Identification of tomato leaf diseases using convolutional neural network with multi-scale and feature reuse

Peng Li[1], Nan Zhong[1*], Wei Dong[2], Meng Zhang[1,2], Dantong Yang[1]

(1. *College of Engineering, South China Agricultural University, Guangzhou 510642, China*;
2. *Agricultural Economy and Information Research Institute, Anhui Academy of Agricultural Sciences, Hefei 230031, China*)

Abstract: Various diseases seriously affect the quality and yield of tomatoes. Fast and accurate identification of disease types is of great significance for the development of smart agriculture. Many Convolution Neural Network (CNN) models have been applied to the identification of tomato leaf diseases and achieved good results. However, some of these are executed at the cost of large calculation time and huge storage space. This study proposed a lightweight CNN model named MFRCNN, which is established by the multi-scale and feature reuse structure rather than simply stacking convolution layer by layer. To examine the model performances, two types of tomato leaf disease datasets were collected. One is the laboratory-based dataset, including one healthy and nine diseases, and the other is the field-based dataset, including five kinds of diseases. Afterward, the proposed MFRCNN and some popular CNN models (AlexNet, SqueezeNet, VGG16, ResNet18, and GoogLeNet) were tested on the two datasets. The results showed that compared to traditional models, the MFRCNN achieved the optimal performance, with an accuracy of 99.01% and 98.75% in laboratory and field datasets, respectively. The MFRCNN not only had the highest accuracy but also had relatively less computing time and few training parameters. Especially in terms of storage space, the MFRCNN model only needs 2.7 MB of space. Therefore, this work provides a novel solution for plant disease diagnosis, which is of great importance for the development of plant disease diagnosis systems on low-performance terminals.

Keywords: tomato diseases, convolutional neural network, confusion matrix, multi-scale, feature reuse
DOI: 10.25165/j.ijabe.20231606.6913

Citation: Li P, Zhong N, Dong W, Zhang M, Yang D T. Identification of tomato leaf diseases using convolutional neural network with multi-scale and feature reuse. Int J Agric & Biol Eng, 2023; 16(6): 226–235.

1 Introduction

Tomato is rich in nutrition and has been widely processed into tomato paste, tomato juice, etc. In the process of tomato planting, a variety of diseases seriously restrict the yield and quality, such as late blight, early blight, leaf mold, mosaic virus, and spot blight[1,2]. Traditional disease identification mainly depends on farmers' experience and usually requires a high professional knowledge of agricultural producers due to the complexity of disease symptoms. However, judging the types of diseases by artificial observation is more subjective, time-consuming, and labor-consuming, which is prone to misdiagnosis. Therefore, automatically and accurately identifying plant leaf diseases has become a research hotspot[3-5].

Plant disease recognition based on traditional machine learning (ML) algorithm includes three steps: disease spot segmentation, feature extraction, and classifier recognition. The research based on this method was carried out earlier and has achieved a lot of outcomes. For example, the classical ML algorithms that were applied to identify plant diseases include Support Vector Machine (SVM)[6], *K*-nearest neighbor (KNN)[7], Random Forest (RF)[8], etc. In these studies, the disease classifiers were constructed by using manually selected features, which required a lot of time for image preprocessing and feature evaluation. Additionally, the disease symptoms would become complex due to the influence of plant type, growth stage, diseased spot, and other factors[9], leading to the difficulty of feature extraction, which limits the application effect of the above methods.

Compared to traditional ML methods, Convolutional Neural Network (CNN) has abandoned complex image preprocessing and manual feature extraction. Since 2012, researchers have proposed a variety of excellent CNN models, such as AlexNet[10], VGGNet[11], ResNet[12], GoogLeNet[13], DenseNet[14], InceptionV3[15], MobileNet[16], and Xception[17]. Research showed that CNN has comprehensively surpassed the traditional ML algorithms in image classification, target detection, and image segmentation. Nowadays, CNN has been widely used in various visual tasks. Meanwhile, the research on automatic diagnosis of plant diseases has been carried out by using CNN models. For instance, Mohanty et al.[18] trained classic models AlexNet and GoogLeNet on the public PlantVillage dataset including 14 crop species and 26 diseases. Thangaraj et al.[4] explored popular CNNs like AlexNet, DenseNet, InceptionV3, ResNet, and MobileNet in tomato leaf disease identification. Rangarajan et al.[19] applied pre-trained models AlexNet and VGG16 to classify six tomato diseases, and AlexNet produced higher accuracy than VGG16. Durmus et al.[20] utilized AlexNet and SqueezeNet models for the identification of nine tomato diseases and reported an accuracy of 95.65% for AlexNet and an accuracy of 94.3% for SqueezeNet. Zhang et al.[21] used AlexNet, GoogLeNet, and ResNet to identify tomato leaf diseases, and ResNet provided the highest accuracy of 97.28%. Ahmad et al.[22] evaluated the performance of VGG-16, VGG-19, ResNet, and Inception V3 on a laboratory-based dataset and self-collected dataset, and Inception V3 was proved to be the optimal model on both datasets. Although

Received date: 2021-07-25 **Accepted date:** 2022-06-30

Biographies: Peng Li, PhD candidate, research interest: smart agriculture, Email: gxsd_roc@163.com; **Wei Dong**, Master, Associate Researcher, research interest: agricultural information technology, Email: dw06@163.com; **Meng Zhang**, PhD candidate, research interest: smart agriculture, Email: zhangmengchn@163.com; **Dantong Yang**, Master, Associate Professor, research interest: agricultural mechanization and automation, Email: yangdt@scau.edu.cn.

***Corresponding author: Nan Zhong**, PhD, Professor, research interest: smart agriculture. College of Engineering, South China Agricultural University, Guangzhou 510642, China. Email: zhongnan@scau.edu.cn.

the above CNN models performed well in the task of plant disease identification, the fixed CNN architectures and huge model parameters limit the actual deployment in the field. In particular, AlexNet and VGGNet require huge calculation costs and storage space, so they cannot be directly deployed on mobile devices. In addition, it must be pointed out that these classic CNNs were designed for a huge number of categories and trained on high-performance platforms and large-scale datasets. For the tomato leaf diseases, the number of categories and the scale of datasets are very small, and the use of such large models is prone to over-fitting. Thus, for the small-scale tomato disease dataset or other similar disease datasets, it is necessary to design a CNN model with weight and high efficiency, which can achieve the same or even better disease recognition effect as the popular CNNs, rather than just relying on these popular CNNs.

At present, researchers have developed some lightweight or low-order CNN models for identifying plant leaf diseases. Yan et al.[23] proposed an apple disease recognition model that was built by using the global pooling layer to replace the fully connected layer of VGG16 to reduce parameters. Liu et al.[24] employed GoogLeNet's Inception module to improve AlexNet model to realize the identification of apple leaf diseases. Zhang et al.[25] designed a light-weight cucumber diseases classification model namely GPDCNN, constructed by using dilated convolution, Inception module, and global pooling to improve AlexNet model. Overall, the model training parameters can be reduced by removing the fully connected layer, reducing channels and convolution kernel size, and introducing an advanced convolution module. On the other side, some low-order CNN models have been developed for plant disease identification, but the recognition accuracy is not high. Agarwal et al.[26] trained an eight-layer CNN on a publicly available tomato disease dataset and reported an accuracy of 91.2% and a storage space of 1.5 MB. Subsequently, Agarwal et al.[27] evaluated this model on a new tomato disease dataset and proved this model was superior to traditional ML approaches as well as popular CNN models, with an accuracy of 98.4%. Geetharamani et al.[28] developed a nine-layer CNN and tested this model on the open PlantVillage dataset, with an accuracy of 96.46% for 39 plant leaf classes. Li et al.[29] developed two shallow models SCNN-KSVM and SCNN-RF for identifying plant leaf diseases, reported the two models outperformed large-scale models Xception and InceptionV3, and suggested developing a shallow CNN for plant disease identification because of its simple structure and low computational cost. In addition, the above works mainly focus on the open plant disease datasets and depend on a layer-by-layer CNN model. This study also utilizes CNN to identify tomato diseases, but the CNN architecture is different from the previous ones, and two types of datasets were considered. The major contributions of this paper can be stated as follows:

1) A light-weight CNN model (MFRCNN) was developed for the identification of tomato leaf diseases, the model was established by the multi-scale convolution and feature reuse structure rather than simply stacking convolution layer by layer.

2) Two types of tomato leaf disease datasets were built, including a laboratory-based dataset and a field-based dataset.

3) Various models were tested on laboratory and field datasets. It turned out that the MFRCNN model outperformed popular CNN models on the metrics of accuracy, precision, recall, and F1-score, with fewer training parameters.

2 Materials and methods

2.1 Data acquisition

A categorized tomato disease dataset is an essential part of evaluating the model performance. Some institutions have developed the standard laboratory-based tomato leaf disease dataset, but these samples were collected in simple background and controlled environment, which deviates from the actual environment. In this work, two types of datasets were considered, one was collected from the laboratory environment, and the other was collected in the field environment.

The laboratory-based data was the public PlantVillage dataset (https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset), including one healthy class and nine disease classes, each category is shown in Figure 1. Because the number of images in each category was not uniformly distributed, the trained model might be biased in identifying different leaf diseases[27]. Data augmentation technology was used to solve this situation, including resizing, flipping, mirroring, and rotating. Afterwards, the total number of tomato leaf images was 17 197 in the laboratory-based dataset.

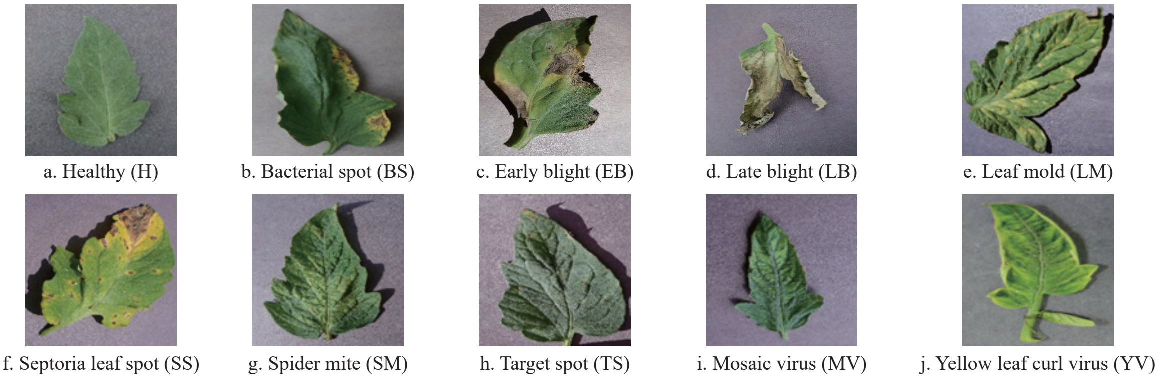
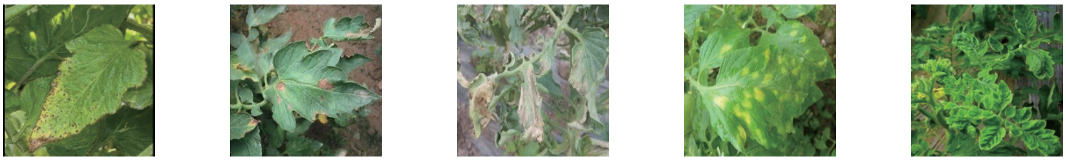

a. Healthy (H) b. Bacterial spot (BS) c. Early blight (EB) d. Late blight (LB) e. Leaf mold (LM)
f. Septoria leaf spot (SS) g. Spider mite (SM) h. Target spot (TS) i. Mosaic virus (MV) j. Yellow leaf curl virus (YV)

Figure 1 Samples in laboratory-based dataset obtained from PlantVillage

The field-based dataset contains five tomato leaf diseases, as shown in Figure 2. Collecting leaf disease images from the field is a challenging task. A total number of 326 disease images were collected in the natural environment using a mobile phone camera (12 megapixels, MI MIX 3). A few numbers of images were not enough to train a deep-learning model. Thus the data augmentation method was employed to raise the number of samples, including resizing, flipping, mirroring, rotating, adjusting brightness and contrast, etc. Finally, the total number of disease images was 4531 in field-based dataset. In addition, all images were uniformly adjusted to 256×256 pixels to be suitable for the input size of the CNN model.



a. BS b. EB c. LB d. LM e. YV

Figure 2 Samples in field-based dataset collected in the natural environment by phone camera

2.2 Datasets

In this work, each dataset was divided into a training set, validation set, and testing set. The details of the laboratory-based dataset and field-based dataset for training, validation, and testing are listed in Table 1. The previous work mainly tested the proposed model on public datasets, which contain images obtained in a controlled environment or perfect conditions. However, in the field scene, it is impossible to acquire such high-quality leaf images. In this study, the field dataset was used to further verify the performance of the model.

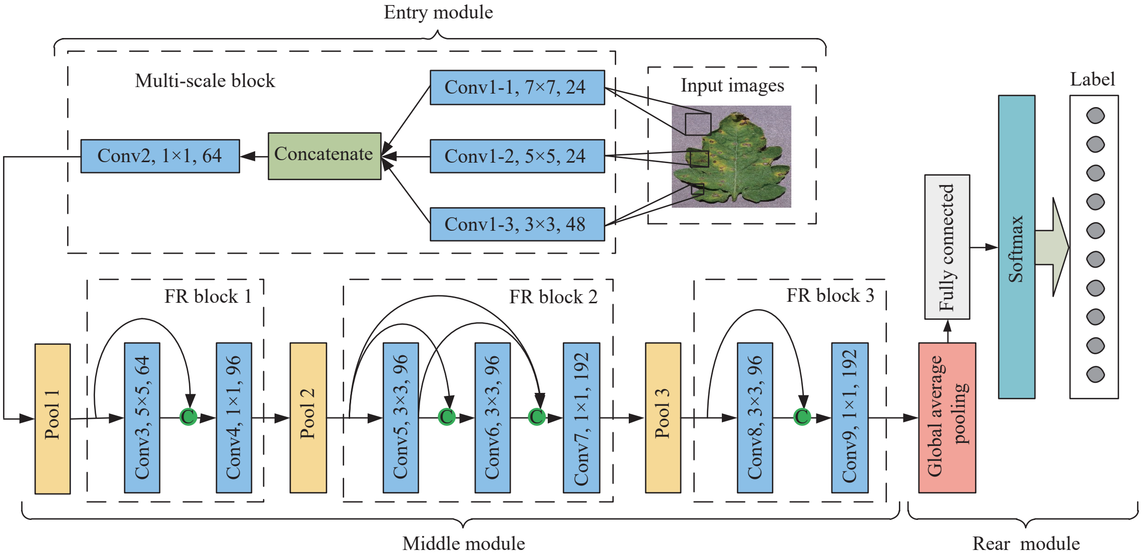
2.3 Proposed model

Inspired by the structure of AlexNet, GoogLeNet, and DenseNet, a lightweight CNN model named MFRCNN was constructed for the diagnosis of tomato leaf diseases. In Figure 3, the proposed MFRCNN includes 3 modules: the first module is the "entry module", its first convolution layer with kernels of size 3×3, 5×5, and 7×7, respectively, and the number of each kernel is 48, 24, and 24, respectively. Then, the output features of three kernels are concatenated into a tensor followed by 64 convolution kernels of size 1×1 to fuse the extracted feature information and reduce the computational complexity of the model. The second module is the "middle module", its core part is feature reuse structure, including FR Block1, FR Block2, and FR Block3. Specifically, the FR Block1 consists of Conv3 and Conv4, with 64 kernels of size 5×5 and 96 kernels of size 1×1, respectively. The FR Block2 consists of Conv5, Conv6, and Conv7, with 96 kernels of size 3×3, 96 kernels of size 3×3, and 192 kernels of size 1×1, respectively. The FR Block3 consists of Conv8 and Conv9, with 96 kernels of size 3×3 and 192 kernels of size 1×1, respectively. The last module is the "rear module", which is composed of a global average pooling layer, a fully connected layer, and a 10-way softmax layer.

Table 1 Details of the laboratory-based dataset and field-based dataset

Class name	Training set		Validation set		Testing set		Total	
	Laboratory	Field	Laboratory	Field	Laboratory	Field	Laboratory	Field
H	1146	--	318	--	127	--	1591	--
BS	1226	630	340	180	136	90	1702	900
EB	1233	641	342	183	136	91	1711	915
LB	1268	588	351	168	140	84	1759	840
LM	1274	663	353	189	141	94	1768	946
SS	1276	--	354	--	141	--	1771	--
SM	1207	--	335	--	134	--	1676	--
TS	1236	--	343	--	137	--	1716	--
MV	1272	651	353	186	141	93	1766	930
YV	1251	--	347	--	139	--	1737	--
Total	12 389	3173	3436	906	1372	452	17 197	4531

Note: The laboratory dataset contains nine leaf diseases, while the field dataset contains five leaf diseases.

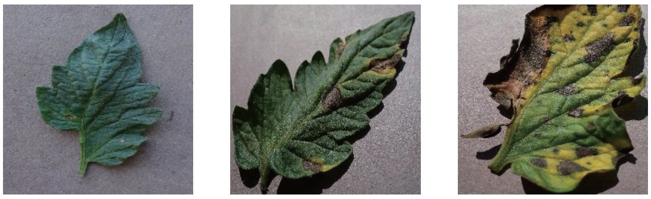


Note: Conv: Convolution; FR: Feature reuse.

Figure 3 Architecture diagram of the MFRCNN model for the diagnosis of tomato leaf diseases

2.3.1 Multi-Scale structure

Generally speaking, the small-scale kernel extracts low-frequency content but high-frequency details of the image are easily lost, while the large-scale kernel extracts big features information of the image but low-frequency contents are often lost. For tomato leaf diseases, the same leaf disease also has significant differences in different infection stages. Figure 4 gives the early, mid, and late infected leaf images of tomato early blight. As seen, the diseased leaves show obvious differences in contour, color, and texture. Therefore, multi-scale kernels are applied in the first layer of CNN to obtain the low-frequency and high-frequency details of the input leaf image. The proposed multi-scale block includes three different convolution kernels with the size of 3×3, 5×5, and 7×7 respectively. All convolution operations are performed at the same time, and then all feature maps are merged together and sent to the layer with a kernel size of 1×1 to complete feature fusion and dimensionality reduction.



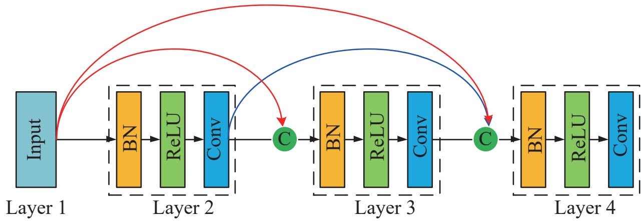
a. Early b. Mid c. Late

Figure 4 Infected images of different stages of early blight

2.3.2 Feature reuse structure

The traditional CNNs usually extract image features layer by layer, so that there is no connection between non-adjacent layers. As CNNs become increasingly deep, the gradient disappearance phenomenon will become more and more obvious. At present, many papers have reported solutions to this problem, and the core idea is to create a short path from the early layer to the later layer[30]. However, the DenseNet showed a novel perspective that each layer can directly access the later layers, and its performance also exceeded the classic model ResNet. However, DenseNet seriously consumes GPU memory, which is a challenge for low-performance computing platforms. For this reason, this study only refers to the design idea in model.

As shown in Figure 5, the feature extraction structure that the current layer can accept the previous *n* layers as additional input was constructed. It is worth noting that the number of input feature maps in the last layer was the sum of the previous *n* layers, which was not conducive to feature extraction in the next stage. Therefore, Layer4 with a small size convolution kernel 1×1 was introduced to reduce the number of feature maps, which not only reduced the amount of calculation but also fused the feature maps of all input channels. In addition, the batch normalization (BN) algorithm[31] was introduced to accelerate the convergence of the model, and the rectified linear unit (ReLU)[32] was selected as the activation function.



Note: BN: Batch Normalization; ReLU: Rectified Linear Unit.

Figure 5 Feature reuse structure

In Figure 5, the feature extraction also can be described by mathematical expression, and the output of feature reuse structure in the *l* layer is shown in Equation (1).

$$x_l = H_l([x_1, x_2, \ldots, x_{l-1}]) \tag{1}$$

where, $H_l(\cdot)$ is the nonlinear transfer function; x_1, x_2, x_3, …, x_l are the output of different layers.

2.4 Performance evaluation metrics

2.4.1 Basic metrics

In the training phase, average recognition accuracy is the primary criterion in judging model performance, and the higher the accuracy is, the better the performance is. Many related works paid more attention to the recognition accuracy and ignored other performance metrics. In view of this, considering the deployment and model calculation on low-cost terminals, the model size also was used as a model evaluation metric. In addition to accuracy and model size, training time was an important metric for evaluating the model performance. If the model has a high accuracy, few parameters, small size, and short training time, it is more conducive to the application of the model in practice.

2.4.2 Additional metrics

The Precision, recall, and F1-score were employed to comprehensively evaluate model classification performance. Among them, precision, recall, and F1-score were derived from the sample number of true positive (TP), false positive (FP), true negative (TN), and false negative (FN) results. These metrics were calculated as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{3}$$

$$\text{F1-score} = \frac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}} \tag{4}$$

where, Precision is the proportion of all positive predictions that are correct; Recall is the proportion of all real positive observations that are correct; F1-score is the harmonic mean of Precision and Recall; TP means the prediction is positive and the ground truth is positive FP means he prediction is positive and the ground truth is negative; FN is the prediction is negative and the ground truth is positive.

3 Results

Reference [33] reports that models like AlexNet, GoogLeNet, VGG16, and ResNet were the most frequently used for plant disease identification. Hence, several popular CNN models include AlexNet, SqueezeNet[34], VGG16, ResNet18, and GoogLeNet were used to verify the effectiveness of the MFRCNN.

3.1 Experimental parameters

The software environment is Ubuntu16.04LTS system, installed Caffe deep learning framework and Python2.7. The hardware platform is a computer equipped with Intel® Core™ i7-9700 @ 3.0 GHz processor, 32 GB memory, and NVIDIA GTX1080Ti.

Stochastic Gradient Descent (SGD) was selected as the optimization algorithm, and the momentum parameter was set to 0.9. The initial learning rate was set to 0.005, the training batch size was set to 32, the test batch size was set to 50, and the maximum iteration was set to 20 000 times.

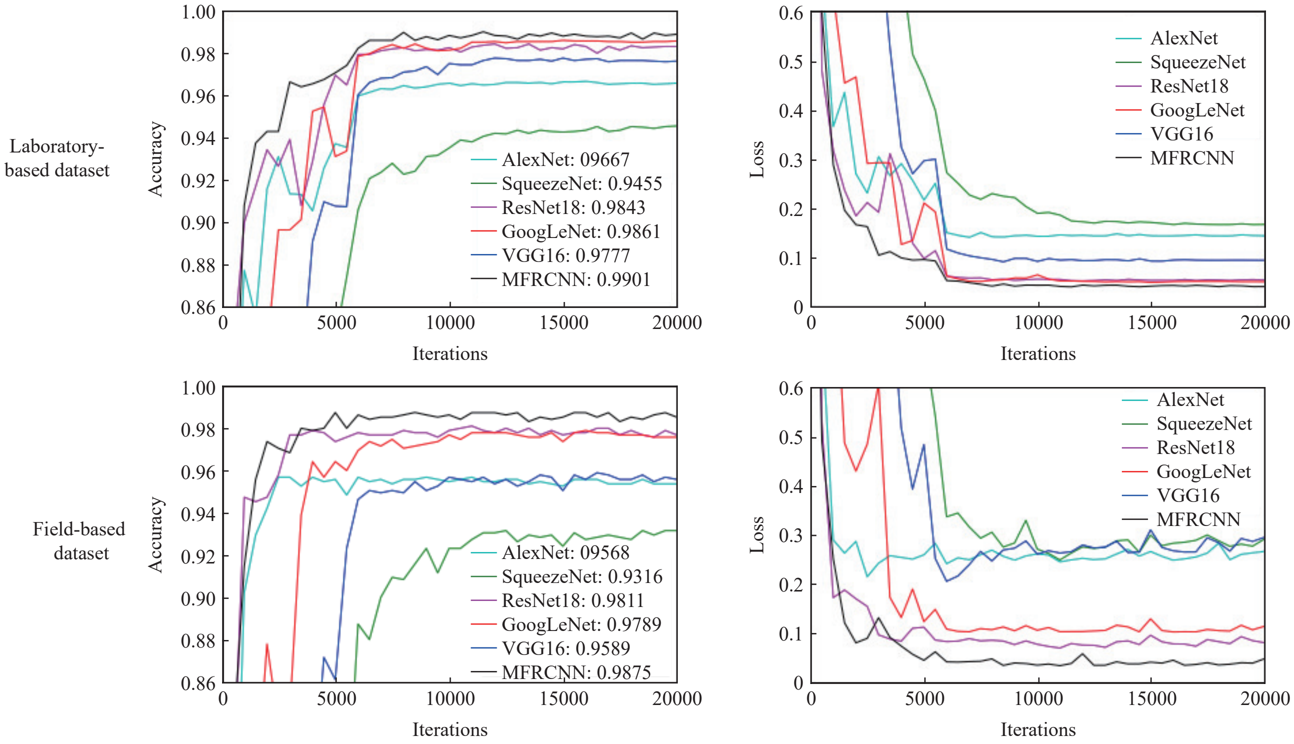
3.2 Training results

Table 2 lists the average accuracy, training time, parameters, and model size of different CNNs. The results on both datasets showed that the proposed MFRCNN model achieved the optimal performance, with the highest accuracy of 99.01% on the laboratory dataset and accuracy of 98.75% on the field dataset, which also indicated the multi-scale structure of the first layer made a better respond to different disease characteristics, and feature reuse structure effectively integrated high and low order features. In fact, the feature reuse structure could be regarded as a special multi-scale feature information fusion, which strengthened the communication between layers and reduced model parameters to a certain extent. In addition, Figure 6 shows the accuracy and loss of MFRCNN as well as five popular models, including AlexNet, SqueezeNet, VGG16, ResNet18, and GoogLeNet. It can be observed that the loss values decreased with the increase in accuracy. After 20 000 iterations, the accuracy and loss value of all models tend to stabilize. MFRCNN achieved the highest accuracy and lowest loss value than other models.

Table 2 Performance comparison of different models

Model	Average accuracy/%		Parameters /Millions	Training time/s	Model size/MB
	Laboratory	Field			
AlexNet	96.67	95.68	~56.91	~901	~222.0
SqueezeNet	94.55	93.16	~0.73	~761	~2.9
VGG16	97.77	95.89	~134.30	~7693	~537.2
ResNet18	98.43	98.11	~11.19	~2441	~44.8
GoogLeNet	98.61	97.89	~5.98	~1875	~24.0
MFRCNN	99.01	98.75	~0.66	~712	~2.7

Table 2 also compares the six models with the number of model parameters, training time, and model size. Figure 7 shows the accuracy and parameter comparison of different models on the validation set of the laboratory dataset and field dataset. It is concluded that the MFRCNN obtained the optimal performance in each metric. On the contrary, the SqueezeNet is a lightweight model, its parameters, training time and model size are slightly more than MFRCNN, but it produced the lowest accuracy on the two datasets. The VGG16 has the most parameters, training time, and model size, but its accuracy on each of the datasets is lower than that of MFRCNN. Both ResNet18 and GoogLeNet could produce satisfactory accuracy on the two datasets, but their parameters, training time, and model size are more than MFRCNN. In fact, the reason why AlexNet and VGG16 have such a large model size is that there are three fully connected layers at the end of the two models, resulting in a mount of model parameters. In general, although the MFRCNN has relatively few model parameters and a short training time, it produced the highest classification accuracy compared to other models. Thus the proposed model MFRCNN with few parameters and high accuracy can provide an important reference for the development of a tomato disease diagnosis system on mobile terminal.



a. Accuracy

b. Loss

Figure 6 Accuracy and loss of different models on laboratory dataset and field dataset

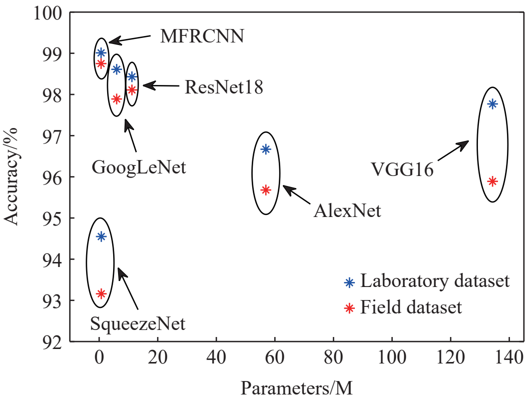


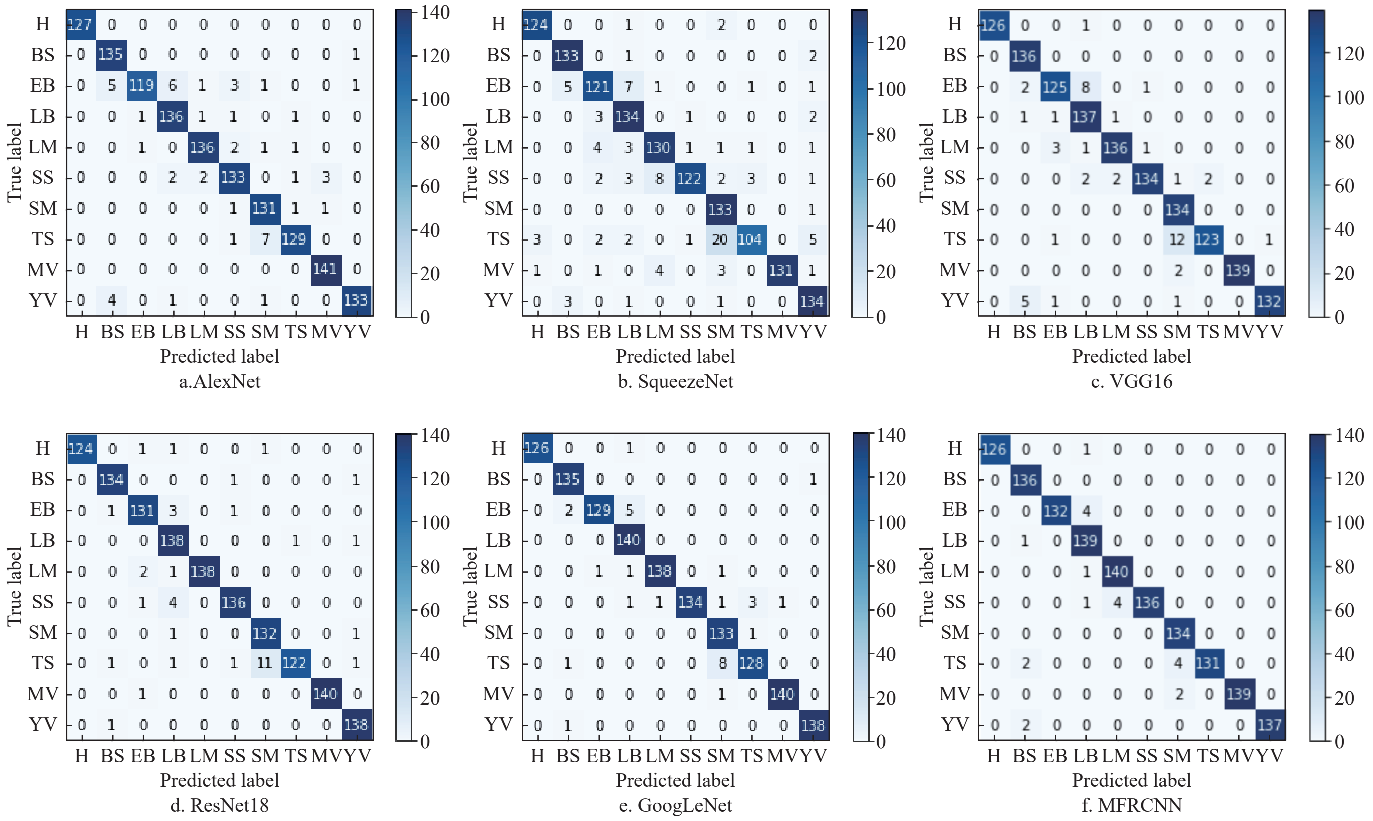
Figure 7 Accuracy and parameters comparison of different models on the laboratory and field dataset

3.3 Testing results

3.3.1 Confusion matrix of different models

The accuracy may be an unreliable performance metric for assessing the identification ability of the model, especially if the samples are unevenly distributed. Thus, the confusion matrix was introduced to further observe the prediction performance of all models for each category. All models were tested on the testing set of the two datasets, and the confusion matrices of all models are shown in Figure 8 and Figure 9. For the laboratory dataset, to intuitively observe the abilities of each model in predicting each disease category, the number of mispredicted samples of each category was counted, as shown in Figure 10.

In Figure 8 and Figure 10, it can be seen that the prediction ability of the six models for each disease category was quite different, but all models have a common ground in that they were difficult to distinguish the diseases EB, SS, and TS. Similarly, for the field dataset, Figure 9 shows that the disease EB is also difficult to identify. In Figure 8, H and YV were completely and correctly recognized by AlexNet. BS and SM were wholly and rightly identified by the MFRCNN as well as the VGG16. LB was completely and rightly distinguished by GoogLeNet. In Figure 9, YV was completely and rightly identified by the MFRCNN, GoogLeNet, and ResNet18.



True label	H	BS	EB	LB	LM	SS	SM	TS	MV	YV
H	127	0	0	0	0	0	0	0	0	0
BS	0	135	0	0	0	0	0	0	0	1
EB	0	5	119	6	1	3	1	0	0	1
LB	0	0	1	136	1	1	0	1	0	0
LM	0	0	1	0	136	2	1	1	0	0
SS	0	0	0	2	2	133	0	1	3	0
SM	0	0	0	0	0	1	131	1	1	0
TS	0	0	0	0	0	1	7	129	0	0
MV	0	0	0	0	0	0	0	0	141	0
YV	0	4	0	1	0	0	1	0	0	133

Predicted label
a.AlexNet

True label	H	BS	EB	LB	LM	SS	SM	TS	MV	YV
H	124	0	0	1	0	0	2	0	0	0
BS	0	133	0	1	0	0	0	0	0	2
EB	0	5	121	7	1	0	0	1	0	1
LB	0	0	3	134	0	1	0	0	0	2
LM	0	0	4	3	130	1	1	1	0	1
SS	0	0	2	3	8	122	2	3	0	1
SM	0	0	0	0	0	0	133	0	0	1
TS	3	0	2	2	0	1	20	104	0	5
MV	1	0	1	0	4	0	3	0	131	1
YV	0	3	0	1	0	0	1	0	0	134

Predicted label
b. SqueezeNet

True label	H	BS	EB	LB	LM	SS	SM	TS	MV	YV
H	126	0	0	1	0	0	0	0	0	0
BS	0	136	0	0	0	0	0	0	0	0
EB	0	2	125	8	0	1	0	0	0	0
LB	0	1	1	137	1	0	0	0	0	0
LM	0	0	3	1	136	1	0	0	0	0
SS	0	0	0	2	2	134	1	2	0	0
SM	0	0	0	0	0	0	134	0	0	0
TS	0	0	1	0	0	0	12	123	0	1
MV	0	0	0	0	0	0	2	0	139	0
YV	0	5	1	0	0	0	1	0	0	132

Predicted label
c. VGG16

True label	H	BS	EB	LB	LM	SS	SM	TS	MV	YV
H	124	0	1	1	0	0	1	0	0	0
BS	0	134	0	0	0	1	0	0	0	1
EB	0	1	131	3	0	1	0	0	0	0
LB	0	0	0	138	0	0	0	1	0	1
LM	0	0	2	1	138	0	0	0	0	0
SS	0	0	1	4	0	136	0	0	0	0
SM	0	0	0	1	0	0	132	0	0	1
TS	0	1	0	1	0	1	11	122	0	1
MV	0	0	1	0	0	0	0	0	140	0
YV	0	1	0	0	0	0	0	0	0	138

Predicted label
d. ResNet18

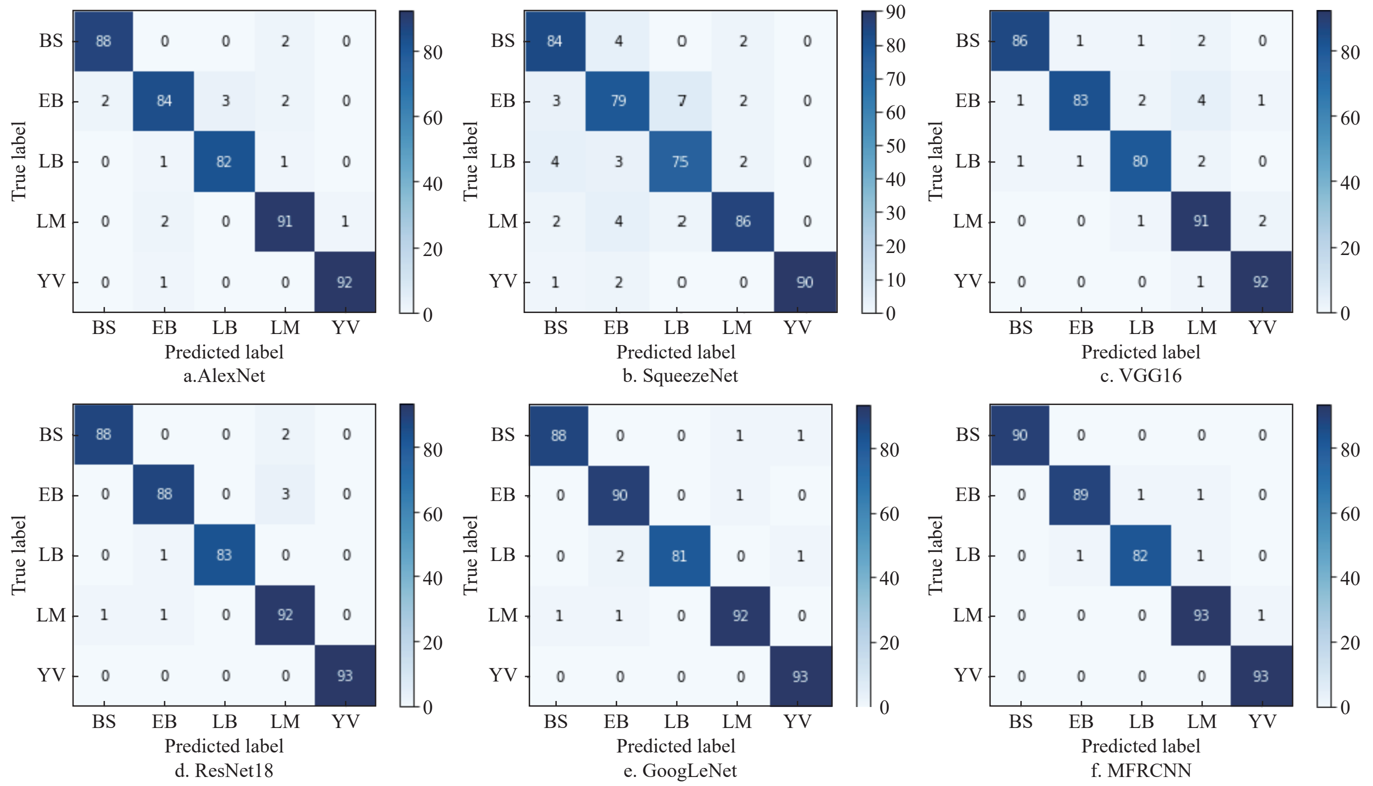
True label	H	BS	EB	LB	LM	SS	SM	TS	MV	YV
H	126	0	0	1	0	0	0	0	0	0
BS	0	135	0	0	0	0	0	0	0	1
EB	0	2	129	5	0	0	0	0	0	0
LB	0	0	0	140	0	0	0	0	0	0
LM	0	0	1	1	138	0	1	0	0	0
SS	0	0	0	1	1	134	1	3	1	0
SM	0	0	0	0	0	0	133	1	0	0
TS	0	1	0	0	0	0	8	128	0	0
MV	0	0	0	0	0	0	1	0	140	0
YV	0	1	0	0	0	0	0	0	0	138

Predicted label
e. GoogLeNet

True label	H	BS	EB	LB	LM	SS	SM	TS	MV	YV
H	126	0	0	1	0	0	0	0	0	0
BS	0	136	0	0	0	0	0	0	0	0
EB	0	0	132	4	0	0	0	0	0	0
LB	0	1	0	139	0	0	0	0	0	0
LM	0	0	0	1	140	0	0	0	0	0
SS	0	0	0	1	4	136	0	0	0	0
SM	0	0	0	0	0	0	134	0	0	0
TS	0	2	0	0	0	0	4	131	0	0
MV	0	0	0	0	0	0	2	0	139	0
YV	0	2	0	0	0	0	0	0	0	137

Predicted label
f. MFRCNN

Figure 8 Confusion matrixes of six models on the testing set of the laboratory dataset



True label	BS	EB	LB	LM	YV
BS	88	0	0	2	0
EB	2	84	3	2	0
LB	0	1	82	1	0
LM	0	2	0	91	1
YV	0	1	0	0	92

Predicted label
a.AlexNet

True label	BS	EB	LB	LM	YV
BS	84	4	0	2	0
EB	3	79	7	2	0
LB	4	3	75	2	0
LM	2	4	2	86	0
YV	1	2	0	0	90

Predicted label
b. SqueezeNet

True label	BS	EB	LB	LM	YV
BS	86	1	1	2	0
EB	1	83	2	4	1
LB	1	1	80	2	0
LM	0	0	1	91	2
YV	0	0	0	1	92

Predicted label
c. VGG16

True label	BS	EB	LB	LM	YV
BS	88	0	0	2	0
EB	0	88	0	3	0
LB	0	1	83	0	0
LM	1	1	0	92	0
YV	0	0	0	0	93

Predicted label
d. ResNet18

True label	BS	EB	LB	LM	YV
BS	88	0	0	1	1
EB	0	90	0	1	0
LB	0	2	81	0	1
LM	1	1	0	92	0
YV	0	0	0	0	93

Predicted label
e. GoogLeNet

True label	BS	EB	LB	LM	YV
BS	90	0	0	0	0
EB	0	89	1	1	0
LB	0	1	82	1	0
LM	0	0	0	93	1
YV	0	0	0	0	93

Predicted label
f. MFRCNN

Figure 9 Confusion matrixes of six models on the testing set of the field dataset

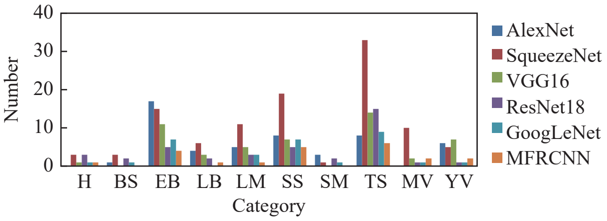


Figure 10 Number of mispredicted samples of each model on the laboratory dataset

Furthermore, although the predictive ability was different in each model, the mispredicted disease category was similar when all models predicted the same disease. For example, all models easily mistake YV for BS, which is due to the disease features being similar. Overall, the models are easily confused in identifying diseases with similar geometric features. In general, the SqueezeNet showed the worst outcome on the two datasets, with 106 mispredicted samples in the testing set of the laboratory dataset, while the MFRCNN produced the best results, with 22 mispredicted samples. MFRCNN can significantly reduce the number of mispredicted samples compared to other models.

3.3.2 Classification performance analysis

To comprehensively evaluate the classification performance of these models, the precision, recall, and F1-Score of all models on both datasets were calculated based on the above confusion matrices, as listed in Table 3. It can be concluded that MFRCNN performed better than other models on both datasets. In both circumstances, the score of MFRCNN, GoogLeNet, ResNet18, and AlexNet achieved on the field-based dataset is superior to that achieved on the laboratory-based dataset. The first reason may be that the laboratory dataset is a standard and comprehensive dataset, including the early, middle, and late stages of each disease. Despite our field dataset being collected in natural conditions, most of the diseases were in late stage and had obvious symptoms, which means that it was easier for the model to learn the key features used to distinguish different diseases. Another reason may be that the number of samples in the testing set was too small to cover comprehensive disease characteristics. Overall, the MFRCNN model is more accurate and effective in identifying tomato leaf diseases.

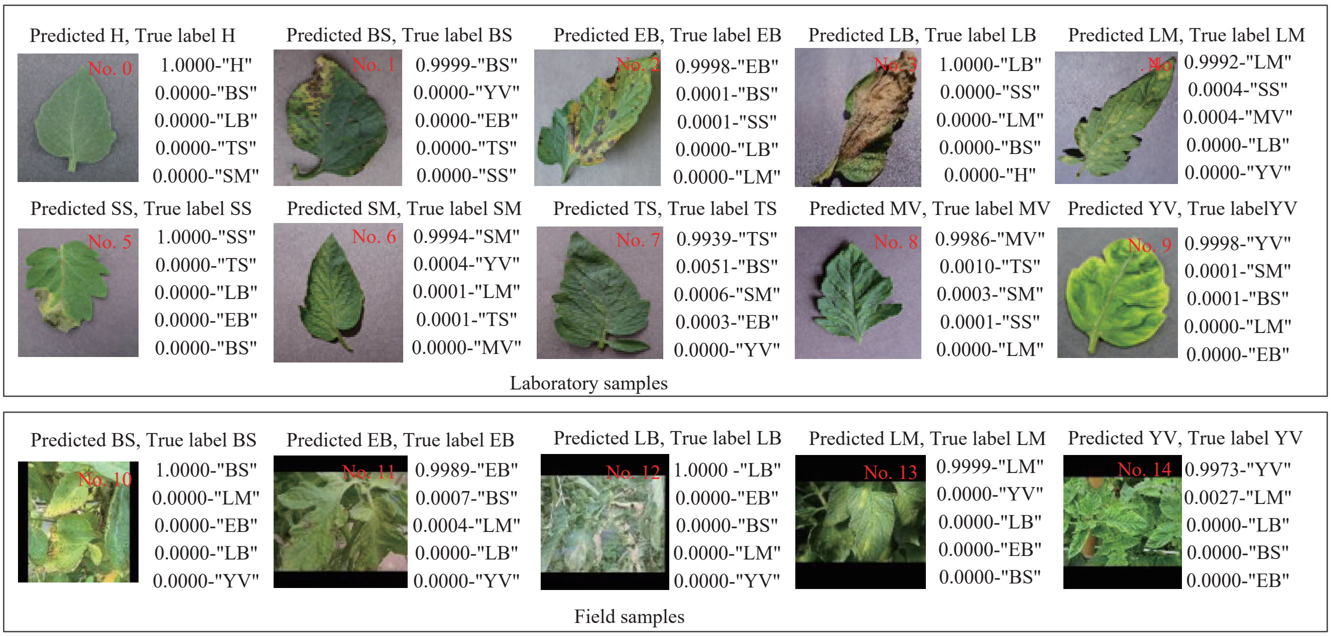
Table 3 Recall, precision, and F1-score on the laboratory dataset and field dataset

Model	Precision/%		Recall/%		F1-score/%	
	Laborat-ory	Field	Laboratory	Field	Laboratory	Field
AlexNet	96.30	96.68	96.23	96.69	96.21	96.68
SqueezeNet	92.69	91.60	92.33	91.54	92.24	91.55
VGG16	96.52	95.69	96.38	95.55	96.37	95.57
ResNet18	97.26	98.30	97.15	98.23	97.15	98.26
GoogLeNet	97.81	98.28	97.75	98.20	97.74	98.23
MFRCNN	98.46	98.90	98.41	98.87	98.41	98.88

3.4 Prediction analysis

To provide some suggestions for building high-quality dataset, this paper also summarizes the causes of sample misprediction on the two datasets. In the process of testing, each image was fed into the program, then five labels with the prediction probability were output in descending order from top to bottom and the label with the highest probability was the prediction result. Figure 11 shows an example of the correctly predicted samples on both datasets. Figure 12 presents the samples that were incorrectly predicted on both datasets.

From Figures 11 and 12, three factors affecting the model performance are summarized. One is that the shape of the leaf image is not standard (e.g., in No.18 19, and 25, the leaf accounts for a merely small space in the whole image.). The second is the color or other characteristics in different disease leaves may be similar (e.g., in No. 15, 16, and 22, they have similar contour features.). The last is the disease spots are not obvious (e.g., in No. 20, the diseased spot area in the leaf is relatively small.). These factors have important reference significance for establishing high-quality plant leaf disease data set and improving the application of deep learning technology in disease control.


Predicted H, True label H

Predicted BS, True label BS

Predicted EB, True label EB

Predicted LB, True label LB

Predicted LM, True label LM

Predicted SS, True label SS

Predicted SM, True label SM

Predicted TS, True label TS

Predicted MV, True label MV

Predicted YV, True labelYV

Laboratory samples

Predicted BS, True label BS

Predicted EB, True label EB

Predicted LB, True label LB

Predicted LM, True label LM

Predicted YV, True label YV

Field samples

Figure 11 Example of correctly predicted samples on both datasets

3.5 Feature visualization

Feature visualization can provide help in debugging the CNN model[35]. Figure 13 is the visualization of the different layers of the MFRCNN. In Figure 13, the shallow Conv1-3 and Relu1-3 extracted obvious edge and contour features from the input image. In the middle layer (Pool1 and Conv6), the feature maps gradually became blurred, which indicated that the local information of the image began to be highlighted, such as shape and texture information in the diseased area. In the high-order layer (Conv8), the feature maps have become more abstract, which means the MFRCNN had a strong response to the diseased area and extracted plenty of disease feature information.

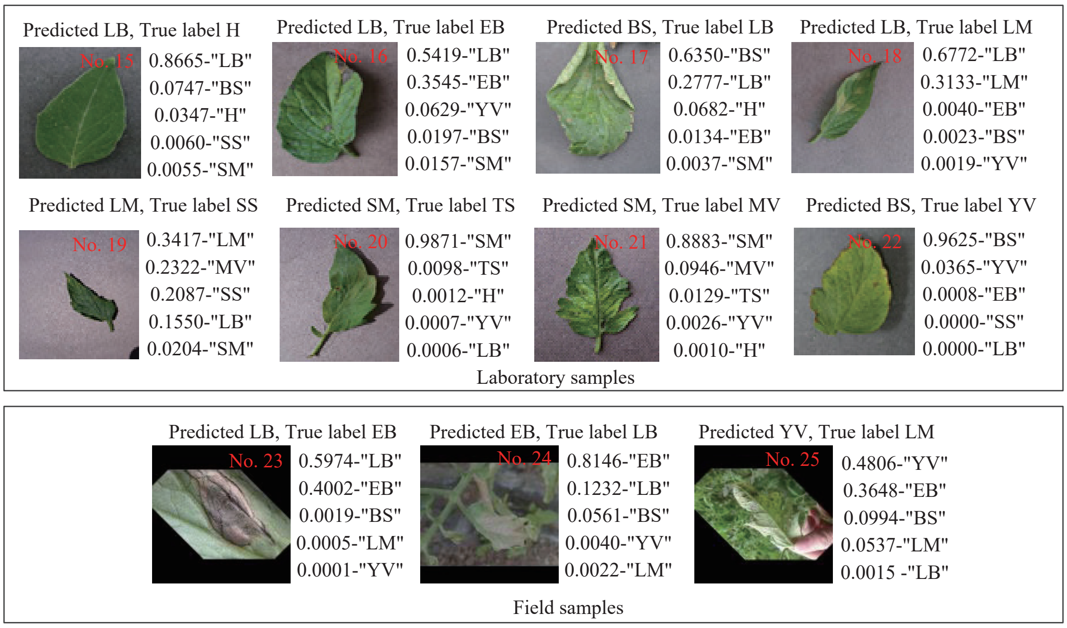
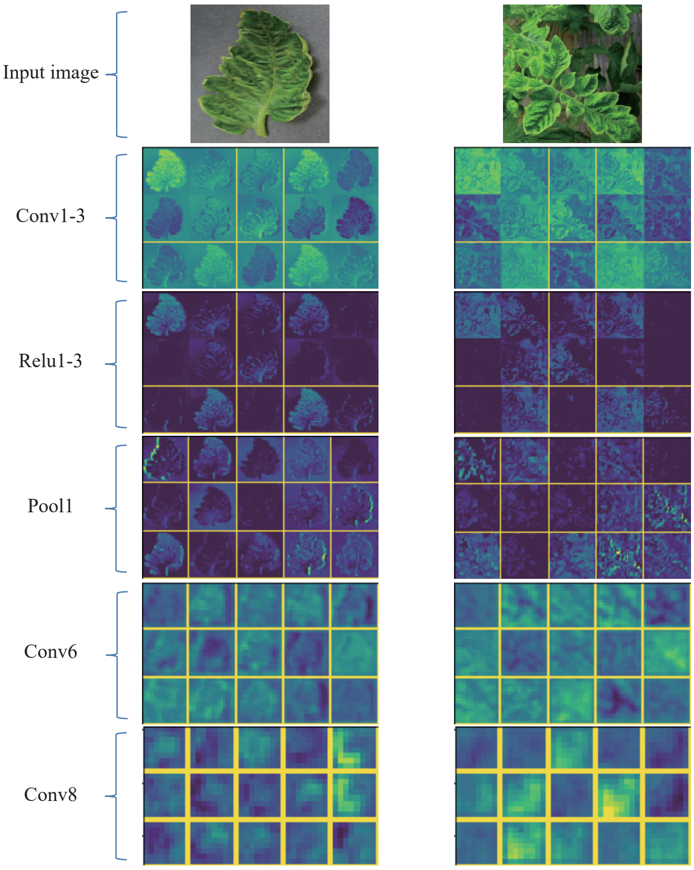


Figure 12 Example of incorrectly predicted samples on both datasets



a. Input image 1 and the visualization of different layers b. Input image 2 and the visualization of different layers

Figure 13 Feature visualization of the input image using MFRCNN

4 Discussion

In this work, several popular CNN models (AlexNet, SqueezeNet, VGG16, ResNet18, and GoogLeNet) were employed to identify tomato diseases. Because these models are designed to classify 1000 categories of the Imagenet dataset, the scale of these models is very large, and each convolution layer has a large number of kernels[27]. Although these models have achieved good results on laboratory and field datasets, they were implemented at the cost of massive calculation time and huge storage space. From the perspective of the future development of smart agriculture, embedded devices are more competitive than high-performance computers and servers in field applications, especially considering portability, economy, and practicality[29]. Therefore, the light-weight CNN model is more suitable than the above models. Fortunately, this study developed a lightweight model MFRCNN, which can provide the basis for the development of disease identification systems in mobile terminals.

As listed in Table 4, researchers used the popular models to classify the laboratory-based tomato leaf diseases. It can be seen that the accuracy of most models is lower than that of the MFRCNN model. In particular, Agarwal et al.[27] also proved that simplified CNN is better than deep models such as VGG16, Inception-V3, and MobileNet. It is noted that this simplified CNN achieved 98.4%, which is lower than the 99.01% obtained by MFRCNN. In addition, deep models like VGG16, MobileNet, and Inception-V3 have millions of training weights. If these models are trained on small-scale datasets, the over-fitting is easier to occur. The MFRCNN model has very few weight parameters, which cannot only save storage space but also prevent over-fitting. Overall, compared with some popular models, the proposed MFRCNN has a good ability to deal with general tomato leaf disease identification tasks and obtains higher accuracy.

Table 4 Accuracy comparison on the laboratory-based dataset

Literature	Class	Samples	Model	Accuracy/%
Durmus et al.[20]	10	18 160	AlexNet SqueezeNet	95.65 94.30
Zhang et al.[21]	9	41 127	AlexNet GoogLeNet ResNet	95.83 95.66 97.28
Rangarajan et al.[19]	7	13 262	AlexNet VGG16	97.29 97.49
Wu et al.[36]	5	4300	AlexNet GoogLeNet ResNet VGG16	89.67 94.33 83.00 79.30
Agarwal et al.[27]	10	18 160	VGG16 Inception-V3 MobileNet Simplified CNN	93.50 77.50 82.60 98.40
Singh et al.[37]	10	18 160	ResNet50 DenseNet121 DenseNet201	98.15 98.54 98.95
Proposed method in this study	10	17 197	AlexNet SqueezeNet VGG16 ResNet18 GoogLeNet MFRCNN	96.67 94.55 97.77 98.43 98.61 99.01

CNN is a powerful and important technology for plant leaf disease identification. As a suggestion, this work calls on more researchers to develop lightweight CNN models instead of relying solely on popular models, especially considering the simple architecture, few parameters, and high accuracy. Further, it can be considered to establish lightweight models by combining advanced convolution modules, such as attention mechanisms and depthwise separable convolutions.

5 Conclusions

This study developed a lightweight CNN model MFRCNN for the identification of tomato leaf diseases. The success of the proposed MFRCNN was compared with the state-of-the-art CNN models on the laboratory-based dataset and field-based dataset using multiple performance metrics. The results show that the performance of the proposed CNN model was better than that of popular CNNs on both datasets. The proposed CNN model with fewer parameters, shorter training time, and higher recognition accuracy, and it provides a theoretical basis for the development of the automatic disease diagnosis system based on low-cost terminals. It is suggested that the lightweight CNN should be first developed for plant disease diagnosis because of low computational costs, good performance, small storage space, etc. If lightweight CNNs can not meet the needs, then further consider using popular CNNs.

Acknowledgements

This work was financially supported by the China Agriculture Research System (CARS-170 404), Qingyuan Science and Technology Plan (Grant No. 2022KJJH063), and Guangzhou Science and Technology Plan (Grant No. 201 903 010 063).

[References]

[1] Hassan S M, Maji A K, Jasiński M, Leonowicz Z, Jasi´nska E. Identification of plant-leaf diseases using CNN and transfer-learning approach. Electronics, 2021; 10(12): 1388.

[2] Karthik R, Hariharan M, Anand S, Mathikshara P, Johnson A, Menaka R. Attention embedded residual CNN for disease detection in tomato leaves. Applied Soft Computing, 2020; 86: 105933.

[3] Bhujel A, Kim N E, Arulmozhi E, Basak J K, Kim H T. A lightweight Attention-based convolutional neural networks for tomato leaf disease classification. Agriculture, 2022; 12(2): 228.

[4] Thangaraj R, Anandamurugan S, Pandiyan P, Kaliappan K V. Artificial intelligence in tomato leaf disease detection: a comprehensive review and discussion. Journal of Plant Diseases and Protection, 2022; 129(3): 469–488.

[5] Ahad M T, Li Y, Song B, Bhuiyan T. Comparison of CNN-based deep learning architectures for rice diseases classification. Artificial Intelligence in Agriculture, 2023; 9: 22–35.

[6] Mokhtar U, Ali M A S, Hassenian A E, Hefny. Tomato leaves diseases detection approach based on Support Vector Machines. In: 2015 11th International Computer Engineering Conference (ICENCO), Cairo: IEEE, 2015; pp.246–250. doi: 10.1109/ICENCO.2015.7416356.

[7] Tian K, Li J H, Zeng J F, Evans A, Zhang L N. Segmentation of tomato leaf images based on adaptive clustering number of K-means algorithm. Computers and Electronics in Agriculture, 2019; 165: 104962.

[8] Chaudhary A, Kolhe S, Kamal R. An improved random forest classifier for multi-class classification. Information Processing in Agriculture, 2016; 3(4): 215–222.

[9] Waheed A, Goyal M, Gupta D, Khanna A, Hassanien A E, Pandey H M. An optimized dense convolutional neural network model for disease recognition and classification in corn leaf. Computers and Electronics in Agriculture, 2020; 175: 105456.

[10] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 2017; 60(6): 84–90.

[11] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 3rd Int. Conf. Learn. Represent. ICLR 2015-Conf. Track Proc., 2014; pp.1–14.

[12] He K M, Zhang X Y, Ren S Q, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas: IEEE, 2016; pp.770–778. doi: 10.1109/CVPR.2016.90.

[13] Szegedy C, Liu W, Jia Y Q, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston: IEEE, 2015; pp.1–9. doi: 10.1109/CVPR.2015.7298594.

[14] Huang G, Liu Z, Van Der Maaten L, Weinberger K Q. Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu: IEEE, 2017; pp.2261–2269,doi: 10.1109/CVPR.2017.243.

[15] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas: IEEE, 2016; pp.2818–2826. doi: 10.1109/CVPR.2016.308.

[16] Howard A G, Zhu M L, Chen B, Kalenichenko D, Wang W J, Weyand T, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv, 2017; arXiv: 1704.04861.

[17] Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu: IEEE, 2017; pp.1800–1807. doi: 10.1109/CVPR.2017.195.

[18] Mohanty S P, Hughes D P, Salathé M. Using deep learning for image-based plant disease detection. Frontiers in Plant Science, 2016; 7: 1419.

[19] Rangarajan, A K, Purushothaman R, Ramesh A. Tomato crop disease classification using pre-trained deep learning algorithm. Procedia Computer Science, 2018; 133: 1040–1047.

[20] Durmus H, Gunes E O, Kirci M. Disease detection on the leaves of the tomato plants by using deep learning. In: 2017 6th International Conference on Agro-Geoinformatics, Fairfax: IEEE, 2017; pp.1–5. doi: 10.1109/Agro-Geoinformatics.2017.8047016.

[21] Zhang K K, Wu Q F, Liu A W, Meng X Y. Can deep learning identify tomato leaf disease? Advances in Multimedia, 2018; 2018: 6710865. doi: 10.1155/2018/6710865.

[22] Ahmad I, Hamid M, Yousaf S, Shah S T, Ahmad M O. Optimizing pretrained convolutional neural networks for tomato leaf disease detection. Complexity, 2020; 2020: 8812019.

[23] Yan Q, Yang B H, Wang W Y, Wang B, Chen P, Zhang J. Apple leaf diseases recognition based on an improved convolutional neural network. Sensors, 2020; 20(12): 3535.

[24] Liu B, Zhang Y, He D J, Li Y X. Identification of apple leaf diseases based on deep convolutional neural networks. Symmetry, 2018; 10(1): 11.

[25] Zhang S W, Zhang S B, Zhang C L, Wang X F, Shi Y. Cucumber leaf disease identification with global pooling dilated convolutional neural network. Computers and Electronics in Agriculture, 2019; 162: 422–430.

[26] Agarwal M, Singh A, Arjaria S, Sinha A, Gupta S. ToLeD: Tomato leaf disease detection using Convolution Neural Network. Procedia Computer Science, 2020; 167: 293–301.

[27] Agarwal M, Gupta S K, Biswas K K. Development of efficient CNN model for Tomato crop disease identification. Sustainable Computign:Informatics and Systems, 2020; 28: 100407.

[28] Geetharamani G, Arun Pandian J. Identification of plant leaf diseases using a nine-layer deep convolutional neural network. Computers & Electrical Engineering, 2019; 76: 323–338.

[29] Li Yang, Nie J, Chao X W. Do we really need deep CNN for plant diseases identification? Computers and Electronics in Agriculture, 2020; 178: 105803.

[30] Huang G, Liu Z, Van Der Maaten L, Weinberger K Q. Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu: IEEE, 2017; pp.2261–2269. doi: 10.1109/CVPR.2017.243.

[31] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. International Conference on Machine Learning (ICML'15), 2015; 448–456.

[32] Ayan E, Erbay H, Varcin F. Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks. Computers and Electronics in Agriculture, 2020; 179: 105809.

[33] Saleem M H, Potgieter J, Arif K M. Plant disease detection and classification by deep learning. Plants, 2019; 8(11): 468.

[34] Iandola F N, Han S, Moskewicz M W, Ashraf K, Dally W J, Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv, 2016; arXiv: 1602.07360v4.

[35] Hang J, Zhang D X, Chen P, Zhang J, Wang B. Classification of plant leaf diseases based on improved convolutional neural network. Sensors, 2019; 19(19): 4161.

[36] Wu Q F, Chen Y P, Meng J. DCGAN-based data augmentation for tomato leaf disease identification. IEEE Access, 2020; 8: 98716–98728.

[37] Singh G B, Rani R, Sharma N, Kakkar D. Identification of tomato leaf diseases using deep Convolutional Neural Networks. International Journal of Agricultural and Environmental Information Systems, 2021; 12(4): oa3.