 29 | 1 | 3 |
| SLB | 2 | 456 | 2 | 1 |
| SR | 0 | 15 | 497 | 4 |
| GLS | 1 | 0 | 0 | 242 |

Vgg11

| | H | SLB | SR | GLS |
|---|---|---|---|---|
| H | 498 | 15 | 1 | 3 |
| SLB | 1 | 478 | 4 | 0 |
| SR | 0 | 7 | 495 | 4 |
| GLS | 1 | 0 | 0 | 243 |

Vit-Base

| | H | SLB | SR | GLS |
|---|---|---|---|---|
| H | 477 | 40 | 6 | 2 |
| SLB | 19 | 450 | 6 | 14 |
| SR | 0 | 5 | 487 | 4 |
| GLS | 4 | 5 | 1 | 230 |

**FIGURE 9 |** Confusion matrices of all models of this article (i.e., Model-1 and Model-2).

**TABLE 3 |** Research of importance of the self-attention.

| | Standard | Model-1 | Model-2 |
|---|---|---|---|
| Accuracy (%) | 98.7 | 97.7 | 90.2 |
| Precision (%) | | | |
| H | 97 | 95 | 85 |
| SCLB | 99 | 98 | 90 |
| SR | 99 | 99 | 94 |
| GLS | 100 | 100 | 96 |
| Recall (%) | | | |
| H | 99 | 99 | 94 |
| SCLB | 97 | 96 | 85 |
| SR | 100 | 99 | 94 |
| GLS | 99 | 94 | 85 |
| F1 (%) | | | |
| H | 98 | 97 | 89 |
| SCLB | 98 | 97 | 87 |
| SR | 100 | 99 | 94 |
| GLS | 100 | 97 | 90 |
| Parameter (M) | 0.65 | 0.66 | 0.66 |
| FLOPs (G) | 1.47 | 1.46 | 0.33 |

of the standard model is mainly focused on the leaf surface. **Figure 12C** shows that the attention is more refined when representing features based on tokens, effectively suppressing the background information and focusing more on the leaf surface lesions. **Figure 12D** shows the attention distribution of classification token to other feature tokens, which is consistent with the area of attention of the model, which also shows that the MSA calculation mechanism that does not increase the number of model parameters effectively enhances the attention of the model to crucial information and suppresses the useless background noise information.

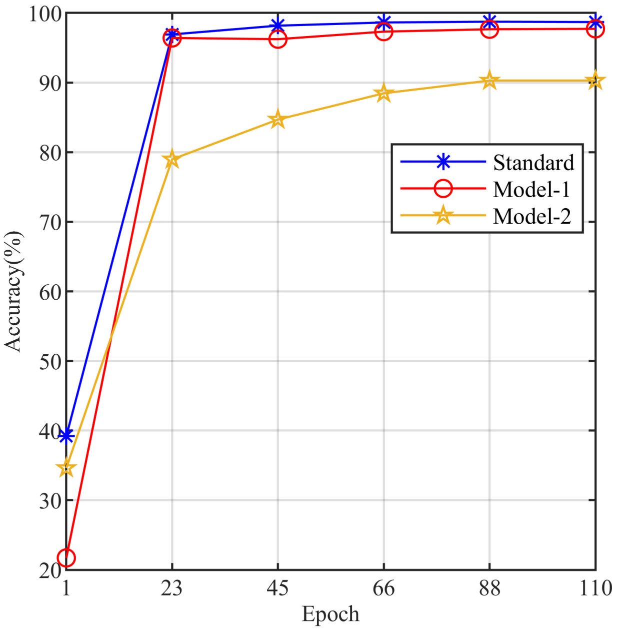


**FIGURE 10 |** The accuracy line chart of the standard model, Model-1, and Model-2.

## DISCUSSION

The common CNNs represent the feature information of an image *via* feature maps, and deepening the depth of the network can generally achieve better performance, but this also increases the number of model parameters and computational effort. They are more suitable for object recognition. The pixels where these objects are located usually do not have similarities, and the overall

**TABLE 4 |** The standard model accuracy results for each train-validation set.

| Train-test split (%) | H | SCLB | SR | GLS | Accuracy |
|---|---|---|---|---|---|
| 10–90 | 243/2,192 | 224/2,019 | 202/1,821 | 100/900 | 0.894 |
| 20–80 | 487/1,948 | 448/1,795 | 404/1,619 | 200/800 | 0.940 |
| 30–70 | 730/1,705 | 672/1,571 | 606/1,417 | 300/700 | 0.967 |
| 40–60 | 974/1,461 | 897/1,346 | 809/1,214 | 400/600 | 0.980 |
| 50–50 | 1,217/1,218 | 1,121/1,122 | 1,011/1,012 | 500/500 | 0.977 |
| 60–40 | 1,461/974 | 1,345/898 | 1,213/810 | 600/400 | 0.980 |
| 70–30 | 1,704/731 | 1,570/673 | 1,416/607 | 700/300 | 0.986 |
| 80–20 | 1,948/487 | 1,794/449 | 1,618/405 | 800/200 | 0.990 |
| 90–10 | 2,191/244 | 2,018/225 | 1,820/203 | 900/100 | 0.989 |
| Total | 2,435 | 2,243 | 2,023 | 1,000 | |

pixels composition of the pattern presents the features of the target object. For maize leaf disease recognition, the pixels where the lesions are located usually have similarities (reflected in the RGB values), which requires a feature representation with higher resolution rather than the feature maps of CNNs. Since the feature maps increases with the number of channels but decreases in width as the network feeds forward. The relationship between lesions information and receptive field becomes blurred. There is no correlation computed between the lesions, so increasing the number of network layers will only bring a slight increase in recognition rate while also increasing the volume and complexity of the network. The model used in this article is entirely different from CNNs in that it is based on tokens to represent the visual information of local areas of the image. Stage 1 encodes the visual information of the receptive field into a matrix of feature tokens. The subsequent network does not perform any compression of this matrix. However, it continuously computes the correlation (attention) between tokens by MSA, making the network pay more attention to information about the lesions useful for classification and suppressing the noisy information in the background. We demonstrated this idea from this article's theoretical, experimental, and visual analysis perspectives. Tokens represent the local feature information of images,

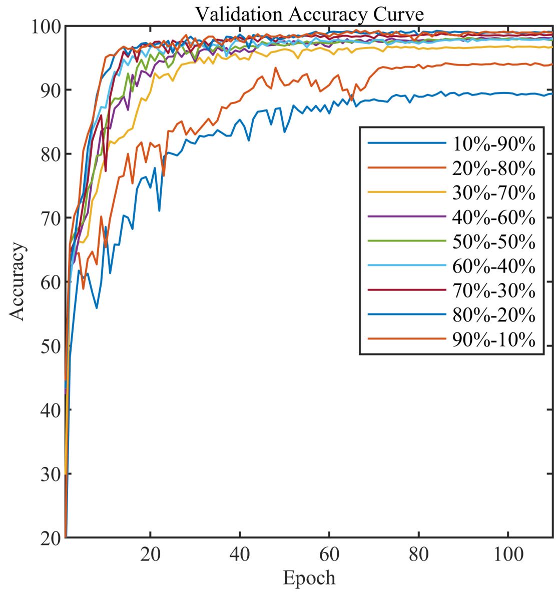


**FIGURE 11 |** The validation accuracy curve of the standard model in nine train-validation sets.

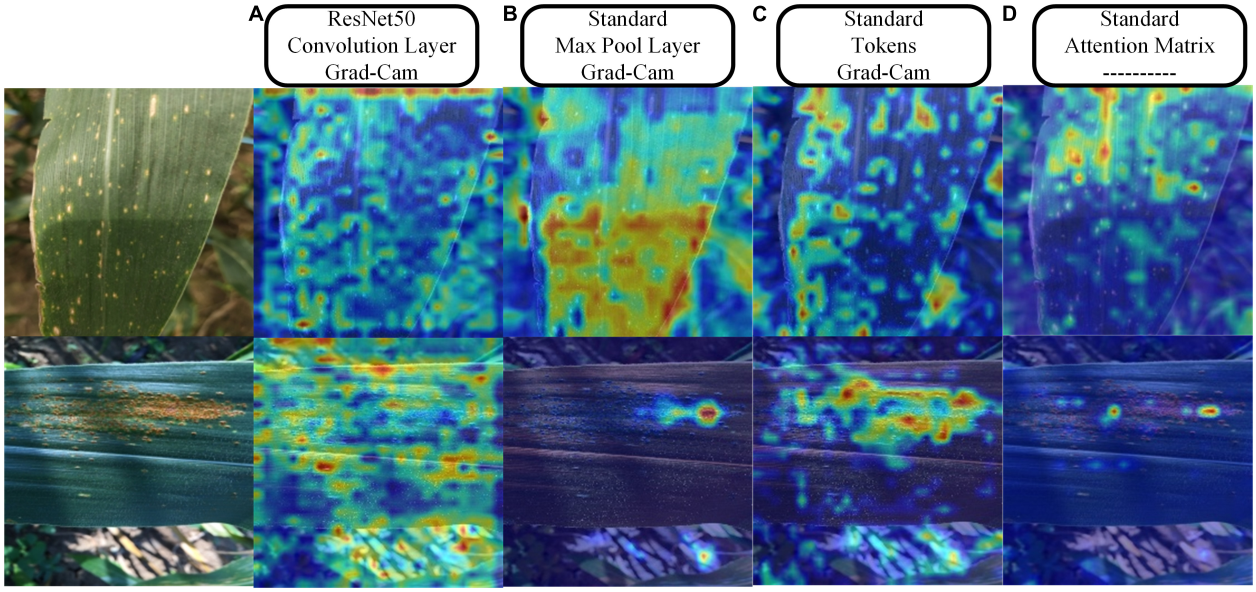


**FIGURE 12 |** Visualization results of the three schemes. **(A)** Grad-Cam method for visualization of the feature maps output by the Resnet50 convolutional layer. **(B)** Grad-Cam method for visualization of the feature maps output from the max-pooling layer of the standard model. **(C)** Grad-Cam method for visualization of the tokens of the standard model. **(D)** Directly map the attention of the classification token on feature tokens to the original image.

and self-attention calculates the correlation of local information, which is more suitable for maize leaf disease identification in complex background. Therefore, guided by the above analysis, we designed a more reasonable model that achieves the best performance with minimal computational cost and number of parameters compared with other mainstream CNNs. However, our model has some limitations. The token (i.e., a single vector) dimension is a hyperparameter. As it increases, the feature information can be represented more abundantly, increasing the attention matrix's scale. Large-scale matrix operations can rapidly increase the computational complexity of the model. Many researchers are now actively working to overcome this challenge (Carion et al., 2020; Liu et al., 2021; Touvron et al., 2021).

In addition, the results above indicate that convolution method outperforms the patch embedding method in encoding maize disease features into feature tokens. Convolution kernel as receptive field extracts visual information by sliding of itself. Two slides of the receptive field have an overlapped area, associating the semantic information of the area. However, the patch embedding method cuts a complete image into many irrelevant patches and directly encodes these patches into tokens, leading to the semantic information of adjacent areas to be lost. Humans tend to process critical vision information instead of all receptive field information, which is mainly limited by the brain's inability to process massive information simultaneously. The mechanism by which humans process visual information is consistent with our model based on the attention mechanism, and they both prefer critical information.

In the field of plant disease identification, the hyperspectral imaging technology is usually used for object detection because the difference in reflectance of plant disease features is slight (Yue et al., 2015; Polder et al., 2019; Wang D. et al., 2019). The investigation of Nagasubramanian et al. (2019) demonstrated that soybeans infected the charcoal rot are more sensitive than healthy soybeans in the wavelengths of visible spectra (400–700 nm). Yang et al. (2021) have achieved good results in the Citrus Huanglongbing detection task by fusing hyperspectral data in CNNs using a multimodal approach. Recent research has shown that the transformer architecture is better suited for multimodal tasks (Frank et al., 2021; Zhang et al., 2021). We will conduct research by extending our model to combine with multimodal approaches for crop disease identification and detection in complex backgrounds in future.

## DATA AVAILABILITY STATEMENT

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

## AUTHOR CONTRIBUTIONS

XQ and KL conceived the study and wrote the manuscript. XQ implemented the algorithm. LC and CZ described the diseases and provided the dataset. All authors contributed to the article and approved the submitted version.

## FUNDING

This work was supported by the Open-End Fund of Information Materials and Intelligent Sensing Laboratory of Anhui Province (IMIS202009) and Anhui Agricultural University Introduction and Stabilization of Talents Research Funding Project (No. yj2020-74).

## REFERENCES

Ahila Priyadharshini, R., Arivazhagan, S., Arun, M., and Mirnalini, A. (2019). Maize leaf disease classification using deep convolutional neural networks. *Neural Comput. Applic.* 31, 8887–8895. doi: 10.1007/s00521-019-04228-3

Aregbesola, E., Ortega-Beltran, A., Falade, T., Jonathan, G., Hearne, S., and Bandyopadhyay, R. (2020). A detached leaf assay to rapidly screen for resistance of maize to Bipolaris maydis, the causal agent of southern corn leaf blight. *Eur. J. Plant Pathol.* 156, 133–145. doi: 10.1007/s10658-019-01870-4

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/1607.06450 (accessed April 1, 2021).

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). "End-to-end object detection with transformers," in *Proceedings of the European Conference on Computer Vision*, (Cham: Springer), 213–229. doi: 10.1007/978-3-030-58452-8_13

Chattopadhay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N. (2018). "Grad-cam++: generalized gradient-based visual explanations for deep convolutional networks," in *Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, (Piscataway, NJ: IEEE), 839–847. doi: 10.1109/WACV.2018.00097

Chen, L., Chen, J., Hajimirsadeghi, H., and Mori, G. (2020). "Adapting Grad-CAM for embedding networks," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, (Piscataway, NJ: IEEE), 2794–2803. doi: 10.1109/WACV45572.2020.9093461

DeChant, C., Wiesner-Hanks, T., Chen, S., Stewart, E. L., Yosinski, J., Gore, M. A., et al. (2017). Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. *Phytopathology* 107, 1426–1432. doi: 10.1094/PHYTO-11-16-0417-R

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: pre-training of deep bidirectional transformers for language understanding. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/1810.04805 (accessed March 25, 2021).

Dhaka, V. S., Meena, S. V., Rani, G., Sinwar, D., Kavita, Ijaz, M. F., et al. (2021). A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. *Sensors* 21:4749. doi: 10.3390/s21144749

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: transformers for image recognition at scale. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/2010.11929 (accessed May 22, 2021).

Frank, S., Bugliarello, E., and Elliott, D. (2021). Vision-and-language or vision-for-language? on cross-modal influence in multimodal transformers. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/2109.04448 (accessed February 22, 2022).

Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., and Wang, Y. (2021). Transformer in transformer. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/2103.00112 (accessed July 31, 2021).

Hassani, A., Walton, S., Shah, N., Abuduweili, A., Li, J., and Shi, H. (2021). Escaping the big data paradigm with compact transformers. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/2104.05704 (accessed August 21, 2021).

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*, (Piscataway, NJ: IEEE), 770–778. doi: 10.1109/CVPR.2016.90

Hendrycks, D., and Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/1606.08415 (accessed August 21, 2021).

Jiang, P.-T., Zhang, C.-B., Hou, Q., Cheng, M.-M., and Wei, Y. (2021). Layercam: exploring hierarchical class activation maps for localization. *IEEE Trans. Image Process.* 30, 5875–5888. doi: 10.1109/TIP.2021.3089943

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Adv. Neural Inform. Process. Syst.* 25, 1097–1105.

Kundu, N., Rani, G., Dhaka, V. S., Gupta, K., Nayak, S. C., Verma, S., et al. (2021). IoT and interpretable machine learning based framework for disease prediction in pearl millet. *Sensors* 21:5386. doi: 10.3390/s21165386

Li, Y., Yang, L., Xu, B., Wang, J., and Lin, H. (2019). Improving user attribute classification with text and social network attention. *Cogn. Comput.* 11, 459–468. doi: 10.1007/s12559-019-9624-y

Liu, J., and Wang, X. (2021). Plant diseases and pests detection based on deep learning: a review. *Plant Methods* 17, 1–18. doi: 10.1186/s13007-021-00722-9

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). Swin transformer: hierarchical vision transformer using shifted windows. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/2103.14030 (accessed September 26, 2021). doi: 10.1109/ICCV48922.2021.00986

Lu, J., Xiong, C., Parikh, D., and Socher, R. (2017). "Knowing when to look: adaptive attention via a visual sentinel for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (Piscataway, NJ: IEEE), 375–383. doi: 10.1109/CVPR.2017.345

Lu, Y., Yi, S., Zeng, N., Liu, Y., and Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. *Neurocomputing* 267, 378–384. doi: 10.1016/j.neucom.2017.06.023

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

Nagasubramanian, K., Jones, S., Singh, A. K., Sarkar, S., Singh, A., and Ganapathysubramanian, B. (2019). Plant disease identification using explainable 3D deep learning on hyperspectral images. *Plant Methods* 15, 1–10. doi: 10.1186/s13007-019-0479-8

Ouppaphan, P. (2017). "Corn disease identification from leaf images using convolutional neural networks," in *Proceedings of the 2017 21st International Computer Science and Engineering Conference (ICSEC)*, (Piscataway, NJ: IEEE), 1–5. doi: 10.1109/ICSEC.2017.8443919

Panigrahi, K. P., Das, H., Sahoo, A. K., and Moharana, S. C. (2020). "Maize leaf disease detection and classification using machine learning algorithms," in *Progress in Computing, Analytics and Networking*, eds P. K. Pattnaik, S. S. Rautaray, H. Das, and J. Nayak (Cham: Springer), 659–669. doi: 10.1155/2022/6504616

Polder, G., Blok, P. M., de Villiers, H. A., van der Wolf, J. M., and Kamp, J. (2019). Potato virus Y detection in seed potatoes using deep learning on hyperspectral images. *Front. Plant Sci.* 10:209. doi: 10.3389/fpls.2019.00209

Ranum, P., Peña-Rosas, J. P., and Garcia-Casal, M. N. (2014). Global maize production, utilization, and consumption. *Ann. N.Y. Acad. Sci.* 1312, 105–112. doi: 10.1111/nyas.12396

Saito, B. C., Silva, L. Q., Andrade, J. A. C., and Goodman, M. M. (2018). Adaptability and stability of corn inbred lines regarding resistance to gray leaf spot and northern leaf blight. *Crop Breed. Appl. Biotechnol.* 18, 148–154. doi: 10.1590/1984-70332018v18n2a21

Saleem, M. H., Potgieter, J., and Mahmood Arif, K. (2019). Plant disease detection and classification by deep learning. *Plants* 8:468. doi: 10.3390/plants8110468

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). "Mobilenetv2: inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (Piscataway, NJ: IEEE), 4510–4520. doi: 10.1109/CVPR.2018.00474

Savary, S., Ficke, A., Aubertot, J.-N., and Hollier, C. (2012). Crop losses due to diseases and their implications for global food production losses and food security. *Food Security* 4, 519–537. doi: 10.1007/s12571-012-0200-5

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*, (Piscataway, NJ: IEEE), 618–626. doi: 10.1109/ICCV.2017.74

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/1409.1556 (accessed September 1, 2021). doi: 10.3390/s21082852

Song, S., Lan, C., Xing, J., Zeng, W., and Liu, J. (2017). "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, (Piscataway, NJ: IEEE).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (Piscataway, NJ: IEEE), 2818–2826. doi: 10.1109/CVPR.2016.308

Tan, M., and Le, Q. (2019). "Efficientnet: rethinking model scaling for convolutional neural networks," in *Proceedings of the International Conference on Machine Learning: PMLR*, (Piscataway, NJ: IEEE), 6105–6114.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. (2021). "Training data-efficient image transformers & distillation through attention," in *Proceedings of the International Conference on Machine Learning: PMLR*, (Piscataway, NJ: IEEE), 10347–10357.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). "Attention is all you need," in *Advances in Neural Information Processing Systems*, eds M. S. Kearns, S. A. Solla, and D. A. Cohn (Cambridge, MA: MIT Press), 5998–6008.

Wang, D., Vinson, R., Holmes, M., Seibel, G., Bechar, A., Nof, S., et al. (2019). Early detection of tomato spotted wilt virus by hyperspectral imaging and outlier removal auxiliary classifier generative adversarial nets (OR-AC-GAN). *Sci. Rep.* 9, 1–14. doi: 10.1038/s41598-019-40066-y

Wang, S., Chen, Z., Tian, L., Ding, Y., Zhang, J., Zhou, J., et al. (2019). Comparative proteomics combined with analyses of transgenic plants reveal Zm REM 1.3 mediates maize resistance to southern corn rust. *Plant Biotechnol. J.* 17, 2153–2168. doi: 10.1111/pbi.13129

Wieczorek, M., Sika, J., Wozniak, M., Garg, S., and Hassan, M. (2021). "Lightweight CNN model for human face detection in risk situations," in *Proceedings of the IEEE Transactions on Industrial Informatics*, (Piscataway, NJ: IEEE). doi: 10.1109/TII.2021.3129629

Yang, D., Wang, F., Hu, Y., Lan, Y., and Deng, X. (2021). Citrus huanglongbing detection based on multi-modal feature fusion learning. *Front. Plant Sci.* 12:809506. doi: 10.3389/fpls.2021.809506

Yang, G., He, Y., Yang, Y., and Xu, B. (2020). Fine-grained image classification for crop disease based on attention mechanism. *Front. Plant Sci.* 11:600854. doi: 10.3389/fpls.2020.600854

Yue, J., Zhao, W., Mao, S., and Liu, H. (2015). Spectral–spatial classification of hyperspectral images using deep convolutional neural networks. *Remote Sens. Lett.* 6, 468–477. doi: 10.1080/2150704x.2015.1047045

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. (2019). "Cutmix: regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (Piscataway, NJ: IEEE), 6023–6032. doi: 10.1109/ICCV.2019.00612

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). "mixup: beyond empirical risk minimization," in *Proceedings of the International Conference on Learning Representations*, (Piscataway, NJ: IEEE).

Zhang, L. N., and Yang, B. (2014). Research on recognition of maize disease based on mobile internet and support vector machine technique. *Adv. Mater. Res.* 905, 659–662. doi: 10.4028/www.scientific.net/amr.905.659

Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L., et al. (2021). "Vinvl: revisiting visual representations in vision-language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (Piscataway, NJ: IEEE), 5579–5588. doi: 10.1109/CVPR46437.2021.00553

Zhang, Z., He, X., Sun, X., Guo, L., Wang, J., and Wang, F. (2015). Image recognition of maize leaf disease based on GA-SVM. *Chem. Eng. Trans.* 46, 199–204.

**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.

ORIGINAL RESEARCH
published: 11 May 2022
doi: 10.3389/frai.2022.871162

# Leveraging Guided Backpropagation to Select Convolutional Neural Networks for Plant Classification

*Sakib Mostafa [1]*, Debajyoti Mondal [1], Michael A. Beck [2,3], Christopher P. Bidinosti [2], Christopher J. Henry [3] and Ian Stavness [1]*

*[1] Department of Computer Science, University of Saskatchewan, Saskatoon, SK, Canada, [2] Department of Physics, University of Winnipeg, Winnipeg, MB, Canada, [3] Department of Applied Science, University of Winnipeg, Winnipeg, MB, Canada*

The development of state-of-the-art convolutional neural networks (CNN) has allowed researchers to perform plant classification tasks previously thought impossible and rely on human judgment. Researchers often develop complex CNN models to achieve better performances, introducing over-parameterization and forcing the model to overfit on a training dataset. The most popular process for evaluating overfitting in a deep learning model is using accuracy and loss curves. Train and loss curves may help understand the performance of a model but do not provide guidance on how the model could be modified to attain better performance. In this article, we analyzed the relation between the features learned by a model and its capacity and showed that a model with higher representational capacity might learn many subtle features that may negatively affect its performance. Next, we showed that the shallow layers of a deep learning model learn more diverse features than the ones learned by the deeper layers. Finally, we propose SSIM cut curve, a new way to select the depth of a CNN model by using the pairwise similarity matrix between the visualization of the features learned at different depths by using Guided Backpropagation. We showed that our proposed method could potentially pave a new way to select a better CNN model.

**Keywords: explainable AI, deep learning—artificial neural network, Guided Backpropagation, neural network visualization, convolutional neural network**

***Edited by:***
*Wenyan Wu,*
*The University of Melbourne, Australia*

***Reviewed by:***
*Lars Linsen,*
*University of Münster, Germany*
*Haibo Chu,*
*Beijing University of Technology, China*

****Correspondence:***
*Sakib Mostafa*
*sakib.mostafa@usask.ca*

***Specialty section:***
*This article was submitted to AI in Food, Agriculture and Water, a section of the journal Frontiers in Artificial Intelligence*

***Received:*** *07 February 2022*
***Accepted:*** *15 April 2022*
***Published:*** *11 May 2022*

***Citation:***
*Mostafa S, Mondal D, Beck MA, Bidinosti CP, Henry CJ and Stavness I (2022) Leveraging Guided Backpropagation to Select Convolutional Neural Networks for Plant Classification. Front. Artif. Intell. 5:871162. doi: 10.3389/frai.2022.871162*

## 1. INTRODUCTION

Deep learning approaches have been widely adopted into agriculture (Weng et al., 2019; Chandra et al., 2020) (i.e., precision agriculture, crop breeding, plant phenotyping) due to their ability to extract complex features from a large amount of data (Montavon et al., 2019). In recent years, the focus has shifted toward developing tools to optimize the performance of the models to help researchers integrate deep learning models easily into their studies (Humphrey et al., 2017; Ubbens and Stavness, 2017; Ubbens et al., 2018). Despite the recent development, deep learning models are often considered as "black box" (Tzeng and Ma, 2005; Oh et al., 2019). To improve the trustworthiness of models and to design them effectively for the unique challenges that appear with specialized datasets, many recent studies have focused on explaining the learning and prediction of deep learning models (Tzeng and Ma, 2005; Mostafa and Mondal, 2021). However, explainable deep learning models in plant phenotyping still remains to be an active field of research with room for improvement (Ubbens and Stavness, 2017; Chandra et al., 2020; Hati and Singh, 2021). Plant image datasets are often different from general image datasets due to small sample sizes, highly self-similar foreground objects, and simplified backgrounds. Therefore, complex deep learning models that are used for general image classification may perform poorly for plant datasets (Mohanty et al., 2016; Zenkl et al., 2022).

Convolutional neural networks (CNN) are one of the most widely used deep learning models in image-based plant phenotyping. A common phenomenon when designing a CNN model is model overfitting. Overfitting in a CNN model occurs when the model approximates or memorizes the training data and fails to generalize to unseen examples in the testing data (Reed and Marks, 1999). A popular way to detect the overfitting is by inspecting the difference between the training and testing accuracy and loss using the accuracy and loss curve (Géron, 2019; Gigante et al., 2019). However, this does not provide insight into the model's learning or which features or part of the image contributed to the model's prediction.

## 1.1. Explainability in CNN

To explain the learning of CNN models, researchers have proposed different feature-map visualization techniques (Springenberg et al., 2014; Bach et al., 2015; Ribeiro et al., 2016; Selvaraju et al., 2016; Lundberg and Lee, 2017). Zeiler and Fergus (2014) proposed deconvolutional networks (Deconvnet) that provide insight into the function of a CNN classifier's intermediate layers by modifying the model's gradient and displaying the visual patterns in the input image that generated the activation. There have been several attempts that deviate from deconvolutional networks. Simonyan et al. (2013) used the gradient of a CNN model's output with respect to the input image's pixel intensities to generate saliency maps. Zhou et al. (2016) and Selvaraju et al. (2017) proposed class activation mapping (CAM), and Gradient-weighted Class Activation Mapping (Grad-CAM), respectively, which helps achieve class-specific feature visualization.

Ghosal et al. (2018) visualized feature maps in various layers that detected the stress regions of a plant leaf. Nagasubramanian et al. (2019) used a saliency map based visualization technique to detect the hyperspectral wavelengths that are responsible for the models' performance. Dobrescu et al. (2017) showed that the model always looks at the leaves in the image in the CNN-based plant classifier. In Dobrescu et al. (2019), the research group used layerwise relevance propagation and GBP to explain the learning of intermediate layers of the CNN model by counting the leaves in an image. Escorcia et al. (2015) studied the visualization of the leaf features and found the existence of attribute-centric nodes, which, rather than learning attributes, learns to detect objects. A more recent work, Lu et al. (2021) used guided upsampling and background suppression to improve models' performance. However, their explanation was limited to the visualization of the instances responsible for the count.

Toneva et al. (2018) explained the learning of the CNN models in terms of forgetting patterns, where at some point during the training, the model correctly predicts an example, but eventually, it is misclassified. Feldman (2020) took a different approach and demonstrated that when there are numerous instances of rare examples in the dataset, the deep learning models must memorize the labels to achieve state-of-the-art performance. Feldman and Zhang (2020) showed that along with memorizing outliers, the deep learning models also memorize training examples and if there are testing examples similar to it and hence overparameterized models perform extraordinarily. Salman and Liu (2019) claimed that overfitting is caused due to the continuous update of a deep learning model's gradient and scale sensitiveness of the loss function. They also proposed a consensus-based classification algorithm for limited training examples.

## 1.2. Contributions

In this study, we focus on the plant species classification, which is relevant in digital agriculture, e.g., precision herbicide application (Weis et al., 2008), and is a prevalent task for employing CNN models (Dyrmann et al., 2016; Azlah et al., 2019). We examine the features learned by the intermediate layers of CNN classifiers to understand the behavior of overfit models and the contribution of image background in overfitting. To examine how the CNN models learn in various conditions (overfit or balanced), we use Guided Backpropagation (GBP) (Springenberg et al., 2014) to visualize the features being learned at different layers of the CNN models. We explore whether the GBP-based feature visualizations could be leveraged to detect the overfitting. We then propose a new technique for model selection that can be used to develop balanced models.

There are three main contributions of this study. First, we visualize the intermediate layers of different CNN models to investigate whether the learning of the features depends on the model's capacity. Second, we propose a novel SSIM-based evaluation technique that relates overfitting to the depth of the model and provides an intuitive way to understand the differences between overfit and balanced models. Here SSIM refers to a measurement of the similarity between two feature map visualizations. Third, we show how our SSIM-based evaluation may help detect potential underfitting or overfitting in the CNN models and allow us to select a balanced model (i.e., a model which is neither overfit nor underfit). In particular, it may suffice to examine models of various depths only at their first training epochs, and the corresponding SSIM-based evaluation may reveal a potential balanced model. This approach can reduce the model selection time by several factors compared to the time needed to train different models to select a preferable depth.

# 2. METHODOLOGY

## 2.1. Guided Backpropagation

The GBP is a gradient-based visualization technique that visualizes the gradient with respect to images when backpropagating through the Relu activation function (Springenberg et al., 2014). GBP allows the flow of only the positive gradients by changing the negative gradient values to zero. This allows visualizing the image features that activate the neurons. Let $f$ be the feature map of any layer $l$ then the forward pass is $f_i^{l+1} = Relu(f_i^l, 0)$. Since GBP only allows the flow of positive gradients, the backward pass of the GBP is $R_i^l = (f_i^l > 0) \cdot (R_i^{l+1} > 0) \cdot (R_i^{l+1})$, where $R$ is an intermediate result on the calculation of the backpropagation for layer $l$. The

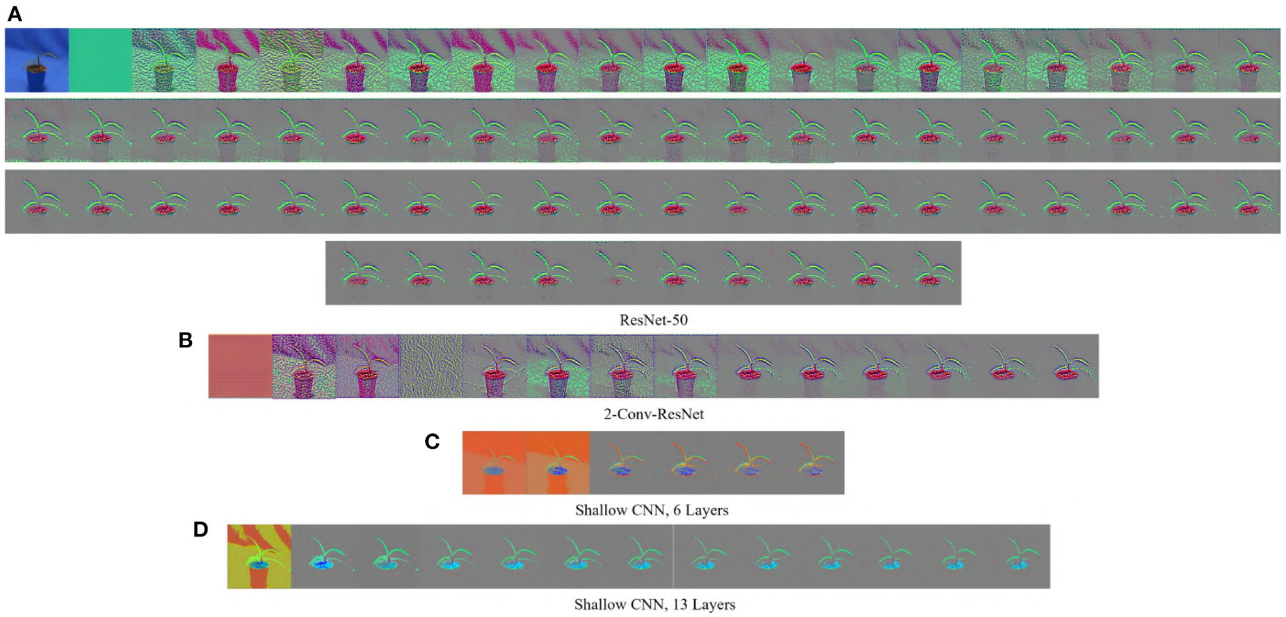


**FIGURE 1 |** GBP-based visualization of the intermediate layers (left to right) of different CNN models for Barnyard Grass of the Weedling dataset. The top-left image of **(A)** is the input image for all models. **(A)** ResNet-50. **(B)** 2-Conv-ResNet. **(C)** Shallow CNN, 6 layers. **(D)** Shallow CNN, 13 layers.

final output of the GBP is an image of the same dimension as the input, displaying the features of the input image that maximized the activation of the feature maps. A major advantage of GBP is that it works for both convolutional layers and fully connected layers. **Figure 1** shows some examples of the visualization generated by GBP for the Weedling dataset using ResNet-50 (He et al., 2016). The gray color in the output of the GBP images (**Figure 1**) represents that the features in those positions of the input image do not contribute to the prediction.

## 2.2. SSIM Cut Curve

We use the GBP approach to visualize the features learned by the intermediate layers of a CNN (e.g., see **Figure 1**). GBP creates an RGB image with the same shape as the input image representing the learned features for every layer. **Figure 2** depicts pairwise SSIM matrices for ResNet-50 and 2-Conv-ResNet models on different datasets, i.e., each entry $(i, j)$ denotes the SSIM value between the GBP visualizations obtained for the $i$th and $j$th convolutional layer of ResNet-50 and 2-Conv-ResNet. Here a darker red indicates higher SSIM. From the color-coding, we can observe that the pairwise SSIM is much lower at the initial layers compared to the layers at a deeper layer. This inspired us to find a way to separate the initial (dissimilar) layers from the later (similar) layers. Let $L_1, L_2, ..., L_n$ be the GBP visualization for different convolutional layers of a CNN model with $n$ convolutional layers. The intuition is that the number $k$, where $1 \leq k \leq n$, with the best separation between $\{L_1, \ldots, L_k\}$ and $\{L_{k+1}, \ldots, L_n\}$ would suggest a reasonable depth for the model to have a good performance.

Given a number $k$ (i.e., a cut position), we first define a *SSIM cut value* $\mathcal{C}_k$ to obtain an estimation of how good the cut is for the value $k$. We define $\mathcal{C}_k$ to be the mean pairwise similarity between $\{L_1, \ldots, L_k\}$ and $\{L_{k+1}, \ldots, L_n\}$:

$$\mathcal{C}_k = \frac{1}{k(n-k)} \sum_{i=1}^{k} \sum_{j=k+1}^{n} s_{i,j} \tag{1}$$

where $s_{i,j}$ is the SSIM between $L_i$ and $L_j$. In the rest of the article, we will refer to the function $\mathcal{C}_k$ with respect to $k$ as the *SSIM cut curve*.

We can observe this phenomenon better by examining the rate of change, as follows. Let $M_i$ be the sum of the SSIM values of $L_i$ with all other layers. Then $\mathcal{C}_k$ can be rewritten as $\mathcal{C}_k = \frac{1}{k(n-k)} \left( \sum_{i=1}^{k} M_i - \sum_{i=1}^{k} \sum_{j=1}^{k} s_{i,j} \right)$. If the curve appears to be flat around the middle cut positions, i.e., when $k \approx (n-k)$, then $\Delta\mathcal{C}_k = \mathcal{C}_{k+1} - \mathcal{C}_k = 0$. In other words, we will have $\Delta\mathcal{C}_k \approx M_{k+1} - 2\sum_{i=1}^{k} s_{i,k+1} = 0$, and hence $\sum_{i=1}^{k+1} s_{i,k+1} = \frac{1}{2} M_{k+1}$. Thus the similarity of $L_{k+1}$ with the earlier layers $\{L_1, \ldots, L_k\}$ will be equal to its similarity with the rest of the layers $\{L_{k+1}, \ldots, L_n\}$.

## 2.3. Research Questions

In a CNN, it is expected that the convolutional layers will learn features from the foreground objects in images that are being classified (Kamal et al., 2021). The background features are considered irrelevant, and often these features are not consistent in the images. However, the features learned by a CNN model are also dictated by the model's capacity. Models with a large number of layers have a very high representational capacity, and therefore

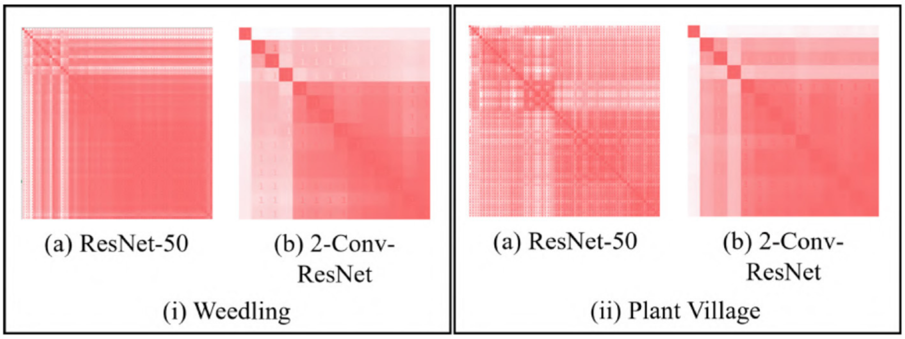


**FIGURE 2 |** SSIM matrix ($s_{i,j}$) generated with GBP images for **(i)** Barnyard Grass of the Weedling dataset **(ii)** Apple leaf of the Plant Village dataset using (a) ResNet-50 and (b) 2-Conv-ResNet. A darker red indicates higher SSIM.

such models are expected to learn diverse features. Although this is widely believed, no formal exploration has been done in the plant phenotyping context. We thus explore the following research question.

**RQ1:** Are the variety of features learned by a model influenced by the model's capacity?

In a model with high representational capacity, the presence of the potential redundant layers can cause the model to overfit by memorizing irrelevant features. As a result, it performs well for the training images but fails to classify the testing images due to the absence of the features present in the training set. To investigate the presence of redundant layers in CNN models, we considered the following question.

**RQ2:** How diverse are the features learned at different depths in a deep CNN model?

The visualizations of the feature maps represent the learning of the layers. The SSIM similarity of these visualizations may be an effective tool to investigate various options for model depth and potentially select a balanced model.

**RQ3:** Can the SSIM-based evaluation of the feature map visualizations be leveraged to select the required depth of a model?

## 2.4. Datasets

The use of deep learning in plant phenotypic tasks are gradually gaining popularity (Scharr et al., 2016; Aich and Stavness, 2017; Ubbens and Stavness, 2017; Aich et al., 2018), and the dataset plays a vital role as it contains a large amount of noise representing the real-world scenarios. Manually measuring the plant traits is a time-consuming process, which is also prone to error. Image-based automated plant trait analysis using deep learning can help overcome these drawbacks (Aich et al., 2018). However, most of the studies explaining the deep learning models use benchmark datasets (e.g., MNIST (Deng, 2012), Fashion-MNIST (Xiao et al., 2017), and so on), and very few studies have attempted to explain the learning using a plant dataset (Dobrescu et al., 2019).

We used three plant datasets: Weedling dataset (Beck et al., 2020, 2021), Plant Village dataset (Mohanty, 2018), and Plant Seedling dataset (Giselsson et al., 2017) which are commonly used for creating deep learning models for plant phenotyping tasks. For all the datasets, 80% of the available images were used for training and 10% for testing and 10% for validation. The detailed overview of the datasets is available in Mostafa et al. (2021).

## 2.5. Deep Learning Models

**ResNet-50:** In this study, we used the ResNet-50 model with random weight initialization and adam optimizer as the optimization function. We also replaced the top layer of the model with a fully connected layer, where Softmax was the activation function, and the number of neurons was the number of classes in a dataset. We trained the model for 100 epochs and only used the model with the highest testing accuracy.

**2-Conv-ResNet, 3-Conv-ResNet, 4-Conv-ResNet:** Keras ResNet-50 model is an implementation of the architecture proposed by He et al. (2016), where the authors used five convolutional blocks. However, we also used smaller versions of the ResNet-50, where we sequentially increased the number of blocks to create 2-Conv-ResNet, 3-Conv-ResNet, and 4-Conv-ResNet. For example, in 2-Conv-ResNet we only used the layers in Conv1 and Conv2_x (see **Table 1**, He et al., 2016), and in 3-Conv-ResNet we used the layers in Conv1, Conv2_x, and Conv3_x. In different models, apart from discarding the convolutional blocks, the rest of the architecture remained the same. We used the modified ResNet models to investigate the relation between the model depth and SSIM cut curve and to see whether decreasing the depth helps avoid overfitting.

**ResNet-50-10% and 2-Conv-ResNet-10%:** In an attempt to create overfit models for this study, we trained the ResNet-50 and 2-Conv-ResNet on 10% training data for the Weedling and Plant Village dataset; but we left out the Plant Seedling dataset due to its small size.

**Shallow CNN:** Along with the ResNet-50, we also used two shallow CNN models for our experiments: one with 6 convolutional layers and the other with 13 convolutional layers, which we named **Shallow CNN, 6 Layers** and **Shallow CNN, 13 Layers**, respectively. In the shallow CNN models, we only used a combination of convolutional layers and avoided using the residual connection. These models aim to examine whether the observations obtained from the comparative analysis between ResNet-50 and 2-Conv-ResNet also hold for shallow CNN models.

For the shallow CNNs, we used categorical cross-entropy as the loss function, random weight initialization, and adam optimizer to optimize the models. Similar to ResNet models, we trained shallow models for 100 epochs with a minibatch size of 16 and chose the model with maximum testing accuracy. While training the shallow CNNs on the Weedling dataset, we resized the images to 512 × 512. For the other datasets, the size of the images was 224 × 224, as it is required for the ResNet models. We used varying zoom range, image flipping, and distorting images along an axis (shear angle) for data augmentation and added an additional batch of augmented images during each epoch. The model architecture and more details of the shallow CNN models are in the **Supplementary Material**.

The training, testing, and validation accuracy of different models on the different datasets are in **Table 1**. In this study, for the Weedling and Plant village dataset, we have considered the ResNet-50-10% and 2-Conv-ResNet-10% as overfit models due to their significant difference between training accuracy and testing accuracy. All the models for the Plant Seedling dataset were overfit except the shallow CNN with 13 convolutional layers, which had a very poor accuracy indicating the model was not optimized for the classification.

**TABLE 1** | Performances of different models for various datasets.

| Dataset name | Model name | Training accuracy (%) | Testing accuracy (%) | Validation accuracy (%) |
|---|---|---|---|---|
| Weedling | ResNet-50 | 98.70 | 96.70 | 96.29 |
| | ResNet-50-10% | 99.89 | 50.70 | 62.88 |
| | 2-Conv-ResNet | 99.88 | 95.53 | 95.62 |
| | 2-Conv-ResNet-10% | 99.89 | 52.10 | 44.68 |
| | 3-Conv-ResNet | 99.93 | 95.14 | 95.13 |
| | 4-Conv-ResNet | 99.75 | 94.21 | 94.41 |
| | Shallow CNN, 6 Layers | 94.00 | 89.60 | 88.73 |
| | Shallow CNN, 13 Layers | 96.23 | 95.45 | 94.91 |
| Plant village | ResNet-50 | 98.59 | 98.04 | 98.27 |
| | ResNet-50-10% | 87.99 | 77.93 | 65.45 |
| | 2-Conv-ResNet | 99.25 | 99.17 | 99.31 |
| | 2-Conv-ResNet-10% | 90.91 | 82.57 | 65.02 |
| | 3-Conv-ResNet | 99.65 | 99.29 | 99.30 |
| | 4-Conv-ResNet | 96.74 | 96.91 | 96.83 |
| | Shallow CNN, 6 Layers | 98.26 | 96.46 | 97.75 |
| | Shallow CNN, 13 Layers | 96.96 | 96.46 | 96.48 |
| Plant seedling | ResNet-50 | 91.26 | 81.90 | 80.38 |
| | 2-Conv-ResNet | 85.16 | 68.75 | 61.67 |
| | Shallow CNN, 6 Layers | 90.51 | 76.79 | 68.79 |
| | Shallow CNN, 13 Layers | 68.41 | 69.22 | 64.54 |

*Bold values represent the highest classification accuracy.*

# 3. RESULT AND DISCUSSION

## 3.1. Learning of Intermediate Layers

A CNN model is expected to extract features from the foreground of the images (Xiao et al., 2020). The foreground of an image is the object that we are performing the task on. Hence we first examined GBP visualization of the features being learned by the intermediate layers in various models.

**Figures 1**, **3** show the GBP visualization of the consecutive layers of different models for the Weedling and Plant Village datasets, respectively. After inspecting the visualized features, we can see that in **Figure 1** the last convolutional layer of ResNet models (ResNet-50 and 2-Conv-ResNet) extracted features from the plant leaf, soil, and plant pot (zoom the figure for a better view) based on which the classification is performed. There is also the influence of background features on the ResNet models, although it is not very strong. On the other hand, the shallow models (Shallow CNN, 6 Layers and Shallow CNN, 13 Layers) learned features from the plant leaf and soil. There is no visible influence of the background features.

For the Plant Village dataset (**Figure 3**), the features extracted by both ResNet models are strongly influenced by the background of the image. The background of the Plant Village dataset consisted of grainy structures, which might have forced the ResNet models to extract features from them. However, the shallow CNN models only extracted features from the leaf edge. If we closely inspect the visualized features, we can see that the shallow models also extracted features from the leaf's veins. In contrast, the ResNet models also depended on the leaf pixels. Between the ResNet models, the ResNet-50 extracted more features from the image background than the 2-Conv-ResNet. This observation is consistent for both datasets.

Analyzing **Figures 1**, **3**, one can observe that the variety of features learned by a model depends on the capacity of the model. ResNet-50 has the highest representational capacity, followed by the 2-Conv-ResNet, Shallow CNN, 13 Layers, and Shallow CNN, 6 Layers. The figures show that the variety of extracted features was higher for the ResNet-50 than other models. Although the background seems uniform in the Weedling dataset, the lighting condition varied for different images. The presence of irregular bright patches might have been deemed as a feature to the ResNet-50 model, which it learned due to its higher representational capacity. The extraction of such features decreased with the decrease of the model capacity. The same trend was followed by the models used for the Plant Village dataset. The analysis of **Figures 1**, **3** reveal that the features learned by a model is influenced by the model's capacity (**RQ1**).

## 3.2. Contribution of Model Depth to Performance

When designing a CNN model, a common practice is to increase the depth of the model to achieve better performance. In this section, we studied whether increasing the depth of the model helps learn better features. From **Figures 1**, **3**, we can see that the ResNet-50 models have the highest number of

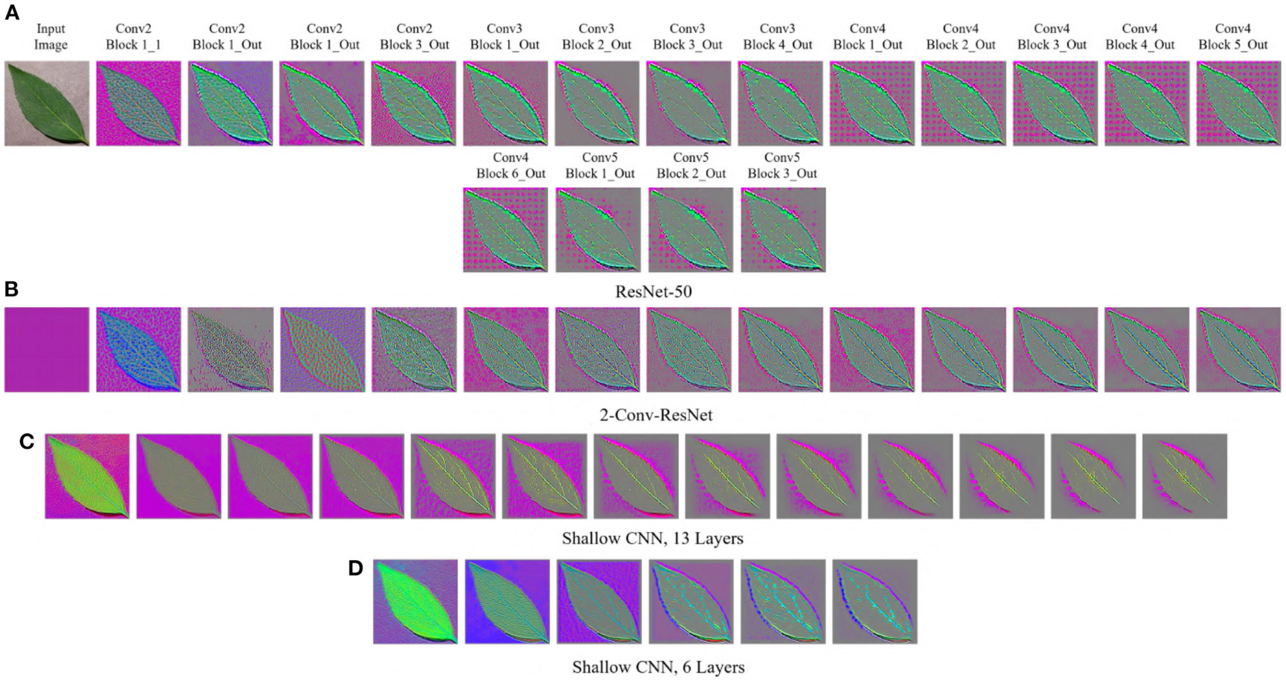


**FIGURE 3 |** GBP-based visualization of the intermediate layers (left to right) of different CNN models for Bean leaf of the Plant Village dataset. The top-left image of **(A)** is the input image for all models. **(A)** ResNet-50. **(B)** 2-Conv-ResNet. **(C)** Shallow CNN, 13 layers. **(D)** Shallow CNN, 6 layers.

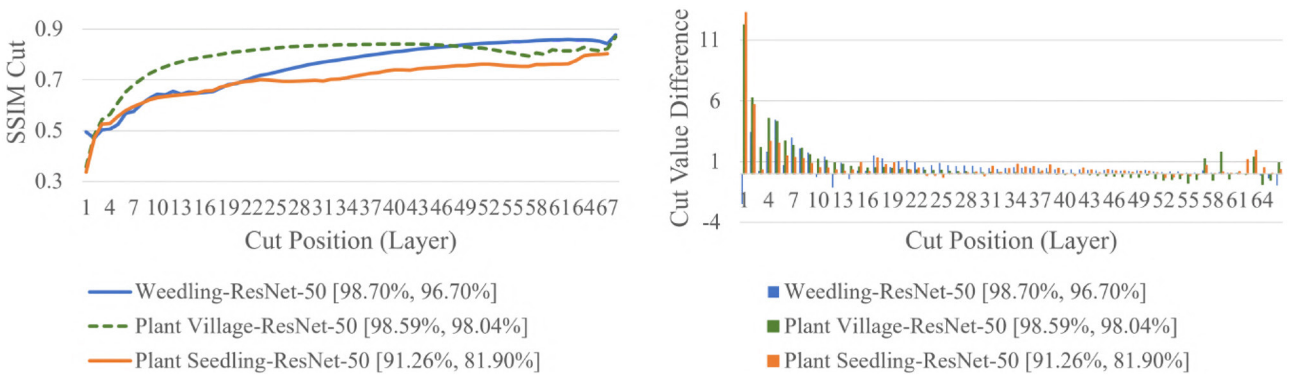


**FIGURE 4 |** Comparison of the Cut Position (Layer) VS SSIM cut curve, and Cut Value Difference for the ResNet-50 models for different datasets. The value in the legend of the chart indicate the training and testing accuracy of the model.

convolutional layers. Examining the features extracted by the convolutional layers, it is evident that for the ResNet-50 model after a certain depth, GBP visualizations are similar. However, for the 2-Conv-ResNet, and both shallow models, the GBP visualizations were dissimilar across all the layers. To quantify the similarity of features extracted by different layers, we propose SSIM cut curve. For every class in a dataset, we randomly selected an image from the testing set and calculated the SSIM cut values for the images (see Section 2.2). Next, we averaged the SSIM cut values over all the images for every layer of a CNN model. Thus for every model, we ended up with an SSIM cut curve.

The SSIM cut curve resembles the "elbow method," commonly used in cluster analysis (Ketchen and Shook, 1996) to choose the number of clusters that optimize the clustering cost. For the SSIM cut curve, the elbow of the curve is a point where moving the cut position more to the right no longer improves the SSIM cut value significantly. **Figure 4** shows the SSIM cut curves of ResNet-50 for different datasets. Initially, every SSIM cut curve shows a sharp positive slope, which indicates the feature visualizations for the initial layers are very dissimilar from the rest of the layers. The slope becomes flatter with the increase in cut position. Thus, in a model with many convolutional layers, the feature visualizations obtained from the shallow layers are

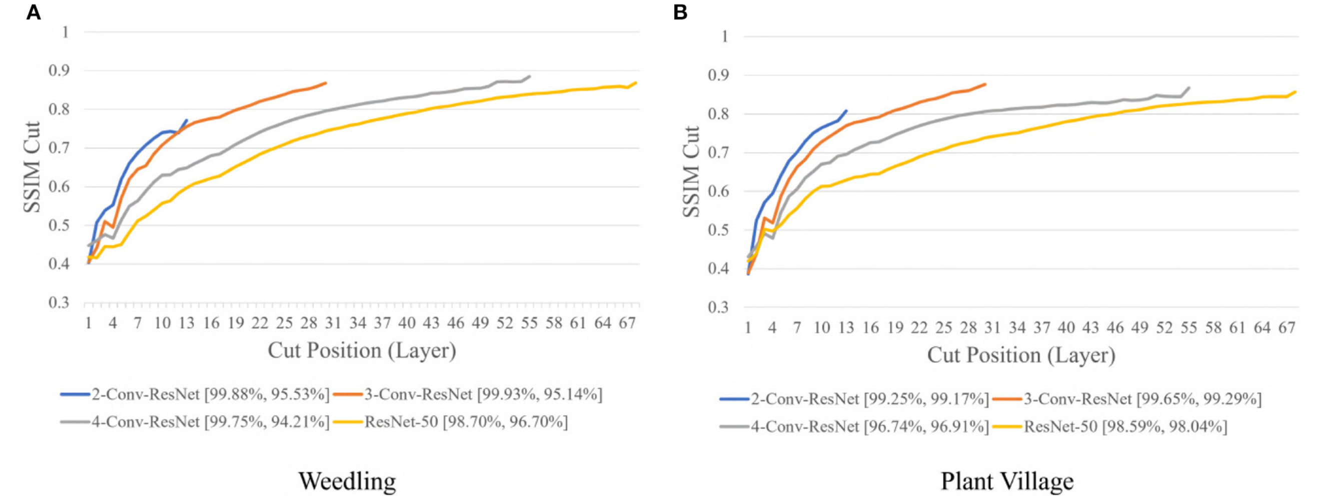


**FIGURE 5 |** Comparison of the Cut Position (Layer) VS SSIM cut curve for ResNet models for **(A)** Weedling and **(B)** Plant Village dataset for Epoch (Best).

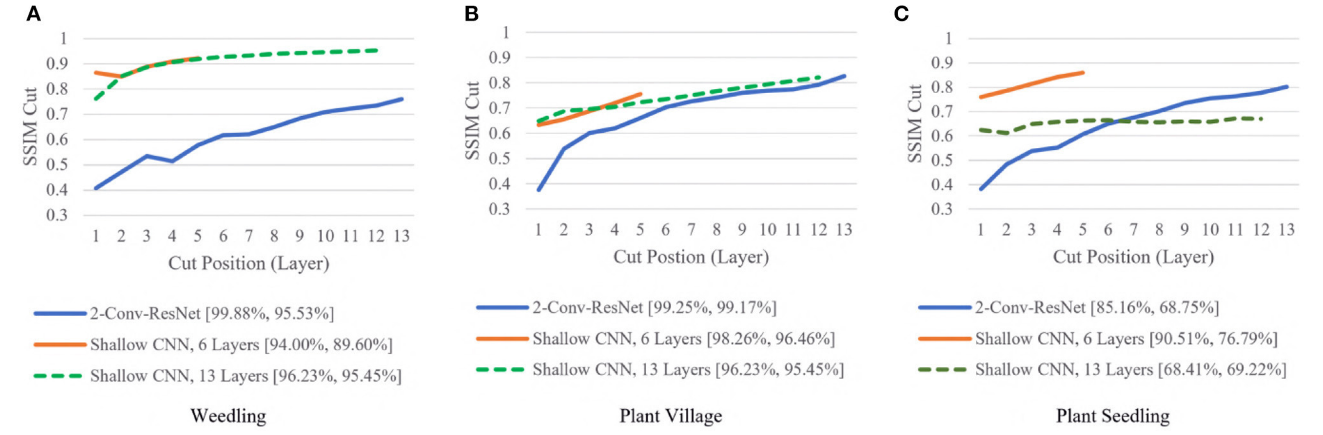


**FIGURE 6 |** Comparison of the Cut Position (Layer) VS SSIM cut curve for the 2-Conv-ResNet, Shallow CNN, 6 Layers, and Shallow CNN, 13 Layers for different datasets. The value in the legend of the chart indicate the training and testing accuracy of the model. **(A)** Weedling. **(B)** Plant village. **(C)** Plant seedling.

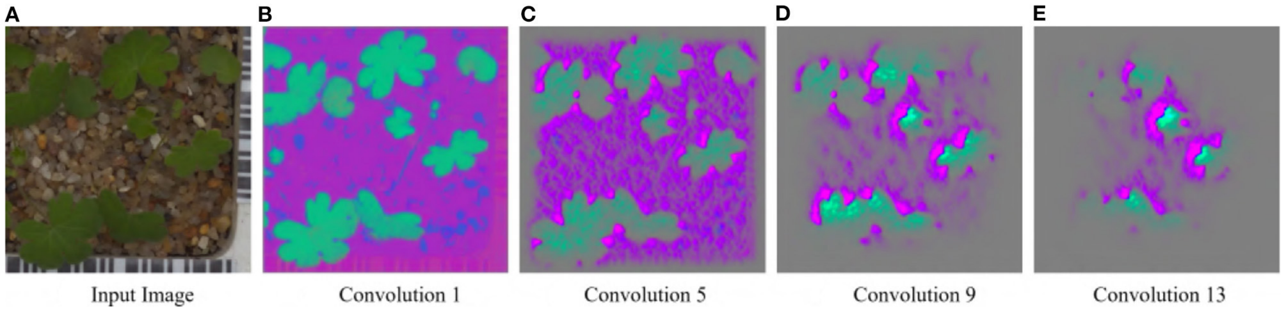


**FIGURE 7 |** GBP visualization of the convolutional layers of the Small-flowered Cranesbill of the Plant Seedling dataset for Shallow CNN, 13 Layers. **(A)** Input image. **(B)** Convolution 1. **(C)** Convolution 5. **(D)** Convolution 9. **(E)** Convolution 13.

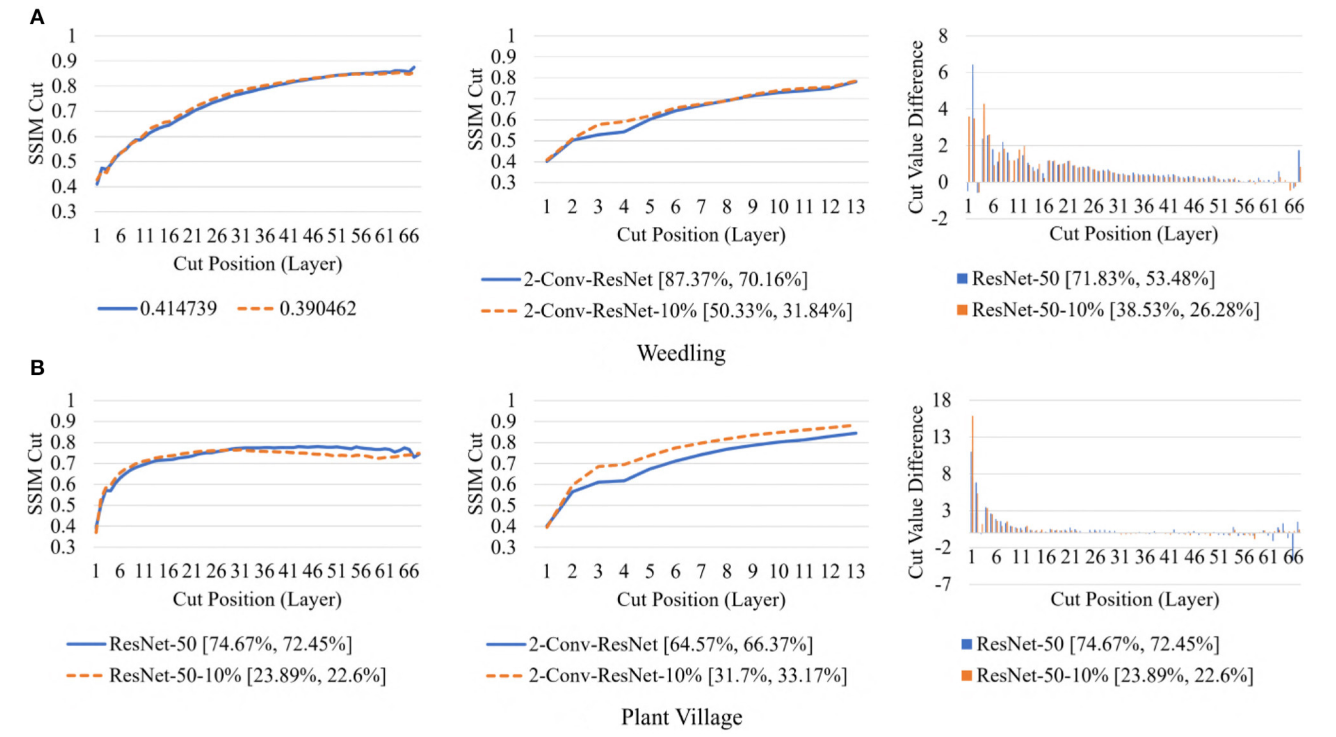


**FIGURE 8 |** Comparison of the Cut Position (Layer) VS SSIM cut curve for the ResNet-50, ResNet-50-10%, 2-Conv-ResNet, and 2-Conv-ResNet-10% models, and Cut Value Difference for the ResNet-50, ResNet-50-10% for different datasets. The value in the legend of the chart indicate the training and testing accuracy of the model. **(A)** Weedling. **(B)** Plant village.

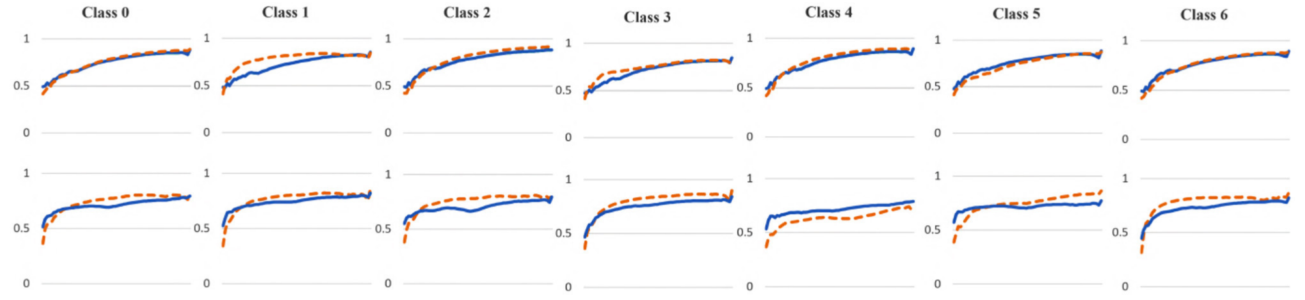


**FIGURE 9 |** Comparison of the Cut Position (Layer) VS SSIM cut curve for different classes of the ResNet-50 (blue), and ResNet-50-10% (orange) models for (top row) Weedling and (bottom row) Plant Village dataset.

more diverse than those from the deeper layers. Furthermore, the diversity of the feature visualizations at a deeper layer is larger in a balanced model compared to those in an overfit model (**RQ2**).

To evaluate whether the SSIM cut curve's elbow point could be used as a guide for selecting the depth of the model, we examined the performances of truncated ResNet-50 (i.e., 2-Conv-ResNet) for the same datasets. We observed that (**Table 1**) 2-Conv-ResNet achieved similar performance when compared with ResNet-50 for the Weedling dataset and even better performance for the Plant Village dataset. For the Seedling dataset, both the ResNet-50 and 2-Conv-ResNet remained overfit.

Next, we varied the number of blocks (see Section 2.5) in the ResNet-50 model to investigate the relation between the depth of a model and the performance of the model and also to see whether SSIM cut curve can help select the model depth more precisely. From **Figure 5**, we can see that the 2-Conv-ResNet and 3-Conv-ResNet have a sharper positive slope than 4-Conv-ResNet and ResNet-50 for both datasets. A flat SSIM cut curve indicates that the convolutional layers are learning less diverse features, indicating that a higher depth model will not always perform better. **Table 1** supports this observation as compared to the larger 4-Conv-ResNet and ResNet-50; the 2-Conv-ResNet and 3-Conv-ResNet performed similarly for the Weedling dataset and better for the Plant Village dataset.

To examine whether shallow models could achieve high performances, we compared the SSIM cut curve for the

2-Conv-ResNet, Shallow CNN with 6 Layers, and Shallow CNN with 13 layers (**Figure 6**).

For the Weedling and Plant Village datasets, the Shallow CNN models achieved comparable performance to the ResNet-50 models. Furthermore, the Shallow CNN models with 6 layers performed similarly to CNN models with 13 layers. We observed a steady increase in the SSIM cut value in both cases. The Shallow CNN with 13 layers performed poorly for the Seedling dataset and relied on the background features (**Figure 7**). The Shallow CNN with 6 layers was overfit, but its training and test accuracy were higher than Shallow CNN with 13 layers.

To examine the diversity of the feature visualizations between balanced and overfit models, we compared the SSIM cut curve of ResNet-50, ResNet-50-10%, 2-Conv-ResNet, and 2-Conv-ResNet-10% models for the Weedling and Plant Village dataset (**Figure 8**). ResNet-50 and 2-Conv-ResNet models were balanced for both datasets, and ResNet-50-10%, and 2-Conv-ResNet-10% models were overfit. In both cases, the SSIM curve of the overfit model had smaller initial SSIM cut values, which increased more sharply than in the balanced models. This can be observed better using the cut value difference plot. The decrease in the cut value difference was sharper for the overfit models for both datasets. This observation indicates that an overfit model may cease extracting new features earlier than a balanced model. A similar trend can also be seen for the per class analysis in **Figure 9**.

## 3.3. Model Selection Using SSIM Cut Curve

This section discusses whether the SSIM cut curve could be leveraged to select an appropriate ResNet-50 model. An ideal situation would be to check all the SSIM cut curves and select the one that learns over all the layers, i.e., where the SSIM cut curve is constantly rising. However, computing the SSIM cut curves for all the models (2-Conv-ResNet, 3-Conv-ResNet, 4-Conv-ResNet, ResNet-50) is infeasible due to the huge amount of time it requires to train these models. For example, it took 72.81 h and 7.80 h to train ResNet-50 for 100 epochs using the Weedling and Plant Village datasets. However, if we can predict the appropriate depth by examining these models only at their first training epochs, we can reduce the time for selecting an appropriate model. This idea would only work if the SSIM cut curves for various models at the first epoch and the best epoch maintained the same shape and relative ordering of the curves.

The rapid increase in the SSIM cut curve indicates that the feature maps are extracting new and diverse features. The curve's saturation indicates that the feature maps may have stopped extracting additional features or learning very subtle features. To select a preferred depth of a model from the SSIM cut curve, we should consider the layers as long as the cut curve is rising, i.e., cut value differences are large. **Figure 10** illustrates the SSIM cut curves for different models at the first epoch, which have a shape similar to the SSIM cut curves computed from the best epoch, e.g., see **Figure 6**. To precisely find the desired model depth, we can use the cut value difference curve and the SSIM cut curve. For the Weedling dataset, in **Figure 12A**, the change of cut value difference is much less after layer 31, which indicates that the rest of the layers might be redundant. A model with around 31 layers is likely to be able to replicate the performance of the ResNet-50 model. From **Figures 6**, **10**, we can see that the SSIM cut curve of the 3-Conv-ResNet is consistently rising, with around 31 layers. From both figures, we can see that the change of the SSIM cut values of the 4-Conv-ResNet in later layers is much smaller, showing that the additional depth of the 4-Conv-ResNet fails to help learn additional features. The SSIM cut curve of the 2-Conv-ResNet always increases, indicating that the feature maps are still learning features, and more layers can help learn additional features. Based on these observations, we can say that the 3-Conv-ResNet is the preferred model for the Weedling dataset.

For the Plant Village dataset, the cut value difference of **Figure 12B** suggests that a model with around 16 layers should be sufficient to perform the task. In **Figures 6**, **10**, the increasing SSIM cut curve of the 2-Conv-ResNet and 3-Conv-ResNet also supports these models being chosen as the preferred model. From **Table 1**, it is evident that the 3-Conv-ResNet outperforms all the models. So, the SSIM cut curve can help us choose the depth of a model.

We now examine the behavior of the SSIM cut curve over various epochs. **Figure 11** shows the SSIM cut curve of ResNet-50 and 2-Conv-ResNet models for Weedling and Plant Village datasets at different epochs. From **Figure 11**, we can see that the SSIM cut curves are similar at different epochs for a model, and the pattern is consistent for both models and both datasets. For both datasets, the SSIM cut curve of the ResNet-50 model suggests an earlier elbow point, whereas the 2-Conv-ResNet shows a steady increase. The stability of the SSIM cut at various epochs gives further evidence that relying on the first training epoch would be sufficient for the SSIM cut curve based model selection.

Next, in **Figure 12**, we compared the SSIM cut curves for the first epoch of ResNet-50, ResNet-50-10%, 2-Conv-ResNet, and 2-Conv-ResNet-10% models along with their cut value difference plots for the ResNet-50 and ResNet-50-10%. From **Figures 8**, **12**, it is evident that shape and relative ordering of the SSIM-cut curves obtained at the first epoch (**Figure 12**) is consistent with the model's best epoch (**Figure 8**).

Since the empirical results show that the SSIM cut curve follows the same trend throughout the training of a CNN, we can choose the preferred model depth by examining the SSIM cut curves for various models at the beginning of their training. So, the SSIM cut curve of the feature map visualizations may help detect potential underfitting or overfitting in the CNN models and allow us to select a balanced model (**RQ3**).

## 3.4. Early Stopping VS SSIM Cut Curve

There are several ways to avoid overfitting, and early stopping is one of them (Ying, 2019). Early stopping is a form of regularization that is used during training in iterative methods to select how long a model is going to be trained (Girosi et al., 1995; Prechelt, 1998). In early stopping, it is possible that the learning of the model is stopped before it is fully optimized (Caruana et al., 2000), which may prevent the model from making accurate predictions. Early stopping is insensitive to

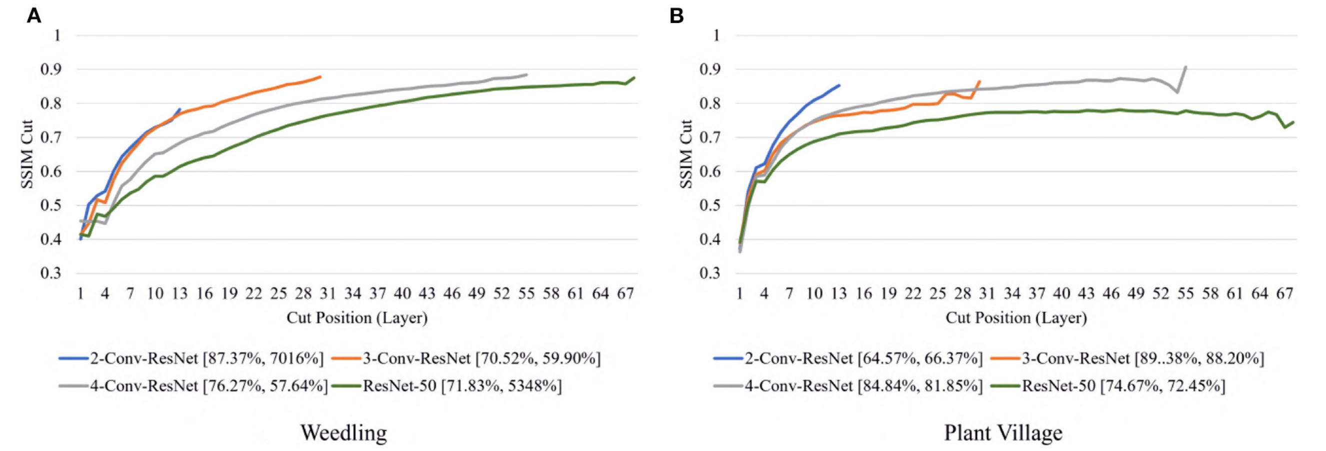


**FIGURE 10 |** Comparison of the Cut Position (Layer) VS SSIM cut curve for ResNet models for **(A)** Weedling and **(B)** Plant Village dataset at Epoch (1). The value in the legend of the chart indicate the training and testing accuracy of the model.

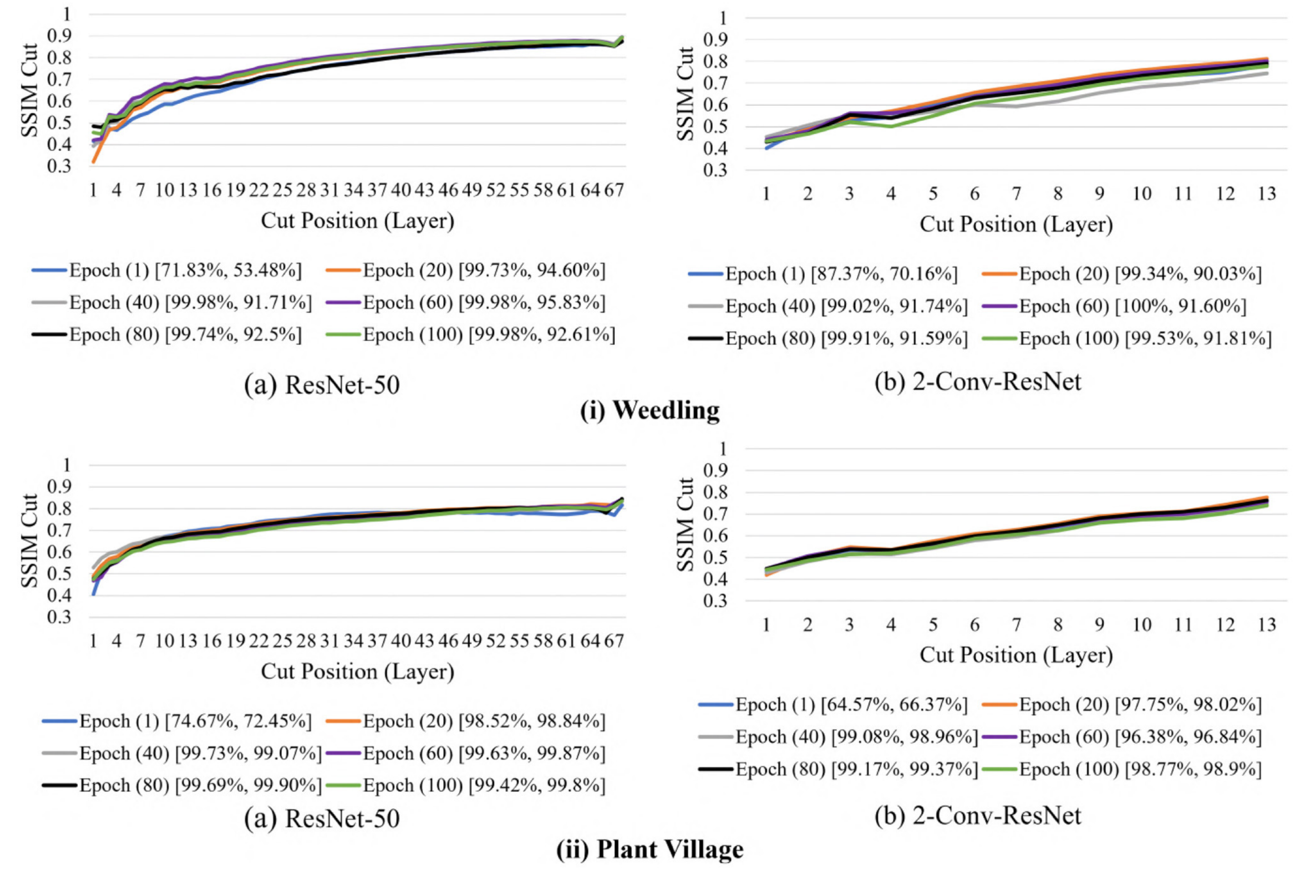


**FIGURE 11 |** Comparison of the Cut Position (Layer) VS SSIM cut for different epochs of **(i)** Weedling and **(ii)** Plant Village dataset using (a) ResNet-50 and (b) 2-Conv-ResNet. The value in the legend of the chart indicate the training and testing accuracy of the model.

the capacity of the model. As a result, the training of the smaller models can be stopped before it is optimized (Caruana et al., 2000).

The advantage of the SSIM cut curve is that it helps select an optimized model. By looking at the SSIM cut of models of various depth, we propose the depth that is likely to provide better

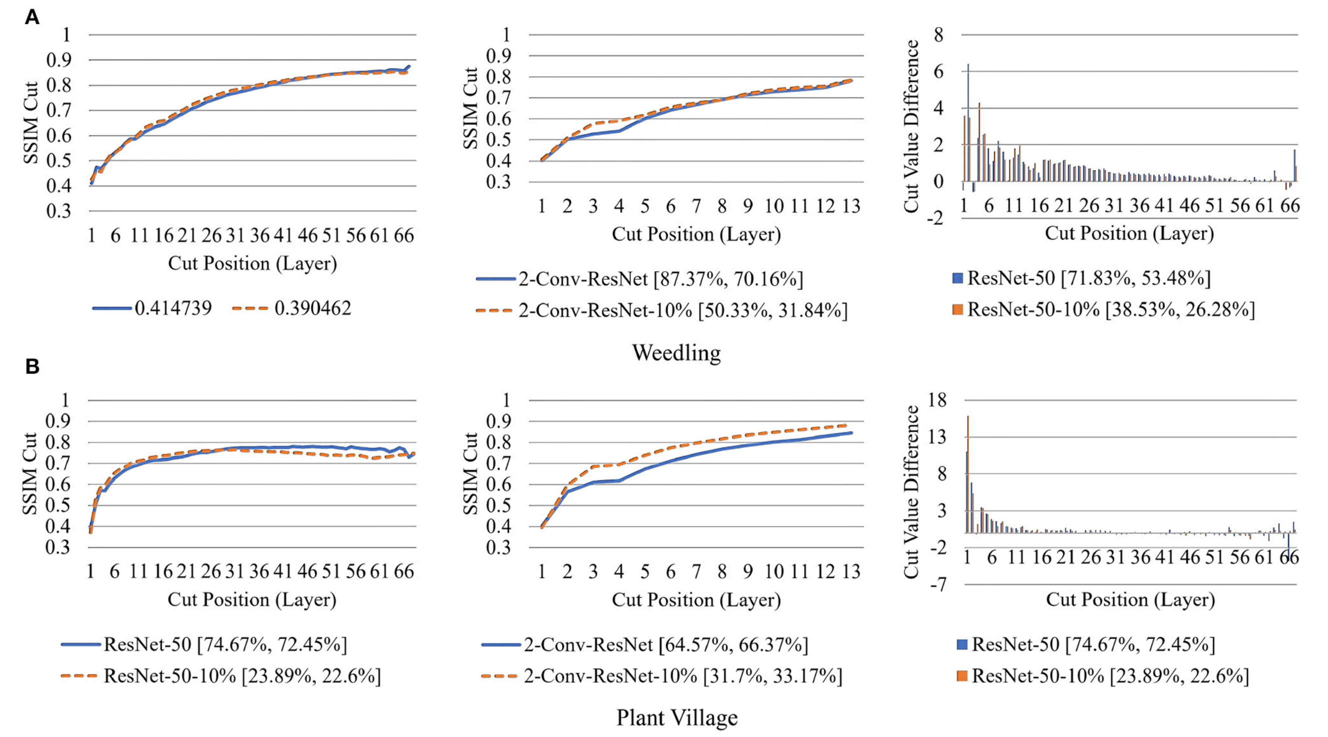


**FIGURE 12 |** Comparison of the Cut Position (Layer) VS SSIM cut curve for the ResNet-50, ResNet-50-10%, 2-Conv-ResNet, and 2-Conv-ResNet-10% models, and Cut Value Difference for the ResNet-50, ResNet-50-10% for different datasets at Epoch (1). The value in the legend of the chart indicate the training and testing accuracy of the model. **(A)** Weedling. **(B)** Plant village.

accuracy. For a fixed model, the shape of the SSIM cut curve remains similar whether we stop early or not (see **Figures 6**, **10**). Therefore, we look at the SSIM cut curves of models built after the first epoch to expedite the process.

Note that early stopping may be used while training the model to expedite the model selection by generating SSIM cut curves. However, here we only used one training epoch to generate the curves. Once we select an appropriate model by examining the generated SSIM cut curves, we do not use any early stopping criteria on the selected model.

## 3.5. Summary

In summary, our experimental results suggest that the extraction of features of a deep learning model depends on the capacity of the model (**RQ1**). Our analysis of the SSIM curve shows that the GBP visualizations of the initial convolutional layers of a model are much more diverse than the GBP visualizations for the deeper layers (**RQ2**). Furthermore, the rate of change and the SSIM cut curve's elbow point can be used for model selection (**RQ3**). Since the SSIM cut curve is consistent for a model throughout different training epochs, we can use it to choose a model depth at an early training stage. This can save a lot of time in a traditional approach that compares models after fully training them.

## 3.6. Testing With Segmented Images

**Table 2** illustrates the accuracy of different models on the Weedling and Plant Village dataset for segmented images. For segmentation, we retained all the green pixels in the image and marked the rest of the pixels as black (see **Supplementary Material** for examples of segmented images). Finally, we ended up with images where only the leaf was present. Next, we used the pre-trained models on the segmented images to calculate the accuracy.

From **Table 2**, we can see that the 3-Conv-ResNet has higher classification accuracy than other models for both Weedling and Plant Village dataset, which implies that the models are more focused on the leaf features than the background features. Also, the low accuracy of the ResNet-50 model indicates background features may more influence it than the 3-Conv-ResNet model. Comparing the accuracy of Epoch (1) to the accuracy of Epoch (Best), we can see that as the training progresses, the models tend to focus more on the leaf features than the background features. However, the results of such experiments can be limited by the quality of the segmented images.

# 4. CONCLUSION

In this article, we explained the overfitting in a CNN model for plant phenotyping by visualizing the intermediate layers' learning. We used guided backpropagation to visualize the learning of the intermediate layer of different CNN models. We used four different models on three different plant classification datasets. We proposed a novel SSIM cut based analysis to measure the similarity among the features learned in the

**TABLE 2 |** Performances of ResNet-50 model with different block for various datasets with segmented images.

| | Epoch (best) | | Epoch (1) | |
|---|---|---|---|---|
| | Training dataset (%) | Testing dataset (%) | Training dataset (%) | Testing dataset (%) |
| **Weedling** | | | | |
| ResNet-50 | 58.87 | 53.61 | 14.01 | 11.99 |
| ResNet-50-10% | 41.95 | 35.49 | 28.43 | 25.60 |
| 2-Conv-ResNet | 33.56 | 24.74 | 29.39 | 25.88 |
| 2-Conv-ResNet-10% | 27.61 | 18.66 | 27.82 | 17.26 |
| 3-Conv-ResNet | **65.50** | **66.64** | 14.08 | 12.08 |
| 4-Conv-ResNet | 30.77 | 23.38 | 13.97 | 11.97 |
| **Plant village** | | | | |
| ResNet-50 | 28.00 | 23.68 | 10.60 | 10.50 |
| ResNet-50-10% | 31.45 | 27.80 | 10.08 | 11.54 |
| 2-Conv-ResNet | 40.78 | 34.71 | 28.66 | 25.35 |
| 2-Conv-ResNet-10% | 24.03 | 23.37 | 28.66 | 25.35 |
| 3-Conv-ResNet | **61.72** | **62.94** | 30.94 | 26.52 |
| 4-Conv-ResNet | 43.90 | 38.09 | 21.48 | 20.56 |

*Bold values represent the highest classification accuracy.*

intermediate layers of a CNN. Our experiments showed that the features extracted by a model depend on its capacity. Our SSIM cut curve revealed that in a more complex model, the shallow layers learn more diverse features as compared to the deeper layers and that a more distinct transition between these regimes is noticeable for overfit models. The SSIM cut curve method can help detect a potential overfit condition or inform a practitioner that a shallower model may be more appropriate for training with a particular dataset. We also showed the usage of the SSIM cut curve in selecting the model depth. It can help reduce a model's training time and resource as we can predict the required model depth at the beginning of training. We believe our study contributes to a better understanding of the behavior of overfit CNN models and provides new directions for creating metrics to detect and avoid model overfitting in plant phenotyping tasks.

Future works may further examine various facets of our SSIM cut curve based analysis. In our SSIM cut curve analysis, the elbow point may not always correspond to a sharp elbow or be identified unambiguously in practice, which is a commonly known limitation of elbow heuristics (Ketchen and Shook, 1996). We envision running a user study involving deep learning experts, where one can show the output of different models by hiding the model's label and recording their opinions to see whether a domain expert can detect an overfit model by only observing the GBP visualization of the intermediate layers. Due to the residual connection in the ResNet models, it might be possible to avoid overfitting and influence the similarity of the GBP visualizations of various layers. Hence it would be interesting to investigate the contribution of the residual connections in an overfit model's performance.

## 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/s.

## AUTHOR CONTRIBUTIONS

SM conceived of the presented idea and performed the computations. SM, DM, and IS developed the theory. DM and IS verified the analytical methods. MB, CB, and CH provided the Weedling dataset and the trained ResNet-50 model for analysis. All authors discussed the results and contributed to the final manuscript.

## FUNDING

This research was undertaken thank in part to funding from the Canada First Research Excellence Fund.

## ACKNOWLEDGMENTS

The content of this manuscript has been presented [IN PART] at the Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, 2021, 1362-1371 (Mostafa et al., 2021).

## SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frai.2022.871162/full#supplementary-material

## REFERENCES

Aich, S., Josuttes, A., Ovsyannikov, I., Strueby, K., Ahmed, I., Duddu, H. S., et al. (2018). "Deepwheat: estimating phenotypic traits from crop images with deep learning," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)* (Lake Tahoe, NV: IEEE), 323–332. doi: 10.1109/WACV.2018.00042

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

Azlah, M. A. F., Chua, L. S., Rahmad, F. R., Abdullah, F. I., and Wan Alwi, S. R. (2019). Review on techniques for plant leaf classification and recognition. *Computers* 8, 77. doi: 10.3390/computers8040077

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* 10, e0130140. doi: 10.1371/journal.pone.0130140

Beck, M. A., Liu, C.-Y., Bidinosti, C. P., Henry, C. J., Godee, C. M., and Ajmani, M. (2020). An embedded system for the automated generation of labeled plant images to enable machine learning applications in agriculture. *PLoS ONE* 15, e0243923. doi: 10.1371/journal.pone.0243923

Beck, M. A., Liu, C.-Y., Bidinosti, C. P., Henry, C. J., Godee, C. M., and Ajmani, M. (2021). Weed seedling images of species common to Manitoba, Canada. doi: 10.5061/dryad.gtht76hhz

Caruana, R., Lawrence, S., and Giles, L. (2000). "Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping," in *Advances in Neural Information Processing Systems*, eds T. Leen, T. Dietterich and V. Tresp (MIT Press), 13. Available online at: https://proceedings.neurips.cc/paper/2000/file/059fdcd96baeb75112f09fa1dcc740cc-Paper.pdf

Chandra, A. L., Desai, S. V., Guo, W., and Balasubramanian, V. N. (2020). Computer vision with deep learning for plant phenotyping in agriculture: a survey. *arXiv[Preprint].arXiv:2006.11391*. Available online at: https://arxiv.org/abs/2006.11391

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process. Mag*. 29, 141–142. doi: 10.1109/MSP.2012.2211477

Dobrescu, A., Giuffrida, M. V., and Tsaftaris, S. A. (2019). "Understanding deep neural networks for regression in leaf counting," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, 2600–2608. doi: 10.1109/CVPRW.2019.00316

Dobrescu, A., Valerio Giuffrida, M., and Tsaftaris, S. A. (2017). "Leveraging multiple datasets for deep leaf counting," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, Venice, 2072–2079. doi: 10.1109/ICCVW.2017.243

Dyrmann, M., Karstoft, H., and Midtiby, H. S. (2016). Plant species classification using deep convolutional neural network. *Biosyst. Eng*. 151, 72–80. doi: 10.1016/j.biosystemseng.2016.08.024

Escorcia, V., Carlos Niebles, J., and Ghanem, B. (2015). "On the relationship between visual attributes and convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1256–1264. doi: 10.1109/CVPR.2015.7298730

Feldman, V. (2020). "Does learning require memorization? A short tale about a long tail," in *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, 954–959. doi: 10.1145/3357713.3384290

Feldman, V., and Zhang, C. (2020). What neural networks memorize and why: discovering the long tail via influence estimation. *arXiv[Preprint].arXiv:2008.03703*. Available online at: https://arxiv.org/abs/2008.03703

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.

Ghosal, S., Blystone, D., Singh, A. K., Ganapathysubramanian, B., Singh, A., and Sarkar, S. (2018). An explainable deep machine vision framework for plant stress phenotyping. *Proc. Natl. Acad. Sci. U.S.A*. 115, 4613–4618. doi: 10.1073/pnas.1716999115

Gigante, S., Charles, A. S., Krishnaswamy, S., and Mishne, G. (2019). Visualizing the phate of neural networks. *arXiv[Preprint].arXiv:1908.02831*. Available online at: https://arxiv.org/abs/1908.02831

Girosi, F., Jones, M., and Poggio, T. (1995). Regularization theory and neural networks architectures. *Neural Comput*. 7, 219–269. doi: 10.1162/neco.1995.7.2.219

Giselsson, T. M., Dyrmann, M., Jørgensen, R. N., Jensen, P. K., and Midtiby, H. S. (2017). A public image database for benchmark of plant seedling classification algorithms. *arXiv[Preprint].arXiv:1711.05458*. doi: 10.48550/arXiv.1711.05458

Hati, A. J., and Singh, R. R. (2021). Artificial intelligence in smart farms: plant phenotyping for species recognition and health condition identification using deep learning. *AI* 2, 274–289. doi: 10.3390/ai2020017

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 770–778. doi: 10.1109/CVPR.2016.90

Humphrey, G. B., Maier, H. R., Wu, W., Mount, N. J., Dandy, G. C., Abrahart, R. J., et al. (2017). Improved validation framework and r-package for artificial neural network models. *Environ. Modell. Softw*. 92, 82–106. doi: 10.1016/j.envsoft,.2017.01.023

Kamal, K. C., Yin, Z., Li, D., and Wu, Z. (2021). Impacts of background removal on convolutional neural networks for plant disease classification *in-situ*. *Agriculture* 11, 827. doi: 10.3390/agriculture11090827

Ketchen, D. J., and Shook, C. L. (1996). The application of cluster analysis in strategic management research: an analysis and critique. *Strat. Manage. J*. 17, 441–458. doi: 10.1002/(SICI)1097-0266(199606)17:6<441::AID-SMJ819>3.0.CO;2-G

Lu, H., Liu, L., Li, Y.-N., Zhao, X.-M., Wang, X.-Q., and Cao, Z.-G. (2021). TasselNETV3: explainable plant counting with guided upsampling and background suppression. *IEEE Trans. Geosci. Remote Sens*. 60, 1–15. doi: 10.1109/TGRS.2021.3058962

Lundberg, S., and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *arXiv[Preprint].arXiv:1705.07874*. Available online at: https://arxiv.org/abs/1705.07874

Mohanty, S. P. (2018). *Plant Village*. Available online at: https://github.com/spMohanty/PlantVillage-Dataset

Mohanty, S. P., Hughes, D. P., and Salathe, M. (2016). Using deep learning for image-based plant disease detection. *Front. Plant Sci*. 7, 1419. doi: 10.3389/fpls.2016.01419

Montavon, G., Binder, A., Lapuschkin, S., Samek, W., and Müller, K.-R. (2019). "Layer-wise relevance propagation: an overview," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, ed. Springer (Cham: Springer), 193–209. doi: 10.1007/978-3-030-28954-6_10

Mostafa, S., and Mondal, D. (2021). On the evolution of neuron communities in a deep learning architecture. *arXiv[Preprint].arXiv:2106.04693*. Available online at: https://arxiv.org/abs/2106.04693

Mostafa, S., Mondal, D., Beck, M., Bidinosti, C., Henry, C., and Stavness, I. (2021). "Visualizing feature maps for model selection in convolutional neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 1362–1371. doi: 10.1109/ICCVW54120.2021.00157

Nagasubramanian, K., Jones, S., Singh, A. K., Sarkar, S., Singh, A., and Ganapathysubramanian, B. (2019). Plant disease identification using explainable 3D deep learning on hyperspectral images. *Plant Methods* 15, 1–10. doi: 10.1186/s13007-019-0479-8

Oh, S. J., Schiele, B., and Fritz, M. (2019). "Towards reverse-engineering black-box neural networks," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning* (Springer), 121–144. doi: 10.1007/978-3-030-28954-6_7

Prechelt, L. (1998). "Early stopping-but when?" in *Neural Networks: Tricks of the Trade*, ed. Springer (Cham: Springer), 55–69. doi: 10.1007/3-540-49430-8_3

Reed, R., and Marks, R. J. II (1999). *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. MIT Press. doi: 10.7551/mitpress/4937.001.0001

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). ""Why should I trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, 1135–1144. doi: 10.1145/2939672.2939778

Salman, S., and Liu, X. (2019). Overfitting mechanism and avoidance in deep neural networks. *arXiv[Preprint].arXiv:1901.06566*. doi: 10.48550/arXiv.1901.06566

Scharr, H., Minervini, M., French, A. P., Klukas, C., Kramer, D. M., Liu, X., et al. (2016). Leaf segmentation in plant phenotyping: a collation study. *Mach. Vision Appl*. 27, 585–606. doi: 10.1007/s00138-015-0737-3

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*, Venicem, 618–626. doi: 10.1109/ICCV.2017.74

Selvaraju, R. R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., and Batra, D. (2016). GRAD-CAM: why did you say that? *arXiv[Preprint].arXiv:1611.07450*. doi: 10.48550/arXiv.1611.07450

Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: visualising image classification models and saliency maps. *arXiv[Preprint].arXiv:1312.6034*. doi: 10.48550/arXiv.1312.6034

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2014). Striving for simplicity: the all convolutional net. *arXiv[Preprint].arXiv:1412.6806*. Available online at: https://doi.org/10.48550/arXiv.1412.6806

Toneva, M., Sordoni, A., Combes, R. T., Trischler, A., Bengio, Y., and Gordon, G. J. (2018). An empirical study of example forgetting during deep neural network learning. *arXiv[Preprint].arXiv:1812.05159*. doi: 10.48550/arXiv.1812.05159

Tzeng, F.-Y., and Ma, K.-L. (2005). "Opening the black box - data driven visualization of neural networks," in *VIS 05. IEEE Visualization, 2005*, 383–390. doi: 10.1109/VISUAL.2005.1532820

Ubbens, J., Cieslak, M., Prusinkiewicz, P., and Stavness, I. (2018). The use of plant models in deep learning: an application to leaf counting in rosette plants. *Plant Methods* 14, 1–10. doi: 10.1186/s13007-018-0273-z

Ubbens, J. R., and Stavness, I. (2017). Deep plant phenomics: a deep learning platform for complex plant phenotyping tasks. *Front. Plant Sci*. 8, 1190. doi: 10.3389/fpls.2017.01190

Weis, M., Gutjahr, C., Ayala, V. R., Gerhards, R., Ritter, C., and Schölderle, F. (2008). Precision farming for weed management: techniques. *Gesunde Pflanzen* 60, 171–181. doi: 10.1007/s10343-008-0195-1

Weng., Y., Zeng, R., Wu, C., Wang, M., Wang, X., and Liu, Y. (2019). A survey on deep-learning-based plant phenotype research in agriculture. *Sci. Sin. Vitae* 49, 698–716. doi: 10.1360/SSV-2019-0020

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv[Preprint].arXiv:1708.07747*. doi: 10.48550/arXiv.1708.07747

Xiao, K., Engstrom, L., Ilyas, A., and Madry, A. (2020). Noise or signal: the role of image backgrounds in object recognition. *arXiv[Preprint].arXiv:2006.09994*. doi: 10.48550/arXiv.2006.09994

Ying, X. (2019). An overview of overfitting and its solutions. *J. Phys. Conf. Ser*. 1168, 022022. doi: 10.1088/1742-6596/1168/2/022022

Zeiler, M. D., and Fergus, R. (2014). "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision* (Zurich: Springer), 818–833. doi: 10.1007/978-3-319-10590-1_53

Zenkl, R., Timofte, R., Kirchgessner, N., Roth, L., Hund, A., Van Gool, L., et al. (2022). Outdoor plant segmentation with deep learning for high-throughput field phenotyping on a diverse wheat dataset. *Front. Plant Sci*. 12, 774068. doi: 10.3389/fpls.2021.774068

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*, Las Vegas, NV, 2921–2929. doi: 10.1109/CVPR.2016.319

**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.

ORIGINAL RESEARCH
published: 24 May 2022
doi: 10.3389/fpls.2022.846767

# CASM-AMFMNet: A Network Based on Coordinate Attention Shuffle Mechanism and Asymmetric Multi-Scale Fusion Module for Classification of Grape Leaf Diseases

*Jiayu Suo[1]†, Jialei Zhan[1]†, Guoxiong Zhou[1]*, Aibin Chen[1], Yaowen Hu[1], Weiqi Huang[1], Weiwei Cai[1], Yahui Hu[2] and Liujun Li[3]*

*[1] College of Computer and Information Engineering, Central South University of Forestry and Technology, Changsha, China, [2] Plant Protection Research Institute, Hunan Academy of Agricultural Sciences (HNAAS), Changsha, China, [3] Department of Civil, Architectural and Environmental Engineering, Missouri University of Science and Technology, Rolla, MO, United States*

OPEN ACCESS

**Edited by:**
*Ian Stavness,*
*University of Saskatchewan, Canada*

**Reviewed by:**
*Marcin Wozniak,*
*Silesian University of Technology, Poland*
*Shangpeng Sun,*
*McGill University, Canada*

***Correspondence:**
*Guoxiong Zhou*
*zhougx01@163.com*

*†These authors have contributed equally to this work*

**Specialty section:**
*This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science*

**Received:** *31 December 2021*
**Accepted:** *24 March 2022*
**Published:** *24 May 2022*

**Citation:**
*Suo J, Zhan J, Zhou G, Chen A, Hu Y, Huang W, Cai W, Hu Y and Li L (2022) CASM-AMFMNet: A Network Based on Coordinate Attention Shuffle Mechanism and Asymmetric Multi-Scale Fusion Module for Classification of Grape Leaf Diseases.*
*Front. Plant Sci. 13:846767.*
*doi: 10.3389/fpls.2022.846767*

Grape disease is a significant contributory factor to the decline in grape yield, typically affecting the leaves first. Efficient identification of grape leaf diseases remains a critical unmet need. To mitigate background interference in grape leaf feature extraction and improve the ability to extract small disease spots, by combining the characteristic features of grape leaf diseases, we developed a novel method for disease recognition and classification in this study. First, Gaussian filters Sobel smooth de-noising Laplace operator (GSSL) was employed to reduce image noise and enhance the texture of grape leaves. A novel network designated coordinated attention shuffle mechanism-asymmetric multi-scale fusion module net (CASM-AMFMNet) was subsequently applied for grape leaf disease identification. CoAtNet was employed as the network backbone to improve model learning and generalization capabilities, which alleviated the problem of gradient explosion to a certain extent. The CASM-AMFMNet was further utilized to capture and target grape leaf disease areas, therefore reducing background interference. Finally, Asymmetric multi-scale fusion module (AMFM) was employed to extract multi-scale features from small disease spots on grape leaves for accurate identification of small target diseases. The experimental results based on our self-made grape leaf image dataset showed that, compared to existing methods, CASM-AMFMNet achieved an accuracy of 95.95%, F1 score of 95.78%, and mAP of 90.27%. Overall, the model and methods proposed in this report could successfully identify different diseases of grape leaves and provide a feasible scheme for deep learning to correctly recognize grape diseases during agricultural production that may be used as a reference for other crops diseases.

**Keywords: CASM-AMFMNet, coordinate attention shuffle mechanism asymmetric, multi-scale fusion module, grape leaf diseases, GSSL, image enhancement**

## INTRODUCTION

Grape is a popular fruit worldwide with multiple nutritional components. The active compounds in grape extracts have antioxidant, antibacterial, anti-inflammatory, and anti-carcinogenic activities and thus utilized to generate products that can alleviate and treat hypertension (Sabra et al., 2021). The continuous improvement of living standards and high demand for grapes have been important driving factors in the progressive development of the grape planting industry and growing areas of grape cultivation over recent years. However, grapes are easily susceptible to weather, environmental variables, insect pests, bacteria, and fungi during the cultivation process (Ampatzidis et al., 2017), with frequent risk of black rot, black measles, leaf blight, downy mildew, and other grape leaf diseases that seriously affect growth and contribute significantly to reduction of grape quality and yield, resulting in huge financial losses to farmers.

Infection patterns of grape diseases are usually manifested on the leaves (Chouhan et al., 2020), which can be easily collected and examined to characterize diseased spots. Traditionally, grape leaf diseases are evaluated *via* visual inspection by fruit farmers and plant protection experts (Pound et al., 2017), which is associated with problems of strong subjectivity, slow speed, a high misidentification rate, poor real-time performance, and high dependence on advice by experts (Bock et al., 2010). Since grape leaves display small spot areas in the early stages of disease, manual detection is difficult. In addition, when collecting grape leaf images in the natural environment, some disease spots of the leaves are obscured, resulting in fewer details of features that are identifiable. Evaluation of leaf disease *via* visual inspection is a considerable challenge (Chouhan et al., 2018). However, accurate early identification of the symptoms of grape disease and effective control spread should aid in successfully minimizing losses. Therefore, timely and efficient machine learning methods to identify the disease spots of grape leaves are extremely helpful for farmers to rapidly assess the disease type and extent of infestation. Appropriate prevention and control can reduce the impact of disease, in turn, improving the yield and quality of grapes and safeguarding the economic benefits of fruit farming. At present, three major problems exist in identification of grape leaf diseases. (1) Imaging of grape leaf has issues of edge blurring and noise. For instance, among the grape leaf images we obtained, inconspicuous contrast, blurred edges, and noise were prevalent, which affect leaf recognition by the network, and in severe cases, the recognition and extraction of disease features, leading to inaccurate classification of grape leaf diseases. (2) Images have background interference. When analyzing grape leaf images, shape, size, and color of spots of different diseases are usually extracted. However, complex backgrounds can affect feature extraction. The network extracts the interference factors in the background as features, leading to inaccurate classification. (3) Grape leaf disease spots are extremely small. Since the grape leaves are relatively small and some disease spots themselves are minute at the beginning, small and dense disease spots may also appear on the same leaf, making detection difficult and leading to lack of extracted feature information. Consequently, misclassification of different grape leaf diseases is relatively common.

To solve the problem of blurred edges and noise in grape leaf images, Liu et al. (2018) proposed a novel adaptive-rendering approach based on feature reconstruction to eliminate Monte Carlo noise while preserving image details. However, the edge information of images obtained with this method becomes blurred. Clinton (2017) used Sobel algorithms to detect the edges of blurred images, which improved image quality and facilitated restoration, but the image edges detected with this method were discontinuous, and the lines were thick, resulting in loss of some edge details. Cruz et al. (2017) applied a small-window median filter to remove noise in the leaf image dataset. This method effectively preserved the sharp edges of plant leaves, but the effect of Gaussian noise removal in the background was not ideal. In this study, the Gaussian filters Sobel smooth de-noising Laplace operator (GSSL) algorithm was applied to preprocess the image and process the grape leaf image using multiple steps, including ideal high-pass filter, Sobel operator, and smooth filter. The images obtained exhibited clear edges, obvious contrast, and less noise. At the same time, the texture features of diseased grape leaves were preliminarily enhanced.

To resolve the problem of image background interference, Gao and Lin (2019) proposed an accurate and fully automatic segmentation method for medicinal plant leaf imaging under complex backgrounds. However, this method was not successful when applied to gray images. An algorithm combining simple linear iterative cluster (SLIC) with support vector machine (SVM) was proposed by Sun and colleagues (Huang et al., 2018) to extract a saliency map of tea leaf disease under complex backgrounds (Sun et al., 2019). This procedure uses simple linear iterative clustering for preprocessing to separate significant regions from the background. However, errors can occur when separating the background and disease regions, resulting in loss of a number of the features at the preprocessing stage.

For the problem of small leaf disease spots, Liu et al. (2020a) proposed improved deep convolutional neural networks based on convolutional neural network (CNN) for grape leaf disease recognition using depthwise separable convolution to establish the first two convolutional layers, designated DICNN. Deep separable convolution is used to reduce the model parameters and over-refinement. Next, the concept structure is employed to improve the extraction performance of multi-scale convolution for disease points. Finally, the dense connection strategy is introduced to promote the fusion of multidimensional features between the concept structures for alleviating the problem of gradient disappearance and facilitating feature reuse and propagation. However, when the simplest CNN is used as the model backbone, the gradient descent algorithm can be easily applied to make the training results converge to the local minimum rather than the global minimum. The pooling layer loses considerable valuable information and overlooks the correlation between the local and global layers. On the other hand, in a complex environment, precise disease location of grape leaves is not achieved and the disease can easily be confused with a similar background, resulting in reduced accuracy of identification. The use of deep separable convolution significantly reduces the model parameters but simultaneously decreases the model capacity, leading to lower accuracy of disease recognition. The presence of accumulating perception structures also increases the difficulty of using the model in downstream tasks and amount of calculation. Xie et al. (2020) proposed a rapid DR-IACNN with higher feature extraction capability to identify grape leaves based on the detection algorithms of GLDD and fast R-CNN. Firstly, the use of Resnet improved the backbone. A double RPN structure was designed to achieve better feature extraction of small lesions through upsampling and downsampling. Disease features were extracted by introducing the inception-v1 and inception-ResNet-v2 modules and Se blocks to obtain further features. While this method facilitates network focus on the diseased points of grape leaves rather than the background, drawbacks of the Resnet model include a large number of parameters and high volume of the model after training. Despite the increased accuracy of identification of diseases from grape leaf images, several problems, such as large network parameters, complex calculations, and poor real-time performance, remain to be resolved. SE blocks only consider reweighting the importance of each channel by simulating the channel relationship while disregarding location information, consistently resulting in significant classification errors for grape leaf diseases.

In view of the above issues, we proposed CASM-AMFMNet based on CoAtNet to improve the identification and classification of grape leaf diseases. Firstly, CoAtNet effectively combined the convolution and attention layers to achieve a better balance between recognition accuracy and efficiency and showed better generalization ability and capacity of the network model. As a backbone, CoAtNet initially extracted the local edge features of disease images, such as contour and color. The CASM module was effectively used to solve the problems existing in the traditional SE block, embed location information into the channel attention system so that the network could perform over a wide range, and avoid computational overheads to accurately locate, capture, and extract feature information used to distinguish between diseases and reduce the interference of complex background information. Finally, the AMFM module, which could process the input image position and semantic information on different scales that were then rescaled and combined with the module input, was introduced to extract multi-scale features of small targets. Our model effectively reduced the quantity of calculations and training time of the network.

The main contributions of this study are as follows:

1. A new algorithm GSSL is proposed to enhance grape leaf imaging. The method initially grayscales the image and subsequently processes high-pass filtering and the grayscale image using the Sobel operator to obtain the mask. Simultaneously, the grayscale image is smoothed and denoised, and the image obtained is processed using the Laplace operator to enhance grape leaf details. Finally, a preliminary texture-enhanced grape leaf image is generated using this image and the mask.
2. To reduce the background interference in grape leaf images and improve the extraction of small disease spots, we have proposed a new network, CASM-AMFMNet. The design is as follows: (a) A coordinate attention shuffle mechanism (CASM) suitable for retaining accurate disease location information along two different spatial directions, and capturing the domain of interest is utilized to extract feature information for disease discrimination. The module uses the input grape leaf disease image feature maps to perform group convolution (GC), and each sub-feature map captures specific semantic information on network training. Meanwhile, adding a channel shuffle at the end of the module can effectively improve the correlations between different channels in the group convolution and integrate feature information on each channel, improve the network fit, and merge the extracted image features with fewer numbers of parameters to obtain higher model accuracy. The module assigns weights to the feature maps according to different semantic information. The weight of the channel in which the grape leaf disease features are located is the largest, which effectively suppresses the interference of analogs and extracts disease features under complex background interference. (b) An asymmetric multi-scale fusion module (AMSM) was designed, which assigns multi-scale perceptual fields in the main network and effectively extracts details such as the shape and contour of small grape leaf disease spots. The ACB on each branch can enhance the robustness of the model to flipped or rotated images, improve training accuracy, and further reduce the number of parameters and computational efforts of the model. The module may be used to better focus on smaller spots that are easily overlooked when performing feature extraction and extract features while improving the precision of small target extraction and reducing the training time required by the network.
3. Our method achieved an accuracy of 95.95% in identification of five grape leaf samples, F1 score of 95.78%, and mAP of 90.27%. Furthermore, the model had a good discriminatory power for distinguishing between healthy and diseased leaves, facilitated classification of grape leaves in complex environments, and effectively extracted small disease spot targets. This technique could also be used with good results in public datasets. Rapid and accurate identification and classification of leaf diseases should effectively reduce loss of grape production in agriculture.

## RELATED WORK

To reduce the harmful effects of diseases in plants, many experts and scholars have recently focused on exploring the utility of artificial intelligence in identifying and classifying plant diseases rapidly and effectively. These studies have made significant contributions to the recognition of plant diseases, especially grape leaf diseases. For instance, Kundu et al. (2021) developed the framework of "automatic and intelligent data collector and classifier" by integrating IoT and deep learning to precisely predict blast and rust diseases in pearl millet. Padol and co-workers used the SVM classification technique for grape leaf diseases. The K-means clustering segmentation algorithm was initially used to identify the region of disease and extract texture and color features, followed by a classification technique for stratification of leaf disease categories (Padol and Yadav, 2016). This method showed an accuracy of 88.89%. The group of Narvekar used the SGDM matrix method to analyze grape leaf diseases and systematically discussed effective methods for disease detection *via* leaf feature inspection (Narvekar et al., 2014). Their method was able to achieve accurate disease detection with little computational effort. Peng et al. (2021) performed extraction with CNN plus support vector machine (SVM) to diagnose grape leaves based on fused deep features. Using this method, the SVM classifier could be trained to achieve the same classification accuracy as the CNN model. Leaf GAN utilized by Liu et al. (2020b) to analyze four different grape leaf disease images successfully overcame the overfitting problem and improved identification accuracy. Jaisakthi and colleagues further used different machine learning techniques such as SVM, AdaBoost, and Random Forest tree (Liu et al., 2020b), to identify grape leaf diseases. Their results showed that SVM was able to achieve 93% accuracy (Jaisakthi et al., 2019). Each of the above methods has its own merits, and network models using SVM classifier and CNN clearly have the ability to successfully classify grape leaf diseases. However, identification of grape leaf diseases needs to be optimized in many areas to achieve higher classification accuracy. In addition, grape leaf images with blank backgrounds have mainly been used as the datasets for experiments to date, which are conducive to classification of simple images and less suitable for complex backgrounds. Meanwhile, the current network is inadequate for recognition of small target diseases and leads to generation of errors. Here, we propose a model better adapted to extract features of grape leaf small target diseases under complex backgrounds based on CASM-AMFMNet. The specific scheme of grape leaf disease identification and classification is presented in **Figure 1**.

# MATERIALS AND METHODS

## Data Acquisition

Grape leaf data used in this study were classified into five categories: (1) healthy, (2) black rot, (3) black measles, (4) leaf blight, and (5) downy mildew. The dataset was mainly derived from two sources. One part was collected from the Tianlu vineyard (Changsha, China), which incorporated images of healthy, black rot, leaf blight, and downy mildew leaves from different periods taken on both sunny and cloudy days. We constantly changed the shooting angles and distances and collected grape leaf images of different colors, sizes, and backgrounds. To ensure accuracy of recognition, the grape leaves filled the image to the maximum extent. The other part of the dataset included leaves of different grape varieties with black measles along with the above diseases from a complex environment, comprising several orchards located using websites such as Kaggle (2021) and Google, among which 2,603 images were screened. Using the available information and by consulting relevant scholars, we reorganized and reclassified the collected images, screened those that were categorized, and deleted blurred images. Ultimately, 3,409 grape leaf images were collected from both dataset sources. The numbers and ratios of different categories of grape leaf images are shown in **Table 1**.

Five types of grape leaves were analyzed in this study. Healthy grape leaves were dark green and palm shaped with a surface free of disease spots and clear veins. Black rot is a fungal disease (Tomoiaga and Chedea, 2020) usually occurring at the leaf margin. After their appearance, disease spots gradually expand to circular spots that are gray-white in the center and brown on the outer edge, with a grayish-brown margin. At the later stages of the disease, small dots arranged in a ring appear on the disease spots. During early infection with black measles caused by fungal complexes, such as Phaeoacremonium (Nerva et al., 2019), light green spots are formed between leaf veins that continue to expand to the end of branches, and eventually become tiger striated. At the initial disease stage, leaf blight caused by fungi, such as Pestalotiopsis (Nuthan et al., 2021), presents as light-brown, irregular, and angular small spots, which then expand into circular or oval brown spots with a brown or tan center and a dark brown margin with a water-stained outer edge. Downy mildew [caused by *Plasmopara viticola* (Berk. & Burt.) Berl. & De Toni belonging to the order Peronosporales, a pathogen of grape-specific oomycetes (Fawke et al., 2015)] produces small, indistinct, yellowish watery spots with indistinct edges in the early stages of infection, which gradually expand into light green or yellow-brown spots on the front of leaves. Images of individual grape leaf diseases clearly show distinct spot characteristics. However, black rot and leaf blight have relatively similar features. Some leaf images show many tiny spots in both the early and later stages of infection, and therefore, extraction of their specific characteristics is important for disease recognition and management.

Convolutional neural networks require a large number of samples for model training, and acquisition of large quantities of disease images is a considerable challenge. Therefore, we expanded the dataset in this study using image transformation algorithms (Ghosal et al., 2018) to increase the sample number, prevent overfitting in the network, and improve the performance of the model (Pawara et al., 2017; Barbedo, 2018). We employed the algorithms of perspective transformation, geometric transformation (Sladojevic et al., 2016) [e.g., horizontal and vertical mirroring flip (Wang et al., 2017)], and intensity transformation (e.g., contrast increase and decrease and brightness enhancement and decrease) (Khan et al., 2018) to increase the number of grape disease images with a view to simulating the real collection environment and improving diversity and accuracy. With the aid of "vertical mirroring," "horizontal mirroring," "contrast reduction by 10%," "contrast increase by 10%," "Grayscale value increase by 45," "Grayscale value reduction by 45," "perspective transformation," and "image transposition" processes, grape leaf diseases were imaged. Taking the grape leaf downy mildew image as an example, the eight

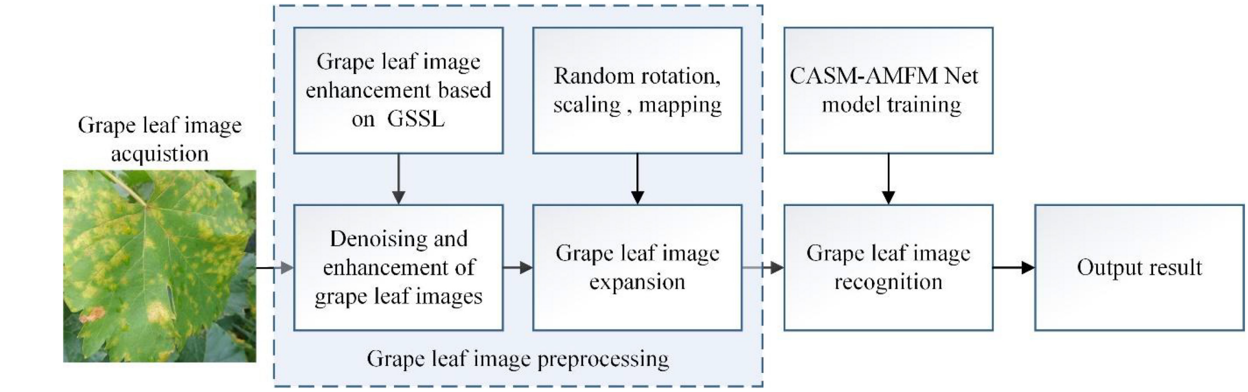
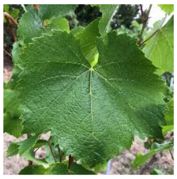
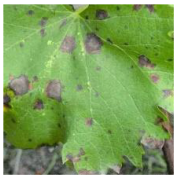
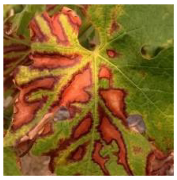
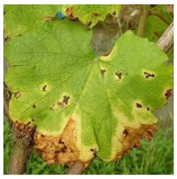
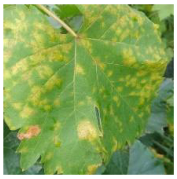

Grape leaf image acquistion
Grape leaf image enhancement based on GSSL
Random rotation, scaling , mapping
CASM-AMFM Net model training
Denoising and enhancement of grape leaf images
Grape leaf image expansion
Grape leaf image recognition
Output result
Grape leaf image preprocessing

**FIGURE 1 |** A working principle diagram of the system.

**TABLE 1 |** Number and proportion of grape leaf images.

| Category | Example | Number (Before) | Proportion/ % (Before) | Number (After) | Proportion/ % (After) |
|---|---|---|---|---|---|
| Healthy |  | 814 | 23.88 | 3,166 | 20.02 |
| Black rot |  | 725 | 21.27 | 3,148 | 19.89 |
| Black Measles |  | 669 | 19.62 | 3,175 | 20.06 |
| Leaf blight |  | 674 | 19.77 | 3,154 | 19.93 |
| Downy Mildew |  | 527 | 15.46 | 3,181 | 20.10 |

transformed images are shown in **Figure 2**. Original downy mildew leaves are usually light green. The disease spots could be enhanced by adjusting the contrast of disease spots and leaf colors. Through perspective transformation, the disease spot could be enlarged, which facilitated observation of the water stain shape. Different angle transformation methods were utilized to examine the shapes of the diseased leaves from different angles. At the same time, the brightness transformation simulated leaf images in different environments, leading to enhancement of disease characteristics.

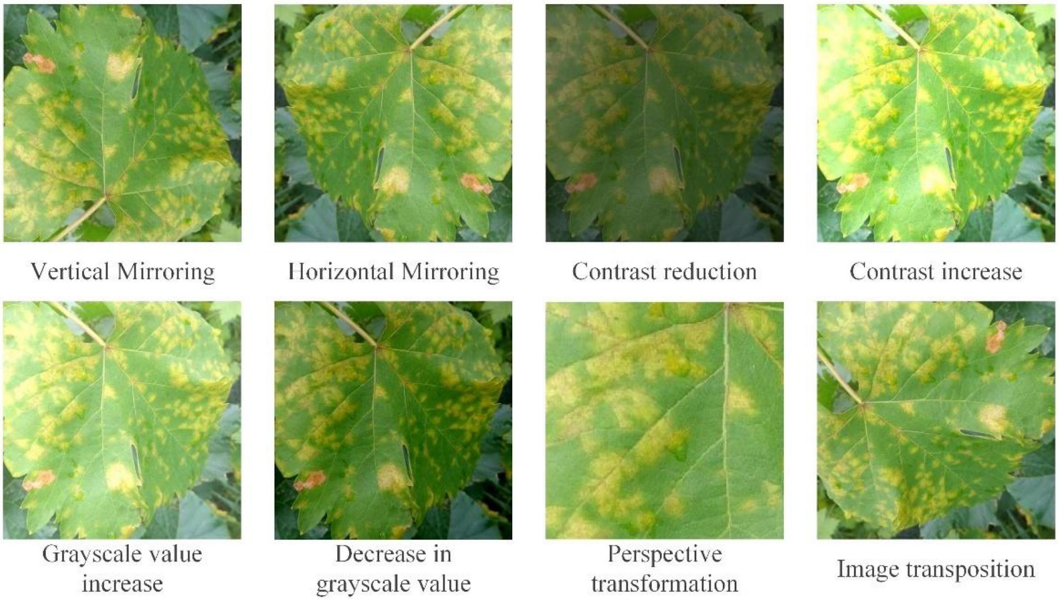


**FIGURE 2 |** Eight transformation images of downy mildew as an example.

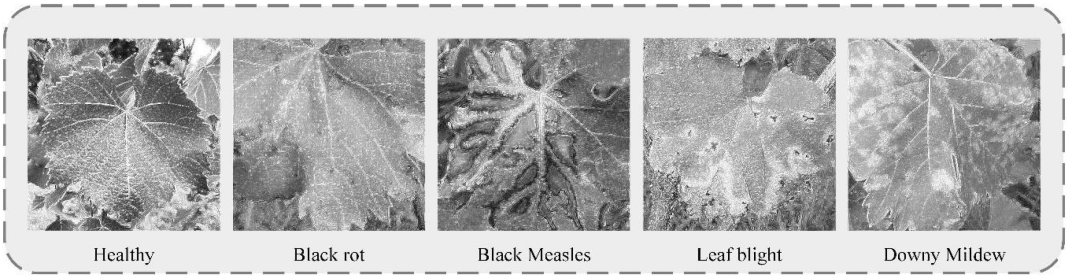


**FIGURE 3 |** A Gaussian filters Sobel smooth de-noising Laplace operator (GSSL) enhancement effect chart for five grape leaves.

## Evaluation Indicators

To determine the effectiveness of our method and quantitatively analyze the accuracy of grape leaf image recognition and classification, evaluation criteria used on the one hand were accuracy Equation (1), precision Equation (2), recall Equation (3), and mAP Equation (4) for assessment of the model performance. On the other hand, considering the limitations of storage and computational power during network operation, FPS (the number of grape leaf images recognized by the model per second, representing speed of detection), recognition time used per batch of images, param, MFLOPs, and FLOPs were also used as criteria for model evaluation.

$$Accuracy = \frac{TF + TP}{FP + TN + TP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$mAP = \int_0^1 P(R)dR \tag{4}$$

*TP* indicates the number of accurately identified grape leaf disease categories, *TN* the number of incorrectly identified non-grape leaf diseases, *FP* the number of correctly identified non-grape leaf diseases, and *FN* the number of grape leaf diseases that were not correctly identified. Precision indicates the proportion of all correctly predicted grape leaf images to the number of true correct samples and incorrectly predicted correct samples within the data. Recall signifies the proportion of grape leaf images of all predicted correct samples in relation to all true correct samples. For comprehensive evaluation of the model, the harmonic average F1 score of precision and recall was applied as the evaluation index, as shown in Equation (5).

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{5}$$

FPS representing the number of images detected by the model per second (speed of detection) can be obtained from Equation (6) below.

$$FPS = \frac{N}{T} \tag{6}$$

In Equation (6), $N$ represents the number of recognized samples and $T$ the time required to test all samples.

For the evaluation index of the image quality, we selected grayscale mean, peak signal-to-noise ratio (PSNR), and entropy to quantitatively analyze the quality of image enhancement and compare with the visual effects. The grape leaf image with a high mean gray value is bright overall, which is easier to identify than an image with a low mean gray value. Larger PSNR corresponds to lower distortion of the grape leaf image. Larger entropy values are correlated with richer texture information. Mean, PSNR, and entropy are calculated using Equations (7–9).

$$mean = \frac{1}{X \times Y} \sum_{i=1}^{X} \sum_{j=1}^{Y} R(i, j) \tag{7}$$

$$PSNR = 10 \times \lg\left(\frac{MAX^2}{MSE}\right) \tag{8}$$

$$entropy = \sum_{i=0}^{225} P(i) \times \log_2 p(i) \tag{9}$$

Here, X × Y represents the total number of pixels in the image, $R(i,j)$ is the pixel value of the image point $(i,j)$, $R(i,j)$, and $f(i,j)$ are grayscale values of the output and input images at point $(i,j)$, respectively, $MSE$ is the mean square error, and 255 is the maximum gray level. $P(i)$ denotes the proportion of pixels with gray value $i$ to total pixel number.

## Gaussian Filters Sobel Smooth De-Noising Laplace Operator

The images of grape leaves in the dataset have a number of issues, such as inconspicuous contrast, blurred edges, and noise. Therefore, the acquired grape leaf images need to be pre-processed *via* filtering, noise reduction, and enhancement.

We have proposed a GSSL algorithm to denoise and enhance grape leaf images in this study. Compared with the traditional preprocessing method, our procedure does not need to segment the background and leaves but deepens the edge contours of grape leaves, reduces image noise, and uses multi-step combination processing, such as ideal high-pass filter, Sobel operator, and smooth filter. Useful information from the image is extracted to the maximum extent possible and the noise reduced. Through image superposition, the authenticity of the original image is retained, and image distortion is effectively prevented while highlighting useful information. The image is obtained as $E(i,j)$, as shown in the Equation (10).

$$E(i, j) = \left\{\sqrt{s_i^2 + s_j^2} \times \left[\sum_{m=-1}^{1} \sum_{n=-1}^{1} k(m, n)p(i - m, j - n) + f(i, j)\right]\right\} + f(i, j) \tag{10}$$

Here, $k(m,n)$ is the Laplace operator mask of 3×3, $p(i,j)$ the gray value after smooth filtering, $s_i$ and $s_j$ the gradients of the image in the horizontal and vertical directions, respectively, and $f(i,j)$ the gray value of the input image at point $(i,j)$. The specific workflow of the GSSL algorithm is as follows:

Step 1: The data of grape leaf images are normalized. Color images are converted into grayscale, and the normalized and grayscale-processed grape leaf images used as the input for subsequent steps.

Step 2: The mask required is obtained and a simple detailed enhancement image acquired in two steps.

1. First, the ideal high-pass filter is used to process the input image. Through this step, the high-frequency part of the grape leaf image in the frequency domain space, i.e., edge details, can be extracted. Next, the image extracted with the ideal high-pass filter is added to the input image to obtain a simple edge enhancement image. The Sobel operator is subsequently used as the convolution kernel for the convolution operation on the image obtained in the previous step to acquire edge information for use as a mask.
2. The input grape leaf image is smoothed, denoised, and processed with the Laplace operator to highlight minor details. Incorporation of this result into the input image generates a preliminary detail-enhanced image.

Step 3: Image calculation. The mask image processed in Step 1 is multiplied by the initial enhanced image obtained in Step 2 for efficient extraction of edge and detailed information from the grape leaf image. The input image is then added to obtain an enhanced grape leaf image. A representative enhanced image obtained with the GSSL algorithm is shown in **Figure 3**.

As observed from the figure, grape leaf image processed using GSSL displays a certain shape of dark spots with an obvious edge contour. For example, “black measles” and “leaf bright” can clearly be utilized to detect the location of the spots. Although the disease contour is not obvious, colors of spots are easily distinguishable from the healthy leaf surface. The veined texture of all enhanced grape leaves is also more prominent, weakening the background-independent factors and reducing noise in the image, which increases the convenience of subsequent extraction of grape leaf characteristics by the neural network model.

## Coordinated Attention Shuffle Mechanism-Asymmetric Multi-Scale Fusion Module Net

In the images of grape leaf disease we collected, most of the grape leaves that have diseases show background interference. This entails that the whole network is vulnerable to impeded recognition, resulting in the incorrect localization of the identified disease areas. In addition, some of the diseased areas are almost integrated with the grape branches in the background, or their leaf shapes and contour after leaf curling are easily confused with the shapes and contours of the flowers in the background, resulting in recognition errors, which reduces the recognition accuracy of the grape leaf diseases.

Therefore, the reduction of the impact of a complex background on disease recognition and the realization of the feature extraction of small disease spots of grape leaf diseases are problems that need urgent solution. In response, this paper designs a CASM-AMFMNet for grape leaf disease identification and classification. First, the network proposes a backbone based on CoAtNet. Then, the CASM module is used to accurately capture location information for grape leaf diseases and to focus on their essential features to reduce complex background information. Finally, AMFM is used to give the main network multi-scale perceptual fields, extracting subtle features such as disease spot shapes and contours in all directions as much as possible to improve the accuracy of the network recognition of small targets, effectively reducing the amount of parameter computation and reducing the training time of the network.

The overall structure of the CASM-AMFMNet is shown in **Figure 4**, which is mainly divided into the following three parts:

1. In the first part, we used CoAtNet as the backbone. It uses convolution for downsampling up to stride = 16 to perform preliminary extraction of the features of grape leaf disease and bring about higher accuracy in the network, better generalization, and larger capacity.

The second part consists of a CASM module and an AMFM. First, the CASM divides the feature map into G groups (see the following text for the definition of G). Then, we used coordinate attention mechanism (CAM) for an average pool of the horizontal and vertical directions; this process assigns different weights to channel and spatial features, suppresses background information that is invalid with respect to the features of grape leaf disease, captures the accurate location information on the disease, and enhances the expressiveness of the network. Next, through three 1*1 convolutions, a feature map of the same size and enhanced representation as the input grape leaf feature image is obtained. Finally, the feature maps obtained from the first layer are added to the module after the attention mechanism for the channel shuffle to enhance the expression of the learned features and use the SELU activation function to enhance the nonlinear expression capability of the network. We added the CASM module at the backbone end of the model to fully consider the global and local texture features of grape leaf disease. AMFM consists of two 1*1 convolutions and *n* 3*3 convolutions at different scales, and the feature fusion of the disease features extracted by backbone can strengthen the recognition capability of small disease targets. The convolution adopts ACB convolution, which is done to reduce the amount of parameter computation and speed up network training.

2. In the third part, the global pooling downsampling layer is connected to the fully connected layer. Finally, the output is transformed into a probability distribution using Softmax to obtain the classification results of grape leaf disease images.

The following three subsections elaborate on the network.

## CoAtNet

ConvNet has good generalization capability and rapid convergence speed. Nevertheless, its perceptual range is limited by the size of the convolution kernel, while its large-scale perceptual ability is conducive to the model to obtain additional contextual information. A transformer tends to have a larger model capacity, but its generalization capability is poor relative to that of ConvNet due to the lack of correct induction deviation. Therefore, this paper uses CoAtNet (Dai et al., 2021) as the backbone, which effectively combines ConvNet with a transformer to achieve a better trade-off between accuracy and efficiency, and its backbone network uses residual connections. As a result, the network structure has sufficient depth to retain additional feature information and facilitate the fusion of feature information at the front and back layers of the network. In addition, the network can mitigate network degradation, including gradient disappearance and explosion during training, which makes the model easier to converge and leads to stronger feature extraction capability. When the data set is large, the network model is enabled to have stronger learning ability and generalization ability so that the network model has better performance on classification tasks.

## Coordinate Attention Shuffle Mechanism

Some of the images in the grape leaf data set that we collected were taken in a complex natural environment. The images have problems such as grape leaf self-obscuring, grape fruit, hand obscuring the disease area, leaf curling, and so forth. In addition, because some of the disease spots first occur at the edge of the leaf, the traditional method shows a large degree of uncertainty in terms of acquiring information about the grape leaf disease area. However, the gaps between different diseases on grape leaves are usually in tiny local details. If it is affected by both the background and the shape of disease spots at the same time, this will lead to increased recognition. In the current study, we found that CAM (Hou et al., 2021) can capture cross-channel information and orientation- and position-aware information, which can help the model locate and identify potential targets more precisely. Second, CAM is an attention method with flexible and lightweight properties that can be easily inserted into classic modules to enhance features by strengthening information representation. Finally, as a pre-trained model, CAM can bring significant gains to downstream tasks based on lightweight networks.

Therefore, in this paper, we propose the CASM module, which is based on CAM, and added it to the CASM-AMFMNet so that the network model can pay closer attention to the grape disease area, distinguish the background interference from the disease, and accurately obtain the detailed feature information for the grape leaf disease area to extract the feature information to distinguish between diseases and improve recognition capability of the local detailed features of the disease. The CASM module is shown in **Figure 5**.

Because the CASM module is proposed according to the cam module, referring to the two steps of the cam module, this paper proposes that group coordinated information on the embedding module (GCM) and the coordinated attention generation shuffle module (CSM) are the main structures of the CASM module.

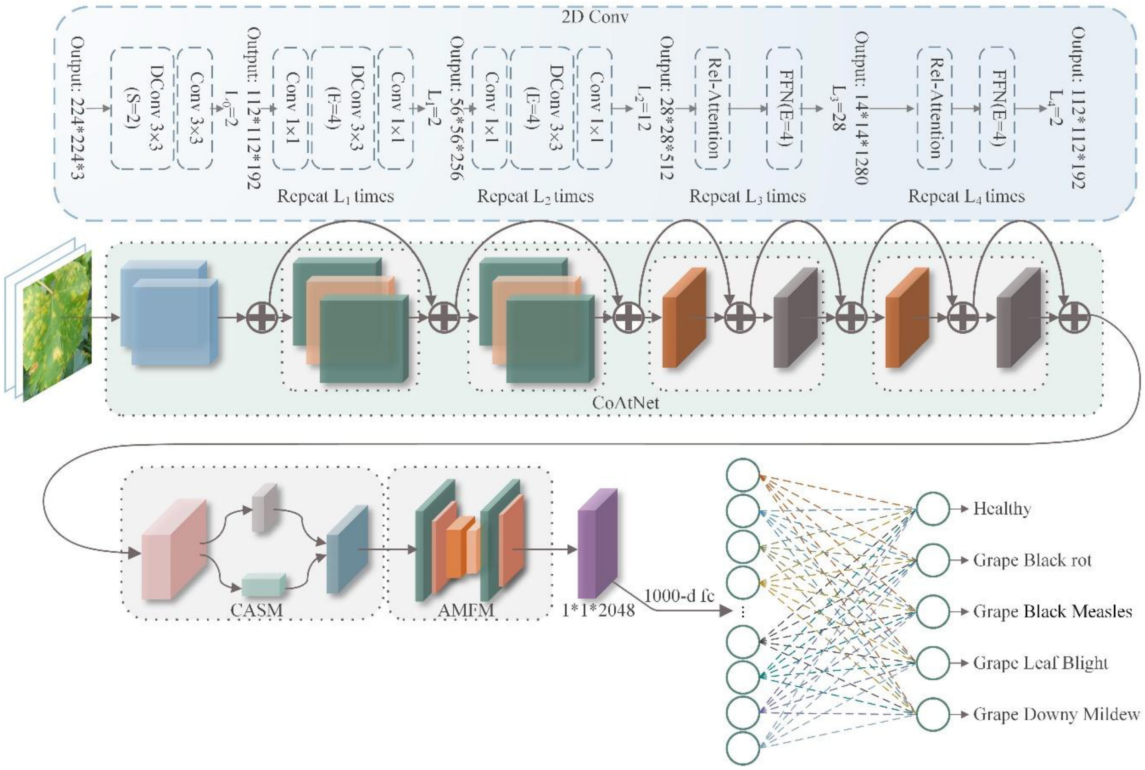


**FIGURE 4 |** Coordinated attention shuffle mechanism-asymmetric multi-scale fusion module net (CASM-AMFMNet) structure.

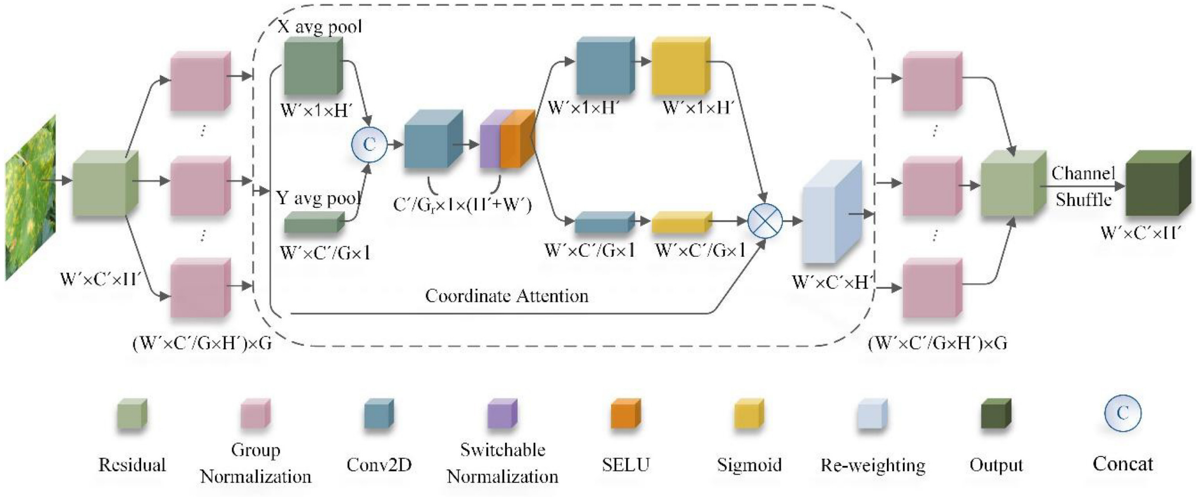


**FIGURE 5 |** Coordinate attention shuffle mechanism (CASM) module structure.

A. Group coordinate information embedding module (GCM)

This operation of the GCM corresponds to group convolution and the two parts X Avg Pool and Y Avg Pool in **Figure 5** above, which is a global sensory field that encodes precise location information. The CASM module proposed in this paper has the following four improvements.

First, we GC the input image feature map of grape leaf disease. In network training, each sub feature map captures specific semantic information. After we performed GC, the parameter quantity became 1/G of the original standard convolution. With the increase in the number of groups, the parameter quantity and calculation quantity are significantly reduced. The G obtained by the experiment is set to 4. In addition, GC cannot easily produce

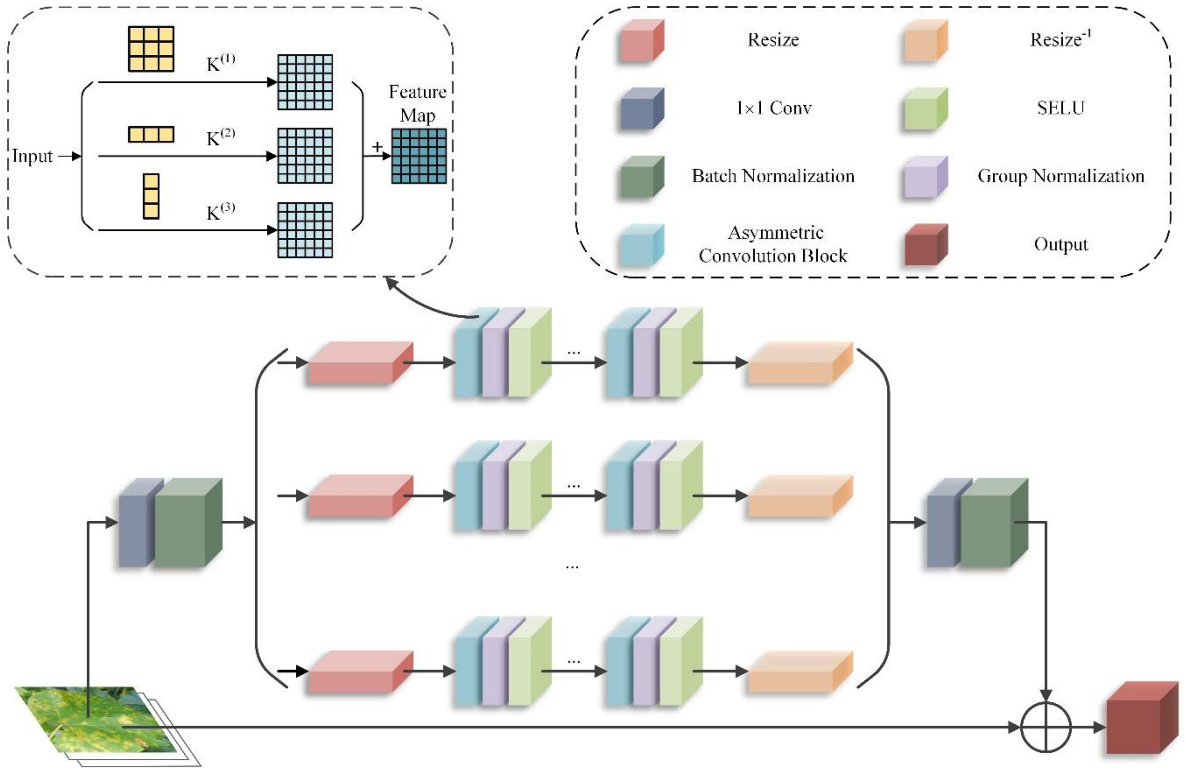


**FIGURE 6 |** Asymmetric multi-scale fusion module (AMFM) structure.

overfitting, and it has the effect of regularization (Krizhevsky et al., 2017). Then, the attention module is induced to capture the remote dependencies with precise location information, and then the pooling kernel with dimensions ($H$, 1) and (1, $W$) is used to encode the sub-feature maps for each channel along the horizontal and vertical coordinates, respectively, so that the output of the $c$th channel can be written as the Equation (11).

$$z_c = \frac{1}{H \times W} \sum_{j=1}^{H} \sum_{j=1}^{W} x_c(i, j) \tag{11}$$

In the Equation (11), $z_c$ denotes the output of the $c$th channel and $x_c(i,j)$ denotes the values of the position characteristic diagram of height coordinate $i$ and the width coordinate $j$ of the $c$th channel. The output of the $c$th channel with height $h$ can be expressed as the Equation (12).

$$Z_c^h = \frac{1}{W} \sum_{0 \le j \le W} x_c(h, j) \tag{12}$$

In the Equation (12), $Z_c^h(h)$ denotes the output with height of the $c$th channel as $h$, and $x_c\left(h, j\right)$ is the value of the feature map with width coordinate $j$ for the $c$th channel with height $h$. The output of the $c$th channel with width $w$ is as shown in the Equation (13).

$$Z_c^w(w) = \frac{1}{H} \sum_{0 \le j \le H} x_c(i, w) \tag{13}$$

In the Equation (13), $Z_c^w(w)$denotes the output with the height of the $c$th channel as $w$; $x_c\,(i, w)$ is the value of the feature map with height coordinate $i$ for the $c$th channel with width $w$, and $H$ and $W$ are the height and width of the feature map, respectively.

The above two transformations aggregate features along two spatial directions and generate direction correlation feature graphs. This is very different from the SE block, which generates a single eigenvector in the channel attention method. These two transformations also allow the attention module to capture long-term dependencies along one spatial direction and preserve precise location information along the other, which helps the network to locate small spots more accurately.

B. Coordinate attention generation shuffle module (CSM)

In Step A, the global sensory field can be easily obtained, and precise positional information encoded. To better integrate the features of grape leaf diseases so that their features can be fully utilized to capture positional information and facilitate more precise localization of ROI regions, we concatenated the aggregation feature maps generated by Equations (12, 13), and we used 1*1 convolution to compress the channel for transformation to obtain Equation (14).

$$f = \delta\left(F_1\left(\left[z^h, z^w\right]\right)\right) \tag{14}$$

In Equation (14), $\left[z^h, z^w\right]$ is a stitching operation along the spatial dimension, $\delta$ uses SELU, and $f \in R^{c/r\times(H+W)}$ is an intermediate feature map that encodes the spatial information in the horizontal and vertical directions. Then, $f$ is decomposed into two separate tensors $f^h \in R^{c/r\times(H+W)}$ and $f^w \in R^{c/r\times(H+W)}$ along the spatial dimension, and two additional 1*1 convolution transforms $f^h$ and $f^w$ are used to transform $F_h$ and $F_w$ into tensors with the same number of channels to the input X, respectively, to obtain Equations (15, 16).

$$g^h = \sigma\left(F_h\left(f^h\right)\right) \tag{15}$$

$$g^w = \sigma\left(F_w\left(f^w\right)\right) \tag{16}$$

In Equations (15, 16), $\sigma$ is the sigmoid activation function to reduce the complexity and computational overhead of the model; an appropriate scaling ratio $r$ is usually used here to reduce the number of channels of $f$. Next, the outputs $g_h$ and $g_w$ are expanded and used as attention weights to generate new feature maps by combining all of the sub-feature maps, as shown in Equation (17).

$$y_c\left(i,j\right) = x_c\left(i,j\right) \times g_c^h\left(i\right) \times g_c^w\left(j\right) \tag{17}$$

Finally, by shuffling the information on the sub-feature map, we strengthened the information exchange between different channels and acted on the input to obtain the output $X = [x_1, x_2, \ldots x_c]$ with the same size of this attention module as the input $Y' = [x_1, x_2, \ldots x_c]$ and with enhanced learning features, as shown in Equation (17).

$$Y' = channel\_shuffle(Y) \tag{18}$$

In GSM, the method used in this paper makes the following innovations.

1 To adapt to complex and variable backgrounds, we used switchable normalization (SN) instead of the traditional batch normalization (BN) layer to make the model more robust to adapt to various scenarios by dynamically adjusting the weights through training. SN calculates the BN, LN, and IN, produces the statistical weighting (weights are calculated by Softmax), and finally calculates the normalized pixel value $\overset{\frown}{h}_{nchw}$ as Equation (18).

$$\overset{\frown}{h}_{nchw} = \gamma\frac{\overset{\frown}{h}_{nchw}\sum_{k\varepsilon\Omega}\omega_k\mu_k}{\sqrt{\sum_{k\varepsilon\Omega}\omega_k^{'}\sigma_k^2+\varepsilon}} \tag{19}$$

In Equation (19), we input a four-dimensional feature vector of a grape leaf image with $n$, $c$, $h$, and $w$, representing the number of samples, channels, height, and width, respectively. $h_{nchw}$ is each pixel on the feature map, $\overset{\frown}{h}_{nchw}$ is the pixel value output after the SN operation on$h_{nchw}$, $\gamma$ is the scaling coefficient; $\beta$ is the offset coefficient, $\mu$is the mean value, $\sigma^2$is the variance, and $\omega_k$ and $\omega_k^{'}$ are the weighting coefficients for weighting the mean and variance, respectively. The weight coefficient $\omega_k$ uses the Softmax function to calculate the control parameters $\lambda_k$ of the three dimensions, as shown in Equation (20).

$$\omega_k = \frac{e^{\lambda_k}}{\sum_{z\varepsilon\Omega} e^{\lambda_k}} \tag{20}$$

In Equation (20), the initial values of the control parameters $\lambda_k$ for each of the 3 dimensions are 1, which are optimized during back propagation with $\sum_{k\varepsilon\Omega}\omega_k = 1$; the value of each weighting factor $\omega_k$ is between 0 and 1.

2 We used the SELU activation function instead of the commonly used ReLU activation function or the Sigmoid activation function to improve the learning convergence effect of the model. The SELU activation function is calculated as follows:

$$SeLU(x) = \lambda_{selu}\begin{cases} x & x \geq 0 \\ \alpha_{selu}\left(\exp\left(x\right) - 1\right) & \text{otherwise}\end{cases} \tag{21}$$

In Equation (21), $\alpha$ and $\lambda$ are hyperparameters, and it is proven that the training effect reaches the best at$\alpha_{selu} \approx 1.6733$, $\lambda_{selu} \approx 1.0507$.

3 Adding a channel shuffle at the end of the module can effectively integrate the feature information on each channel, strengthen the information exchange between channels, and better enable the network fit of the extracted image features with fewer parameters to obtain higher model accuracy, improve the efficiency of the model operation, and enhance the classification effects.

## Asymmetric Multi-Scale Fusion Module

Compared to the entire image, the diseased area on a grape leaf image is tiny, so the size of the disease spot used for the extraction itself is necessarily small. After CoAtNet, the semantic information of the small targets in the grape leaf features map almost disappears at this time, which increases the difficulty of the network to recognize small spots. The black rot spots and leaf bright spots are small and dense, the black measles spots are similar to stripes, and the frosty mildew spots are irregular in shape. To address the problem of small target recognition in grape leaves, we extracted and fused the shape and contour features of grape leaf spots at different scales, effectively improving network accuracy and enhancing the feature expression capability of the convolution kernel to achieve accurate recognition of small targets. A single-scale convolution kernel is not efficient for sensing multi-scale lesion points. Therefore, this paper proposes AMFM, which uses MSFM (Wang and Wang, 2020) as the framework for extracting the features of multi-scale lesions and partially improves it. AMFM can extract the small lesion features of grape leaves to a greater extent without increasing the amount of calculation and improve the robustness of the model to image reversal.

Asymmetric multi-scale fusion module divides the feature map obtained after 1*1 convolution into n scales equally. One of the 3*3 convolutions is replaced using an asymmetric convolution block (ACB) (Ding et al., 2019), which can still extract features correctly for flipped images to improve the network's training accuracy and reduce the parameters of the model training and the required computational effort. On the other hand, the use of the SELU activation function instead of the ReLU activation function can better fit the training, extract the features of the grape leaf spots, and improve the learning convergence of the network. The model for AMFM is shown in **Figure 6**.

First, the input grape leaf images are convoluted with 3*3 convolution kernels, 1*3 convolution kernels, and 3*1 convolution kernels, which produce three different shapes to extract different branch features, as shown in Equation (22). Then, the different branches are fused using convolution's additivity to obtain the fused feature output. To make the module lightweight while maintaining the dimensionality of the fused output features consistent with that of the input features, the residual bottleneck structure is utilized. This structure refines the module input according to the channel and then feeds into the branches. Finally, the branch input is resized using bilinear interpolation, and its elements are returned to their original size using the same method, as shown in Equation (23).

$$M(x) = x + U\{C[F_1(S(x)), F_2(S(x)), \ldots F_n(S(x))]\} \quad (22)$$

$$F_n(a) = R_n^{-1}\left(CGN_{n,i}\left(CGN_{n,i-1}\left(\ldots\left(CGN_{n,i}\left(R_n(a)\right)\right)\right)\right)\right) \quad (23)$$

In Equations (22, 23), x is the input grape blade, $M(x)$ is the output, $S()$ is the extrusion module that makes the input x thinner, $F_n()$ is the branching operation, $C()$ is the combination function, and $U()$ is the unsqueezed module that restores the branching output depth to be the same as $x$. $CGN_{n,i}$ is the result of the extrusion module, $R_n()$ is the resize function on the nth branch, $a = S(x)$ is the normalized nonlinear operation on the nth branch of the $i$th ACB group, and ${R_n}^{-1}$ is the resize function to restore the feature dimensions (height and width). The computational volume equation after applying the ACB is as shown in Equation (24).

$$I * K_1 + I * K_2 = I * (K_1 \oplus K_2) \quad (24)$$

In Equation (24), $I$ is the input feature map matrix; $K_1$ and $K_2$ are two convolution kernels; $\oplus$denotes added corresponding positions of the convolution kernels. In the feature fusion process of asymmetric convolution processing, the feature information is superimposed based on standard 3*3 convolution processing with feature information extracted by two dimensions of asymmetric convolution. Compared to the 3*3 convolution with 3*3 multiplications, the number of asymmetric convolution operations is 2*3 multiplications, and the amount of network operations is reduced by 1/3.

# RESULTS AND ANALYSIS

This section verifies the effectiveness of the CASM-AMFMNet in the identification and classification of grape leaves through experiments and designs experiments to use the test set in other models together with the model in this paper to compare the effectiveness of different models. This section describes the experimental environment, the experimental setup, the evaluation metrics, the effectiveness analysis of each module of CASM-AMFMNet, the ablation experiments, and the comparison experiments between different models.

**TABLE 2 |** Hardware and software environment.

| Hardware environment | CPU | Intel Core i7-6800 K 3.40 GHz 15 MB |
|---|---|---|
| | RAM | 64 GB |
| | Video memory | 32 GB |
| | GPU | NVIDIA GTX 2080ti |
| Software environment | Operating system | Windows 10 |
| | CUDA Toolkit | V11.1 |
| | CUDNN | V8.0.4 |
| | Python | 3.8.8 |
| | Torch | 1.8.1 |
| | Torch vision | 0.9.1 |
| | Matlab | 2020a |

**TABLE 3 |** Parameter setting.

| Parameter category | Parameter name | Parameter setting |
|---|---|---|
| AdamW | Initial learning rate | 0.001 |
| | Weight decay | $1 \times 10^{-4}$ |
| | Momentum | 0.9 |
| | Learning rate decay | 0.1 |
| Input data parameters | Size of input images | (224,224) |
| | Minibatch | 32 |
| | Iteration Epochs | 30 |
| | Iteration Number | 37,950 |

## Experimental Environment and Data Preparation

To verify the performance of the CASM-AMFMNet proposed in this paper, all experiments were carried out in the same hardware and software environment, with the specific environmental parameters shown in **Table 2**.

## Experimental Settings

The self-made data set used in the experiments in this paper contained five categories of grape leaves: healthy, black rot, black measles, leaf blight, and downy mildew. The size of the unified image input is adjusted to 224*224 to improve the efficiency of the image processing technology, minimize the calculation cost, and reduce the time spent with the training model and classification. After pre-processing, we obtained a total of 15,824 images of grape leaves. The number of images for the five diseases was evenly distributed, all in the range of 19–21%. The data sets in this paper were divided in the following ratio: the training set: the validation set: the test set = 3:1:1, with 9,480 images of the five grape leaves in the training set and 3,160 images in the test set.

In the deep learning training, the hyperparameter selection is difficult and time-consuming because the optimal combination of hyperparameters depends not only on the model itself but also on the software and hardware environment. In this paper, the hyperparameters of the CASM-AMFMNet were determined through multiple fine adjustments, as shown in **Table 3**. When

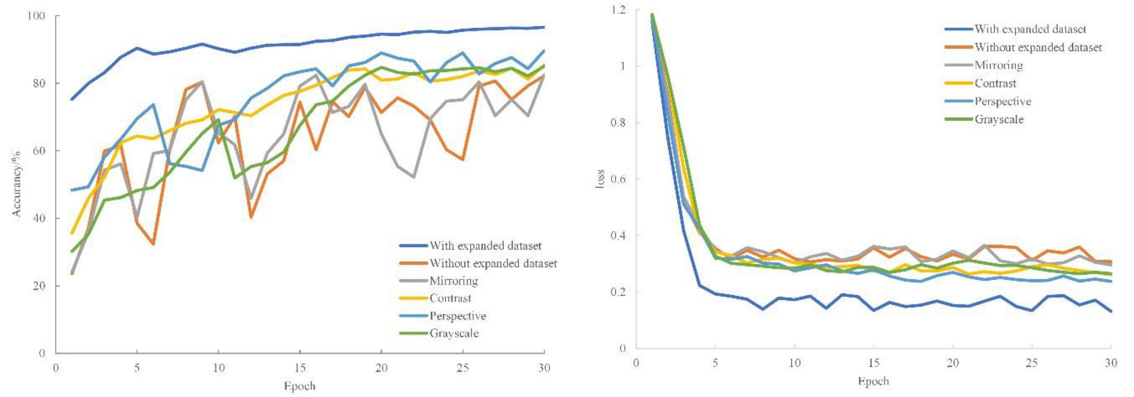


**FIGURE 7 |** Comparison experiments of different data enhancement effects.

training with the model, batch training was adopted to randomly divide the training and validation sets into multiple batches, with a training batch (Minibatch) of 32 and a round batch (epoch) of 30 and 1,265 iterations per round, for a total of 37,950 iterations. We verified once every 1,000 iterations; the initial learning rate was set to 0.001, and the weight decay value was $1 \times 10^{-4}$.

To investigate the effects of different optimizers on model performance, three commonly used optimizers were selected for model training, and model accuracy was obtained under different optimizers. For validation accuracy, the AdamW (Kingma and Ba, 2014) optimizer value is 1.03 and 1.42% higher than those for the SGDM and RMSprop optimizers, respectively; for testing accuracy, the AdamW optimizer value is 1.71% and 2.10% higher than those for the SGDM and RMSprop optimizers, respectively. Therefore, the AdamW optimizer with the driving volume is more suitable for this study model. Under the same experimental conditions, the accuracy of the models obtained by the three optimizers differed significantly. Regarding training duration, the three methods are relatively close to one another, all occupying around 3 h, although the AdamW optimizer takes the shortest time.

Training parameters of the model are set as shown in **Table 3**.

## Individual Modules Effectiveness Analysis

### Impact of Data Enhancement on Recognition Performance

In this section, we used digital image processing to expand the collected grape leaf image data sets and then trained the original data set, the flip expanded data set, the contrast expanded data set, the gray expanded data set, the perspective expanded data set, and the common expanded data set using the model CASM-AMFMNet, proposed in this paper. The experimental results for accuracy and loss are compared to evaluate the impact of data enhancement on the classification accuracy of grape leaf diseases, as shown in **Figure 7**. Compared to the original image data set, the training accuracy of different expansion methods is improved by 0.30, 3.17, 2.86, 7.49, and 14.42 percentage points, respectively. The training accuracy in the case of the flipping expansion is not different from that in the original data set as the flipping operation shows little change in image quality due to multi-angle shooting. However, the training accuracy of other expanded data sets is significantly higher than that of the original data set. The reason for this is that the original training sample set is too small, and the data expansion provides the necessary amount of data for model training. In particular, the recognition accuracy of the jointly expanded data set is much better than for that of the non-expanded data set. The loss function curve shows that the training loss value of the expanded data set is lower, and the model converges rapidly; it can well fit the characteristics of grape leaf disease. Data expansion increases the diversity of data, the parameters of the classification model are fully trained, and the network model has better feature extraction ability when trained on large data sets. More importantly, the enhancement of the data set can better simulate the real environment of grape leaves and improve the model robustness.

### Effectiveness of Gaussian Filters Sobel Smooth De-Noising Laplace Operator

To more objectively evaluate the feasibility of the method studied in this paper, the GSSL algorithm compares the grape leaf images processed by the GSSL algorithm with five filter enhancement and comparison algorithms, and the grape leaf image test set is enhanced for comparison experiments and analysis. The parameters of the gray level mean, peak signal-to-noise ratio (PSNR), and entropy of the six algorithms are shown in **Table 4**.

**TABLE 4 |** Enhanced image quality parameters.

| Method | Mean | PSNR | Entropy |
|---|---|---|---|
| Original image | 132.17 | 27.49 | 7.55 |
| EGIF (Wu et al., 2021) | 135.91 | 29.63 | 7.64 |
| WGIF (Mu et al., 2021) | 110.23 | 31.08 | 7.25 |
| HSFGTF (Joseph et al., 2021) | 128.25 | 28.70 | 7.00 |
| GFCBH (Pashaei, 2021) | 95.16 | 35.12 | 6.99 |
| WLS (Singh et al., 2022) | 119.32 | 35.04 | 7.81 |
| GSSL | 148.61 | 37.87 | 7.94 |

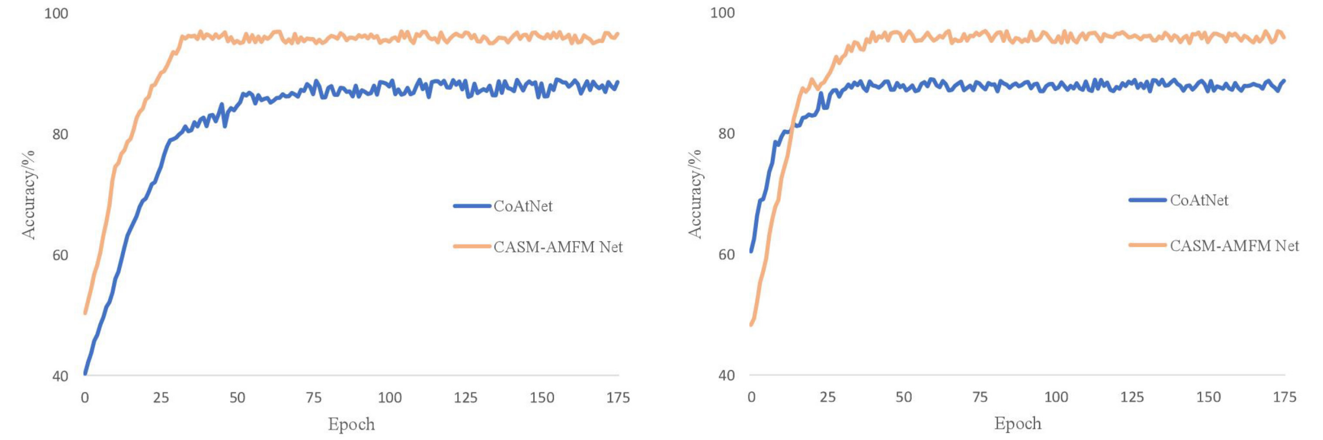


**FIGURE 8 |** Accuracy curves of the CoAtNet and the coordinated attention shuffle mechanism-asymmetric multi-scale fusion module net (CASM-AMFMNet).

From the data obtained in **Table 4**, it can be observed that the grape leaf image derived from this experiment is significantly improved relative to the original image. The PSNR of the enhanced image obtained by this method is 8.24, 6.79, 9.17, 2.75, and 2.83 dB higher than those for EGIF, WGIF, HSFGTF, GFCBH, and WLS, respectively, and 10.38 dB higher than that of the original image, indicating that the image enhanced with the algorithm used in this paper has less distortion and higher quality; the obtained entropy values of the enhanced image are 0.3 bit, 0.69 bit, 0.94 bit, 0.95 bit, and 0.13 bit larger than those for other methods and 0.39 bit larger than that of the original image, resulting in improved image quality and a greater amount of information. At the same time, the means of the enhanced image in this paper are 12.7, 38.38, 20.36, 53.45, and 29.29 higher than those of other methods and slightly higher than that of the original image and 16.44 higher than that of the original image, which makes the enhanced image brighter and more appreciable. The extraction of image edge details at the same time ensures the authenticity of the image information, effectively overcomes the impact of noise in the image, and makes the leaf details clearer, the image brighter, and the grape leaf spot texture obvious and readable.

## Self-Contrasting Experiments

We carried out self-comparison experiments on the grape leaf dataset for the underlying network CoAtNet model and our CASM-AMFMNet network model. First, the training set is used for training, and then the obtained model is tested against the test set. By contrast, the network model in this paper shows some improvement in recognition speed and accuracy compared to the CoAtNet network model.

It can be seen from **Figure 8** that CoAtNet iterations tend to converge 50 times, and the final training accuracy is 88.56%, while CASM-AMFMNet iterations tend to converge 30 times, and the final training accuracy is 96.58%, which is higher than CoAtNet. Because Sn and GN are added to the CASM-AMFMNet algorithm, the convergence speed of the model may accelerate. The accuracy rate for the CoAtNet test set is 88.74%, and it is 95.95% for the CASM-AMFMNet test set. Because the proposed algorithm incorporates contextual and location information among the grape leaf disease regions, the accuracy rate on the test set is 7.21% higher than that of CoAtNet.

It can be seen from **Table 5** that the number of parameters of our improved CASM-AMFMNet network is much smaller than the number from before the improvement (−2 M). In addition, ACB divides the standard convolution into 1*3 and 3*1, which further reduces some parameters of the original convolution layer and greatly improves the overfitting-prone characteristics of the complex network. When the number of parameters is reduced, the recognition accuracy is improved (+10.78%). In terms of program running time, with 32 samples per training batch, the original model takes 586 s, while the improved model takes only 270 s (−316 s). The difference in the total program time is even more obvious, with a 109.61-min difference in the time spent to train 30 epochs, reflecting the improved performance of the updated model in terms of the training cost. Our improved model significantly lowered the number of parameters. Its effectiveness is reflected not only in preventing overfitting and thus improving test accuracy but also in the time cost required for the training, which is highly practical.

## Backbone Comparison Experiment

To determine the choice of the model backbone in this paper, under the framework of CASM-AMFMNet, models with backbone from CoAtNet −1 to 7 are experimentally compared. The experimental results are shown in **Table 6**.

When the size of the grape leaf image data set is the same, the width of the network increases with the increase in the CoAtNet model. The network params and flops between CoAtNet −1 and 5 are very small, but the recognition accuracy differs by more than 1 percentage point. From CoAtNet −5, the recognition accuracy improves slightly. Where the depth of the network layer is deepened, CoAtNet −6 and CoAtNet −7

**TABLE 5 |** Performance comparison of CoAtNet and coordinated attention shuffle mechanism-asymmetric multi-scale fusion module net (CASM-AMFMNet).

| Method | CoAtNet | CASM-AMFMNet |
|---|---|---|
| Accuracy | 88.74% | 95.95% |
| mAP | 79.49% | 90.27% |
| FPS | 44 | 85 |
| Param | 168 M | 166 M |
| FLOPs | 189.5 B | 187.8 B |
| MFLOPs | 632.79 MB | 4.67 MB |
| Running time per batch | 586 s | 270 s |
| Time required per epoch | 169.86 min | 60.25 min |

increase the accuracy of the network model weight file by 0.03 percentage points after 109 M and 0.06 percentage points after 164 M. There is little difference in the recognition accuracy among CoAtNet −6, CoAtNet −7, and CoAtNet −5. Therefore, CoAtNet −5 with a moderate size of params and flops and a high recognition accuracy for the grape leaf image data set is used in all subsequent experiments.

## Effectiveness of Coordinate Attention Shuffle Mechanism

To verify the effectiveness of the CASM module, we first experimented with the settings of the grouping parameter G (see a below), and then verified the effects of the activation function, shuffling strategy, and the attention mechanism in the CASM module on the model through three experiments (see b–d below).

(a) Effects of different grouping numbers on the CASM module. The grouping number G is set to 2, 4, and 8. The analysis data are grape leaves, and the results of the analysis of different parameters are shown in **Table 7**.

As shown in **Table 7**, the test accuracy of the model is close to 90% for all three cases, with a different number of groups. With the increase in the number of groups, the params (−0.22 M, −0.11 M) and flops (−0.6 B, −0.3 B) of the network model are significantly reduced. Although the increase in the number of groupings reduces the computational and parametric quantities of the model, its intensive operations lower the computing and storage access efficiency and extend the actual running time. Therefore, in practical applications, combining the above reasons and experimental data, we set the number of groups of CASM to G = 4.

(b) The influence of SN and SELU activation functions on the model is used in the CASM module. To verify the feasibility and effectiveness of SN and SELU, this paper validates them in CoAtNet in terms of training time and training accuracy with different batching methods and activation functions, and the results are shown in **Table 8**.

In **Table 8**, the SN + SELU combination is shown to be better than BN/SN + ReLU, BN/SN + Sigmoid, and BN + SELU in mAP, with increases of +1.36%, +0.76%, +1.91%, + 1.31%, and +.69%, respectively, and compared to the most common BN + ReLU combination; its param is also reduced by about 0.78 M, and the training time is reduced by 30 min and 20 s, which makes the optimization learning and solving model convergence easier.

**TABLE 6 |** Experimental results for different backbone networks.

| Models | Eval size | Params | FLOPs | Accuracy |
|---|---|---|---|---|
| CoAtNet-1 | $224^2$ | 55 M | 49.8 B | 89.56% |
| CoAtNet-2 | $224^2$ | 75 M | 96.7 B | 92.45% |
| CoAtNet-3 | $224^2$ | 96 M | 126.1 B | 93.98% |
| CoAtNet-4 | $224^2$ | 121 M | 149.8 B | 94.77% |
| CoAtNet-5 | $224^2$ | 166 M | 187.8 B | 95.95% |
| CoAtNet-6 | $224^2$ | 275 M | 289.8 B | 95.98% |
| CoAtNet-7 | $224^2$ | 330 M | 360.9 B | 96.01% |

**TABLE 7 |** Comparison result of different groups.

| Group number | Test accuracy | mAP | Testing time | Params | Flops |
|---|---|---|---|---|---|
| G = 2 | 95.95 | 90.25% | 11.33 | 166.63 M | 189.5 B |
| G = 4 | 95.95 | 90.27% | 10.87 | 166.41 M | 188.9 B |
| G = 8 | 95.95 | 90.23% | 11.83 | 166.30 M | 188.6 B |

(c) The CASM module introduces the effects of channel shuffling strategies on the model. The grouping of grape leaf features generates a large amount of group convolutional stacking, which leads to feature information loss and the obstruction of the interactive flow of feature information between channels, as well as seriously affecting the feature characterization ability. In this paper, we introduce a channel-mixing strategy into the proposed module and compare the same module without adding the mixing operation to verify the impact of adding channel mixing on grape leaf disease identification. The model uses CoAtNet as the backbone network for comparative analysis of channel-mixing additions in the framework of CASM-AMFMNet.

The experimental results in **Table 9** indicate that the inclusion of the shuffling operation in the model does not generate additional parameters or computational effort, and the presence of the shuffle channel was effective in improving the average identification accuracy of grape leaf diseases. The use of channel shuffling after all convolutional layers using grouped convolution improves the accuracy of the model by 0.34 percentage points. Channel shuffling enhances the flow of feature information between channels, and plays a positive role in the interaction of feature information obtained from

**TABLE 8 |** Exploring the combination of normalized processing and activation functions.

| Method | mAP | Param | Training time |
|---|---|---|---|
| BN+ReLU | 80.31% | 167.33 M | 4 h 48 min 29 s |
| BN+Sigmoid | 79.85% | 167.95 M | 4 h 58 min 57 s |
| BN+SELU | 81.07% | 166.53 M | 4 h 16 min 42 s |
| SN+ReLU | 80.91% | 167.35 M | 4 h 50 min 03 s |
| SN+Sigmoid | 80.45% | 167.97 M | 4 h 59 min 44 s |
| SN+SELU | 81.67% | 166.55 M | 4 h 18 min 09 s |

**TABLE 9 |** Effect of shuffle on the model.

| | CASM-AMFMNet (no Shuffle) | CASM-AMFMNet (with Shuffle) |
|---|---|---|
| mAP | 89.93% | 90.27% |
| FLOPs | 189.5 B | 189.5 B |
| param | 166 M | 166 M |

group convolution, which makes the model more efficient in its use of feature information in different channels that are at the same spatial location after group convolution, improving the experimental accuracy.

(d) The CASM module uses the attention module for the impact on the model. While accomplishing network light weighting, channel attention is particularly important for ensuring network accuracy. This set of experiments is conducted to verify the effect of the CAM and CASM modules on the model, as well as the effect of different positions and quantities of CASM on the grape leaf disease data set. Under the same training environment, the performance of different attention modules of the CASM module and CAM on the model recognition ability is shown in **Figure 9**. The CASM module was added at different positions of CASM-AMFMNet to study the correlation between the recognition ability of the model and increasing the number of CASM attentions. The experimental results are shown in **Table 10**.

As can be seen in **Figure 9**, for the data set in this paper, comparing the CASM module and CAM, the mAP and FPS of the CASM module are higher than those of CAM. Therefore, this paper uses the CASM module to fuse with the feature extraction network. From **Table 10**, we can see that the feature blending effect generated by redundant features has an impact on extraction accuracy, and the use of the attention mechanism also generates additional computational overhead, and the complexity of the network model increases with the number of CASM modules inserted (+0.6 B). Therefore, after experimental comparison, adding the CASM module after the CoAtNet module gives the best experimental results and an accuracy of 90.27%, which is 0.65 and 0.24% higher than the accuracy of the other two models. In this paper, the CASM attention module is embedded into the feature extraction network, which has the effect of suppressing invalid leaf and background features and enhancing effective grape leaf disease features, as well as improving the performance of correctly capturing the disease location information of grape leaf feature extraction network.

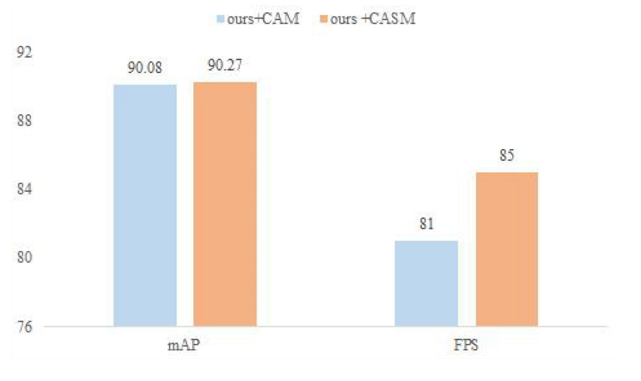


**FIGURE 9 |** Effects of different attention modules of coordinate attention shuffle mechanism (CASM) and CAM on model recognition ability.

**TABLE 10 |** Experimental results of adding an attention mechanism to different positions and numbers.

| Location | Number | mAP | FLOPs |
|---|---|---|---|
| Add CASM module to CoAtNet | ×1 | 89.62% | 189.5 B |
| | ×2 | 89.45% | 190.1 B |
| Add CASM module after CoAtNet | ×1 | 90.27% | 189.5 B |
| | ×2 | 90.10% | 190.1 B |
| Add CASM module after AMFM | ×1 | 90.03% | 189.5 B |
| | ×2 | 89.86% | 190.1 B |

## Effectiveness of Asymmetric Multi-Scale Fusion Module

To study the effectiveness of each part of AMFM on the grape leaf data set, this paper takes CoAtNet as the backbone, and AMFM is used as a single ablation experiment on the grape leaf data set, i.e., comparing MSFM, replacing 3*3 convolution of MSFM with ACB, replacing RELU of MSFM with SELU, and using AMFM to perform the experiment, with the experimental results shown in **Table 11**.

As seen in the experimental results given in **Table 11**, the AMFM proposed in this paper has a significant effect on the improvement of network identification performance. After adding AMFM to the CoAtNet network, the mAP is increased by 3.27%, the params are reduced by 0.79 M, the FLOPs are reduced by 0.8 B, and the test time is reduced by 6.28 s relative to adding MSFM, which indicates that the use of ACB and SELU in the AMFM can significantly reduce the number of params and the number of operations and improve the network training efficiency. When the traditional standard convolution in MSFM is experimentally replaced with ACB, the accuracy of using ACB is slightly improved compared to the traditional convolution (+1.34%), the params decrease by 0.33 M, the FLOPs decrease by 0.7 B, and the test time decreases by 2.08 s, which indicates that ACB can improve the performance of the underlying model. When ReLU is replaced with SELU in MSFM, mAP increases by 0.93%, params decrease by 0.46 M, FLOPs decrease by 1.1 B, and test time decreases by 3.33 s, which shows that the activation function SELU can better improve the convergence speed and recognition accuracy of the model compared to ReLU. The experimental results fully demonstrate the effectiveness of the AMFM proposed in this paper and further enhance the richness and representation capability of the features extracted by the model. In addition, adding the module further improves the results of grape leaf disease recognition in several comparative experiments.

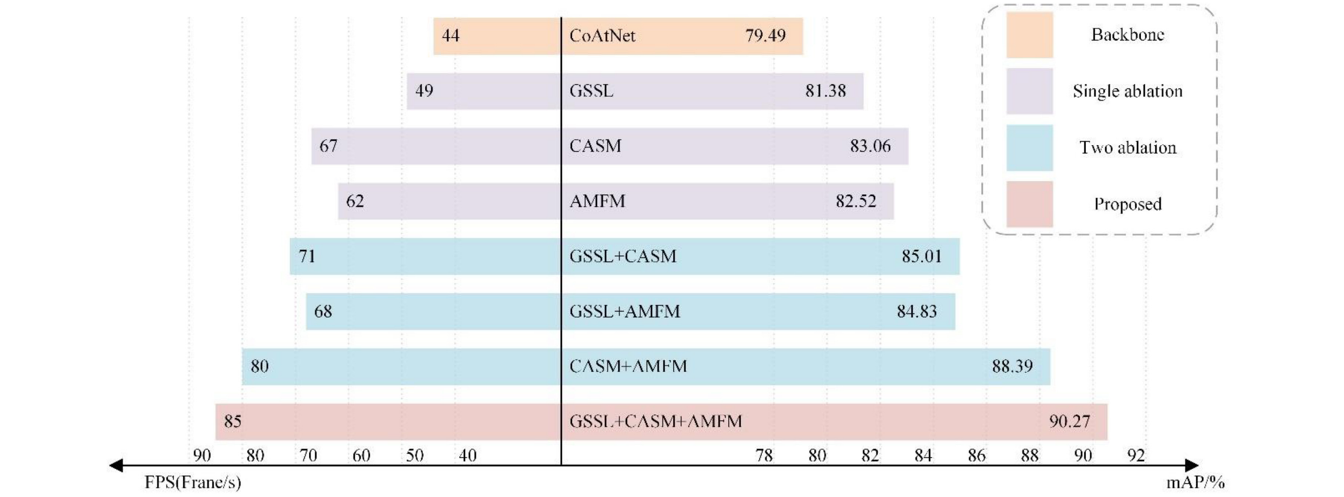


**FIGURE 10 |** Ablation experiments.

## Ablation Experiments

To verify the effectiveness of the CASM-AMFMNet, ablation experiments are conducted on the proposed CASM-AMFMNet network for the grape leaf image data set. Taking CoAtNet as the backbone, to which GSSL, CASM, and AMFM are gradually added, the performance of each module is analyzed by comparing the differences in detection accuracy and FPS. The overall ablation experiments are shown in **Figure 10**.

As can be seen from the ablation experiments, the model performance of the GSSL algorithm on the basis of the backbone improves by about 1.89% in mAP and about five in FPS. After adding only CASM, the map quality increases by about 3.57% in mAP and about 23 in FPS; adding AMFM alone increases it by about 3.03% in mAP and about 18 in FPS. To sum up, the CASM-AMFMNet increases 10.78% in mAP and increases in recognition speed (+41) compared to CoAtNet. The above seven sets of experimental results demonstrate the effectiveness of GSSL, CASM, and AMFM. This illustrates the high accuracy and speed of the network used in this paper for the identification of grape leaf diseases.

## Comparison of Coordinated Attention Shuffle Mechanism-Asymmetric Multi-Scale Fusion Module Net With Other Classification Models

Overall, 670 black rot, 647 black measles, 604 leaf blight, 564 downy mildew, and 675 healthy grape leaf images were selected as a fixed test dataset. All images in the dataset were not involved in the training of the model. Therefore, the generalization ability of the model was tested based on recognition accuracy, i.e., whether the model had the same high recognition accuracy for grape leaf images not involved in training. The performance of CASM-AMFMNet was further compared with the other three networks using the confusion matrix, as shown in **Figure 11**. Diagonal cells in the confusion matrix indicate the number of test samples correctly predicted by the model and non-diagonal cells the number of samples incorrectly predicted by the model.

Through the confusion matrix, we identified that, among the 670 black rot test samples of our CASM-AMFMNet network, four were incorrectly identified as healthy grape leaves, 10 as black measles, and 15 as leaf blight. Among the 604 leaf blight samples, only four were wrongly identified as healthy leaves and 14 as black measles. Six of the 564 downy mildew samples were incorrectly identified as healthy leaves and 13 of the 647 black measles test samples as healthy leaves. During classification of the four major diseases and healthy grape leaves, on the one hand, since disease spots of black rot and leafy blight were too small and limited in number at the early stage of the onset, similar to the images of healthy leaves, errors inevitably occurred in the identification of categories. On the other hand, misidentification phenomena were commonly encountered. (1) Black rot and leaf blight were easily misclassified as black measles due to connections in the later disease stages. Specifically, shape characteristics were similar to black measles, and color differentiation was not high, leading to classification errors. (2) Black rot manifested as small brown spots at the beginning, which could easily be confused with leaf blight. (3) Black measles presented as long brown spots on the leaf surface and leaf blight showed similar characteristics to black measles in terms of spot color, shape, and texture at the margins of the grape leaf surface. However, downy mildew differed significantly from the other three diseases in terms

**TABLE 11 |** A single ablation experiment of asymmetric multi-scale fusion module (AMFM).

| Method | mAP | Params | FLOPs | Testing time |
|---|---|---|---|---|
| CoAtNet with MSFM | 82.10% | 169.31 M | 191.0 B | 35.41 s |
| CoAtNet with MSFM (ACB) | 83.44% | 168.98 M | 190.3 B | 33.33 s |
| CoAtNet with MSFM (SELU) | 84.03% | 168.85 M | 189.9 B | 32.08 s |
| CoAtNet with AMFM | 85.37% | 168.52 M | 189.2 B | 29.13 s |

**TABLE 12 |** Performance evaluation of four types of networks.

| Class | Evaluation metrics | Black rot | Black measles | Leaf blight | Downy mildew | Healthy leaves | Average value |
|---|---|---|---|---|---|---|---|
| DICNN | Accuracy | 97.31% | 96.68% | 97.63% | 97.56% | 96.71% | 97.18% |
| | Precision | 93.28% | 94.44% | 93.71% | 92.73% | 93.63% | 93.56% |
| | Recall | 94.27% | 95.32% | 96.75% | 97.21% | 86.22% | 93.95% |
| | F1 Score | 93.77% | 94.88% | 95.21% | 94.92% | 89.77% | 93.71% |
| DMS-R Alexnet (Lv et al., 2020) | Accuracy | 97.31% | 96.68% | 97.63% | 97.56% | 96.71% | 97.18% |
| | Precision | 93.43% | 92.58% | 95.53% | 90.96% | 92.15% | 92.93% |
| | Recall | 93.85% | 91.31% | 92.32% | 95.18% | 92.42% | 93.02% |
| | F1 Score | 93.64% | 91.94% | 93.90% | 93.02% | 92.28% | 92.96% |
| Faster DR-IACNN | Accuracy | 98.23% | 98.20% | 98.16% | 97.28% | 95.41% | 97.46% |
| | Precision | 93.58% | 94.28% | 93.05% | 93.26% | 93.93% | 93.62% |
| | Recall | 94.71% | 96.06% | 97.91% | 95.46% | 85.91% | 94.01% |
| | F1 Score | 94.14% | 95.16% | 95.42% | 94.35% | 89.74% | 93.76% |
| CASM-AMFM Net (ours) | Accuracy | 98.01% | 98.42% | 97.91% | 98.86% | 98.70% | 98.38% |
| | Precision | 94.63% | 95.98% | 95.53% | 97.87% | 96.00% | 96.00% |
| | Recall | 95.92% | 96.28% | 93.67% | 95.83% | 97.89% | 95.92% |
| | F1 Score | 95.27% | 96.13% | 94.59% | 96.94% | 95.96% | 95.78% |

**TABLE 13 |** Comparison of the main performance of different methods.

| Method | A | $P_A$ | $R_A$ | $F1_A$ | mAP | Training time |
|---|---|---|---|---|---|---|
| DCNN (Ma et al., 2021) | 83.87% | 84.73% | 81.29% | 82.97% | 80.77% | 4 h 04 min 12 s |
| MediNET (Bhuiyan et al., 2021) | 76.99% | 76.83% | 77.29% | 77.06% | 78.39% | 5 h 58 min 27 s |
| YoloV4 (Richey and Shirvaikar, 2021) | 63.42% | 59.21% | 68.35% | 63.45% | 71.29% | 3 h 6 min 45 s |
| VirLeafNet (Joshi et al., 2021) | 85.12% | 84.54% | 77.87% | 81.06% | 81.73% | 4 h 28 min 03 s |
| BGCNN (Hridoy and Rakshit, 2022) | 91.59% | 91.20% | 91.00% | 91.10% | 84.44% | 3 h 32 min 57 s |
| DCGAN (Zhao et al., 2021) | 83.79% | 82.31% | 83.54% | 82.92% | 85.89% | 4 h 38 min 27 s |
| OPNN (Akanksha et al., 2021) | 82.38% | 81.16% | 83.28% | 82.21% | 81.23% | 3 h 50 min 3 s |
| DICNN | 93.58% | 93.56% | 93.95% | 93.71% | 84.81% | 3 h 30 min 51 s |
| DMS-R Alexnet | 92.94% | 92.93% | 93.02% | 92.96% | 85.84% | 4 h 26 min 9 s |
| Faster DR-IACNN | 93.64% | 93.62% | 94.01% | 93.76% | 87.48% | 3 h 48 min 13 s |
| CASM-AMFM Net(ours) | 95.95% | 96.00% | 95.92% | 95.78% | 90.27% | 3 h 13 min 27 s |

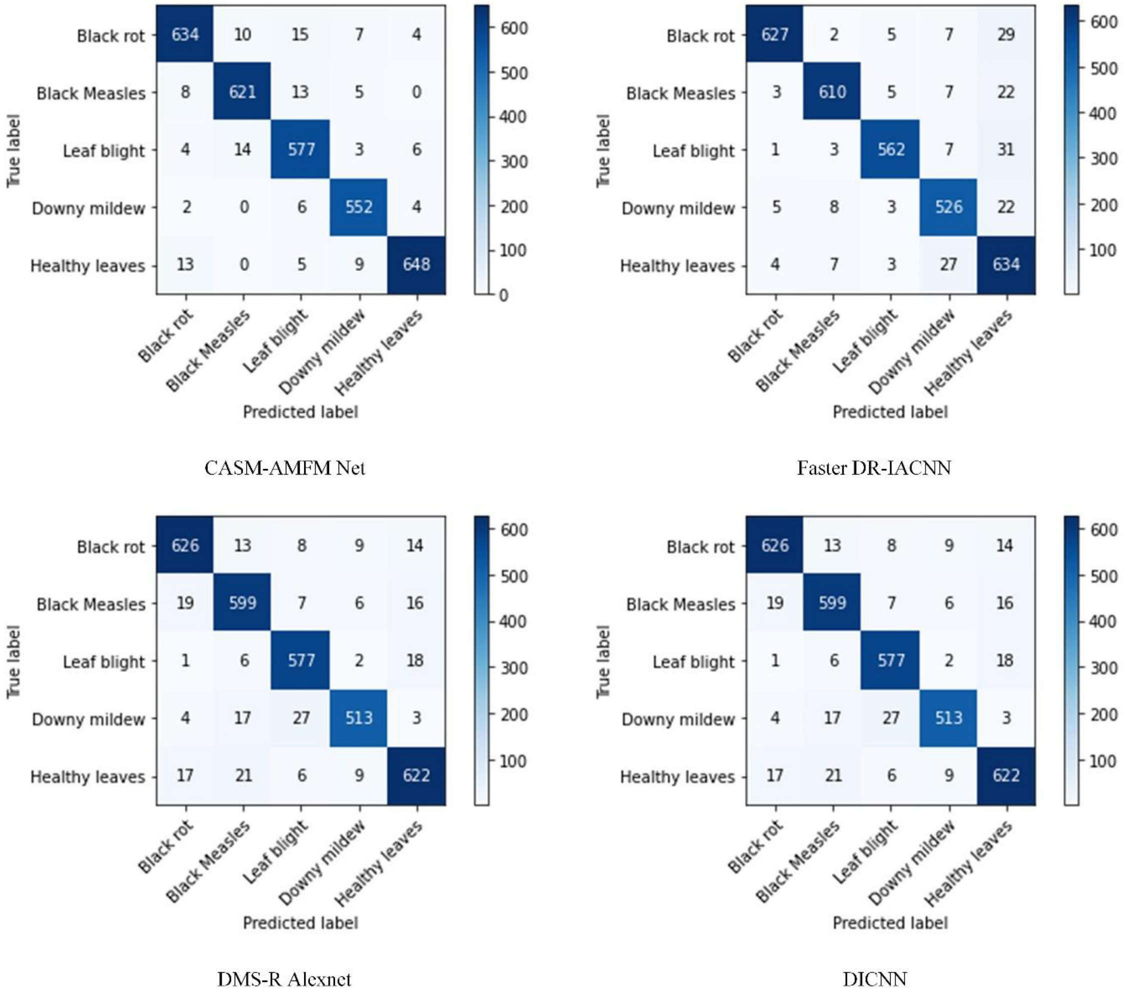
of both color and shape characteristics, resulting in a higher recognition rate. Among the five categories of grape leaves, 3,032 were correctly identified and classified with our network model, 2,959 with Faster DR-IACNN, 2,937 with DMS-R Alexnet, and 2,957 with DICNN. Based on the comparison data, we conclude that our newly developed network model has the highest feature recognition and classification efficiency.

Accuracy, precision, recall, and F1 scores of the four network models for four grape leaf diseases and healthy grape leaves were calculated using the confusion matrix as the model performance evaluation index (**Table 12**).

As shown in **Table 12**, the average accuracy of the model was 98.38%, which was 0.92, 1.2, and 1.2% higher than that of Faster DR-IACNN, DMS-R Alexnet, and DICNN, respectively. The average precision rate of 96.00% was 2.38, 3.07, and 2.44% higher relative to the above three models, respectively. Average recall value was 95.92%, which was 1.91, 2.9, and 1.97% higher, and the average F1 score of 95.78% was 2.05, 2.82, and 2.07% higher compared to the other three models, respectively. In summary, our model shows good recognition accuracy for grape leaf diseases.

To further validate the effectiveness of our model, the methods used by other researchers to resolve the image recognition problem of plant leaf datasets were introduced for comparison. Overall, 10 deep network models were selected, and experimental results are shown in **Table 13**.

As evident from **Table 13**, YoloV4 and MediNET had relatively low recognition accuracy of <80% for grape leaf disease images. The two networks are less focused on the context and location information between disease regions in the recognition process, and, therefore, recognition effects are poor. Accuracy levels of DCNN, VirLeafNet, DCGAN, and OPNN in the test set were estimated as 83.87, 85.12, 83.79, and 82.38%, respectively, which have a deeper network structure and can extract deep-seated grape leaf disease features but still do not consider the contextual and location information among disease regions. The algorithm proposed in this study incorporating contextual and location information among disease regions achieved 95.95% accuracy on the test set, which is higher than all the other network models examined (BGCNN, DICNN, DMS-R Alexnet, Faster DR-IACNN), with >90% accuracy (+4.36, +2.37, +3.01, +2.31%, respectively). CASM-AMFMNet also achieved more



CASM-AMFM Net

Faster DR-IACNN

DMS-R Alexnet

DICNN

**FIGURE 11 |** A confusion matrix for the identification of grape leaf diseases.

accurate localization compared to the training time required for CASM-AMFMNet and other deep models. The training time required for CASM-AMFMNet for grape leaf images was 3 h 13 min 27 s, which was significantly lower than the time taken by other models. Our findings clearly demonstrate enhanced recognition performance and robustness of CASM-AMFMNet developed in this study in terms of training time, recognition accuracy, precision, recall, and F1 score.

## DISCUSSION

Here, we constructed a CASM-AMFMNet model capable of effectively extracting shape, color, and texture features of grape leaf images to automatically improve identification and classification of healthy and diseased leaves. Application of the model to grape leaf images from the public PlantVillage Dataset (Kaggle, 2019) led to the recognition of four types of grape leaf diseases and healthy grape leaves with an accuracy of 97.21%. We further applied this novel model to the self-made banana leaf image dataset collected from Guangdong Province along with leaf images of apple, corn, and cherry from the PlantVillage dataset. The average classification accuracy of different diseases of the leaves from various plant species reached 94.41, 96.09, 94.77, and 95.92% for banana, apple, corn, and cherry, respectively. Comparative analysis suggested that the actual effect of these kinds of blades using our model is inferior to that of other methods for the above blades, and accuracy is additionally lower. Overall, the accuracy of our CASM-AMFMNet model in identifying grape leaf diseases was greatly enhanced compared with the other leaf types, and its classification effect was superior. As the shape of grape leaf edges is not a regular oval, the majority of disease spots are water stains and the edge contours are obvious. The color of disease spots is clearly distinct from that of the leaf surface, which is not observed for other leaf types.

The accuracy of grape leaf disease identification with the CASM-AMFMNet network was significantly higher than that with existing methods and solved the problem of low accuracy of multi-classification grape leaf disease identification to a certain extent, but further studies are necessary to resolve a number of issues. (1) The model training speed could potentially be reduced through more advanced parallel processing. (2) At present, we are limited to extraction and identification of the characteristics of only single grape leaf diseases using this method. Features that could enhance identification of two or more similar mixed diseases (Barbedo, 2016) or other diseases of grape leaves require further investigation. (3) Grape leaf contours are valuable for studying disease types, and methods to segment out the disease spots and leaf contours will be a focus of future research. The algorithm proposed in this study still needs further fine-tuning to improve the recognition rate of diseases from images with inconspicuous features. In particular, leaf features at the early onset of the grape disease onset are inconspicuous, and some disease characteristics are more similar, resulting in low recognition rates at the early stages of infection.

## CONCLUSION

To address the challenges of identification of grape leaf diseases, which are easily confused with the background, and difficulty of detection of small spots under complex backgrounds, we first constructed a dataset for grape leaf disease target recognition and classification, comprising a total of 15,824 images. Next, the GSSL algorithm was used to enhance the texture of grape leaves on the original image. After processing, this technique increased the map of the network by 1.89% and FPS by five. We further applied the CASM-AMFMNet model, which reduced the background interference in feature extraction without segmenting the background of grape leaves. The CASM-AMFMNet model was improved based on CoAtNet. The CASM module captured and pinpointed leaf diseases and effectively prevented confusion with the background, following which AMFM facilitated the identification of smaller target spots, which improved model recognition performance to a greater extent. Addition of CASM to CoAtNet increased mAP by 3.57% and FPS by 23, and adding AMFM to CoAtNet increased mAP by 3.03% and FPS by 18. Overall, CASM-AMFMNet was effective in identifying four grape diseases, specifically black rot, black measles, leaf blight, and downy mildew, with 98.01, 98.42, 97.91, and 98.86% accuracy, respectively, and healthy grape leaves with 98.7% accuracy. The average recognition accuracy of the five categories of grape leaves was >98%. Our collective results demonstrate enhanced performance of CASM-AMFMNet in identifying grape leaf spots and diseases with good accuracy and speed.

The CASM-AMFMNet model can be successfully applied for real-time disease identification from images of grape crop leaves under complex backgrounds, which is crucial for timely diagnosis and control of foliar pests and diseases that affect cultivated grape vines. In future studies, we plan to focus on application of the model to identify more leaf disease types and further improve the network by enhancing the feature extraction ability, reducing the recognition time and increasing accuracy. In addition, we will consider the transplantation of this model to cell phone platforms to enable more effective immediate identification of grape leaf diseases for raising agricultural productivity.

## 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

JS: methodology, writing–original draft preparation, conceptualization, and data curation. JZ: software, data acquisition, and investigation. GZ: validation and project administration. AC: supervision and funding acquisition. YoH: software. WH: writing, review, and editing. WC: model guidance. YhH: formal analysis and resources. LL: visualization. All authors contributed to the article and approved the submitted version.

## FUNDING

This work was supported by the Scientific Research Project of Education Department of Hunan Province (Grant No. 21A0179), in part by Changsha Municipal Natural Science Foundation (Grant No. kq2014160), in part by the Natural Science Foundation of Hunan Province (Grant No. 2021JJ41087), in part by the Natural Science Foundation of China (Grant No. 61902436), and in part by Hunan Key Laboratory of Intelligent Logistics Technology (2019TP1015).

## ACKNOWLEDGMENTS

We are grateful to all the members of the Food Collage of Central South University of forestry and technology for their advice and assistance in the course of this research.

## REFERENCES

Akanksha, E., Sharma, N., and Gulati, K. (2021). "OPNN: optimized probabilistic neural network based automatic detection of maize plant disease detection," in *Proceedings of the 2021 6th International Conference on Inventive Computation Technologies (ICICT)*, Lalitpur, 1322–1328. doi: 10.1109/ICICT50816.2021.9358763

Ampatzidis, Y., De Bellis, L., and Luvisi, A. (2017). iPathology: robotic applications and management of plants and plant diseases. *Sustainability* 9:1010. doi: 10.3390/su9061010

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

Barbedo, J. G. A. (2016). A review on the main challenges in automatic plant disease identification based on visible range images. *Biosyst. Eng.* 144, 52–60. doi: 10.1016/j.biosystemseng.2016.01.017

Bhuiyan, M., Abdullahil-Oaphy, M., Khanam, R. S., and Islam, M. (2021). "MediNET: a deep learning approach to recognize Bangladeshi ordinary medicinal plants using CNN," in *Soft Computing Techniques and Applications*, eds S. Borah, R. Pradhan, N. Dey, and P.

Gupta (Singapore: Springer), 371–380. doi: 10.1007/978-981-15-7394-1_35

Bock, C. H., Poole, G. H., Parker, P. E., and Gottwald, T. R. (2010). Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging. *Crit. Rev. Plant Sci.* 29, 59–107. doi: 10.1080/07352681003617285

Chouhan, S. S., Kaul, A., Singh, U. P., and Jain, S. (2018). Bacterial foraging optimization based radial basis function neural network (BRBFNN) for identification and classification of plant leaf diseases: an automatic approach towards plant pathology. *IEEE Access* 6, 8852–8863. doi: 10.1109/access.2018.2800685

Chouhan, S. S., Singh, U. P., and Jain, S. (2020). Applications of computer vision in plant pathology: a survey. *Arch. Comput. Methods Eng.* 27, 611–632. doi: 10.1007/s11831-019-09324-0

Clinton, S. (2017). *Implementasi Deteksi Tepi Berbasis Algoritma Sobel*. Available online at: http://eprints.dinus.ac.id/id/eprint/22238 (accessed November 2021).

Cruz, A. C., Luvisi, A., De Bellis, L., and Ampatzidis, Y. (2017). X-FIDO: an effective application for detecting olive quick decline syndrome with deep learning and data fusion. *Front. Plant Sci.* 8:1741. doi: 10.3389/fpls.2017.01741

Dai, Z., Liu, H., Le, Q. V., and Tan, M. (2021). Coatnet: marrying convolution and attention for all data sizes. *Adv. Neural Inf. Process. Syst.* 34, 3965–3977. doi: 10.48550/arXiv.2106.04803

Ding, X., Guo, Y., Ding, G., and Han, J. (2019). "Acnet: strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*, Seoul, 1911–1920. doi: 10.1109/ICCV.2019.00200

Fawke, S., Doumane, M., and Schornack, S. (2015). Oomycete interactions with plants: infection strategies and resistance principles. *Microbiol. Mol. Biol. Rev.* 79, 263–280. doi: 10.1128/MMBR.00010-15

Gao, L., and Lin, X. (2019). Fully automatic segmentation method for medicinal plant leaf images in complex background. *Comput. Electronics Agric.* 164:104924. doi: 10.1016/j.compag.2019.104924

Ghosal, S., Blystone, D., Singh, A. K., Ganapathysubramanian, B., Singh, A., and Sarkar, S. (2018). An explainable deep machine vision framework for plant stress phenotyping. *Proc. Natl. Acade. Sci. U.S.A.* 115, 4613–4618. doi: 10.1073/pnas.1716999115

Hou, Q., Zhou, D., and Feng, J. (2021). "Coordinate attention for efficient mobile network design," in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, TN, 13713–13722. doi: 10.48550/arXiv.2103.02907

Hridoy, R. H., and Rakshit, A. (2022). "BGCNN: a computer vision approach to recognize of yellow mosaic disease for black gram," in *Computer Networks and Inventive Communication Technologies*, eds S. Smys, R. Bestak, R. Palanisamy, I. Kotuliak (Singapore: Springer), 189–202. doi: 10.1007/978-981-16-3728-5_14

Huang, T., Yang, R., Huang, W., Huang, Y., and Qiao, X. (2018). Detecting sugarcane borer diseases using support vector machine. *Inf. Process. Agric.* 5, 74–82. doi: 10.1016/j.inpa.2017.11.001

Jaisakthi, S. M., Mirunalini, P., and Thenmozhi, D. (2019). "Grape leaf disease identification using machine learning techniques," in *Proceedings of the 2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*, Chennai, 1–6. doi: 10.1109/ICCIDS.2019.8862084

Joseph, S. I. T., Sasikala, J., Juliet, D. S., and Velliangiri, S. (2021). Hybrid spatio-frequency domain global thresholding filter (HSFGTF) model for SAR image enhancement. *Patt. Recogn. Lett.* 146, 8–14. doi: 10.1016/j.patrec.2021.02.023

Joshi, R. C., Kaushik, M., Dutta, M. K., Srivastava, A., and Choudhary, N. (2021). VirLeafNet: automatic analysis and viral disease diagnosis using deep-learning in *Vigna mungo* plant. *Ecol. Inf.* 61:101197. doi: 10.1016/j.ecoinf.2020.101197

Kaggle (2019). *PlantVillage Dataset*. Available online at: https://www.kaggle.com/abdallahalidev/plantvillage-dataset (accessed November 2021).

Kaggle (2021). *PlantifyDr Dataset*. Available online at: https://www.kaggle.com/lavaman151/plantifydr-dataset (accessed November 2021).

Khan, M. A., Akram, T., Sharif, M., Awais, M., Javed, K., Ali, H., et al. (2018). CCDF: automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep CNN features. *Comput. Electron. Agric.* 155, 220–236. doi: 10.1016/j.compag.2018.10.013

Kingma, D. P., and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv* [Preprint]. doi: 10.48550/arXiv.1412.6980

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

Kundu, N., Rani, G., Dhaka, V. S., Gupta, K., Nayak, S. C., Verma, S., et al. (2021). IoT and interpretable machine learning based framework for disease prediction in pearl millet. *Sensors* 21:5386. doi: 10.3390/s21165386

Liu, B., Ding, Z., Tian, L., He, D., Li, S., and Wang, H. (2020a). Grape leaf disease identification using improved deep convolutional neural networks. *Front. Plant Sci.* 11:1082. doi: 10.3389/fpls.2020.01082

Liu, B., Tan, C., Li, S., He, J., and Wang, H. (2020b). A data augmentation method based on generative adversarial networks for grape leaf disease identification. *IEEE Access* 8, 102188–102198. doi: 10.1109/ACCESS.2020.2998839

Liu, Y., Zheng, C., Zheng, Q., and Yuan, H. (2018). Removing Monte Carlo noise using a Sobel operator and a guided image filter. *Vis. Comput.* 34, 589–601. doi: 10.1007/s00371-017-1363-z

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

Ma, L., Guo, X., Zhao, S., Yin, D., Fu, Y., Duan, P., et al. (2021). Algorithm of strawberry disease recognition based on deep convolutional neural network. *Complexity* 2021:6683255. doi: 10.1155/2021/6683255

Mu, Q., Wang, X., Wei, Y., and Li, Z. (2021). Low and non-uniform illumination color image enhancement using weighted guided image filtering. *Comput. Vis. Media* 7, 529–546. doi: 10.1007/s41095-021-0232-x

Narvekar, P. R., Kumbhar, M. M., and Patil, S. N. (2014). Grape leaf diseases detection & analysis using SGDM matrix method. *Int. J. Innov. Res. Comput. Commun. Eng.* 2, 3365–3372. doi: 10.21090/ijaerd.01099

Nerva, L., Turina, M., Zanzotto, A., Gardiman, M., Gaiotti, F., Gambino, G., et al. (2019). Isolation, molecular characterization and virome analysis of culturable wood fungal endophytes in esca symptomatic and asymptomatic grapevine plants. *Environ. Microbiol.* 21, 2886–2904. doi: 10.1111/1462-2920.14651

Nuthan, B. R., Meghavarshinigowda, B. R., Maharachchikumbura, S. S. N., Mahadevakumar, S., Marulasiddaswamy, K. M., Sunilkumar, C. R., et al. (2021). Morphological and molecular characterization of Neopestalotiopsis vitis associated with leaf blight disease of *Manilkara zapota*—a new record from India. *Lett. Appl. Microbiol.* 73, 352–362. doi: 10.1111/lam.13521

Padol, P. B., and Yadav, A. A. (2016). "SVM classifier based grape leaf disease detection," in *Proceedings of the 2016 Conference on Advances in Signal Processing (CASP)*, Lisbon, 175–179. doi: 10.1109/CASP.2016.7746160

Pashaei, E. (2021). "Medical Image Enhancement using Guided Filtering and Chaotic Inertia Weight Black Hole Algorithm," in *Proceedings of the 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Ankara, 37–42. doi: 10.1109/ISMSIT52890.2021.9604701

Pawara, P., Okafor, E., Schomaker, L., and Wiering, M. (2017). "Data augmentation for plant classification," in *International Conference on Advanced Concepts for Intelligent Vision Systems*, eds J. Blanc-Talon, R. Penne, W. Philips, D. Popescu and P. Scheunders (Cham: Springer), 615–626. doi: 10.1007/978-3-319-70353-4_52

Peng, Y., Zhao, S., and Liu, J. (2021). Fused-deep-features based grape leaf disease diagnosis. *Agronomy* 11:2234. doi: 10.3390/agronomy11112234

Pound, M. P., Atkinson, J. A., Townsend, A. J., Wilson, M. H., Griffiths, M., Jackson, A. S., et al. (2017). Deep machine learning provides state-of-the-art performance in image-based plant phenotyping. *Gigascience* 6:gix083. doi: 10.1093/gigascience/gix083

Richey, B., and Shirvaikar, M. V. (2021). "Deep learning based real-time detection of Northern Corn Leaf Blight crop disease using YoloV4," in *Proceedings of the Real-Time Image Processing and Deep Learning*, (Tyler, TX: The University of Texas), 1173606. doi: 10.1117/12.2587892

Sabra, A., Netticadan, T., and Wijekoon, C. (2021). Grape bioactive molecules, and the potential health benefits in reducing the risk of heart diseases. *Food Chem. X* 12:100149. doi: 10.1016/j.fochx.2021.100149

Singh, P., Bhandari, A. K., and Kumar, R. (2022). Naturalness balance contrast enhancement using adaptive gamma with cumulative histogram and median filtering. *Optik* 251:168251. doi: 10.1016/j.ijleo.2021.168251

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., and Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Comput. Intell. Neurosci.* 2016, 1–11. doi: 10.1155/2016/3289801

Sun, Y., Jiang, Z., Zhang, L., Dong, W., and Rao, Y. (2019). SLIC_SVM based leaf diseases saliency map extraction of tea plant. *Comput. Electron. Agric.* 157, 102–109. doi: 10.1016/j.compag.2018.12.042

Tomoiaga, L., and Chedea, V. S. (2020). The Behaviour of Some Grapevine Varieties to the *Guignardia bidwellii* Fungus Attack. *Bull. Univ. Agric. Sci. Vet. Med. Cluj Napoca Hortic.* 77, 122–127. doi: 10.15835/buasvmcn-hort

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

Wang, X., and Wang, C. (2020). *MSFM: Multi-Scale Fusion Module for Object Detection*. Available online at: https://openreview.net/references/pdf?id=IblXk1C75Q (accessed November 2021).

Wu, J., Li, G., Wang, C., Liu, H., Zhang, S., and Zhang, G. (2021). "Extended guided image filtering for contrast enhancement," in *Proceedings of the 2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, Shenzhen, 1–4. doi: 10.1109/ICMEW53276.2021.9455989

Xie, X., Ma, Y., Liu, B., He, J., Li, S., and Wang, H. (2020). A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks. *Front. Plant Sci.* 11:751. doi: 10.3389/fpls.2020.00751

Zhao, Y., Chen, Z., Gao, X., Song, W., Xiong, Q., Hu, J., et al. (2021). "Plant Disease Detection using Generated Leaves Based on DoubleGAN," in *Proceedings of the 2021 IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Piscataway, NJ. doi: 10.1109/TCBB.2021.3056683

**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.

frontiers | Frontiers in Plant Science

ORIGINAL RESEARCH
published: 06 June 2022
doi: 10.3389/fpls.2022.806711

# Multi-Information Model for Large-Flowered Chrysanthemum Cultivar Recognition and Classification

Jue Wang[1]†, Yuankai Tian[1]†, Ruisong Zhang[2], Zhilan Liu[1], Ye Tian[2]* and Silan Dai[1]*

[1] Beijing Key Laboratory of Ornamental Plants Germplasm Innovation and Molecular Breeding, Beijing Laboratory of Urban and Rural Ecological Environment, Key Laboratory of Genetics and Breeding in Forest Trees and Ornamental Plants of Ministry of Education, National Engineering Research Center for Floriculture, School of Landscape Architecture, Beijing Forestry University, Beijing, China, [2] College of Technology, Beijing Forestry University, Beijing, China

OPEN ACCESS

**Edited by:**
Valerio Giuffrida,
Edinburgh Napier University,
United Kingdom

**Reviewed by:**
Nguyen Quoc Khanh Le,
Taipei Medical University, Taiwan
Damiano Perri,
University of Florence, Italy

***Correspondence:**
Ye Tian
tytoemail@bjfu.edu.cn
Silan Dai
silandai@sina.com

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

**Specialty section:**
This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science

**Received:** 01 November 2021
**Accepted:** 16 March 2022
**Published:** 06 June 2022

**Citation:**
Wang J, Tian Y, Zhang R, Liu Z, Tian Y and Dai S (2022) Multi-Information Model for Large-Flowered Chrysanthemum Cultivar Recognition and Classification. Front. Plant Sci. 13:806711. doi: 10.3389/fpls.2022.806711

The traditional Chinese large-flowered chrysanthemum is one of the cultivar groups of chrysanthemum (*Chrysanthemum* × *morifolium* Ramat.) with great morphological variation based on many cultivars. Some experts have established several large-flowered chrysanthemum classification systems by using the method of comparative morphology. However, for many cultivars, accurate recognition and classification are still a problem. Combined with the comparative morphological traits of selected samples, we proposed a multi-information model based on deep learning to recognize and classify large-flowered chrysanthemum. In this study, we collected the images of 213 large-flowered chrysanthemum cultivars in two consecutive years, 2018 and 2019. Based on the 2018 dataset, we constructed a multi-information classification model using non-pre-trained ResNet18 as the backbone network. The model achieves 70.62% top-5 test accuracy for the 2019 dataset. We explored the ability of image features to represent the characteristics of large-flowered chrysanthemum. The affinity propagation (AP) clustering shows that the features are sufficient to discriminate flower colors. The principal component analysis (PCA) shows the petal type has a better interpretation than the flower type. The training sample processing, model training scheme, and learning rate adjustment method affected the convergence and generalization of the model. The non-pre-trained model overcomes the problem of focusing on texture by ignoring colors with the ImageNet pre-trained model. These results lay a foundation for the automated recognition and classification of large-flowered chrysanthemum cultivars based on image classification.

**Keywords: large-flowered chrysanthemum, image classification, cultivar recognition, cultivar classification, deep learning**

## INTRODUCTION

The traditional Chinese large-flowered chrysanthemum (larger-flowered chrysanthemum) is a particular group of chrysanthemum (*Chrysanthemum* × *morifolium* Ramat.) derived from wild Chrysanthemum species through domestication and selection for over 2,600 years in China (Dai et al., 2012). The cultivar group of large-flowered chrysanthemum has over 3,000 cultivars to date, and they exhibit a rich diversity in floral morphology. Thus, this cultivar group possesses excellent aesthetic value and prospects for the market (Zhang et al., 2014a; Dai and Hong, 2017; Su et al., 2019).

Cultivar identification and classification are very important for production and communication (Yu, 1963). Similar to the cultivar classification system for other ornamental plants, such as Lily,[1] Rosa,[2] Daffodils,[3] and Peony,[4] the classification of large-flowered chrysanthemum is also based on critical morphological traits, such as flower color (Hong et al., 2012), flower type (Zhang et al., 2014b), petal type (Song et al., 2018), and leaf type (Song et al., 2021), and classifies a considerable number of cultivars into multiple groups with high similarity within-group and high variation between groups.

At present, the researchers widely use the classification system of 9-color series based on flower color (Hong et al., 2012), five petal types, and 30 flower types based on flower shape (Wang, 1993). In the former, the large-flowered chrysanthemum is divided into nine color groups by quantitative classification. In the latter, the petal type (flat, spoon, tubular, anemone, and peculiar) is the first criteria of the classification, and the second is the flower shape (the petal details and the combination relationship among petals). The above two systems determine the distribution of floral characteristics in large-flowered chrysanthemum (**Figure 1**).

However, faced with the vast number of large-flowered chrysanthemum cultivars, morphological variability within a cultivar, and similarity to other cultivars, the above classification system's efficiency, and accuracy are often challenged.

Deep learning is an emerging area of machine learning for tackling large data analytics problems (Ubbens and Stavness, 2017). As one of the most popular branches of machine learning research, deep learning has been widely employed and has attracted more attention from various domains, such as protein prediction (Le and Huynh, 2019; Tng et al., 2022), plant disease detection (Abade et al., 2021), plant yield, growth prediction (Ni et al., 2020; Shibata et al., 2020), and animal identification (Norouzzadeh et al., 2018; Spiesman et al., 2021).

A significant trend in plant recognition in recent years has been to use deep learning for plant image classification (Wldchen and Mder, 2018). The network that applies to deep learning for plant images classification is the deep convolutional neural network (DCNN), which establishes the classification model by extracting features of plant images. So far, DCNN derived a series of network structures, such as VGG (Visual Geometry Group) (Simonyan and Zisserman, 2015), GoogleNet (Szegedy et al., 2015), and ResNet (Residual Neural Network) (He et al., 2016). ResNet is the first classification network that surpasses human accuracy in classification tasks (Russakovsky et al., 2015). At present, the ResNet-based classification model has been widely used in plant image research, such as plant age judgment (Yue et al., 2021), flowering pattern analysis (Jiang et al., 2020), and root image analysis (Wang et al., 2020).

For plant recognition, users only need to provide images of plant organs, such as leaves (Zhang et al., 2020) and flowers (Seeland et al., 2017; Liu et al., 2019), to complete the recognition of plants. With the application of related research and development, over 30,000 species of plants could be recognized (Mäder et al., 2021). However, because of the difficulties in data acquisition, many deep learning methods used pre-training models based on ImageNet (He et al., 2015, 2016; Russakovsky et al., 2015; Chattopadhay et al., 2018) as the backbone network, but ImageNet-trained CNNs are strongly biased toward recognizing textures and not sensitive to color (Geirhos et al., 2018). When using the classifier constructed by the ImageNet-trained model to test the images of large-flowered chrysanthemum in 2008, we also found that the top-5 results and test images have similar textures but a difference in color (**Figure 2**).

A number of previous studies utilized deep learning for plant image classification, which only provided single taxonomic information of plants (Waeldchen and Maeder, 2018). A recent study on large-flowered chrysanthemum image classification (Liu et al., 2019) established a recognition model with the output of cultivar name. It cannot fully meet the requirements of large-flowered chrysanthemum recognition and classification. It is important to recognize large-flowered chrysanthemum, and the classification according to corresponding petal type and flower type is also necessary for practical application. These results have practical value in market communication and landscape application.

We also consider the large intra-cultivar visual variation. The large-flowered chrysanthemum belonging to the same cultivars may show considerable differences in their morphological characteristics depending on their different abiotic factors, development stage, and opening periods, which is a challenge to the generalization of the model (Liu et al., 2019).

Based on previous research, for large-flowered chrysanthemum recognition and classification, and to overcome the bias to texture, we proposed a multi-information classification model that can output flower type, petal type, and cultivar name. We also tested the model's generality on the datasets of different years.

## MATERIALS AND METHODS

### Plant Material

According to the previous classification system of flower type and color (Wang, 1993; Hong et al., 2012), we selected large-flowered chrysanthemums in the chrysanthemum resource nursery (in Dadongliu nursery in Beijing) of the research group. To cover all flower colors, petal types, and flower types of chrysanthemum cultivars, we selected 126 cultivars

[1] https://www.lilies.org/culture/types-of-lilies

[2] https://www.rose.org/single-post/rose-classifications

[3] https://thedaffodilsociety.com/a-guide-to-dafodils/classification-system

[4] https://americanpeonysociety.org/learn/herbaceous-peonies/flower-types-anatomy

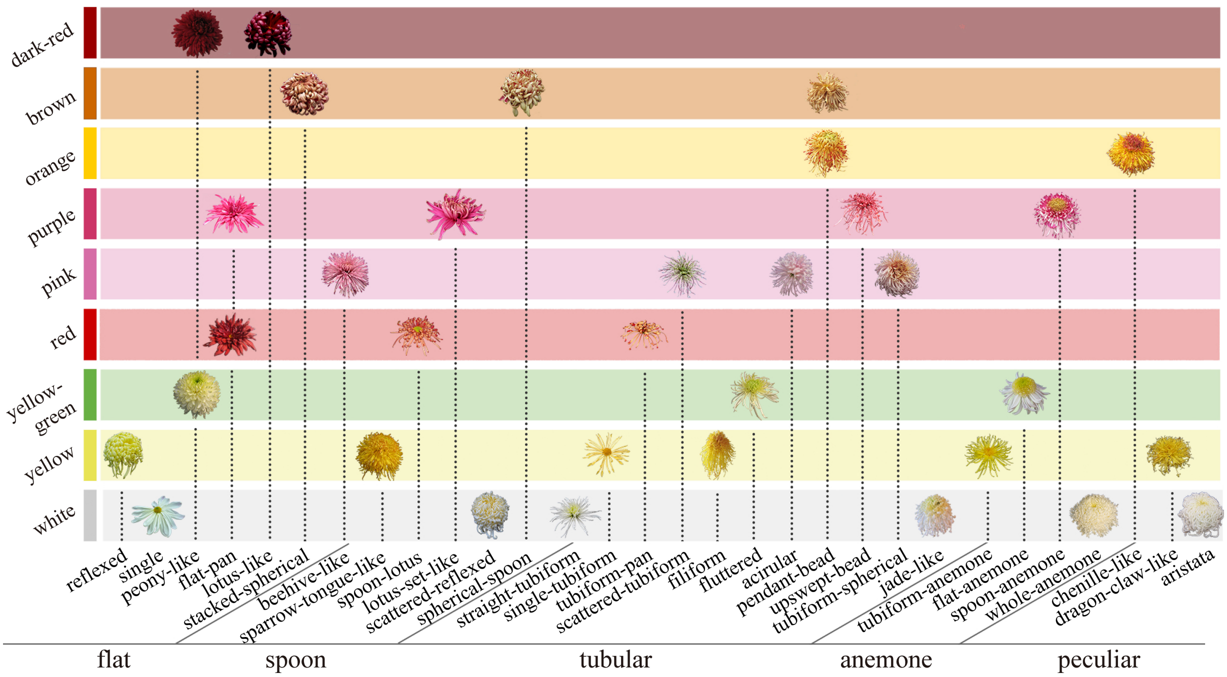


**FIGURE 1 |** Classification system of large-flowered chrysanthemum based on flower color, petal type, and flower type.

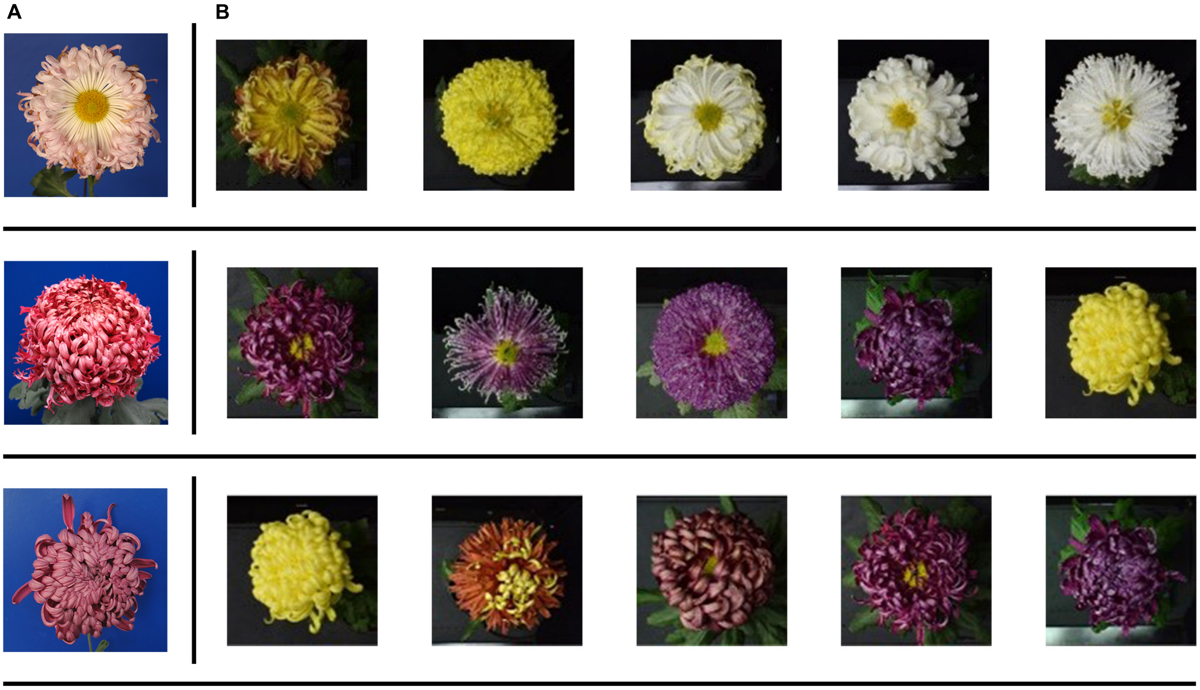


**FIGURE 2 |** Examples of misclassified images by the pre-training model. **(A)** The test images. **(B)** The Top-5 results.

in 2018 (as shown in **Supplementary Table 1**) and 117 cultivars in 2019 (as shown in **Supplementary Table 2**). After removing the duplication, there were 213 cultivars in 2018 and 2019.

Referring to the Chinese Chrysanthemum book (Zhang and Dai, 2013), we accomplished the cultivation and management of large-flowered chrysanthemum in the Dadongliu nursery in Beijing in 2018 and 2019, respectively.

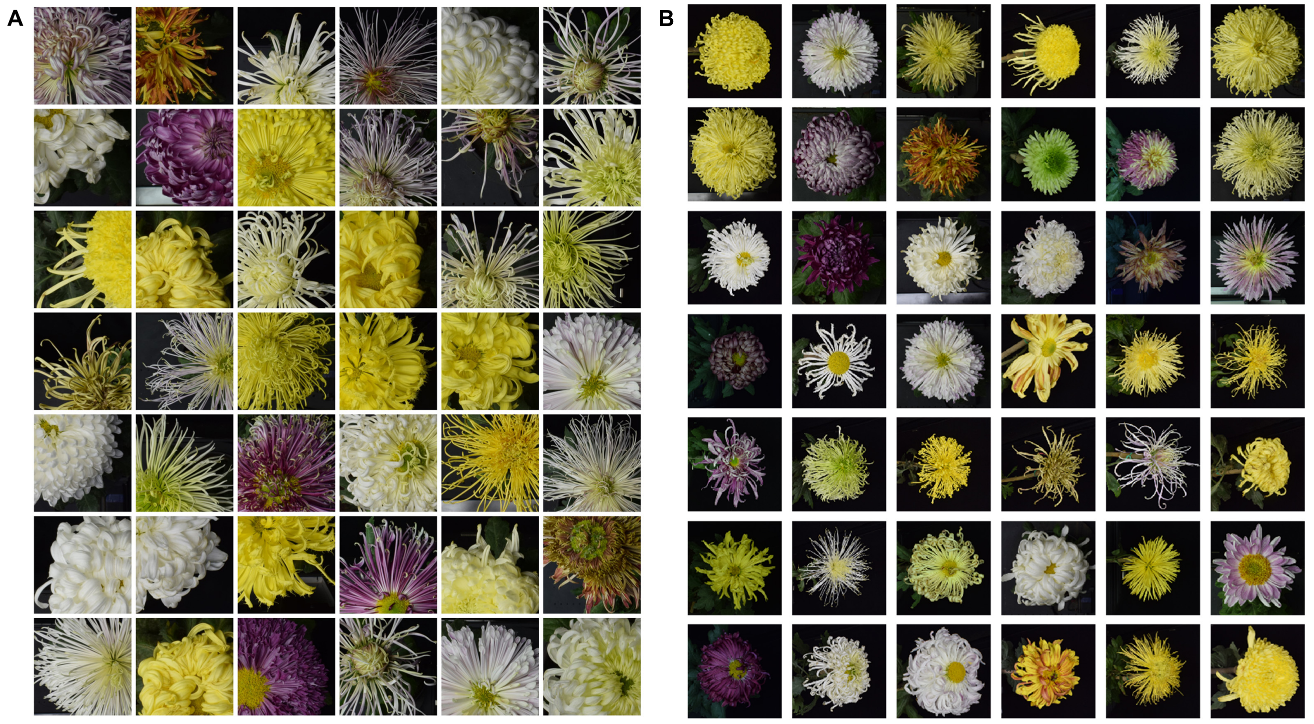


**FIGURE 3 |** Sample images from dataset training and validation. **(A,B)** Are training and validation dataset, respectively.

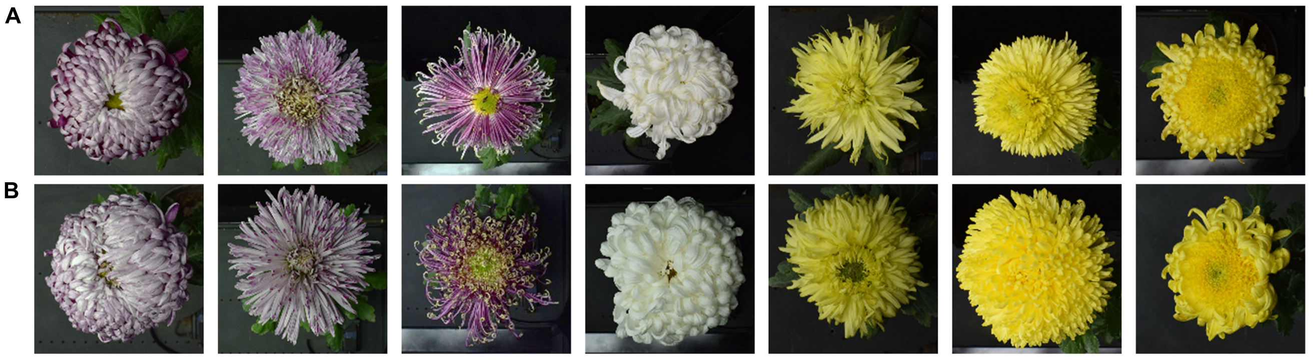


**FIGURE 4 |** Some sample images in the 2018 and 2019 datasets. Row **(A,B)** belong to the 2018 and 2019 datasets, respectively.

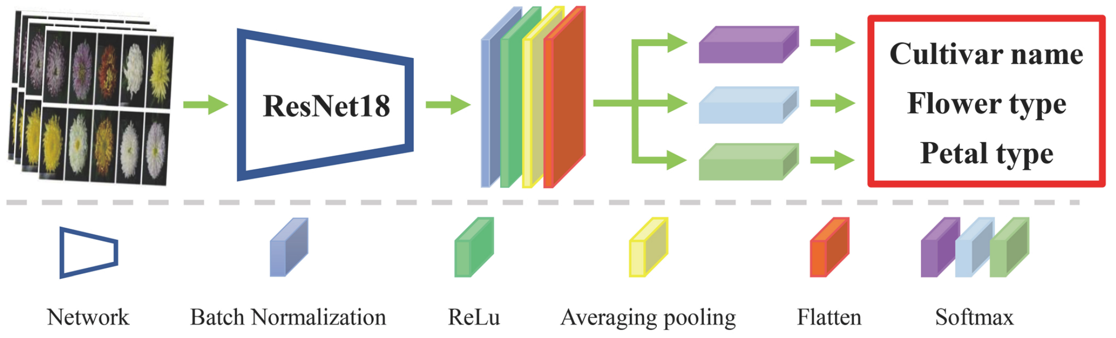


**FIGURE 5 |** Architecture of the network model.

## Image Acquisition and Labeling

The image acquisition of large-flowered chrysanthemum was carried out during flowering periods in November to December in 2018 and 2019. The image acquisition device and image acquisition process are the same as the study by Liu et al. (2019). The image resolution was 6,000 × 6,000 pixels, and the format was PNG. In the gathered images, each cultivar had at least 2–3 individuals. We photographed each individual from the top view and oblique views while ignoring the background.

All the collected images were accurately and uniformly marked using LabelImg v1.7.0 software. See **Supplementary Table 3** for cultivar name, petal type, and flower type marking.

## Dataset Construction

### 2018 Dataset (Training Dataset and Validation Dataset)

The 2018 dataset contained 126 cultivars. To balance the samples, we randomly selected 80 images from each cultivar for 10,080 images. A total of 80% of the images were used for training and 20% for validating. Data augmentation plays a crucial role in improving classification performances. However, large-flowered chrysanthemum recognition is similar to face recognition, so some common data enhancement methods are not adopted. For example, color is essential information for flowers. Therefore, methods such as color jitter and gray scales are unsuitable for actually identified scenes. In addition, the symmetry of chrysanthemum structure, rotation, and flip operations are invalid. For large-flowered chrysanthemum image recognition, the deep network model focuses on the local information of the image (Liu et al., 2019), so the cropping is only for the training dataset. The original image of the training dataset is scaled to 256 × 256 pixels then randomly cropped to 224 × 224 pixels image patch. By random cropping, we expanded the number of the training dataset (**Figure 3A**) by 10 times to 80,640. The validation dataset (**Figure 3B**) is 2,016 images used to determine model architecture and hyper-parameters.

### 2019 Dataset (Test Dataset)

The 2019 dataset contained 2,556 images belonging to 117 cultivars, including 640 images of 30 similar cultivars as the 2018 dataset. This dataset formed the test dataset. **Figure 4** shows some of the same cultivars in 2018 and 2019. Because of the differences in climate environment and photo time in 2018 and 2019, the flower type of the same cultivar has changed to some extent, which can test the model's generalization.

## Devices

The models were built and trained on the Ubuntu 16.04 system, based on Intel Xeon Gold 5120 CPU and 4 NVIDIA Titan Xp 16GB GPU hardware platform.

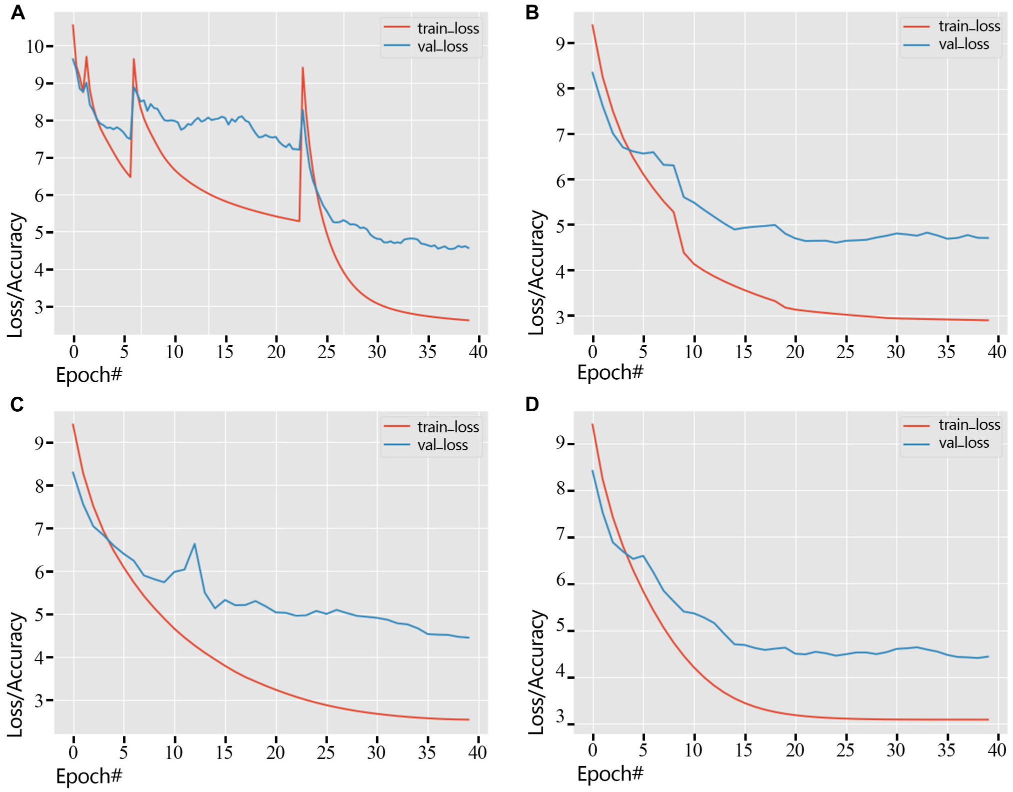


**FIGURE 6 |** Error variation of training dataset and validation dataset under different strategies (2018 dataset). **(A)** Standard-decay. **(B)** Step-decay. **(C)** Line-decay. **(D)** Poly-decay.

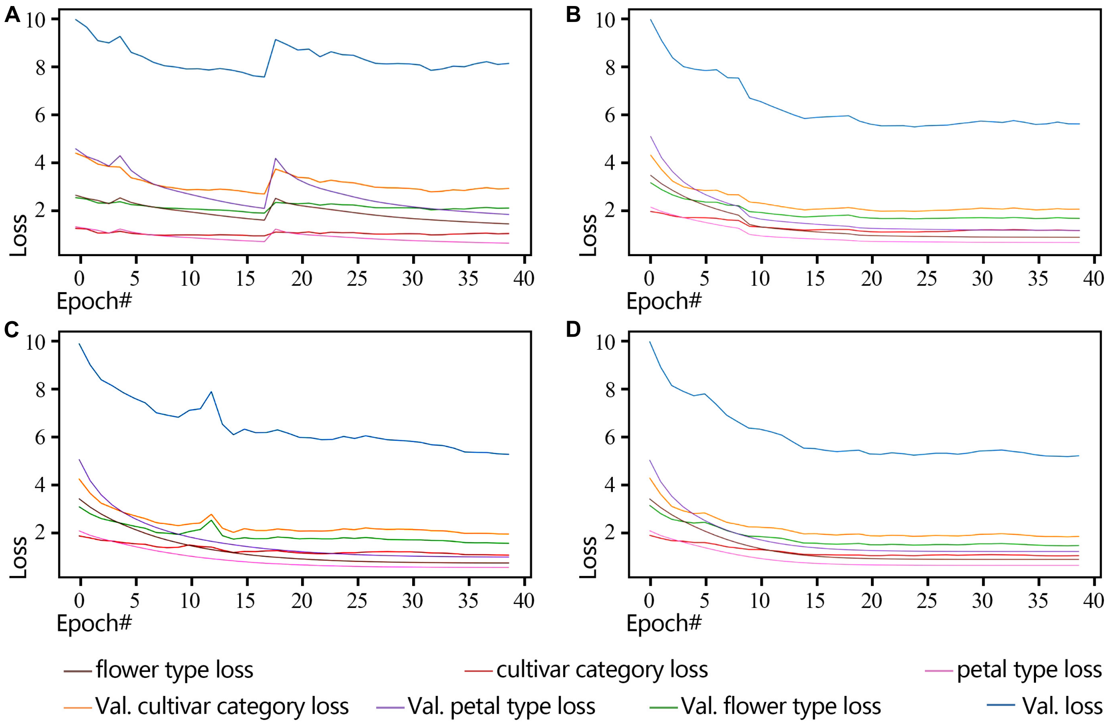


**FIGURE 7 |** ResNet18 Loss curve of a non-pre-training model during training and validation (2018 dataset). **(A)** Standard-decay. **(B)** Step-decay. **(C)** Line-decay. **(D)** Poly-decay.

**TABLE 1 |** Comparison of decay strategy performance in the validation dataset.

| Decay strategy | Cultivar name classification | | | | Petal type classification | | | | Flower type classification | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Top-1 (%) | Top-5 (%) | recall | F1-score | Top-1 (%) | Top-5 (%) | recall | F1-score | Top-1 (%) | Top-5 (%) | recall | F1-score |
| Standard-decay | 73.96 | 94.98 | 0.69 | 0.68 | 78.27 | 98.34 | 0.78 | 0.77 | 72.68 | 95.08 | 0.71 | 0.7 |
| Step-decay | 74.15 | 95.45 | 0.72 | 0.71 | **82.43** | 99.29 | **0.82** | 0.81 | 72.92 | 96.69 | 0.71 | 0.7 |
| Line-decay | 76.04 | 96.54 | 0.74 | 0.73 | 81.49 | 99.34 | 0.8 | 0.79 | 74.91 | 97.3 | 0.73 | 0.72 |
| Poly-decay | **77.61** | **97.16** | **0.78** | **0.77** | 81.68 | **99.72** | 0.8 | **0.79** | **76.42** | **98.44** | **0.75.** | **0.74** |

*Text in bold indicates the best value in each category.*

## Approach

Many researchers used the pre-trained network model on the ImageNet dataset to extract image features. As mentioned above, ImageNet-trained CNNs are strongly biased toward recognizing textures and ignoring color. However, for large-flowered chrysanthemum, color is an essential characteristic for classification. We used the script of non-pre-trained ResNet18 (He et al., 2016) as the backbone network. Due to the limited amount of data, we abandoned the deeper network, such as ResNet50. The network comprised three parallel softmax classifiers to get a richer feature representation of large-flowered chrysanthemum images. It means that the network model has three outputs about botanical information of cultivar name, flower type, and petal type, respectively. **Figure 5** shows the network structure. The features of the images are flatten-layer output with 512-dimensions. Keras was used for our experiments. The DCNN was initialized by He initialization (He et al., 2015).

## Model Training

The total loss function (1) includes cultivar name loss, flower type loss, and petal type loss.

$$\boldsymbol{Loss} = \boldsymbol{Loss}_{\boldsymbol{name\_cultivar}} + \boldsymbol{Loss}_{\boldsymbol{types\_flower}} + \boldsymbol{Loss}_{\boldsymbol{types\_petal}} \tag{1}$$

The label smoothing (Szegedy et al., 2016) was used to increase the convergence rate in the training phase. The optimization method was the Stochastic Gradient Descent (SGD), Momentum of 0.9, training used a batch size of 32, and was terminated after 40 epochs.

In training, the network's weights are updated according to a certain strategy. The weight update function is defined as (2).

$$\mathbf{W} + = \alpha \times \boldsymbol{gradient} \tag{2}$$

α is the learning rate, the gradient is the corresponding weight gradient.

**TABLE 2 |** Influence of cropping or non-cropping models on a generalization of 2019 cultivars classification.

| | Cultivar name classification accuracy (%) | | Flower type classification accuracy (%) | | Petal type classification accuracy (%) | |
|---|---|---|---|---|---|---|
| | **Top-1** | **Top-5** | **Top-1** | **Top-5** | **Top-1** | **Top-5** |
| Non-cropping | 8.15 | 25.00 | **75.64*** | **95.4*** | 61.5 | 88.37 |
| Cropping | **32.81*** | **70.62*** | 61.25 | 84.37 | **70.12*** | **96.34*** |

*The superscript "*" denotes a significant difference ($P < 0.05$) between cropping or non-cropping using one-way ANOVA.*
*The highest accuracy is marked in bold face.*

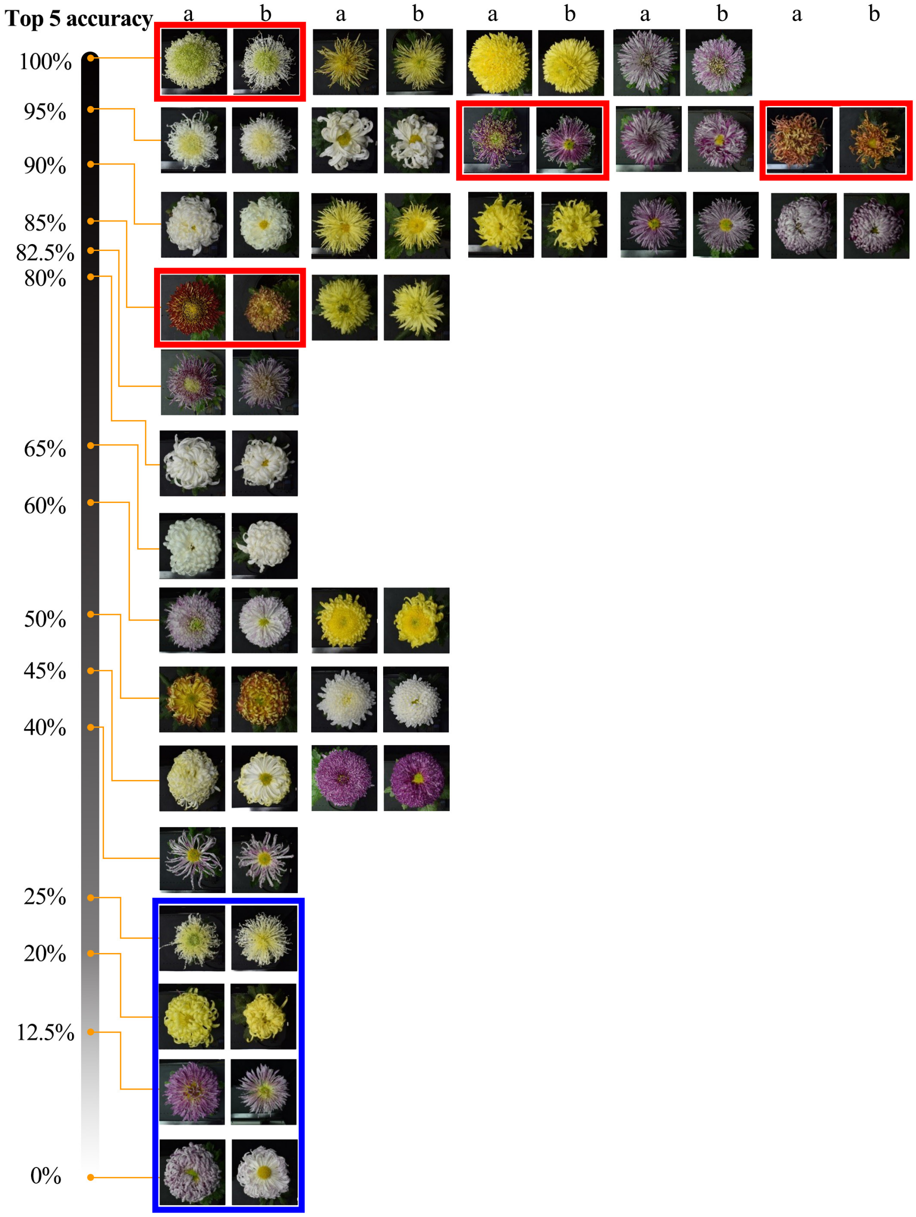


**FIGURE 8 |** Top-5 recognition results of 2019 dataset. The two adjacent columns are the same cultivar, a and b belong to the 2019 and 2018 datasets. Red and blue frames indicate some cultivars with noticeable morphological changes but with high and low identification accuracy.

Because the model used the script of the ResNet18 network with nearly 32M parameters and 80,640 pictures in the training dataset, the network was easy to overfit. To avoid overfitting, we used different

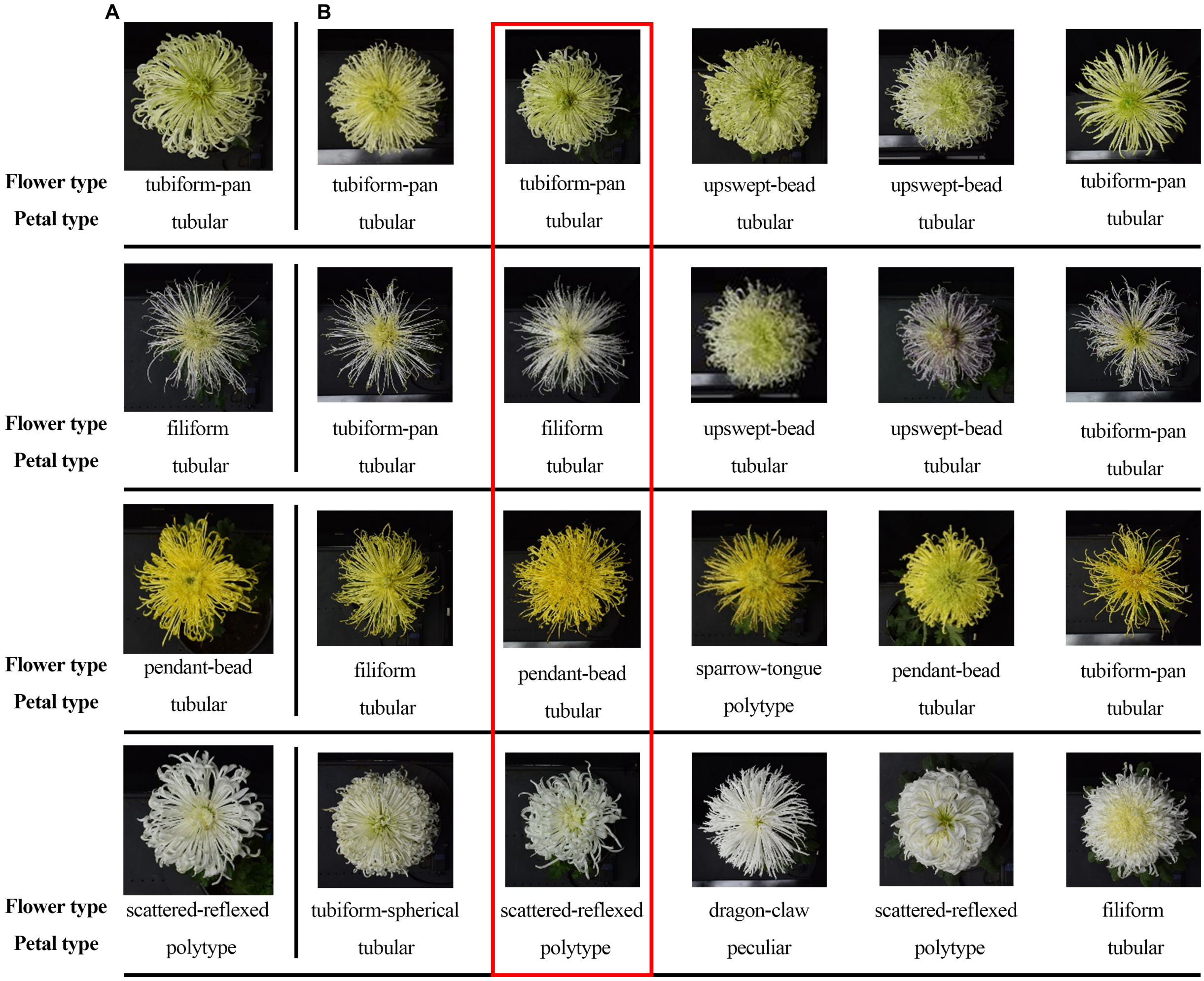


**FIGURE 9 |** Top-5 results of some cultivars. **(A)** is test images, **(B)** is Top-5 results, and the possibility from left to right gradually reduced. The text below the picture means the corresponding flower type and petal type for each cultivar.

learning rate adjustment methods (Rosebrock, 2021) for comparison.

In the standard-decay strategy, the initial init_lr = 1e-2, the function is (3).

$$\alpha = init\_lr \times \frac{1.0}{1.0 + decay \times iterations}, decay = \frac{init\_lr}{echos} \tag{3}$$

In the step-decay strategy, the initial init_lr = 1e-2, the formula is (4).

$$\alpha_{E+1} = \alpha_I \times F^{(1+E)/D} \tag{4}$$

$\alpha_I$ represents the initial learning rate, $F$ represents the learning rate factor, $F = 0.25$, $E$ represents the current *echo*, $D$ represents each *echo* to adjust the learning rate, $D = 10$.

In the line-decay strategy, the initial *init_lr* = 1e-2, the formula is (5).

$$\alpha_{E+1} = \alpha_I \times (1-\frac{E}{echos})^{pow} \tag{5}$$

When $pow = 1$, it means the line-decay approach.

In the poly-decay strategy, the initial init_lr = 1e-2 (when pow = 5, it means the poly-decay strategy), the formula is (5).

## Evaluation of Results

We used the Top-k accuracy to evaluate the model. If the K results include the correct categories, we consider the results valid. It took the average value of all images in each cultivar test dataset as the Top-1 and Top-5 accuracy. In addition, we used the F1-score and recall as the evaluation metrics.

## Feature Analysis

After training, we extracted the 512-dimensional image features for the 126 cultivars in the 2018 validation dataset and 87 cultivars in the 2019 test dataset. By the AP clustering (Frey and Dueck, 2007), the correlation between image features and large-flowered chrysanthemum phenotype was analyzed. To explore the class separability of cultivars in the PCA space, we colored the points according to their characteristic labels.

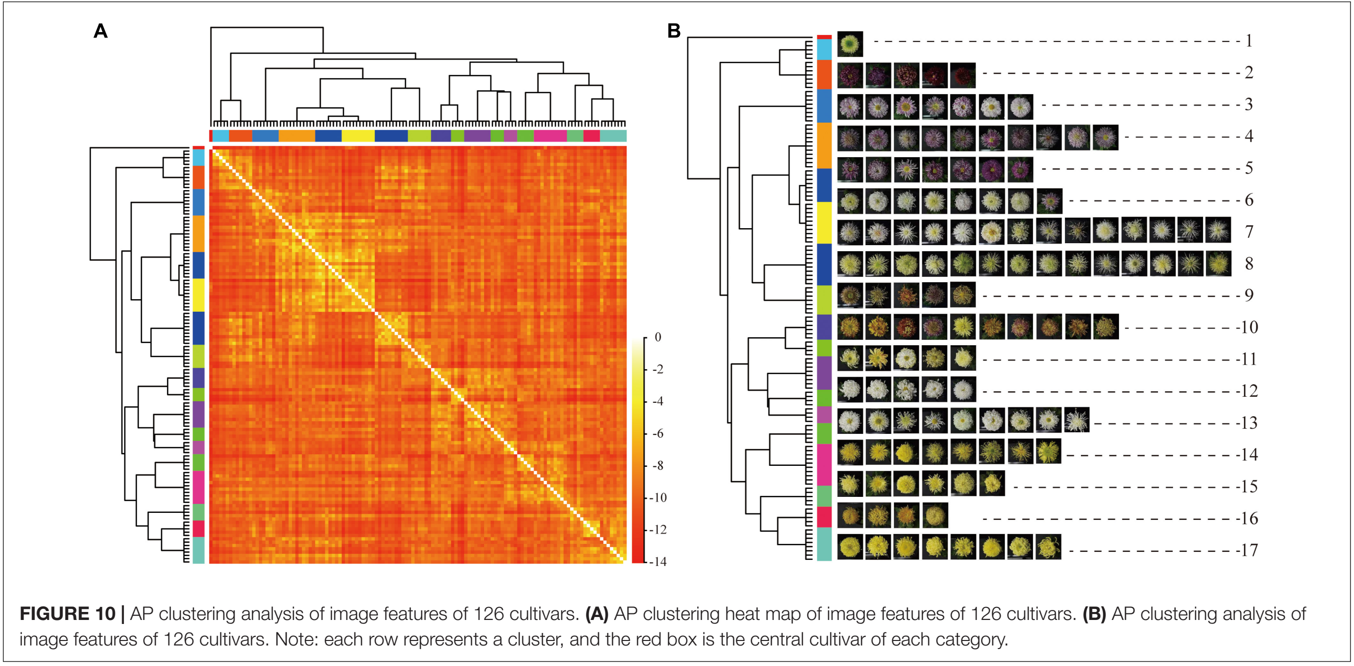


**FIGURE 10 |** AP clustering analysis of image features of 126 cultivars. **(A)** AP clustering heat map of image features of 126 cultivars. **(B)** AP clustering analysis of image features of 126 cultivars. Note: each row represents a cluster, and the red box is the central cultivar of each category.

**TABLE 3 |** Interpretation of principal component analysis.

| Principal components | Eigenvalues | Importance of components | Cumulative proportion |
|---|---|---|---|
| PC1 | 57.437 | 11.218 | 11.218 |
| PC2 | 49.524 | 9.673 | 20.891 |
| PC38 | 1.707 | 0.333 | 90.286 |
| PC185 | 0.050 | 0.010 | 99.003 |
| PC492 | 0.003 | 0.001 | 99.990 |

# RESULTS

## Model Training

We showed the loss change process of the training dataset and the validation dataset by different learning rate adjustment methods in **Figure 6**. As shown in **Figure 6A**, Standard-decay caused the convergence to be unstable, causing the loss curve to spike twice. At the same time, other learning rate adjustment methods were relatively stable, and the loss curve maintained stability. In the training process, the loss of the validation dataset gradually decreases, and the loss of the training process using the poly-decay strategy decreases most smoothly (**Figure 6D**), which reflects that the poly-decay strategy can ensure the convergence of the network.

By observing the loss changes of the three classifiers, i.e., cultivar name, flower type, and petal type, we obtain the best convergence result using the poly-decay strategy (**Figure 7**). Among the three classifiers, the petal type classifier has the highest accuracy because the model pays more attention to local features such as petal type than global features such as cultivar name and flower type (**Figure 7D**).

The validation accuracies of Top-1 and Top-5 of cultivar name, flower type, and petal type of 2018 validation dataset using four kinds of decay strategies are shown in **Table 1**. Except that the petal type of Top-1 accuracy is lower than the step-decay strategy, the other accuracies of the poly-decay are the highest, poly-decay also had the highest recall and F1-score.

## Model Generalization Performance

The generalization ability to the 640 images of the same cultivars in the 2019 dataset and the 2018 dataset, while comparing the influence of image cropping, is shown in **Table 2**. For petal type, a local feature, the model's accuracy with cropping image patches is higher than that without cropping. Compared with the petal type, the flower type, which reflects the overall features of the large-flowered chrysanthemum, the model's accuracy with cropping image patches is lower than that without cropping. For cultivar name, the model's accuracy with cropping image patches is four times higher than without cropping, confirming the previous conclusion that the large-flowered chrysanthemum classification model focuses on the local (Liu et al., 2019).

The model could accurately identify the morphological changes of the same cultivars to some extent (**Figure 8**). The results show that among 30 cultivars, the accuracy of 9 cultivars is greater than or equal to 90%, and 18 cultivars are greater than or equal to 80%. For some cultivars (red frames marked) with significant morphological changes, the model can identify them with a high accuracy rate (over 85%). We also found some cultivars (blue frames marked) that changed to the degree that

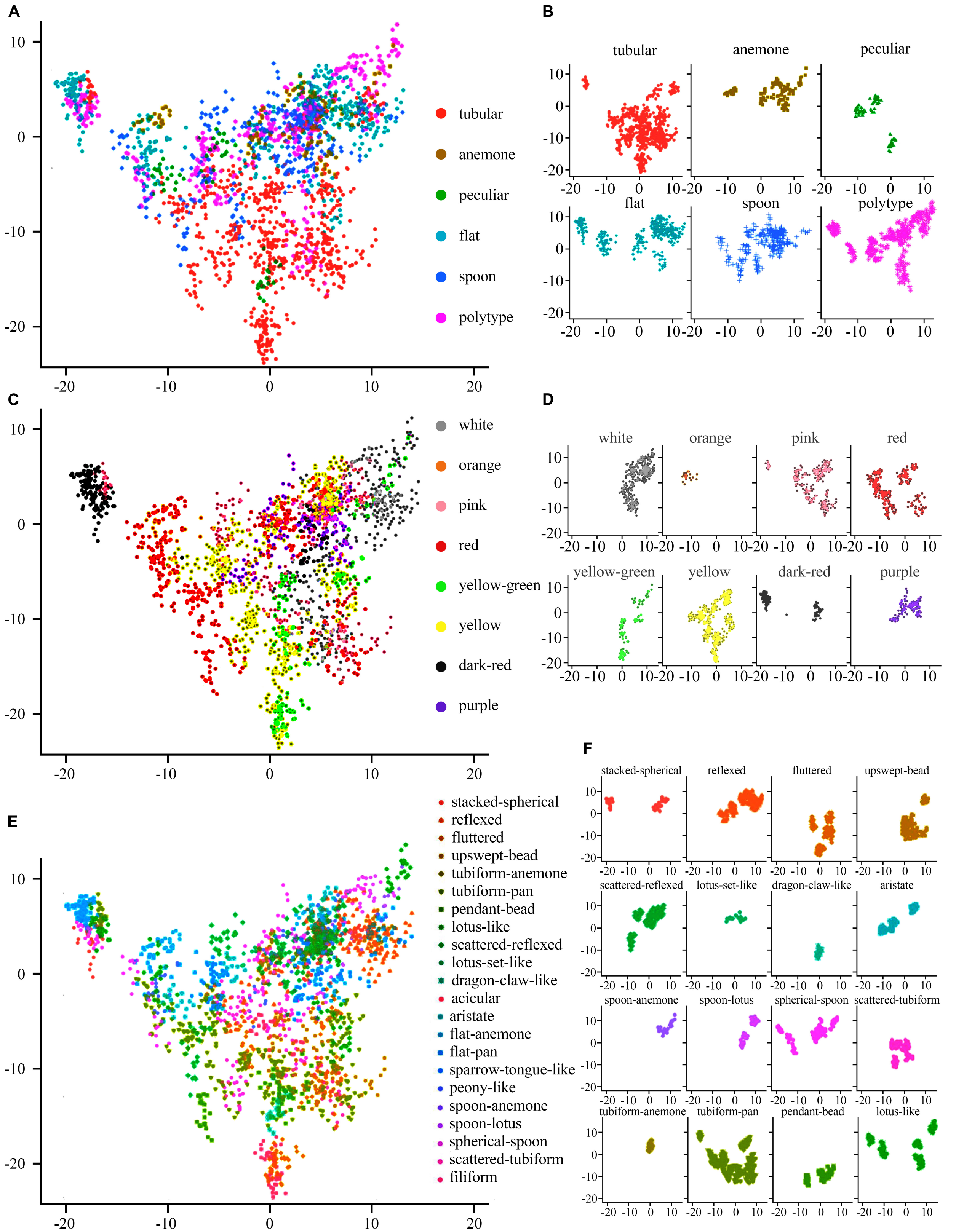
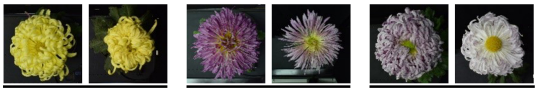


**FIGURE 11 |** Distribution of various flower colors, petal types, and flower types on the first two PCs. **(A,C,E)** The comprehensive distribution results of flower colors, petal types, and flower types, respectively. **(B,D,F)** The individual distribution results of flower colors, petal types, and flower types, respectively.



| | | | |
|---|---|---|---|
| Top-1 | 0% | 2.5% | 0% |
| Top-5 | 20% | 12.5% | 0% |

FIGURE 12 | Top-1 and Top-5 identification accuracy of some cultivars. Two adjacent images are the same cultivars from datasets in 2018 and 2019, respectively.

exceeded the model's generalization, and we will further analyze this problem in our discussion.

The Top-5 results' details of some cultivars are shown in **Figure 9**. The Top-5 results are consistent in the flower color, which indicates that the model can recognize the vital character of flower color. When focusing on petal type and flower type, we found the model has higher accuracy for petal type.

## Affinity Propagation Cluster Analysis

The AP clustering algorithm was used to cluster and compare the image features of the 2018 validation dataset (**Figure 10**). The maximum number of iterations is 1,000, the attenuation coefficient is $\lambda = 0.9$, and the convergence condition is $\delta = 100$. The Ap clustering result is shown in **Figure 10B**. The 126 cultivars in 2018 were automatically clustered into 17 categories and the central cultivars of each category were extracted, as shown in the red frame.

The large-flowered chrysanthemum images were not labeled with any color information, and the AP clustering still showed prominent clustering features by color. The second cluster belonged to dark red, the 12th cluster belonged to white, and the 14th, 15th, and 17th clusters belonged to yellow. While in other clusters, although they were not all the same color, the cultivars in the same clusters belonged to similar colors, such as the third, fourth, and fifth clusters belonged to pink-purple; sixth and seventh clusters are yellow-white; 14th, 15th, 16th, and 17th are yellow-orange.

For the petal type features, the image features of tubular petals were the most discriminative; most were clustered in clusters seventh and eighth, while the flat petal, the spoon petal, and the peculiar petal had poor clustering distribution. The spoon petal was the most relaxed.

## Principal Components Analysis

Based on the non-pre-training Resnet18, we extracted the image feature of 87 new cultivars from the test dataset in 2019 for PCA. We show the results in **Table 3**. The interpretation degree of the first principal component (PC1) to the original data is 11.22%, and PC2 to the original data is 9.67%. The first two principal components (PCs) accounted for 20.89% of the total variance, while the first 38 PCs account for approximately 90% of the total variance in the original data.

The 1,916 image features were visualized on the first two PCs, and colored by flower color (**Figures 11C,D**), petal type (**Figures 11A,B**), and flower type (**Figures 11E,F**), respectively. The image features of white, orange, purple, yellow, and pink large-flowered chrysanthemum cultivars were concentrated. The image features of yellow-green, red, and dark-red large-flowered chrysanthemum cultivars were scattered.

For the distribution of petal types, we can clearly distinguish the tubular from other petal types on the distribution map (**Figure 11A**), while other petal types were mixed above the distribution map.

For the flower types (**Figures 11E,F**), it shows that the first two PCs have a certain explanatory effect on the image characteristic flower types of large-flowered chrysanthemum cultivars. Among flower types, Lotus set like, dragon claw like, and tubiform anemone have the best aggregation. However, some flower types, such as tubiform pan, Lotus like, and spherical spoon, not only have poor aggregation but also have a great overlap with each other.

# DISCUSSION

## Dynamic Identification of Large-Flowered Chrysanthemum Image

For the 30 same cultivars in 2018 and 2019, the results of the Top-1 and Top-5 were only 32.81% and less than 70%. It was apparent that the vast differences in the images of large-flowered chrysanthemum obtained at different flowering stages directly affected the results of recognition and classification. For some cultivars with low recognition rates, the images obtained in different years have apparent flower color and flower type changes. Sometimes this change shows a comprehensive and complex shift in floral characteristics, but the model lacks sufficient generalization for this circumstance (**Figure 12**).

The large-flowered chrysanthemum belonging to the same cultivars may show considerable changes in different developmental stages, nutrient levels, or stress conditions. The single classification network cannot be accurate enough for this dynamic process. In terms of dynamic phenotypic identification, good progress has been made in related studies on leaves. By obtaining leaf images at different growth stages (Zhang et al., 2021) or under different stress conditions (Hao et al., 2020), researchers have established a classification model that can recognize the changing leaves through multi-feature or multi-scale input. However, compared to leaf images, flowers are only available during a short period of the year. Due to being complex 3D objects, there is a considerable number of variations in viewpoint, occlusions, and scale of flower images. In the future, one problem to be solved is establishing a dynamic identification model that can accurately identify the 2D images of large-flowered chrysanthemum in various states.

## Deep Features Analysis

It is difficult to interpret the principle of deep network decision-making and analyze the deep features (Castelvecchi, 2016; Lipton, 2018). When the primary purpose is to advance biological research based on accurate prediction, the interpretability of the deep learning model becomes crucial. Whether the classification model of large-flowered chrysanthemum has botanical application value, it needs to analyze whether the classification model is based on the relevant botanical characters and quantify botanical characters' importance. Liu trained the VGG-16 network with the transfer learning method and extracted 4096-dimensional features, but the clustering results and visual analysis of extracted deep features did not reflect the general distribution rule (Liu et al., 2019). In this paper, the AP clustering and PCA analyzed the 512-dimensional features clustering, and the AP clustering showed that flower color presents high aggregation. For PCA, PC1, and PC2 only account 20.89% of the original data. The image features of chrysanthemum cultivars of each flower color, petal type, and flower type overlap in the first two PCs, and the boundary between different groups is not obvious. It shows that PC1 and PC2 are still not enough to explain the petal type, flower type, and flower color. Further research still needs more principal components.

## CONCLUSION

The results show that the Top-5 accuracy of the ResNet18 non-pre-training model based on the poly-decay strategy is 70.62%. The image processing, model training scheme, and learning rate adjustment method significantly influence the model's generalization performance. The AP clustering was used to analyze the deep features. The AP clustering result showed that the 126 cultivars in the 2018 dataset were divided into 17 clusters; the flower color and the petal type clustering effect were better than the flower type. Because the structure of the classification network limits the number of categories, it cannot meet the requirement of category increase. In this case, metric learning is a solution.

## DATA AVAILABILITY STATEMENT

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

## AUTHOR CONTRIBUTIONS

YT conceived the study. SD completed the work of cultivars' confirmation, guided sample selection, and cultivation. JW and ZL undertook the cultivation, sample collection, and data labeling of the chrysanthemum. YT and RZ performed the experiments, analyzed the data, and prepared the figures and tables. JW and YKT completed the first draft. SD and YT revised the manuscript. All authors read and approved the final manuscript.

## FUNDING

This study was performed under the National Natural Science Foundation of China (Nos. 31530064 and 31471907), the National Key Research and Development Plan (No. 2018YFD1000405), the Beijing Science and Technology Project (No. Z191100008519002), and the Major Research Achievement Cultivation Project of Beijing Forestry University (No. 2017CGP012).

## ACKNOWLEDGMENTS

We are particularly indebted to Beijing Dadongliu Nursery for providing test sites. We thank Yushan Ji and Shuo Wang for their guidance on plant material cultivation.

## SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2022.806711/full#supplementary-material

## REFERENCES

Abade, A., Ferreira, P. A., and de Barros Vidal, F. (2021). Plant diseases recognition on images using convolutional neural networks: a systematic review. *Comp. Elect. Agricult.* 185:6125. doi: 10.1016/j.compag.2021.106125

Castelvecchi, D. (2016). Can we open the black box of AI? *Nature* 538, 20–23. doi: 10.1038/538020a

Chattopadhay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N. (2018). "Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. (Lake Tahoe, NV).

Dai, S., and Hong, Y. (2017). *Chrysanthemum: rich diversity of flower color and full possibilities for flower color modification*. Leuven: ISHS, 193–208.

Dai, S., Zhang, L., Luo, X., Bai, X., Xu, Y., and Liu, Q. (2012). *Advanced research on chrysanthemum germplasm resources in china*. Leuven: ISHS, 347–354.

Frey, B. J., and Dueck, D. (2007). Clustering by passing messages between data points. *Science* 315, 972–976. doi: 10.1126/science.1136800

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2018). *ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness*. Vienna: ICLR

Hao, X., Jia, J., Gao, W., Guo, X., Zhang, W., Zheng, L., et al. (2020). MFC-CNN: an automatic grading scheme for light stress levels of lettuce (Lactuca sativa L.) leaves. *Comp. Elect. Agricult.* 179:5847. doi: 10.1016/j.compag.2020.105847

He, K., Zhang, X., Ren, S., and Jian, S. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *2015 IEEE International Conference on Computer Vision (ICCV)*. Piscataway.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (IEEE Computer Society)*, Piscataway. 770–778.

Hong, Y., Bai, X., Sun, W., Jia, F., and Dai, S. (2012). The numerical classification of chrysanthemum flower color phenotype. *Acta Horticult. Sin.* 39, 1330–1340.

Jiang, Y., Li, C., Xu, R., Sun, S., Robertson, J. S., and Paterson, A. H. (2020). DeepFlower: a deep learning-based approach to characterize flowering patterns of cotton plants in the field. *Plant Methods* 16:156. doi: 10.1186/s13007-020-00698-y

Le, N. Q. K., and Huynh, T. T. (2019). Identifying SNAREs by incorporating deep learning architecture and amino acid embedding representation. *Front. Physiol.* 10:1501. doi: 10.3389/fphys.2019.01501

Lipton, Z. C. (2018). The mythos of model interpretability. *Commun. ACM* 61, 36–43. doi: 10.1145/3233231

Liu, Z., Wang, J., Tian, Y., and Dai, S. (2019). Deep learning for image-based large-flowered chrysanthemum cultivar recognition. *Plant Methods* 15:7. doi: 10.1186/s13007-019-0532-7

Mäder, P., Boho, D., Rzanny, M., Seeland, M., Wittich, H. C., Deggelmann, A., et al. (2021). The Flora Incognita app – Interactive plant species identification. *Methods Ecol. Evol.* 44, 1131–1142. doi: 10.1111/2041-210x.13611

Ni, X., Li, C., Jiang, H., and Takeda, F. (2020). Deep learning image segmentation and extraction of blueberry fruit traits associated with harvestability and yield. *Horticult. Res.* 7:3. doi: 10.1038/s41438-020-0323-3

Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C., et al. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proc. Nat. Acad. Sci.* 115, E5716–E5725. doi: 10.1073/pnas.1719367115

Rosebrock, A. (2021). *Deep Learning for Computer Vision with Python,* 3rd Edn. Available at: https://www.kickstarter.com/projects/adrianrosebrock/deep-learning-for-computer-vision-with-python-eboo/description (accessed March 6, 2022).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). ImageNet large scale visual recognition challenge. *Int. J. Comp. Vis.* 115, 211–252. doi: 10.1007/s11263-015-0816-y

Seeland, M., Rzanny, M., Alaqraa, N., Waldchen, J., and Mader, P. (2017). Plant species classification using flower images-A comparative study of local feature representations. *PLoS One* 12:e0170629. doi: 10.1371/journal.pone.0170629

Shibata, S., Mizuno, R., and Mineno, H. (2020). Semisupervised deep state-space model for plant growth modeling. *Plant Phenom.* 2020:4261965. doi: 10.34133/2020/4261965

Simonyan, K., and Zisserman, A. (2015). "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR*. (Vienna: International Conference on Learning Representations, ICLR). doi: 10.3390/s21082852

Song, X., Gao, K., Fan, G., Zhao, X., Liu, Z., and Dai, S. (2018). Quantitative classification of the morphological traits of ray florets in large-flowered chrysanthemum. *Hortscience* 53, 1258–1265. doi: 10.21273/hortsci13069-18

Song, X., Gao, K., Huang, H., Liu, Z., Dai, S., and Ji, Y. (2021). Quantitative definition and classification of leaves in large- flowered chinese chrysanthemum based on the morphological traits. *Chin. Bull. Bot.* 56, 10–24. doi: 10.11983/cbb20014

Spiesman, B. J., Gratton, C., Hatfield, R. G., Hsu, W. H., Jepsen, S., McCornack, B., et al. (2021). Assessing the potential for deep learning and computer vision to identify bumble bee species from images. *Sci. Rep.* 11:1. doi: 10.1038/s41598-021-87210-1

Su, J., Jiang, J., Zhang, F., Liu, Y., Lian, D., Chen, S., et al. (2019). Current achievements and future prospects in the genetic breeding of chrysanthemum: a review. *Horticult. Res.* 6:8. doi: 10.1038/s41438-019-0193-8

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Boston, MA).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). "Rethinking the Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Las Vegas, NV).

Tng, S. S., Le, N. Q. K., Yeh, H. Y., and Chua, M. C. H. (2022). Improved prediction model of protein lysine crotonylation sites using bidirectional recurrent neural networks. *J. Proteome Res.* 21, 265–273. doi: 10.1021/acs.jproteome.1c00848

Ubbens, J. R., and Stavness, I. (2017). Deep plant phenomics: a deep learning platform for complex plant phenotyping tasks. *Front. Plant Sci.* 8:1190. doi: 10.3389/fpls.2017.01190

Waeldchen, J., and Maeder, P. (2018). Machine learning for image based species identification. *Methods Ecol. Evol.* 9, 2216–2225. doi: 10.1111/2041-210x.13075

Wang, C., Li, X., Caragea, D., Bheemanahallia, R., and Jagadish, S. V. K. (2020). Root anatomy based on root cross-section image analysis with deep learning. *Comp. Elect. Agricult.* 175:5549. doi: 10.1016/j.compag.2020.105549

Wang, J. (1993). Classification of chrysanthemum in China. *Proc. Chin. Chrysanth. Res.* 3, 58–60.

Wldchen, J., and Mder, P. (2018). Plant species identification using computer vision techniques: a systematic literature review. *Arch. Comp. Methods Eng.* 25, 507–543. doi: 10.1007/s11831-016-9206-z

Yu, D. (1963). Problems on the classification and nomination of garden plants. *Acta Horticult.* 2, 225–232. doi: 10.1515/9780824839154-014

Yue, J., Li, Z., Zuo, Z., Zhao, Y., Zhang, J., and Wang, Y. (2021). Study on the identification and evaluation of growth years for Paris polyphylla var. yunnanensis using deep learning combined with 2DCOS. *Spectrochim. Acta Part A: Mol. Biomol. Spectroscop.* 261:120033. doi: 10.1016/j.saa.2021.120033

Zhang, S., and Dai, S. (2013). *Chinese chrysanthemum book*. Beijing: China Forestry Publishing House.

Zhang, S., Huang, W., Huang, Y., and Zhang, C. (2020). Plant species recognition methods using leaf image: overview. *Neurocomputing* 408, 246–272. doi: 10.1016/j.neucom.2019.09.113

Zhang, Y., Dai, S., Hong, Y., and Song, X. (2014a). Application of Genomic SSR Locus polymorphisms on the identification and classification of chrysanthemum cultivars in china. *Plos One* 9:4856. doi: 10.1371/journal.pone.0104856

Zhang, Y., Luo, X., Zhu, J., Wang, C., Hong, Y., Lu, J., et al. (2014b). A classification study for chrysanthemum (Chrysanthemum x grandiflorum Tzvelv.) cultivars based on multivariate statistical analyses. *J. Syst. Evol.* 52, 612–628. doi: 10.1111/jse.12104

Zhang, Y., Peng, J., Yuan, X., Zhang, L., Zhu, D., Hong, P., et al. (2021). MFCIS: an automatic leaf-based identification pipeline for plant cultivars using deep learning and persistent homology. *Hortic. Res.* 8:172. doi: 10.1038/s41438-021-00608-w

**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.

ORIGINAL RESEARCH
published: 24 June 2022
doi: 10.3389/fpls.2022.886804

# Comparison of Remote Sensing Methods for Plant Heights in Agricultural Fields Using Unmanned Aerial Vehicle-Based Structure From Motion

***Ryo Fujiwara, Tomohiro Kikawada, Hisashi Sato† and Yukio Akiyama****

*Hokkaido Agricultural Research Center, National Agriculture and Food Research Organization (NARO), Sapporo, Japan*

OPEN ACCESS

***Edited by:***
*Hanno Scharr,*
*Helmholtz Association of German Research Centres (HZ), Germany*

***Reviewed by:***
*Mario Fuentes Reyes,*
*German Aerospace Center (DLR), Germany*
*Lukas Roth,*
*ETH Zürich, Switzerland*

***Correspondence:***
*Yukio Akiyama*
*akky@affrc.go.jp*

**†*Present address:***
*Hisashi Sato,*
*Kyushu-Okinawa Agricultural Research Center, NARO, Koshi, Japan*

***Specialty section:***
*This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science*

***Received:*** *01 March 2022*
***Accepted:*** *20 May 2022*
***Published:*** *24 June 2022*

***Citation:***
*Fujiwara R, Kikawada T, Sato H and Akiyama Y (2022) Comparison of Remote Sensing Methods for Plant Heights in Agricultural Fields Using Unmanned Aerial Vehicle-Based Structure From Motion.*
*Front. Plant Sci. 13:886804.*
*doi: 10.3389/fpls.2022.886804*

Remote sensing using unmanned aerial vehicles (UAVs) and structure from motion (SfM) is useful for the sustainable and cost-effective management of agricultural fields. Ground control points (GCPs) are typically used for the high-precision monitoring of plant height (PH). Additionally, a secondary UAV flight is necessary when off-season images are processed to obtain the ground altitude (GA). In this study, four variables, namely, camera angles, real-time kinematic (RTK), GCPs, and methods for GA, were compared with the predictive performance of maize PH. Linear regression models for PH prediction were validated using training data from different targets on different flights ("different-targets-and-different-flight" cross-validation). PH prediction using UAV-SfM at a camera angle of –60° with RTK, GCPs, and GA obtained from an off-season flight scored a high coefficient of determination and a low mean absolute error (MAE) for validation data ($R^2_{val}$ = 0.766, MAE = 0.039 m in the vegetative stage; $R^2_{val}$ = 0.803, MAE = 0.063 m in the reproductive stage). The low-cost case (LC) method, conducted at a camera angle of –60° without RTK, GCPs, or an extra off-season flight, achieved comparable predictive performance ($R^2_{val}$ = 0.794, MAE = 0.036 m in the vegetative stage; $R^2_{val}$ = 0.749, MAE = 0.072 m in the reproductive stage), suggesting that this method can achieve low-cost and high-precision PH monitoring.

**Keywords: unmanned aerial vehicle, structure from motion, remote sensing, plant height, 3D structure analysis, maize**

## INTRODUCTION

Remote sensing is a key technology for the sustainable management of agricultural fields. Agricultural management based on remote sensing strengthens food production and reduces natural resource use. Thus, remote sensing technologies have found applications, such as growth monitoring, irrigation management, weed detection, and yield prediction

**Abbreviations:** CHM, crop height model; DSM, digital surface model; DTM, digital terrain model; GA, ground altitude; GCP, ground control point; GNSS, global navigation satellite system; HC, highest-cost case; LC, low-cost case; LiDAR, light detection and ranging; MAE, mean absolute error; MAPE, mean absolute percentage error; PH, plant height; RMSE, root mean squared error; ROI, region of interest; RTK, real-time kinematic; SfM, structure from motion; UAV, unmanned aerial vehicle.

(Sishodia et al., 2020). Furthermore, the applications of remote sensing in agriculture have gained widespread attention in recent years (Weiss et al., 2020).

Unmanned aerial vehicles (UAVs) are commonly used for the remote sensing of agricultural fields owing to their high-resolution imagery and cost-effectiveness. Sensors (e.g., RGB or multispectral cameras, laser scanning devices, etc.) and processing strategies (e.g., vegetation index calculation, machine learning, 3D structure analysis, etc.) have been combined to solve problems in remote sensing applications (Tsouros et al., 2019).

Three-dimensional (3D) structural analysis is useful for determining plant height (PH) and volume, which reflect the growth and biomass of crops (Yao et al., 2019). Strategies for 3D structure analysis include generating 3D models from multiview aerial images of UAVs using structure from motion (SfM) algorithms or obtaining 3D point clouds with light detection and ranging (LiDAR) systems (Paturkar et al., 2021). The SfM approach with UAV imagery (UAV-SfM) can be conducted at a relatively low cost using normal RGB cameras to suit the requirements of agricultural applications. The UAV-SfM approach for monitoring growth or biomass has been applied to various crops, such as wheat (Holman et al., 2016; Madec et al., 2017; Volpato et al., 2021), barley (Bendig et al., 2013, 2014), rice (Jiang et al., 2019; Kawamura et al., 2020; Lu et al., 2022), and maize (Li et al., 2016; Ziliani et al., 2018; Tirado et al., 2020).

However, the UAV-SfM approach has a problem with regard to balancing precision and cost. During the SfM process, matching features over multiple images are detected, camera positions are estimated, and dense point clouds are generated (Westoby et al., 2012). Ground control points (GCPs), which are points whose coordinates are known from ground surveys, are often used to correct the camera positions. The coordinates of each aerial image surveyed with the global navigation satellite system (GNSS) are commonly recorded in the metadata of the image and can be used for the SfM process. SfM analysis without GCPs often faces coordinate errors owing to the uncertainty of GNSS (Wu et al., 2020) or an SfM-specific distortion termed the central “doming” effect (Rosnell and Honkavaara, 2012). However, the installation and maintenance of GCPs on the ground require time and effort. Additionally, annotation of GCPs on aerial images also takes time if weeds or reflection of sunlight interfere with the auto-detection algorithms for GCPs.

An orthomosaic (an orthographic image composed of geometrically corrected aerial images) and a digital surface model (DSM; a representation of elevation on the 3D model) are constructed from the point clouds. When aerial images of an on-season agricultural field are taken and processed, a DSM obtained by the SfM process shows an elevation that includes the PH. To extract PH from the DSM, data for ground altitude (GA), such as that obtained from a digital terrain model (DTM; digital topographic maps indicating ground surface), are necessary.

Ground altitude is mainly obtained by processing off-season (pre-germination or post-harvest) images to create a DTM (Roth and Streit, 2018; Ziliani et al., 2018; Jiang et al., 2019; Kawamura et al., 2020) or by extracting the altitude of the soil surface in the on-season DSM (Tirado et al., 2020). The former method (processing off-season images) requires an extra flight, along with the SfM process. In addition, the models obtained from different flights generally deviate from each other owing to the uncertainty of the coordinates and distortion of the 3D models mentioned above. Such deviations cause errors in PH prediction and affect reproducibility. Therefore, correction with GCPs is crucial for this method. The latter method (extracting soil altitude) requires the presence of a bare soil surface in the seasonal DSM. However, this method may be difficult to apply when the ground is fully covered by plants. To obtain GA in plants, extracted soil coordinates were interpolated to generate a DTM (Murakami et al., 2012; Gillan et al., 2014; Iqbal et al., 2017). Such interpolation methods can achieve low-cost and accurate PH predictions when adequate soil coordinates are extracted for the interpolation algorithm. Although other sources, such as airborne laser scanning (ALS), can be used to determine the altitude of the terrain (Li et al., 2016), the applicable scope is mostly restricted because of the equipment costs and time demands for aerial scanning.

The precision of UAV-SfM analysis is determined by the camera angles, real-time kinematics (RTK), and GCPs. SfM point clouds generated from aerial images taken at diagonal camera angles can have fewer errors that result from the “doming” effect (James and Robson, 2014). Furthermore, UAVs equipped with high-precision positioning systems using RTK-GNSS have become increasingly popular, although the initial and operational costs for RTK-UAVs remain higher. Finally, GCPs are generally used to correct camera positions, as mentioned above. In addition to these three parameters, methods for obtaining GA need to be considered for PH prediction. Although comparative studies have been conducted on one or a few of these variables (Holman et al., 2016; Xie et al., 2021; Lu et al., 2022), their effects on the precision of PH monitoring have not been fully investigated.

In this study, four variables, namely, camera angles, RTK, GCPs, and methods for obtaining GA, were compared for PH prediction in maize. Two existing methods for obtaining GA, using off-season DSMs (method M1) and extracting the altitude of the soil surface (method M2), were demonstrated (**Figure 1**). In addition, an interpolation method for obtaining GA (method M3) was considered, wherein the coordinates of the terrain around the field were obtained and fitted to a polynomial surface. The surface can then be used as a DTM, and the DTM subtracted from a DSM provides a crop height model (CHM), representing the PH of the crop (Chang et al., 2017). This method can achieve high precision without GCPs, even when the inside of the field is covered with plants.

## MATERIALS AND METHODS

### Data Collection and Structure From Motion Process

The data were collected from the Hokkaido Agricultural Research Center (Hokkaido, Japan). Two maize fields (Fields 1 and 2) under variety tests were used in this study. An overview of these two fields is presented in **Figure 2**. Field 1 was used for method M3, and Field 2 was used for validation of PH prediction under all conditions of camera angles, RTK, GCPs, and methods for

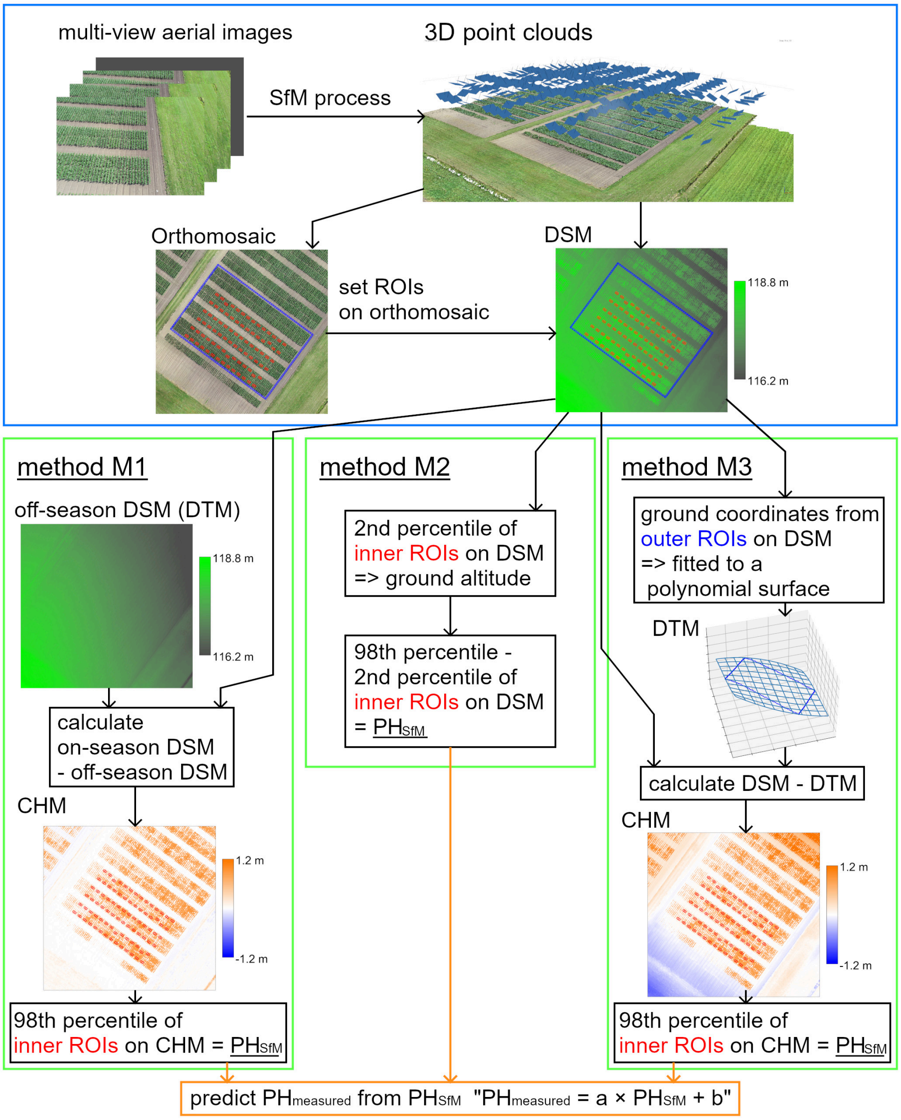

multi-view aerial images
SfM process
3D point clouds
Orthomosaic
DSM
set ROIs on orthomosaic
118.8 m
116.2 m
method M1
off-season DSM (DTM)
118.8 m
116.2 m
calculate on-season DSM - off-season DSM
CHM
1.2 m
-1.2 m
98th percentile of inner ROIs on CHM = PHSfM
method M2
2nd percentile of inner ROIs on DSM => ground altitude
98th percentile - 2nd percentile of inner ROIs on DSM = PHSfM
method M3
ground coordinates from outer ROIs on DSM => fitted to a polynomial surface
DTM
calculate DSM - DTM
CHM
1.2 m
-1.2 m
98th percentile of inner ROIs on CHM = PHSfM
predict PHmeasured from PHSfM "PHmeasured = a × PHSfM + b"

**FIGURE 1 |** An overview of the SfM process and three methods for obtaining ground altitude (GA); methods M1, M2, and M3. A DSM is a digital surface model that represents elevation on the 3D model, a DTM is a digital terrain model that represents elevation without plants, and a CHM is a crop height model that represents plant height of crop.

obtaining GA. There were 42 plots (14 varieties) in Field 1 and 84 plots (21 varieties) in Field 2. Each plot had four rows, and each row contained 18 plants. The row spacing was 0.75 m and the in-row plant spacing was 0.18 m (7.41 plants per square meter) in both fields.

Nine checkerboard square markers used as GCPs for the SfM process were placed in Field 2. Eight GCPs were located around the field and one GCP was located at the center of the field. The locations of the GCPs are shown in **Figure 2**. The coordinates of the GCPs were surveyed using D-RTK 2 (SZ DJI Technology, Nanshan, Shenzhen, China).

The ground truth of PH was measured using rulers as the height from the ground to the highest point of the plant at two different growth stages. During the vegetative stage, the highest point was at the apex of the top leaf, while during the reproductive stage, the highest point was at the apex of the tassel. Actual measurements were conducted in the middle two of the four rows in each plot. The PH of five consecutively placed plants in each row was measured and averaged. The average from one row was regarded as one sample ($PH_{\text{measured}}$). Subsequently, $PH_{\text{measured}}$ was obtained from 84 rows in Field 1 and 168 rows in Field 2. The schedule of the actual PH measurements and UAV image acquisition is presented in **Table 1**.

A DJI Phantom 4 RTK (SZ DJI Technology) with a mounted camera (lens: 8.8 mm focal length, sensor: 1” CMOS 20 M) was used for image acquisition. The analysis region (35.5 m × 54 m) for the SfM process in each field was determined, as shown in **Figure 2**. The flight plan was generated automatically using

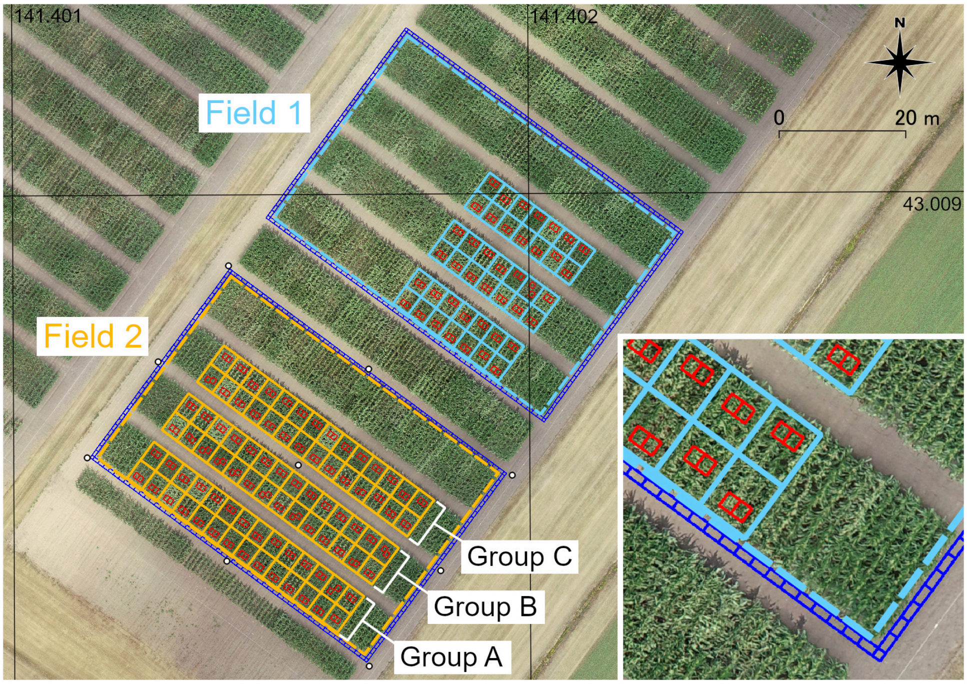


**FIGURE 2 |** An overview of two fields of maize used in this study. Light blue (Field 1) and orange (Field 2) solid lines show the plots. Dash lines show the analysis region of each field. White circles around or in Field 2 show the locations of GCPs. Red rectangles show ROIs on rows (inner ROIs). Blue rectangles show ROIs around the field (outer ROIs).

**TABLE 1 |** The schedule of data collection.

| Stage | Process | Field 1 | | Field 2 | |
|---|---|---|---|---|---|
| | Sowing | 2021/5/12 | | 2021/5/12 | |
| Pre-germination | Image acquisition | – | | 2021/5/14 | 4 flights<br>(4 conds. × 1 rep.)* |
| | | | | 2021/5/17 | 8 flights<br>(4 conds. × 2 reps.)* |
| Vegetative stage | Image acquisition | 2021/6/28 | 3 flights<br>(3 reps.) | 2021/6/30 | 12 flights<br>(4 conds. × 3 reps.) |
| | Measurement | 2021/6/28 | | 2021/6/30 | |
| Reproductive stage | Image acquisition | 2021/8/4 | 3 flights<br>(3 reps.) | 2021/9/2 | 12 flights<br>(4 conds. × 3 reps.) |
| | Measurement | 2021/8/6 | | 2021/9/3 | |

**Three repetitions of image acquisition in the pre-germination stage (Field 2) were conducted over 2 days: one rep on 2021/5/14, and the remaining two reps on 2021/5/17. The details of UAV image acquisition are shown in* ***Supplementary Table 1*** *(Field 1) and* ***Supplementary Table 2*** *(Field 2).*

DJI GS RTK (SZ DJI Technology) to cover the analysis region of each field with an adequate margin. The flight height was 25 m, the forward overlapping rate was 80%, and the side-overlapping rate was 60%. For the flight in Field 1, the camera angle (angles of a camera's forward direction from a horizontal plane) was −90°, and RTK was not used. For the flight in Field 2, the camera angle was set to −60° or −90° as shown in **Figure 3**, and RTK was switched on or off. Therefore, four flight conditions were applied. The details of UAV image acquisition are shown in **Supplementary Table 1** (Field 1) and **Supplementary Table 2** (Field 2).

In Field 1, three flight repetitions were conducted at each growth stage. For Field 2, image acquisition was conducted in three stages, namely, the pre-germination stage (for obtaining off-season data of method M1), the vegetative stage, and the reproductive stage. At each stage, 12 flights (four conditions with three flight repetitions) were arranged using a randomized block design.

The SfM process was conducted using the Agisoft Metashape Professional 1.7.3 (Agisoft LLC, St. Petersburg, Russia). Three-dimensional point clouds were generated from the image sets, and orthomosaic images and DSMs were constructed. The

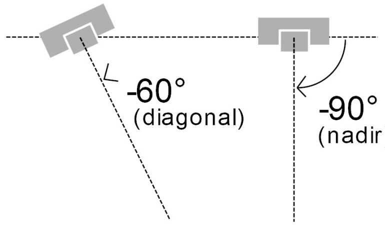


**FIGURE 3 |** An illustration of the camera angle.

**TABLE 2 |** Parameters of the SfM process.

| Process | Parameter | Setting |
|---|---|---|
| Aligning photos | Accuracy | High |
| | Generic preselection | Yes |
| | Key point limit | 40,000 |
| | Tie point limit | 4,000 |
| Building dense point cloud | Quality | High |
| | Depth filtering modes | Mild |
| Building digital elevation model | Source data | Dense cloud |
| | Interpolation | Enabled |
| Building orthomosaic | Surface | Digital elevation model |
| | Blending mode | Mosaic |
| | Hole filling | Enabled |

parameters selected for the process are listed in **Table 2**. From each image set of Field 2, another Metashape project file was created for the GCP-corrected analysis. In the project file, markers were set at the locations of nine GCPs, the coordinates of GCPs by the ground survey were input, and the SfM process was conducted in a similar manner. The conditions for the SfM products are summarized in **Supplementary Table 3**.

## Methods for Analysis of Digital Surface Model

Each analysis row in each field was divided into three blocks. Using QGIS Desktop 3.16.8, one polygon enclosing each field (Field 1 or 2) and polygon enclosing blocks were created on each orthomosaic image and written to a shapefile (the locations of polygons are shown in **Supplementary Figure 1**). Regions of interest (ROIs) in rows (inner ROIs, for all methods) and around the field (outer ROIs, for method M3) were determined using the shapefile. For the inner ROIs (red rectangles in **Figure 2**), the coordinates were calculated with the locations of the plants measured in the blocks. For outer ROIs (blue rectangles), coordinates of 180 rectangular areas with a size of approximately 1 × 0.5 m (on the corner: 0.5 × 0.5 m) enclosing each field were calculated. The coordinates were saved as CSV files. In **Figure 2**, the locations of the ROIs are drawn on an orthomosaic according to the coordinates used for visualization.

For method M1 (only Field 2), an off-season DSM was used as the DTM, which was subtracted from the on-season DSM. For each on-season DSM, an off-season DSM under the same conditions and repetition was applied; thus, the same number of CHMs (24 CHMs for Field 2) were obtained.

The 90th–99th percentiles have often been applied as representative values of PH (Holman et al., 2016; Malambo et al., 2018; Tirado et al., 2020), as they express height (altitude of the highest point) better than a mean and are subjected to less noise than a maximum. In this study, the size of an inner ROI was approximately 2,500 pixels, and noises that were blobs of adjacent 30 pixels or smaller were observed on a DSM (**Figure 4**). Therefore, to exclude noise less than 2% of the ROI (50 pixels), the 98th percentile was used in this study. For the CHMs, the 98th percentile from the inner ROIs was calculated as $PH_{\mathrm{SfM}}$.

For method M2 (only Field 2), the 2nd and 98th percentiles from the inner ROIs were calculated as the altitude of the ground and plant apex, respectively. These percentiles were applied for the same reason mentioned above. The difference between the 2nd and 98th percentiles was obtained as $PH_{\mathrm{SfM}}$.

For method M3 (Fields 1 and 2), a DTM was obtained by polynomial fitting with the coordinates of the terrain around the field on an on-season DSM. The median altitude was calculated from each outer ROI as the $z$-coordinate of the area, and the center of gravity of the rectangle was calculated as the $x$ and $y$-coordinates. A total of 180 points ($x$, $y$, $z$) were fitted to an $n$-dimensional polynomial surface (1) using the least squares method.

$$z = \sum_{k=0}^{n} \sum_{i=0}^{k} a_{ki} x^{k-i} y^{i} \tag{1}$$

where $n$ is the dimension of the polynomial surface, $k$ and $i$ are the indices for summation, and $a_{ki}$ is the parameter to be estimated. This polynomial surface was used as the DTM. This DTM was subtracted from the original DSM to obtain a CHM. On the CHMs, the 98th percentiles from the inner ROIs were calculated as $PH_{\mathrm{SfM}}$ (PH obtained from UAV-SfM analysis).

The dimension of the polynomial ($n$) was set to 0–4 for data from Field 1. The dimension that achieved the strongest correlation between the measured PH ($PH_{\mathrm{measured}}$) and $PH_{\mathrm{SfM}}$ in Field 1 was applied for a comparative study of Field 2.

The analysis in this and the following sections were conducted using Python 3.6.8 and QGIS.

## Correlation Analysis and Cross-Validation of Regression Models

The Pearson correlation coefficient ($r$) between $PH_{\mathrm{measured}}$ and $PH_{\mathrm{SfM}}$ in each dataset and the bias ( $PH_{\mathrm{SfM}} - PH_{\mathrm{measured}}$) were calculated. For each condition, the three coefficients obtained from the repetitions were averaged.

Cross-validation of the linear regression models to predict $PH_{\mathrm{measured}}$ from $PH_{\mathrm{SfM}}$ was conducted using data from Field 2. As the SfM point clouds from different flights could deviate from each other, validation with different target data on a different flight's point cloud was needed ("different-targets-and-different-flight" validation) to ensure that a regression model from one flight can be applied to as training data to unknown data. The 168 data points from Field 2 were divided into three groups (the grouping in Field 2 is shown in **Figure 2**). A linear regression

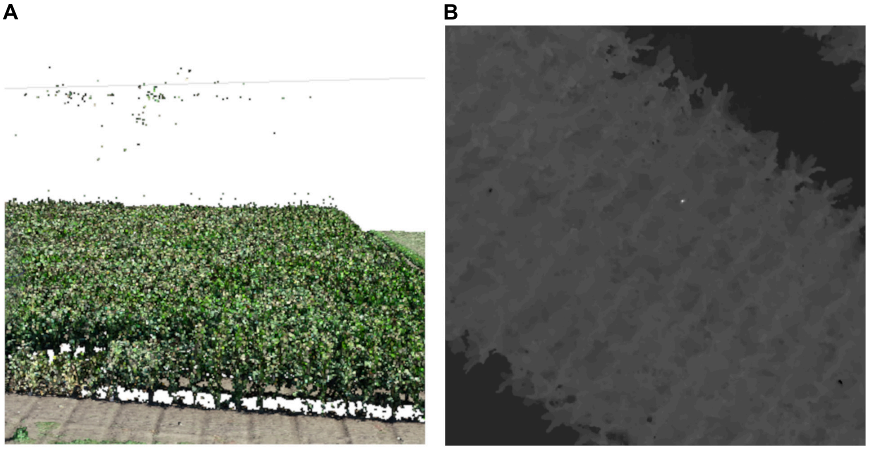


**FIGURE 4 |** An example of noises in point clouds. **(A)** A point cloud with noises (some points floating over plants). **(B)** A DSM with noises (an extremely high area near the center). The DSM is shown in grayscale; when the pixel is white, the altitude is high.

model (2) was fitted with the least-squares method using data from two groups (112 training data points).

$$PH_{measured} = a \times PH_{\mathrm{SfM}} + b \tag{2}$$

where $a$ and $b$ are the parameters to be estimated ($a$: slope and $b$: intercept, respectively). A total of 56 data points from different flight repetition groups were used for validation. There were six flight repetitions and three groups for training and validation; thus, 18 sets of validations were conducted for each condition (all sets of flight repetitions and sample groups for cross-validation are shown in **Supplementary Table 4**). The coefficient of determination ($R^2$), mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE) were calculated using the validation data (equations of these evaluation metrics are shown in **Supplementary Table 5**).

# RESULTS

## Consideration of Method M3

A summary of the measured PH ($PH_{\mathrm{measured}}$) is provided in **Table 3**. The range of the measured PH was 0.632–1.190 m in the vegetative stage and 2.18–3.14 m in the reproductive stage. The standard deviations in Fields 1 and 2 were 0.075 and 0.102 m in the vegetative stage and 0.126 and 0.181 m in the reproductive stage, respectively. These results indicate high variability in PH in the fields.

The DSM of Field 1 shows ground inclination, with the northeast being lower and the southwest being higher (**Figure 5**). The CHM calculated from the 0 dim (flat) DTM left the inclination, as the 0 dim DTM cannot model such a tilted plane. The CHM calculated from the 1 dim (plane) DTM did not leave the inclination but left the central bulge. The 2 and 3 dim DTMs fitted better to the true terrain. The CHMs from the 2 and 3 dim DTMs left neither the inclination nor bulge inside the ground ROIs. The 4 dim DTM, however, overfitted the sample points of the ground, and thus, the CHM from the DTM was strongly distorted.

The mean correlation coefficients ($r$) between $PH_{\mathrm{measured}}$ and $PH_{\mathrm{SfM}}$ on the CHMs of Field 1 from the three flight repetitions are summarized in **Table 4**. The correlation was stronger with the 3 dim DTM in both growth stages. Thus, the 3 dim DTM was applied to method M3 in validation with Field 2.

## Comparative Correlation Analysis Between $PH_{\mathrm{measured}}$ and $PH_{\mathrm{SfM}}$ on All Conditions

$PH_{\mathrm{SfM}}$ of Field 2 was calculated for 24 conditions that differed in the four parameters, namely, camera angles, RTK, GCPs, and methods for obtaining GA (method M1: using off-season DSM, M2: extracting altitude of the soil surface, and M3: fitting coordinates of the terrain around the field to a polynomial surface). The correlation coefficients ($r$) between $PH_{\mathrm{measured}}$ and $PH_{\mathrm{SfM}}$ were compared at each growth stage (vegetative stage, **Table 5**; reproductive stage, **Table 6**).

In the vegetative stage (**Table 5**), the correlation was stronger when a camera angle of −60° (diagonal) and method M3 were applied, even without RTK or GCPs. Using method M1, the correlation was stronger when RTK or GCPs were present. With

**TABLE 3 |** Descriptive statistics of measured PH.

| | Vegetative stage | | Reproductive stage | |
|---|---|---|---|---|
| | Field 1 | Field 2 | Field 1 | Field 2 |
| Date of measurement | 28-Jun | 30-Jun | 6-Aug | 3-Sep |
| Number of samples | 84 | 168 | 84 | 168 |
| Mean (m) | 0.809 | 0.898 | 2.73 | 2.68 |
| Minimum (m) | 0.634 | 0.632 | 2.47 | 2.18 |
| Maximum (m) | 0.982 | 1.190 | 3.05 | 3.14 |
| Standard deviation (m) | 0.075 | 0.102 | 0.126 | 0.181 |

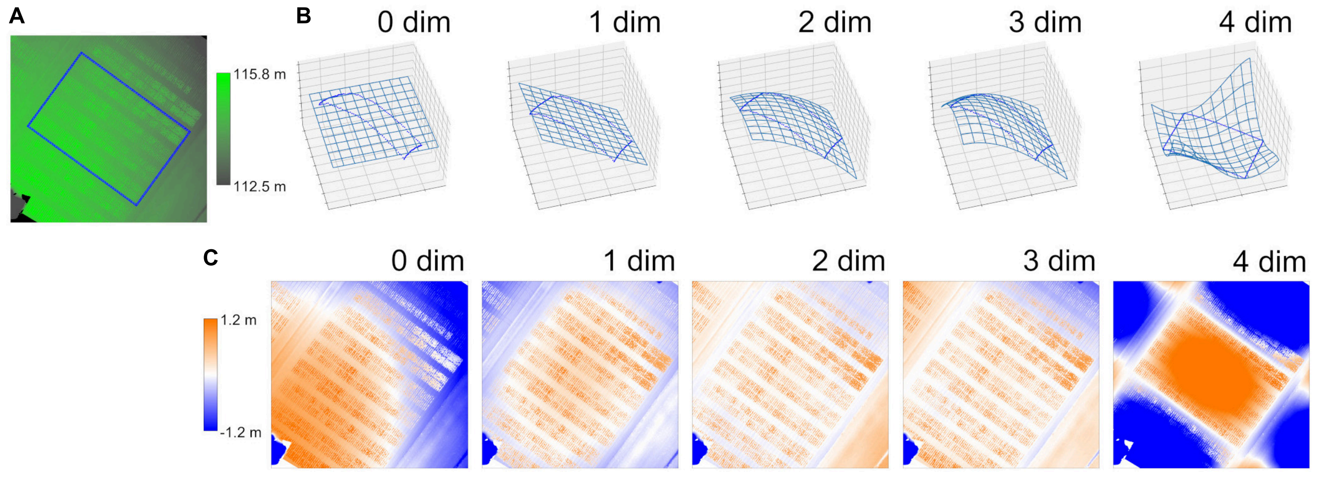


**FIGURE 5 |** An example of the process involved in method M3. **(A)** A digital surface model (DSM) of Field 1. Blue rectangles show ROIs around the field (outer ROIs). **(B)** Digital terrain models (DTMs) fitted to polynomial surfaces. Blue points show coordinates of the outer ROIs and meshes show the fitted DTMs. "*n* dim" shows the dimension of the polynomial surface. **(C)** Crop height models (CHMs) calculated as the difference between DSM and DTMs. These figures were created with the dataset of rep. 1 on Field 1 in the vegetative stage.

method M2, although the correlation was strong at a −90° (nadir) camera angle, it was weak at −60°. The bias ($PH_{\mathrm{SfM}} - PH_{\mathrm{measured}}$) was negative (−0.07 to −0.09 m) for the highest four conditions in correlation coefficients. $PH_{\mathrm{SfM}}$ tended to be lower than $PH_{\mathrm{measured}}$.

In the reproductive stage (**Table 6**), for both methods M1 and M3, the correlation was strong with −60° camera angle and RTK. When RTK and GCPs were not applied, the correlation was stronger using method M3. In the M2 method, the correlation was weak. $PH_{\mathrm{SfM}}$ tended to be lower than $PH_{\mathrm{measured}}$ during the reproductive stage (bias = approx. −0.2 to −0.3 m under higher correlation coefficient conditions).

## Cross-Validation of Plant Height Regression Models

A linear regression model was necessary for PH prediction with UAV-SfM because $PH_{\mathrm{SfM}}$ tended to be lower than $PH_{\mathrm{measured}}$. Simple regression models for PH prediction were trained and the "different-targets-and-different-flight" cross-validation was conducted under all conditions (vegetative stage: **Table 7**, reproductive stage: **Table 8**). In the reproductive stage, method M2 was omitted because the correlation between $PH_{\mathrm{measured}}$ and $PH_{\mathrm{SfM}}$ was weak (**Table 6**).

**TABLE 4 |** Correlation coefficients (*r*) between $PH_{\mathrm{measured}}$ and $PH_{\mathrm{SfM}}$ on the CSMs of Field 1.

| Dimension of polynomial surface | 0 dim | 1 dim | 2 dim | 3 dim | 4 dim |
|---|---|---|---|---|---|
| Vegetative stage | 0.469 | 0.776 | 0.832 | **0.839** | 0.153 |
| Reproductive stage | 0.346 | 0.816 | 0.866 | **0.873** | 0.272 |

*Each value is the mean of 3 flight repetitions. 3 dim (bolded) was highest in the correlation.*

In the vegetative stage (**Table 7**), the coefficient of determination of the validation data ($R^2_{\mathrm{val}}$) was high when the −60° camera angle and method M3 were applied, as was the correlation coefficient between $PH_{\mathrm{measured}}$ and $PH_{\mathrm{SfM}}$ (**Table 5**). Even in the "Low-cost case (LC)" (camera angle: −60°, RTK: unused [−], GCPs: unused [−], method: M3), which can be conducted with minimum equipment and without an extra flight, the regression model showed a high predictive performance ($R^2_{\mathrm{val}}$ = 0.794, MAE = 0.036 m). In the "Highest-cost case (HC)" (camera angle: −60°, RTK: used [+], GCPs: used [+], method: M1), which seems to achieve high-precision sensing with method M1, the $R^2_{\mathrm{val}}$ was 0.766, which was lower than that of LC. With method M1, $R^2_{\mathrm{val}}$ was high only when GCPs were used.

The predictive performance of HC was highest in the reproductive stage (**Table 8**) ($R^2_{\mathrm{val}}$ = 0.803, MAE = 0.063 m). $R^2_{\mathrm{val}}$ was also high when the camera angle was −60° and method M3 was applied, including LC ($R^2_{\mathrm{val}}$ = 0.749, MAE = 0.072 m). With method M1, the predictive performance was low when GCPs were not used, similar to the vegetative stage. Although the overall mean absolute errors in the validation data (MAEs) in the reproductive stage were larger than those in the vegetative stage, the mean absolute percentage errors (MAPEs) in the reproductive stage were smaller (MAPE = 2–3% in the six highest conditions in $R^2_{\mathrm{val}}$).

Examples of cross-validation, that is, scatterplots between the measured PH ($PH_{\mathrm{measured}}$) and PH predicted by the regression model from $PH_{\mathrm{SfM}}$ on the validation data, are shown in **Figure 6**. From 18 validation cases for each condition, the average case (nearest to the mean in $R^2_{\mathrm{val}}$) was selected for the figure.

# DISCUSSION

The correlation between $PH_{\mathrm{measured}}$ and $PH_{\mathrm{SfM}}$ was stronger with a −60° camera angle than with −90°, except for method M2

**TABLE 5 |** Correlation analysis between $PH_{\text{measured}}$ and $PH_{\text{SfM}}$ in the vegetative stage of Field 2.

| | Camera angle | RTK | GCP | Method* | $r^{\dagger}$ | Bias (m)$^{\dagger}$ |
|---|---|---|---|---|---|---|
| 1 | −60° | + | + | M3 | **0.914** | −0.078 |
| 2 | −60° | + | − | M3 | **0.913** | −0.080 |
| 3 | −60° | − | + | M3 | **0.906** | −0.087 |
| 4 | −60° | − | − | M3 | **0.906** | −0.087 |
| 5 | −60° | + | + | M1 | **0.903** | −0.103 |
| 6 | −60° | + | − | M1 | **0.903** | −0.135 |
| 7 | −60° | − | + | M1 | **0.896** | −0.115 |
| 8 | −90° | + | + | M3 | **0.894** | −0.055 |
| 9 | −90° | + | − | M3 | **0.891** | −0.070 |
| 10 | −90° | − | + | M3 | **0.888** | −0.058 |
| 11 | −90° | + | + | M2 | **0.886** | −0.035 |
| 12 | −90° | − | − | M3 | **0.885** | −0.077 |
| 13 | −90° | + | + | M1 | **0.885** | −0.089 |
| 14 | −90° | + | − | M1 | **0.874** | 0.842 |
| 15 | −90° | − | + | M2 | **0.874** | −0.040 |
| 16 | −90° | − | + | M1 | **0.874** | −0.091 |
| 17 | −90° | − | − | M2 | **0.868** | −0.058 |
| 18 | −60° | − | − | M1 | **0.856** | −1.023 |
| 19 | −90° | + | − | M2 | **0.850** | −0.052 |
| 20 | −90° | − | − | M1 | **0.810** | 0.334 |
| 21 | −60° | − | − | M2 | **0.539** | −0.096 |
| 22 | −60° | − | + | M2 | **0.494** | −0.095 |
| 23 | −60° | + | − | M2 | **0.333** | −0.111 |
| 24 | −60° | + | + | M2 | **0.330** | −0.106 |

**Method for obtaining ground altitude (GA); M1, using off-season DSM; M2, extracting altitude of the soil surface; M3: fitting coordinates of the terrain around the field to a polynomial surface.*

*$^{\dagger}$r is the correlation coefficient between $PH_{measured}$ and $PH_{SfM}$, and bias is the mean of the difference between $PH_{measured}$ and $PH_{SfM}$ ($PH_{SfM}$ − $PH_{measured}$). Each value of r and bias is the mean of three flight repetitions.*

*The rows are sorted by r (bolded) in a descending order.*

**TABLE 6 |** Correlation analysis between $PH_{\text{measured}}$ and $PH_{\text{SfM}}$ in the reproductive stage of Field 2.

| | Camera angle | RTK | GCP | Method* | $r^{\dagger}$ | Bias (m)$^{\dagger}$ |
|---|---|---|---|---|---|---|
| 1 | −60° | + | + | M1 | **0.907** | −0.262 |
| 2 | −60° | + | − | M1 | **0.906** | −0.306 |
| 3 | −60° | + | + | M3 | **0.899** | −0.227 |
| 4 | −60° | + | − | M3 | **0.899** | −0.233 |
| 5 | −60° | − | + | M1 | **0.894** | −0.267 |
| 6 | −60° | − | + | M3 | **0.886** | −0.232 |
| 7 | −60° | − | − | M3 | **0.883** | −0.236 |
| 8 | −90° | + | − | M1 | **0.869** | 0.790 |
| 9 | −90° | + | + | M1 | **0.863** | −0.210 |
| 10 | −60° | − | − | M1 | **0.862** | −1.181 |
| 11 | −90° | + | − | M3 | **0.846** | −0.212 |
| 12 | −90° | + | + | M3 | **0.845** | −0.166 |
| 13 | −90° | − | + | M1 | **0.843** | −0.208 |
| 14 | −90° | − | + | M3 | **0.829** | −0.167 |
| 15 | −90° | − | − | M3 | **0.824** | −0.213 |
| 16 | −90° | − | − | M1 | **0.809** | 0.045 |
| 17 | −90° | + | − | M2 | **0.327** | −1.745 |
| 18 | −90° | + | + | M2 | **0.316** | −1.736 |
| 19 | −60° | + | − | M2 | **0.281** | −1.821 |
| 20 | −60° | − | − | M2 | **0.259** | −1.836 |
| 21 | −60° | + | + | M2 | **0.257** | −1.790 |
| 22 | −90° | − | − | M2 | **0.243** | −1.783 |
| 23 | −90° | − | + | M2 | **0.229** | −1.769 |
| 24 | −60° | − | + | M2 | **0.208** | −1.828 |

**Method for obtaining ground altitude (GA); M1, using off-season DSM; M2, extracting altitude of the soil surface; M3, fitting coordinates of the terrain around the field to a polynomial surface.*

*$^{\dagger}$r is the correlation coefficient between $PH_{measured}$ and $PH_{SfM}$, and bias is the mean of the difference between $PH_{measured}$ and $PH_{SfM}$ ($PH_{SfM}$ − $PH_{measured}$). Each value of r and bias is the mean of three flight repetitions.*

*The rows are sorted by r (bolded) in a descending order.*

(**Tables 5**, **6**). This tendency for a strong correlation of −60° was observed even when GCPs were used. Therefore, the diagonal camera angle could both suppress the "doming" effect and grasp the 3D structures of the plants well with a lateral view. However, in the vegetative stage using method M2, the soil surface in some plots was difficult to image at a diagonal camera angle, and the low accuracy of the GA appeared to result in a weak correlation. In some examples, a DSM with a −60° camera angle had wider plant areas than a DSM with a −90° camera angle and a hidden soil surface (**Figure 7**). In the reproductive stage, the inner ROI was mostly covered with plants, and thus, M2 was difficult to apply with both −60° and −90° camera angles.

The $R^2_{\text{val}}$ ranks of the regression models differed from those of the correlation coefficients ($r$) between $PH_{\text{measured}}$ and $PH_{\text{SfM}}$ (**Tables 5–8**). For example, in the vegetative stage, the condition with camera angle: −60°, RTK: used [+], GCPs: unused [−], and method: M1 scored high correlation coefficients ($r$ = 0.903, rank = 6; **Table 5**) and thus high goodness of fit for the training data ($R^2_{\text{train}} =$ 0.814; **Table 7**). However, the regression models had low predictive performance on "different-targets-and-different-flight" validation data ($R^2_{\text{val}} =$ 0.401, rank = 17; **Table 7**). Although a strong correlation in one flight leads to high goodness of fit for the training data, the model showed low predictive performance on unknown data from different flights.

The predictive performance on unknown data was higher with the M1 and GCP methods or with method M3 (**Tables 7**, **8**). With method M1, GCPs seemed to prevent the deviation between 3D models from different flights and contributed to the high predictive performance. For method M3, the predictive performance was not affected by such deviation, even without GCPs. The ground surface was determined for each on-season flight by using method M3. In this process, the effect of the overall deviation was reduced.

The contribution of RTK positioning to the precision of PH monitoring was restricted in this study, although only small effects were observed. With RTK used [+], GCPs unused [−], and method M1 applied, the $R^2_{\text{val}}$ was lower despite some improvement by a diagonal (−60°) camera angle (**Tables 7**, **8**). Although RTK positioning installed on UAVs enables centimeter-level precision on a DSM (Forlani et al., 2018), the differences in PH were also at the centimeter level. Moreover, SfM photogrammetry based on RTK positioning has larger vertical errors than horizontal errors (Forlani et al., 2018; Štroner et al., 2020). A previous

**TABLE 7 |** Evaluation metrics on cross-validation of PH regression models in the vegetative stage of Field 2.

| | Camera angle | RTK | GCP | method* | | $R^2_{\mathrm{train}}$† | $R^2_{\mathrm{val}}$† | MAE (m)† | RMSE (m)† | MAPE† |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | −60° | − | + | M3 | | 0.820 | **0.799** | 0.036 | 0.044 | 4.08 |
| 2 | −60° | − | − | M3 | (LC)‡ | 0.820 | **0.794** | 0.036 | 0.045 | 4.14 |
| 3 | −60° | − | + | M1 | | 0.801 | **0.786** | 0.037 | 0.046 | 4.17 |
| 4 | −60° | + | + | M3 | | 0.835 | **0.770** | 0.038 | 0.047 | 4.36 |
| 5 | −60° | + | − | M3 | | 0.832 | **0.769** | 0.038 | 0.047 | 4.34 |
| 6 | −60° | + | + | M1 | (HC)‡ | 0.815 | **0.766** | 0.039 | 0.048 | 4.36 |
| 7 | −90° | − | + | M3 | | 0.786 | **0.758** | 0.038 | 0.049 | 4.38 |
| 8 | −90° | − | − | M3 | | 0.782 | **0.756** | 0.039 | 0.049 | 4.43 |
| 9 | −90° | + | − | M3 | | 0.793 | **0.745** | 0.039 | 0.049 | 4.42 |
| 10 | −90° | + | + | M3 | | 0.799 | **0.737** | 0.040 | 0.050 | 4.56 |
| 11 | −90° | + | + | M2 | | 0.783 | **0.729** | 0.041 | 0.051 | 4.63 |
| 12 | −90° | + | + | M1 | | 0.781 | **0.728** | 0.041 | 0.051 | 4.61 |
| 13 | −90° | − | + | M2 | | 0.762 | **0.722** | 0.041 | 0.052 | 4.61 |
| 14 | −90° | − | + | M1 | | 0.762 | **0.720** | 0.042 | 0.052 | 4.69 |
| 15 | −90° | − | − | M2 | | 0.753 | **0.703** | 0.042 | 0.053 | 4.74 |
| 16 | −90° | + | − | M2 | | 0.727 | **0.669** | 0.043 | 0.057 | 4.83 |
| 17 | −60° | + | − | M1 | | 0.814 | **0.401** | 0.063 | 0.073 | 7.14 |
| 18 | −60° | − | + | M2 | | 0.253 | **0.200** | 0.065 | 0.090 | 7.32 |
| 19 | −60° | − | − | M2 | | 0.317 | **0.167** | 0.063 | 0.091 | 7.04 |
| 20 | −60° | + | − | M2 | | 0.146 | **−0.045** | 0.077 | 0.102 | 8.65 |
| 21 | −60° | + | + | M2 | | 0.142 | **−0.066** | 0.077 | 0.103 | 8.71 |
| 22 | −60° | − | − | M1 | | 0.731 | **−40.07** | 0.582 | 0.585 | 65.78 |
| 23 | −90° | + | − | M1 | | 0.764 | **−74.45** | 0.774 | 0.776 | 87.23 |
| 24 | −90° | − | − | M1 | | 0.658 | **−99.50** | 0.791 | 0.796 | 88.84 |

**Method for obtaining ground altitude (GA); M1, using off-season DSM; M2, extracting altitude of the soil surface; M3, fitting coordinates of the terrain around the field to a polynomial surface.*

*†$R^2_{train}$ and $R^2_{val}$ are the coefficients of determination for the training and validation datasets, respectively. MAE is the mean absolute error, RMSE is the root mean squared error, and MAPE is the mean absolute percentage error of the validation data. Each value represents the mean of the 18 validation cases.*

*‡LC means "Low-cost case" (camera angle: −60°, RTK: unused [−], GCPs: unused [−], method: M3), and HC means "Highest-cost case" (camera angle: −60°, RTK: used [+], GCPs: used [+], method: M1).*

*The rows are sorted by $R^2_{val}$ (bolded) in a descending order.*

**TABLE 8 |** Evaluation metrics on cross-validation of PH regression models in the reproductive stage of Field 2.

| | Camera angle | RTK | GCP | Method* | | $R^2_{\mathrm{train}}$† | $R^2_{\mathrm{val}}$† | MAE (m)† | RMSE (m)† | MAPE† |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | −60° | + | + | M1 | (HC)‡ | 0.821 | **0.803** | 0.063 | 0.078 | 2.36 |
| 2 | −60° | + | + | M3 | | 0.810 | **0.791** | 0.066 | 0.081 | 2.47 |
| 3 | −60° | + | − | M3 | | 0.808 | **0.781** | 0.067 | 0.083 | 2.52 |
| 4 | −60° | − | + | M1 | | 0.799 | **0.771** | 0.068 | 0.085 | 2.55 |
| 5 | −60° | − | + | M3 | | 0.785 | **0.753** | 0.071 | 0.089 | 2.67 |
| 6 | −60° | − | − | M3 | (LC)‡ | 0.782 | **0.749** | 0.072 | 0.089 | 2.70 |
| 7 | −90° | + | + | M1 | | 0.745 | **0.640** | 0.086 | 0.105 | 3.23 |
| 8 | −90° | + | + | M3 | | 0.718 | **0.601** | 0.091 | 0.111 | 3.40 |
| 9 | −90° | − | + | M1 | | 0.712 | **0.597** | 0.089 | 0.112 | 3.32 |
| 10 | −90° | + | − | M3 | | 0.721 | **0.592** | 0.093 | 0.113 | 3.47 |
| 11 | −90° | − | − | M3 | | 0.680 | **0.576** | 0.093 | 0.115 | 3.48 |
| 12 | −90° | − | + | M3 | | 0.691 | **0.562** | 0.093 | 0.116 | 3.48 |
| 13 | −60° | + | − | M1 | | 0.821 | **0.548** | 0.097 | 0.115 | 3.64 |
| 14 | −60° | − | − | M1 | | 0.746 | **−14.240** | 0.637 | 0.644 | 23.83 |
| 15 | −90° | + | − | M1 | | 0.756 | **−15.510** | 0.608 | 0.619 | 22.74 |
| 16 | −90° | − | − | M1 | | 0.653 | **−49.841** | 1.115 | 1.122 | 41.62 |

**Method for obtaining ground altitude (GA); M1, using off-season DSM; M3, fitting coordinates of the terrain around the field to a polynomial surface.*

*†$R^2_{train}$ and $R^2_{val}$ are the coefficients of determination for the training and validation datasets, respectively. MAE is the mean absolute error, RMSE is the root mean squared error, and MAPE is the mean absolute percentage error of the validation data. Each value represents the mean of the 18 validation cases.*

*‡LC means "Low-cost case" (camera angle: −60°, RTK: unused [−], GCPs: unused [−], method: M3), and HC means "Highest-cost case" (camera angle: −60°, RTK: used [+], GCPs: used [+], method: M1).*

*The rows are sorted by $R^2_{val}$ (bolded) in a descending order.*

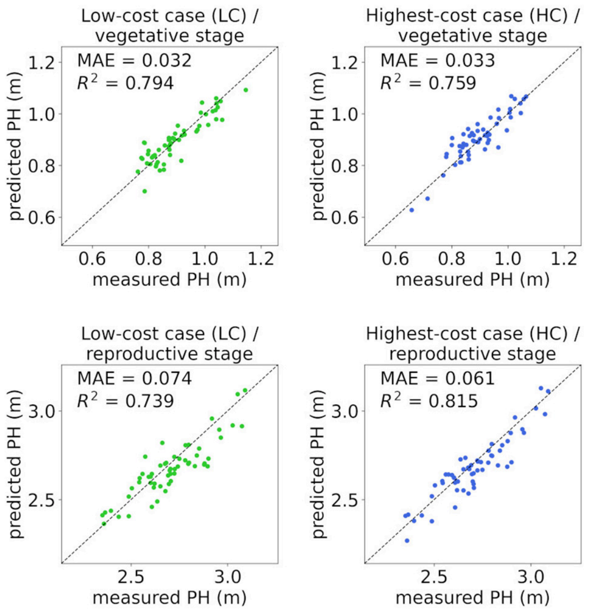


**FIGURE 6 |** Scatterplots between measured PH ($PH_{\text{measured}}$) and PH predicted by the linear regression model from $PH_{\text{SfM}}$ on validation data. "Low-cost case (LC)" is the condition with camera angle: −60°, RTK: unused [–], GCPs: unused [–], method: M3; and "Highest-cost case (HC)" is the condition with camera angle: −60°, RTK: used [+], GCPs: used [+], method: M1. In each set of analysis conditions (LC or HC) and stage (vegetative or reproductive), the nearest to the mean in $R^2_{\text{val}}$ was selected from 18 validation cases.

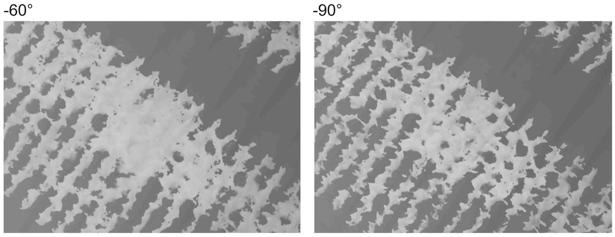


**FIGURE 7 |** A comparison of DSMs with −60° and −90° for the camera angle. The DSMs are shown in grayscale; when the pixel is white, the altitude is high.

study on a paddy field under similar conditions of UAV image acquisition reported a 0.031 m vertical coordinate error (MAE) with a −60° camera angle and 2.10 m with a −90° camera angle on UAV-SfM point clouds with RTK without GCPs (Fujiwara et al., in press). A slight vertical deviation of point clouds with RTK positioning may cause low predictive performance for different flight data. For UAV-SfM reproducibility between different flights, GCPs seem to be more reliable than RTK.

In this study, the regression models in the two growth stages were trained separately. In contrast, a common model across growth stages scored a high coefficient of determination ($R^2$) in several studies (Madec et al., 2017; Tirado et al., 2020; Lu et al., 2022). Considering that data from multiple growth stages have high variance, the proportion of the variation explained by the model could be large. That is, when the data obtained from an early stage (e.g., PH = 0.5–1 m) and from a later stage (e.g., PH = 2–3 m) are mixed and fitted to a model, a high coefficient of determination is expected. However, errors such as MAE and RMSE may be larger than those specific to the growth stage. Stage-specific models are important for simultaneous evaluation. In this study, PH prediction of centimeter-level accuracy was made possible by separating the models from the vegetative and reproductive stages. Strategies should be selected by considering the target and accuracy.

With method M3, the coordinates of the terrain are extracted only around a field, and thus, this method is applicable to a field covered with plants. In this study, although the trial fields had passages without plants (**Figure 2**), these passages were not used as GA. This was because of the assumption of production fields without such a passage. It was shown that method M3 worked when the inside of the field was covered with plants.

In this study, all outer ROIs were used for polynomial fitting because bare soil was always visible, that is, weeds were few. When such bare soil areas around a field are unavailable, it may be better to eliminate some outer ROI areas. For a field completely covered with plants without any margin, applying method M3 would be difficult. In such a situation, method M1 with RTK, GCP, or both may be more suitable.

The shapefiles for ROIs were created on each orthomosaic in this study; thus, the horizontal deviation of the 3D models did not affect the predictive performance. However, when common ROIs in a field are used for different flights, such horizontal deviations can cause errors. To reduce the cost of creating ROIs on time-series datasets, high-precision positioning with GCPs or RTK could be beneficial, regardless of the method used to obtain GA.

Three-dimensional structural analysis with UAV-SfM is applicable to PH monitoring, yield prediction (Bendig et al., 2014; Li et al., 2016; Roth and Streit, 2018; Jiang et al., 2019; Karunaratne et al., 2020), and lodging detection (Chu et al., 2017; Yang et al., 2017). Moreover, PH data from UAV-SfM, such as the mean, percentiles, and coefficient of variation, can be combined with RGB and multispectral data for crop-monitoring systems (Li et al., 2016; Jiang et al., 2019; Karunaratne et al., 2020). The PH obtained using a high-precision and low-cost method is the basis for advanced demonstrations. The UAV-SfM methods demonstrated in this study can be applied to various targets and analytical strategies.

In this study, to evaluate the predictive performance of unknown data from another flight, a "different-targets-and-different-flight" cross-validation was conducted. It was suggested that method M1 with GCPs and method M3 could build regression models with the goodness of fit to unknown data. Particularly, with method M3, the predictive performance was high on "LC" without the use of GCPs or RTK. Therefore, this could work as a high-precision and low-cost method for general analysis based on UAV-SfM. Three-dimensional structural analysis using this method may prove useful for remote sensing of production fields.

## 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/s.

## AUTHOR CONTRIBUTIONS

RF, TK, HS, and YA designed this study and performed experiments. RF has contributed new analytical tools and analyzed the data. RF and YA wrote the manuscript. All authors have contributed to the manuscript and approved the submitted version.

## ACKNOWLEDGMENTS

We thank K. Ushijima, M. Suzuki, and K. Uesaka for their assistance in conducting the measurements.

## SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2022.886804/full#supplementary-material

## REFERENCES

Bendig, J., Bolten, A., and Bareth, G. (2013). UAV-based imaging for multi-temporal, very high resolution crop surface models to monitor crop growth variability. *PFG Photogramm Fernerkund Geoinformation* 6, 551–562. doi: 10.1127/1432-8364/2013/0200

Bendig, J., Bolten, A., Bennertz, S., Broscheit, J., Eichfuss, S., and Bareth, G. (2014). Estimating biomass of barley using crop surface models (CSMs) derived from UAV-based RGB imaging. *Remote Sens.* 6, 10395–10412. doi: 10.3390/rs61110395

Chang, A. J., Jung, J. H., Maeda, M. M., and Landivar, J. (2017). Crop height monitoring with digital imagery from Unmanned Aerial System (UAS). *Comput. Electron. Agric.* 141, 232–237. doi: 10.1016/j.compag.2017.07.008

Chu, T. X., Starek, M. J., Brewer, M. J., Murray, S. C., and Pruter, L. S. (2017). Assessing lodging severity over an experimental maize (*Zea mays* L.) field using UAS images. *Remote Sens.* 9:923. doi: 10.3390/rs9090923

Forlani, G., Dall'Asta, E., Diotri, F., Morra di Cella, U., Roncella, R., and Santise, M. (2018). Quality assessment of DSMs produced from UAV flights georeferenced with On-Board RTK positioning. *Remote Sens.* 10:311. doi: 10.3390/rs10020311

Fujiwara, R., Yasuda, H., Saito, M., Kikawada, T., Matsuba, S., Sugiura, R., et al. (in press). Investigation of a method to estimate culm length of rice based on aerial images using an unmanned aerial vehicle (UAV) equipped with high-precision positioning system. *Breed. Res.* doi: 10.1270/jsbbr.21J09 [Epub ahead of print].

Gillan, J. K., Karl, J. W., Duniway, M., and Elaksher, A. (2014). Modeling vegetation heights from high resolution stereo aerial photography: an application for broad-scale rangeland monitoring. *J. Environ. Manage.* 144, 226–235. doi: 10.1016/j.jenvman.2014.05.028

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

Iqbal, F., Lucieer, A., Barry, K., and Wells, R. (2017). Poppy crop height and capsule volume estimation from a single UAS flight. *Remote Sens.* 9:647. doi: 10.3390/rs9070647

James, M. R., and Robson, S. (2014). Mitigating systematic error in topographic models derived from UAV and ground-based image networks. *Earth Surf. Process. Landf.* 39, 1413–1420. doi: 10.1002/esp.3609

Jiang, Q., Fang, S. H., Peng, Y., Gong, Y., Zhu, R. S., Wu, X. T., et al. (2019). UAV-based biomass estimation for rice-combining spectral, TIN-based structural and meteorological features. *Remote Sens.* 11:890. doi: 10.3390/rs11070890

Karunaratne, S., Thomson, A., Morse-McNabb, E., Wijesingha, J., Stayches, D., Copland, A., et al. (2020). The fusion of spectral and structural datasets derived from an airborne multispectral sensor for estimation of pasture dry matter yield at paddock scale with time. *Remote Sens.* 12:2017. doi: 10.3390/rs12122017

Kawamura, K., Asai, H., Yasuda, T., Khanthavong, P., Soisouvanh, P., and Phongchanmixay, S. (2020). Field phenotyping of plant height in an upland rice field in Laos using low-cost small unmanned aerial vehicles (UAVs). *Plant Prod. Sci.* 23, 452–465. doi: 10.1080/1343943X.2020.1766362

Li, W., Niu, Z., Chen, H. Y., Li, D., Wu, M. Q., and Zhao, W. (2016). Remote estimation of canopy height and aboveground biomass of maize using high-resolution stereo images from a low-cost unmanned aerial vehicle system. *Ecol. Indic.* 67, 637–648. doi: 10.1016/j.ecolind.2016.03.036

Lu, W., Okayama, T., and Komatsuzaki, M. (2022). Rice height monitoring between different estimation models using UAV photogrammetry and multispectral technology. *Remote Sens.* 14:78. doi: 10.3390/rs14010078

Madec, S., Baret, F., de Solan, B., Thomas, S., Dutartre, D., Jezequel, S., et al. (2017). High-throughput phenotyping of plant height: comparing unmanned aerial vehicles and ground LiDAR estimates. *Front. Plant Sci.* 8:2002. doi: 10.3389/fpls.2017.02002

Malambo, L., Popescu, S. C., Murray, S. C., Putman, E., Pugh, N. A., Horne, D. W., et al. (2018). Multitemporal field-based plant height estimation using 3D point clouds generated from small unmanned aerial systems high-resolution imagery. *Int. J. Appl. Earth Obs. Geoinf.* 64, 31–42. doi: 10.1016/j.jag.2017.08.014

Murakami, T., Yui, M., and Amaha, K. (2012). Canopy height measurement by photogrammetric analysis of aerial images: application to buckwheat (*Fagopyrum esculentum* Moench) lodging evaluation. *Comput. Electron. Agric.* 89, 70–75. doi: 10.1016/j.compag.2012.08.003

Paturkar, A., Sen Gupta, G., and Bailey, D. (2021). Making use of 3D models for plant physiognomic analysis: a review. *Remote Sens.* 13 doi: 10.3390/rs13112232

Rosnell, T., and Honkavaara, E. (2012). Point cloud generation from aerial image data acquired by a Quadrocopter type micro unmanned aerial vehicle and a digital still camera. *Sensors (Basel)* 12, 453–480. doi: 10.3390/s120100453

Roth, L., and Streit, B. (2018). Predicting cover crop biomass by lightweight UAS-based RGB and NIR photography: an applied photogrammetric approach. *Precis. Agric.* 19, 93–114. doi: 10.1007/s11119-017-9501-1

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

Štroner, M., Urban, R., Reindl, T., Seidl, J., and Brouèek, J. (2020). Evaluation of the georeferencing accuracy of a photogrammetric model using a Quadrocopter with onboard GNSS RTK. *Sensors (Basel).* 20:2318. doi: 10.3390/s20082318

Tirado, S. B., Hirsch, C. N., and Springer, N. M. (2020). UAV-based imaging platform for monitoring maize growth throughout development. *Plant Direct* 4:e00230. doi: 10.1002/pld3.230

Tsouros, D. C., Bibi, S., and Sarigiannidis, P. G. (2019). A review on UAV-based applications for precision agriculture. *Information* 10:349. doi: 10.3390/info10110349

Volpato, L., Pinto, F., González-Pérez, L., Thompson, I. G., Borém, A., Reynolds, M., et al. (2021). High throughput field phenotyping for plant height using UAV-based RGB imagery in wheat breeding lines: feasibility and validation. *Front. Plant Sci.* 12:591587. doi: 10.3389/fpls.2021.591587

Weiss, M., Jacob, F., and Duveiller, G. (2020). Remote sensing for agricultural applications: a meta-review. *Remote Sens. Environ.* 236:111402. doi: 10.1016/j.rse.2019.111402

Westoby, M. J., Brasington, J., Glasser, N. F., Hambrey, M. J., and Reynolds, J. M. (2012). 'Structure-from-Motion' photogrammetry: a low-cost, effective tool for geoscience applications. *Geomorphology* 179, 300–314. doi: 10.1016/j.geomorph.2012.08.021

Wu, W., Guo, F., and Zheng, J. (2020). Analysis of Galileo signal-in-space range error and positioning performance during 2015–2018. *Satell. Navig.* 1, 1–13. doi: 10.1186/s43020-019-0005-1

Xie, T. J., Li, J. J., Yang, C. H., Jiang, Z., Chen, Y. H., Guo, L., et al. (2021). Crop height estimation based on UAV images: methods, errors, and strategies. *Comput. Electron. Agric.* 185:13. doi: 10.1016/j.compag.2021.106155

Yang, M. D., Huang, K. S., Kuo, Y. H., Tsai, H. P., and Lin, L. M. (2017). Spatial and spectral hybrid image classification for rice lodging assessment through UAV imagery. *Remote Sens.* 9:583. doi: 10.3390/rs9060583

Yao, H., Qin, R. J., and Chen, X. Y. (2019). Unmanned aerial vehicle for remote sensing applications—a review. *Remote Sens.* 11:1443. doi: 10.3390/rs11121443

Ziliani, M. G., Parkes, S. D., Hoteit, I., and McCabe, M. F. (2018). Intra-season crop height variability at commercial farm scales using a fixed-wing UAV. *Remote Sens.* 10:2007. doi: 10.3390/rs10122007

**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.

ORIGINAL RESEARCH
published: 04 July 2022
doi: 10.3389/frai.2022.872858

# Comparing Deep Learning Approaches for Understanding Genotype × Phenotype Interactions in Biomass Sorghum

*Zeyu Zhang*[1], *Madison Pope*[2], *Nadia Shakoor*[3], *Robert Pless*[1], *Todd C. Mockler*[3] and *Abby Stylianou*[2]*

*[1] Department of Computer Science, George Washington University, Washington, DC, United States, [2] Department of Computer Science, Saint Louis University, Saint Louis, MO, United States, [3] Donald Danforth Plant Science Center, Mockler Lab, Saint Louis, MO, United States*

OPEN ACCESS

**Edited by:**
*Ian Stavness,*
*University of Saskatchewan, Canada*

**Reviewed by:**
*Bedrich Benes,*
*Purdue University, United States*
*Andrew P. French,*
*University of Nottingham,*
*United Kingdom*

***Correspondence:**
*Abby Stylianou*
*abby.stylianou@slu.edu*

**Specialty section:**
*This article was submitted to*
*AI in Food, Agriculture and Water,*
*a section of the journal*
*Frontiers in Artificial Intelligence*

***Received:*** *10 February 2022*
***Accepted:*** *09 June 2022*
***Published:*** *04 July 2022*

**Citation:**
*Zhang Z, Pope M, Shakoor N, Pless R, Mockler TC and Stylianou A (2022) Comparing Deep Learning Approaches for Understanding Genotype × Phenotype Interactions in Biomass Sorghum. Front. Artif. Intell. 5:872858. doi: 10.3389/frai.2022.872858*

We explore the use of deep convolutional neural networks (CNNs) trained on overhead imagery of biomass sorghum to ascertain the relationship between single nucleotide polymorphisms (SNPs), or groups of related SNPs, and the phenotypes they control. We consider both CNNs trained explicitly on the classification task of predicting whether an image shows a plant with a reference or alternate version of various SNPs as well as CNNs trained to create data-driven features based on learning features so that images from the same plot are more similar than images from different plots, and then using the features this network learns for genetic marker classification. We characterize how efficient both approaches are at predicting the presence or absence of a genetic markers, and visualize what parts of the images are most important for those predictions. We find that the data-driven approaches give somewhat higher prediction performance, but have visualizations that are harder to interpret; and we give suggestions of potential future machine learning research and discuss the possibilities of using this approach to uncover *unknown* genotype × phenotype relationships.

**Keywords: deep learning, convolutional neural networks, explainable AI, visualization, single nucleotide polymorphism, phenotyping, sorghum, TERRA-REF**

## 1. INTRODUCTION

Sorghum is a cereal crop, used worldwide for a variety of purposes including for use as grain and as a source of biomass for bio-energy production. For biofuel production, the goal of both plant growers and breeders is to produce sorghum crops that grow as big as possible, as quickly as possible, with as few resources as possible. Plant breeders produce new lines of sorghum by crossing together candidate lines that have desirable traits, or known genes that correspond to desirable traits.

Understanding the relationship between genetics and traits is key to improving the breeding process, and to understanding of plant biology in general. High throughput phenotyping (Araus and Cairns, 2014) takes advantage of progress in sensor platforms able to measure data about plant growth and traits at large scale to better understand these relationships.

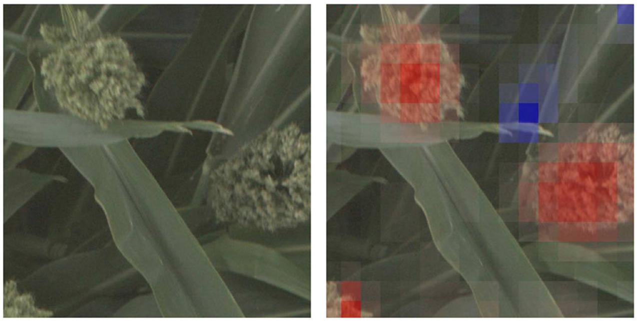


**FIGURE 1 |** We train deep convolutional neural network classifiers to predict whether an image of a sorghum crop contains a reference or alternate version of particular genetic marker, and then visualize why the network makes that prediction. In this figure, we show the visualization for why the neural network predicted an image showed a plant with an alternate version of a SNP that controls, among other phenotypes, panicle shape (Hilley et al., 2017)—the visualization highlights (in red) the panicle as an important feature in the networks prediction.

In this paper, we propose using deep convolutional neural networks (CNNs) as a computational platform to understand and identify interesting genetic markers that control visually observable traits. The pipelines we present can be leveraged by plant geneticists and breeders to understand the relationship between single nucleotide polymorpishms (SNPs, locations in the organism's DNA that vary between different members of the population), or groups of related SNPs, and the phenotypes that they impact. We explore these genotype × phenotype relationships by training CNNs to predict whether images of biomass sorghum show plants that have reference or alternate versions of different genetic markers, and then making visualizations that highlight the image features that lead to the predictions. For models that can perform this classification task with high accuracy, the visualizations highlight phenotypes that correlate with the genetic marker. **Figure 1** shows such a visualization for a genetic marker that controls panicle shape—the visualization shows that the machine learning model learned to focus on the panicles, while not focusing on other plant parts.

We consider two approaches to performing this classification and visualization task. The first approach directly trains a CNN to classify images by their genetic variations. The second approach involves first learning an embedding that can distinguish between different varieties of sorghum, and then training different classifiers on top of that embedding. In both cases, we can quantitatively evaluate how well the models can be used to predict genetic variations and qualitatively assess whether the visualizations provide meaningful and biologically relevant information about the genotype × phenotype relationship.

We demonstrate the feasibility and utility of these pipelines on a number of SNPs identified in the sorghum Bioenergy Association Panel (Brenton et al., 2016) (BAP), a set of 390 sorghum cultivars whose genomes have been fully sequenced and which show promise for bio-energy usage. We focus on SNPs and groups of SNPs with known phenotypic expression in order to validate our approach. We highlight both quantitative results, demonstrating that classification and embedding networks can successfully be trained to predict genetic variation in biomass sorghum, and present example visualizations which highlight that the relevant features learned by these networks correspond to features documented in existing literature about the different genetic markers. The success of this approach on genetic markers with known genotype × phenotype relationships indicates that the same approach could be extended to genetic markers whose phenotypic expression is less well understood, which could help to accelerate crop breeding programs.

## 2. BACKGROUND

### 2.1. Sorghum and Polymorphisms

Sorghum is a diploid species, meaning that it has two copies of each of its 10 chromosomes. Each chromosome consists of DNA, the genetic instructions for the plant. The DNA itself is made up of individual nucleotides, sequences of which tell the plant precisely which proteins to make. Variations in these sequences, called single nucleotide polymorphisms, can result in changes to the proteins the plant is instructed to make, which in turn can have varying degrees of impact on the structure and performance of the plant. Understanding the impact that specific genes have on plants and how they interact with their environment is a fundamental problem and area of study in plant biology (Bochner, 2003; Schweitzer et al., 2008; Cobb et al., 2013; Boyles et al., 2019; Mural et al., 2021).

Single nucleotide polymorphisms (SNPs) are specific variations that exist between different members of a population at a single location on the chromosome, where one adenine, thymine, cytosine or guanine nucleotide in one plant may be have one or more different nucleotides in a different plant. This variation can exist on one or both copies of the chromosome. A cultivar that has the "original" version of the SNP on both copies of the chromosome is referred to as being homozygous reference; a cultivar that has variant on both copies of the chromosome is referred to as being homozygous alternate; and a cultivar that has one normal and one variant version of the SNP is called heterozygous. In this paper we consider only the homozygous cases, and how deep convolutional neural networks can be used to predict whether imagery of sorghum plants shows a plant with a reference or alternate version of a particular SNP or family of related SNPs.

### 2.2. TERRA-REF

We work with data collected by the Transportation Energy Resources from Renewable Agriculture Phenotyping Reference Platform, or TERRA-REF, project which was funded by the Advanced Research Project Agency–Energy (ARPA-E) in 2016 (Burnette et al., 2018; LeBauer et al., 2020). The TERRA-REF platform is a state-of-the-art gantry based system for monitoring the full growth cycle of over an acre of crops with a cutting-edge suite of imaging sensors, including stereo-RGB, thermal, short- and long-wave hyperspectral cameras, and laser 3D-scanner sensors. The goal of the TERRA-REF gantry was to perform in-field automated high throughput plant phenotyping,

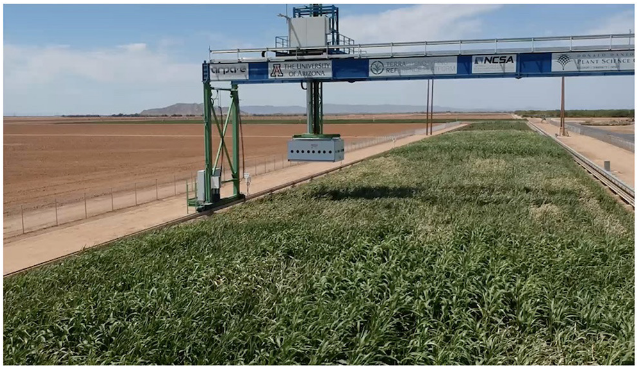


**FIGURE 2** | The TERRA-REF Field and Gantry-based Field Scanner in Maricopa, Arizona, with sorghum being grown in the field.

the process of making phenotypic measurements of the physical properties of plants at large scale and with high temporal resolution, for the purpose of better understanding the difference between crops and facilitating rapid plant breeding programs. The TERRA-REF field and gantry system are shown in **Figure 2**.

Since 2016, the TERRA-REF platform has collected petabytes of sensor data capturing the full growing cycle of sorghum plants from the sorghum Bioenergy Association Panel (Brenton et al., 2016), a set of 390 sorghum cultivars whose genomes have been fully sequenced and which show promise for bio-energy usage. The full, original TERRA-REF dataset is a massive public domain agricultural dataset, with high spatial and temporal resolution across numerous sensors and seasons, and includes a variety of environmental data and extracted phenotypes in addition to the sensor data. More information about the dataset and access to it can be found in LeBauer et al. (2020).

## 2.3. Deep Learning for Agriculture

To our knowledge, ours is the first work that trains classifiers on visual sensor data to predict whether an image shows organisms with a reference or alternate version of a genetic marker in order to better understand the genotype × phenotype relationship. There is related work in genomic selection that attempts to predict end-of-season traits like leaf or grain length and crop yield (Sandhu et al., 2021) from genetic information, and in using 3D reconstructions of plants to identify leaf-angle related loci in the sorghum genome (Tross et al., 2021). In Liu et al. (2019), the most related work to ours, the authors train CNNs to predict quantitative traits from SNPs, and use a visualization approach called saliency maps to highlight the *SNPs* that most contributed to predicting a particular trait (as opposed to predicting whether a SNP is reference or alternate, and what visual components led to that classification). There is additionally work that attempts to use deep learning to predict the relative functional importance of specific genetic markers and mutations in plants (Wang et al., 2020), without focusing on visualizing their specific impact on the expressed phenotypes.

There is generally significantly more work in applying deep learning for a wide variety of plant phenotyping and agriculture tasks that do not incorporate the underlying genetics—for example, deep CNNs have successfully been used for fruit detection (Sa et al., 2016; Bargoti and Underwood, 2017; Lim and Chuah, 2018; Koirala et al., 2019; Wan and Goudos, 2020), cultivar and species identification (Barré et al., 2017; Lim and Chuah, 2018; Van Horn et al., 2018; Ashqar et al., 2019; Osako et al., 2020; Heidary-Sharifabad et al., 2021; Ren et al., 2021), plant disease classification (Mohanty et al., 2016; Wang et al., 2017; Ferentinos, 2018; Too et al., 2019), leaf counting (Aich and Stavness, 2017; Dobrescu et al., 2017; Giuffrida et al., 2018; Ubbens et al., 2018; Miao et al., 2021), yield prediction (Wang et al., 2018; Chen et al., 2019; Nevavuori et al., 2019; Maimaitijiang et al., 2020), and stress detection (Anami et al., 2020; Butte et al., 2021; Chandel et al., 2021), among other phenotyping tasks. These deep learning approaches are sensitive to the amount of labeled data available, and the previous works take advantage of a combination of fine-tuning CNN networks trained for other tasks, heroic efforts to hand-label sufficient data to support the learning tasks, or working with existing high-throughput phenotyping data to bootstrap the learning process.

## 2.4. Latent Space Learning and Embedding Networks

When there are too few labels for standard deep learning approaches to work, there are sometimes widely available labels that are still somehow related. These can support alternative ways to train a CNN. One approach is called Deep Metric Learning, and this takes advantages of circumstances when there are sets of images whose labels are unknown, but known to be the same as each other. For example, if you have sets of images that are known to be from the same sorghum cultivar, then you know that those images have the same (but unknown) genetic markers as each other. For such data, deep metric learning trains convolutional neural networks to extract output features from images so that input data from the same class produce similar output features, and input data from different classes produce different output features.

Many approaches to solve this problem have been proposed in recent years, both varying specific loss functions to define the embedding (Hadsell et al., 2006; Sohn, 2016; Ge, 2018; Kim et al., 2018; Xuan et al., 2018), and proposing interesting datasets along with loss functions (Schroff et al., 2015; Song et al., 2016). In this work we use a variation called the Proxy Loss approach described in (Movshovitz-Attias et al. (2017) and Boudiaf et al. (2020), which was recently used for plant-recognition based on flower images (Zhang et al., 2021). This trains an embedding network so that images taken from the same field plot are mapped closer together than images taken from different field plots; this source of weak labeling would apply to any situation where field plots consist of unique cultivars.

The idea of embedding images into a feature space that captures fundamental variations in crop varieties was proposed as "Latent Space Phenotyping" (Ubbens et al., 2020), where the authors used a similar approach to automatically find image features that highlight differentiated response to treatment effects. In their case, the embedding network is trained to learn

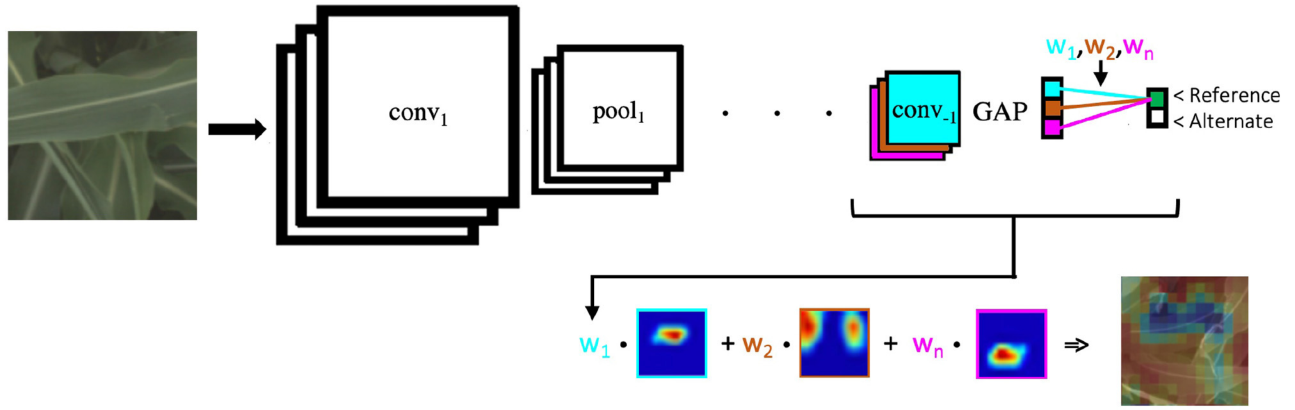


**FIGURE 3 |** We use a standard ResNet-50 architecture, which like many deep convolutional neural networks consists of alternating convolutional and pooling layers (with interspersed activation functions). The network ends with a final convolutional layer (*conv*$_{-1}$), a global average pooling (GAP) operation, and then a fully connected layer, the output of which is used to make our prediction of whether an image shows a plant with a reference or alternate version of a particular genetic marker. We use the class activation mapping approach described in Zhou et al. (2016), in which the filters in the last convolutional layer are multiplied by the corresponding weights between the respective layer and the predicted output node. These weighted filters are then added up to produce a heatmap that has its highest values in important regions.

image features that best capture how the plants in the dataset respond to the experimental treatment (such as drought stress or nitrogen deficiency), to discover image features that might not correlate to standard phenotypes. In our case, we build a network that embeds images into a latent space that helps differentiate many different cultivars, and show that this latent space supports classification of cultivars based on several genetic markers.

## 2.5. Visualization Approaches

A common strategy for making deep convolutional neural networks and their decisions more interpretable is to produce automatically generated visualizations that highlight the most important regions in images for a particular output. There are a variety of different approaches for making these visualizations, including output-agnostic approaches that generate a binary relevancy map by thresholding the values of a feature map from a given layer in the network (Zhou et al., 2015; Bau et al., 2017) or incorporate deconvolutional neural networks to transform activation maps into the original pixel space (Zeiler and Fergus, 2014).

One of the most common styles of visualizations that is output-specific is the Class Activation Map (CAM) (Zhou et al., 2016), which were shown to produce discriminative visualizations. CAMs are generated by taking a weighted sum of the feature maps produced by the last convolutional layer in the network, using the weights of the global pooled feature with respect to the target class as a multiplier (as shown in **Figure 3**. An extension of CAM, GradCAM (Selvaraju et al., 2017) generalizes this framework for different network layers and architectures, weighting the feature maps by the gradients with respect to the target class.

For embedding networks there are fewer visualization approaches. In Chen et al. (2020), the authors extend the GradCAM approach to embedding networks by averaging the gradients from sampled training triplets. To produce the visualization of a test image, the gradients of the most similar training image are used for the weighted sum of the feature maps. In Stylianou et al. (2019), the authors introduce a method for generating heatmaps from a pair of images which highlight the regions that contribute the most to their pairwise similarity by decomposing the similarity calculation across each spatial location in the final feature maps of both images.

In this paper, we focus on the Class Activation Map style visualization to understand the predictions of deep convolutional neural networks relative to particular families of genetic markers in biomass sorghum.

## 3. DATASET DETAILS

To support our study on the usage of deep convolutional neural networks to understand the genotype × phenotype relationship in biomass sorghum, we leverage RGB imagery from the TERRA-REF gantry described in Section 2.2. We specifically focus on images from the 2017 growing season, when cultivars from the sorghum Biomass Association Panel (BAP) (Brenton et al., 2016) were grown. Each cultivar was grown in two spatially separated plots.

The original TERRA-REF dataset provides raw RGB images that are 3296 × 2016 pixels. There are approximately 11 images that mostly or completely image each plot for a given day. In pre-processing the raw imagery for our task, images that cross the plot boundary are cropped into multiple images that each contain pixels of plants from only one plot. This data is then organized into various datasets for our specific task of understanding the genotype × phenotype relationship.

Our study focuses on two different strategies for training CNNs for this task—the first approach directly trains CNNs to classify images as having the "reference" or "alternate" version of

**TABLE 1 |** Details about the genetic marker families of interest.

| Genetic marker family | SNP details | | | |
|---|---|---|---|---|
| | Chromosome | Gene | Position | Known controlled phenotype |
| Leaf wax | 1 | 001G269200 | 51,588,525 | Wax composition (Uttam et al., 2017) |
| | 1 | 001G269200 | 51,588,838 | |
| | 1 | 001G269200 | 51,589,143 | |
| | 1 | 001G269200 | 51,589,435 | |
| dw | 6 | 006G067700 | 42,805,319 | Plant height and structure, stem length and internode length (Yamaguchi et al., 2016; Hilley et al., 2017) |
| | 6 | 006G067700 | 42,804,037 | |
| Dry Stalk (d) locus | 6 | 006G147400 | 50,898,459 | Plant height and structure, and sugar composition (Xia et al., 2018) |
| | 6 | 006G147400 | 50,898,536 | |
| | 6 | 006G147400 | 50,898,315 | |
| | 6 | 006G147400 | 50,898,231 | |
| | 6 | 006G147400 | 50,898,523 | |
| | 6 | 006G147400 | 50,898,525 | |
| ma | 6 | 006G057866 | 40,312,463 | Flowering time and maturity (Murphy et al., 2014; Wang et al., 2015; Cuevas et al., 2016) |
| | 6 | 006G004400 | 2,697,734 | |
| tan | 9 | 009G229800 | 57,040,680 | Pigmentation and tannin production (Wu et al., 2012) |

*Single nucleotide polymorphisms are grouped by the phenotypes they control, and classification is performed by genetic marker family. Cultivars are defined as reference if they have the reference version of all SNPs on both copies of the chromosomes, and as alternate if they have the alternate version of all SNPs on both copies of the chromosomes (we do not consider heterozygous cultivars).*

a particular genetic marker or family of related SNPs; the second approach first trains a genetic-marker agnostic embedding, where images from the same plot are encouraged to have features that are similar and images from different plots are encouraged to have features that are dissimilar. A genetic-marker specific classifier is then trained on top of the genetic-marker agnostic embedding model. Below we describe the specific datasets used for the classification and embedding tasks.

## 3.1. Classification Dataset

In the classification setting, we train a neural network directly on the task of predicting whether an image fed into the network shows a plant that is homozygous reference or homozygous alternate for a particular genetic markers.

In this paper, we focus on the five genetic markers listed in **Table 1**. Each genetic marker is defined by one or more related SNPs, which have been identified in prior work as having a particular phenotype that is impacted depending on whether the cultivar being grown has the reference or alternate version of the marker.

For a cultivar to be labeled reference for a particular genetic marker, it must have the reference version of all SNPs in the family; cultivars are labeled alternate if they have the alternate version of any of the SNPs in the family—this is because even one polymorphism can significantly impact the phenotype being controlled. (We do not consider heterozygous cultivars.)

For each genetic marker, we then count the total number of reference and the total number of alternate cultivars; the minimum count determines the number of cultivars that are put into the genetic marker family specific training and testing sets—the testing set includes half of the cultivars from whichever class has fewer cultivars, and an equal number cultivars from the more represented class.

We additionally balance our testing set such that there are an equal number of reference and alternate images from an equal number of reference and alternate cultivars (both images and cultivars are randomly selected from the initial test set to guarantee this balance). This guarantees that the performance of a random classifier would be at 50% if predicting either per-image or per-cultivar classification accuracy.

All remaining cultivars are put into the training set, without limiting the number of images per cultivar—this allows us to use a large number of training examples, even if there may be imbalance in the number of images per class (reference vs. alternate) or per cultivar. This imbalance is dealt with at training time by an imbalanced sampler per batch, which selects roughly equal numbers of images from the population of reference and alternate examples.

There is no overlap between the training and testing cultivars.

## 3.2. Embedding Dataset

For the embedding approach, we first train a deep CNN to learn a genetic-marker agnostic representation. To do this, we use all available plot-cropped RGB images from the June 2017 TERRA-REF dataset. These images are labeled by plot. This Embedding Pre-training Dataset contains images from both the classification

**TABLE 2** | Dataset statistics.

| Genetic marker family | # Train cultivars | | # Test cultivars | | # Train images | | # Test images | |
|---|---|---|---|---|---|---|---|---|
| | Ref | Alt | Ref | Alt | Ref | Alt | Ref | Alt |
| Leaf wax | 67 | 114 | 34 | 34 | 6,700 | 11,400 | 3,400 | 3,400 |
| dw | 80 | 105 | 40 | 40 | 8,000 | 10,500 | 4,000 | 4,000 |
| Dry Stalk (d) locus | 43 | 127 | 21 | 21 | 4,300 | 12,700 | 2,100 | 2,100 |
| ma | 21 | 167 | 10 | 10 | 2,100 | 16,700 | 1,000 | 1,000 |
| tan | 133 | 53 | 27 | 27 | 13,300 | 5,300 | 2,700 | 2,700 |

*The number of cultivars and images used in the training and testing sets for each of the genetic marker families.*

training and testing set, but no knowledge of the data's genetic marker labels is used to learn the representation.

After the pre-training stage, we are able to then train genetic marker family specific classifiers on top of the embedding model. Details of these classifiers and how they are trained are discussed in more detail in Section 4.2. The test datasets used to evaluate these classifiers are the same as in the classification pipeline. This is acceptable despite the existence of these testing images in the Embedding Pre-Training Dataset as we only use the *plot* labels to pre-train the network; the genetic marker labels are unseen during this stage. Genetic marker dataset splitting that is based on cultivars also assures the plot label pre-training does not force the model to map train and test images together.

**Table 2** shows the exact number of cultivars and images used in the classification training and testing sets for each genetic marker family (the Embedding Pre-training Dataset consists of all available plot-cropped images). We only consider images from June of 2017, mid-way through the growing season when plants are not too small, exhibiting many of the phenotypes of interest, and not yet lodging (falling over) on top of each other.

## 4. METHODS

Our approach to gaining understanding about the genotype × phenotype relationship in biomass sorghum is to train deep convolutional neural networks to predict whether an image shows a sorghum cultivar with the reference or alternate version of a specific SNP or group of related SNPs, and to then visualize the specific features the network focuses on when making that determination. If the classifier can perform well above chance performance on this classification task, then it is learning something that is significantly correlated with the genetics being considered, and the visualizations can help us glean insights into precisely what those correlations are.

### 4.1. Training Pipeline 1: Classification

We train a ResNet-50 model (He et al., 2016), pre-trained on the ImageNet dataset (Deng et al., 2009), with a single fully connected layer on the reference vs. alternate classification task. A general overview of this type of network architecture is shown at the top of **Figure 3**.

For all families of genetic markers, the network is trained on 512 × 512 plot-cropped RGB images from the datasets described in Section 3. The weights of the entire network are trained using the adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.0001 for 20 epochs. For data augmentation, we subtract by dataset channel means and divide by dataset channel standard deviations, and during training we perform random horizontal flips. The 512 × 512 pixel images are extracted by resizing the image on its largest side to 512 and extracting a random crop at training time, and a center crop at testing time. We use imbalanced batch sampling during training to fill 100 image batches with a roughly equal number of reference and alternate images per batch, even if there is an imbalance in the number of reference and alternate images in the training set.

### 4.2. Training Pipeline 2: Embedding

#### 4.2.1. Pre-training

As in the classification pipeline, we start from a ResNet-50 model pre-trained on ImageNet. Instead of having a two-dimensional output (as we have in the classification pipeline), the output is 700-dimensional, and the network's task is to correctly classify which of the 700 field plots an image came from.

During the pre-training, we use 25 images per batch, with each image labeled by plot number.

Our embedding network loss function uses a cross-entropy variant of Proxy Loss (Movshovitz-Attias et al., 2017; Boudiaf et al., 2020), optimize the network using SGD (Sutskever et al., 2013) with an initial learning rate of 0.01, learning rate decay of 0.1 every 10 epochs, and a momentum term of 0.9. We train for 40 epochs, stopping based on training loss convergence. We use the same data augmentation strategies as in the classification pipeline.

#### 4.2.2. Genetic Marker Prediction Using Embedding Model

Once this pre-training is complete, we freeze the weights of the network and the plot-level classification layer is chopped off, yielding a network that ends with the 2,048-dimensional output of the ResNet-50's Global Average Pooling (GAP) layer, which we use as our feature embedding. This output of the GAP layer is established to be an excellent representation across datasets and problem domains in Vo and Hays (2019). This embedding feature can then either be used directly in inferring genetic marker labels (for example, using k-Nearest Neighbors) or fed into a classifier (for example, a support vector machine or a new classification head on the pre-trained neural network). We discuss these approaches below.

**k-Nearest Neighbors:** In order to predict a genetic marker label using k-Nearest Neighbors, we first extract the 2,048-dimensional embedding feature for each of the images in both the classification training and the testing sets. For every feature in the test set, we look up its k-nearest neighbors in the training set and infer whether the test image is reference or alternate from the mode of the nearest neighbors. We use the value $k = 11$ in all experiments based on empirical testing.

**Support Vector Machine:** To predict a genetic marker label with a support vector machine, we first extract the 2,048-dimensional embedding feature for each of the images in the classification training and testing sets. We use PCA to reduce the dimensionality of these features from 2,048 to 60, and then use the classification training images and labels to train a support vector machine with a radial basis function kernel, and evaluate performance on the classification test set.

**Classification Head on Embedding Network:** For each genetic marker, we take the pre-trained embedding network and add a fully connected layer with a 2-dimensional output. We fine-tune this fully connected layer using the images and labels from the classification training set (the preceding network weights remain frozen). Performance is evaluated on the classification test set. We use SGD with a learning rate of 0.1 learning rate and 0.1 learning rate decay every 5 epochs during training (with no momentum). We stop training based on training accuracy convergence.

#### 4.2.3. Evaluation Settings

When computing the accuracy of each approach on the classification test set, we can consider accuracy per image, per cultivar and per plot-day. Accuracy per image is computed by simply measuring the average accuracy of predicting the correct label over all images in a test set. Accuracy per cultivar is computed by making per-image predictions for all images from a cultivar in a test set, and selecting the mode from those predictions as the cultivar label. This setting does require knowledge of the test set cultivar labels.

Accuracy per plot-day is computed by taking all of the 2,048-dimensional embedding features from a specific plot on a specific day and averaging them together to produce a plot-day embedding feature. This feature can then be used in place of the original embedding features as the input to the k-Nearest Neighbor or SVM classification (this setting is not applicable for the approach where a fully connected layer is added to the embedding model and trained for each genetic marker).

We discuss the relative classification accuracy of each of the genetic marker prediction approaches and each of the evaluation settings on the genetic marker classification task in Section 5.1.

### 4.3. Visualization Pipeline

It is not our ultimate goal to merely show which of the above strategies yields the highest quantitative performance at predicting whether an image shows a plant that has the reference or alternate version of a particular genetic marker. Instead, we hope to clarify the genotype × phenotype relationship that each of these genetic markers. In order to do this, we propose to automatically highlight the visual features that the neural networks learn are most important in accurately predicting reference vs. alternate. Those visual features are correlated with the genetic markers, and reviewing them can provide insights about what phenotypes the genetic markers are controlling.

In order to make such visualizations, we use the Class Activation Mapping approach described in Zhou et al. (2016), which highlights the image regions that most contributed to a classification of the neural network. This approach is detailed in the bottom of **Figure 3**, where the filters in the last convolutional layer are multiplied by the corresponding weights between the respective layer and the predicted output node. These weighted filters are then added up to produce a heatmap that has its highest values in important regions (e.g., the red regions in **Figure 1**). We use this approach to compare the predictions among different methods on a particular genetic marker family to understand the different visual traits correlated with being either reference or alternate.

We are able to use this visualization strategy both for the classification pipeline, as well as the version of the embedding pipeline where we train a genetic marker specific fully connected layer at the end of the embedding network. We compare the visualizations from these different approaches and discuss the biological relevance of them in Section 5.2.

## 5. RESULTS

### 5.1. Genetic Marker Prediction Accuracy

In **Table 3** we show the test set classification accuracy for all five genetic markers using both the classification and embedding pipelines. We compute the accuracy per image as well as the accuracy achieved by taking the mode of the predictions from all images of a cultivar, as described in Section 4.2.3. Taking the mode per cultivar outperforms the per image accuracy for all but the ma genetic marker. This is possibly due to the large imbalance in the number of images per class in the ma training set (the ratio between reference and alternate images of ma is 1:8, as seen in **Table 2**). This significant imbalance may lead the classifiers that utilize the training set (the k-NN and SVM approaches) to be biased toward predicting the alternate class, resulting in roughly chance performance.

Overall the best classification performance is achieved by the approach where we train a fully connected layer on top of the pre-trained embedding model for each genetic marker. This indicates, for single genetic marker prediction task, the embedding network extracts richer features than the direct classification approach.

#### 5.1.1. Per Plot-Day Results

As discussed in Section 3, there are multiple images per plot on any given day in the dataset due to the configuration of the TERRA-REF field and imaging protocols. Any one of these pictures shows only a subset of the plants in a specific plot, and it may be the case that one picture contains relevant visual features for the plot that are not present in a different picture (e.g., one picture might show a particularly indicative panicle while others do not). This suggests that an approach that aggregates features across all of the images from a plot could achieve superior performance.

**TABLE 3 |** Classification accuracy by image and by cultivar.

| Genetic marker | Classification | | Embedding + k-NN | | Embedding + SVM | | Embedding + fc | |
|---|---|---|---|---|---|---|---|---|
| | Image | Cultivar | Image | Cultivar | Image | Cultivar | Image | Cultivar |
| Leaf Wax | 0.611 | 0.706 | 0.641 | 0.632 | 0.656 | 0.647 | **0.668** | ***0.721*** |
| dw | 0.600 | 0.650 | 0.660 | ***0.750*** | **0.676** | 0.738 | 0.655 | 0.713 |
| d locus | 0.642 | 0.762 | 0.669 | 0.667 | 0.665 | 0.667 | **0.734** | ***0.833*** |
| ma | 0.629 | 0.600 | 0.556 | 0.500 | 0.570 | 0.500 | **0.630** | ***0.650*** |
| tan | 0.646 | ***0.796*** | **0.682** | 0.741 | **0.682** | 0.704 | 0.667 | 0.704 |

*For each genetic marker, we compare the accuracy of the direct classification approach with each of the approaches that use the embedding pre-training [k-NN, SVM and adding a fully connected (fc) layer]. Accuracy per image is computed on each image in the test set separately. Accuracy per cultivar is computed by taking the mode of the image predictions from each cultivar. The test set for each genetic marker family is balanced such that the classification accuracy by both image and by cultivar are 0.5. For each genetic marker, the highest accuracy per image is shown in bold text, while the highest accuracy per cultivar is shown in italicized bold text.*

**TABLE 4 |** Comparison with per plot-day features.

| Genetic marker | Per image | Per plot-day |
|---|---|---|
| Leaf wax | 0.656 | 0.699 |
| dw | 0.676 | 0.685 |
| Dry Stalk (d) locus | 0.665 | 0.741 |
| ma | 0.570 | 0.761 |
| tan | 0.682 | 0.733 |

*Comparison of the accuracy of the SVM classification approach using the embedding features for individual images as input vs. using plot-day aggregated features (generated using the average pooling described in Section 4.2.3) as inputs.*

In **Table 4**, we compare the accuracy of the SVM approach using the embedding features for individual images as input vs. using plot-day aggregated features (generated using the average pooling described in Section 4.2.3) as inputs in both training and testing. This plot-day aggregation over all of the images from a plot yields significant improvement for all of the genetic markers. The most noticeable improvement comes from the ma marker. This indicates that the most important visual features for the ma marker may only be present in a subset of the plot images.

This significant improvement in classification accuracy for the SVM approach, suggests that it would be beneficial to similarly aggregate features across all of the plot images in the pipeline where we train a fully connected layer on top of the pre-trained embedding. While we cannot use the same average pooling of the embedding features that we employ in this paper, one possible approach for such cross-image aggregation was described in Ren et al. (2021), and presents an interesting direction for future work.

## 5.2. Visualizations of Genetic Markers

In the following sections, we discuss the visualizations produced by the classification models. We focus on the biological relevance of the produced visualizations, as well as a comparison between the visualizations produced by the direct classification model vs. the embedding model.

### 5.2.1. Visualizations From Classification Network

In **Figure 4**, we show 9 of the most activated and correctly predicted reference and alternate images and their corresponding heatmaps for each of the genetic markers (limiting our selection to images that aren't extremely over-saturated or under-exposed). These visualizations provide compelling insights into what the networks have learned to focus on, and therefore what visual plant features are highly correlated with a plant either being reference or alternate for a particular genetic marker. In the following paragraphs, we will discuss notable observations from these visualizations and how they correspond to the phenotypes these markers are known to control. In all visualizations, red regions indicate visual features that are important in leading to the correct classification, while blue regions actively detract from the correct class.

In the d_locus and dw visualizations in **Figure 4**, the alternate visualizations appear to frequently focus on particular panicles at different growth stages (the panicles focused on for the dw and ma genetic markers are earlier in their life cycle when compared to the panicles in the d locus visualizations). This corresponds to the knowledge that polymorphisms in these genetic markers control features like plant growth rate (SNPs in the dw and d_locus families are considered "dwarfing" markers, controlling growth rate and ultimate plant height), flowering time and maturity. The d_locus reference visualizations also appear to focus on particular leaf shapes—the ends of broad leaves—which similarly may relate to the fact that the markers are known to exhibit control over plant structure, and the mid-rib of the leaf. This is consistent with existing knowledge about the phenotype controlled by the d_locus marker as described in Xia et al. (2018): "Dry Stalk (D) locus controls a qualitative difference between juicy green (dd) and dry white (D-) stalks and midribs, and co-localizes with a quantitative trait locus for sugar yield."

In the leaf wax visualizations in **Figure 4**, we see the most confident correct predictions for the leaf wax genetic marker family. Cultivars with the reference version of these SNPs are known to be more waxy, while the alternate versions are less waxy. In the reference heat maps, the important (red) regions are often diffuse, covering much of the leaf, while the alternate visualizations are very focused on the spine of the leaf.

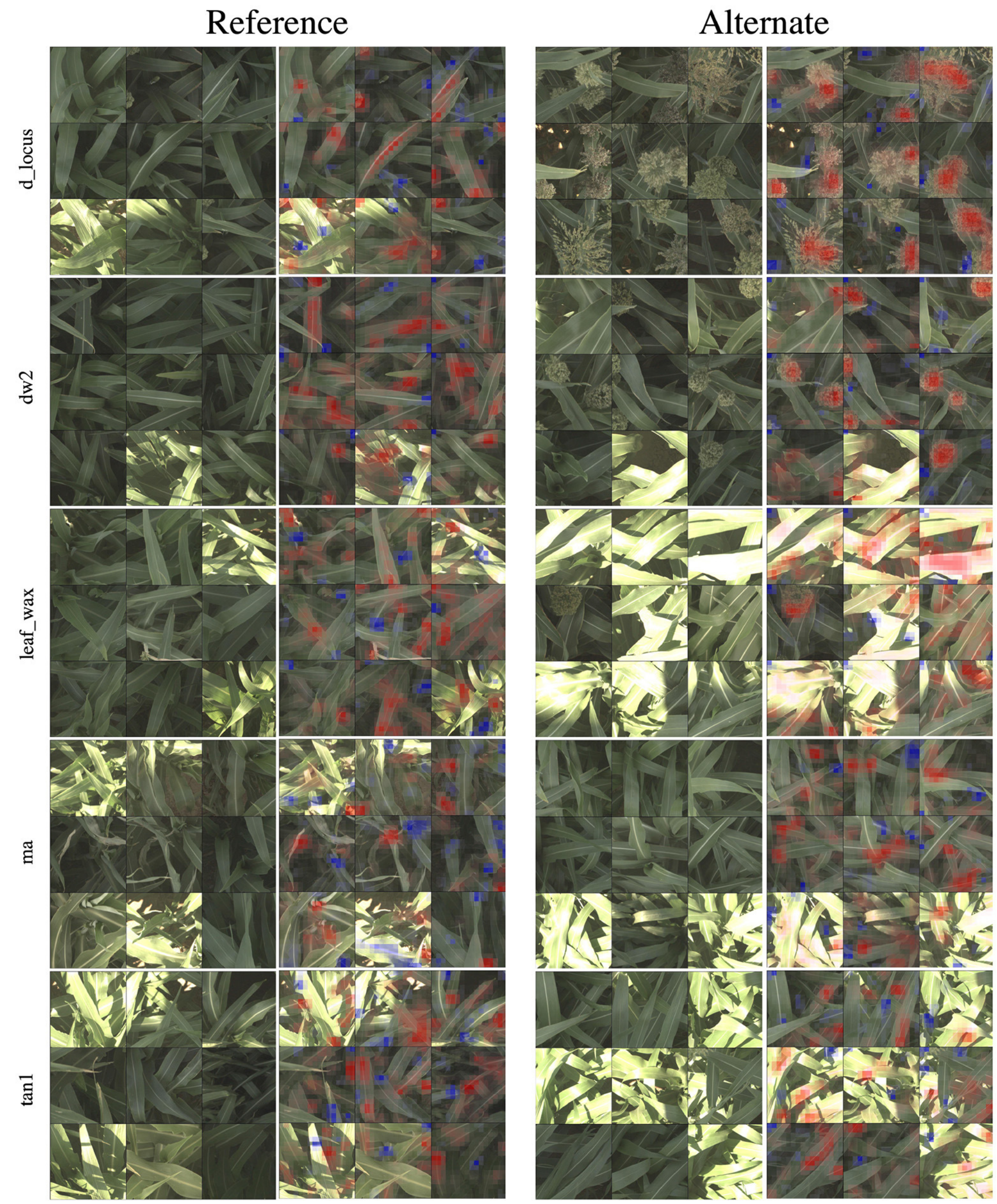


**FIGURE 4 |** For all five families of genetic markers, we can visualize highly activated and correctly classified images from the "reference" and "alternate classes," and their corresponding classification visualization that highlights the features that led to the networks classification. Features highlighted in red are those that led the network to make its correct classification, while features in blue are those which detracted from the correct classification.

We zoom in on a selection of these leaf wax images in **Figure 5**, where it is apparent that in the alternate images, this spine is more brightly differentiated from the rest of the leaf, while in the reference images the spine has less contrast. This corresponds to the wax build up on the leaf in the reference images, which cause the overall leaf to be whiter, resulting in lower contrast on the spine. The reference visualizations also often focus specifically on the interface between the sorghum plant spine and leaf. When reviewing these visualizations with a biologist on our team that does in-field ground truth phenotyping of traits including leaf wax, they said: "That's exactly the place I look at when determining waxiness in the field—it's where the wax is most obvious!" Excitingly, this indicates that the network has learned, without explicit direction, to focus on the same plant parts as expert humans.

In the ma visualizations in **Figure 4**, we see reference heat maps that highlight the ends and edges of leaves that are old, damaged or browning, and the alternate heatmaps show red highlights on the edges of smoother, apparently healthier leaves, which correlates with impact of this particular genetic marker on the growth stage and maturity of the plants, or the "time to maturity" described in Wang et al. (2015) to be controlled in part by the ma genetic markers.

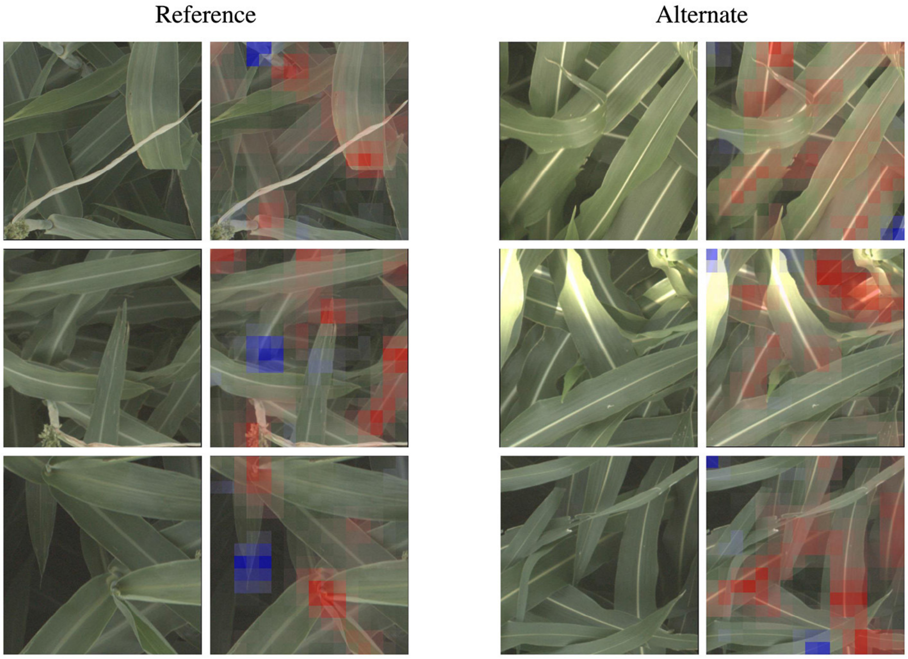


**FIGURE 5 |** The classification network trained on the leaf wax SNPs learned to focus on specific features for the reference and alternate class. When classifying an image as the higher wax content "reference" class, the network often focuses on the interface between the stem and either leaves or panicles, where the wax build up is most high. When classifying an image as "alternate", the network instead often focuses on the vivid mid-vein of the leaf that is more obvious when leaf wax content is lower. These features correspond to phenotypes that field biologists observe in the field. Features highlighted in red are those that led the network to make its correct classification, while features in blue are those which detracted from the correct classification.

### 5.2.2. Visualizations From Embedding Networks

In **Figure 6**, we show the same nine highly activated reference images from **Figure 4**, however this time we show both the visualization produced by the classification model and the visualization produced by the embedding model. While the embedding-based approach achieves higher accuracy, as discussed in Section 5.1, the visualizations are generally less coherent. The classification visualizations often focus on specific and isolated visual features, such as a single panicle or the vein down the center of a leaf.

By comparison, the contributions to the correct prediction highlighted by the embedding visualizations are often much more scattered, highlighting various different visual features simultaneously. The embedding features are trained for the more difficult task of differentiating images of plants in different plots that may look overall quite similar. It is likely that the features learned by the network are good in the aggregate, but individual features may represent combinations of image properties (e.g., "bright midline or wavy leaves or dark shadows") that are more broadly active across the image. The stronger classification results of the embedding features suggests that it is learning more comprehensive visual features; but additional work may be necessary for this improved performance to also include more interpretable visualizations.

In **Figure 7**, we highlight three specific examples for the d_locus marker (reference class) that show this difference in the coherence of the visualizations. The classification visualization clearly focuses on panicles in the first two examples and on the leaf mid-rib in the third; by comparison, the embedding visualization on the other hand highlights various parts of multiple leaves in all three examples. In addition to the classification visualization showing consistent, specific features like the mid-rib and panicles, it highlights a relatively small amount of the image as affecting the classification (either positively or negatively). In contrast, the embedding visualizations shows more overall regions of the image with small amount of impact on the classification.

## 6. CONCLUSIONS AND FUTURE WORK

In this paper, we compare two different pipelines to understand the genotype × phenotype relationship in sorghum. The first pipeline directly creates an image classifier by training on images of cultivars with and without a particular genetic marker, and the second trains an embedding that differentiates a wide variety of cultivars and then uses features in that embedding to predict the presence or absence of genetic markers in images of specific plants. We show the embedding approach has an overall better accuracy on genetic marker prediction tasks.

We also visualize the network by showing activation maps which highlight the most important parts of the images that led to the decision of the network. For several genetic markers, the classification approach leads to maps that seem to give

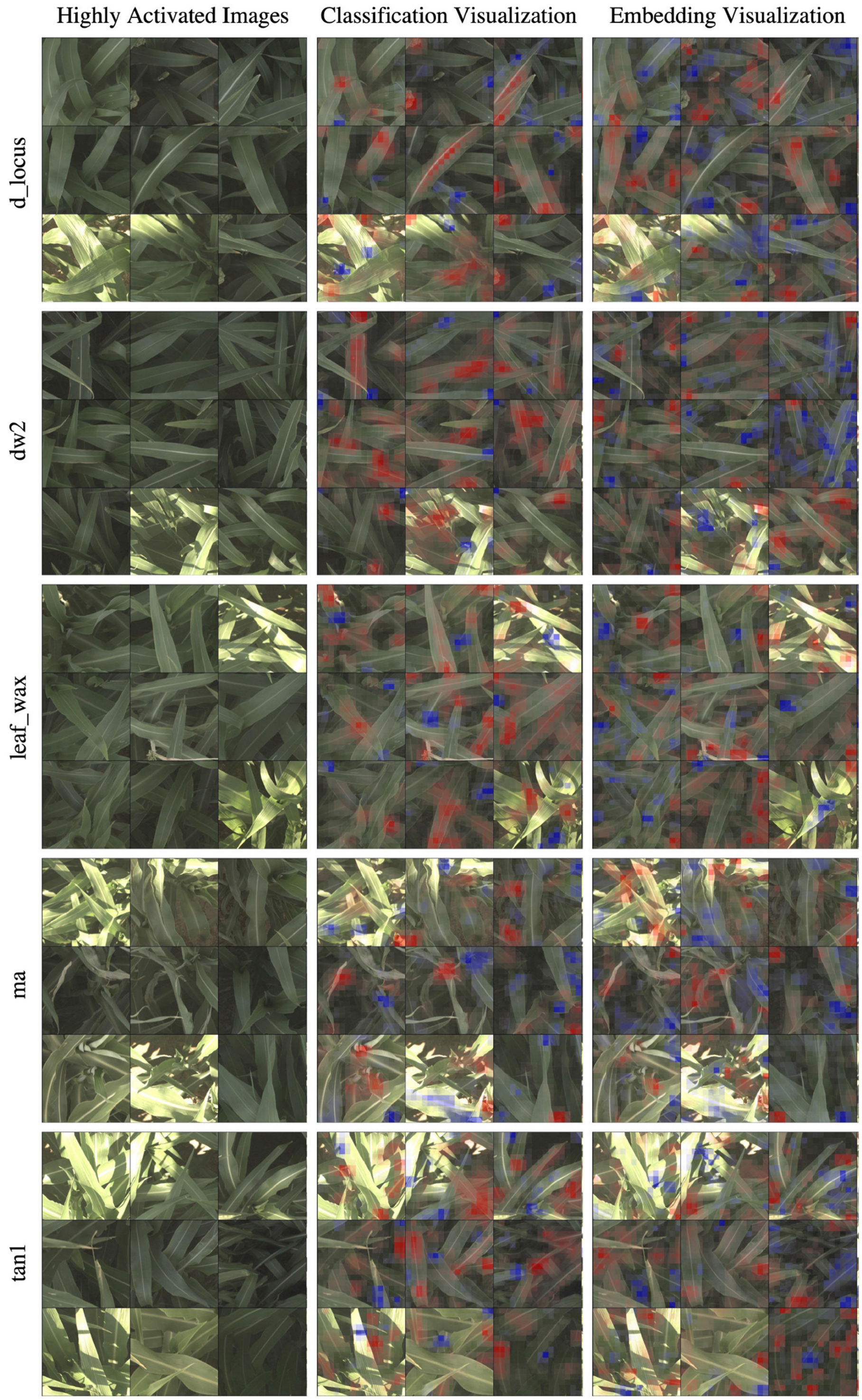


**FIGURE 6 |** In this figure, we compare the "reference" visualizations from the classification and embedding models over all of the markers. Features highlighted in red are those that led the network to make its correct classification, while features in blue are those which detracted from the correct classification. In general, the classification visualizations focus on specific and more readily identifiable features, while the embedding visualization appears to encompass more diverse but less obvious features. Specific examples of this for the d_locus marker are highlighted in **Figure 7**.

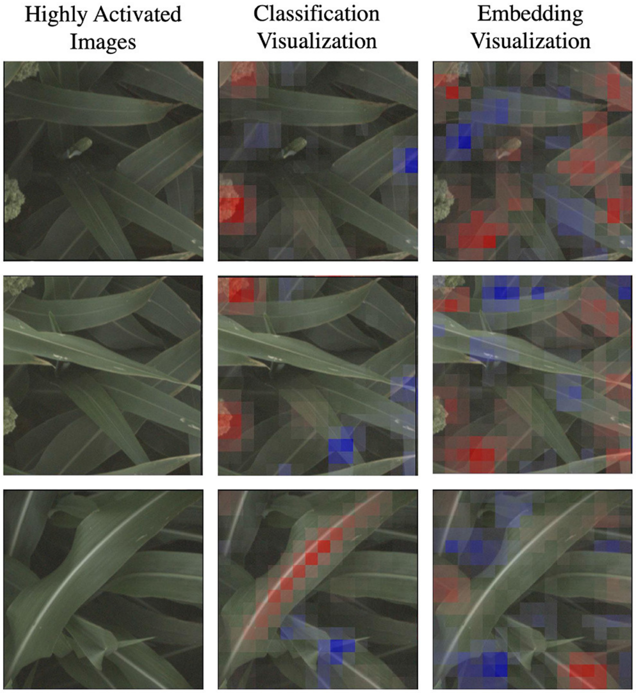


**FIGURE 7 |** Here we focus on a comparison of the classification and embedding visualizations for highly activated reference images for the d_locus genetic marker family. Features highlighted in red are those that led the network to make its correct classification, while features in blue are those which detracted from the correct classification. This classification visualization clearly highlights specific features such as the panicle or the vein down the center of the leaf, while the embedding visualizations are more diffuse, indicating that the model achieves its higher accuracy by learning more varied (but less readily interpretable) features.

clear explanations, as shown, for example, in **Figure 5**. However, the activation maps created in the embedding approach are more complicated. This is because the embedding network learns features to differentiate many different plots instead of features focused entirely on differentiating one genetic marker. Because each feature may contribute to differentiating many different plots, it may represent a mixture of different kinds of image features and therefore be less interpretable. In future work, a finer grain visualization tool like the one proposed by Zhao et al. (2021) may help to understand and explain the visual features that extracted by the embedding network, and loss functions that encourage sparse representation may make those features more interpretable. Additionally, it may be beneficial to consider visualization strategies that do not simply localize the most salient features, but rather try to disentangle their semantic relevance, such as in the Explaining-in-Style approach proposed in Lang et al. (2021).

We demonstrated the feasibility of our pipeline to help understand the genotype × phenotype relationship in sorghum by training deep convolutional neural networks on visual sensor data to predict whether different crops have reference or alternate versions of particular genetic markers. We show for several genetic markers that whose phenotypic expression is well understood that these networks can achieve well-above chance performance on this task, and that visualizations that highlight the most important parts of the images that led to the classification correspond with the known phenotypes.

This approach can be extended to not only help better understand well-established genotype × phenotype relationships, but to explore new, less well understood relationships. The same approach could be deployed for SNPs and families of SNPs whose phenotypic expression is *not* understood, to uncover the importance of new, unstudied polymorphisms. Such discovery would be achieved by first starting with a list of candidate SNPs from sequencing whose phenotypic expression are not well understood; then, for each one, a classifier would be trained to predict whether images show a plant with the reference or alternate version. If a classifier achieves significantly above random-chance performance on this task, then there is some visual feature that is correlated with the marker. The visualizations of the most salient features for the classifier can then be used to determine precisely what the most important plant features are for that genetic marker, to help drive understanding of these as yet unknown genotype × phenotype relationships. We acknowledge that this approach is limited in terms of determining causation as opposed to correlation—there are often substantial correlations between genetic variation in cultivars making it challenging to attribute changes to individual mutations. However, even correlations provide useful evidence for an investigator seeking to better understand the genotype × phenotype relationship. The pre-trained embedding models that achieved high performance in this study could be used in these explorations of new genotype × phenotype relationships, and our pre-trained models and training code are available in our GitHub code repository, which can be found at https://github.com/GWUvision/sorghum-snp-classification. If an investigator is seeking to generalize this pipeline to new species or to sorghum lines and phenotypes that are not present in the BAP, it may be necessary to re-train on representative data.

In this paper, we focused on a relatively limited time period of high resolution data from the TERRA-REF gantry system (data from the entire month of June, mid-way through the growing season in 2017). We recognize that not all phenotypes, however, are observable during this time period. Especially when considering unknown genetic markers, it may be beneficial to consider longer time periods including both early and late growing periods when different phenotypes are expressed. This is a direction for future work: longer time periods may require more complex training protocols that more explicitly incorporate time—for example, using recurrent approaches, or training a multi-headed network that simultaneously predicts the genetic class and the date. Additional work could focus on extending the approach to sensors other than RGB cameras, as some phenotypes may be more readily observed in different sensing modalities, such as hyperspectral or thermal imagery, or in the structural information from the 3D laser scanner.

## DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/GWUvision/sorghum-snp-classification.

## AUTHOR CONTRIBUTIONS

AS, NS, RP, and TM contributed to conception and design of the study. AS and ZZ performed analyses. MP performed literature review. NS and TM provided review of biological relevance of visualizations. AS, RP, and ZZ wrote sections of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

## FUNDING

This work was supported by the Advanced Research Projects Agency-Energy (ARPA-E)/US Department of Energy under grant numbers DE-AR0000594 and DE-AR0001101.

## REFERENCES

Aich, S., and Stavness, I. (2017). "Leaf counting with deep convolutional and deconvolutional networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (CVPRW) (Honolulu, HI: IEEE), 2080–2089.

Anami, B. S., Malvade, N. N., and Palaiah, S. (2020). Deep learning approach for recognition and classification of yield affecting paddy crop stresses using field images. *Artif. Intell. Agric*. 4, 12–20. doi: 10.1016/j.aiia.2020.03.001

Araus, J. L., and Cairns, J. E. (2014). Field high-throughput phenotyping: the new crop breeding frontier. *Trends Plant Sci*. 19, 52–61. doi: 10.1016/j.tplants.2013.09.008

Ashqar, B. A., Abu-Nasser, B. S., and Abu-Naser, S. S. (2019). "Plant seedlings classification using deep learning," in *International Journal of Academic Information Systems Research* (IJAISR) (Bowling Green, KY).

Bargoti, S., and Underwood, J. (2017). "Deep fruit detection in orchards," in *IEEE International Conference on Robotics and Automation (ICRA)* (Singapore: IEEE), 3626–3633.

Barré, P., Stöver, B. C., Müller, K. F., and Steinhage, V. (2017). Leafnet: a computer vision system for automatic plant species identification. *Ecol. Inform*. 40, 50–56. doi: 10.1016/j.ecoinf.2017.05.005

Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. (2017). "Network dissection: quantifying interpretability of deep visual representations," in *Proceedings of the Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 6541–6549.

Bochner, B. R. (2003). New technologies to assess genotype-phenotype relationships. *Nat. Rev. Genet*. 4, 309–314. doi: 10.1038/nrg1046

Boudiaf, M., Rony, J., Ziko, I. M., Granger, E., Pedersoli, M., Piantanida, P., et al. (2020). "A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses," in *Proceedings of the European Conference on Computer Vision*, 548–564.

Boyles, R. E., Brenton, Z. W., and Kresovich, S. (2019). Genetic and genomic resources of sorghum to connect genotype with phenotype in contrasting environments. *Plant J*. 97, 19–39. doi: 10.1111/tpj.14113

Brenton, Z. W., Cooper, E. A., Myers, M. T., Boyles, R. E., Shakoor, N., Zielinski, K. J., et al. (2016). A genomic resource for the development, improvement, and exploitation of sorghum for bioenergy. *Genetics* 204, 21–33. doi: 10.1534/genetics.115.183947

Burnette, M., Kooper, R., Maloney, J. D., Rohde, G. S., Terstriep, J. A., Willis, C., et al. (2018). "TERRA-REF data processing infrastructure," in *Proceedings of the Practice and Experience on Advanced Research Computing*, ed S. Sanielevici (New York, NY: ACM).

Butte, S., Vakanski, A., Duellman, K., Wang, H., and Mirkouei, A. (2021). Potato crop stress identification in aerial images using deep learning-based object detection. *arXiv preprint arXiv:2106.07770*. doi: 10.1002/agj2.20841

Chandel, N. S., Chakraborty, S. K., Rajwade, Y. A., Dubey, K., Tiwari, M. K., and Jat, D. (2021). Identifying crop water stress using deep learning models. *Neural Comput. Appl*. 33, 5353–5367. doi: 10.1007/s00521-020-05325-4

Chen, L., Chen, J., Hajimirsadeghi, H., and Mori, G. (2020). "Adapting grad-cam for embedding networks," in *IEEE Winter Conference on Applications of Computer Vision* (Snowmass, CO: IEEE), 2794–2803.

Chen, Y., Lee, W. S., Gan, H., Peres, N., Fraisse, C., Zhang, Y., et al. (2019). Strawberry yield prediction based on a deep neural network using high-resolution aerial orthoimages. *Remote Sens*. 11, 1584. doi: 10.3390/rs11131584

Cobb, J. N., DeClerck, G., Greenberg, A., Clark, R., and McCouch, S. (2013). Next-generation phenotyping: requirements and strategies for enhancing our understanding of genotype-phenotype relationships and its relevance to crop improvement. *Theor. Appl. Genet*. 126, 867–887. doi: 10.1007/s00122-013-2066-0

Cuevas, H. E., Zhou, C., Tang, H., Khadke, P. P., Das, S., Lin, Y.-R., et al. (2016). The evolution of photoperiod-insensitive flowering in sorghum, a genomic model for panicoid grasses. *Mol. Biol. Evol*. 33, 2417–2428. doi: 10.1093/molbev/msw120

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). "Imagenet: a large-scale hierarchical image database," in *Proceedings of the Conference on Computer Vision and Pattern Recognition* (Miami, FL), 248–255.

Dobrescu, A., Valerio Giuffrida, M., and Tsaftaris, S. A. (2017). "Leveraging multiple datasets for deep leaf counting," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. (Honolulu, HI), 2072–2079.

Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric*. 145, 311–318. doi: 10.1016/j.compag.2018.01.009

Ge, W., Huang, W., Dong, W., Scott, D., and Scott, M. R. (2018). "Deep metric learning with hierarchical triplet loss," in *Proceedings of the European Conference on Computer Vision* (Munich), 269–285.

Giuffrida, M. V., Doerner, P., and Tsaftaris, S. A. (2018). Pheno-deep counter: a unified and versatile deep learning architecture for leaf counting. *Plant J*. 96, 880–890. doi: 10.1111/tpj.14064

Hadsell, R., Chopra, S., and LeCun, Y. (2006). "Dimensionality reduction by learning an invariant mapping," in *Proceedings of the Conference on Computer Vision and Pattern Recognition, Vol. 2* (New York, NY: IEEE), 1735–1742.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV), 770–778.

Heidary-Sharifabad, A., Zarchi, M. S., Emadi, S., and Zarei, G. (2021). An efficient deep learning model for cultivar identification of a pistachio tree. *Br. Food J*. 123, 3592–3609. doi: 10.1108/BFJ-12-2020-1100

Hilley, J. L., Weers, B. D., Truong, S. K., McCormick, R. F., Mattison, A. J., McKinley, B. A., et al. (2017). Sorghum dw2 encodes a protein kinase regulator of stem internode length. *Sci. Rep*. 7, 4616. doi: 10.1038/s41598-017-04609-5

Kim, W., Goyal, B., Chawla, K., Lee, J., and Kwon, K. (2018). "Attention-based ensemble for deep metric learning," in *Proceedings of the European Conference on Computer Vision* (Munich), 736–751.

Kingma, D. P., and Ba, J. (2015). "Adam: a method for stochastic optimization," in *International Conference on Learning Representations*, eds Y. Bengio and Y. LeCun (San Diego, CA).

Koirala, A., Walsh, K., Wang, Z., and McCarthy, C. (2019). Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of "mangoyolo". *Precision Agric*. 20, 1107–1135. doi: 10.1007/s11119-019-09642-0

Lang, O., Gandelsman, Y., Yarom, M., Wald, Y., Elidan, G., Hassidim, A., et al. (2021). Explaining in style: training a gan to explain a classifier in stylespace. *arXiv preprint arXiv:2104.13369*. doi: 10.1109/ICCV48922.2021.00073

LeBauer, D., Burnette, M. A., Demieville, J., Fahlgren, N., French, A. N., Garnett, R., et al. (2020). *TERRA-REF, An Open Reference Data Set From High Resolution Genomics, Phenomics, and Imaging Sensors*. Available online at: https://datadryad.org/stash/dataset/ doi: 10.5061/dryad.4b8gtht99

Lim, M. G., and Chuah, J. H. (2018). "Durian types recognition using deep learning techniques," in *2018 9th IEEE Control and System Graduate Research Colloquium (ICSGRC)* (Shah Alam: IEEE), 183–187.

Liu, Y., Wang, D., He, F., Wang, J., Joshi, T., and Xu, D. (2019). Phenotype prediction and genome-wide association study using deep convolutional neural network of soybean. *Front. Genet*. 10, 1091. doi: 10.3389/fgene.2019.01091

Maimaitijiang, M., Sagan, V., Sidike, P., Hartling, S., Esposito, F., and Fritschi, F. B. (2020). Soybean yield prediction from uav using multimodal data fusion and deep learning. *Remote Sens. Environ*. 237, 111599. doi: 10.1016/j.rse.2019.111599

Miao, C., Guo, A., Thompson, A. M., Yang, J., Ge, Y., and Schnable, J. C. (2021). Automation of leaf counting in maize and sorghum using deep learning. *Plant Phenome J*. 4, e20022. doi: 10.1002/ppj2.20022

Mohanty, S. P., Hughes, D. P., and Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Front Plant Sci*. 7, 1419. doi: 10.3389/fpls.2016.01419

Movshovitz-Attias, Y., Toshev, A., Leung, T. K., Ioffe, S., and Singh, S. (2017). "No fuss distance metric learning using proxies," in *Proceedings of the International Conference on Computer Vision* (Venice).

Mural, R. V., Grzybowski, M., Miao, C., Damke, A., Sapkota, S., Boyles, R. E., et al. (2021). Meta-analysis identifies pleiotropic loci controlling phenotypic trade-offs in sorghum. *Genetics* 218, iyab087. doi: 10.1093/genetics/iyab087

Murphy, R. L., Morishige, D. T., Brady, J. A., Rooney, W. L., Yang, S., Klein, P. E., et al. (2014). Ghd7 (ma6) represses sorghum flowering in long days: Ghd7 alleles enhance biomass accumulation and grain production. *Plant Genome* 7, plantgenome2013.11.0040. doi: 10.3835/plantgenome2013.11.0040

Nevavuori, P., Narra, N., and Lipping, T. (2019). Crop yield prediction with deep convolutional neural networks. *Comput. Electron. Agric*. 163:104859. doi: 10.1016/j.compag.2019.104859

Osako, Y., Yamane, H., Lin, S.-Y., Chen, P.-A., and Tao, R. (2020). Cultivar discrimination of litchi fruit images using deep learning. *Sci. Hortic*. 269:109360. doi: 10.1016/j.scienta.2020.109360

Ren, C., Dulay, J., Rolwes, G., Pauli, D., Shakoor, N., and Stylianou, A. (2021). "Multi-resolution outlier pooling for sorghum classification," in *Agriculture-Vision Workshop in IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (Nashville, TN).

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

Sandhu, K. S., Lozada, D. N., Zhang, Z., Pumphrey, M. O., and Carter, A. H. (2021). Deep learning for predicting complex traits in spring wheat breeding program. *Front. Plant Sci*. 11, 2084. doi: 10.3389/fpls.2020.613325

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). "Facenet: a unified embedding for face recognition and clustering," in *Proceedings of the Conference on Computer Vision and Pattern Recognition* (Boston, MA).

Schweitzer, J. A., Bailey, J. K., Fischer, D. G., LeRoy, C. J., Lonsdorf, E. V., Whitham, T. G., et al. (2008). Plant-soil-microorganism interactions: heritable relationship between plant genotype and associated soil microorganisms. *Ecology* 89, 773–781. doi: 10.1890/07-0337.1

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 International Conference on Computer Vision* (Venice), 618–626.

Sohn, K. (2016). "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems* (Barcelona), 1857–1865.

Song, H. O., Xiang, Y., Jegelka, S., and Savarese, S. (2016). "Deep metric learning via lifted structured feature embedding," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*. (Las Vegas, NV).

Stylianou, A., Souvenir, R., and Pless, R. (2019). "Visualizing deep similarity networks," in *IEEE Winter Conference on Applications of Computer Vision (WACV)* (Waikoloa, HI: IEEE), 2029–2037.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). "On the importance of initialization and momentum in deep learning," in *International Conference on Machine Learning* (Atlanta, GA: PMLR), 1139–1147.

Too, E. C., Yujian, L., Njuki, S., and Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric*. 161, 272–279. doi: 10.1016/j.compag.2018.03.032

Tross, M. C., Gaillard, M., Zwiener, M., Miao, C., Grove, R. J., Li, B., et al. (2021). 3d reconstruction identifies loci linked to variation in angle of individual sorghum leaves. *PeerJ*. 9, e12628. doi: 10.7717/peerj.12628

Ubbens, J., Cieslak, M., Prusinkiewicz, P., Parkin, I., Ebersbach, J., and Stavness, I. (2020). Latent space phenotyping: automatic image-based phenotyping for treatment studies. *Plant Phenomics* 2020, 5801869. doi: 10.34133/2020/5801869

Ubbens, J., Cieslak, M., Prusinkiewicz, P., and Stavness, I. (2018). The use of plant models in deep learning: an application to leaf counting in rosette plants. *Plant Methods* 14, 1–10. doi: 10.1186/s13007-018-0273-z

Uttam, A., Madgula, P., Rao, Y., Tonapi, V., and Madhusudhana, R. (2017). Molecular mapping and candidate gene analysis of a new epicuticular wax locus in sorghum (sorghum bicolor l. moench). *Theor. Appl. Genet*. 130, 2109–2125. doi: 10.1007/s00122-017-2945-x

Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., et al. (2018). "The inaturalist species classification and detection dataset," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*. (Salt Lake City, UT), 8769–8778.

Vo, N., and Hays, J. (2019). "Generalization in metric learning: Should the embedding layer be embedding layer?" in *IEEE Winter Conference on Applications of Computer Vision (WACV)* (Waikoloa, HI: IEEE), 589–598.

Wan, S., and Goudos, S. (2020). Faster r-cnn for multi-class fruit detection using a robotic vision system. *Comput. Netw*. 168, 107036. doi: 10.1016/j.comnet.2019.107036

Wang, A. X., Tran, C., Desai, N., Lobell, D., and Ermon, S. (2018). "Deep transfer learning for crop yield prediction with remote sensing data," in *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*. (San Jose, CA), 1–5.

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

Wang, H., Cimen, E., Singh, N., and Buckler, E. (2020). Deep learning for plant genomics and crop improvement. *Curr. Opin. Plant Biol*. 54, 34–41. doi: 10.1016/j.pbi.2019.12.010

Wang, Y., Tan, L., Fu, Y., Zhu, Z., Liu, F., Sun, C., et al. (2015). Molecular evolution of the sorghum maturity gene ma3. *PLoS ON*. 10, e0124435. doi: 10.1371/journal.pone.0124435

Wu, Y., Li, X., Xiang, W., Zhu, C., Lin, Z., Wu, Y., et al. (2012). Presence of tannins in sorghum grains is conditioned by different natural alleles of tannin1. *Proc. Natl. Acad. Sci. U.S.A*. 109, 10281–10286. doi: 10.1073/pnas.1201700109

Xia, J., Zhao, Y., Burks, P., Pauly, M., and Brown, P. J. (2018). A sorghum nac gene is associated with variation in biomass properties and yield potential. *Plant Direct* 2, e00070. doi: 10.1002/pld3.70

Xuan, H., Souvenir, R., and Pless, R. (2018). "Deep randomized ensembles for metric learning," in *Proceedings of the European Conference on Computer Vision* (Munich).

Yamaguchi, M., Fujimoto, H., Hirano, K., Araki-Nakamura, S., Ohmae-Shinohara, K., Fujii, A., et al. (2016). Sorghum dw1, an agronomically important gene for lodging resistance, encodes a novel protein involved in cell proliferation. *Sci. Rep*. 6, 28366. doi: 10.1038/srep28366

Zeiler, M. D., and Fergus, R. (2014). "Visualizing and understanding convolutional networks," in *Proceedings of the European Conference on Computer Vision* (Zurich: Springer), 818–833.

Zhang, R., Tian, Y., Zhang, J., Dai, S., Hou, X., Wang, J., et al. (2021). Metric learning for image-based flower cultivars identification. *Plant Methods* 17, 1–14. doi: 10.1186/s13007-021-00767-w

Zhao, W., Rao, Y., Wang, Z., Lu, J., and Zhou, J. (2021). "Towards interpretable deep metric learning with structural matching," in *Proceedings of the International Conference on Computer Vision*. (Montreal), 9887–9896.

Zhou, B., Khosla, A., A., L., Oliva, A., and Torralba, A. (2016). "Learning deep features for discriminative localization," in *Proceedings of the Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV).

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2015). "Object detectors emerge in deep scene CNNS," in *International Conference on Learning Representations* (San Diego, CA).

**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.

frontiers | Frontiers in Plant Science

ORIGINAL RESEARCH
published: 08 July 2022
doi: 10.3389/fpls.2022.869524

# MDAM-DRNet: Dual Channel Residual Network With Multi-Directional Attention Mechanism in Strawberry Leaf Diseases Detection

*Tingjing Liao*[1], *Ruoli Yang*[1], *Peirui Zhao*[2]*, *Wenhua Zhou*[2], *Mingfang He*[1] *and Liujun Li*[3]

[1] *College of Computer and Information Engineering, Central South University of Forestry and Technology, Changsha, China,*
[2] *College of Food Science and Engineering, Central South University of Forestry and Technology, Changsha, China,*
[3] *Department of Civil, Missouri University of Science and Technology, University of Missouri-Rolla, Rolla, MO, United States*

**Edited by:**
*Hanno Scharr,*
*Helmholtz Association of German Research Centres (HZ), Germany*

**Reviewed by:**
*Marcin Wozniak,*
*Silesian University of Technology, Poland*
*Talha Ilyas,*
*Jeonbuk National University, South Korea*

***Correspondence:**
*Peirui Zhao*
*znlkd666@163.com*

**Specialty section:**
*This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science*

**Received:** *04 February 2022*
**Accepted:** *10 June 2022*
**Published:** *08 July 2022*

**Citation:**
*Liao T, Yang R, Zhao P, Zhou W, He M and Li L (2022) MDAM-DRNet: Dual Channel Residual Network With Multi-Directional Attention Mechanism in Strawberry Leaf Diseases Detection. Front. Plant Sci. 13:869524. doi: 10.3389/fpls.2022.869524*

The growth of strawberry plants is affected by a variety of strawberry leaf diseases. Yet, due to the complexity of these diseases' spots in terms of color and texture, their manual identification requires much time and energy. Developing a more efficient identification method could be imperative for improving the yield and quality of strawberry crops. To that end, here we proposed a detection framework for strawberry leaf diseases based on a dual-channel residual network with a multi-directional attention mechanism (MDAM-DRNet). (1) In order to fully extract the color features from images of diseased strawberry leaves, this paper constructed a color feature path at the front end of the network. The color feature information in the image was then extracted mainly through a color correlogram. (2) Likewise, to fully extract the texture features from images, a texture feature path at the front end of the network was built; it mainly extracts texture feature information by using an area compensation rotation invariant local binary pattern (ACRI-LBP). (3) To enhance the model's ability to extract detailed features, for the main frame, this paper proposed a multidirectional attention mechanism (MDAM). This MDAM can allocate weights in the horizontal, vertical, and diagonal directions, thereby reducing the loss of feature information. Finally, in order to solve the problems of gradient disappearance in the network, the ELU activation function was used in the main frame. Experiments were then carried out using a database we compiled. According to the results, the highest recognition accuracy by the network used in this paper for six types of strawberry leaf diseases and normal leaves is 95.79%, with an F1 score of 95.77%. This proves the introduced method is effective at detecting strawberry leaf diseases.

**Keywords: detection of strawberry leaf diseases, color feature path, texture feature path, multidirectional attention mechanism, multidirectional attention mechanism dual channel residual network, ELU**

## INTRODUCTION

Strawberry is a sweet and sour delicious fruit prized by consumers that have high nutritional content and commercial value (Skrovankova et al., 2015). Strawberry has since become an important cash fruit crop in China (Lei et al., 2021). With the popularization of greenhouse cultivation technology, strawberries can be harvested year-round, and their cultivation area in China is expanding. However, high temperature and humidity in greenhouses offer favorable conditions for diseases and their outbreaks, leading to infections of strawberry plants that can seriously affect their yield of strawberry fruit (Wang et al., 2015). Because the symptomatic leaf spots of diseased strawberries show complex characteristics in both color and texture, their manual recognition method is time-consuming and laborious (Xiao et al., 2021) and it is thus more likely to miss the best time to intervene with control measures. Therefore, the development of a quick and reliable strawberry disease identification method could help fruit farmers implement timely control measures to reduce losses caused by disease, whose application value could be wide-ranging.

Color and texture are the two main visual attributes used to describe the disease spots that appear on infected plants. In traditional strawberry disease recognition, the types of disease spots are mainly determined manually, according to these two visual attributes. However, manual detection has several drawbacks, namely its slow speed, low accuracy, and large subjective error. In this respect, the field of plant science has advanced vigorously in recent years. Many researchers have proposed disease detection methods that rely instead on machine vision. For example, Kusumandari et al. (2019) proposed a strawberry leaf spot detection method based on color segmentation, for which the results showed a good detection effect. Yet, although this method can distinguish the diseased leaves from the background, this detection becomes impaired when the diseased spots are enlarged or the image quality is not sufficiently high. Robust strawberry disease image recognition inevitably requires fine-grained image classification, with more colors and irregular textures distinguishable. Therefore, during image processing, much color and texture information is apt to get lost, making accurate recognition more difficult. In addition, conventional image processing methods struggle to extract deeper feature information and often are not readily applicable to real environment settings. Recently, Huang et al. (2020) proposed PCNN-IPELM to detect peach diseases, and its detection effect is considered good. However, convolution only uses local information to calculate the target pixel, possibly leading to a loss of information given the lack of global features. Therefore, the key current problems in strawberry leaf disease identification are as follows: (1) the color and texture features of strawberry leaf disease spots are complex, and it is difficult to completely retain their crucial information. (2) It is hard to obtain deeper-level feature information using typical image processing methods, and their practical extension is weak. (3) In the process of image recognition, information loss can arise in the absence of global features.

To solve the problem (1), Kavitha and Suruliandi (2016) used GLCM and a color histogram to respectively extract the texture and color features of the image and then classified the image accordingly. Their experimental results demonstrated the classification effect is stronger when the texture feature is combined with RGB color space. However, GLCM entails abundant calculations, requiring much time. Fekriershad and Tajeripour (2017) had proposed using hybrid color local binary patterns (HCLBP), based on local binary patterns (LBP), to extract color and texture features, reducing the sensitivity of LBP to noise. They introduced an effective point selection algorithm to select the key points of the image and thus reduce the computational complexity; however, some color and texture features were abandoned when selecting such keys.

To solve the problem (2), and thereby extend the model's applicability to automated agricultural systems, Li and Chao (2021) proposed a semi-supervised small sample learning method to identify plant leaf diseases, which outperformed other related methods when less marker training data is available. While adding unlabeled data could improve the accuracy of that model in some cases, it may also render the model worse in other aspects. In the case of plant disease identification, marker data can also be used. Lv et al. (2020) designed DMS-Robust Alexnet, based on the backbone AlexNet structure. Combining extended convolution and multi-scale convolution, improved the feature extraction ability and showed strong robustness when applied to corn disease images collected in a natural setting. Although the extended convolution does increase the receptive field, not all inputs are involved in the calculation because of a gap in the convolution kernel. Zhang et al. (2020) proposed the FCM-NPGA algorithm to segment the image, to retain important texture information while removing noise points and edge points, finding it has high accuracy for detecting defects in apple fruit. But due to the partial loss of color and texture, that model is still limited for extracting key feature information.

To solve the problem (3), Wang et al. (2018) proposed a non-local module, to help the algorithm learn the relationship between different pixel positions, with promising results in the fields of action recognition, image classification, and target detection. This method, however, does not consider the relationship between different regional locations. Chen X. et al. (2020) introduced the channel attention mechanism into the dual-channel residual network and proposed B-ARNet, which can effectively improve the fine-grained classification effect. A drawback to this method is that multi-directional feature sequences are not well accounted for.

Accordingly, to tackle and simultaneously address all three primary problems, this paper proposes a new detection model for strawberry leaf diseases. Based on the ResNeXt network structure, this paper constructs a parallel color feature path and texture feature path at the front end of the network, which can retain the color and texture information in the original image more completely than traditional image processing methods. The two channels converge into a main frame road, to further extract the deep features. In this main road path, MDAM is introduced to improve the network's ability to extract critical features. The model can effectively detect strawberry leaf diseases and has high application value in agricultural automation systems. The main contributions and innovations of this paper are summarized as follows:

(1) The color feature path was constructed by combining the color diagram and ResNeXt structure, enabling the effective extraction and description of the color feature map of a strawberry leaf disease image. It can obtain pivotal preliminary color feature information that then improves the disease detection ability of the network.
(2) The texture feature path was constructed by combining ACRI-LBP and ResNeXt structure, to effectively extract and describe the texture feature map of a strawberry leaf disease image. Effective preliminary texture feature information is obtained, improving the ability of subsequent network detection.
(3) This paper proposed a new attention mechanism—MDAM, used to obtain the weights of the feature layer in the road path of the main frame. The feature layer fused by color feature and texture feature path is inputted into MDAM, and the weights of different feature; information are obtained through a multi-directional comprehensive analysis. This method is helpful for extracting pertinent features, reducing the loss of main features from the strawberry leaf disease image, and improving the adaptability of the model to a complex environment. At the same time, the ELU activation function was used in MDAM-DRNet, adequately inhibiting the disappearance of the gradient.
(4) The recognition rate of seven kinds of strawberry leaf images was 95.79%, and the F1 value was 95.77%. This indicates our model can accurately distinguish among strawberry leaf images displaying similar characteristics. Because of its robust classification performance in a complex natural environment, fruit farmers can use this method to judge whether strawberry leaves are infected with diseases, and to prevent and control strawberry diseases in advance, thereby ensuring the growth of strawberries and mitigating the economic losses caused by strawberry diseases.

## RELATED WORK

In recent years, with the rapid development of machine vision technology, image processing techniques, and machine learning algorithms have been widely incorporated for the detection and classification of leaf diseases (Dhaka et al., 2021). Image processing techniques such as denoising and enhancement are the main methods applied to improve image quality. The use of appropriate image processing methods is conducive to improving recognition accuracy. Many researchers have made outstanding contributions in the field of image processing. Liu et al. (2021) proposed a self-attentional negative feedback network (SRAFBN) capable of achieving a real-time image super-resolution. This model can reconstruct the image texture more in line with human visual perception and has a better image enhancement effect. Chakraborty et al. (2021) proposed an apple leaf disease prediction method based on a multi-class support vector machine. To do this, first, the Otsu threshold algorithm and histogram equalization are used to segment the apple's infected disease area, and then a support vector machine identifies the disease type. Notably, the recognition accuracy achieved was high. To enable the accurate detection of plant diseases, researchers began to use the deep learning method to extract deep-seated features from images of diseased leaves. In this respect, Kundu et al. (2021) proposed a pearl millet disease prediction framework, based on the "internet of things" and interpretable machine learning, which can be used for accurate prediction of pearl millet outbreak and rust disease. Kim et al. (2021) proposed an improved vision-based strawberry disease detection method. Its PlantNet used in this method has a good ability to capture plant domain information. Xie et al. (2020b) proposed a real-time detector of grape leaf disease based on an improved deep-convolution neural network. The detection model Faster DR-IACNN achieved high accuracy when tested against a grape leaf disease data set. Finally, Yang et al. (2022) proposed a strawberry disease classification system that is based on deep learning; it provides a non-destructive, fast, and convenient classification scheme for diseases likely to occur in the process of strawberry planting. However, regarding the above plant disease detection methods, few studies have made full use of the color and texture features of disease spots that appear on leaves. In addition, the existing networks still face hurdles in fine-grained image recognition and their applied use in complex agricultural environments. Therefore, this paper proposes a new detection framework for strawberry leaf diseases that is based on a dual-channel residual network with a multi-directional attention mechanism (MDAM-DRNet). Our experimental results show that this method performs well in the fine-grained classification of strawberry leaf diseases, whose process is depicted in **Figure 1**.

## MATERIALS AND METHODS

### Data Acquisition

To compile the data set used in experiments, online sources and orchard fields were used. The websites included Kaggle and social media, among others, yielding 2,753 photographs. Field images were collected from several strawberry picking gardens: Qingqing Strawberry Garden in Wangcheng District, Changsha City, Hunan Province; Zihui Farm Strawberry Picking Garden in Changsha Economic and Technological Development Zone; and the Shifang strawberry base at Hunan Agricultural University. The camera used was a Canon EOS R6, with an image pixel size of 2,400 × 1,600, and 3,841 strawberry disease images were taken. Because some websites lacked strict disease classification, some classification errors are inevitable. By consulting materials and asking experienced fruit farmers, we eliminated those images with poor quality and unclear objectives and reclassified the pictures having classification errors. Then, the above two data parts were integrated, for a total of 4,362 images, of which 1,106 were of early stages of strawberry leaf diseases. Because a large amount of data is needed for model training, the data was augmented by rotation, flip, random clipping, and brightness transformation tools. In this way, 17,440 images were finally obtained in a database. **Table 1** lists the disease

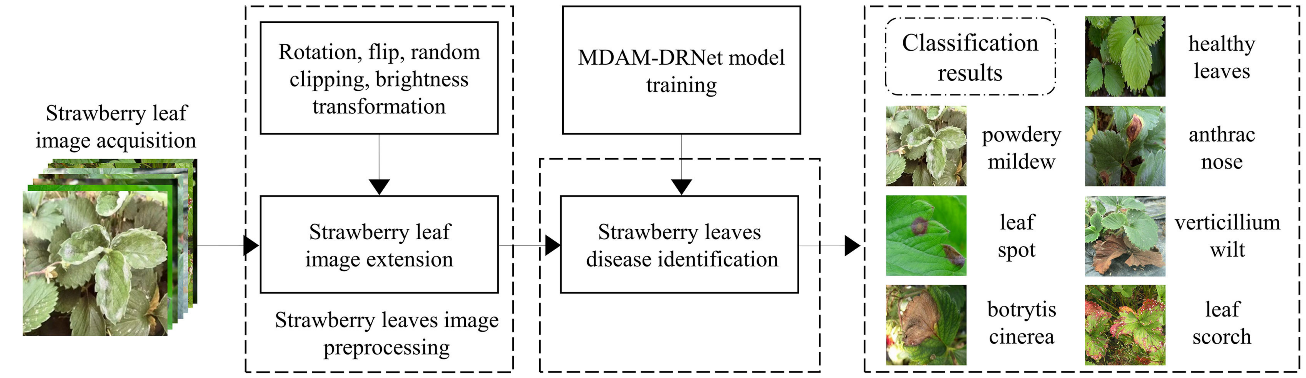
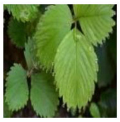

Strawberry leaf image acquisition
Rotation, flip, random clipping, brightness transformation
Strawberry leaf image extension
Strawberry leaves image preprocessing
MDAM-DRNet model training
Strawberry leaves disease identification
Classification results
healthy leaves
powdery mildew
anthrac nose
leaf spot
verticillium wilt
botrytis cinerea
leaf scorch

**FIGURE 1 |** Schematic diagram of strawberry leaves disease identification.

**TABLE 1 |** Quantitative distribution of seven strawberry leaf images.

| Disease category | Example | Number | Proportion (%) |
|---|---|---|---|
| *healthy leaves* |  | 2,591 | 14.86 |
| *powdery mildew* | | 2,503 | 14.35 |
| *leaf spot* | | 2,455 | 14.08 |
| *Botrytis cinerea* | | 2,416 | 13.85 |
| *anthracnose* | | 2,562 | 14.69 |
| *verticillium wilt* | | 2,434 | 13.96 |
| *leaf scorch* | | 2,479 | 14.21 |

categories and corresponding data distribution of strawberry leaves used in this paper, including that for healthy strawberry, strawberry powdery mildew, strawberry leaf spot, strawberry *Botrytis cinerea*, strawberry anthracnose, strawberry verticillium wilt, and strawberry leaf scorch.

Combined with the six different disease images in **Table 1**, the leaf image characteristics of six strawberry diseases were analyzed. The above six diseases can differ starkly in the color and texture of their leaf spot symptoms. Their color characteristics are as follows: (1) Powdery mildew spots are white. (2) Leaf spot is purplish-red in the initial stage, gray in the center, turning purplish brown at the edge after expansion. (3) *Botrytis cinerea* spots appear yellowish-brown. (4) Anthracnose spots are reddish-brown or black in the early stage, brown in the center, and reddish-brown at the edge after expansion. (5) At the initial stage of verticillium wilt disease, its leaf spots are black-brown, but after expansion, they turn yellow-brown between leaf edges and leaf veins, with the new tender leaves appearing grayish-green or light brown. (6) The leaf scorch spot is purple to brown. The texture features of the leaf spots caused by different disease categories are as follows: (1) Powdery mildew is nearly round in the initial stage, whose edge is indistinct after radial expansion. (2) Leaf spot is a small round spot in the initial stage, taking the shape of a snake eye after expansion, and its wheel lines are fine and dense. In severe cases, the disease spots fuse together and the leaf dies. (3) *Botrytis cinerea* spots are large and "V"-shaped, and infected leaves die in severe cases. (4) Anthracnose spots are spindle-shaped with an uneven texture. (5) The initial stage of the verticillium wilt spot manifests a long strip shape in leaves; these wither in severe cases. (6) Leaf scorch leaves shrink, turn brown and inward, and wither with the severity. By comparing their respective color and texture, we can distinguish these six strawberry leaf diseases from healthy strawberry leaves. Therefore, this paper first extracts the texture and color features of strawberry leaves to obtain the shape feature information of a given disease. Next, in the subsequent identification of different strawberry leaf diseases by the neural network, the accuracy of strawberry leaf image classification can be significantly improved.

## Strawberry Leaf Disease Identification Based on the MDAM-DRNet Network

As seen in **Table 1**, the diseases of strawberries are characterized by inconspicuous leaf spots small in area, hindering the manual diagnosis of the disease present and inevitably complicating

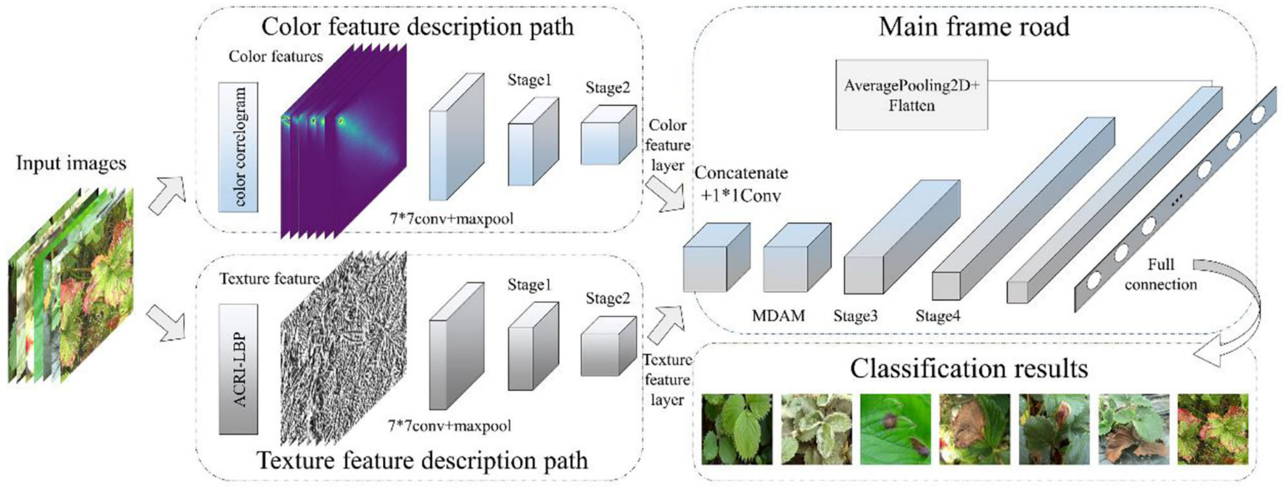


**FIGURE 2 |** Structural diagram of MDAM-DRNet network.

disease identification, making it more likely to overshoot the best period to enact control measures. Therefore, the early monitoring of strawberry leaf diseases has more practical significance; more detailed features of this development stage ought to be extracted by a deep neural network. To solve the above problems, this paper proposes the MDAM-DRNet network, whose overall structure is illustrated in **Figure 2.** Firstly, the input data set passes through the color feature path and texture feature path in parallel. The color feature path includes the color correlogram and stage1 and stage2 of ResNeXt, through which the color feature layer can be extracted. The texture feature path includes ACRI-LBP and stage1 and stage2 of ResNeXt, through which the texture feature layer can be extracted. Then, the two-color feature and texture feature layers are merged into the main frame road path *via* concatenation, after which the MDAM attention mechanism is added to improve the recognition accuracy of a strawberry leaf disease image. The output of MDAM enters stage3 and stage4 of ResNeXt and continues to extract deep-seated feature information. Finally, after extracting the feature information from the network, the types of strawberry leaf disease mapped are categorized using the "softmax" classifier, whose classification results are outputted. The implementation process for the color feature path, texture feature path, and main frame road path is detailed below.

**TABLE 2 |** Pseudo code of color correlation diagram.

**Algorithm 1 color correlogram**

**Input:** Color image *img*, Space distance *d*, Image length *L*, Image width *W*, Number of image channels *N*

**Output:** Color correlogram cgram

```
Begin
  for $x \leftarrow 0$ to $L-1$ do
    for $y \leftarrow 0$ to $W-1$ do
      for $t \leftarrow 0$ to $N-1$ do
        /*Step 1: Take a point as the central pixel and obtain its pixel value*/
        color_i ← Gets the pixel value of the (x, y, t) point
        /*Step 2: Obtain the eight field coordinates of the center point*/
        neighbors ← Obtain the coordinates of 8 field points of (x, y, t)
        for neighbors ← neighbors[0] to neighbors[7] do
          /*Step3: Gets the pixel value of a field*/
          color_j ← Gets the pixel value of the i point
          /*Step 4: Record the number of color pairs (color_i, color_j)*/
          cgram[color_i, color_j] ← cgram[color_i, color_j] + 1
        end for
      end for
    end for
  end for
  return cgram
End
```

## Color Feature Path

The color feature path is composed of the color correlogram and stage1 and stage2 of ResNeXt. Among them, the color correlogram is mainly used to extract and describe the color features in images of strawberry leaf diseases. Therefore, the following mainly introduces the implementation process of color feature extraction as well as describes the color correlogram.

The color correlogram is an expression of image color distribution. This feature not only describes the proportion of pixels of a certain color within the whole image but also reflects the spatial correlation between different color pairings (Jing et al., 1997). Research shows that a color correlation map has higher retrieval efficiency than does a color histogram or color aggregation vector (Wei-Ying and Hong Jiang, 1998). A color correlogram can express the proportion of pixels of a certain color in the whole image and the spatial correlation between pairs of a different color. Because the disease spots on strawberry leaves are small, the local correlation between colors is a more important consideration. Therefore, in order to reduce the space and time requirements, this paper sets the spatial distance D to a fixed value. The specific color extraction steps applied to a strawberry disease leaf image are shown in **Table 2.**

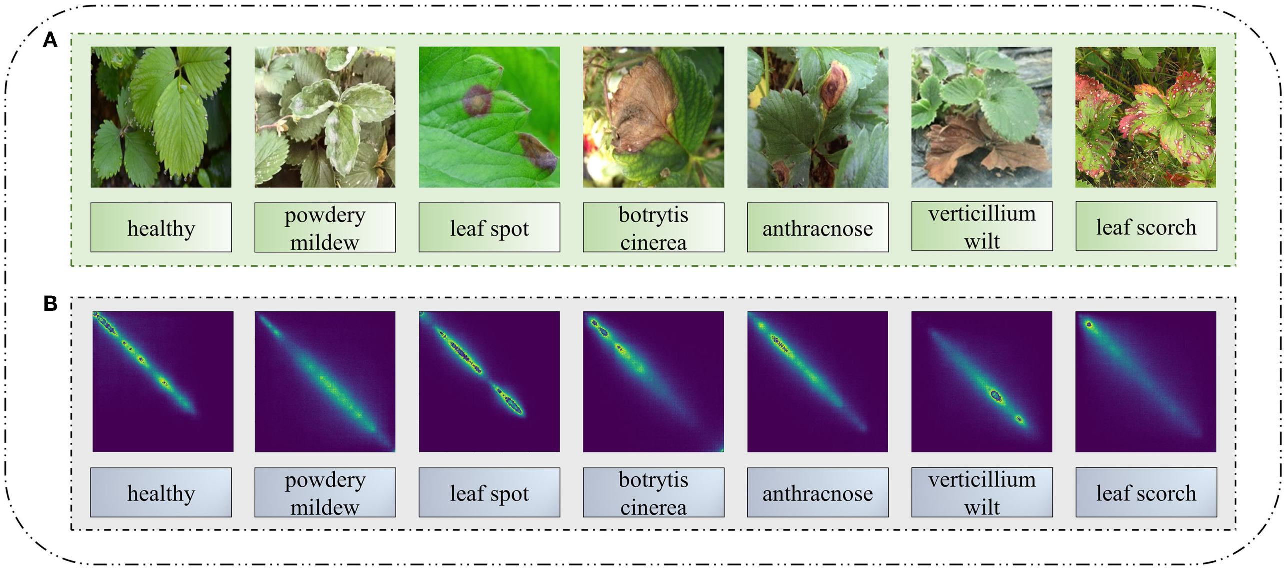


**FIGURE 3 | (A,B)** Color feature map corresponding to five different strawberry leaf disease images.

A color correlogram can be understood as a table indexed by color pairs $\langle x, y \rangle$. Because the color correlogram only considers the local correlation between colors, this method is relatively simple and less computational taxing than extracting all color features of strawberry disease leaf images. Finally, the color features of different kinds of strawberry leaf disease images are extracted by the color correlogram (as seen in **Figure 3**). From the color features in that figure, they evidently differ considerably among the six diseases, indicating high discrimination.

## Texture Feature Path

The texture feature pathway is composed of ACRI-LBP, in addition to stage1 and stage2 of ResNeXt. The ACRI-LBP algorithm is mainly used to describe and extract the texture features from images of strawberry leaf diseases. Therefore, the following focuses on the implementation process of texture feature description and extraction by ACRI-LBP.

LBP is a classical method for describing texture features (Tu et al., 2016). The original LBP operator is defined as taking the center pixel of the window as the threshold in a 3 × 3-window and comparing the gray value of eight adjacent pixels with it. If a surrounding pixel value is greater than the center pixel value, the position of that pixel is marked as 1; otherwise, it is 0. In this way, the eight points in the 3 × 3 neighborhood can be compared, to generate 8-bit binary numbers (usually converted into decimal numbers; i.e., LBP codes, for a total of 256); that is, the LBP value of the window's central pixel is derived, which may be used to reflect the texture information of the region. To resolve the issue arising when the LBP feature coding errs when the scale of the image changes, Guo et al. (2010) proposed the circular LBP (CLBP), which extends the 3 × 3 neighborhood to any neighborhood, by replacing the square neighborhood with a circular one, so as to obtain the LBP Operator with P sampling points in the circular region with a radius R. However, that LBP value will change once the image is rotated. Researchers have extended the LBP Operator to include rotation invariance (Mäenpää and Pietikäinen, 2005). Specifically, by continuously rotating the circular neighborhood the minimum LBP value is obtained, which then serves as the LBP feature of the central pixel. No matter how the image is rotated, the minimum eigenvalue in the field is finally found. For example, an initial LBP value in the circular neighborhood of 225, a series of LBP eigenvalues obtained after image rotation are 240, 120, 60, 30, 15, 135, and 195 respectively. In this group of LBP eigenvalues, if the smallest LBP eigenvalue is 15, the LBP characteristic for the central pixel of that circular neighborhood is 15. Given that the minimum value of the circular field corresponding to each pixel is different and fixed after rotation, the difference between pixels can also be clearly expressed by using the obtained minimum value. Therefore, RI-LBP (rotation invariant LBP) has rotation invariance and high description ability.

Now, considering that the LBP feature value obtained by RI-LBP is the minimum value obtained after the rotation of the circular field, the image may nonetheless be too dark because the feature value is too small, thus obscuring the texture features. Therefore, this paper adds area gray compensation to RI-LBP and proposes ACRI-LBP to extract the texture features. The schematic diagram of ACRI-LBP is presented in **Figure 4.** Specific steps for extracting texture features from a strawberry leaf image by ACRI-LBP are as follows:

(a) Firstly, the color image is transformed into a gray image.
(b) Divide the image into non-overlapping small areas, each 16 × 16 in size. Select the average value of *Agmax* and the maximum value of *Agmin* in the gray area of a pixel, and calculate the minimum value of *Agmax* in the gray area. The regional gray compensation value can then be calculated this way:

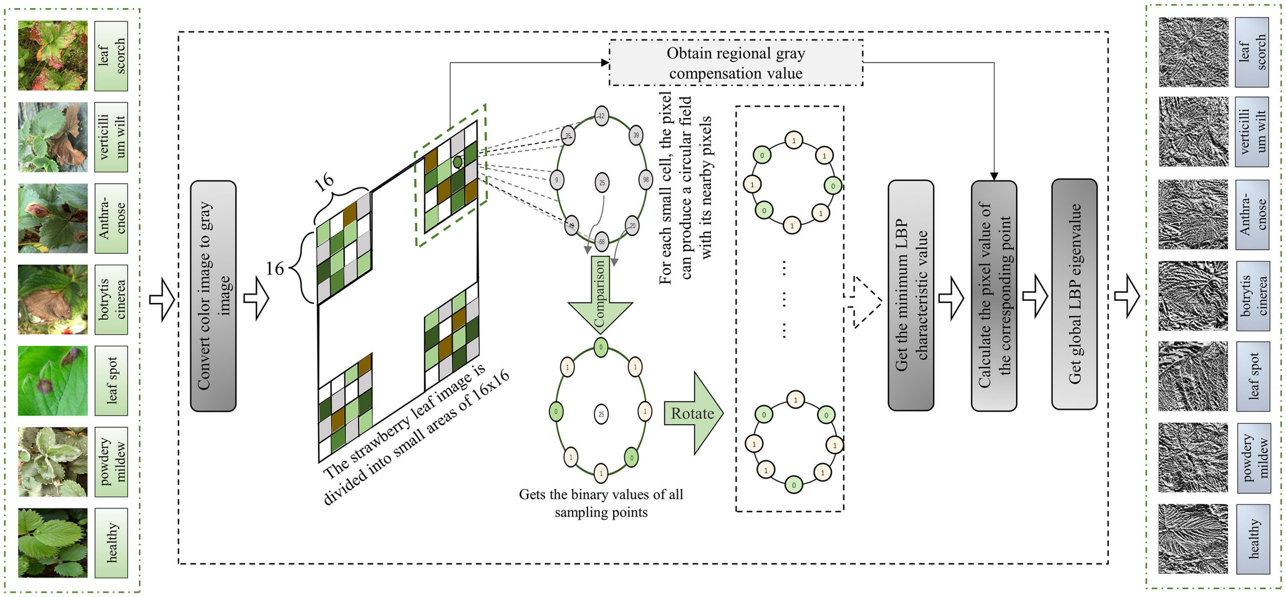

leaf scorch
verticillium wilt
Anthra-cnose
botrytis cinerea
leaf spot
powdery mildew
healthy
Convert color image to gray image
16
16
The strawberry leaf image is divided into small areas of 16x16
Obtain regional gray compensation value
For each small cell, the pixel can produce a circular field with its nearby pixels
Comparison
Gets the binary values of all sampling points
Rotate
Get the minimum LBP characteristic value
Calculate the pixel value of the corresponding point
Get global LBP eigenvalue
leaf scorch
verticillium wilt
Anthra-cnose
botrytis cinerea
leaf spot
powdery mildew
healthy

**FIGURE 4 |** Schematic diagram of ACRI-LBP.

$$\mathrm{Ac} = \frac{\mathrm{Agmin}}{\mathrm{Agmax}}^{*}\mathrm{Ag} \tag{1}$$

(c) Select a pixel point in the region as the center point, whose coordinates are expressed as $(x_c, y_c)$. Taking $(x_c, y_c)$ as the center, draw a circle with radius $R$, select $P$ sampling points with that circular area, and sampling points' coordinates as follows:

$$\mathrm{x_p} = \mathrm{x_c} + \mathrm{Rcos}\left(\frac{2\pi \mathrm{p}}{\mathrm{P}}\right) \tag{2}$$

$$\mathrm{y_p} = \mathrm{y_c} - \mathrm{Rsin}\left(\frac{2\pi \mathrm{p}}{\mathrm{P}}\right) \tag{3}$$

(d) If the coordinates of a sampling point are not at the center of the pixel, the bilinear interpolation method is used to obtain the coordinates of that sampling point. Set the coordinates of the four pixels around the sampling point as $Q_{11} = (x_1, y_1)$, $Q_{12} = (x_1, y_2)$, $Q_{21} = (x_2, y_1)$, $Q_{22} = (x_2, y_2)$, then derive its pixel value this way:

$$\mathrm{f}(\mathrm{x}, \mathrm{y}) = [\mathrm{x_2} - \mathrm{x}\ \mathrm{x} - \mathrm{x_1}] \begin{bmatrix} \mathrm{f}(\mathrm{Q_{11}}) & \mathrm{f}(\mathrm{Q_{12}}) \\ \mathrm{f}(\mathrm{Q_{21}}) & \mathrm{f}(\mathrm{Q_{22}}) \end{bmatrix} \begin{bmatrix} \mathrm{y_2} - \mathrm{y} \\ \mathrm{y} - \mathrm{y_1} \end{bmatrix} \tag{4}$$

(e) Next, compare the gray value of a point in the neighborhood with it. If the surrounding pixel value is greater than the central pixel value, then the position of that point is marked as 1; otherwise, it is marked as 0. In this way, the $P$-point in the neighborhood can generate a $P$-bit binary number after comparison; that is, the LBP value of the central pixel is obtained:

$$\mathrm{LBP}\left(\mathrm{x_c}, \mathrm{y_c}\right) = \sum_{\mathrm{p}=0}^{\mathrm{p}-1} 2^{\mathrm{p}} \mathrm{s}\left(\mathrm{i_p}, \mathrm{i_c}\right) \tag{5}$$

$$\mathrm{s}(\mathrm{x}) = \begin{cases} 1 & \text{if } \mathrm{i_p} \geq \mathrm{i_c} \\ 0 & \text{else} \end{cases} \tag{6}$$

where $i_p$ denotes a pixel value of a neighborhood, and $i_p$ denotes the center pixel value.

(f) Then, the binary values of the left turn bits are recycled, and then the decimal minimum value is taken as the eigenvalue of the current point.

(g) The eigenvalues of each pixel in each region are calculated. The pixel values of the corresponding points can be obtained by adding the eigenvalues of each pixel to the regional gray compensation value. Finally, the texture feature map is obtained by combining each pixel.

Because the radius is the amount actually selected according to the data set, and the smaller the radius, the finer the image texture, and the smaller the number of neighborhoods, the lower the brightness of the image.

## The Main Frame Road Path

The main frame, road path entails the merged texture feature path and color feature path, the MDAM, and the stage3 and stage4 of ResNeX. The output feature layer of the color feature path and that of the texture feature path are fused after entering the main frame road path; hence, the fused output conveys the characteristics of color and texture. However, given the different contribution weights of these two features in the subsequent deep information extraction and disease classification performed by stage3 and stage4 of ResNeXt, we introduce MDAM to assign specific weights to different regions in the fused feature layer. Although stage1 and stage2 of the ResNeXt framework are both distributed in texture feature path and color feature path, stage3 and stage4 for extracting deep information and realizing classification functions are located in the main frame road path. Therefore, the frame of ResNeXt is introduced in the main frame road. To sum up, the following describes the implementation process of the main functions in the main frame road.

### *ResNeXt*

When using a simple neural network for feature extraction, it is easy to lose the main features and thereby alter the classification effect. The deeper the network, the greater the possibility of decomposing the gradient. Residual network (ResNet) (He et al., 2016) can resolve this problem well. However, to improve model accuracy, the traditional ResNet needs to deepen the network. When deepening the network with more super parameters (such as the number of channels, filter size, etc.), the difficulty and computational overhead of network design will increase in tandem. Therefore, this paper uses the ResNeXt structure of Xie et al. (2017) as the basic framework for identifying strawberry leaf diseases. The introduction of ResNeXt not only can retain the residual structure of ResNet to preserve its excellent performance capabilities, but it also improves the recognition accuracy of strawberry leaves without exacerbating parameter complexity, by reducing and minimizing the number of super parameters needed and simplifying the network.

ResNeXt is based on ResNet, but the concept of cardinality is proposed on the structure of ResNet. Each layer of ResNet50 includes two modules: the identity block and the convolution block. The latter can change the network's dimensions but cannot be connected in series continuously, while the former is used to deepen the network and are connectable in a series. With the deepening of the network level, the things learned to become more complex, and more output channels arise. Therefore, while using identity blocks to deepen the network, it is also necessary to use convolution blocks to convert the dimensions, so that those features in the network's front can be transmitted to the feature layer in its back. Compared with previous networks, ResNeXt remains a popular network because of its few parameters, deep layers, and excellent classification ability and recognition effect.

### *MDAM*

The attention mechanism (Luong et al., 2015; Cohn et al., 2016; Tu et al., 2016) originates from the simulation of the visual signal processing mechanism in humans. When people observe and recognize a target, they will focus on its prominent part and ignore some global and background information. This selective attention mechanism is consistent with the characteristics of the identification part in fine-grained image classification. Therefore, in order to extract the features of strawberry leaf images more thoroughly, a new attention mechanism—MDAM, is introduced here into ResNeXt. The MDAM is added after establishing the overall image feature connection layer. The MDAM model performs shallow mining on the overall image features *via* two-layer convolution. First, to each feature, MDAM assigns four weight coefficients (i.e., horizontal weight coefficient, vertical weight coefficient, left diagonal weight coefficient, and right diagonal weight coefficient). Because the attention mechanism in the Graph Attention Network (GAT) algorithm can assign different weights to adjacent nodes, it provides a way to extract deep-seated mutual features in adjacent directions (Velickovic et al., 2017). Therefore, we use the attention mechanism in the GAT algorithm to assign mutual weights to the four directional features. Next, each weighting coefficient is extended, by using the weight multiplication strategy and the maximum matching strategy, and the depth feature is generated through the convolution layer and an average pool. Meanwhile, the relationship between depth features is constructed using the MDAM algorithm (the weight feature map generated by the "softmax" function serves as the adjacency matrix of MDAM). By expanding the relationship between feature weight coefficients and structural features, MDAM can contribute to enhancing the overall classification of the whole strawberry leaf image. This should improve the final effect of fine-grained classification.

The MDAM proposed in this paper has three parts. Its algorithm block diagram appears in **Figure 5**.

Firstly, four primary image weight features *CCCC* are generated in the horizontal, vertical, left diagonal, and right diagonal directions. When a first-order image weight feature is generated, the feature vector set of the vertices of the first layer is $h = [h_1, h_2, ..., h_N]^{\mathrm{T}}$, where $N$ is the number of nodes in the graph (there are only four directions here, so $N = 4$). A weight matrix is needed to obtain the eigenvector of the next layer, so the weight matrix required is $W$. Then the feature vector set of the next layer can be obtained: $h^{'} = [h_1^{'}, h_2^{'}, ..., h_N^{'}]$. For each node, the corresponding attention coefficient can be trained accordingly. The attention coefficient is thus given expressed as $e_{i,j} = a(W^T h_i, W^T h_j)$. Next, the weight assigned by each vertex node $i$ in each direction to node $j$ on the feature sequence is obtained. Finally, the "softmax" function is implemented to regularize the attention coefficient, as shown in formula (7). The features extracted in multiple directions are highly complete, which is more conducive to extracting effective image features. Further, the weight distribution across multiple directions is more conducive to extracting the disease characteristics in different directions.

$$\mathrm{C_i} = \sum\nolimits_{\mathrm{j}=1}^{\mathrm{n}} \frac{\exp(\mathrm{e_{i,j}})}{\sum_{\mathrm{k}=1}^{\mathrm{n}} \exp(\mathrm{e_{i,k}})} \mathrm{h_j} \tag{7}$$

Secondly, the horizontal and vertical weight features are used to obtain the first secondary weight, $C_{21}$, *via* the weight multiplication strategy given by Formula **(8)**. This weight multiplication strategy can mine the deep feature information and expand the weight coefficient by multiplying it with the minimum penalty. The weight multiplication can further amplify the influence of a weight coefficient—for example, the small weight may be 0.1*0.2, while the large weight may be 0.9*0.7– so as to obtain the extended feature. The horizontal and vertical weight features are then used to obtain the second secondary image weight, $C_{22}$, by applying a maximum weight strategy. In the latter's Formula **(9)**, the maximum feature is deemed an effective feature and consistent with the minimum feature $\alpha$ multiple addition, where $\alpha$ is a decimal number between 0 and 1. This method takes the maximum as the main factor and considers another feature to obtain the comprehensive feature. The weight

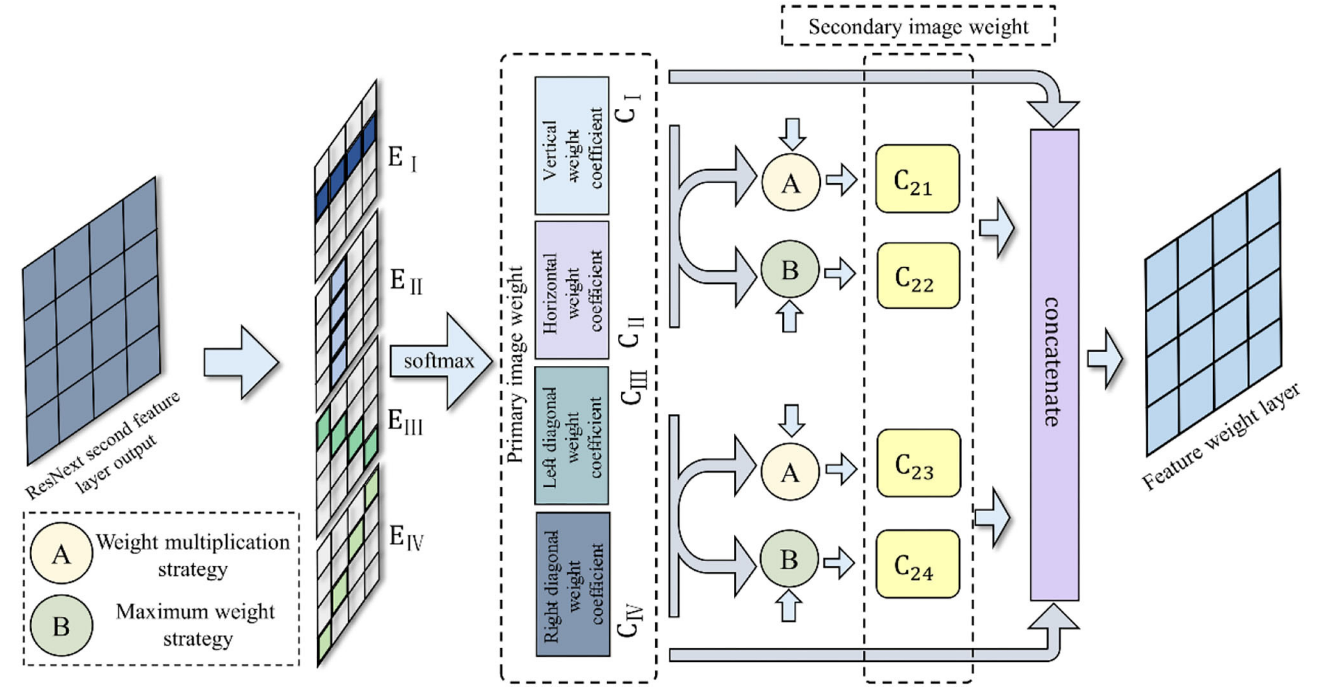

ResNext second feature layer output
E I
E II
E III
E IV
softmax
Primary image weight
Vertical weight coefficient
C I
Horizontal weight coefficient
C II
Left diagonal weight coefficient
C III
Right diagonal weight coefficient
C IV
Secondary image weight
A
B
A
B
C21
C22
C23
C24
concatenate
Feature weight layer
A Weight multiplication strategy
B Maximum weight strategy

**FIGURE 5 |** Algorithm block diagram of MDAM.

features of the left diagonal and right diagonal are multiplied to obtain the third secondary image weight, $C_{23}$. Likewise, the weight features of the left diagonal and right diagonal are also multiplied to obtain the third secondary image weight $C_{24}$. The specific formulae for obtaining $C_{21}, C_{22}, C_{23}, C_{24}$ weight are as follows:

$$C_{21} = C^{*}C - \min(C, C) \tag{8}$$

$$C_{22} = \max(C, C) + \alpha^{*} \min(C, C) \tag{9}$$

$$C_{23} = C^{*}, C - \min(C, C) \tag{10}$$

$$C_{24} = \max(C, C) + \alpha^{*} \min(C, C) \tag{11}$$

Finally, the four types of weight features $C_{21}, C_{22}, C_{23}, C_{24}$ are matched to obtain the maximum value, which is used to supplement the results of four primary image weight coefficients $C, C, C, C$. In MDAM, these eight different image weight coefficients integrate the processed feature information in series through the concatenate function.

$$\mathrm{MDAM} = \mathrm{concatenate}\ ([C, C, C, C, C_{21}, C_{22}, C_{23}, C_{24}]) \tag{12}$$

### *ELU Function*

The output of upper nodes in ResNeXt and the input of lower nodes are connected by a ReLU activation function. Still, some neurons in ReLU may never get activated. Compared with ReLU, the ELU function does not have this “dead” problem, and it can effectively solve the problem of gradient disappearance (Clevert et al., 2015). Therefore, this paper selects the ELU activation function to replace ReLU. The ELU function is expressed this way:

$$f(x) = \begin{cases} x & \text{if } (x > 0) \\ \alpha\,(e^{x} - 1) & \text{otherwise} \end{cases} \tag{13}$$

According to that formula, an output from ELU is maintained even if the input is negative. This ensures the advantages of the ReLU function are inherited while letting the ELU function solve the problem of gradient explosion in the network. Further, because the output mean of ELU is close to zero, its convergence speed is faster than that of the ReLU function. In addition, for the MDAM-DRNet network in this paper, without batch normalization, the ReLU network with > 30 layers will not converge, whereas incorporating the ELU function enables the network to reach high convergence despite more layers.

# RESULTS

## Laboratory Environment

The experimental works were carried out on Windows 10 64-bit operating system equipped with a Core i9-9980xe CPU and NVIDA GeForce RTX 2080ti GPU. The software environment consisted of a CUDA Toolkit (v10.2), CUDNN (v7.6.5), Pycharm (v2019.3), Python (v3.7), and torch (v1.9.1), Numpy (v1.21.4), and OpenCV (v4.5.4.60). The experiments in this paper were all carried out in the same computing environment.

The unified input size of each image is 224*224. During the input process, the data set was expanded by horizontal flipping, small-angle rotation, and scaling, generating a total of 17440

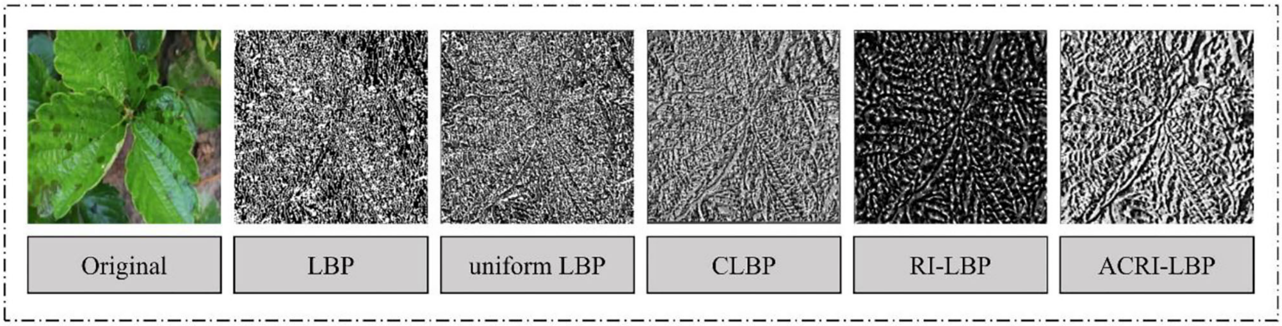


**FIGURE 6 |** Texture feature images extracted by different methods.

images for analysis. This augmented data set was divided into a training set, testing set, and verification set in a ratio of 3:1:1. There are 10,464 images in the training set and 3,488 images in each test set and verification set.

Considering the hardware performance and training effect, the random gradient descent method was used to train the network. To do this, the size of each training and test was set to 24, that is, the batch size is 24, the epoch count is 200, and the momentum parameter is set to 0.9. The model used an Adam optimizer (Kingma and Ba, 2014), because the setting of a learning rate will affect the convergence speed and stability of the model. A callback function was added, the learning rate of the first 60 epochs was set to 0.0001, and the learning rate of the last 60 epochs reduced 10-fold; doing this increased the fitting speed and set the weight falloff to 0.0005.

## Evaluation Indicators

To evaluate the classification effect of the model, we selected accuracy, precision, recall, and F1 score as evaluation indicators. For a single category, the corresponding calculation formulae were as follows:

$$\text{Accuracy} = \frac{\text{TF+TP}}{\text{FP+TN+TP+FN}} \tag{14}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{15}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{16}$$

$$\text{F1} = 2^{*}\frac{\text{Precision}^{*}\text{Recall}}{\text{Precision+Recall}} \tag{17}$$

where TP is the number of strawberry leaf disease samples predicted to be of class A that are actually class A (i.e., positive samples are tested as positive samples). FP is the number of strawberry leaf disease samples that are not predicted as class A but actually are of class A (negative samples are tested as positive samples). FN is the number of samples of strawberry leaf diseases predicted to be of class A yet is not actually class A (if no positive sample is detected, it is designated a positive sample). *Accuracy* corresponds to the proportion of samples correctly classified among all samples attempted; *Precision* is used to measure the number of correctly predicted samples whose predictions were positive; *Recall* is used to measure the number of correct predictions among the real positive samples; the $F1$ score is used to weigh precision and recall in the case of binary classification. For the case of multiple categories, the F1 score must synthesize the calculation results of the evaluation indicators of each category. Such a macro-F1 has the advantage of treating all categories equally (Opitz and Burst, 2019); hence, Macro-F1 was selected as the evaluation index, expressed by $F1_{macro}$, whose calculation formula is as follows:

**TABLE 3 |** Experimental results using different texture extraction methods.

| Radius | Accuracy (%) | Precision$_{\text{macro}}$ (%) | Recall$_{\text{macro}}$ (%) | F1$_{\text{macro}}$ (%) |
|---|---|---|---|---|
| LBP | 85.75 | 85.76 | 85.73 | 85.75 |
| Uniform LBP | 87.90 | 87.93 | 87.91 | 87.92 |
| CLBP | 91.06 | 91.07 | 91.14 | 91.11 |
| RI-LBP | 92.57 | 92.59 | 92.61 | 92.60 |
| ACRI-LBP | 95.79 | 95.76 | 95.79 | 95.77 |

$$\text{Precision}_{\text{macro}} = \frac{\sum_{i=1}^{n}\text{Precision}_{i}}{n} \tag{18}$$

$$\text{Recall}_{\text{macro}} = \frac{\sum_{i=1}^{n}\text{Recall}_{i}}{n} \tag{19}$$

$$\text{F1}_{\text{macro}} = 2^{*}\frac{\text{Precision}_{\text{macro}}^{*}\text{Recall}_{\text{macro}}}{\text{Precision}_{\text{macro}}+\text{Recall}_{\text{macro}}} \tag{20}$$

where $i$ represents class $i$, $Precision_{macro}$ can be regarded as averaging the precision of $i$ categories, and $Recall_{macro}$ can be regarded as averaging the recall of $i$ categories.

# Performance and Analysis

## Comparative Experiment Between ACRI-LBP and Other Texture Extraction Methods

In order to verify the performance of ACRI-LBP, four methods—LBP, CLBP, uniform LBP (Ojala et al., 2002), and RI-LBP—were compared with ACRI-LBP to test the effect of different methods for extracting texture features from the images (**Figure 6**). We can see that the texture features extracted by the CLBP, RI-LBP, and ACRI-LBP methods are relatively clear. To test the influence of different texture feature extraction methods on the network recognition rate, we replaced ACRI-LBP in MDAM-DRNet with LBP, CLBP, uniform LBP, or RI-LBP, and carried out experiments

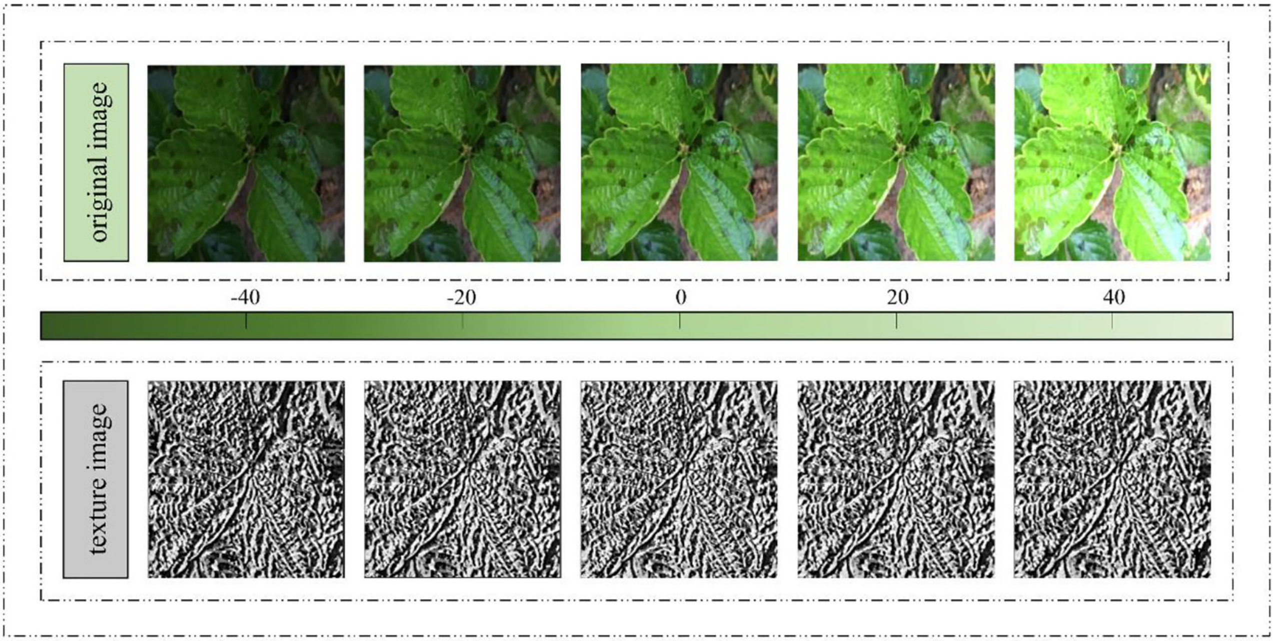


FIGURE 7 | Texture feature map of the same image extracted under different brightness.

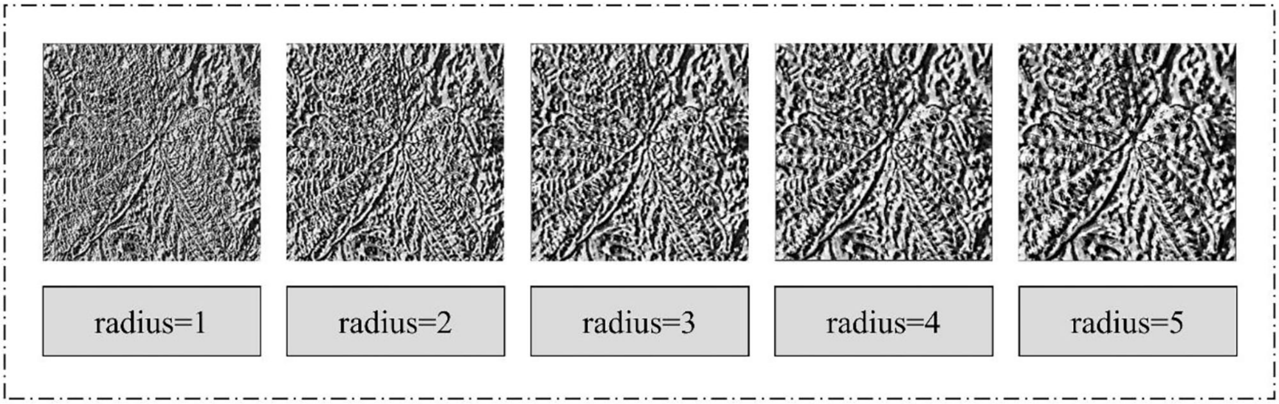


FIGURE 8 | Texture feature map extracted from the same image after selecting different domain radius.

on the same self-made data set. The scores of accuracy, precision, recall, and F1 were the evaluation indicators.

According to these experimental results (**Table 3**), the accuracy attained by the RI-LBP method was 95.79%, and whose F1 score was 95.77%. Compared with that, our ACRI-LBP had an accuracy of 3.22% higher and an F1 score of 3.17% higher. Compared with the original LBP, accuracy increased by 10.04%, and F1 increased by 10.02% when using ACRI-LBP. This proved that ACRI-LBP is effective at improving the accuracy of network recognition and robust method to extract image texture features.

## Verification Experiment for the Illumination Robustness of ACRI-LBP

According to the above description of ACRI-LBP's properties, it only considers the size relationship between the center and adjacent pixel intensity, thus being invariant to a uniform change in whole-image intensity and robust to illumination changes. Therefore, different brightness processing was applied to the same image of a diseased strawberry leaf to obtain five pictures of increasing brightness, from left to right, as shown in **Figure 7.** Then, ACRI-LBP was used to extract their texture features, whose final texture features also appear in **Figure 7**.

**TABLE 4 |** Experimental results with different radius.

| Radius | Accuracy (%) | Precision$_{macro}$ (%) | Recall$_{macro}$ (%) | F1$_{macro}$ (%) |
|---|---|---|---|---|
| 1 | 90.31 | 90.34 | 90.32 | 90.33 |
| 2 | 93.92 | 93.95 | 93.94 | 93.95 |
| 3 | 95.79 | 95.76 | 95.79 | 95.77 |
| 4 | 94.52 | 94.56 | 94.53 | 94.55 |
| 5 | 91.51 | 91.54 | 91.52 | 91.53 |

In that figure, the texture features extracted by ACRI-LBP are consistent across images differing in brightness. This indicated that extraction is not affected by light, and that texture features are clearly extractable even for strawberry leaf images obtained under low light conditions. Therefore, ACRI-LBP's texture extraction can safeguard the utility of the image from low image quality caused by uneven illumination or low brightness. This proved that texture feature extraction *via* ACRI-LBP can effectively improve the subsequent recognition rate of various strawberry leaf disease images.

## Effect of the ACRI-LBP Domain Radius on Image Recognition of Strawberry Leaf Diseases

Changing the domain radius can generate different scale texture features. Accordingly, we selected four different radii (1, 2, 3, and 4) as the domain radius to extract the texture features of a strawberry leaf disease image. When different texture features are inputted into the MDAM-DRNet network, different strawberry leaf disease image recognition rates are obtained. **Figure 8** shows the texture feature map extracted for the same image after applying the different field radii.

Evidently, the feature texture extracted is clearest when the domain radius is 3. The specific reason for this is that the regional feature information associated with other parts cannot be extracted in a domain radius that is too small; conversely, extracting more detailed location feature information is precluded when too large a domain radius is used. To check whether the texture features obtained when the radius is 3 are indeed more effective at improving the image recognition accuracy, we set different radii and conducted experiments on self-made data sets with MDAM-DRNet. These experimental results are shown in **Table 4**.

We see that when the domain radius is set to 3, the recognition accuracy of MDAM-DRNet is 95.79%, and the F1 score is 95.77%, each exceeding that when the radius is set to other values. Therefore, the texture features obtained when the radius is 3 are optimal for enabling the network to extract the key information for strawberry leaf diseases' classification.

## Test Experiment for the Optimal Value of $\alpha$

In equation 4, the $\alpha$ value is multiplied by the minimum characteristic to compensate. To determine the appropriate $\alpha$ value, we set different $\alpha$ values and applied MDAM-DRNet to the strawberry leaf data set we collected. This experiment's results are shown in **Figure 9**.

Evidently, the greatest recognition accuracy was obtained when the x value is 0.3. If the value of $\alpha$ is too small, attention is focused on the global features, and the smaller features go ignored. But if the value of $\alpha$ is too large, attention is shifted to focus on the smaller features, while ignoring the effective features. Both cases will impact the extraction of important feature information by MDAM.

## Experiment for the Recognition Effect of MDAM-DRNet on Early Disease Images of Strawberry Leaves

In order to test the recognition effect of the model on the early disease of strawberry leaves, we screened the images in the self-made data set. A total of 3,768 images of early strawberry leaf diseases were obtained, these were then divided into a training set, test set, and verification set according to a 3:1:1 ratio. There were 2,260 images in the training set and 754 images in each test set and verification set. We tested the recognition effect of ResNeXt and this paper's proposed MDAM-DRNet model for the early incidence of six disease types of strawberry leaves in the data set. According to the experimental results in **Table 5**, the recognition accuracy for early diseases of strawberry leaves is significantly improved when using MDAM-DRNet compared with ReNeXt, by about 9.16%. The recognition accuracy of both models was lower for white spot and brown spot because these two diseases cause dark, small round spots in their early stage, whose color and texture are difficult to distinguish. However, the recognition accuracy of MDAM-DRNet for these two kinds of diseases was still >90%, indicating our proposed model proposed is well able to distinguish similar features. In addition, MDAM-DRNe had a high recall and F1 scores for each category, indicating this new method is adept at recognizing leaf diseases in their early stage of development.

## The Comparative Experiment of MDAM-DRNet and ResNeXt

We next conducted a comparative experiment between MDAM-DRNet and ResNeXt, using the self-made data set, to verify the optimization of MDAM-DRNet relative to ResNeXt. This paper verifies the optimization effect of the network by comparing the evaluation indicator values obtained for MDAM-DRNet and ResNeXt applied to the same strawberry leaf disease data set.

According to the experimental results in **Table 6**, the recognition accuracy of MDAM-DRNet for early-stage diseases of strawberry leaves is significantly improved over ResNeXt, by about 9.67%. The recognition accuracy of MDAM-DRNet for healthy leaves, powdery mildew, leaf spot, *Botrytis cinerea*, anthracnose, verticillium wilt, and leaf scorch was > 95%, and the recognition accuracy of leaf spot and *Botrytis cinerea* was relatively low. This is because the early symptoms of the white spot are not readily apparent, allowing it to be mistaken for another kind of disease, and the early disease images of gray mold accounted for a high proportion, along with texture features that are relatively complex. The recognition accuracy of MDAM-DRNet for each category is at least 93%, and its recall is above 95%, with an F1 score higher than 94%, thus indicating the network has a good classification effect for strawberry leaf diseases. **Figure 10** shows the loss and accuracy curves obtained when the MDAM-DRNet and ResNeXt were trained on the same date set, for six strawberry leaf disease images and one healthy strawberry leaf image.

The experimental results in **Figure 11** show that when the epoch number of the MDAM-DRNet network reaches 75, the accuracy curve converges and flattens, and its highest recognition accuracy exceeds 95%. When the number of iterations of the ResNeXt network reaches 50, the accuracy curve converges and flattens, and its highest recognition accuracy is more than 85%, which this lower than that of the MDAM-DRNet. The convergence speed of MDAM-DRNet was slightly slower than that of ResNeXt, but it significantly improved the accuracy of strawberry leaf disease identification.

## Ablation Experiment

This was done to verify the effect of incorporating the MDAM attention mechanism and dual-channel structure of ACRI-LBP and the color correlogram into the ResNeXt model on the image recognition accuracy of strawberry leaf diseases. Through ablation experiments, we compared the recognition ability of strawberry leaf disease images of the following five models and conducted experiments on the same data set under the same experimental environment.

According to their results in **Table 7**, ResNeXt's strawberry leaf disease recognition accuracy is the lowest among the five models, and the single-use of color or texture features for classification tasks did little to improve accuracy. But after adding MDAM or dual-channel structure to the ResNeXt model, although the number of parameters and training time both increased, overall accuracy is greatly improved. The final superposition effect is more than adequate, having a recognition accuracy of at least 95%, with an F1 score of 95.77%. This proves the modified model is effective for the identification of various diseases afflicting strawberry plants.

## An Experiment Comparing the Recognition Rate With Other Networks

To verify the performance of the MDAM-DRNet model in the current network, the classification performance of the MDAM-DRNet network model was tested vis-à-vis an existing partial supervised model and a semi-supervised model. Among the models selected for this experiment, AlexNet (Krizhevsky et al., 2012), VGG16 (Simonyan and Zisserman, 2014), Efficientnet-B5 (Tan and Le, 2019), and ResNet50, ResNeXt, and DensNet-161 (Huang et al., 2017) are the most widely used supervision models at present. Noisy Student Training (Xie et al., 2020a), Meta Pseudo Labels (Pham et al., 2021), and SimCLRv2 (Chen T. et al., 2020) are advanced semi-supervised models developed in the past two years; likewise, the B-ARNet model, DMS-Robust Alexnet model, and NFNet (Brock et al., 2021) model are advanced supervision models proposed in the last two years. The respective recognition accuracy of the above 13 models for seven kinds of strawberry leaf images (6 strawberry leaf diseases and a control image [healthy strawberry leaf] in the same strawberry leaf disease data set is conveyed in **Table 8**.

Compared with other network models, the MDAM-DRNet model proposed in this paper has higher recognition accuracy for the six diseases, being higher than 93%. Thus, the application value of the MDAM-DRNet model for strawberry leaf disease detection is confirmed. To further evaluate the performance of this model, accuracy, precision, recall, and F1 scores as evaluation indicators were also compared among models: these results are in **Table 9**.

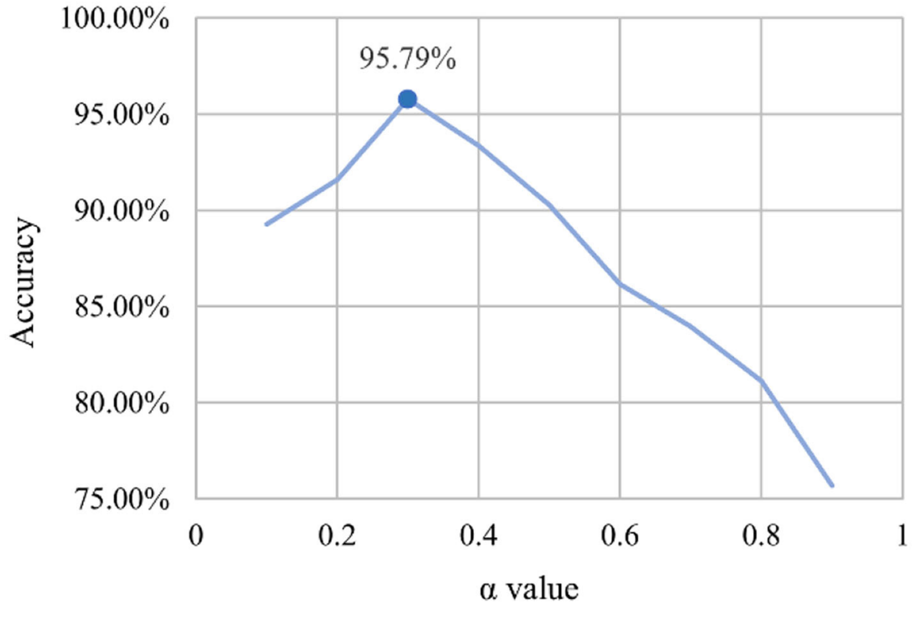


**FIGURE 9 |** Corresponding relationship between $\alpha$ value and accuracy.

**TABLE 5 |** Effect of MDAM-DRNet model on early disease identification of strawberry leaves.

| Methods | Categories | Number of pictures tested | Precision (%) | Recall (%) | F1 (%) | Accuracy (%) |
|---|---|---|---|---|---|---|
| ResNeXt | Powdery mildew | 645 | 83.72 | 80.60 | 82.13 | 83.00 |
| | Leaf spot | 597 | 79.85 | 82.40 | 81.10 | |
| | Botrytis cinerea | 642 | 81.25 | 83.87 | 82.54 | |
| | Anthracnose | 622 | 83.06 | 79.85 | 81.42 | |
| | Verticillium wilt | 573 | 82.61 | 84.82 | 83.70 | |
| | Leaf scorch | 689 | 81.16 | 80.58 | 80.87 | |
| MDAM-DRNet | Powdery mildew | 645 | 93.02 | 90.91 | 91.95 | 92.16 |
| | Leaf spot | 597 | 90.76 | 92.31 | 91.53 | |
| | Botrytis cinerea | 642 | 92.97 | 93.70 | 93.33 | |
| | Anthracnose | 622 | 93.55 | 92.80 | 93.17 | |
| | Verticillium wilt | 573 | 92.17 | 91.38 | 91.77 | |
| | Leaf scorch | 689 | 90.58 | 91.91 | 91.24 | |

**TABLE 6 |** Recognition results of ResNeXt and MDAM-DRNet for strawberry leaf diseases.

| Methods | Categories | Precision (%) | Recall (%) | F1 (%) | Accuracy |
|---|---|---|---|---|---|
| ResNeXt | Healthy leaves | 89.19 | 87.33 | 88.25 | 86.12 |
| | Powdery mildew | 86.23 | 86.75 | 86.49 | |
| | Leaf spot | 84.11 | 85.33 | 84.72 | |
| | Botrytis cinerea | 84.06 | 84.94 | 84.50 | |
| | Anthracnose | 86.52 | 87.38 | 86.95 | |
| | Verticillium wilt | 86.03 | 85.69 | 85.86 | |
| | Leaf scorch | 86.49 | 85.29 | 85.89 | |
| MDAM-DRNet | Healthy leaves | 98.07 | 95.31 | 96.97 | 95.79 |
| | Powdery mildew | 96.01 | 96.78 | 96.39 | |
| | Leaf spot | 94.30 | 95.07 | 94.68 | |
| | Botrytis cinerea | 93.58 | 95.97 | 94.76 | |
| | Anthracnose | 95.90 | 97.04 | 96.46 | |
| | Verticillium wilt | 95.69 | 95.49 | 95.59 | |
| | Leaf scorch | 96.77 | 94.86 | 95.81 | |

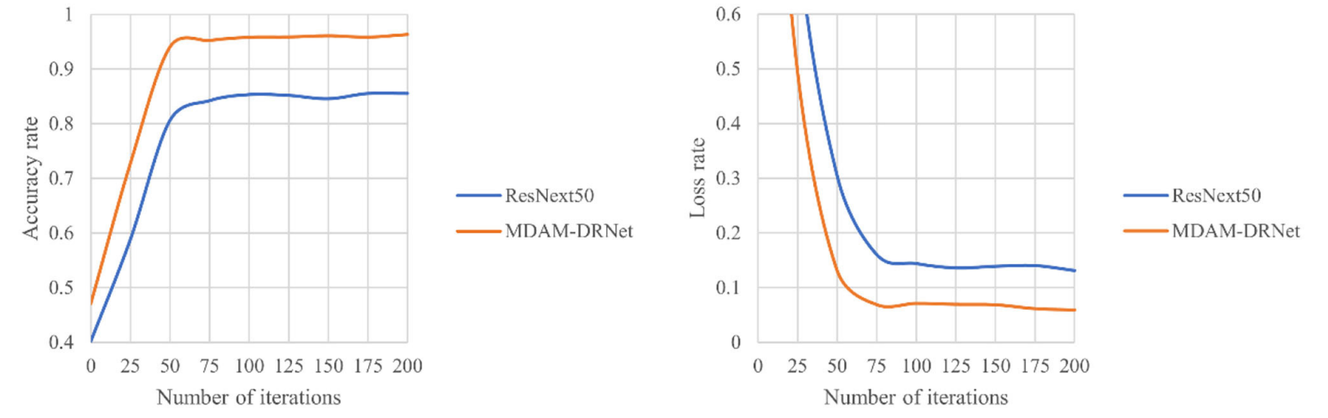


**FIGURE 10 |** The identification accuracy curves of MDAM-DRNet and ResNeXt for strawberry leaf diseases under the same data set.

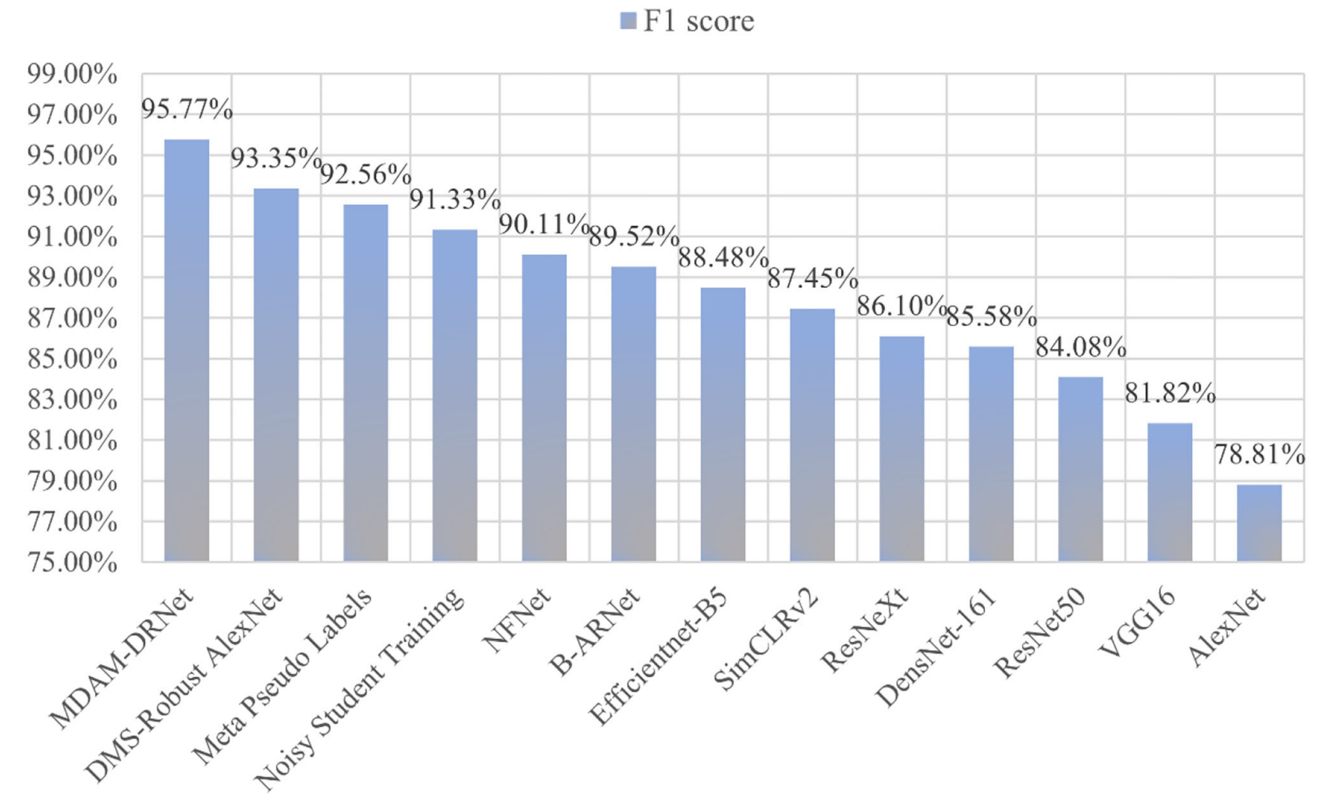


**FIGURE 11 |** F1 score histogram of each method.

**TABLE 7 |** Each model corresponds to the recognition accuracy of each strawberry leaf image.

| Network | Accuracy (%) | Precision$_{macro}$ (%) | Recall$_{macro}$ (%) | F1$_{macro}$ (%) | Parameters | Training time |
|---|---|---|---|---|---|---|
| ResNeXt | 84.09 | 84.06 | 84.09 | 84.08 | 25M | 9 h 8 min |
| ResNeXt+color correlogram | 84.58 | 84.58 | 84.57 | 84.57 | 25M | 9 h 49 min |
| ResNeXt+ACRI-LBP | 83.52 | 83.51 | 83.48 | 83.50 | 25M | 10 h 20 min |
| MDAM-RNet | 90.22 | 90.19 | 90.22 | 90.21 | 28M | 9 h 39 min |
| DRNet | 88.16 | 88.12 | 88.14 | 88.13 | 27M | 11 h 36 min |
| MDAM-DRNet | 95.79 | 95.76 | 95.79 | 95.77 | 30M | 12 h 10 min |

We can see that the accuracy, precision, recall, and F1 score of our proposed MDAM-DRNet proposed, respectively, were 95.79, 95.76, 95.79, and 95.77%, exceeding those of other models. This substantiates the MDAM-DRNet model's excellent recognition ability for strawberry leaf diseases. To more intuitively compare model accuracy, histograms were drawn (**Figure 11**); from these, one can clearly see that the MDAM-DRNet has outstanding recognition accuracy.

Among the evaluation indicators of machine learning, in addition to those listed in **Table 9**, there is also a confusion matrix (also known as a possibility table or error matrix). It is a specific matrix used to visualize the performance of an algorithm, usually one under supervised learning (for unsupervised learning, it is usually called a matching matrix). Each column represents the predicted value and each row represents the actual category. This is very important because, in the actual classification, TP and FP values are the most direct indicators that ultimately determine whether the classification is indeed correct, and the F1 value comprehensively embodies these two critical indicators. As **Figure 12** shows, we calculated the confusion matrix based on the experimental results of the MDAM-DRNet, NFNet, ResNeXt, and AlexNet models.

In this confusion matrix, the values on the diagonal are all the correct prediction results, and the remaining values are the wrong prediction results arising from the model's misjudgment. Each row of the matrix represents the real category, and each column of the matrix represents the prediction label of the model. Evidently, the MDAM-DRNet proposed in this paper has a robust classification effect for strawberry leaf diseases:

**TABLE 8 |** The recognition accuracy of 13 models for 7 categories of the same strawberry leaf disease data set.

| Network | Healthy leaves (%) | Powdery mildew (%) | Leaf spot (%) | *Botrytis cinerea* (%) | Anthracnose (%) | Verticillium wilt (%) | Leaf scorch (%) |
|---|---|---|---|---|---|---|---|
| AlexNet | 81.85 | 76.65 | 80.86 | 79.92 | 77.93 | 76.39 | 78.02 |
| VGG16 | 83.40 | 81.04 | 82.08 | 81.99 | 82.27 | 81.11 | 80.85 |
| ResNet50 | 86.87 | 83.83 | 81.87 | 81.78 | 84.18 | 83.78 | 86.09 |
| ResNeXt | 89.19 | 86.23 | 84.11 | 84.06 | 86.52 | 86.04 | 86.49 |
| DensNet-161 | 88.80 | 85.63 | 83.71 | 83.64 | 85.94 | 85.63 | 85.69 |
| Efficientnet-B5 | 92.28 | 87.62 | 87.78 | 86.96 | 88.87 | 87.68 | 88.10 |
| Noisy Student Training | 94.40 | 91.02 | 89.82 | 91.10 | 91.21 | 91.38 | 90.32 |
| Meta Pseudo Labels | 95.37 | 92.42 | 92.46 | 91.72 | 92.58 | 91.99 | 91.33 |
| SimCLRv2 | 91.12 | 88.62 | 86.35 | 85.51 | 87.70 | 86.45 | 86.29 |
| B-ARNet | 92.66 | 89.62 | 87.37 | 87.99 | 90.63 | 89.53 | 88.71 |
| DMS-Robust Alexnet | 95.75 | 92.81 | 93.08 | 92.34 | 92.38 | 93.22 | 93.75 |
| *NFNet* | 93.24 | 90.42 | 87.78 | 87.78 | 90.23 | 89.12 | 92.14 |
| *MDAM-DRNet* | 98.07 | 96.01 | 94.30 | 93.58 | 95.90 | 95.69 | 96.77 |

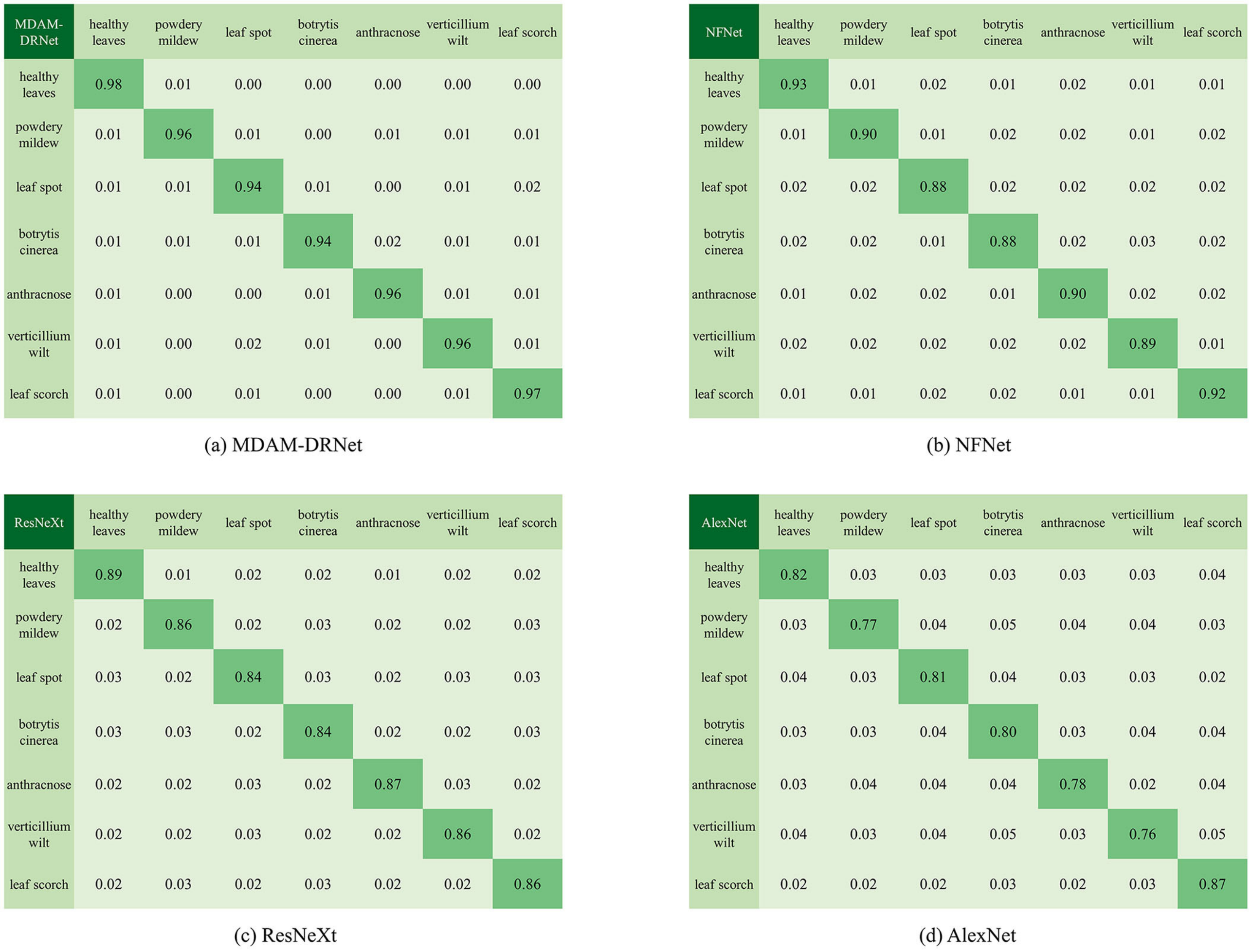
**TABLE 9 |** Test results of 13 models on the same strawberry leaf disease data set.

| Network | Accuracy (%) | Precision$_{macro}$ (%) | Recall$_{macro}$ (%) | F1$_{macro}$ (%) |
|---|---|---|---|---|
| AlexNet | 78.81 | 78.80 | 78.81 | 78.81 |
| VGG16 | 81.82 | 81.81 | 81.83 | 81.82 |
| ResNet50 | 84.09 | 84.06 | 84.09 | 84.08 |
| ResNeXt | 86.12 | 86.09 | 86.10 | 86.10 |
| DensNet-161 | 85.61 | 85.58 | 85.58 | 85.58 |
| Efficientnet-B5 | 88.50 | 88.47 | 88.49 | 88.48 |
| Noisy Student Training | 91.34 | 91.32 | 91.34 | 91.33 |
| Meta Pseudo Labels | 92.57 | 92.55 | 92.56 | 92.56 |
| SimCLRv2 | 87.47 | 87.43 | 87.46 | 87.45 |
| B-ARNet | 89.54 | 89.50 | 89.54 | 89.52 |
| DMS-Robust Alexnet | 93.35 | 93.99 | 93.36 | 93.35 |
| NFNet | 90.14 | 90.10 | 90.12 | 90.11 |
| MDAM-DRNet | 95.79 | 95.76 | 95.79 | 95.77 |

compared with NFNet, ResNeXt, and AlexNet, the number of successful predictions on the diagonal is higher than that attained by other models. Importantly, its performance excelled at detecting/identifying leaf spot diseases with small, cryptic symptoms. Notably, the network framework of MDAM-DRNet was able to correctly classify (more than 94% of cases) two easily confused diseases, leaf spot, and anthracnose. This is because, in its algorithm, the comprehensive weight obtained by MDAM from weights in different directions is soundly aggregated, enabling it to learn the contextual relationship of strawberry leaf diseases, mitigating their similarity to enhance their classification accuracy.

## DISCUSSION

In order to better verify the generalization ability of our MDAM-DRNet model, this paper conducted supplementary experiments on three open data sets of leaf diseases: PlantVillage (Rauf et al., 2019), Citrus (Singh et al., 2020), and PlantDoc (Tan and Le, 2019). Among them, PlantVillage is a multi-category laboratory data set, citrus is a laboratory data set with a small number of categories, and PlantDoc is a multi-category non-laboratory dataset. As neither PlantVillage nor Citrus is already divided into a training set and test set, we divided their data sets into two parts using this training set: a test set ratio of 8:2.



| MDAM-DRNet | healthy leaves | powdery mildew | leaf spot | botrytis cinerea | anthracnose | verticillium wilt | leaf scorch |
|---|---|---|---|---|---|---|---|
| healthy leaves | 0.98 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| powdery mildew | 0.01 | 0.96 | 0.01 | 0.00 | 0.01 | 0.01 | 0.01 |
| leaf spot | 0.01 | 0.01 | 0.94 | 0.01 | 0.00 | 0.01 | 0.02 |
| botrytis cinerea | 0.01 | 0.01 | 0.01 | 0.94 | 0.02 | 0.01 | 0.01 |
| anthracnose | 0.01 | 0.00 | 0.00 | 0.01 | 0.96 | 0.01 | 0.01 |
| verticillium wilt | 0.01 | 0.00 | 0.02 | 0.01 | 0.00 | 0.96 | 0.01 |
| leaf scorch | 0.01 | 0.00 | 0.01 | 0.00 | 0.00 | 0.01 | 0.97 |

(a) MDAM-DRNet

| NFNet | healthy leaves | powdery mildew | leaf spot | botrytis cinerea | anthracnose | verticillium wilt | leaf scorch |
|---|---|---|---|---|---|---|---|
| healthy leaves | 0.93 | 0.01 | 0.02 | 0.01 | 0.02 | 0.01 | 0.01 |
| powdery mildew | 0.01 | 0.90 | 0.01 | 0.02 | 0.02 | 0.01 | 0.02 |
| leaf spot | 0.02 | 0.02 | 0.88 | 0.02 | 0.02 | 0.02 | 0.02 |
| botrytis cinerea | 0.02 | 0.02 | 0.01 | 0.88 | 0.02 | 0.03 | 0.02 |
| anthracnose | 0.01 | 0.02 | 0.02 | 0.01 | 0.90 | 0.02 | 0.02 |
| verticillium wilt | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.89 | 0.01 |
| leaf scorch | 0.01 | 0.01 | 0.02 | 0.02 | 0.01 | 0.01 | 0.92 |

(b) NFNet

| ResNeXt | healthy leaves | powdery mildew | leaf spot | botrytis cinerea | anthracnose | verticillium wilt | leaf scorch |
|---|---|---|---|---|---|---|---|
| healthy leaves | 0.89 | 0.01 | 0.02 | 0.02 | 0.01 | 0.02 | 0.02 |
| powdery mildew | 0.02 | 0.86 | 0.02 | 0.03 | 0.02 | 0.02 | 0.03 |
| leaf spot | 0.03 | 0.02 | 0.84 | 0.03 | 0.02 | 0.03 | 0.03 |
| botrytis cinerea | 0.03 | 0.03 | 0.02 | 0.84 | 0.02 | 0.02 | 0.03 |
| anthracnose | 0.02 | 0.02 | 0.03 | 0.02 | 0.87 | 0.03 | 0.02 |
| verticillium wilt | 0.02 | 0.02 | 0.03 | 0.02 | 0.02 | 0.86 | 0.02 |
| leaf scorch | 0.02 | 0.03 | 0.02 | 0.03 | 0.02 | 0.02 | 0.86 |

(c) ResNeXt

| AlexNet | healthy leaves | powdery mildew | leaf spot | botrytis cinerea | anthracnose | verticillium wilt | leaf scorch |
|---|---|---|---|---|---|---|---|
| healthy leaves | 0.82 | 0.03 | 0.03 | 0.03 | 0.03 | 0.03 | 0.04 |
| powdery mildew | 0.03 | 0.77 | 0.04 | 0.05 | 0.04 | 0.04 | 0.03 |
| leaf spot | 0.04 | 0.03 | 0.81 | 0.04 | 0.03 | 0.03 | 0.02 |
| botrytis cinerea | 0.03 | 0.03 | 0.04 | 0.80 | 0.03 | 0.04 | 0.04 |
| anthracnose | 0.03 | 0.04 | 0.04 | 0.04 | 0.78 | 0.02 | 0.04 |
| verticillium wilt | 0.04 | 0.03 | 0.04 | 0.05 | 0.03 | 0.76 | 0.05 |
| leaf scorch | 0.02 | 0.02 | 0.02 | 0.03 | 0.02 | 0.03 | 0.87 |

(d) AlexNet

**FIGURE 12 |** Confusion matrix corresponding to MDAM-DRNet, B-ARNet, ResNeXt, and CNN models.

**TABLE 10 |** Category of three public data sets, number of training set pictures, and number of test set pictures.

| Dataset | Category | Training | Testing |
|---|---|---|---|
| PlantVillage | 38 | 43,447 | 10,862 |
| Citrus | 5 | 487 | 122 |
| PlantDoc | 27 | 2,334 | 236 |

The categories of these three public data sets, their number of training set images, and their number of test set images are in **Table 10**.

A total of 12 models—AlexNet, VGG16, ResNet50, ResNeXt, DensNet-161, Efficientnet-B5, Noisy Student Training, Meta Pseudo Labels, SimCLRv2, B-ARNet, DMS-Robust Alexnet, and NFNet—were selected and tested against the three public data sets. The experimental results are presented in **Table 11.** According to these, the recognition accuracy of our proposed network on PlantVillage, Citrus, and PlantDoc data sets is 98.04, 98.36, and 90.16% respectively. The test of MDAM-DRNet using the laboratory data sets revealed a good recognition effect, which is equivalent to the recognition accuracy of an advanced network. The recognition accuracy on the non-laboratory data set (PlantDoc) was >90%, exceeding that of the other model networks, indicating that this paper's proposed network is applicable to real-world environments.

**TABLE 11 |** Recognition accuracy of 13 models tested on three public data sets.

| Network | PlantVillage | Citrus | PlantDoc |
|---|---|---|---|
| | Classification accuracy (%) | Classification accuracy (%) | Classification accuracy (%) |
| AlexNet | 96.11 | 97.46 | 68.85 |
| VGG16 | 94.53 | 96.61 | 65.57 |
| ResNet50 | 97.66 | 98.31 | 79.51 |
| ResNeXt | 98.49 | 98.73 | 82.79 |
| DensNet-161 | 95.27 | 97.46 | 72.13 |
| Efficientnet-B5 | 98.40 | 99.15 | 83.61 |
| Noisy Student Training | 91.85 | 93.22 | 82.79 |
| Meta Pseudo Labels | 93.11 | 94.49 | 85.25 |
| SimCLRv2 | 90.59 | 92.37 | 84.43 |
| B-ARNet | 99.03 | 99.15 | 87.70 |
| DMS-Robust Alexnet | 98.18 | 98.73 | 86.89 |
| NFNet | 99.34 | 99.58 | 89.34 |
| MDAM-DRNet | 98.26 | 99.15 | 90.16 |

## CONCLUSION

In tackling the current problem of recognition accuracy of strawberry leaf disease by image recognition models that are not high, leaving it difficult to distinguish the early-stage disease categories, our paper improves the functioning of ResNeXt for this task. The key innovations of the image recognition network MDAM-DRNet designed here for strawberry leaf diseases are as follows:

(1) The color feature path is added to obtain the color features in a strawberry leaf disease image. The color feature path combines the color correlogram and ResNeXt structure to analyze the texture features, which effectively reduces the color interference of other objects in the background and greatly reduces the difficulty of recognition in color extraction.

(2) The texture feature path is added to obtain the texture features in a strawberry leaf image. The texture feature path combines ACRI-LBP and ResNeXt structure to analyze the texture features, enabling deeper feature extraction, effectively filtering out the interference of non-feature texture information, which greatly reduces the difficulty of recognition in texture extraction.

(3) MDAM is introduced into the main frame road path to extract multi-directional attention, which can dynamically weigh the characteristic data of the region of interest from different directions. This improves the attention of the network to the key region and overcomes the identification difficulty caused by the target's small size. Meanwhile, in the main frame, the ELU function is applied to improve the anti-interference ability of the network.

Compared with the traditional ResNeXt model, the newly designed MDAM-DRNet network in this paper strengthens the recognition accuracy of strawberry leaf diseases, and our model's effectiveness is corroborated by a suite of experiments. In this paper, images of strawberry leave in different periods and regions were collected in representative strawberry planting areas in southern China. Through deep learning and comparison of different models, strawberry leaf diseases in their natural environmental settings are identified and detected, for which high accuracy is achievable. Hence, the MDAM-DRNet network in this paper can aid fruit farmers in accurately monitoring the disease situation of leaves in strawberry orchards, for timely control of disease according to its type, by curtailing its spread. In follow-up work, the new model will be tested in real-world agricultural situations, to contribute to the economic production of strawberries and realize its potentially broader benefits for society as soon as possible.

## DATA AVAILABILITY STATEMENT

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

## AUTHOR CONTRIBUTIONS

TL: methodology, writing—original draft preparation, and conceptualization. RY: software, data acquisition, and formal analysis. WZ: model guidance and resources. MH: validation, project administration, funding acquisition, and supervision. LL: visualization and writing—review and editing. All authors have read and agreed to the published version of the manuscript.

## FUNDING

This work was supported by Changsha Municipal Natural Science Foundation (Grant No. kq2014160); in part by the Natural Science Foundation of Hunan Province (Grant No. 2021JJ41087); in part by Hunan Key Laboratory of Intelligent Logistics Technology (Grant No. 2019TP1015).

## ACKNOWLEDGMENTS

We are grateful to all members of the Forestry Information Research Centre for their advice and assistance in the course of this research.

## REFERENCES

Brock, A., De, S., Smith, S. L., and Simonyan, K. (2021). "High-performance large-scale image recognition without normalization", in *International Conference on Machine Learning*: *PMLR* PMLR (New York, NY: PMLR), 1059–1071.

Chakraborty, S., and Paul, S., and Rahat-uz-Zaman, M. (2021). "Prediction of Apple Leaf Diseases Using Multiclass Support Vector Machine", in *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)* (Dhaka: IEEE), 147–151.

Chen, X., Zhou, G., Chen, A., Yi, J., Zhang, W., Hu, Y., et al. (2020). Identification of tomato leaf diseases based on combination of ABCK-BWTR and B-ARNet. *Comput. Electron. Agricult.* 178, 105730. doi: 10.1016/j.compag.2020.105730

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. (2020). Big self-supervised models are strong semi-supervised learners. *Adv. Neural Inf. Process. Syst.* 33, 22243–22255. doi: 10.48550/ARXIV.2006.10029

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2015). Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *arXiv [Preprint]*. arXiv:1511.07289.

Cohn, T., Hoang, C. D. V., Vymolova, E., Yao, K., Dyer, C., Haffari, G., et al. (2016). *Incorporating Structural Alignment Biases into an Attentional Neural Translation Model*. Lisbon, Portugal: Association for Computational Linguistics, 876–885.

Dhaka, V. S., Meena, S. V., Rani, G., and Sinwar, D. Kavita, Ijaz, M.F., et al. (2021). A Survey of Deep Convolutional Neural Networks Applied for Prediction of Plant Leaf Diseases. *Sensors* 21, 4749. doi: 10.3390/s21144749

Fekriershad, S., and Tajeripour, F. (2017). Color texture classification based on proposed impulse-noise resistant color local binary patterns and significant points selection algorithm. *Sens. Rev.* 37, 33–42. doi: 10.1108/SR-07-2016-0120

Guo, Z., Zhang, L., and Zhang, D. (2010). A completed modeling of local binary pattern operator for texture classification. *IEEE Trans. Image Process.* 19, 1657–1663. doi: 10.1109/TIP.2010.2044957

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

Huang, G., Liu, Z., Maaten, L. V. D., and Weinberger, K. Q. (2017). "Densely connected convolutional networks", in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI: IEEE), 2261–2269.

Huang, S., Zhou, G., He, M., Chen, A., Zhang, W., Hu, Y., et al. (2020). Detection of peach disease image based on asymptotic non-local means and PCNN-IPELM. *IEEE Access*. 8, 136421–136433. doi: 10.1109/ACCESS.2020.3011685

Jing, H., Kumar, S. R., Mitra, M., Wei-Jing, Z., and Zabih, R. (1997). "Image indexing using color correlograms", in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (San Juan: IEEE), 762–768.

Kavitha, J. C., and Suruliandi, A. (2016). "Texture and color feature extraction for classification of melanoma using SVM", in *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)* (Kovilpatti: IEEE), 1–6.

Kim, B. Han, Y-. K., Park, J-. H., and Lee, J. (2021). Improved Vision-Based Detection of Strawberry Diseases Using a Deep Neural Network. *Front. Plant Sci.* 11, 559172. doi: 10.3389/fpls.2020.559172

Kingma, D. P., and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv preprint arXiv:*1412, 6980.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 25. 60, 84–90. doi: 10.1145/3065386

Kundu, N., Rani, G., Dhaka, V. S., Gupta, K., Nayak, S. C., Verma, S., et al. (2021). IoT and interpretable machine learning based framework for disease prediction in pearl millet. *Sensors* 21, 5386. doi: 10.3390/s21165386

Kusumandari, D. E., Adzkia, M., Gultom, S. P., Turnip, M., and Turnip, A. (2019). Detection of strawberry plant disease based on leaf spot using color segmentation. *J. Phys. Conf. Ser.* 1230, 012092. doi: 10.1088/1742-6596/1230/1/012092

Lei, J. J., Jiang, S., Ma, R. Y., Xue, L., Zhao, J., Dai, H. P., et al. (2021). *"Current status of strawberry industry in China": International Society for Horticultural Science (ISHS)* (San Juan: ISHS), 349–352.

Li, Y., and Chao, X. (2021). Semi-supervised few-shot learning approach for plant diseases recognition. *Plant Meth.* 17, 68. doi: 10.1186/s13007-021-00770-1

Liu, X., Chen, S., Song, L., Wozniak, M., and Liu, S. (2021). Self-attention negative feedback network for real-time image super-resolution. *J. King Saud Univ. Comput. Inf. Sci.* doi: 10.1016/j.jksuci.2021.07.014

Luong, T., Pham, H., and Manning, C. D. (2015). *Effective Approaches to Attention-Based Neural Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, 1412–1421. doi: 10.18653/v1/D15-1166

Lv, M., Zhou, G., He, M., Chen, A., Zhang, W., Hu, Y., et al. (2020). Maize leaf disease identification based on feature enhancement and DMS-robust alexnet. *IEEE Access*. 8, 57952–57966. doi: 10.1109/ACCESS.2020.2982443

Mäenpää, T., and Pietikäinen, M. (2005). "Texture analysis with local binary patterns", in *Handbook of Pattern Recognition and Computer Vision*. Singapore: World Scientific, 197-216.

Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transac. Pattern Anal. Mac. Intell.* 24, 971–987. doi: 10.1109/TPAMI.2002.1017623

Opitz, J., and Burst, S. (2019). Macro F1 and Macro F1. *arXiv [Preprint]*. arXiv:1911.03347.

Pham, H., Dai, Z., Xie, Q., and Le, Q. V. (2021). "Meta pseudo labels", in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Nashville, TN: IEEE), 11552–11563.

Rauf, H. T., Saleem, B. A., Lali, M. I. U., Khan, M. A., Sharif, M., Bukhari, S. A. C., et al. (2019). A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning. *Data Brief*. 26, 104340. doi: 10.1016/j.dib.2019.104340

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:*1409, 1556.

Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., Batra, N., et al. (2020). "Plantdoc: a dataset for visual plant disease detection," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD* (New York, NY: Association for Computing Machinery), 249–253.

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, 24673–706. doi: 10.3390/ijms161024673

Tan, M., and Le, Q. (2019). "Efficientnet: rethinking model scaling for convolutional neural networks", in *International conference on machine learning: PMLR* (New York, NY: PMLR), 6105-6114.

Tu, Z., Lu, Z., Liu, Y., Liu, X., and Li, H. (2016). *Modeling Coverage for Neural Machine* Translation. *arXiv preprint arXiv:1601.04811.*

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., et al. (2017). Graph attention networks. *Stat*. 1050, 20. doi: 10.48550/ARXIV.1710.10903

Wang, X., Girshick, R., Gupta, A., and He, K. (2018). "Non-local neural networks", in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 7794–7803.

Wang, Z., Cang, T., Qi, P., Zhao, X., Xu, H., Wang, X., et al. (2015). Dissipation of four fungicides on greenhouse strawberries and an assessment of their risks. *Food Control* 55, 215–220. doi: 10.1016/j.foodcont.2015.02.050

Wei-Ying, M., and Hong Jiang, Z. (1998). "Benchmarking of image features for content-based retrieval", in *Conference Record of Thirty-Second Asilomar Conference on Signals, Systems and Computers (Cat. No.98CH3628), vol. 251*, 253–257.

Xiao, J-. R., Chung, P-. C., Wu, H-. Y., Phan, Q-. H., Yeh, J-. L. A., Hou, M. T., et al. (2021). Detection of strawberry diseases using a convolutional neural network. *Plants*. 10, 31. doi: 10.3390/plants10010031

Xie, Q., Luong, M.-T., Hovy, E., and Le, Q.V. (2020a). "Self-training with noisy student improves imagenet classification", in: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (Seattle, WA: IEEE), 10687–10698. doi: 10.1109/CVPR42600.2020.01070

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). "Aggregated residual transformations for deep neural networks", in *Proceedings of the IEEE conference on computer vision and pattern recognition* (Honolulu, HI: IEEE), 1492–1500.

Xie, X., Ma, Y., Liu, B., He, J., Li, S., Wang, H., et al. (2020b). A Deep-Learning-Based Real-Time Detector for Grape Leaf Diseases Using

Improved Convolutional Neural Networks. *Front. Plant Sci.* 11, 751. doi: 10.3389/fpls.2020.00751

Yang, G.-f., Yang, Y., He, Z.-k., Zhang, X.-y., and He, Y. (2022). A rapid, low-cost deep learning system to classify strawberry disease based on cloud service. *J. Integr. Agricult.* 21, 460–473. doi: 10.1016/S2095-3119(21)63604-3

Zhang, W., Hu, J., Zhou, G., and He, M. (2020). Detection of Apple Defects Based on the FCM-NPGA and a Multivariate Image Analysis. *IEEE Access*. 8, 38833–38845. doi: 10.1109/ACCESS.2020.2974262

**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.

ORIGINAL RESEARCH
published: 14 July 2022
doi: 10.3389/fpls.2022.857104

# A Dataset for Forestry Pest Identification

*Bing Liu [1,2], Luyang Liu [1], Ran Zhuo [1], Weidong Chen [1], Rui Duan [1] and Guishen Wang [1]**

*[1] School of Computer Science and Engineering, Changchun University of Technology, Changchun, China, [2] College of Computer Science and Technology, Jilin University, Changchun, China*

The identification of forest pests is of great significance to the prevention and control of the forest pests' scale. However, existing datasets mainly focus on common objects, which limits the application of deep learning techniques in specific fields (such as agriculture). In this paper, we collected images of forestry pests and constructed a dataset for forestry pest identification, called Forestry Pest Dataset. The Forestry Pest Dataset contains 31 categories of pests and their different forms. We conduct several mainstream object detection experiments on this dataset. The experimental results show that the dataset achieves good performance on various models. We hope that our Forestry Pest Dataset will help researchers in the field of pest control and pest detection in the future.

**Keywords: forestry pest identification, deep learning, forestry pest dataset, object detection, transformer**

OPEN ACCESS

***Edited by:***
*Hanno Scharr,*
*Helmholtz Association of German Research Centres (HZ), Germany*

***Reviewed by:***
*Gilbert Yong San Lim,*
*SingHealth, Singapore*
*Sai Vikas Desai,*
*Indian Institute of Technology Hyderabad, India*

****Correspondence:***
*Guishen Wang*
*wangguishen@ccut.edu.cn*

***Specialty section:***
*This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science*

***Received:*** *18 January 2022*
***Accepted:*** *13 June 2022*
***Published:*** *14 July 2022*

***Citation:***
*Liu B, Liu L, Zhuo R, Chen W, Duan R and Wang G (2022) A Dataset for Forestry Pest Identification. Front. Plant Sci. 13:857104. doi: 10.3389/fpls.2022.857104*

## 1. INTRODUCTION

It is well known that the untimely control of pests will cause serious damage and loss of commercial crops (Estruch et al., 1997). In recent years, the scope and extent of forestry pest events in China have increased dramatically, resulting in huge economic losses (Gandhi et al., 2018; FAO, 2020). The identification and detection of pests play a crucial role in agricultural pest control, providing a strong guarantee for crop yield growth and the agricultural economy (Fina et al., 2013). Traditional forestry pest identification relies on a small number of forestry protection workers and insect researchers (Al-Hiary et al., 2011), generally based on the appearance of insects, through manual inspection, visual inspection of insect wings, antennae, mouthparts, feet, etc. to complete the identification of insects, but Due to the wide variety of pests and the small differences between the species, this method has major defects in practice. With the development of machine learning and computer vision technology, automatic pest identification has received more and more attention.

Most of the early pest identification work was done by using a machine learning framework, which consists of two modules: (1) hand-made feature extractors based on GIST (Torralba et al., 2003), Scale-Invariant Feature Transform(SIFT) (Lowe, 2004), and (2) machine learning classifiers, including support vector machine (SVM) (Ahmed et al., 2012) and k-nearest neighbor (KNN) (Li et al., 2009) classifiers. The goodness of the hand-designed feature components will affect the accuracy of the model. If incomplete or incorrect features are extracted from pest images, subsequent classifiers may not be able to distinguish between similar pest species.

With the continuous development of science and technology, deep learning technology has become a research hot spot of artificial intelligence. Image recognition technology based on deep learning improves the efficiency and accuracy of recognition, shortens the recognition time, reduces the workload of staff greatly, and lowers the cost. At present, pest identification methods based on deep learning technology are becoming more and more mature, and the scope of the research includes crops, plants, and fruits (Li and Yang, 2020; Liu and Wang, 2020; Zhu J. et al., 2021). However, the detection of forest pests faces many difficulties due to the lack of effective datasets. Some datasets are too small to meet the detection needs. Furthermore, most pest datasets are collected through traps or controlled laboratory environments, but they lack consideration of the real environment (Sun et al., 2018; Hong et al., 2021). Different species of pests may have a similar appearance. The same species may have different morphologies (nymphs, larvae, and adults) at different times (Wah et al., 2011; Krause et al., 2013; Maji et al., 2013).

For solving the problems mentioned above, we proposed a new forestry pest dataset for the forestry pest identification task. We collected pest data by searching through Google search engine and major forestry control websites. After filtering, we collected 2,278 original pest images covering adults, larvae, nymphs, and eggs of various pests. To alleviate the problem of category imbalance and improve the performance of the dataset for generalization ability, we took data enhancement operations, After data enhancement operations, the total amount of data increased to 7,163. For our pest dataset, we invited three experts in the field to assist us in classifying pests with the help of authoritative websites. Under the premise of knowing the category, we use the LabelImg annotation tool to annotate the image.

Our dataset covers 31 common forestry pests. We collected the forms of pests in different periods in the real wild environment. It meets the basic requirements of forestry pest identification. **Figure 1** shows some examples of the dataset. To explore the application value of our proposed dataset, we use popular object detection algorithms to test the dataset.

The contributions of this work are summarized as follows:

1) We construct a new forestry pest dataset for the target detection task.
2) We tested our dataset on several popular object detection models. The results indicate that the dataset is challenging and creates new research opportunities. We hope this work will help advance future research on related fundamental issues as well as forestry pests identification tasks.

## 2. RELATED WORKS

In this section, we introduce the related work of agricultural pest identification and review the existing data sets.

### Pest Identification of Agriculture

Pest identification helps researchers improve the quality and yield of agricultural products. Earlier pest identification models are mainly based on machine learning techniques. For example, Le-Qing and Zhen (2012) utilizes local average color features and SVM to diagnose 10 insect pests based on a dataset containing 579 samples. Fina et al. (2013) combined K-mean clustering algorithm with adaptive filter for crop pest identification. Zhang et al. (2013) designed a field pest identification system and their dataset comprises approximately 270 training samples. Ebrahimi et al. (2017) used a differential kernel function SVM method for classification and detection, but the evaluated dataset is small, containing just 100 samples. Wang et al. (2018) uses digital morphological features and K-means to segment pest images.
The above traditional pest identification algorithms have been studied with good results, but all of them have limitations, and their detection performance depends on the performance of the pre-designed manual feature extractor and the selected classifier.

Convolutional neural network (CNN) has strong image feature learning capability, such as ResNet (He et al., 2016) and GoogleNet (Szegedy et al., 2015) can learn deep higher-order features from images and can automatically learn shape, color, and texture of complex images and other multi-level features, overcoming the traditional manually designed feature extractors' limitations and subjectivity. It has obvious advantages in target detection, segmentation, classification of complex images, *etc*.

Liu and Wang (2020) constructed a tomato diseases and pests dataset and improved the YOLOV3 algorithm to detect tomato

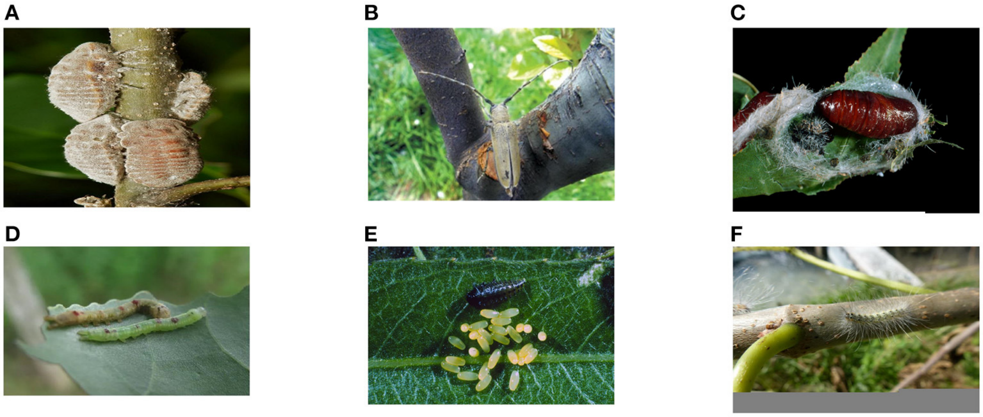


**FIGURE 1 |** Sample images of Forestry Pest Dataset. ***(A)*** Drosicha contrahens; ***(B)*** Apriona germari; ***(C)*** Hyphantria cunea; ***(D)*** Micromelalopha troglodyta(Graeser); ***(E)*** Plagiodera versicolora(Laicharting); and ***(F)*** Hyphantria cunea larvae.

pests. Wang et al. (2020) introduced an attention mechanism in residual networks for improving the recognition accuracy of small targets. A two-stage aphid detector named Coarse-to-Fine Network (CFN) is proposed by Li et al. (2019) to detect aphids with different distributions. Zhu J. et al. (2021) uses super-resolution image enhancement technology and an improved YOLOv3 algorithm to detect black rot on grape leaves.

In general, CNN-based pest identification work can well avoid the limitations of traditional methods and improve the performance of pest identification. However, most target detection models have applied many hand-crafted components.To some extent, the parameters of the manual components increase the workload. To eliminate the impact of manual components on the model, researchers have considered using the versatile and powerful relational modeling capabilities of the transformer to replace the hand-crafted components. Carion et al. (2020) put forward the end-to-end object detection with transformers (DETR) by combining the convolutional neural network and the transformer, which built the first complete end-to-end target detection model and achieved highly competitive performance.

## Related Datasets

At present, deep learning-based agricultural pest identification and classification is maturing. The research scope includes a variety of cash crops such as crops, vegetables, and fruits, and relevant datasets have also been constructed.

Wu et al. (2019) constructed the IP102 pest dataset, which covers more than 70,000 images of 102 common crop pests. Wang et al. (2021) constructed the Agripest field pest dataset, which includes more than 49,700 images of pests in 14 categories. Hong et al. (2020) constructed a moth dataset by pheromone traps, which were labeled with four classes, including three moth classes and an unknown class of non-target insects. As a result of data collection and labeling, a total of 1,142 images were obtained. Liu Z. et al. (2016) constructed a rice pest dataset. The data were collected from image search databases of Google, Naver, and FreshEye, including 12 typical species of paddy field pest insects with a total of over 5,000 images. He et al. (2019) designed an oilseed rape pest image database, including a total of 3,022 images with 12 typical oilseed rape pests. Lim et al. (2018) build an insects dataset by specimens and Internet. The dataset consists of about 29,000 image files for 30 classes. Baidu constructed a forestry pest dataset that includes over 2,000 images for 7 classes through the specimen and traps. Chen et al. (2019) build a garden pests datasets. The dataset consists of about 9,070 image files for 38 classes. Liu et al. (2022) constructed a representative dataset of forest pests classification, including 67 categories and 67,953 original images. However, so far, only the dataset of Liu et al. (2022) is available for the detection of forest pests.

In conclusion, the research on crop diseases and insect pests based on deep learning covers a wide range, but in forestry, the detection and control of forest diseases and insect pests is still a challenge.

# 3. OUR FORESTRY PEST DATASET

## Data Collection and Annotation

We collect and annotate the dataset with following four stages: 1) taxonomic system establishment, 2) image collection, 3) preliminary data filtering, 4) Data Augmentation, and 5) professional data annotation.

### Taxonomic System Establishment

We have established hierarchical classification criteria for the Forestry Pest Dataset. We asked three forestry experts to help us discuss common forest pest species. In addition, to better meet the needs of forest pest control, we use the larvae, eggs, and nymphs of each pest as subclasses, specifically, *Sericinus montela* and *Sericinus montela(larvae)* according to our The standards are divided into two categories. There are 31 classes finally obtained and they present a hierarchical structure as shown in **Figure 2**.

### Image Collection

We utilize the Internet and forestry pest databases as the main sources of dataset images. We use the Chinese and scientific names of pests to search and save on common image search engines and also search for their corresponding eggs, larvae, and other images. Afterward, we searched for corresponding images from specialized agricultural and forestry pest websites.

### Preliminary Data Filtering

From candidate images obtained from various websites and databases, we organized four volunteers to manually screen images. With the assistance of forestry experts, volunteers removed invalid and duplicate images that did not contain pests and repaired damaged images. And establish the initial category information. Specifically, in the initial pest collection work, we collected according to 15 categories, the purpose of this is to enhance the balance of data in the next step. Finally, we obtained 2,278 original images.

### Data Augmentation

To ensure the effectiveness of the model and improve the generalization ability of the dataset, we use 7 image enhancement techniques such as rotation, noise, and brightness transformation to expand our dataset. For the species with less data, we adopt 7 methods for augmentation, and our purpose is to balance the number of pest images for each category. **Figure 3** shows some examples of data augmentation. At the same time, we extract subclasses such as eggs, larvae, and nymphs under each category to establish subclass information. Finally, we obtained a forestry pest dataset of 31 categories (including 16 sub-categories) with a total of 7,163 images. **Table 1** shows specific data for each category.

### Professional Data Annotation

For object detection tasks, annotation information is very important, which is related to the recognition accuracy of the model. The first is to classify the collected pests. In the image collection stage, we already have the initial classification information. On this basis, our three experts first need to independently determine whether the image conforms to the

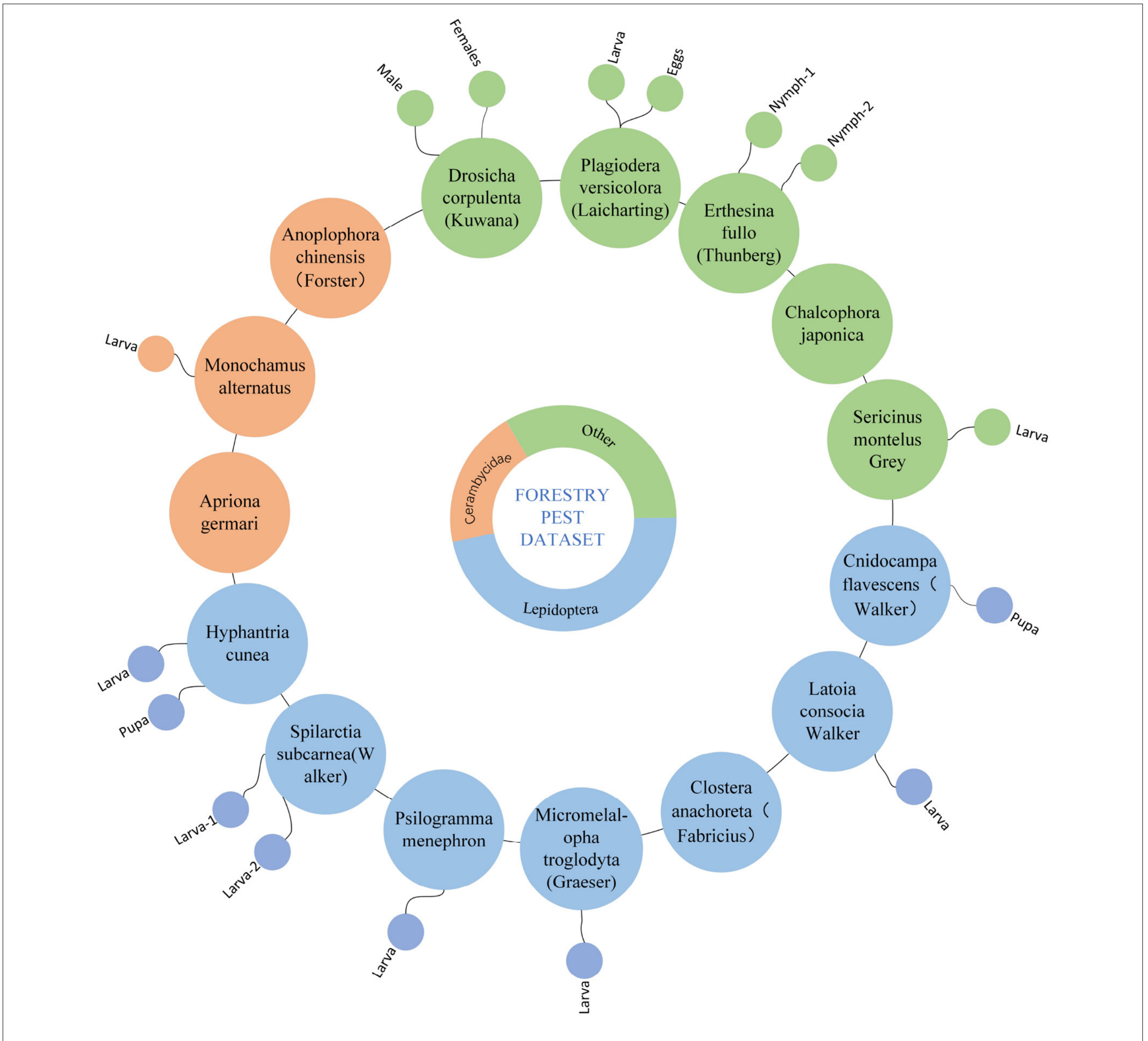


**FIGURE 2 |** The classification structure of Forestry Pest Dataset.

category. Uncertain images are eliminated by three experts. The location information of pests is also very important, which can help forestry protection workers better find the specific location of pests. On the premise of understanding the types of pests, we use the LabelImg tool to label the images, mainly labeling the types and locations of pests.

We recruited three volunteers to assist us in the annotation of the data. First, each volunteer will receive guidance and training from three forestry professionals to understand the basic characteristics of each type of pest. After that, we will train the three volunteers to use the LabelImg tool. Volunteers need to master the basic usage of LabelImg, including importing files and adding, modifying, and deleting annotation information. Experts will assist volunteers to annotate some images in the early stage, and then volunteers will independently complete subsequent image annotations. In the process of annotation, images that are difficult to identify or annotate will be resolved through consultation by three experts. After all image annotations are completed, volunteers use the annotation visualization to check whether there is any wrong or defective annotation information and submit it to experts for the final ruling.

## Dataset Split

Our Forestry Pest Dataset contains 7,163 images and 31 pest species. To ensure the training results, we randomly divide according to the following ratio: (Train: Val=9: 1): Test=9:

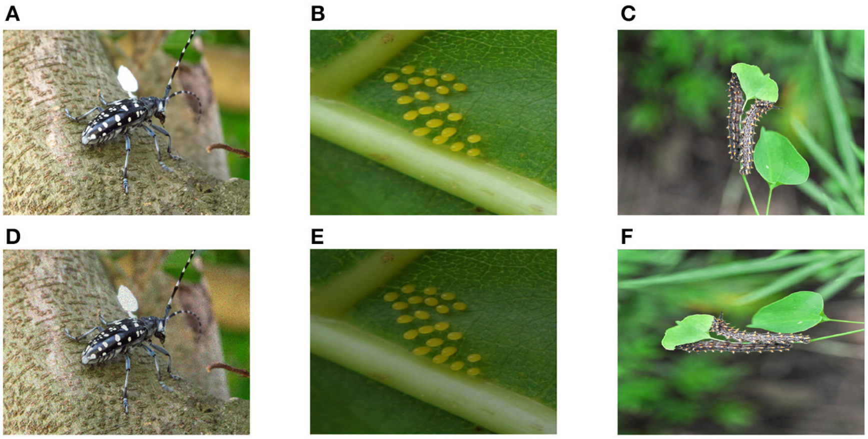


**FIGURE 3 |** Example of image data enhancement method. The first row is the original image, and the second row corresponds to the enhanced image. **(A)** Original image, **(B)** Original image, **(C)** Original image, **(D)** Noise, **(E)** Brightness transformation, and **(F)** Rotation.

**TABLE 1 |** Details of Forestry Pest Dataset.

| Class index | Pest | Sample size |
|---|---|---|
| 0 | *Drosicha contrahens (female)* | 218 |
| 1 | *Drosicha contrahens (male)* | 210 |
| 2 | *Chalcophora japonica* | 158 |
| 3 | *Anoplophora chinensis* | 426 |
| 4 | *Psacothea hilaris(Pascoe)* | 218 |
| 5 | *Apriona germari(Hope)* | 342 |
| 6 | *Monochamus alternatus* | 184 |
| 7 | *Plagiodera versicolora(Laicharting)* | 306 |
| 8 | *Latoia consocia(Walker)* | 290 |
| 9 | *Hyphantria cunea* | 303 |
| 10 | *Cnidocampa flavescens(Walker)* | 290 |
| 11 | *Cnidocampa flavescens(Walker) (pupa)* | 176 |
| 12 | *Erthesina fullo* | 280 |
| 13 | *Erthesina fullo (nymph)* | 156 |
| 14 | *Erthesina fullo (nymph 2)* | 192 |
| 15 | *Spilarctia subcarnea(Walker)* | 188 |
| 16 | *Psilogramma menephron* | 218 |
| 17 | *Sericinus montela* | 364 |
| 18 | *Sericinus montela (larvae)* | 200 |
| 19 | *Clostera anachoreta* | 294 |
| 20 | *Micromelalopha troglodyta(Graeser)* | 238 |
| 21 | *Latoia consocia(Walker) (larvae)* | 204 |
| 22 | *Plagiodera versicolora(Laicharting) (larvae)* | 196 |
| 23 | *Plagiodera versicolora(Laicharting) (ovum)* | 134 |
| 24 | *Spilarctia subcarnea(Walker) (larvae)* | 186 |
| 25 | *Spilarctia subcarnea(Walker) (larvae 2)* | 164 |
| 26 | *Psilogramma menephron (larvae)* | 208 |
| 27 | *Cerambycidae (larvae)* | 196 |
| 28 | *Micromelalopha troglodyta(Graeser) (larvae)* | 226 |
| 29 | *Hyphantria cunea (larvae)* | 224 |
| 30 | *Hyphantria cunea (pupa)* | 174 |

1. Specifically, the Forestry Pest Dataset is split into 5,801 training, 645 validation, and 717 testing images for the object detection task.

## Comparison With Other Forestry Pest Datasets

In **Table 2**, we compare our dataset with some existing datasets related to forestry pest identification tasks. Sun et al. (2018) and Hong et al. (2021) created related datasets using pheromone trap collection, but their datasets only deal with specific species of pests. The forestry pest dataset proposed by Baidu is processed and collected in a controlled laboratory environment. Due to these limitations, these related datasets are difficult to apply to practical applications. Chen et al. (2019) and Liu et al. (2022) focus on the classification of forest pests. Their dataset is rich in pest species and has a sufficient number of samples, which has played a huge role in practical applications. However, they have not made relevant attempts on pest detection tasks, and the relevant datasets have not been published.

## Diversity and Difficulty

Pests with different life cycles have different degrees of damage to forestry, so we retained images of these different morphological pests during data collection and annotation. However, due to the small differences between classes (similar features) and large differences within classes (there are many stages in the life cycle) of pests, accurate classification of their features is a difficult task in detection tasks. In addition, the imbalanced data distribution brings challenges to the feature learning of the model, and the imbalanced data will cause the learning results of the model to be biased toward a relatively large number of classes.

# 4. EXPERIMENT

To explore the application value of our proposed dataset, we evaluate several popular object detection algorithms on this

**TABLE 2 |** Comparison with existing forestry pest datasets.

| Dataset | Year | Class | Sample size | Avg | Public |
|---|---|---|---|---|---|
| Sun et al. (2018) | 2018 | 1 | 2,183 | - | Y |
| BaiDu | 2019 | 7 | 2,183 | 311 | Y |
| Chen et al. (2019) | 2019 | 38 | 9,072 | 238 | N |
| Hong et al. (2021) | 2021 | 1 | 50 | - | N |
| Liu et al. (2022) | 2022 | 67 | 67,953 | 1,014 | N |
| Ours | 2022 | 31 | 7,163 | 231 | Y |

*The "Class" denotes the number of categories. The "Public" indicates if the dataset is open source and available. The "Y" and "N" denote "yes" and "no," respectively. The "Avg" denotes average numbers of samples per class.*

**TABLE 3 |** Configuration of experimental environment.

| Hardware | Model |
|---|---|
| CPU | i7–8,700 |
| Memory | 64GB |
| GPU | RTX 3,090 24GB |
| Hard disk | 2.5TB |

dataset. Based on the two-stage approach of Faster RCNN (Ren et al., 2015), they scan the feature maps for potential objects by sliding windows, then classify them and regress the corresponding coordinate information. YOLOV4 (Bochkovskiy et al., 2020) and SSD (Liu W. et al., 2016) based on one-stage methods directly regress category and location information. In addition, we also evaluate the transformer-based end-to-end object detection algorithm Deformable DETR (Zhu X. et al., 2021).

## Experimental Settings

The framework used for this experiment is python3.8, torch1.9, cuda11.1. The experimental hardware is shown in **Table 3**.

## Object Detection Algorithms

After the accumulation of R-CNN and Fast RCNN, Faster RCNN integrates feature extraction (feature extraction), proposal extraction, bounding box regression (rect refine), and classification into one network in structure, which greatly improves the comprehensive performance., especially in terms of detection speed. SSD is a single-stage target detection algorithm, which uses convolutional neural network for feature extraction, and takes different feature layers for detection output. SSD is a multi-scale detection method.Based on the original YOLO target detection architecture, the YOLOV4 algorithm adopts the best optimization strategy in the CNN field in recent years, and has different degrees of optimization in terms of data processing, backbone network, network training, activation function, loss function, etc., achieving the perfect balance of speed and precision. Based on DETR, Deformable DETR improves the calculation method of the attention mechanism through sparse sampling, reduces the amount of calculation, and greatly reduces the training time of the model while ensuring accuracy.

**TABLE 4 |** Model parameter settings of SSD, Faster RCNN, and YOLOV4.

| Name | Value |
|---|---|
| Batch size | 16 |
| Epoch | 150 |
| Learn rate | 0.0001 |
| NMS | 0.3 |
| Match threshold | 0.5 |

**TABLE 5 |** Model parameter settings of Deformable DETR.

| Name | Value |
|---|---|
| Batch size | 2 |
| Epoch | 150 |
| Learn rate | 0.00002 |

## Parameters of Model Training

SSD, Faster RCNN, YOLOV4, and Deformable DETR initial model parameter settings are shown in **Tables 4**, **5**. To take into account the accuracy and training time, in the previous Deformable DETR model training process, the model reached convergence around 150 epoch, therefore, we chose 150 epoch, and Deformable DETR performed a learning rate decay every 40 epoch, so we chose 80 epoch as the intermediate result, Compare the performance of the four models on the dataset. At the same time, to maintain the consistency of the training cycle, we set the same epoch as Deformable DETR for the other three models.

## Evaluation Metrics

We use *mAP* and *Recall* as evalution metrics which are two widely used metrics in target detection. *mAP* and *Recall* are calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$mAP_{\alpha} = \frac{1}{N} \sum\nolimits_{n=1}^{N} AP_{\alpha}^{n} \tag{3}$$

$$mAP_{multi-scale} = \frac{1}{N_{ms}} \frac{1}{10} \sum\nolimits_{n_{ms}}^{N_{ms}} \sum\nolimits_{\alpha=0.5,step=0.05}^{0.95} mAP_{ms}^{\alpha} \tag{4}$$

Where, $TP$ is a positive sample predicted by the model as a positive class, $FP$ is a negative sample predicted as a positive class by the model, $FN$ is a negative class predicted by the model positive sample. Each class can calculate its Precision and Recall, and each class can get a PR curve, and the area under the curve is $AP$. $mAP_{\alpha}$ and $mAP_{multi-scale}$ are the average of all classes $AP$ at different confidence levels $\alpha$ and different scales value.

In the MS COCO dataset, objects with an area less than 32*32 are considered small objects, while objects with an area greater than 32*32 and less than 96*96 are considered medium objects.

## Experimental Results

Average precision performance of object detection methods under different IoU thresholds. The results are shown in **Table 6**.

From the experimental results in **Table 6**, it can be seen that the dataset in this paper has good accuracy on mainstream target detection models under short-time training. The recently proposed Deformable DETR can also be used on the dataset in this paper. Achieve roughly the same performance as SSD, Faster RCNN, and YOLOV4. An example of the detection of the model is shown in **Figure 4**.

From the above results, Deformable DETR based on Transformer architecture does not perform as well as YOLOV4 or even Faster RCNN in some cases. Based on our analysis, there are the following reasons.

1) Deformable DETR has no prior information. Whether it is YOLOV4 or Faster RCNN, they all have a part of prior information input, such as the clustering results of the coordinate information of the dataset, which can help the model find the target faster.
2) Although the attention mechanism calculation of Deformable DETR has been improved, its essence is still based on pixel calculation, which leads to a huge amount of calculation for high-resolution images. Deformable DETR does not have a feature fusion module similar to YOLOV4, which is detrimental to the detection of small objects.
3) Deformable DETR uses the Hungarian matching algorithm to match the prediction and ground truth, which cannot guarantee the convergence and accuracy of the model to a certain extent.

**TABLE 6 |** mAP$_\alpha$ values of different models on Forestry Pest Dataset.

| Model | Epoch | $mAP_{0.5}$ | $mAP_{0.75}$ |
|---|---|---|---|
| SSD | 80 | 96.6 | 80.6 |
| Faster RCNN | 80 | 96.8 | 83.6 |
| YOLOV4 | 80 | 98.8 | 70.2 |
| Deformable DETR | 80 | 96.6 | 89.8 |
| SSD | 150 | 98.1 | 91.1 |
| Faster RCNN | 150 | 97.5 | 85.2 |
| YOLOV4 | 150 | 99.7 | 88.3 |
| Deformable DETR | 150 | 97.1 | 90.4 |

## Confusion Matrix

The confusion matrix in target detection is very similar to that in classification, but the difference is that the object of the classification task is a picture, while the detection task is different. It includes two tasks of positioning and classification, and the object is each target in the picture. Therefore, to be able to draw positive and negative examples in the confusion matrix, it is necessary to distinguish which results are correct and which are wrong in the detection results. At the same time, the detection of errors also needs to be classified into different error categories. How to judge whether a detection result is correct, the most common way at present is to calculate the IOU of the detection frame and the real frame, and then judge whether the two frames match according to the IOU. For some targets below the threshold or not detected, they will be considered as the background class. The confusion matrix results of the model on the test set are shown in **Figure 5**.

## Case Study: Experiment on Large, Medium, and Small Targets

Small targets have always been a difficult task in target detection due to their small size and lack of feature information.

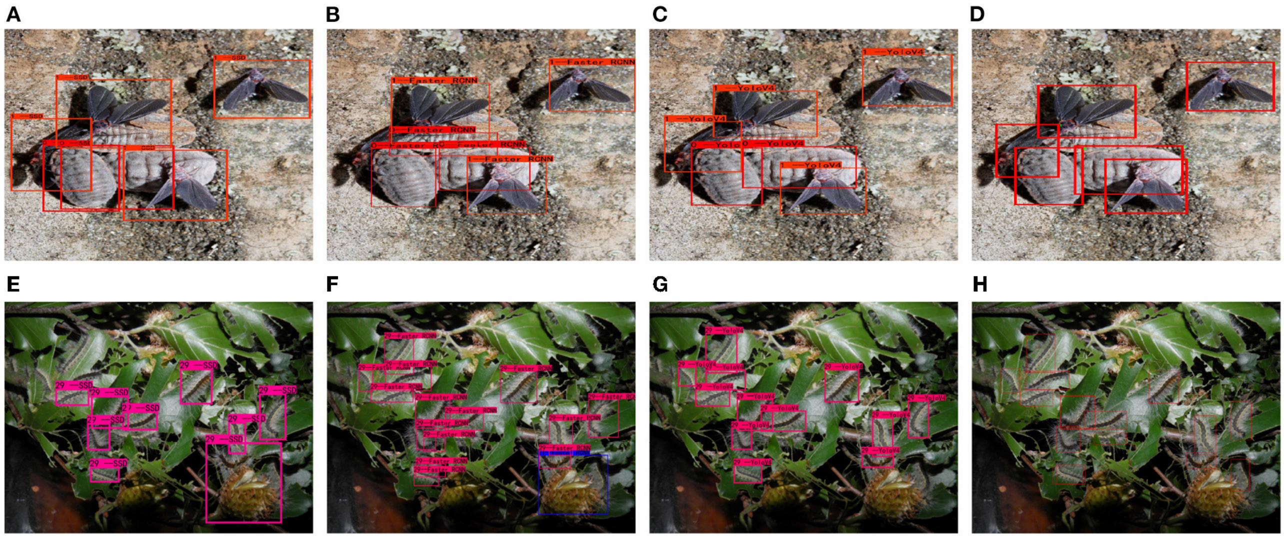


**FIGURE 4 |** Sample detection results on adults and larvae. From left to right are **(A,E)** SSD, **(B,F)** Faster RCNN, **(C,G)** YOLOV4, and **(D,H)** Deformable DETR.

In the field of forest pest detection, the detection of small targets is also a difficult task due to the real complexity. Our dataset contains small objects such as larvae and eggs. We also consider the model's ability to detect small objects in our dataset. The results are shown in **Tables 7**, **8**. The detection example of each model on small targets is shown in **Figure 6**.

As can be seen from the above table, YOLOV4 significantly leads the rest of the models in the detection of small targets, thanks to its powerful network structure and feature fusion, Deformable DETR is based on the attention mechanism of pixel-level computing, and the detection of small targets is not very friendly.

# 5. CONCLUSION AND FUTURE DIRECTIONS

## Conclusion

In this work, we collect a dataset, for forest insect pest recognition, including over 7,100 images of 31 classes. Compared with previous datasets, our dataset focuses on a variety of forestry pests, meets the detection needs of both real and experimental environments, and also includes pest forms in different periods, which some previous forestry pest datasets neglected. Meanwhile, we also evaluate some state-of-the-art recognition methods on our dataset. Exceptionally, this dataset has received good feedback on some mainstream object detection algorithms. However, in the detection of small objects, the existing deep learning methods cannot achieve the desired accuracy. Inspired by the success of the application in computer vision of the Transformer model, we also introduced the Transformer model to solve the forestry pest identification problem. We hope this work will help advance future research on related fundamental issues as well as forestry pests identification tasks.

## Future Directions

To better promote the development of forestry pest identification, we will continue to collect forestry pest data and expand the dataset to 99 categories. For pests that have occurred or diseases caused by pests, there is a lack of relevant data sets and research support. In response, we will collect images of diseases caused by insect pests.

Although the existing deep learning models have achieved good results in forest pest identification, small target recognition is still a challenge. We will optimize and improve the model in

**TABLE 7 |** mAP$_{multi-scale}$ values of multi-scale results achieved by different models on Forestry Pest Dataset.

| Model | Epoch | $mAP_{small}$ | $mAP_{medium}$ | $mAP_{large}$ |
|---|---|---|---|---|
| SSD | 80 | 27.4 | 53.5 | 72.7 |
| Faster RCNN | 80 | 14.2 | 49.0 | 74.0 |
| YOLOV4 | 80 | 49.4 | 57.0 | 62.2 |
| Deformable DETR | 80 | 28.0 | 61.6 | 87.1 |
| SSD | 150 | 35.2 | 65.4 | 84.7 |
| Faster RCNN | 150 | 30.0 | 48.9 | 76.5 |
| YOLOV4 | 150 | 56.2 | 63.1 | 73.2 |
| Deformable DETR | 150 | 30.3 | 63.8 | 87.7 |

**TABLE 8 |** Recall$_{multi-scale}$ values of multi-scale results achieved by different models on Forestry Pest Dataset.

| Model | Epoch | $Recall_{small}$ | $Recall_{medium}$ | $Recall_{large}$ |
|---|---|---|---|---|
| SSD | 80 | 41.2 | 61.5 | 77.1 |
| Faster RCNN | 80 | 23.8 | 55.8 | 78.1 |
| YOLOV4 | 80 | 53.0 | 61.0 | 67.7 |
| Deformable DETR | 80 | 31.4 | 68.8 | 91.3 |
| SSD | 150 | 44.9 | 69.6 | 87.4 |
| Faster RCNN | 150 | 38.8 | 54.7 | 80.0 |
| YOLOV4 | 150 | 60.2 | 67.5 | 77.0 |
| Deformable DETR | 150 | 34.3 | 71.1 | 91.6 |

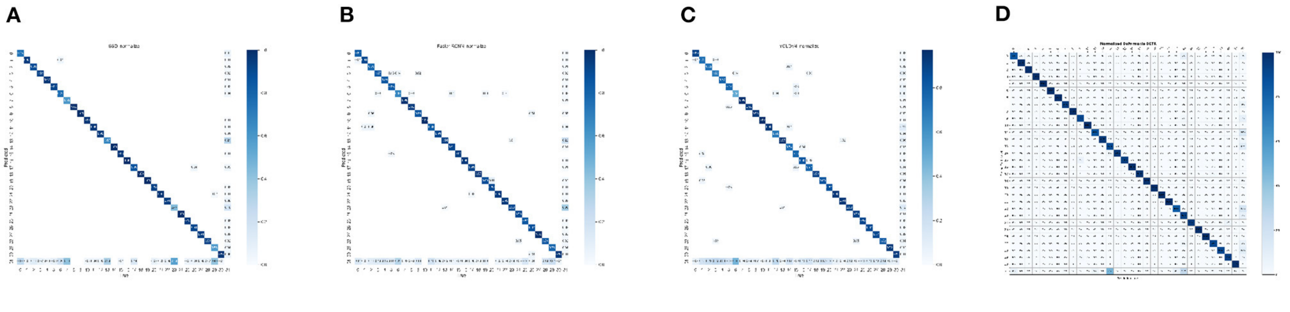


**FIGURE 5 |** Confusion matrix of the mo del on the test set. Epoch=150, Iou-threshold=0.5, Index=31 means background. **(A)** SSD, **(B)** Faster RCNN, **(C)** YOLOV4, and **(D)** Deformable DETR.

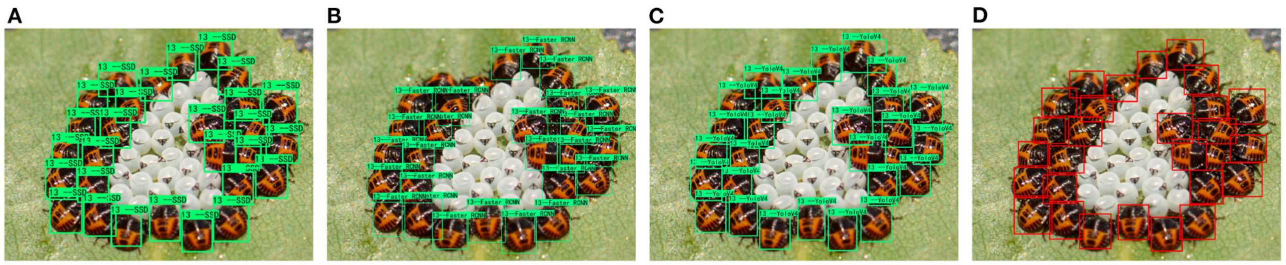


**FIGURE 6 |** Small sample detection results on our Forestry Pest Dataset. **(A)** SSD, **(B)** Faster RCNN, **(C)** YOLOV4, and **(D)** Deformable DETR.

the follow-up to further improve the model's ability to detect small targets.

## DATA AVAILABILITY STATEMENT

The datasets for this study can be found in the https://drive.google.com/drive/folders/1WnNDLEZCNpXKwJzjnJsQKSAYKljIIRCH?usp=sharing.

## AUTHOR CONTRIBUTIONS

GW designed research and revised the manuscript. LL and BL conducted experiments, data analysis, and wrote the manuscript. RZ collected pest data. WC and RD revised the paper. All authors contributed to the article and approved the submitted version.

## FUNDING

The research is supported by the Educational Department of Jilin Province of China (Grant No. JJKH20210752KJ). The research is also supported by the project of research on independent experimental teaching mode of program design foundation based on competition to promote learning which is the Higher Education Research of Jilin Province of China (Grant No. JGJX2021D191).

## REFERENCES

Ahmed, F., Al-Mamun, H. A., Bari, A. H., Hossain, E., and Kwan, P. (2012). Classification of crops and weeds from digital images: a support vector machine approach. *Crop Prot.* 40, 98–104. doi: 10.1016/j.cropro.2012.04.024

Al-Hiary, H., Bani-Ahmad, S., Reyalat, M., Braik, M., and Alrahamneh, Z. (2011). Fast and accurate detection and classification of plant diseases. *Int. J. Comput. Appl.* 17, 31–38. doi: 10.5120/2183-2754

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

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., Chen, L., and Wang, S. (2019). Pest image recognition of garden based on improved residual network. *Trans. Chin. Soc. Agric. Machi* 50, 187–195. doi: 10.6041/j.issn.1000-1298.2019.05.022

Ebrahimi, M., 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

Estruch, J. J., Carozzi, N. B., Desai, N., Duck, N. B., Warren, G. W., and Koziel, M. G. (1997). Transgenic plants: an emerging approach to pest control. *Nat. Biotechnol.* 15, 137–141. doi: 10.1038/nbt0297-137

FAO (2020). *New Standards to Curb the Global Spread of Plant Pests and Diseases.* Available online at: https://www.fao.org/news/story/en/item/1187738/icode/

Fina, F., Birch, P., Young, R., Obu, J., Faithpraise, B., and Chatwin, C. (2013). Automatic plant pest detection and recognition using k-means clustering algorithm and correspondence filters. *Int. J. Adv. Biotechnol. Res.* 4, 189–199.

Gandhi, R., Nimbalkar, S., Yelamanchili, N., and Ponkshe, S. (2018). "Plant disease detection using cnns and gans as an augmentative approach," in *2018 IEEE International Conference on Innovative Research and Development (ICIRD)* (Bangkok: IEEE), 1–5.

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

He, Y., Zeng, H., Fan, Y., Ji, S., and Wu, J. (2019). Application of deep learning in integrated pest management: a real-time system for detection and diagnosis of oilseed rape pests. *Mobile Inf. Syst.* 2019, 4570808. doi: 10.1155/2019/4570808

Hong, S.-J., Kim, S.-Y., Kim, E., Lee, C.-H., Lee, J.-S., Lee, D.-S., et al. (2020). Moth detection from pheromone trap images using deep learning object detectors. *Agriculture* 10, 170. doi: 10.3390/agriculture10050170

Hong, S.-J., Nam, I., Kim, S.-Y., Kim, E., Lee, C.-H., Ahn, S., et al. (2021). Automatic pest counting from pheromone trap images using deep learning object detectors for matsucoccus thunbergianae monitoring. *Insects.* 12, 342. doi: 10.3390/insects12040342

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. (2013). "3D object representations for fine-grained categorization," in *Proceedings of the IEEE International Conference on Computer Vision Workshops* (Sydney, NSW: IEEE), 554–561.

Le-Qing, Z., and Zhen, Z. (2012). Automatic insect classification based on local mean colour feature and supported vector machines. *Orient Insects.* 46, 260–269. doi: 10.1080/00305316.2012.738142

Li, R., Wang, R., Xie, C., Liu, L., Zhang, J., Wang, F., et al. (2019). A coarse-to-fine network for aphid recognition and detection in the field. *Biosyst. Eng.* 187, 39–52. doi: 10.1016/j.biosystemseng.2019.08.013

Li, X.-L., Huang, S.-G., Zhou, M.-Q., and Geng, G.-H. (2009). "Knn-spectral regression lda for insect recognition," in *2009 First International Conference on Information Science and Engineering* (Nanjing: IEEE), 1315–1318.

Li, Y., and Yang, J. (2020). Few-shot cotton pest recognition and terminal realization. *Comput. Electron. Agric.* 169, 105240. doi: 10.1016/j.compag.2020.105240

Lim, S., Kim, S., Park, S., and Kim, D. (2018). "Development of application for forest insect classification using cnn," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)* (Singapore: IEEE), 1128–1131.

Liu, J., and Wang, X. (2020). Tomato diseases and pests detection based on improved yolo v3 convolutional neural network. *Front. Plant Sci.* 11, 898. doi: 10.3389/fpls.2020.00898

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* (Amsterdam: Springer), 2–37.

Liu, Y., Liu, S., Xu, J., Kong, X., Xie, L., Chen, K., et al. (2022). Forest pest identification based on a new dataset and convolutional neural network model with enhancement strategy. *Comput. Electron. Agric.* 192, 106625. doi: 10.1016/j.compag.2021.106625

Liu, Z., Gao, J., Yang, G., Zhang, H., and He, Y. (2016). Localization and classification of paddy field pests using a saliency map and deep convolutional neural network. *Sci. Rep.* 6, 1–12. doi: 10.1038/srep20410

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 60, 91–110. doi: 10.1023/B:VISI.0000029664.99615.94

Maji, S., Rahtu, E., Kannala, J., Blaschko, M., and Vedaldi, A. (2013). Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*. doi: 10.48550/arXiv.1306.5151

Ren, S., He, K., Girshick, R., and Sun, J. (2015). "Faster r-cnn: towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28 (NIPS 2015)* (Montreal, QC), 28.

Sun, Y., Liu, X., Yuan, M., Ren, L., Wang, J., and Chen, Z. (2018). Automatic in-trap pest detection using deep learning for pheromone-based dendroctonus valens monitoring. *Biosyst. Eng.* 176, 140–150. doi: 10.1016/j.biosystemseng.2018.10.012

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition* (Boston, MA: IEEE), 1–9.

Torralba, A., Murphy, K. P., Freeman, W. T., and Rubin, M. A. (2003). "Context-based vision system for place and object recognition," in *Computer Vision, IEEE International Conference on, Vol. 2* (Nice: IEEE), 273–273.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. (2011). The caltech-ucsd birds-200-2011 dataset. California, CA.

Wang, F., Wang, R., Xie, C., Yang, P., and Liu, L. (2020). Fusing multi-scale context-aware information representation for automatic in-field pest detection and recognition. *Comput. Electron. Agric.* 169, 105222. doi: 10.1016/j.compag.2020.105222

Wang, R., Liu, L., Xie, C., Yang, P., Li, R., and Zhou, M. (2021). Agripest: A large-scale domain-specific benchmark dataset for practical agricultural pest detection in the wild. *Sensors* 21, 1601. doi: 10.3390/s21051601

Wang, Z., Wang, K., Liu, Z., Wang, X., and Pan, S. (2018). A cognitive vision method for insect pest image segmentation. *IFAC PapersOnLine* 51, 85–89. doi: 10.1016/j.ifacol.2018.08.066

Wu, X., Zhan, C., Lai, Y.-K., Cheng, M.-M., and Yang, J. (2019). "Ip102: a large-scale benchmark dataset for insect pest recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 8787–8796.

Zhang, H. T., Hu, Y. X., and Zhang, H. Y. (2013). Extraction and classifier design for image recognition of insect pests on field crops. *Adv. Mater. Res.* 756, 4063–4067. doi: 10.4028/www.scientific.net/AMR.756-759.4063

Zhu, J., Cheng, M., Wang, Q., Yuan, H., and Cai, Z. (2021). Grape leaf black rot detection based on super-resolution image enhancement and deep learning. *Front. Plant Sci.* 12, 695749. doi: 10.3389/fpls.2021.695749

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2021). "Deformable {detr}: Deformable transformers for end-to-end object detection," in *International Conference on Learning Representations* (Vienna).

**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.

frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 08 September 2022
DOI 10.3389/frai.2022.868926

OPEN ACCESS

EDITED BY
Ian Stavness,
University of Saskatchewan, Canada

REVIEWED BY
Mario Merone,
Campus Bio-Medico University, Italy
Michael Pound,
University of Nottingham, United Kingdom
Dmitry Murashov,
Federal Research Center Computer Science and Control (RAS), Russia

*CORRESPONDENCE
Massimo Martinelli
massimo.martinelli@isti.cnr.it

†These authors have contributed equally to this work

SPECIALTY SECTION
This article was submitted to AI in Food, Agriculture and Water, a section of the journal Frontiers in Artificial Intelligence

RECEIVED 08 February 2022
ACCEPTED 03 August 2022
PUBLISHED 08 September 2022

CITATION
Bruno A, Moroni D, Dainelli R, Rocchi L, Morelli S, Ferrari E, Toscano P and Martinelli M (2022) Improving plant disease classification by adaptive minimal ensembling. *Front. Artif. Intell.* 5:868926. doi: 10.3389/frai.2022.868926

# Improving plant disease classification by adaptive minimal ensembling

Antonio Bruno[1]†, Davide Moroni[1]†, Riccardo Dainelli[2], Leandro Rocchi[2], Silvia Morelli[3], Emilio Ferrari[3], Piero Toscano[2]† and Massimo Martinelli[1]*†

[1]Institute of Information Science and Technologies, National Research Council, Pisa, Italy, [2]Institute of BioEconomy, National Research Council, Firenze, Italy, [3]Barilla G. e R. Fratelli S.p.A., Parma, Italy

A novel method for improving plant disease classification, a challenging and time-consuming process, is proposed. First, using as baseline EfficientNet, a recent and advanced family of architectures having an excellent accuracy/complexity trade-off, we have introduced, devised, and applied refined techniques based on transfer learning, regularization, stratification, weighted metrics, and advanced optimizers in order to achieve improved performance. Then, we go further by introducing adaptive minimal ensembling, which is a unique input to the knowledge base of the proposed solution. This represents a leap forward since it allows improving the accuracy with limited complexity using only two EfficientNet-b0 weak models, performing ensembling on feature vectors by a trainable layer instead of classic aggregation on outputs. To the best of our knowledge, such an approach to ensembling has never been used before in literature. Our method was tested on PlantVillage, a public reference dataset used for benchmarking models' performances for crop disease diagnostic, considering both its original and augmented versions. We noticeably improved the state of the art by achieving 100% accuracy in both the original and augmented datasets. Results were obtained using PyTorch to train, test, and validate the models; reproducibility is granted by providing exhaustive details, including hyperparameters used in the experimentation. A Web interface is also made publicly available to test the proposed methods.

KEYWORDS

**plant diseases, image classification, deep learning-artificial neural network (DL-ANN), adaptive ensemble, Convolutional Neural Networks (CNN)**

## 1. Introduction

Early detection of plant stress is one of the most crucial practices in agriculture (Nagaraju and Chawla, 2020). Biotic stress in plants is caused by living organisms, specifically viruses, bacteria, fungi, nematodes, insects, arachnids, and weeds, while abiotic stress is caused by environmental factors such as drought, heat, cold, strong wind, flooding, and nutrient deficiencies. In agriculture, both kinds of stress are a significant cause of crop yield and quality loss leading to serious monetary harm when limits for the occurrence of the stress are exceeded (Kashef, 2020; Pantazi et al., 2020).

Although over the years, genetics has made available cultivars that are increasingly resistant to various types of stress, the issue of yield and quality losses remains crucial on a global scale, especially since climate change leads to the co-occurrence of abiotic and biotic stresses (Pandey et al., 2017). Even today, the majority of the inspections are done manually by direct visual analysis, which may not make it easy to identify the disease and its type. Indeed, farmers use their naked eyes for plant inspection, which needs constant observation, high skills, and experience. Some of them are supported by guidelines with basic concepts and aiding materials (pictures/notes to identify symptoms and patterns of stress) that are relevant to distinguish between biotic and abiotic injuries and determine the possible cause and solution to adopt. At other times, farmers might require technical support to achieve a formal and complete diagnosis. In all these cases, the methodologies adopted are time-consuming and expensive (Zhang et al., 2020), often not viable for large farms or not affordable for small farms. Even the identification of weeds typology—broadleaf or grassy—is difficult in their early stages (from germination to the development of the first four/six leaves), i.e., exactly when it would be the most suitable time to counter them. This issue has increased the importance of automated infection recognition and compelled researchers to devise methods or systems that can more accurately diagnose the problem (Ma et al., 2017). In addition, the increased public concern about environmental conservation coupled with the need for more efficient agriculture necessary to cope with the simultaneous increase in population and reduction of available land) demands the introduction of new cost-effective and sustainable methods and solutions to support farmers in their daily work. In this context, machine learning techniques can finally trigger a revolution for the timely suppression of organisms harmful to plants and keep the use of chemical treatment and other forms of intervention to economically and ecologically justified levels.

Computer vision-based methods are now being considered a key enabler in this revolution. The problem has a relatively long history, including several attempts based on the use of particular imaging technologies such as thermal and stereo images (Prince et al., 2015), color and depth images (Rousseau et al., 2012), or even fluorescence imaging spectroscopy (Wetterich et al., 2013) coupled with *ad hoc* image processing pipelines. Such advanced imaging modalities might provide very specific and accurate analysis suitable for particular, especially high-revenue, crops in precision agriculture. However, standard RGB images might be preferable for the broader adoption of vision-based methods for fighting plant diseases even in low-resource and low-income areas of the world. Progresses in artificial intelligence and their excellent classification capabilities on standard images have encouraged several research lines. For instance, neural networks for plant disease classification have been used before (Huang, 2007) making use in most cases of handcrafted features and conventional computer vision pipelines. Indeed, independently of the application domain, typical computer vision techniques are composed of a pipeline of phases that almost equally contribute to the quality of the final result. In the case of image classification, in particular, the phases are (i) *preprocessing* for improving the image quality (e.g., denoising, color enhancement/balancing); *segmentation* for isolating the foreground from the background, to focus only on the useful information;*feature extraction* for obtaining only the relevant information of the foreground represented in a numeric vector (i.e., feature vector), mostly performed by a domain expert, and (iv) *classification* for learning and performing a mapping between the input feature vector and output classes.

In the last years, the paradigm shift proposed by *deep learning* (LeCun et al., 2015), consisting of a way to perform *representation learning* i.e., obtaining the data feature vector without involving a domain expert, has allowed embedding and automatically performing all the phases described above.

Convolutional Neural Networks (CNNs or ConvNets) represent Deep Learning in the scope of Computer Vision and are state-of-the-art (SOTA) in most tasks (Khan et al., 2020). Even if there are many CNN archetypes, all of them are essentially composed by stacking a variable number of modules (that usually share parameters to reduce complexity) consisting of the following layers applied sequentially:

- convolutional layers: they apply several adaptive filters to regions of the image obtaining their abstract representation;
- pooling layers: they perform aggregations which have the 2-fold effect of summarizing data, picking only relevant elements, leading to dimensionality reduction;
- non-linear activation layers: they are used to obtain a more powerful and expressive representation, reaching different levels of abstraction.

At the end of an architecture composed of the layers mentioned above, one or more fully connected layers can be stacked. This organization allows automatic preprocessing, segmentation, and feature extraction whilst classification/regression is feasible, putting a dedicated output module at the top of the architecture. The very first conceived CNN was LeNet (LeCun et al., 1989) more than 30 years ago. Again, only in the last 10 years, CNNs have been experiencing massive use and success, frequently improving the SOTA on different tasks (Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; Szegedy et al., 2015; He et al., 2016; Howard et al., 2017; Hu et al., 2018; Wang et al., 2020).

Convolutional Neural Networks have been adopted to tackle the problem of plant disease classifications. For instance, Wang et al. (2017) have applied transfer learning and fine-tuning of general-purpose architectures to provide fine-grained disease severity classification in the case of the apple black rot images dataset, obtaining a best 90.16% performance using VGG16.

Similarly, Ferentinos (2018) used AlexNet and GoogleNet, training the models with the use of an open database of 87,848 images, containing 25 different plants in a set of 58 distinct classes of [*plant*, *disease*] pairs, achieving the best performance of 99.53%. For the training and validation of deep learning paradigms, several datasets are available (Lu and Young, 2020). However, all of them have some limitations e.g., in size, variety of plants, disease coverage, and extrinsic shooting conditions (i.e., varying illumination and backgrounds). Among them, PlantVillage (Hughes and Salathe, 2015a,b) has emerged as a *de facto* open reference dataset for plant disease classification and, as such, it is considered a benchmark in this article, although it shares limitations of other datasets and, notably, the presence of standard backgrounds instead of real-world ones. It should be noticed that a large-scale benchmark dataset has been recently proposed (Liu et al., 2021), together with a new approach to disease recognition. Still, such a dataset is not freely available and has not yet gained reference value. In general, previous methods addressing the classification of PlantVillage images achieve good performance, however, they often do not sufficiently address the efficiency and complexity of the employed paradigms. Indeed, in order to achieve more significant penetration and broader adoption of the methods, the proposed paradigms should be capable of running on low resource hardware, especially on smartphones, even in the absence of remote clouds.

In view of the above consideration, in this study, we propose a new approach to plant disease classification based on adaptive minimal ensembling. The main contribution is represented by a novel approach to ensembling: different weak classifiers are trained and then combined to obtain a new combined classifier. The novelties of the proposed approach are at least 2-fold: from one side, we propose a fully trainable combination layer, granting end-to-end differentiability of the global architecture; then, in our approach, the combination layer does not act on the output layers of the weak classifiers as in other classical approaches, but the weak classifiers are truncated before. Namely, the final fully-connected layers of each weak classifier are removed, and the combination happens directly at the *deep feature* level.In addition, such an approach is brought into practice by adopting a family of SOTA models, namely EfficientNet (Tan and Le, 2019), known for their optimal complexity/performance trade-off, for each weak classifier. EfficientNet is refined by applying advanced techniques on data and processing, significantly improving the classification task. Namely, besides using ensembling, we perform transfer learning from ImageNet and introduce a novel optimizer as well as a novel validation scheme together with other minor tricks. From an experimental point of view, the article provides an advance since it shows that adaptive minimal ensembling can be used to reach top performance with a minimal computational burden compared to other promising schemes in the literature. Indeed, improving state-of-the-art, we achieve 100% accuracy on the PlantVillage dataset using an ensembling of only two weak classifiers (and thus minimal) while at the same time requiring less computational resources than the previous methodologies tested on the PlantVillage dataset. As a final contribution, carrying out the experiments both on the original PlantVillage dataset and its augmented version, we show that our method based on minimal ensembling is less sensitive to data augmentation with respect to other methods reported in the literature, in which performance significantly drops when training is not performed on the augmented dataset.

The article is organized as follows. In Section 2, we describe our designing strategy in detail (including the proposed models and the validation phases), focusing on the novelty aspects of the solution. The experimental setup is then introduced in Section 3 in which the number and typologies of experimental runs, including hyperparameters and other details, are reported in order to guarantee reproducibility. In Section 4, the obtained experimental results are reported and discussed, while Section 5 ends the article with ideas for future research.

## 2. Materials and methods

Among the pool of CNN architectures available in the SOTA for image classification, it was decided to use the EfficientNet (Tan and Le, 2019) family as the core component in this study. This was motivated by several factors.

First, as the name suggests, EfficientNet improves the classification quality without having huge complexity with respect to the models having similar classification performances. EfficientNet family consists of 8 progressively improved versions (b0-b7) with limited complexity growth, all of them having the inverted bottleneck MBConv (first introduced in MobileNetV2) as the core module, which expands and compresses channels reducing the complexity of convolution. The real novelty introduced is the way EfficientNets perform scaling of the network to achieve optimal performances given a predefined complexity. In the CNN literature, there are 3 main types of scaling as shown in Figure 1:

- *depth scaling*, which consists in increasing the number of layers in the CNN; it is the most popular scaling method in the literature and allows to catch features at more levels of abstraction;
- *width scaling* means increasing the number of convolutional kernels and parameters or channels, giving the model the capability to represent different features at the same level;
- *input scaling* means increasing the size/resolution of the input images, allowing to capture more details.

Each of these scalings can be set manually or by a grid search, but there are two problems with the traditional scaling

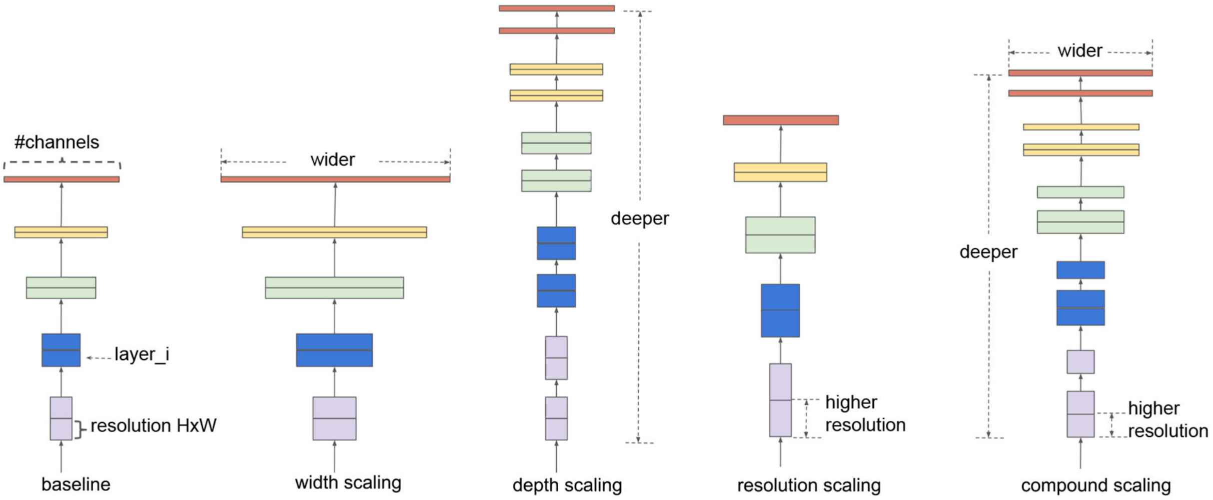


FIGURE 1
Example of scaling types, from left to right: a baseline network example, conventional scaling methods that only increase one network dimension (width, depth, and resolution), and in the end the EfficientNet compound scaling method that uniformly scales all three dimensions with a fixed ratio. Image taken from the original article (Tan and Le, 2019).

method: first, they increase the model complexity, usually exponentially, with tons of new parameters to tune and, second, after a certain level, experiments show that scaling does not improve performances. The scaling method introduced in the article is named *compound scaling* and suggests that strategically performing all scaling together delivers better results because it is observed that they are dependent. Intuitively, Tan and Le (2019) introduce a compound coefficient $\phi$ representing the amount of resources available to the model and find the optimal scaling combination using that amount of resources following the rules:

$$\text{depth: } d = \alpha^{\phi} \qquad \text{width: } w = \beta^{\phi} \qquad \text{resolution: } r = \gamma^{\phi}$$
$$\text{such that} \quad \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \quad \text{and} \quad \alpha \geq 1, \beta \geq 1, \gamma \geq 1$$

In this way, the total complexity of the network is approximately proportional to $2^{\phi}$ (refer to the original article for more details). In the following sections, our strategy is illustrated, highlighting the differences from the previously cited works.

## 2.1. Input preprocessing

In many applications, the models are not fed directly with the images provided by the datasets, but images are preprocessed to improve the performances. In our study, since the image quality of the dataset of interest is already sufficient, we opted not to perform any image enhancement or further augmentation because an augmented version of the dataset already exists.

The only preprocessing we applied is the normalization, in order to have all data described under the same distribution (pixel values in the $[0, 1]$ range and centered around the mean) which improves the stability and convergence of the training.

## 2.2. Transfer learning

Transfer learning (Weiss et al., 2016) is the technique of taking knowledge gained while solving one problem and applying it to a different but related problem. In this case, like most cases for image classification, the stored knowledge is brought by pre-trained models from ImageNet (Deng et al., 2009) task, since it has more than 14 million images belonging to 1,000 generic classes (including plants).

## 2.3. Avoid overfitting

In order to prevent overfitting (i.e., avoid the model being too specialized to the data from the training set with poor performances on *unknown* data), during training we use early stopping (i.e., training is interrupted after no improvements on the validation set after a certain number of epochs, called *patience*, is achieved) and regularization (i.e., adding noise to the loss, usually proportional to the norm of the model parameter vector, in order to keep parameters with low values).

## 2.4. Ensembling

Ensembling is the technique that combines several base models, called *weak*, in order to produce one optimal model

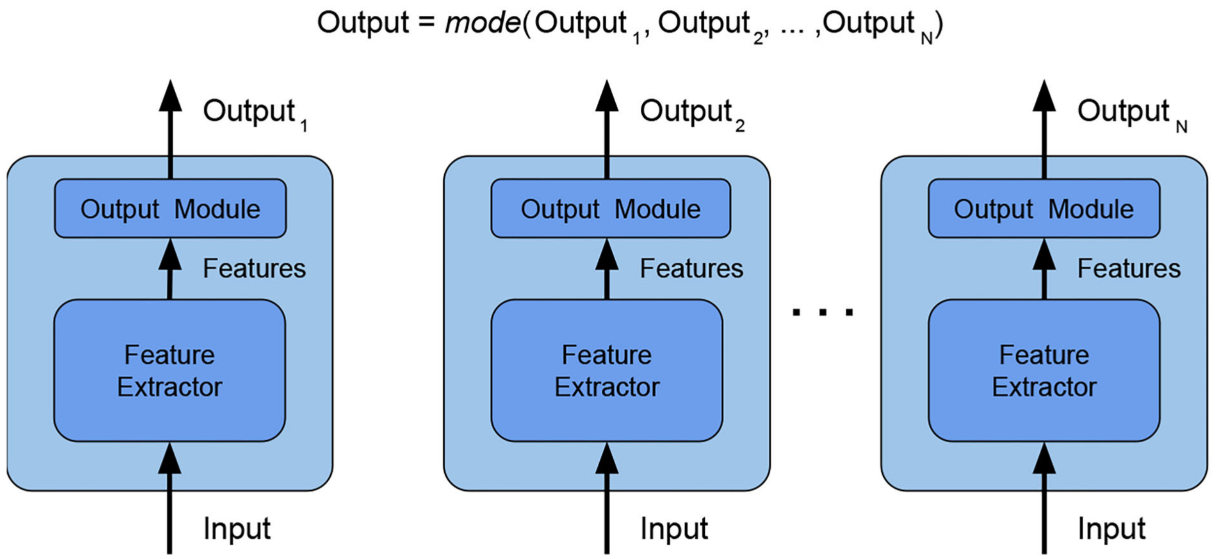


FIGURE 2
Ensemble by voting—the final label is obtained by picking the most frequent label among the weak models. In this way, the weak models are independent and the ensemble is effective with a high number of heterogeneous weak models. Weak models are CNN architectures since now represented by the sequence of Feature Extractor + Output module.

to achieve a better performance than any of the constituent models alone (Opitz and Maclin, 1999). The studies (Sagi and Rokach, 2018; Dong et al., 2019) provide a comprehensive study on different ensembling methods supported by empirical results. Instead of performing a sort of validation to obtain the best combination of ensembling, we adopt the following heuristic choices:

- *ensemble main category*: Due to its simplicity, we decided to use *bagging*, which consists in training several independent weak models on different subsets of data. Since randomness (Ho, 1995) and heterogeneity (Gashler et al., 2008) are known to lead to good quality ensembling, subsets are picked totally random;
- *ensemble size*: The study in Bonab and Can (2016) provides the number of weak models to use for obtaining the ideal ensemble model, however, the study in Bonab and Can (2019) proves that a small number of weak models is enough to achieve high performances with low complexity. We, thus, decided to consider an ensemble size equal to 2 (i.e., the ensemble is composed of two weak models only, being therefore minimal).
- *combination type*: The typical way of combining weak models is to perform voting/averaging as shown in Figure 2 (predicting the output from all weak models and then picking the most frequent output/average of outputs), respectively for classification/regression. However, in previous study, the ensemble is only a static aggregator. In our method, we opted to perform an adaptive combination of the weak models; in addition, instead of combining the outputs (Figure 3) of weak models, the features that the CNNs extract from the input (Figure 4) are combined. In this way, the complexity of the ensemble is further reduced without diminishing its power and expressiveness. Indeed the combination layer is of the same type of the output layer as the weak models (i.e., Linear + LogSoftmax) and keeping both would introduce an unnecessary redundancy. In particular, the mechanism adopted for the fusion of the ensemble is performed first by concatenating the characteristics and then applying a linear transformation to match the output size (i.e., we perform a kind of weighted sum on the concatenated features).
- *weak models training*: Even if the study in Sollich and Krogh (1995) shows that overfitted weak models might also lead to a good adaptive ensemble, we decided to train the weak models by avoiding overfitting to save precious training time and to have weak models of higher quality.

## 2.5. Validation phases

The validation of every single model is divided into two main phases: first *end-to-end training* is performed and then followed by output module *fine-tuning*. For the first phase, transfer learning starting from the ImageNet pre-trained model is applied, introducing a new output module to adjust the

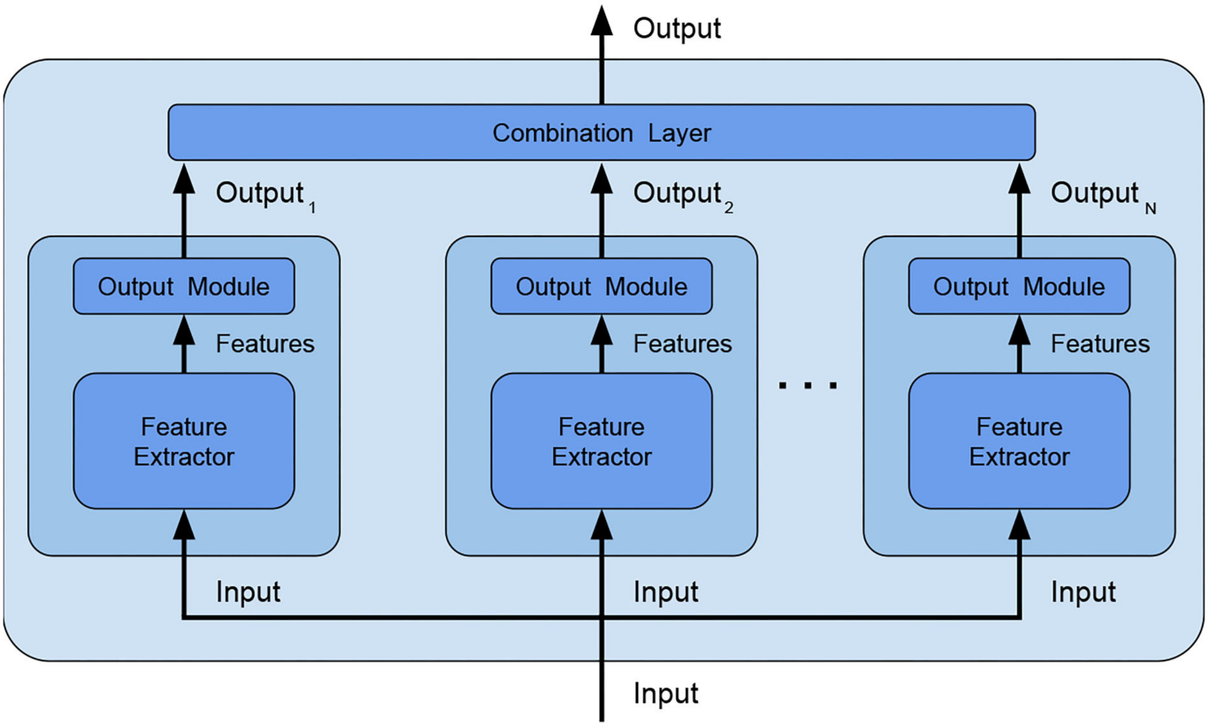


FIGURE 3
Ensemble by output combination–an additional combination layer is fed with the outputs of the weak models and combines them, in this way, the weak models are no longer independent and the combination layer can be trained to better adapt to data.

output size from the 1,000 classes in the ImageNet task to the number of classes in the PlantVillage dataset. A training phase is performed using AdaBelief (Zhuang et al., 2020) optimizer which guarantees both fast convergence and generalization. The parameters used in AdaBelief are the default ones, i.e., learning rate equal to $5 \cdot 10^{-4}$, betas $(0.9, 0.999)$, eps $10^{-16}$, using weight decoupling without rectifying. After such training is concluded, the second phase starts: all the internal layers (i.e., the layers performing feature extraction) of the model obtained with the previous step are frozen, and a new training by Stochastic Gradient Descent (SDG) with a learning rate $3 \cdot 10^{-3}$ and momentum 0.9 is performed. This leads to the fine-tuning of the output module of each classifier.

These steps conclude the validation phase for every single model. When going further to ensembling, each resulting single model is regarded as a weak model of a combined classifier and an additional dedicated pipeline is introduced for training the ensemble. First, the two best performing models are selected and truncated dropping their output module, which is replaced by a common combination layer. Then, ensemble fine-tuning (i.e., only the adaptive combination layer is trained) is performed using the same optimizer setting of the first validation phase. The reasons why we perform a dedicated pipeline for the ensemble are described in Section 3.3.

## 2.6. The plantVillage dataset

The PlantVillage dataset (Hughes and Salathe, 2015a,b) is a dataset for multiclass image classification tasks having 55,448 images (61.486 in its augmented version) divided into 39 classes representing background-only (out of domain images e.g., animals, buildings), healthy and diseased plants.

Table 1 shows that images span 14 plant species: Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Bell Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, and Tomato and contains images of 17 fungal diseases, 4 bacterial diseases, 2 mold (Oomycete) diseases, 2 viral diseases, and 1 disease caused by a mite (some examples are shown in Figure 5).

# 3. Experimental setup

In this section, we describe the design choices justified by prior observations. All the reported experimental results were

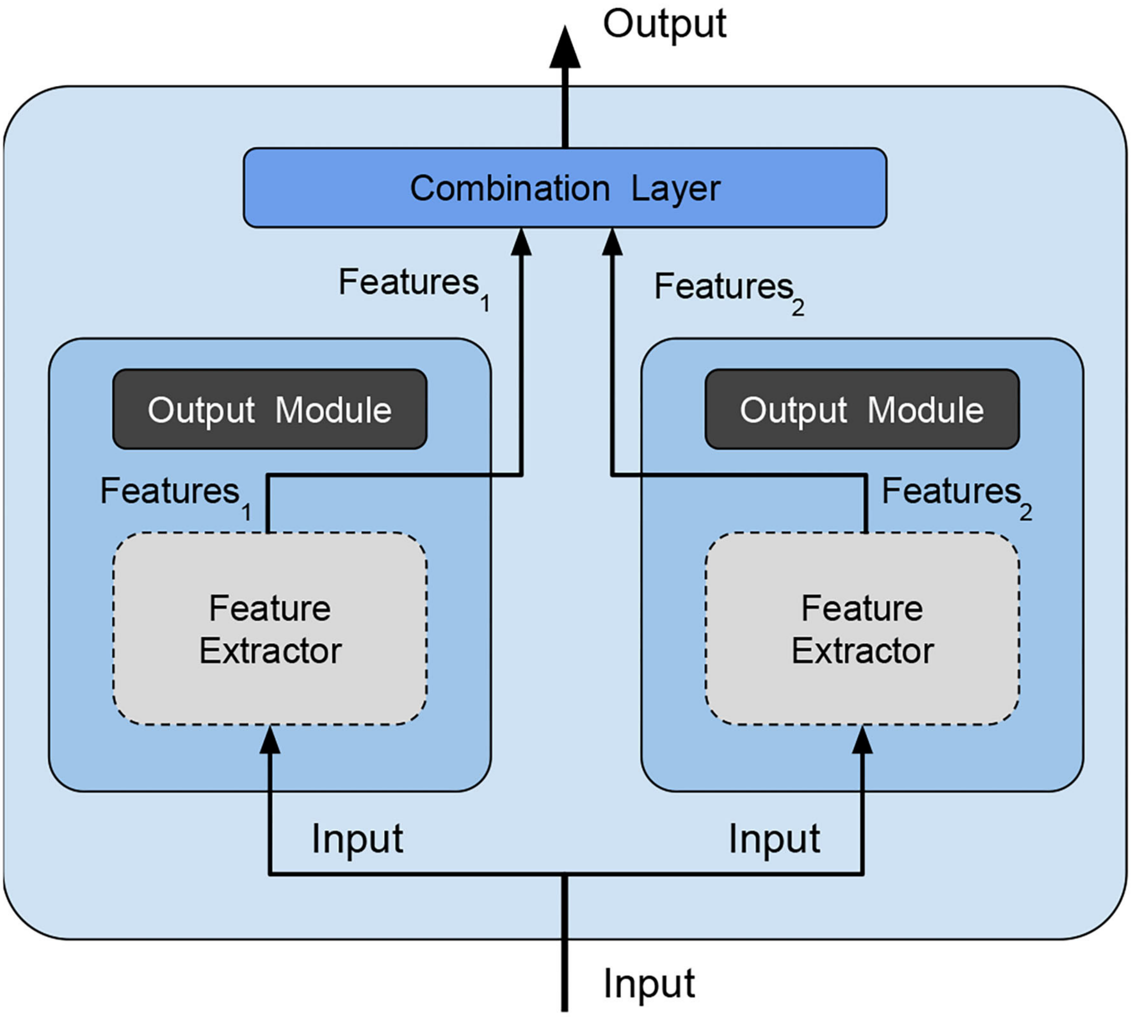


FIGURE 4
Our ensemble method—is an optimized version of the method shown in Figure 3 because we avoid redundancy and reduce complexity by deleting the output module (dark gray filled) of weak models and feeding the combination layer directly with the features extracted by each weak model. Feature extraction modules (light gray filled with dashed borders) have the parameters frozen during ensemble training.

obtained using the PyTorch (Paszke et al., 2019) open-source machine learning framework.

A somewhat non-conventional training/validation/test splitting has been used in the experiments to reproduce the conditions closest to the study in Ümit Atila et al. (2021) representing the SOTA for PlantVillage while doing this work. More in detail, datasets have been split into training (90%), validation (7%), and test (3%). While obviously splits have the same sizes as previous articles, however, since the picks are random, the actual elements in each subset may vary. Moreover, we performed stratification (i.e., preserving the classes ratio). Besides the non-conventional split, in the result and discussion section, a classic split is also considered to show the suitability of our strategy also in this case.

## 3.1. Loss and metrics

**Training Loss:** Due to the multiclass nature of the problem, the Cross-Entropy Loss (which exponentially penalizes differences between predicted and true values, expressed as the probability of class belonging) is used. For this reason, the model output has a size of 39 (i.e., number of classes), and each element output[$i$] represents the probability that the input model belongs to class $i$.

**Validation and test metrics:** For the validation set evaluation, we decided to use the Weighted F1-score because this takes into account both correct and wrong predictions (true/false positive/negative) and weighting allows us to manage any imbalance of the classes (more representative classes have a

TABLE 1 Class labels distribution of PlantVillage dataset.

| Class Name | Class frequency | Class name | Class frequency |
|---|---|---|---|
| Apple scab | 630 | Pepper healthy | 1,478 |
| Apple black rot | 621 | Potato early blight | 1,000 |
| Apple cedar apple rust | 275 | Potato healthy | 1,000 |
| Apple healthy | 16,45 | Potato late blight | 152 |
| Background without leaves | 1,143 | Raspberry healthy | 371 |
| Blueberry healthy | 1,502 | Soybean healthy | 5,090 |
| Cherry powdery mildew | 1,052 | Squash powdery mildew | 1,835 |
| Cherry healthy | 854 | Strawberry healthy | 1,109 |
| Corn gray leaf spot | 513 | Strawberry leaf scorch | 456 |
| Corn common rust | 1,192 | Tomato bacterial spot | 2,127 |
| Corn northern leaf blight | 985 | Tomato early blight | 1,000 |
| Corn healthy | 1,162 | Tomato healthy | 1,591 |
| Grape black rot | 1,180 | Tomato late blight | 1,909 |
| Grape black measles | 1,383 | Tomato leaf mold | 952 |
| Grape leaf blight | 985 | Tomato septoria leaf spot | 1,771 |
| Grape healthy | 1,162 | Tomato spider mites | 1,676 |
| Orange haunglongbing | 5,507 | Tomato target spot | 1,404 |
| Peach bacterial spot | 2,297 | Tomato mosaic virus | 373 |
| Peach healthy | 360 | Tomato yellow leaf curl virus | 5,357 |
| Pepper bacterial spot | 997 | | |

Some diseases are typical of particular plant phenotypes, there are also healthy leaf and background-only images. The frequency values refer to the standard dataset, while in the case of its augmented version only the classes with a size less than 1,000 were augmented to reach 1,000 images (classes having more images are not modified).

greater contribution). On the other hand, in order to compare our results with previous studies, we used accuracy to evaluate the test set.

## 3.2. Hyperparameters

In order to save time, after an initial coarse search, we fixed some hyperparameters:

- Early-stopping patience set at 10: Because deep models have relatively fast convergence and they usually start overfitting after convergence, so there is no need to have much patience;
- Batch size set at 32: Because it is the maximum size allowed on the GPUs we used to perform model training and lower values showed no improvement (and would make training slower). Stratification even inside the batches would have been desirable but this is possible only if the batch size is greater than the number of the classes, which is not our case;
- Input image size fixed to 256: In order to preserve input image quality, a larger size would ruin the images, the lower size would reduce details;
- Mean and SD used for normalization:

$\mu = [0.4683, 0.5414, 0.4477]$ $\sigma = [0.2327, 0.2407, 0.2521]$ for augmented dataset

$\mu = [0.4685, 0.5424, 0.4491]$ $\sigma = [0.2337, 0.2420, 0.2531]$ for original dataset

Moreover, we observed that regularization was not needed during the end-to-end training phase, and, in some cases, it even led to worse results. Regularization is then used only in weak models (not used for ensemble) fine-tuning step, with the following hyperparameters:

- regularization type: Lasso (L1), Ridge (L2)
- regularization factor ($\lambda$): 0, $10^{-4}$, $5 \cdot 10^{-4}$, $10^{-3}$.

For each combination, we used 3 different random seeds in order to obtain different model parameter initialization values, and different train/valid/test splits (useful to obtain random heterogeneous weak models for ensemble).

Considering each dataset, we had the following combinations:

- End-to-end phase: 8 (EfficientNet architectures ranging from b0 to b7) × 3 (random seeds) = 24 combinations;
- Fine-tuning phase: 24 (end-to-end phase results) × 2 (regularization types) × 4 (regularization factors) = 192 combinations;

Since there are two datasets (standard, augmented), the total number of runs is equal to $2 \times (192) = 384$, excluding ensembling which is addressed in Section 3.3 below.

## 3.3. Adaptive minimal ensembling, improving performances with minimum complexity

The last experimental step is to evaluate the performance of ensembling. We opted to perform this step using only EfficientNet-b0 as weak models and not the full family of weak models trained in Section 3.2. This choice was motivated by two

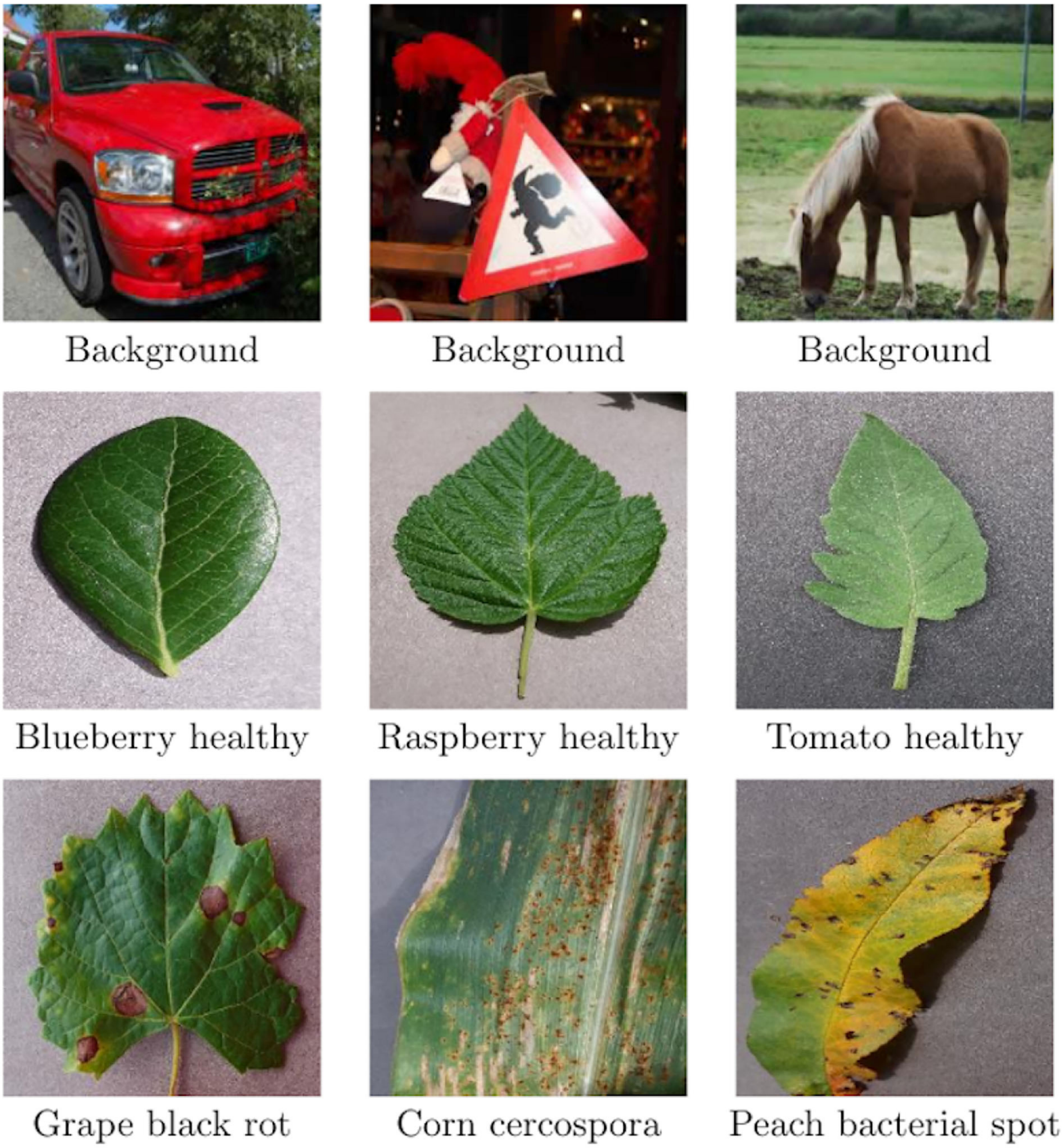


FIGURE 5
Some PlantVillage image examples. From top to bottom: 3 examples of background, 3 examples of healthy leaves, and 3 examples of diseases.

observations: first of all, the performances of all EfficientNet variants after the end-to-end phase are very similar as we show in Table 2, but the b0 variant is much simpler ($\approx$ 5M parameters vs $\approx$ 66.7M parameters of b7), so we decided to investigate on the simplest type of ensemble, even in terms of the number of parameters (and not just in terms of ensemble size using only 2 weak models). Moreover, we skipped the fine-tuning phase as it is used to optimize those parameters that are removed during ensembling because they are redundant (notice, however, that we performed fine-tuning anyway before, in order to collect experimental results to allow fair comparison and show the improvements ensembling can

TABLE 2 Table with best Weighted F1-score results, for each EfficientNet variant, after the first phase of validation (i.e., end-to-end training).

| | | Original | | | Augmented | |
|---|---|---|---|---|---|---|
| **Model** | **Test** | **Valid** | **Train** | **Test** | **Valid** | **Train** |
| EfficientNet-b0 | 99.6995 | 99.8454 | 99.9960 | 99.7832 | 99.8374 | 99.9982 |
| EfficientNet-b1 | 99.5793 | 99.8454 | 100.000 | 99.8916 | 99.8141 | 99.9928 |
| EfficientNet-b2 | 99.5192 | 99.7681 | 99.9140 | 99.7832 | 99.8374 | 99.9982 |
| EfficientNet-b3 | 99.6394 | 99.8712 | 99.9860 | 99.8374 | 99.9303 | 99.9964 |
| EfficientNet-b4 | 99.6995 | 99.8454 | 99.9980 | 99.5664 | 99.8606 | 99.9982 |
| EfficientNet-b5 | 99.7596 | 99.7939 | 99.9920 | 99.9458 | 99.8606 | 99.9982 |
| EfficientNet-b6 | **99.7596** | **99.8712** | **99.9880** | 99.7290 | 99.8141 | 99.9675 |
| EfficientNet-b7 | 99.5192 | 99.8454 | 99.9960 | **99.8916** | **99.9303** | **99.9982** |

The values of the best architectures are in bold.

TABLE 3 Table with Weighted F1-score of the models (best to worst) of the 5 end-to-end training runs using EfficientNet-b0 variants only that will be the weak models of the simplest ensemble.

| | | Original | | | Augmented | |
|---|---|---|---|---|---|---|
| **Model** | **Test** | **valid** | **Train** | **Test** | **valid** | **Train** |
| EfficientNet-b0 | 99.8197 | 99.9485 | 100.000 | 99.7832 | 100.000 | 100.000 |
| EfficientNet-b0 | 99.8197 | 99.8969 | 100.000 | 99.6748 | 99.8838 | 100.000 |
| EfficientNet-b0 | 99.7596 | 99.8454 | 100.000 | 99.8374 | 99.8374 | 99.9980 |
| EfficientNet-b0 | 99.7596 | 99.7423 | 100.000 | 99.8374 | 99.6515 | 100.000 |
| EfficientNet-b0 | 99.5793 | 99.9227 | 99.9960 | 99.7832 | 99.6747 | 100.000 |

TABLE 4 Table with the Weighted F1-score of the 5 ensemble runs (best to worst) composed of the two best EfficientNet-b0 variants.

| | | Original | | | Augmented | |
|---|---|---|---|---|---|---|
| **Model** | **Test** | **valid** | **Train** | **Test** | **valid** | **Train** |
| EfficientNet-b0 ensemble | 100.000 | 100.000 | 100.000 | 100.000 | 100.000 | 100.000 |
| EfficientNet-b0 ensemble | 100.000 | 100.000 | 100.000 | 100.000 | 100.000 | 100.000 |
| EfficientNet-b0 ensemble | 100.000 | 100.000 | 100.000 | 100.000 | 100.000 | 100.000 |
| EfficientNet-b0 ensemble | 100.000 | 100.000 | 100.000 | 100.000 | 100.000 | 100.000 |
| EfficientNet-b0 ensemble | 100.000 | 100.000 | 99.9980 | 100.000 | 99.9768 | 100.000 |

offer over single models). For the ensemble phase, we followed another validation scheme that is, for each version of the dataset, the following:

1. Five end-to-end training of EfficientNet-b0 with different initializations and data splits results in Table 3;
2. No fine-tuning because the parameters involved in this phase would be removed during ensemble;
3. Five fine-tuning of a minimal ensemble composed of the two best weak models obtained at point 1, using different initializations and data splits.

In this way, only 10 runs per dataset are performed, which are drastically fewer than 192 as described in Section 3.2 and every run is much faster. All training runs had the same configuration: AdaBelief optimizers with a learning rate $5 \cdot 10^{-4}$, betas (0.9, 0.999), eps $10^{-16}$, using weight decoupling without rectifying, and Weighted F1-score as validation metric.

It must be noted that using different seeds for each validation phase, both for end-to-end and for ensemble fine-tuning, produces different dataset splits: this can be, therefore, viewed as cross-validation and, by averaging the values in Table 4, we obtain the same results proving that our solution is consistent.

## 4. Results and discussion

Every validation step results in incremental improvements: We discuss them one by one in the following.

TABLE 5 Table with best Weighted F1-score results, for each EfficientNet variant, after the second phase of validation (i.e., end-to-end training + fine-tuning).

| Model | Original<br>Test | Original<br>valid | Original<br>Train | Augmented<br>Test | Augmented<br>valid | Augmented<br>Train |
|---|---|---|---|---|---|---|
| EfficientNet-b0 | 99.6995 | 99.8454 | 100.000 | 99.8916 | 99.8838 | 100.000 |
| EfficientNet-b1 | 99.6995 | 99.8969 | 100.000 | 99.8916 | 99.8374 | 99.9982 |
| EfficientNet-b2 | 99.5793 | 99.8712 | 100.000 | 99.7832 | 99.9303 | 99.9982 |
| EfficientNet-b3 | 99.7596 | 99.8712 | 99.9900 | 99.8374 | 99.9303 | 99.9964 |
| EfficientNet-b4 | 99.6995 | 99.8454 | 99.9980 | 99.5664 | 99.8606 | 99.9982 |
| EfficientNet-b5 | 99.7596 | 99.7939 | 99.9920 | 99.9458 | 99.8606 | 99.9982 |
| EfficientNet-b6 | 99.8197 | 99.8712 | 99.9900 | 99.7290 | 99.8141 | 99.9675 |
| EfficientNet-b7 | **99.8197** | **99.8712** | **100.000** | **99.8916** | **99.9303** | **100.000** |

The values of the best architectures are in bold.

TABLE 6 Table comparing complexity (measured as the number of parameters) of the SOTA models on PlantVillage task.

| Model | Dataset<br>Original | Dataset<br>Augmented |
|---|---|---|
| Mohanty et al. (2016) (GoogleNet) | ≈ 7M | - |
| Too et al. (2019) (DenseNets-121) | ≈ 7.9M | - |
| Chen et al. (2020) (MobileNet-Beta) | ≈ 3.7M | - |
| Ümit Atila et al. (2021) (EfficientNet) | ≈ 30.5M | ≈ 19.5M |
| End-to-end (ours) | ≈ 43.2M | ≈ 66.7M |
| Fine-tuning (ours) | ≈ 66.7M (100k) | ≈ 66.7M (100k) |
| Minimal ensemble (ours) | ≈ 10M (100k) | ≈ 10M (100k) |

The minimal ensemble is the least complex because even if it has 10M parameters it can be considered as 5M because the weak models are independent and can be executed in parallel. Moreover, during training, only the 100k parameters of the combination layers are trained.

**End-to-end phase:** as shown in Table 2 results are very similar, moreover considering both datasets there is no architecture being always the best/worst in both cases. It is also important to say that already in this phase we improved the SOTA: indeed, the best results until this study were obtained by Ümit Atila et al. (2021) with a Validation Accuracy of 97.62% and Test Accuracy of 98.31% for the standard dataset, Validation Accuracy of 98.97% and Test Accuracy of 99.38% for the augmented dataset. We relate this improvement to our design choices: AdaBelief optimizer, performing stratification during dataset, using Weighted F1-score as validation metric, and using normalization parameters taken from datasets instead of ImageNet defaults.

**Fine-tuning phase:** As shown in Table 5, EfficientNet-b7 is the best architecture in both datasets. In major cases, this phase led to no improvements, in a few cases, it improves performances on training data without getting worse on validation/test (i.e., improvements without overfitting). The improvements are more noticeable for the standard dataset (because in the first phase, results were lower), especially for the b7 variant obtaining improvements overall. We observed no differences among different regularization types and factors, but it was needed (because with $\lambda = 0$, we got no improvements).

**Ensemble:** This phase gave a huge peak of improvement in both datasets, obtaining a perfect 100% accuracy on both versions of the dataset (Table 4) being much less complex (10M vs. 66.7M total parameters of EfficientNet-b7 as shown in Table 6).

We finally summarize the design choices and the improvements they led to.

**Transfer learning:** Helped to speed up and optimize (because training from scratch done in the preliminary analysis always led to poor results) the end-to-end training phase;

**Adabelief optimizer:** Allowed to reach lower minimal points due to its high convergence speed without losing generalization power (previous SOTA work used Adam);

**Stratification and weighted F1-score:** Reduced the problems due to high data imbalance, indeed in the augmented dataset, there is less imbalance and with the same condition, there are better performances on it (previous SOTA work used normal accuracy);

**Regularization:** Harmful during end-to-end training but essential during fine-tuning, even if there is no seeming difference among regularization types or factors (previous SOTA work does not seem to use regularization);

**Ensembling:** Using two weak models is enough to have meaningful improvements if the models are heterogeneous enough (i.e., trained on different subsets of data) even if they are very simple. This avoids overfitting since the training of weak models increased the base quality and reduced overall execution time. Finally, performing ensembling on features instead of outputs further reduced the complexity and deleted redundancies.

Now, we consider the comparison between our solution and the models representing the SOTA on the PlantVillage task over the years: Tables 6, 7 show that our design choices, different from previous studies (i.e., AdeBelief optimizer, stratification, weighted-F1, regularization) improved performances if we consider single models. Our minimal ensemble method introduced in this study had a 2-fold improvement: perfect accuracy score without increasing model complexity. Moreover, the feature extractor modules are frozen making the real trainable parameters number very low even in the ensemble (100k trainable parameters over the 10M in total), and the execution of weak models can be performed in parallel since they are independent (thus, the execution time for a 10M parameters ensemble is close to the execution time of a single 5M parameters model).

As said before, we performed a split to make a fair comparison with the SOTA (i.e., 90% train, 7% validation, and 3% test). However, to prove its robustness, our solution has also been tested using a traditional 80/10/10 split: Few weak models were trained and then two best were used to run some ensemble fine-tuning, for each dataset version. The results in Table 8 prove that our solution is suitable also for traditional data splits.

A web application was also implemented to show the results, allowing to pick an image from the datasets and showing its classification and probabilities; this is publicly accessible at the following address: http://plantvillage.isti.cnr.it:9090.

TABLE 7 Table comparing accuracies (measured as correct prediction over the whole dataset) of the SOTA models on the PlantVillage task.

| Model | Dataset | |
|---|---|---|
| | Original | Augmented |
| Mohanty et al. (2016) (GoogleNet) | 99.3500% | - |
| Too et al. (2019) (DenseNets-121) | 99.7500% | - |
| Chen et al. (2020) (MobileNet-Beta) | 99.8500% | - |
| Ümit Atila et al. (2021) (EfficientNet) | 99.9100% | 99.9700% |
| End-to-end (ours) | 99.9729% | 99.9904% |
| Fine-tuning (ours) | 99.9856% | 99.9919% |
| Minimal ensemble (ours) | 100.000% | 100.000% |

Since the end-to-end phase, our study was shown to improve the SOTA.

## 5. Conclusion

Identifying plant diseases and devising optimal adaptive countermeasures can bring significant improvement in crop quality and yields. In this context, expert systems based on artificial intelligence can be a valid aid to farmers, yet there are still no operational services for most crops. This article contributed to the creation of artificial intelligence modules for plant disease classification with high accuracy and efficiency. Indeed, it has been described how specific target design choices can lead to a relevant performance improvement over an already top-rated solution without efficiency loss. The first improvement of the state of the art was reached by using a different optimizer (i.e., Adabelief) in combination with techniques to deal with unbalanced data (i.e., stratification and Weighted F1-score). A family of classifiers based on the EfficientNet architecture has been proposed with similar accuracy but increasing complexity.

The second gain in performance was obtained by introducing a minimal adaptive ensemble model using the combination of the features of the two least complex weak models: while the number of total parameters doubled with respect to the least complex model, perfect accuracy was achieved. To the best of our knowledge, none of the above-mentioned techniques have been ever used before. Doubling the number of total parameters, however, did not increase the total complexity for two reasons: First, only a tiny part of parameters is trained during the ensemble training step (100k over 10M), and, second, the weak models can process input in parallel (therefore, the overall execution time is very close to the execution time of a single weak model). In addition, by minimal ensembling, the performance gap between original and augmented datasets is reduced; it could be argued that this type of ensembling can be helpful in cases where data balancing and augmentation are not feasible or not convenient in terms of computational time/resources. These perspectives will be studied in the future, considering other disparate domains and reference benchmarking datasets. In particular, we are currently investigating the use of adaptive minimal ensembling and the gains it is possible to achieve both in absolute average precision and in the ratio between precision and complexity, toward more sustainable use of artificial intelligence.

Regarding the precision agriculture domain, having achieved top performance on the *de facto* benchmarking

TABLE 8 Table showing the accuracies of the two weak models and the fine-tuning ensembling them, using a traditional 80/10/10 split.

| Model | Original | | | Augmented | | |
|---|---|---|---|---|---|---|
| | Test | valid | Train | Test | valid | Train |
| EfficientNet-b0 weak1 | 99.8738 | 99.8557 | 100.000 | 99.8536 | 99.8536 | 100.000 |
| EfficientNet-b0 weak2 | 99.8377 | 99.9278 | 100.000 | 99.7886 | 99.9187 | 100.000 |
| Ensemble (weak1 + weak2) | 100.000 | 100.000 | 99.9864 | 100.000 | 100.000 | 100.000 |

dataset, the research will pursue the possibility to provide operational services to farmers to identify and recognize plant diseases. To this end, a participatory approach is being followed to gather a large dataset in the specific domain of durum wheat crop culture from pictures taken in the field by farmers, also using mobile devices. This initiative is leading to a realistic and more complex dataset to champion the methods proposed in this article.

## 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

All the authors contributed to this article that was coordinated by PT and MM. The data analysis and the experiments were done and manuscript was written by AB, DM, and MM. The article drafting was done by AB, DM, PT, and MM. All the authors revised the manuscript several times and approved the article.

## Funding

This study was partially supported by the *Barilla Agrosatplus* research project DBA.AD001.034.

## Acknowledgments

The authors wish to thank various colleagues for their contributions to this study: Simone Agostinelli, Paolo La Cava, Rolando Maglione, Enrico Polverigiani, Gustavo Savino, and Marco Silvestri for their very valuable technical support, patient guidance, and useful critiques of this research work. The authors are also particularly grateful to Ovidio Salvetti for his useful and constructive recommendations and enthusiastic encouragement on this project.

## Conflict of interest

Authors SM and EF are employed by Barilla G. e R. Fratelli S.p.A.

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

## Publisher's note

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

## References

Bonab, H., and Can, F. (2019). Less is more: A comprehensive framework for the number of components of ensemble classifiers. *IEEE Trans. Neural Netw. Learn. Syst*. 30, 2735–2745. doi: 10.1109/TNNLS.2018.2886341

Bonab, H. R., and Can, F. (2016). "A theoretical framework on the ideal number of classifiers for online ensembles in data streams," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM '16* (New York, NY: Association for Computing Machinery), 2053–2056.

Chen, J., fu Zhang, D., and Nanehkaran, Y. A. (2020). Identifying plant diseases using deep transfer learning and enhanced lightweight network. *Multimedia Tools Appl.* 79, 31497–31515. doi: 10.1007/s11042-020-09669-w

Deng, J., Dong, W., Socher, R., Li, L., Kai, L. I., and Li, F. F. (2009). "Imagenet: a large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition* (Miami, FL: IEEE), 248–255.

Dong, X., Yu, Z., Cao, W., Shi, Y., and Ma, Q. (2019). A survey on ensemble learning. *Front. Comput. Sci.* 14, 241–258. doi: 10.1007/s11704-019-8208-z

Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* 145, 311–318. doi: 10.1016/j.compag.2018.01.009

Gashler, M., Giraud-Carrier, C., and Martinez, T. (2008). "Decision tree ensemble: small heterogeneous is better than large homogeneous," in *2008 Seventh International Conference on Machine Learning and Applications* (New York, NY), 900–905.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV: IEEE), 770–778.

Ho, T. K. (1995). "Random decision forests," in *Proceedings of the Third International Conference on Document Analysis and Recognition, Vol. 1, ICDAR '95* (Montreal, QC: IEEE Computer Society), 278.

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 preprint arXiv:1704.04861*. doi: 10.48550/arXiv.1704.04861

Hu, J., Shen, L., and Sun, G. (2018). "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 7132–7141.

Huang, K.-Y. (2007). Application of artificial neural network for detecting phalaenopsis seedling diseases using color and texture features. *Comput. Electron. Agric.* 57, 3–11. doi: 10.1016/j.compag.2007.01.015

Hughes, D., and Salathe, M. (2015b). An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060*.

Hughes, D. P., and Salathe, M. (2015a). An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing. *CoRR, abs/1511.08060*.

Kashef, R. (2020). *Adopting Big Data Analysis in the Agricultural Sector: Financial and Societal Impacts*. Singapore: Springer Singapore.

Khan, A., Sohail, A., Zahoora, U., and Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev*. 53, 5455–5516. doi: 10.1007/s10462-020-09825-6

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Commun. ACM*. 60, 84–90. doi: 10.1145/3065386

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., et al. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Comput*. 1, 541–551. doi: 10.1162/neco.1989.1.4.541

Liu, X., Min, W., Mei, S., Wang, L., and Jiang, S. (2021). Plant disease recognition: a large-scale benchmark dataset and a visual region and loss reweighting approach. *IEEE Trans. Image Process*. 30, 2003–2015. doi: 10.1109/TIP.2021.3049334

Lu, Y., and Young, S. (2020). A survey of public datasets for computer vision tasks in precision agriculture. *Comput. Electron. Agric*. 178, 105760. doi: 10.1016/j.compag.2020.105760

Ma, J., Du, K., Zhang, L., Zheng, F., Chu, J., and Sun, Z. (2017). A segmentation method for greenhouse vegetable foliar disease spots images using color information and region growing. *Comput. Electron. Agric*. 142, 110–117. doi: 10.1016/j.compag.2017.08.023

Mohanty, S. P., Hughes, D. P., and Salath,é, M. (2016). Using deep learning for image-based plant disease detection. *Front. Plant Sci*. 7, 1419. doi: 10.3389/fpls.2016.01419

Nagaraju, M., and Chawla, P. (2020). Systematic review of deep learning techniques in plant disease detection. *Int. J. Syst. Assurance Eng. Manag*. 11, 547–560. doi: 10.1007/s13198-020-00972-1

Opitz, D. W., and Maclin, R. (1999). Popular ensemble methods: An empirical study. *J. Artif. Intell. Res*. 11, 169–198. doi: 10.1613/jair.614

Pandey, P., Irulappan, V., Bagavathiannan, M. V., and Senthil-Kumar, M. (2017). Impact of combined abiotic and biotic stresses on plant growth and avenues for crop improvement by exploiting physio-morphological traits. *Front Plant Sci*. 8, 537. doi: 10.3389/fpls.2017.00537

Pantazi, X. E., Moshou, D., and Bochtis, D. (2020). "Chapter 3-utilization of multisensors and data fusion in precision agriculture," in *Intelligent Data Mining and Fusion Systems in Agriculture*, eds X. E. Pantazi, D. Moshou, and D. Bochtis (Cambridge, MA: Academic Press), 103–173.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). "Pytorch: an imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, eds H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Red Hook, NY: Curran Associates, Inc.), 8024–8035.

Prince, G., Clarkson, J. P., Rajpoot, N. M., et al. (2015). Automatic detection of diseased tomato plants using thermal and stereo visible light images. *PLoS ONE* 10, 1–20. doi: 10.1371/journal.pone.0123262

Rousseau, D., Lucidarme, P., Bertheloot, J., Caffier, V., Morel, P., Chapeau-Blondeau, F., et al. (2012). On the use of depth camera for 3d phenotyping of entire plants.

Sagi, O., and Rokach, L. (2018). Ensemble learning: a survey. *Wiley Interdisc. Rev*. 8, e1249. doi: 10.1002/widm.1249

Simonyan, K., and Zisserman, A. (2015). "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations* (San Diego, CA).

Sollich, P., and Krogh, A. (1995). "Learning with ensembles: how over-fitting can be useful," in *Proceedings of the 8th International Conference on Neural Information Processing Systems, NIPS'95* (Cambridge, MA: MIT Press), 190–196.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). "Going deeper with convolutions," in *Computer Vision and Pattern Recognition (CVPR)*.

Tan, M., and Le, Q. (2019). "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research*, eds K. Chaudhuri and R. Salakhutdinov (Long Beach, CA: PMLR), 6105–6114.

Too, E. C., Yujian, L., Njuki, S., and Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric*. 161, 272–279. doi: 10.1016/j.compag.2018.03.032

Ümit Atila, Uçar, M., Akyol, K., and Uçar, E. (2021). Plant leaf disease classification using efficientnet deep learning model. *Ecol. Inform*. 61, 101182. doi: 10.1016/j.ecoinf.2020.101182

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

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., et al. (2020). Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell*. 43, 3349–3364. doi: 10.1109/TPAMI.2020.2983686

Weiss, K., Khoshgoftaar, T., and Wang, D. (2016). A survey of transfer learning. *J. Big Data* 3, 9. doi: 10.1186/s40537-016-0043-6

Wetterich, C. B., Kumar, R., Sankaran, S., Junior, J. B., Ehsani, R., and Marcassa, L. G. (2013). "A comparative study on application of computer vision and fluorescence imaging spectroscopy for detection of citrus huanglongbing disease in usa and Brazil," in *Laser Science* (Orlando, FL: Optical Society of America), JW3A-26.

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

Zhuang, J., Tang, T., Ding, Y., Tatikonda, S., Dvornek, N., Papademetris, X., et al. (2020). "Adabelief optimizer: Adapting stepsizes by the belief in observed gradients," in *Conference on Neural Information Processing Systems*.

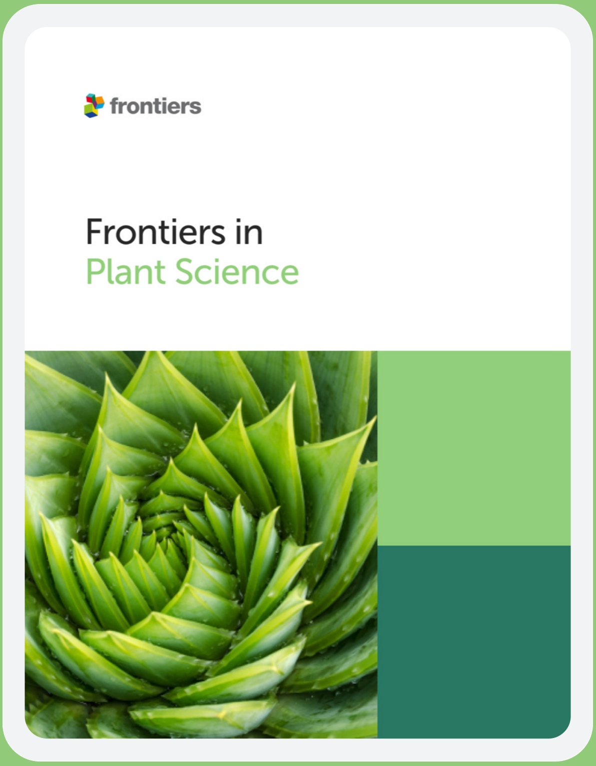
# Frontiers in
# Plant Science

**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

Frontiers in
Plant Science

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us
+41 (0)21 510 17 00
frontiersin.org/about/contact