Double-branch deep convolutional neural network-based rice leaf diseases recognition and classification

Xiong Bi, Hongchun Wang

School of Mathematical Sciences, Chongqing Normal University, Chongqing, China

Abstract

Deep convolutional neural network (DCNN) has recently made significant strides in the classification and recognition of rice leaf disease. The majority of classification models perform disease image recognitions using collocation patterns including pooling layers, convolutional layers, and fully connected layers, followed by repeating this structure to complete depth increase. However, the key information of the lesion area is locally limited. That is to say, in the case of only performing feature extraction according to the above-mentioned model, redundant and low-correlation image feature information with the lesion area will be received, resulting in low accuracy of the model. For improvement of the network structure and accuracy promotion, here we proposed a double-branch DCNN (DBDCNN) model with a convolutional block attention module (CBAM). The results show that the accuracy of the classic models VGG-16, ResNet-50, ResNet50+CBAM, MobileNet-V2, GoogLeNet, EfficientNet-B1 and Inception-V2 is lower than the accuracy of the model in this paper (98.73%). Collectively, the DBDCNN model here we proposed might be a better choice for classification and identification of rice leaf diseases in the future, based on its novel identification strategy for crop disease diagnosis.

Correspondence: Hongchun Wang, School of Mathematical Sciences, Chongqing Normal University, Chongqing, 401331, China.
E-mail: wanghc@cqnu.edu.cn

Key words: convolutional block attention module (CBAM); deep convolution neural network (DCNN); rice leaf diseases classification.

Acknowledgments: this project was supported by the Philosophy and Social Science Foundation of China (No. 13BTJ008), the Key Project of Humanities and Social Sciences of Chongqing Municipal Education Commission (No. 22SKGH081), and Chongqing Education Reform Project (No. 213139).

Conflict of interest: the authors declare no potential conflict of interest.

Funding: none.

Received: 14 January 2023.
Accepted: 27 March 2023.

Journal of Agricultural Engineering 2024; LV:1544
doi:10.4081/jae.2023.1544

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.

Introduction

Rice is one of the most important food crops in the world, and its yield and quality are directly related to global food supply and livelihoods. Especially in China, rice is already an essential food crop, and its planting area and total output rank second in the world (Chakraborty and Newton, 2011; Ray *et al*., 2012; Huang *et al*., 2016; Savary *et al*., 2019). However, rice is vulnerable to diseases during growth and development, among which leaf diseases (such as Bacterial blight, Blast, Brown spot and Tungro) (Skamnioti and Gurr, 2009; Valent and Khang, 2010; Sundaram *et al*., 2014; Savary *et al*., 2019) are one of the main diseases of rice. According to statistics, due to the impact of these rice diseases, the annual rice yield usually decreases by 10-15% (Peng *et al*., 2009), and it will also cause certain damage to the surrounding soil environment. Therefore, it is urgent to develop an efficient and accurate method for rice leaf disease identification.

At present, there have been some researches on rice leaf disease identification methods, but most of them are based on traditional manual detection methods, which have problems such as low efficiency, unstable results and high misdiagnosis rate. In addition, manual diagnostic methods are particularly difficult to implement for some difficult-to-distinguish disease types. Therefore, it has extremely important theoretical and practical significance to develop an automatic identification system that can accurately identify rice leaf disease types.

This research aims to develop a rice leaf disease recognition system based on computer vision (CV) and deep learning technology (DLT) to solve the problems of low efficiency and high misdiagnosis rate of current manual diagnosis. Specifically, this paper will collect a large number of rice leaf images and combine them with deep learning algorithms to train an efficient and accurate disease identification model to realize automatic identification and classification of different types of rice leaf diseases. The research results are expected to provide a more reliable and efficient technical means for rice disease monitoring and control, which will help improve the production efficiency and quality of rice and promote the sustainable development of food production.

Currently, due to the rapid development of CV, digital image processing technology (Barbedo and Garcia, 2013) and DLT, a large number of traditional machine learning algorithms have been used to identify disease types in rice and other crops. For example, Nandhini and Bhavani (2020) first extracted color and shape features from rice diseased leaves and subsequently used traditional machine learning algorithms such as Support vector machine (SVM), Decision tree, and K-nearest neighbor to classify the extracted features, and their experiment results demonstrated that the SVM-based classification was the best. Zhang and Zhang (2010) used three SVMs with different kernel functions [including Radial basis function (RBF), polynomial function, as well as Sigmoid kernel function] for classification and identification of cucumber leaf diseases, and their experimental results indicated that SVM-based RBF kernel function had the highest recognition rate. Panchal *et al*. (2019) initially segmented the diseased areas of plant leaves using K-Means clustering and HSV color space, followed by extracting features of diseased areas according to the grey-level co-occurrence matrix, and then identifying the disease species based on random forest. However, when the above machine learning algorithm was adopted for crop pest and disease image classification, the feature extractor should be at first set up manually, in addition, many other limited difficulties (*e.g.*, limited data processing power, *etc*.) were also encountered, which negatively affect the running capacity of the model and thereby the classification accuracy. On the other hand, due to the continuous and rapid development of machine learning technology, deep learning has attracted increased attention in crop pest and disease identification field. To date, numerous researchers have conducted extensive in-depth studies involved in crop pests and diseases identifications using deep neural networks [convolutional neural networks (CNN) particularly] and corresponding optimization algorithms. For example, Jiang *et al*. (2020) integrated SVM with CNN to establish a disease recognition model, using CNN for image features extraction of rice leaf diseases, and SVM for diseases classification and prediction. The results indicated that the average accuracy of this model could reach 96.8%, and the classification effect was more satisfactory. Deb *et al*. (2021) first introduced the impact of rice diseases on agricultural production and the limitations of traditional detection methods, and then proposed the possibility of using artificial intelligence technology for early diagnosis. At the same time, the performance of 5 different CNN models (namely Inception-V3, VGG-16, AlexNet, MobileNet V2 and ResNet-18) on the rice leaf disease image dataset is introduced in detail. The authors noted that the Inception-V3 model performed best in terms of accuracy, reaching 96.23%. Deb *et al*. (2022) proposed a new CNN model LS-Net to solve the problem of rose plant leaf segmentation. The authors pointed out that traditional methods based on CNN have difficulties in processing images with multiple overlapping leaves and complex backgrounds, and cannot produce high accuracy, so the LS-Net network model is proposed. At the same time, it is compared with 4 existing CNN segmentation models (*i.e.*, DeepLab V3+, Seg Net, Fast-FCN with pyramid pooling module and U-Net). Experimental results show that the proposed model has a better segmentation effect. Chen *et al*. (2020) first investigated the transfer learning of deep CNNs, and then proposed a VGGNet-based INC-VGGN model. The model is pre-trained on ImageNet and Inception modules to identify plant leaf disease types. Compared with other state-of-the-art techniques (DenseNet-201, ResNet-50, Inception V3 and VGGNet-19), our proposed method achieves substantial performance improvement. Pandian *et al*. (2022) proposed a 5-layer CNN model based on data augmentation and hyperparameter optimization techniques to identify plant leaf disease images. Among them, image enhancement techniques include generative adversarial networks, neural style transfer, principal component analysis, color enhancement, and location enhancement. In addition, hyperparameters are optimized by random search techniques. At the same time, the proposed model was compared with advanced methods such as VGG16, Inception-v3, and ResNet-50. The experimental results show that the average accuracy of the proposed model on the test set can reach 98.41%, which is better than other methods. Kaur *et al*. (2022) Based on InceptionNet, ResNet V2 and transfer learning, a new CNN model (*i.e.*, the modified InceptionResNet-V2 model) was proposed and used to identify diseases in tomato leaf images. The model was trained and tested on public datasets Plantvillage and PlantDoc as well as custom datasets, and the final recognition accuracy rate was 98.92%. The experimental results show that the model has a good effect on the identification of tomato leaf diseases. Omer *et al*. (2022) proposed a tuned CNN model (both 1-CNN and 2-CNN implementations) and used it to identify healthy and diseased cucumber leaves. The recognition process is divided into three steps, namely data augmentation, feature extraction and classification. Among them, data augmentation is used to expand the dataset, while CNN is used for feature extraction and classification. The experimental results on the custom data set show that the proposed model has the highest accuracy rate, which is better than the accuracy rate of the comparison models (such as Inception-V3, ResNet-50 and AlexNet). Zhao *et al*. (2021) combined an attention module with a deep CNN to propose a recognition system model (SE-ResNet50) for diagnosing tomato leaf diseases. Specifically, the proposed network model mainly consists of a residual block and an attention module, capable of complex feature extraction and classification for various diseases. The results of a large number of comparative experiments on the tomato leaf disease dataset show that the average accuracy of the proposed model is 96.81%, which is higher than the comparison models (GoogleNet, ResNet-101, Xception and VGG-19). Ghosal and Sarkar (2020) adopted a migration learning approach to fine-tune a pre-trained model (VGG-16) and subsequently applied it on a rice leaf disease dataset with a small sample size, and ultimately achieving an accuracy of 92.46%. Bharali *et al.* (2019) proposed a PDDNN network system (7 conv layers+2 fc layers) for crop-disease identification and obtained an accuracy of 86% on a given dataset. Subetha *et al*. (2021) used the deep learning algorithms ResNet-50 and VGG-19 for predictive classification of apple leaf disease datasets in Kaggle, respectively. A comparison analysis results showed that the former ResNet-50 had a higher prediction accuracy, with an overall accuracy of 87.7%. Sethy *et al*. (2020) first extracted the depth features of rice leaf disease region using 11 classical CNNs (such as VGG-16, VGG-19, GoogLeNet, *etc*.) separately, and then used SVM as a classifier to classify the features. The results suggested that the combination of ResNet-50 with SVM achieved the highest accuracy, while GoogLeNet combined with SVM led to the worst classification result. These experiments have explored and tested how to choose the best network model for depth features extractions when SVM is adopted as a classifier. Waheed *et al*. (2020) further proposed an optimized DenseNet network system and simultaneously applied it with other CNN models (including VGG-19, XceptionNet, NasNet and EfficientNet-B0) for maize leaf disease dataset, respectively. Data showed that the optimized lightweight network DenseNet achieved the highest accuracy, with a far-batter prediction accuracy than other comparison controls. Rahman *et al*. (2020) designed a network model called Simple CNN with fewer parameters based on a novel training method (*i.e.*, a two-stage training method) induced from fine-tuning. They demonstrated that the model achieved 93.30% accuracy in rice pests and diseases identification, outperforming the existing MobileNet-V2, NasNetMobile, SqueezeNet-V1.1 and VGG-16. Zeng and Li (2020) proposed a Self-attentive CNN model combining self-attentive (SA) and CNN, where CNN was adopted for extracting global features of images while SA was used for obtaining local features of crop disease regions. Based on the predictive classification of maize leaf diseases according to the natural environment crop disease dataset (AES-CD9214) and the public dataset (MK-D2), obtained accuracies were 95.33% and 98.0% respectively, outperforming the current-existed classical network models. This suggests that neural network with an attention module is far better able to concentrate on most key regions within images thereby significantly and effectively promoting classification accuracy. Liu *et al*. (2018) removed some original fully connected layers and supplemented some pooling layers based on AlexNet, introducing the GoogLeNet Inception structure to ultimately develop a Deep CNN (DCNN), in which the optimization of model parameters was performed using Numerical Algorithm Group algorithm. AlexNet, GoogLeNet, VGG-16, and ResNet-20 were all adopted as comparison control models in the classification and recognition of apple leaf disease dataset. The results demonstrated that recognition accuracy could reach 97.62%, which was more significantly higher than those detected from other control models. Zhang *et al*. (2019) designed a global pooled dilation CNN system by combining the dilation convolutional layer and global pooling. By doing so, it is capable of aggregating multiscale contextual information and promoting the recognition rate. AlexNet, DCNN (Khan *et al*., 2018), a probabilistic neural network (Shi *et al*., 2015) were once incorporated as comparison control models to identify cucumber leaf disease dataset in the experimental stage, consequently, the proposed model achieved the highest accuracy (94.65%).

As we all know, the attention mechanism is an image-processing technique widely used in the field of deep learning. It mainly has the following advantages: i) it helps the neural network automatically focus on important information in the input data, thereby improving the performance of tasks such as classification or regression; ii) it can adapt to different features in the input data, so it can be flexibly applied to different types of tasks and datasets; iii) it helps researchers understand the behavior and decision-making of neural networks by providing visualization results.

Now we move on to CBAM, an attention mechanism module for convolutional neural networks. Compared with the general attention mechanism, its advantages are the following: i) based on the attention mechanism, a channel and spatial attention mechanism is introduced, which can simultaneously pay attention to the channel and spatial information in the input data, and realize adaptive feature selection and fusion without introducing additional parameters; ii) the parameters of the model are effectively reduced, and the calculation efficiency and generalization ability of the model are improved; iii) good performance has been achieved in multiple vision tasks, demonstrating the potential of the CBAM model in areas such as image classification, object detection, and image segmentation.

The relationship between CBAM and DCNN can be interpreted as the former being a submodule or plug-in of the latter, which can be embedded into different layers of DCNN to improve the performance of the network. For example, CBAM can be naturally inserted into deep models such as ResNet, VGG, Inception, *etc*., and will not have a major impact on their overall structure. In addition, the structure of DCNN is usually composed of multiple convolutional layers and pooling layers, so as to gradually extract higher-level features to realize the classification, recognition or other tasks of input data. If combined with the advantages of CBAM, it can better help the network to pay attention to the important information in the input data, and further improve the expressive ability of features and the accuracy of the model. In conclusion, CBAM is an attention mechanism module that can be embedded into DCNN, which can adaptively focus on important information in input data and further improve the performance and robustness of DCNN. Because of the close relationship between CBAM and DCNN, they can cooperate with each other to better achieve deep learning tasks.

Through the above discussion and analysis, this study proposes a new method for identifying rice leaf diseases, which is to introduce the dual-branch DCNN identification model of CBAM. We mainly focused on the following three aspects: i) we developed and proposed a novel double-branch DCNN (DBDCNN) based on residual network and verified the plausibility of its improvements in three aspects. The experimental results suggested that it could obtain high accuracy in extracting depth features of key regions and classification, implying practical significance for classification and recognition of rice disease images; ii) the effects of CBAM and secondary screening depth features on recognition performance were explored and discussed based on our experimental results compared with different models using the method proposed in this paper; iii) multiple sets of experimental results indicated that our proposed DBDCNN method presented with high accuracy of recognition and strong robustness. Compared with other deep learning models (ResNet-50, MobileNet-V2, VGG-16, ResNet50+CBAM, GoogLeNet, EfficientNet-B1, and Inception-V2), our model performed better on the rice leaf disease dataset.

Pre-requisite knowledge

Convolutional block attention module

CBAM (Chen *et al*., 2017; Woo *et al*., 2018) has been proposed as a lightweight attention mechanism that focuses not only on the attention information on the feature map channel, but also on the large amount of attention information inside the channel, hence effectively combining both channel and spatial dimensions. Compared with SENet (Hu *et al*., 2020), the Spatial Attention Module (SAM) or Channel Attention Module (CAM) alone, was more capable of emphasizing meaningful features in both channel and spatial dimensions, pointing out what and where to pay attention to in order to obtain more reliable and comprehensive information (Yu *et al*., 2020). CBAM successively integrated the spatial attention module SAM as well as the channel attention module CAM. In CAM, feature maps were initially subjected to global maximum and average pooling respectively, followed by feeding the pooling results into another double-layer fully connected neural network. Then output features were subjected to element-wise summation-based operations and sigmoid activation operations, and ultimately channel attention features $M_c(F)$ were obtained. The flow of this module can be seen in Equation (1), where F^c_{avg}, F^c_{max}, denote the features after global average and maximum pooling, respectively, W_0, and W_1 denote weight parameters in the neural network. σ represents the ultimate sigmoid activation function. In SAM, the output feature map from CAM was also subjected to the above two pooling operations. Then the two pooled features were stitched together for input into a convolutional layer for feature dimensionality reduction. The same sigmoid activation operation was subsequently performed to obtain spatial attention features $M_s(F)$. Equation (2) embodies the detailed process of SAM. *f* indicates the convolution operation using a 7×7 convolution kernel, and [ ] is the channel-based splicing operation. Finally, the final feature map (*i.e.*, the output feature map after CBAM) was obtained by multiplying this vector with the input feature map.

$$M_c(F) = \sigma(W_1(W_0(F^c_{avg})) + W_1(W_0(F^c_{max})) \tag{1}$$

$$M_s(F) = \sigma(f^{7\times7}([F^c_{avg}; F^c_{max}])) \tag{2}$$

Derivation of the network structure of the proposed model

During the construction of DBDCNN, the rationality of this network structure was verified according to three aspects as following: i) the rationality of CBAM location introduction; ii) the rationality of CBAM number introduction; iii) the rationality of dual branch path parallelization.

Verify the rationality of the convolutional block attention module insertion position

Four locations were selected as insertion points for CBAM in the network structure (Figure 1). After Conv1, Conv2, Conv3 and Conv4, these locations were denoted as L1, L2, L3 and L4, respectively. Meanwhile, classification recognition of disease images under the condition of inserting CBAM at different positions was also tested. According to Table 1, classification accuracy obtained by inserting CBAM after L2, *i.e.*, Conv2, was 0.9447, significantly much higher than those detected from the other three locations. Therefore, the attention module CBAM at L2 was inserted in order to get more feature information of key areas more often and faster to improve the classification and recognition of disease images.

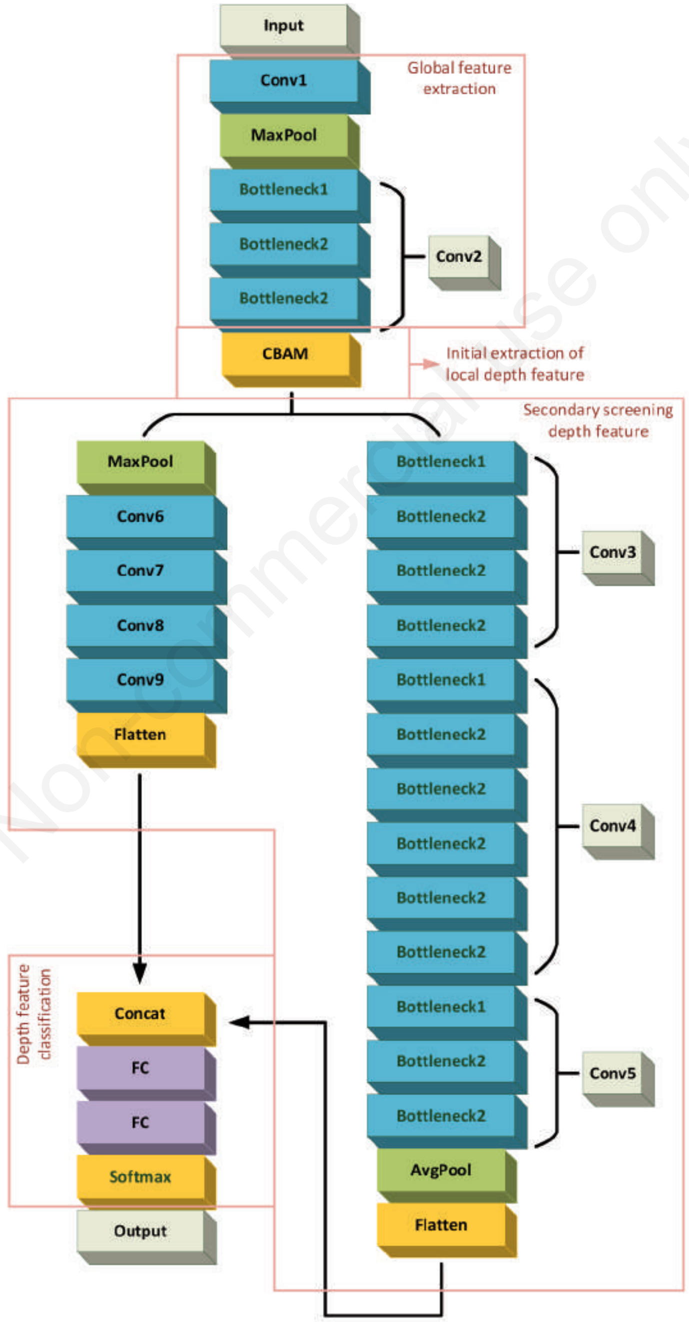

Figure 1. Network structure of the proposed double-branch deep convolutional neural network model.

Validate the reasonableness of the quantity introduced by convolutional block attention module

According to the previous description, the best insertion position of CBAM was located at L2. Therefore, we further introduced different numbers of CBAMs to analyze the variation of their classification accuracy. Here the number of introductions was set to 1, 2, and 3, where the case of number 2 was subsequently divided into two types of introductions, *i.e.*, series introduction and parallel introduction (Figure 2). According to Table 2, when the number of CBAM was 1, the classification accuracy was 0.9482, higher than other numbers. This was because by adding a CBAM, global feature information about the lesion region could be better extracted from the depth features. If too many attention modules were inserted, weakness of the input global feature information and the depth feature extraction of the focus region might be affected or reversed, leading to an eventual degradation of overall classification accuracy. Moreover, an increase in the number of introductions might cause computational overburden in computers, as well as memory wastage and prolonging the training time, hence was not a wise choice for the experiment. Accordingly, more attention modules inserted in the network might not only necessarily facilitate the improvement of the experimental results, but might also be counterproductive. Therefore, the most optimal choice should be made only according to the experimental equipment and network structure.

Verify the rationality of parallelization of double-branch paths

Based on the original ResNet-50 network structure, here we introduced another new path for depth feature extraction. The left branch path in the Secondary screening depth feature part was illustrated in Figure 1. To filter out weakly correlated feature information and obtain more significant local depth feature information, the branch first performs maximum pooling on the input feature map and then acts on four successive convolutional layers. The results showed that a dual-branch path was significantly and much better than a single path in deep feature extraction and classification. However, the introduction of new feature extraction path significantly increased the number of network parameters, making more time spent on training than the original structure. According to Table 3, although the newly proposed method increased the parameter layers and expanded parameter numbers, the classification accuracy of this model was constructively improved and thereby classification task could be efficiently completed.

Materials and Methods

In this study, we developed a dual-branch DCNN model and applied it to a rice leaf disease dataset for automatic identification and classification of leaf diseases. The overall workflow of our proposed model is illustrated in Figure 3. The process was roughly divided into four parts: i) data pre-processing and dataset division; ii) data augmentation; iii) construction and training of DBDCNN; iv) implementation of disease image classification.

Data pre-processing and dataset division

5932 rice leaf disease images were collected from the Kaggle dataset (https://www.kaggle.com/minhhuy2810/rice-diseases-image-dataset, https://www.kaggle.com/vbookshelf/rice-leaf-diseases) as images in this dataset, and these images with different labels were used to train the proposed model. In Figure 4, each type of rice leaf disease image sample is shown, and all disease images in the dataset are divided into four categories: i) Bacterial blight (including 1584 images); ii) Blast (including 1440 images);

Table 1. Classification accuracy obtained by inserting convolutional block attention module at different positions.

Insert position	L1	L2	L3	L4
Classification accuracy	0.9158	0.9447	0.9402	0.9273

Table 2. Classification accuracy corresponds to different numbers of convolutional block attention module at L2.

Number of CBAM introduction	One	Connecting two in series	Connecting two in parallel	Connecting three in series
Classification accuracy	0.9482	0.9376	0.9425	0.9284

CBAM, convolutional block attention module.

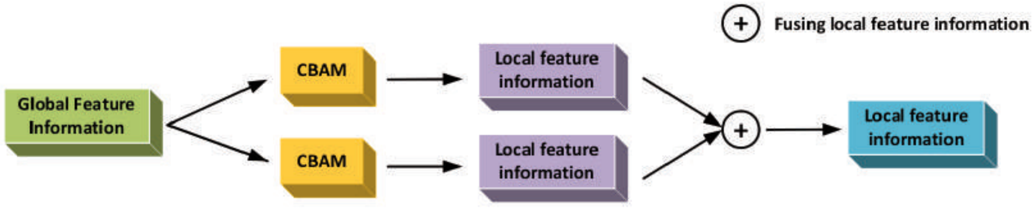
Fusing local feature information
Global Feature Information
CBAM
CBAM
Local feature information
Local feature information
Local feature information

Figure 2. Schematic diagram of convolutional block attention module's parallel access method in the network.

iii) Brown spot (including 1600 images); iv) Tungro (including 1308 images). Firstly, pre-processing operations such as check-weighing on an acquired original dataset were conducted to remove duplicate and unrecognizable images. Secondly, since images in the dataset were of different sizes, a resize operation was performed to make all images into an acceptable 224×224 uniform size for network input. Then the processed dataset was divided into two parts, namely training set and test set respectively, with a distributed ratio of 8:2.

Data augmentation

To make the network model and training process run smoothly and fast, images in both test and training sets should be normalized. Since the number of images was not too large in the training set to meet the requirements of deep neural networks, data augmentations (*e.g.*, vertical flip, horizontal flip, rotation 90°, rotation 270°, and Gaussian noise) were conducted on the training set, and the augmentation process was detailed in https://github.com/ aleju/imgaug.

The data distributions of the test set as well as the enhanced training set were shown in Table 4. Obviously, image data could significantly improve the accuracy of the training process after augmentation.

Construction and training of double-branch deep convolutional neural network

The network structure of our proposed DBDCNN model is indicated in Figure 1. In DBDCNN, the main network structure of ResNet-50 was used, which at the same time was optimized and improved. Specifically, the optimization and improvement of the original Resnet-50 network structure in this article reflected in the introduction of the attention module CBAM and another deep convolutional extraction path, thereby forming a DBDCNN model proposed in this article. Where Conv1, MaxPool and Avgpool were set to the same parameter values as in ResNet-50, while Conv6 to Conv9 were four consecutive convolutional layers with 3×3 convolutional kernels (s=1). As for Conv2 to Conv5, they contain a large number of Bottleneck1 and Bottleneck2 structures, which are called bottleneck structures. Through Figure 5, we can clearly know the internal structure of Bottleneck1 and Bottleneck2.

In Bottleneck2, the 1×1 convolutional layer is used to reduce the dimension of the input data (that is, reduce the number of input

Table 3. Performance analysis of single and double-feature extraction paths.

Type	Single path	Double branch path
Classification accuracy	0.9347	0.9495
Number of layers added	-	4
Number of participants added (in millions)	-	0.3882

Table 4. The detailed distribution of this dataset.

	Number of original images	Number of images after data deduplication	Number of images used for testing	Number of augmented training images
Bacterial blight	1584	1393	303	5450
Blast	1440	1107	264	4215
Brown spot	1600	1321	305	4215
Tungro	1308	1308	261	5235

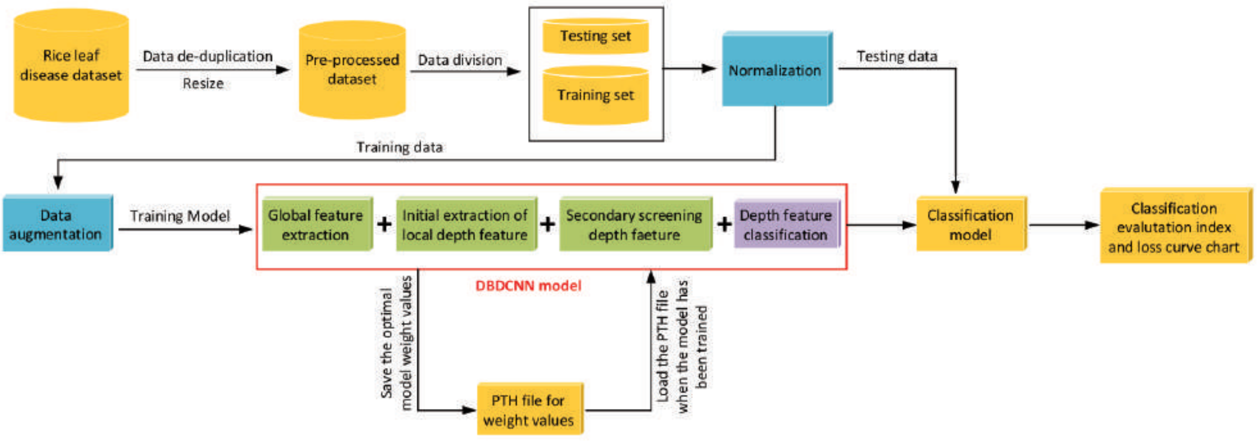
Rice leaf disease dataset
Data de-duplication
Resize
Pre-processed dataset
Data division
Testing set
Training set
Normalization
Testing data
Training data
Data augmentation
Training Model
Global feature extraction
+
Initial extraction of local depth feature
+
Secondary screening depth faeture
+
Depth feature classification
DBDCNN model
Classification model
Classification evalutation index and loss curve chart
Save the optimal model weight values
PTH file for weight values
Load the PTH file when the model has been trained

Figure 3. Flow chart of rice leaf disease classification model.

channels), the 3×3 convolutional layer is used to capture the spatial characteristics of the data (that is, perform feature extraction), and the final 1×1 convolutional layer is used to restore the dimensionality of the output data (*i.e.*, the number of recovered output channels). In Bottleneck1, only the last 1×1 convolutional layer on the left increases the number of channels of the input data, while the first two convolutional layers do not have any processing on the number of channels. Similarly, the 1×1 convolutional layer on the right also increases the number of channels of the input data by the same magnitude. It can be seen from this that when using the Bottleneck2 structure, the number of input and output channels of the data does not change, but when using the Bottleneck1 structure, the number of output channels is not equal to the number of input channels (specifically, the former is 4 times the latter). At the same time, in the above two Bottleneck structures, the activation function we use is the Sigmoid Linear Unit (SiLU) (Elfwing *et al.*,

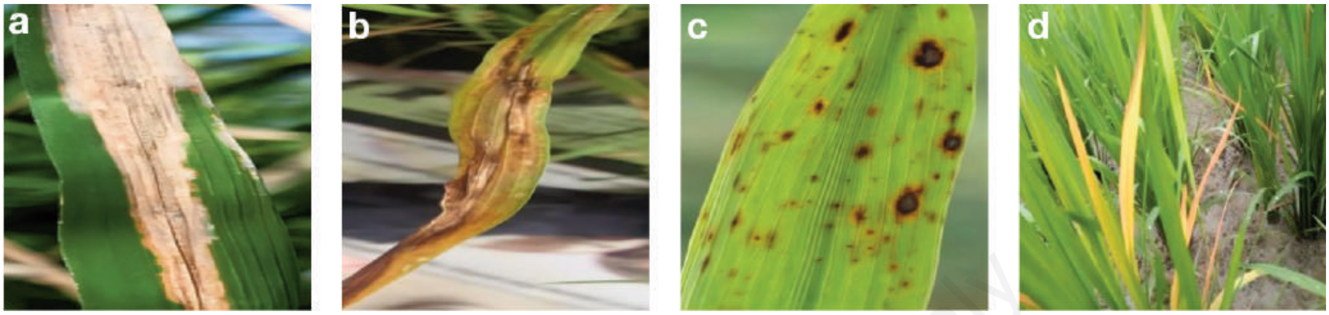

Figure 4. Four types of rice leaf disease image samples in this dataset (the images in this data set are all from the Kaggle dataset): (**a**) Bacterial blight, (**b**) Blast, (**c**) Brown spot, (**d**) Tungro.

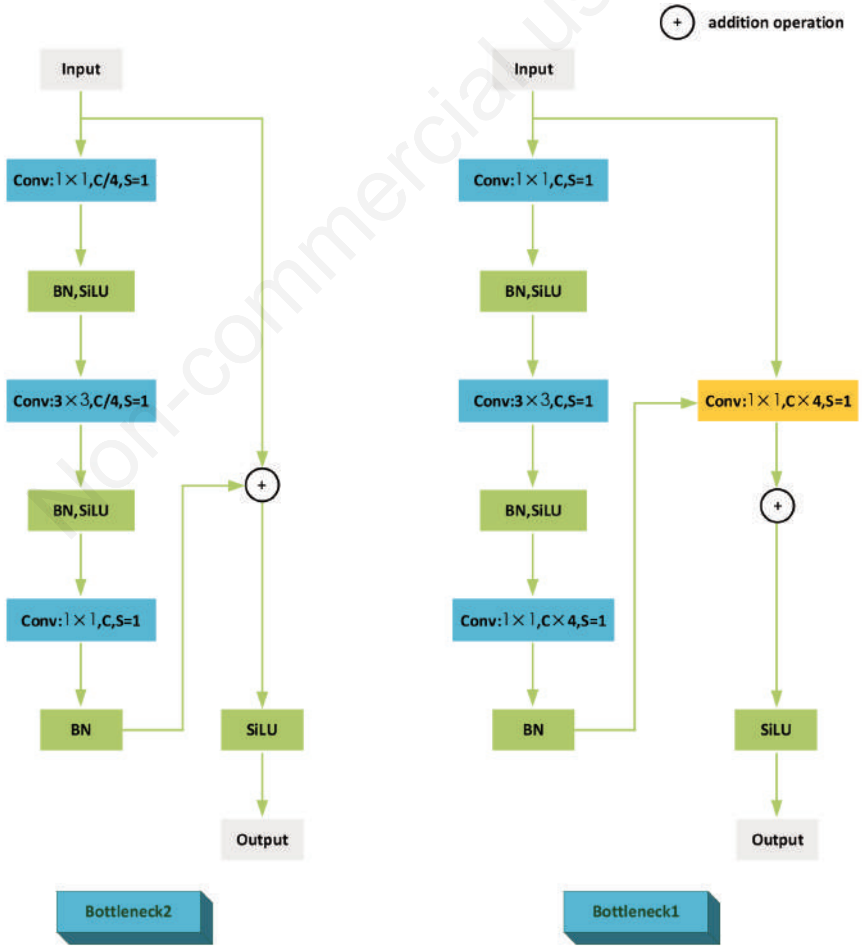

Figure 5. Schematic diagram of the structures of Bottleneck1 and Bottleneck2. Among them, S represents stride, C represents the number of channels, SiLU represents the activation function, 1×1 and 3×3 both represent the size of the convolution kernel, and BN represents the Batch Normalization layer. SiLU, Sigmoid Linear Unit.

2018), and there is a Batch normalization (BN) (Ioffe and Szegedy, 2015) operation before it. The detailed calculation process can be clearly known from Figure 5. In addition, for the use of activation functions, it was still Relu in Conv1 and Relu and sigmoid in CBAM. In this paper, using the BN operation helps normalize the data, avoiding unstable network performance caused by too large data before performing ReLU or SiLU. In addition, the relevant parameters of each layer of the model are shown in Figure 6. Among them, F-E, F-C, F-F, and C-P represent feature extraction, feature compression, feature fusion, and classification prediction, respectively. And Fil, K-S, S, P, Act, P-S, B-S, I-F, O-F respectively represent filter, kernel size, stride, padding, activation, pool size, batch size, in-features and out-features. And "Conv2-B1-L1" means the first convolutional layer on the left side of the Bottleneck1 structure in Conv2, "Conv2-B1-R1" means the first convolutional layer on the right side of the Bottleneck1 structure in Conv2, "Conv2- B21-L1" indicates the first convolutional layer on the left in the first Bottleneck2 structure in Conv2 and "Conv2-B22-L1" indicates the first convolutional layer on the left in the second Bottleneck2 structure in Conv2. Other descriptions have similar meanings. As can be seen from the table, except for various pooling layers, our parameter layer has a total of 62 layers, namely 58 convolutional layers and 4 fully connected layers. Among them, the attention mechanism CBAM contributes a convolutional layer and 2 fully connected layers.

Once the network layer is built, we can train this model using disease images in the training set. Parameter values of the weights corresponding to each epoch online were saved, and the optimal one in the PTH file was stored.

Implementation of disease image classification

At the end of training, all optimal weight parameter values in PTH files were loaded and the classification of the disease images in the test set was started. The process of disease image classification and recognition using the proposed model could be divided into four stages, as indicated in the details below.

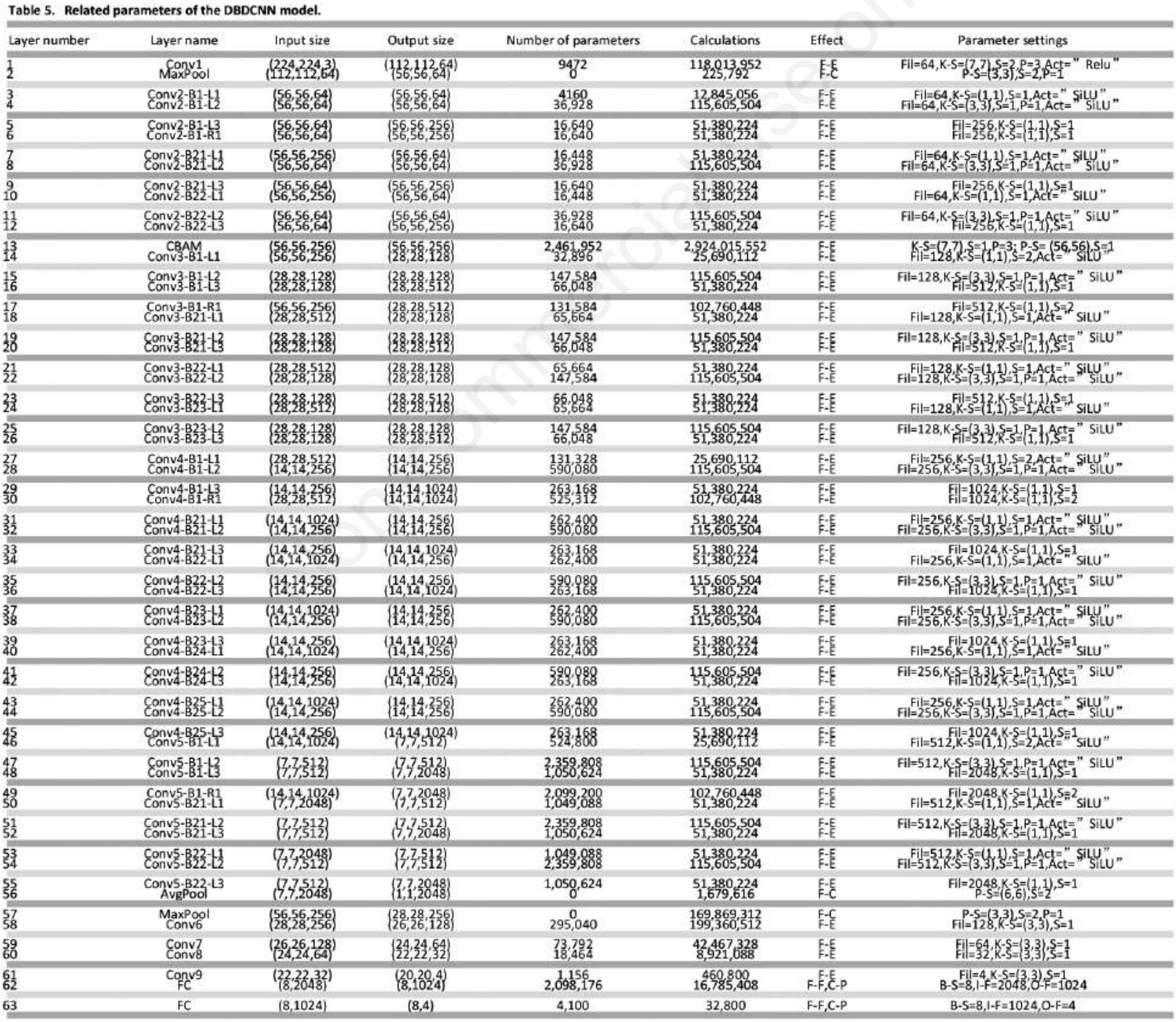

Table 5. Related parameters of the DBDCNN model.

Layer number	Layer name	Input size	Output size	Number of parameters	Calculations	Effect	Parameter settings
1	Conv1	(224,224,3)	(112,112,64)	9472	118,013,952	F-E	Fil=64,K-S=(7,7),S=2,P=3,Act=" Relu"
2	MaxPool	(112,112,64)	(56,56,64)	0	225,792	F-C	P-S=(3,3),S=2,P=1
3	Conv2-B1-L1	(56,56,64)	(56,56,64)	4160	12,845,056	F-E	Fil=64,K-S=(1,1),S=1,Act=" SiLU"
4	Conv2-B1-L2	(56,56,64)	(56,56,64)	36,928	115,605,504	F-E	Fil=64,K-S=(3,3),S=1,P=1,Act=" SiLU"
5	Conv2-B1-L3	(56,56,64)	(56,56,256)	16,640	51,380,224	F-E	Fil=256,K-S=(1,1),S=1
6	Conv2-B1-R1	(56,56,64)	(56,56,256)	16,640	51,380,224	F-E	Fil=256,K-S=(1,1),S=1
7	Conv2-B21-L1	(56,56,256)	(56,56,64)	16,448	51,380,224	F-E	Fil=64,K-S=(1,1),S=1,Act=" SiLU"
8	Conv2-B21-L2	(56,56,64)	(56,56,64)	36,928	115,605,504	F-E	Fil=64,K-S=(3,3),S=1,P=1,Act=" SiLU"
9	Conv2-B21-L3	(56,56,64)	(56,56,256)	16,640	51,380,224	F-E	Fil=256,K-S=(1,1),S=1
10	Conv2-B22-L1	(56,56,256)	(56,56,64)	16,448	51,380,224	F-E	Fil=64,K-S=(1,1),S=1,Act=" SiLU"
11	Conv2-B22-L2	(56,56,64)	(56,56,64)	36,928	115,605,504	F-E	Fil=64,K-S=(3,3),S=1,P=1,Act=" SiLU"
12	Conv2-B22-L3	(56,56,64)	(56,56,256)	16,640	51,380,224	F-E	Fil=256,K-S=(1,1),S=1
13	CBAM	(56,56,256)	(56,56,256)	2,461,952	2,924,015,552	F-E	K-S=(7,7),S=1,P=3; P-S= (56,56),S=1
14	Conv3-B1-L1	(56,56,256)	(28,28,128)	32,896	25,690,112	F-E	Fil=128,K-S=(1,1),S=2,Act=" SiLU"
15	Conv3-B1-L2	(28,28,128)	(28,28,128)	147,584	115,605,504	F-E	Fil=128,K-S=(3,3),S=1,P=1,Act=" SiLU"
16	Conv3-B1-L3	(28,28,128)	(28,28,512)	66,048	51,380,224	F-E	Fil=512,K-S=(1,1),S=1
17	Conv3-B1-R1	(56,56,256)	(28,28,512)	131,584	102,760,448	F-E	Fil=512,K-S=(1,1),S=2
18	Conv3-B21-L1	(28,28,512)	(28,28,128)	65,664	51,380,224	F-E	Fil=128,K-S=(1,1),S=1,Act=" SiLU"
19	Conv3-B21-L2	(28,28,128)	(28,28,128)	147,584	115,605,504	F-E	Fil=128,K-S=(3,3),S=1,P=1,Act=" SiLU"
20	Conv3-B21-L3	(28,28,128)	(28,28,512)	66,048	51,380,224	F-E	Fil=512,K-S=(1,1),S=1
21	Conv3-B22-L1	(28,28,512)	(28,28,128)	65,664	51,380,224	F-E	Fil=128,K-S=(1,1),S=1,Act=" SiLU"
22	Conv3-B22-L2	(28,28,128)	(28,28,128)	147,584	115,605,504	F-E	Fil=128,K-S=(3,3),S=1,P=1,Act=" SiLU"
23	Conv3-B22-L3	(28,28,128)	(28,28,512)	66,048	51,380,224	F-E	Fil=512,K-S=(1,1),S=1
24	Conv3-B23-L1	(28,28,512)	(28,28,128)	65,664	51,380,224	F-E	Fil=128,K-S=(1,1),S=1,Act=" SiLU"
25	Conv3-B23-L2	(28,28,128)	(28,28,128)	147,584	115,605,504	F-E	Fil=128,K-S=(3,3),S=1,P=1,Act=" SiLU"
26	Conv3-B23-L3	(28,28,128)	(28,28,512)	66,048	51,380,224	F-E	Fil=512,K-S=(1,1),S=1
27	Conv4-B1-L1	(28,28,512)	(14,14,256)	131,328	25,690,112	F-E	Fil=256,K-S=(1,1),S=2,Act=" SiLU"
28	Conv4-B1-L2	(14,14,256)	(14,14,256)	590,080	115,605,504	F-E	Fil=256,K-S=(3,3),S=1,P=1,Act=" SiLU"
29	Conv4-B1-L3	(14,14,256)	(14,14,1024)	263,168	51,380,224	F-E	Fil=1024,K-S=(1,1),S=1
30	Conv4-B1-R1	(28,28,512)	(14,14,1024)	525,312	102,760,448	F-E	Fil=1024,K-S=(1,1),S=2
31	Conv4-B21-L1	(14,14,1024)	(14,14,256)	262,400	51,380,224	F-E	Fil=256,K-S=(1,1),S=1,Act=" SiLU"
32	Conv4-B21-L2	(14,14,256)	(14,14,256)	590,080	115,605,504	F-E	Fil=256,K-S=(3,3),S=1,P=1,Act=" SiLU"
33	Conv4-B21-L3	(14,14,256)	(14,14,1024)	263,168	51,380,224	F-E	Fil=1024,K-S=(1,1),S=1
34	Conv4-B22-L1	(14,14,1024)	(14,14,256)	262,400	51,380,224	F-E	Fil=256,K-S=(1,1),S=1,Act=" SiLU"
35	Conv4-B22-L2	(14,14,256)	(14,14,256)	590,080	115,605,504	F-E	Fil=256,K-S=(3,3),S=1,P=1,Act=" SiLU"
36	Conv4-B22-L3	(14,14,256)	(14,14,1024)	263,168	51,380,224	F-E	Fil=1024,K-S=(1,1),S=1
37	Conv4-B23-L1	(14,14,1024)	(14,14,256)	262,400	51,380,224	F-E	Fil=256,K-S=(1,1),S=1,Act=" SiLU"
38	Conv4-B23-L2	(14,14,256)	(14,14,256)	590,080	115,605,504	F-E	Fil=256,K-S=(3,3),S=1,P=1,Act=" SiLU"
39	Conv4-B23-L3	(14,14,256)	(14,14,1024)	263,168	51,380,224	F-E	Fil=1024,K-S=(1,1),S=1
40	Conv4-B24-L1	(14,14,1024)	(14,14,256)	262,400	51,380,224	F-E	Fil=256,K-S=(1,1),S=1,Act=" SiLU"
41	Conv4-B24-L2	(14,14,256)	(14,14,256)	590,080	115,605,504	F-E	Fil=256,K-S=(3,3),S=1,P=1,Act=" SiLU"
42	Conv4-B24-L3	(14,14,256)	(14,14,1024)	263,168	51,380,224	F-E	Fil=1024,K-S=(1,1),S=1
43	Conv4-B25-L1	(14,14,1024)	(14,14,256)	262,400	51,380,224	F-E	Fil=256,K-S=(1,1),S=1,Act=" SiLU"
44	Conv4-B25-L2	(14,14,256)	(14,14,256)	590,080	115,605,504	F-E	Fil=256,K-S=(3,3),S=1,P=1,Act=" SiLU"
45	Conv4-B25-L3	(14,14,256)	(14,14,1024)	263,168	51,380,224	F-E	Fil=1024,K-S=(1,1),S=1
46	Conv5-B1-L1	(14,14,1024)	(7,7,512)	524,800	25,690,112	F-E	Fil=512,K-S=(1,1),S=2,Act=" SiLU"
47	Conv5-B1-L2	(7,7,512)	(7,7,512)	2,359,808	115,605,504	F-E	Fil=512,K-S=(3,3),S=1,P=1,Act=" SiLU"
48	Conv5-B1-L3	(7,7,512)	(7,7,2048)	1,050,624	51,380,224	F-E	Fil=2048,K-S=(1,1),S=1
49	Conv5-B1-R1	(14,14,1024)	(7,7,2048)	2,099,200	102,760,448	F-E	Fil=2048,K-S=(1,1),S=2
50	Conv5-B21-L1	(7,7,2048)	(7,7,512)	1,049,088	51,380,224	F-E	Fil=512,K-S=(1,1),S=1,Act=" SiLU"
51	Conv5-B21-L2	(7,7,512)	(7,7,512)	2,359,808	115,605,504	F-E	Fil=512,K-S=(3,3),S=1,P=1,Act=" SiLU"
52	Conv5-B21-L3	(7,7,512)	(7,7,2048)	1,050,624	51,380,224	F-E	Fil=2048,K-S=(1,1),S=1
53	Conv5-B22-L1	(7,7,2048)	(7,7,512)	1,049,088	51,380,224	F-E	Fil=512,K-S=(1,1),S=1,Act=" SiLU"
54	Conv5-B22-L2	(7,7,512)	(7,7,512)	2,359,808	115,605,504	F-E	Fil=512,K-S=(3,3),S=1,P=1,Act=" SiLU"
55	Conv5-B22-L3	(7,7,512)	(7,7,2048)	1,050,624	51,380,224	F-E	Fil=2048,K-S=(1,1),S=1
56	AvgPool	(7,7,2048)	(1,1,2048)	0	1,679,616	F-C	P-S=(6,6),S=2
57	MaxPool	(56,56,256)	(28,28,256)	0	169,869,312	F-C	P-S=(3,3),S=2,P=1
58	Conv6	(28,28,256)	(26,26,128)	295,040	199,360,512	F-E	Fil=128,K-S=(3,3),S=1
59	Conv7	(26,26,128)	(24,24,64)	73,792	42,467,328	F-E	Fil=64,K-S=(3,3),S=1
60	Conv8	(24,24,64)	(22,22,32)	18,464	8,921,088	F-E	Fil=32,K-S=(3,3),S=1
61	Conv9	(22,22,32)	(20,20,4)	1,156	460,800	F-E	Fil=4,K-S=(3,3),S=1
62	FC	(8,2048)	(8,1024)	2,098,176	16,785,408	F-F,C-P	B-S=8,I-F=2048,O-F=1024
63	FC	(8,1024)	(8,4)	4,100	32,800	F-F,C-P	B-S=8,I-F=1024,O-F=4

Figure 6. Released parameters of double-branch deep convolutional neural network model.

Stage 1: global feature extraction. First, original feature map of 224×224×3 was input into the network, and the 56×56×256 feature map was output under the action of Conv1, MaxPool and Conv2. The feature information acquired at this time was the global feature information on disease image, rather than effective area. To locate the lesion area more precisely and to obtain more information on local focus features, attention mechanism and deep feature screening should be implemented subsequently.

Stage 2: initial extraction of local depth feature. The global feature information obtained above was input into CBAM. After going through CAM and SAM, feature information on key areas of the disease image was subsequently obtained, while redundant or irrelevant feature information was removed. Meanwhile, although the initial extraction of image depth features was completed, it was necessary for another feature filtering as feature information presented with little relevance to the specific lesion area.

Stage 3: secondary screening depth feature. The output feature map obtained from CBAM was used for depth feature re-extraction to filter out less relevant depth features and subsequently removed, which was implemented as a deep convolution operation based on a double branch path. As indicated in Figure 1, a convolutional block in ResNet-50 for deep feature extraction was used in the right path, followed by an average pooling and Flatten operation to obtain a feature vector. On the other hand, MaxPool was performed on the input feature map in the left path, followed by convolutional layers used for overlap and Flatten expansion, which also ends up with a feature vector.

Stage 4: depth feature classification. The two obtained feature vectors were fused in a Concat (Dettmers *et al*., 2018) manner. Integrated feature information was subsequently fed into two consecutive fully connected layers with Softmax for classification and recognition.

Results

Experimental platform and evaluation indicators

Our proposed classification model was implemented based on the deep learning framework pytorch. And because the configuration of the laptop is not enough to support the efficient and stable operation of the experiment, we debug and run the code on Google Colaboratory to ensure that the experiment is completed quickly and stably. Finally, to evaluate the performance of different models, we selected 4 image classification evaluation metrics (Zeng *et al*., 2022), including Accuracy, Precision, Recall and F1-Score. And formulas (3) to (6) give the calculation methods of these indicators, where *TP*, *TN*, *FP* and *FN* represent true positive, true negative, false positive and false negative, respectively.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (3)$$

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

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

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

Details of the training process

Before training the model in this paper, we first set the training parameters. The specific training parameter values are shown in Table 5. In this table, except that Optimizer and Min-Max learning rate have multiple values, the rest of the parameter values are determined. During the training process, we randomly combined different hyperparameter values, and a total of 6 parameter value combinations can be obtained. Based on these 6 combinations, we conducted 6 independent trainings on the model in this paper and the comparison model, and recorded the train loss and val loss obtained by each model in each training in the images shown in the *Appendix*. At the same time, the time spent by different models in each training process is recorded in Figure 7. The purpose of this is to use each training time as an indicator for evaluating model performance, which is convenient for comparing performance differences between different models.

Experimental results

After the model in this paper completes 6 independent trainings, we compared and analyzed these results, and finally selected the optimal combination of stochastic gradient descent and 0.00001-0.001, and recorded the corresponding optimal train loss and val loss in Figure 8. Next, load the weight parameter file corre-

Table 5. The training parameters of the proposed model.

Hyperparameter	Value
Dataset	Rice leaf disease dataset (from Kaggle dataset)
Image size	224×224
Min-max learning rate	0.0001-0.01, 0.00001-0.001, 0.000001-0.0001
Optimizer	SGD, Adam
Batch size	8
Epochs	200
Momentum	0.9
Weight decay	0.0005
Num-workers	4
Learning rate decay type	cos

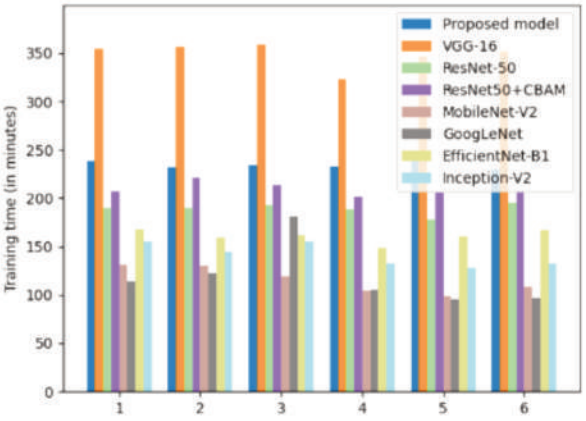

Figure 7. Comparison of the training time of different models in the 6 independent training processes, and the statistics are made in minutes.

sponding to the above combination into the network model to identify and classify the disease images on the test set. When the disease images were input into this network for testing, output feature maps were obtained and shown in Figure 9. Due to the large number of output channels in the network, the number of output feature maps is too large, which will make the size of each output feature map distributed on an image smaller, and it is difficult to carefully observe the feature information expressed by the feature map. In order to illustrate the feature information extracted by the network model from the input image, we only show the output feature maps of the Conv1 layer. The results indicated that as the depth deepens, extracted feature information becomes more concentrated and the depth feature information on key regions was much richer. At the end of the test, the Confusion Matrix obtained was presented in Figure 10, suggesting that the two disease images labeled Bacterial blight were incorrectly identified as Brown spot disease type. Similarly, for the disease images of type Blast, a total of 8 were misclassified as Bacterial blight and 1 was misclassified as Brown spot. For Brown spot and Tungro, there were respectively 2 and 1 disease images incorrectly identified as Blast and Bacterial blight. However, in terms of the effect presented by confusion matrix, the misclassified disease images account for only a small fraction of the number of test samples, and the overall classification effect was still quite satisfactory. In addition, the above evaluation metrics calculated based on this matrix were indicated in Figure 11, demonstrating that Tungro obtained 1 for all three metrics. The lowest Precision and Recall values were detected in Bacterial blight and Blast, respectively, and both were 0.97. The remaining disease types were evaluated with index values ranging from 0.98 to 1. In addition, the proposed model also achieved ideal recognition results, with an accuracy rate as high as 98.73% (Table 6).

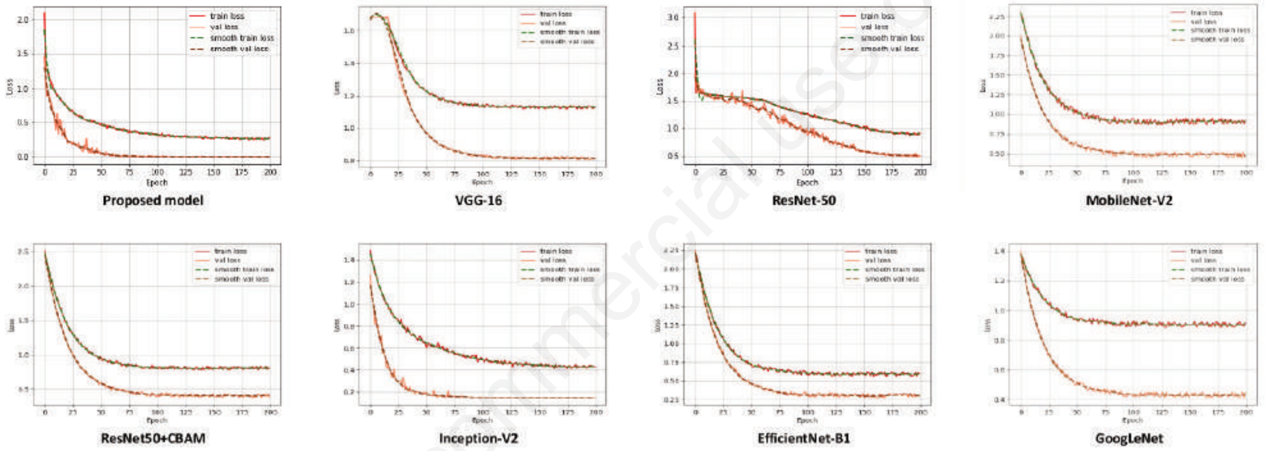

Figure 8. The optimal train loss and val loss curves of different models during the training process, where the values of the training parameters are from Table 5.

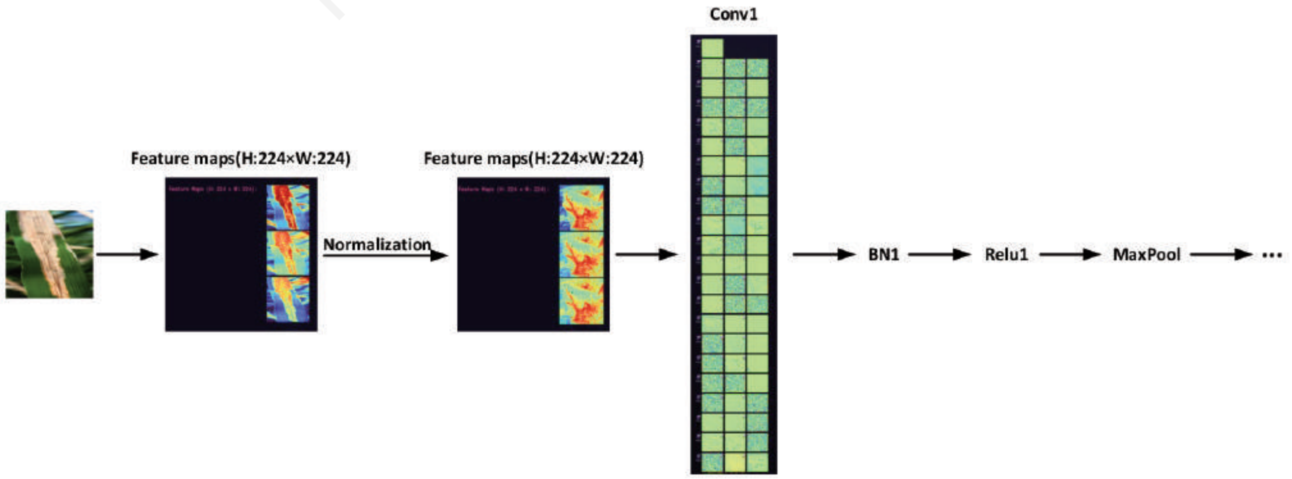

Figure 9. Example of output feature maps of Conv1.

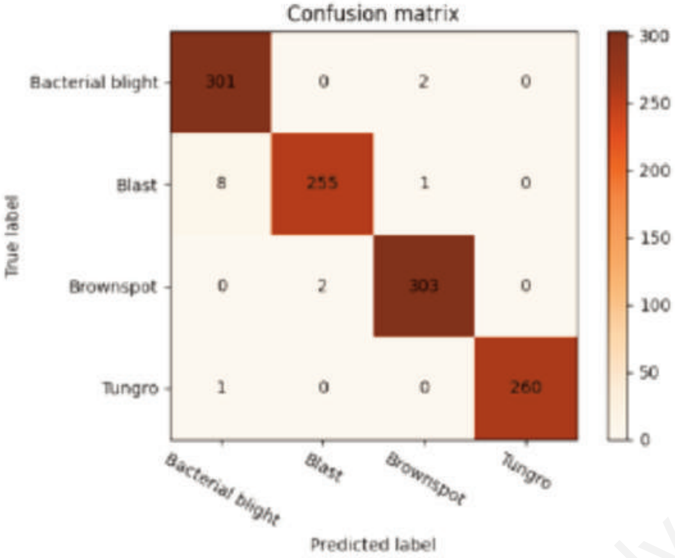

Figure 10. The confusion matrix of the model of this article.

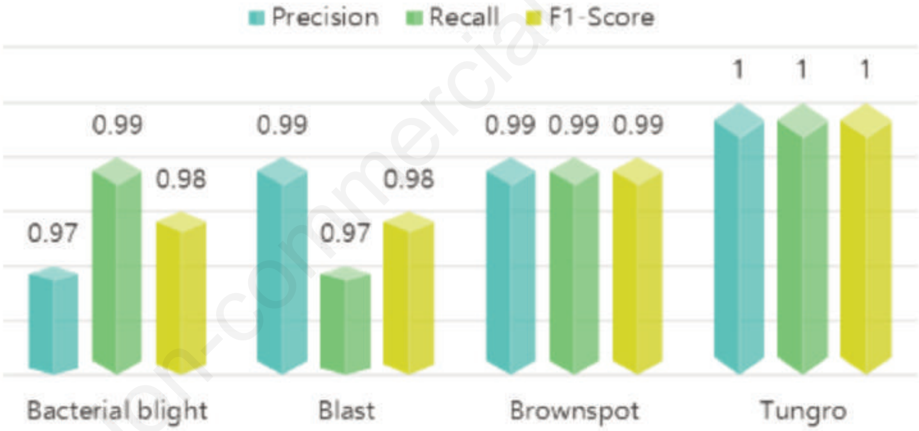

Figure 11. Different evaluation metrics for the proposed model in this study.

Table 6. Performance comparison of different models.

Model	Layers	Model description	Precision	Recall	F1-score	Accuracy (%)
Proposed model	62	58conv+ 4fc layers	0.99	0.99	0.99	98.73
VGG-16	16	13conv+ 3fc layers	0.85	0.83	0.84	82.97
ResNet-50	49	49conv layers	0.91	0.91	0.91	91.09
MobileNet-V2	17	17conv layers	0.95	0.94	0.94	94.35
ResNet50+CBAM	56	51conv+ 5fc layers	0.96	0.95	0.95	95.23
Inception-V2	23	22conv+ 1fc layers	0.97	0.95	0.96	97.04
EfficientNet-B1	25	24conv+ 1fc layers	0.95	0.96	0.95	96.27
GoogLeNet	23	22conv+ 1fc layers	0.95	0.94	0.94	94.83

Comparative experiments

Comparative experiments were performed using seven classification models, VGG-16, ResNet-50, ResNet50+CBAM, MobileNet-V2, GoogLeNet, EfficientNet-B1 and Inception-V2. ResNet50+CBAM is to add CBAM directly to the backbone structure of ResNet-50, and the rest of the network structure remains unchanged. In the comparison experiment, in order to ensure the consistency of the experimental conditions, not only the experiment was carried out on the same device, but also the values of the training hyperparameters were selected in Table 5. At the same time, these comparison models also have the same approach as the model in this paper. That is, after completing 6 independent trainings, the obtained results are compared and analyzed, and then a set of optimal weight parameter files are saved and reloaded into the network to realize the identification and classification of the disease images in the test set. In addition, the optimal train loss and val loss of each comparison model are also recorded in Figure 8. In this way, we tested the accuracy of different models and other evaluation index values (here each index is the average value), see Table 6 for details.

Discussion

Based on the deep residual network (He *et al*., 2016), this study developed and proposed a new DBDCNN network system for the recognition and classification of rice leaf disease images. This paper conducts comparative experiments and analyzes on VGG-16, ResNet-50, ResNet50+CBAM, MobileNet-V2, GoogLeNet, EfficientNet-B1, Inception-V2 and DBDCNN models. The number of layers (especially the parameter layer), model description, average values of the three evaluation metrics, and accuracy of these models are shown in Table 6. Among them, the accuracy rate of this model in this image classification task is 98.73%, which is significantly higher than that of VGG-16 (82.97%), ResNet-50 (91.09%), MobileNet-V2 (94.35%), ResNet50+CBAM (95.23%), GoogLeNet (94.83%), EfficientNet-B1 (96.27%) and Inception-V2 (97.04%) accuracy rates. Since the VGG-16 model has fewer parameter layers and insufficient depth, the extracted image feature information is not significant and rich, making the accuracy and other index values of the classification stage far inferior to other comparative models. Due to the introduction of a linear bottleneck, inverted residuals, and the use of average pooling instead of fully connected layers in MobileNet-V2, the obtained accuracy is significantly higher, although overall much lower than the classification results observed in this study. For ResNet-50, ResNet50+CBAM, and our proposed model, the latter two introduce CBAM compared to ResNet-50, thereby leading to a gap in the feature information of key regions in extracted images, and consequently an inferior accuracy compared to the latter two. Compound scaling allows the EfficientNet-B1 model to better balance width, depth, and resolution under limited computing resources. At the same time, the Mobile Inverted BottleNeck Conv structure also improves the computational efficiency of the model. Combining these two factors, we naturally know that the EfficientNet-B1 model can achieve good accuracy in this classification task, which is better than most of the comparison models. Since the GoogLeNet and Inception-V2 models both use the network architecture based on the Inception module, this architecture has better feature extraction and expression capabilities, and can more accurately identify and classify disease images, resulting in higher classification accuracy. However, compared with the former, the latter has improved in terms of model structure and training skills. Specifically, the latter has a deeper network structure than the former, uses more Inception modules, and can capture more complex features. Second, it introduces some new features and techniques, such as batch normalization, factorized convolution, *etc*., which help to improve the efficiency and accuracy of the model. Finally, Inception-V2 also uses new training techniques, such as Dropout, data enhancement, *etc*., which can alleviate the problem of overfitting and improve the generalization ability and robustness of the model. So, it achieves higher accuracy than the former on the test set. The ideal recognition effect of the model in this paper mainly depends on the dual-path deep feature extraction of the key lesion area, and then the fusion of the respective obtained features, and finally the disease classification. The experimental results show that adding the attention module (CBAM) to the network structure can greatly improve the performance of the model, strengthen the ability of the model to extract deep features and obtain more key information in the image.

Conclusions

In conclusion, here we proposed a novel model (*i.e.*, DBDCNN) for four rice leaf diseases (*i.e.*, Bacterial blight, Blast, Brown spot and Tungro) classification. The main body network structure of ResNet-50 was utilized for introduction of module CBAM and the operation of secondary screening depth features on top of it. The model was divided into four stages when performing classification recognition, namely Global feature extraction, Initial extraction of local depth feature, Secondary screening depth feature, and Depth feature classification respectively. Compared with the classical models (such as VGG-16, ResNet-50, ResNet50+CBAM, MobileNet-V2, GoogLeNet, EfficientNet-B1 and Inception-V2), our newly proposed DBDCNN method is able to extract much more accurate depth feature information of key regions within the image after the first three stages mentioned above, making it possible to achieve 98.73% accuracy during the final classification stage, and was proved better compared with other comparative models. Besides, the model's other classification assessment metrics are significantly higher than comparative models.

The advantages of this study are mainly reflected in the fact that we used a large-scale rice leaf disease data set, which includes a variety of representative disease types, so that the proposed model has high accuracy and generalization performance. In addition, by introducing an efficient feature extraction module (CBAM), we can automatically extract key features from rice leaf disease images, providing more powerful support for the subsequent feature classification stage. However, limitations and deficiencies also exist at any time, the specific situation is as follows: i) there is a certain imbalance between the number of disease images of different categories in the data set, which has a certain impact on the training and testing of the model; ii) the model and identification method in this paper also have poor scalability, and it is difficult to directly apply to the disease classification tasks of other crops; iii) the proposed model is complex and requires high computing resources and training time, which will limit its feasibility in practical applications.

With the continuous development of the field of deep learning, we can conduct more in-depth research from many aspects in the future to further improve this research. For example, in terms of data augmentation, we can use more complex enhancement techniques (such as stretching and zooming, *etc*.). In addition, you can also try transfer learning technology to transfer existing models and knowledge to rice leaf disease classification tasks to improve the generalization performance and scalability of the model. It can also integrate various modal information such as image, voice, video, *etc*., so as to improve the accuracy and robustness of rice leaf disease classification. These future research trends are worth exploring and deepening.

References

Barbedo A., Garcia J. 2013. Digital image processing techniques for detecting, quantifying and classifying plant diseases. SpringerPlus. 2:1-12.

Bharali P., Bhuyan C., Boruah A. 2019. Plant disease detection by leaf image classification using convolutional neural network. Comm. Com. Inf. Sc. 1025:194-205.

Chakraborty S., Newton A.C. 2011. Climate change, plant diseases and food security: an overview. Plant. Pathol. 60:2-14.

Chen J.D., Chen J.X., Zhang D.F., Sun Y.D., Nanehkaran Y.A. 2020. Using deep transfer learning for image-based plant disease identification. Comput. Electron. Agr. 173:105393.

Chen L., Zhang H.W., Xiao J., Nie L.Q., Shao J., Liu W., Chua T.S. 2017. SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning. Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017. pp. 6298-306.

Deb M., Dhal K.G., Mondal R., Gálvez J. 2021. Paddy Disease Classification Study: A Deep Convolutional Neural Network Approach. Opt. Memory Neural. 30:338-57.

Deb M., Garai A., Das A., Dhal K.G. 2022. LS-Net: a convolutional neural network for leaf segmentation of rosette plants. Neural. Comput. Appl. 34:18511-24.

Dettmers T., Minervini P., Stenetorp P., Riedel S. 2018. Convolutional 2D knowledge graph embeddings. 32nd AAAI Conf. Artif. Intell. AAAI. pp. 1811-8.

Elfwing S., Uchibe E., Doya K. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural Networks. 107:3-11.

Ghosal S., Sarkar K. 2020. Rice Leaf Diseases Classification Using CNN with Transfer Learning. 2020 IEEE Calcutta Conf. CALCON 2020 - Proc. 230-6.

He K.M., Zhang X.Y., Ren S.Q., Sun J. 2016. Deep residual learning for image recognition. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. pp. 770-8.

Huang J., Wang X., Rozelle S. 2016. Technological innovations, downside risk, and the modernization of agriculture. J. Dev. Econ. 118:207-21.

Hu J., Shen L., Albanie S., Sun G., Wu E.H. 2020. Squeeze-and-Excitation Networks. IEEE Trans. Pattern Anal. Mach. Intell. 42:2011-23.

Ioffe S., Szegedy C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 32nd Int. Conf. Mach. Learn. ICML. 1:448-56.

Jiang F., Lu Y., Chen Y., Cai D., Li G.F. 2020. Image recognition of four rice leaf diseases based on deep learning and support vector machine. Comput. Electron. Agr. 179:105824.

Kaur P., Harnal S., Gautam V., Singh M.P., Singh S.P. 2022. A novel transfer deep learning method for detection and classification of plant leaf disease. J. Amb. Intel. Hum. Comp. 14:12407-24.

Khan M.A., Kim Y.H., Choo J. 2018. Intelligent Fault Detection via Dilated Convolutional Neural Networks. Proc. - 2018 IEEE Int. Conf. Big Data Smart Comput. Big Comp. pp. 729-31.

Liu B., Zhang Y., He D.J., Li Y.X. 2018. Identification of apple leaf diseases based on deep convolutional neural networks. Symmetry. 10.

Nandhini N., Bhavani R. 2020. Feature extraction for diseased leaf image classification using machine learning. 2020 Int. Conf. Comput. Commun. Informatics, ICCCI. pp. 22-5.

Omer S.M., Ghafoor K.Z., Askar S.K. 2022. An Intelligent System for Cucumber Leaf Disease Diagnosis Based on the Tuned Convolutional Neural Network Algorithm. Mob. Inf. Syst. 2022:1-16.

Panchal P., Raman V.C., Mantri S. 2019. Plant Diseases Detection and Classification using Machine Learning Models. CSITSS 2019 - 2019 4th Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solut. Proc. 4:1-6.

Pandian A.J., Kanchanadevi K., Kumar D.V., Jasinska E., Gono R., Leonowicz Z., Jasinski M. 2022. A Five Convolutional Layer Deep Convolutional Neural Network for Plant Leaf Disease Detection. Electronics-Switz. 11.

Peng S.B., Tang Q.Y., Zou Y.B. 2009. Current status and challenges of rice production in China. Plant. Prod. Sci. 12:3-8.

Rahman C.R., Arko P.S., Ali M.E., Iqbal Khan M.A., Apon S.H., Nowrin F., Wasif A. 2020. Identification and recognition of rice diseases and pests using convolutional neural networks. Biosyst. Eng. 194:112-20.

Ray D.K., Ramankutty N., Mueller N.D., West P.C., Foley J.A. 2012. Recent patterns of crop yield growth and stagnation. Nat. Commun. 3:1293.

Savary S., Willocquet L., Pethybridge S.J., Esker P., McRoberts N., Nelson A. 2019. The global burden of pathogens and pests on major food crops. Nat. Ecol. Evol. 3:430-9.

Sethy P.K., Barpanda N.K., Rath A.K., Behera S.K. 2020. Deep feature-based rice leaf disease identification using support vector machine. Comput. Electron. Agr. 175:105527.

Shi Y., Wang X.F., Zhang S.W., Zhang C.L. 2015. PNN based crop disease recognition with leaf image features and meteorological data. Int. J. Agric. Biol. Eng. 8:60-8.

Skamnioti P., Gurr S.J. 2009. Against the grain: safeguarding rice from rice blast disease. Trends Biotechnol. 27:141-50.

Subetha T., Khilar R., Subaja Christo M. 2021. WITHDRAWN: A comparative analysis on plant pathology classification using deep learning architecture – Resnet and VGG19. Mater. Today Proc.

Sundaram R.M., Vishnupriya M.R., Biradar S.K., Thakur R.P., Rao G.J. 2014. Molecular mapping of quantitative trait loci for blast resistance in rice. Rice. 7:1-12.

Valent B., Khang C.H. 2010. Recent advances in rice blast effector research. Curr. Opin. Plant. Biol. 13:434-41.

Waheed A., Goyal M., Gupta D., Khanna A., Hassanien A.E., Pandey H.M. 2020. An optimized dense convolutional neural network model for disease recognition and classification in corn leaf. Comput. Electron. Agr. 175:105456.

Woo S., Park J., Lee J.Y., Kweon I.S. 2018. CBAM: Convolutional block attention module. ECCV. pp. 3-19.

Yu Y.Y., Liu M.Z., Feng H.J., Xu Z.H., Li Q. 2020. Split-Attention Multiframe Alignment Network for Image Restoration. IEEE Access. 8:39254-72.

Zeng W.H., Li M. 2020. Crop leaf disease recognition based on Self-Attention convolutional neural network. Comput. Electron. Agr. 172:105341.

Zeng W.H., Li H.D., Hu G.S., Liang D. 2022. Lightweight densescale network (LDSNet) for corn leaf disease identification. Comput. Electron. Agr. 197:106943.

Zhang J., Zhang W. 2010. Support vector machine for recognition of cucumber leaf diseases. Proc. - 2nd IEEE Int. Conf. Adv. Comput. Control. ICACC. 5:264-6.

Zhang S.W., Zhang S.B., Zhang C.L., Wang X.F., Shi Y. 2019. Cucumber leaf disease identification with global pooling dilated convolutional neural network. Comput. Electron. Agr. 162:422-30.

Zhao S.Y., Peng Y., Liu J.Z., Wu S. 2021. Tomato Leaf Disease Diagnosis Based on Improved Convolution Neural Network by Attention Module. Agriculture. 11:651.

Appendix