Research Article

An Improved Crop Disease Identification Method Based on Lightweight Convolutional Neural Network

Tingzhong Wang, Honghao Xu, Yudong Hai, Yutian Cui, and Ziyuan Chen

School of Information Technology, Luoyang Normal University, Luoyang 471934, China

Correspondence should be addressed to Tingzhong Wang; wangtingzhong@lynu.edu.cn

Received 11 February 2022; Revised 7 March 2022; Accepted 28 March 2022; Published 12 April 2022

1. Introduction

Many crop diseases break out and spread quickly, which reduce production greatly and thus do harm to agricultural economy. It is necessary to adopt effective methods for identifying disease timely. The existing methods usually diagnose disease by observing morphological character of crop, because diseased crop has some common features. However, these methods have three shortcomings listed as follows. First, these methods highly depend on experience accumulated by farmers of past dynasties. Many inaccurate viewpoints may be collected and used to identify disease. Second, these methods are time consuming and laborious, indicating that they have poor real-time performance. Furthermore, there are not enough agricultural technicians to identify disease for crop based on these methods of observing morphological character. To improve both the accuracy and efficiency of identifying diseases of crop, an accurate, intelligent, and less time-consuming method should be constructed.

The development of precision and ecological agriculture greatly depends on nondestructive detection and early identification of crop diseases. There are many methods that have been proposed to identify crop disease, such as field observation based on farmers, light spectrum technology [1, 2], detection based on visible light [3], traditional image processing [4], and deep learning–based methods [5]. Light spectrum of crop leaves contains the information about the
condition of crop growing; therefore, this technology can be applied to identify crop disease. Yang and He [1] proposed a diagnosis model of LDA rice blast disease based on hyperspectral technology and the diagnosis accuracy reached 96%. Zhao et al. [2] extracted the chlorophyll and carotenoids from cucumber leaves, and disease of angular leaf spot was diagnosed based on the levels of chlorophyll and carotenoids. Zhong et al. [4] applied greedy learning strategy to improve dynamic Bayesian network (DBN) by changing the monotonicity of the hidden layer, and the improved network has been used for image recognition. Raza et al. [5] designed a remote-detection software system of tomato powdery mildew by combing near infrared visible image depth information and machine learning methods. It can be found that the above-mentioned methods must be performed based on expensive equipment such as spectrometer, which limits their practical applications.

Traditional image recognition methods usually extract feature by hands and then these features are input into classifier for crop disease diagnosis. Much research has been done on feature extraction by hands for crop disease diagnosis and many models have been proposed. Among these diagnosed models, support vector machine (SVM) is intensively studied and used to diagnose disease. Monzurul et al. [6] classified potato plants in the PlantVillage dataset by combining image segmentation and multiclass support vector machine, and the results showed that the accuracy rate of disease classification was 95% for more than 300 images. For example, Tian et al. [7] extracted color of diseased cucumber, constructed the matrix of chromaticity, and then diagnosed disease of cucumber through SVM. Sun [8] collected apple leaf images of three kinds of disease and she conducted research about image acquisition, image preprocessing, eigenvalue extraction, image classification, and diagnosis. In her research, SVM was used for diagnosis and particle swarm optimization algorithm was also used to select optimal parameters for SVM. The accuracy of apple disease images classification reaches 96.9%. Although traditional image recognition is able to satisfy the reequipment of diagnosis, this method is exhausting and time consuming because of feature extraction by hands. With the development of machining learning, many disease diagnosis methods based on deep learning networks can be found.

Deep learning networks are consisting of many layer networks, which are constructed by stacking many layers of neural network. Deep learning networks are trained based on amount of data, and during training, these networks are able to learn features from data effectively. The well-trained networks can be used for forecasting or classification. There are many kinds of neural networks with different structures. Among these networks, backpropagation (BP) neural network is one of the most classical networks. BP neural network usually has three layers and is suitable for extracting shallow features rather than complex features. Huang et al. [9] applied deep learning network to process the rice panicle blast disease images after normalization and the diagnosis accuracy is 92.0%. Zhou [10] proposed a diagnosis method using landsat-8 remote sensing image by combining BP neural network with hyperspectral characteristic analysis. Zhao et al. [11] used BP neural network to diagnose rice leaf curling based on rice characteristic data and the diagnosis accuracy is larger than 90% verified based on 300 data samples. Of course, networks with more layers seem to have better diagnosis performance. But adding layers is a double-edged sword, which brings many issues, simultaneously such as overfitting and lots of parameters to be determined. Therefore, a new network, named convolutional neural network (CNN), that needs less parameters was constructed and studied by researches. CNN was proposed by LeCun et al. [12] and has many merits. First, the original data can be regarded as the input of CNN and features can be automatically extracted from the amount of trained data. Second, CNN has many characteristics including local connection, weight sharing, and pool layer, which helps reduce the complexity of network and the number of parameters to be trained. Furthermore, CNN is robust to pan, twist, and zoom of image data. In addition, original images are directly input into CNN for feature extraction without manual operation, so the shortcomings of traditional methods that feature extraction by hands can be well addressed. After pool operation of the extracted features, the output layer of a classifier is used to diagnose disease by outputting the type of disease.

In recent years, much research has been done on the crop disease diagnosis using CNN. For instance, Krizhevsky et al. [13] diagnosed rice disease using CNN with a classifier of AlexNet achieving good performance with accuracy larger than 90%. Ren et al. [14] introduced deep learning to crop disease diagnosis based on satellite images obtained by UAV Remote Sensing in a long time. In his research, CNN was used to extract features from these images and disease diagnosis model with high accuracy was obtained by tuning parameters and network structure optimizing. The accuracy of the constructed model achieved 97.75% on a dataset of three kinds of crops and outperformed other traditional methods including SVM and BP. CNN is suitable for image recognition, because CNN has many advantages in visual information processing and image recognition shares many common characteristics with pattern recognition. Anand et al. [15] presented two different deep architectures for detecting the type of infection in tomato leaves. Experiments were conducted using PlantVillage dataset comprising three diseases, namely, early blight, late blight, and leaf mold. The proposed work exploited the features learned by the CNN at various processing hierarchy using the attention mechanism and achieved an overall accuracy of 98% on the validation sets in the 5-fold cross-validation. Li et al. [16] proposed two methods which could outperform other deep-learning models on three different datasets, considering several evaluation indicators, and the combination of shallow CNN and classic machine learning classification algorithm is a positive attempt to deal with the plant diseases identification in a simple manner.

This paper proposes a stable and feasible method for crop disease identification. We use deep learning technology [17–19] to learn plant disease characteristics and use TensorFlow [20, 21] to build a convolutional neural network.
model that could accurately identify disease categories. The proposed model is constructed following three steps. First, the PlantVillage dataset including images of apple, cherry, corn, grape, orange, peach, pepper, potato, strawberry, and tomato is preprocessed by cutting normalization and image enhancement is used to remove noise for improving the generalization ability and diagnosis accuracy of CNN. Second, an improved neural network consists of 9 layers that are trained based on these images. During the training, Adam optimizer combining momentum with RMSprop algorithm is used to tune learning rate dynamically, which leads to faster convergence of loss function and improves the learning efficiency. The PlantVillage dataset including 38 kinds of plant categories is used to demonstrate the effectiveness of the proposed method and the diagnosis accuracy is over 95%. Therefore, the proposed method is effective and can be used for crop disease diagnosis. The main contributions of our method are shown as follows:

1. The dataset used in our paper contains three pre-processing steps, including image cutting, normalization processing, and data enhancement, which could greatly accelerate the convergence rate and improve the generalization ability and robustness of the network.
2. Adam optimizer combining momentum algorithm and RMSProp algorithm is used to dynamically adjust the learning rate, and the combination of the two algorithms makes the loss function converge to the lowest point faster.
3. The feature map after each convolution is obtained by using transferred revolution, and the model is adjusted according to the feature map to further improve the effect of model recognition.

2. Reviews on Crop Disease Diagnosis Using CNN

The research on artificial intelligence has grown rapidly since Hinton proposed the concept of deep learning. Compared with traditional machine learning, deep learning [22–25] is helpful for both timeliness and accuracy. CNN is a special neural network which consists of convolution calculation and deep network structures, which is one of the most used artificial intelligent networks nowadays and has achieved good performance in many computer vision tasks including image classification and target detection [26, 27]. CNN was proposed by a Japanese scholar named Fukushima [28] and a LeNet-5 CNN was constructed to address the issue of number recognition by LeCun [29]. In 2021, Amara et al. proposed a classical model called AlexNet [30] which made great breakthroughs in the field of image recognition. Long et al. [31] designed a LeNet system for classification based on CNN for diagnosing bananas disease. Kawasaki et al. [32] employed deep CNN to extract features of diseased camellia oleifera and then applied the image dataset to train AlexNet with the help of transfer learning methods, which was able to diagnose disease with high accuracy. Ramcharan et al. [33] designed a plant disease detection system based on CNN. The system was applied to three kinds of cucumber leaf images with a number of 800 and accuracy of 94.9% was obtained.

LeCun et al. [12] performed transfer learning based on CNN for diagnosing disease including brown cercosporin leaf spot (BLS), red spider mite damage (RMD), green spider mite damage (GMD), cassava brown streak disease (CBSD), and mosaic disease (CMD) of Cassava. It can be found that transfer learning with Inception V3 can achieve the best performance during training and the test accuracy is 93%. CNN is effective to improve the efficiency and accuracy of diagnosis. Many new networks have been proposed based on CNN such as LeNet [34], AlexNet [30], GoogLeNet [35], ResNet [36], and VGG [37]. The structure of networks tends to be deeper and more complex, which addresses the issue of gradient disappearance and explosion. Oppenheim et al. [19] captured images by low-cost standard RGB sensors and 2465 images were collected, classified, and labeled by agricultural expert. Then the CNN was used to diagnose potato diseases. Liu et al. [38] used CNN to diagnose rice sheath blight and compared its performance with SVM. The result showed that CNN achieves an accuracy of 97%, while the performance of SVM is 95%. Chen et al. [39] proposed a convolutional neural network model for diagnosing corn plant diseases based on the combination of data enhancement and transfer learning. The algorithm first improves the generalization and accuracy of the model through data enhancement and then constructs a convolutional neural network model based on transfer learning to extract disease image features, which accelerates the training process of convolutional neural network and reduce the overfitting degree of the network. Finally, the model was applied to the accurate recognition of corn pest images. The results show that the average recognition accuracy of the convolution neural network optimization algorithm based on data enhancement and migration learning is 96.6%. Compared with the single convolutional neural network, the accuracy is improved by 25.6%, and the processing time of each image is 0.28 s, which is nearly 10 times shorter than the traditional neural network. The accuracy and training speed of the algorithm are significantly higher than the traditional convolutional neural network, which provides a new method for diagnosing plant diseases of crops such as corn.

Zhang et al. [40] proposed a wheat disease diagnosis method based on convolutional neural network. Firstly, based on the image data of wheat disease, the method used median filtering method and histogram threshold method to preprocess the image, such as background removal, denoising, and disease spot segmentation, for obtaining a sample database. Then, a five-layer deep learning model is constructed based on convolution neural network for feature extraction and the random gradient descent method is used to control the learning process. The experimental results at the sample points show that this method can effectively diagnose commonly seen diseases of wheat sheath blight, stripe rust, leaf rust, stem rust, scab, and powdery mildew. Bao [41] designed a feature fusion module to concatenate the downsampled feature maps output by inverted residual blocks and the average pooling features of the feature maps,
use the input inverted residual blocks to realize the fusion between features of different depths to reduce the loss of the detailed features of wheat ear diseases caused by the networks in the downsampling process, and solve the disappearance of disease features in the process of image feature extraction. Experimental results show that the proposed SimpleNet model achieved an identification accuracy of 94.1% on the test dataset. In summary, much work has been done based on CNN for diagnosing disease of different kinds of crops and all of them achieved good performance which is verified by experiments. Considering many great advantages holding by CNN, an improved CNN network, i.e., stacked CNN, is constructed in this paper. The PlantVillage dataset including leaf images of apple, cherry, corn, grape, orange, peach, pepper, potato, strawberry, and tomato is used to verify the effectiveness of the proposed method.

3. Data Preprocessing

Being the fundament of neural network, data preprocessing is included in the proposed method to improve the quality of dataset. Data preprocessing contains three steps including image cutting, normalization processing, and data enhancement. Image cutting is used to obtain images consisting of the size of input nodes, which not only ensure the trainability of neural networks but also reduce useless pixels of images. Normalization processing is useful to accelerate the convergence rate and avoid gradient disappearance. Data enhancement can be used to enlarge dataset and thus enhance the generation and accuracy of the constructed model. The steps are detailed in Figure 1 and described as follows.

3.1. Image Cutting. Parameters weight and bias contained in the full-connection layer will be considered during the training of networks. The number of these parameters depends on the size of inputting images. The changing dimensions of weight and bias influence the speed of learning and convergence. To ensure the trainability of network, images should be processed by image cutting. Furthermore, image cutting is able to remove redundancy features and make the training less time consuming; Figure 2 shows the image cutting of some plant leaves.

3.2. Normalization Processing. Normalization processing is used to map features of image into one limited domain. The dataset after normalization processing can remove the negative influence of anomaly data, which makes all the data have similar distribution, accelerates the convergence rate, and avoids gradient disappearance. In the proposed method, the disease image data are normalized to $[-1, 1]$ with average value of 0 and standard deviation of 1; the normalization processing is shown as:

$$X = \frac{(X - \mu)}{\sigma},$$  \hspace{1cm} (1)

Here $\mu$ represents the average value and $\sigma$ denotes the standard deviation; Figure 3 shows the effect of data normalization on gradients.

3.3. Data Enhancement. A neural network with good performance usually contains amounts of parameters, the number of which may achieve the order of millions, and enough data should be presented to trained neural networks by fitting these parameters. Data enhancement can be used to process images for obtaining amounts of trained samples and thus the generalization and accuracy can be improved. The data enhancement used in the proposed method includes four aspects shown as follows.

3.3.1. Contrast Transformation. This operation is used to change the brightness, saturation, and contrast of image. Concretely, in the HSV color space of the image, change the saturation $S$ and brightness $V$, while keeping the hue $H$ unchanged. The exponential operation is applied to $S$ and $V$ of each pix and the enhanced factor varies from 0.25 to 4 to increase the brightness. Figure 4 is the example of contrast transform.

3.3.2. Image Scale Change. Scale and aspect ratio change of length and width: images are scaled according to the determined factor; then, a setting factor is used for image filtering and constructing scale space, changing the size and fuzzy degree as shown in Figure 5.

3.3.3. Image Geometry Transformation. This transformation contains flip, translate, etc. For one image, first flip the image along the horizontal or vertical direction, and the enhancement factor is 2. Second, translation translates the image in a certain way on the image plane. The translation range and step are random, and the translation is carried out along random direction. Last, change the position of the image content as shown in Figure 6.

Experiments show that data after processing could greatly improve the generalization ability and robustness of the network and achieve better performance. The expanded dataset contains 236965 images, including 10 crop classifications which totally have 27 disease types. Based on the data...
Figure 2: Image cutting.

(a) Nonuse Normalization processing

(b) Use Normalization processing

Figure 3: The effect of data normalization on gradients.

Figure 4: Contrast transform.

Figure 5: Scale transformation.
4. Crop Disease Identification Based on an Improved CNN

4.1. Overview of the Improved CNN. The structure of the proposed CNN is shown as Figure 7. The diagnosis method is constructed based on CNN which consists of average pooling, batch normalization, ReLU, dropout, and data enhancement. Batch normalization is used after each convolution layer. Each convolution layer of the proposed model includes convolution operation and local normalization. After convolution operation, ReLU activation function, average pooling, and dropout operation are applied to features. The parameter settings are shown in Table 1; the first layer convolution uses 16 convolution kernels with the size of $7 \times 7$ for feature extraction and the same type of padding is used. Since the size and input of the feature matrixes are the same after the “same” convolution, 16 feature matrixes with $64 \times 64$ are obtained after the first convolution. After convolution, applying batch normalization will increase the value of local larger neurons, reduce the value of smaller neurons, and increase the generalization ability of the model. The processing method of the second layer convolution is the same as above, but the application of ReLU activation function is carried out after convolution to increase the expression ability of the model. The expression of ReLU activation function is shown in

$$f(x) = \max(0, x). \quad (2)$$

After ReLU operation, the feature map is pooled, and the feature map is scaled through the pooled operation. The size of the pooled output feature map is $64 \times 64 \times 16$. Then, the dropout with a scale of 0.5 is used to prevent overfitting of the model. Afterwards, the convolution layer structure is the same as the above, but just it adjusts the number of convolution cores. The model adds global average pooling at the end of convolution operation and applies SoftMax regression to the feature graph after pooling operation. The SoftMax function is shown as follows:

$$y_k = \frac{\exp(a_k)}{\sum_{i=1}^{n} \exp(ai)}. \quad (3)$$

A vector with a size of $(X, 1)$ could be obtained and $X$ denotes the number of categories. The output of neural network represents the probability value of being the corresponding category. There are totally 715958 parameters, of which 714486 parameters need to be trained.

4.2. Trained Process. The CNN model used in this paper includes nine convolution layers, and the training process is consisting of three stages. The first stage is to load data samples and divide dataset into the training set and verification set, respectively. The second stage is to set the learning rate, image enhancement, and training of CNN model. Last, loss and accuracy data are saved in the training and evaluate the accuracy of the CNN model; the process of the model training is shown in Figure 8.

The implementation is detailed as follows:

(1) The first stage: all the experimental samples are loaded and divided into training set and verification set according to the proportion of 8:2. LabelBinarizer is used for the label samples, while binarization is carried out to process samples.

(2) The second stage: the neural network is trained by using the training set under supervised learning. First, the forward propagation stage is performed, which initializes the weight of the network, and the input data are propagated forward through the convolution layer, the lower sampling layer, and the full-connection layer to obtain the output value. Then, in the backpropagation stage, the error between the output value and the target value of the network is calculated, and the weight of the network is updated according to the obtained error. Through the above operations, the weight is continuously updated based on the algorithm of gradient descent. The difference between the predicted value and the real value of the model is defined as the loss function. The goal of gradient descent is to find the optimized parameters with the minimum loss function, under which the predicted value is closest to the real value.
Figure 7: Overview of the improved CNN.

Table 1: Parameters setting in the proposed CNN.

| Parameters                  | Information          |
|-----------------------------|----------------------|
| Activation function         | ReLU                 |
| Optimizer                   | Adam                 |
| Convolution type            | Same                 |
| Convolution kernel          | $3 \times 3$, $5 \times 5$, $7 \times 7$ |
| Convolution stride          | 1                    |
| Pooling type                | Average pooling, global average pooling |
| Pooling stride              | 2                    |
| Pooling size                | $2 \times 2$          |

Figure 8: The flowchart of model training.
The activation function selects the ReLU as the activation function, which can speed up the descent of the gradient and prevent the gradient from disappearing. Feature extraction adopts convolution and pooling operation. The feature extraction formula of convolution is given by

$$F = f \left( \sum_{i=1}^{m} \text{con}(M_i, C_i) + B_i \right).$$

(4)

There are two important parameters including weight (w) and bias (b) in the loss function. One is to control the weight of the input data, and the other is to adjust the deviation between the function and the real value. The iteration is used for minimizing loss function by finding the optimal w and b. For each sample data, a weight gradient can be obtained. Add up the weight gradients of each sample data, and calculate their average value which can be considered as the weight gradient of the whole sample. After determining the descending direction of the overall gradient, the learning rate is required to determine the step size of each gradient descent. \(w_i\) represents the weight value after each update and \(\alpha\) represents the learning rate, and the iterative formula is shown as

$$w_{i+1} = w_i - \alpha \frac{\partial L}{\partial w_i}.$$

(5)

In order to accelerate the convergence speed of the loss function, the proposed method uses the Adam optimizer to combine the momentum algorithm and RMSprop algorithm, dynamically adjusting the learning rate to minimize the loss function. As a result, the proposed method has faster convergence and less oscillation during training, which greatly reduces the time required for training and improves the accuracy of model recognition. The process can be calculated according to the following equations:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_w f (W_t),$$

(6)

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \nabla_w f (W_t)^2,$$

(7)

$$m_t^\wedge = \frac{m_t}{1 - \beta_1^t},$$

(8)

$$v_t^\wedge = \frac{v_t}{1 - \beta_2^t},$$

(9)

$$W_t = W_{t-1} - \eta \frac{m_t^\wedge}{\sqrt{v_t^\wedge} + \epsilon},$$

(10)

(3) Stage 3: use categorical crossentropy as the loss function of multiclassification and save the loss and accuracy data generated during model training.

4.3. Optimization of Neural Network. As the core parameter of neural network training, learning rate directly affects the consuming time of model training and the accuracy of the model. If a large learning rate is considered during training, the real minimum value of the loss function may be missed. Conversely, the minimum value is obtained after amounts of iterations under a too small learning rate. In addition, Adam optimizer is used to combine momentum algorithm and RMSprop algorithm to minimize the loss function by dynamically adjusting the learning rate, which is able to improve the performance of the proposed method.

4.3.1. Momentum Algorithm. Momentum algorithm is performed by introducing the concept of momentum in physics. The gradient of each iteration is calculated by considering the gradient of the previous several iterations. That is to say, a new variable \(V\) (velocity) is introduced, which is the accumulation of the previous gradient, but there is a certain attenuation in each iteration. We call \(V\) as the momentum term, which is equivalent to adding inertia to the gradient descent. In this case, the calculated gradient not only is based on the current position, but still has the previous gradient amount. Then once the valley terrain is encountered, according to the previous gradient speed, there is still a certain gradient in the direction of the minimum value, so the iteration will be faster to the minimum value. When gradient descent reaches the local optimum, the gradient is 0, and since there is a momentum term, it is equivalent to having a certain initial velocity at this time, so it can rely on “inertia” to rush out of the local optimum point and then update to the minimum value. This algorithm largely eliminates the local optimization phenomenon in the process of gradient descent and makes the gradient update move in a more and more clear direction.

In each iteration, we randomly select a batch of samples \(\{x_1, x_2, \ldots, x_n\}\) with a capacity of \(m\) from the training set and the related output \(y_i\), and then calculate the gradient and error. The calculation formula is as follows:

$$\hat{g} \leftarrow \frac{1}{m} \nabla \theta \sum_i L(f(x_i; \theta), y_i).$$

(11)

Here \(\epsilon, \theta, v, \alpha\) represent learning rate, initial parameters, initial velocity, and momentum attenuation parameter, respectively. Update the speed \(v\) and parameters \(\theta\) the iterative update formula is as follows:

$$v_t \leftarrow \alpha v_{t-1} - \epsilon \hat{g},$$

(12)

$$\theta \leftarrow \theta - v_t.$$
faster convergence and less oscillation. Compared with the standard gradient descent algorithm, momentum algorithm has faster convergence speed, shown in Figure 9.

4.3.2. RMSprop Algorithm. Root mean square prop (RMSprop) can be used to further optimize the loss function, which addresses the issue that the swing amplitude is too large during updating. Furthermore, RMSprop further accelerates the convergence speed of the function by introducing an attenuation coefficient to make \( r \) attenuate with a certain proportion in each iteration.

For each iteration, a batch of samples \( \{x_1, x_2, x_m\} \) with a capacity of \( M \) is randomly selected from the training set, as well as the relevant output. The gradient and error are calculated as follows:

\[
\hat{\theta} \leftarrow \frac{1}{m} \sum_i \nabla \theta L(f(x_i; \theta), y_i) \tag{13}
\]

Here \( \epsilon \) denotes global learning rate, \( \theta \) is initial parameters, \( \delta \) is numerical stability, \( \rho \) is decay rate, and \( y \) represents gradient cumulant, which usually initialized as 0. Update the parameters according to the calculation results using the following equations:

\[
r \leftarrow r + (1 - \rho) \delta \odot \hat{\theta} \\
\Delta \theta = -\frac{\epsilon}{\delta + \sqrt{r}} \odot \hat{\theta} \tag{14}
\]

\[
\theta \leftarrow \theta + \Delta \theta.
\]

Before updating the gradient, first use the gradient to calculate a weighted average, which is equivalent to normalizing the gradient. This avoids the learning rate become smaller after each iteration, and thus it can adjust the learning rate adaptively. Specifically, it helps to eliminate the direction with large swing amplitude and is used to correct the swing amplitude, so that the swing amplitude of each dimension is small. On the other hand, it also makes the convergence of network function faster. Therefore, RMSprop can adjust the step size in different dimensions, reducing the oscillation amplitude and accelerating the convergence speed; the updating is described using Figure 10.

4.4. PlantVillage Dataset. In this paper, two datasets, including PlantVillage [40] and image database in crop disease recognition competition in global AI challenge, will be used to verify the effectiveness of the proposed method. These two datasets contain a variety of leaf images of crop health and disease and are widely studied in the research of crop disease diagnosis. Bao et al. [41] performed image recognition on the PlantVillage dataset with an accuracy of 93.95%. Zhang et al. [42] diagnosed crop disease dataset in global AI challenge with an accuracy of 83%. PlantVillage is build based on the crop Q and a forum jointly established by epidemiologists David Hughes and Marcel Salthe of Binzhou University. The dataset contains a total of 14 plant categories such as apple, blueberry, cherry, peach, corn, grape, and orange. A total of 54306 crop leaf images are collected with 26 kinds of crop disease and 12 kinds of health plant. Some datasets are shown in Figure 11.

Moreover, the image database contains the health and disease images of ten crops including apple, cherry, corn, grape, orange, peach, pepper, potato, strawberry, and tomato. According to the priority of species, disease and degree, all the datasets can be divided into 61 classifications, including 27 diseases and 10 plant classifications. The crop leaves are located in the center position of each picture. The dataset contains 47393 pictures, which are subdivided into four subdatasets: training, test a, test b, and verification.

The display of positive and negative samples between different species in the data set is shown in Figure 12. The corresponding relationship between tag ID and tag name of some images in the training, testing, and verification dataset is shown in Table 2. The distribution of number for ten crops is shown in the left of Figure 13, while the number of each category in 61 classifications of the dataset according to species, disease, and degree is shown in the right side of Figure 13.

4.5. Computing Platform. The proposed method and compared methods are run based on the software including Python 3.7 and TensorFlow 1.13. In addition to TensorFlow 1.13, deep learning framework also includes keras 2.2.0, Opencv-python 4.3.1. The hardware environment is listed as follows: operating system win10 64-bit, six core processor, 32g memory, graphics card NVIDIA GeForce GTX 2080ti, CUDA API with the version of 11.1.102, cuDNN with the version of 8.0.4.

4.6. Results Analysis. The experiment was conducted for six hours on the workstation and 70 iterations were carried out on the full PlantVillage dataset, with 43442 training set samples and 10861 verification set samples in total. The accuracy and loss are shown in Figure 14; it could be seen that the blue curve in the figure is the training set and the red curve is the test set. With the increase of iterations, the accuracy of training and verification increases step by step, tends to be stable after 60 iterations, and reaches a satisfying accuracy. At this time, the convolution rate of the super convolution parameter set has been accurately adjusted by using the number of times of the training set, but the deconvolution rate has not been improved by a little. At this time, the deconvolution rate of the super convolution parameter set has been adjusted by using the results of each iteration, and the deconvolution rate has not been improved by a little, so as to improve the accuracy of model recognition. After several iterations, the model tends to be stable in both the training set and test set, without large fluctuations, and achieves an ideal convergence effect. Finally, after all the iterations are completed, the accuracy of the model is about 95%. It can also be seen intuitively from the loss curve that the loss rate of training and verification decreases with the increase of iteration times, tending to a stable state after
Figure 9: Optimization of neural network.

Figure 10: Optimization of neural network.

Figure 11: Partial data of PlantVillage.

Figure 12: Partial dataset of PlantVillage.
Table 2: Some labels of PlantVillage dataset.

| Label ID | Label name                        |
|----------|-----------------------------------|
| 0        | apple healthy                     |
| 1        | Apple_Scab general                |
| 2        | Apple_Scab serious                |
| 3        | Apple Frogeye Spot               |
| 4        | Cedar Apple Rust general          |
| 5        | Cedar Apple Rust serious          |
| ...      | ...                               |
| 55       | Tomato_Septoria Leaf Spot Fungus serious |
| 56       | Tomato Spider Mite Damage general |
| 57       | Tomato Spider Mite Damage serious |
| 58       | Tomato YLCV Virus general         |
| 59       | Tomato YLCV Virus serious         |
| 60       | Tomato Tomv                       |

Figure 13: Plant species distribution and disease species distribution.

Figure 14: Curves of accuracy and loss.
60 iterations, and the loss value also decreases and reach a small value.

After the verification using all the dataset, we further carry out analysis on the proposed method based on single data subset. First, a separate result analysis of the apple subset in PlantVillage is used to show the performance of the proposed method.

The apple subset contains four classifications: apple scab, apple black rot, apple cedar rust, and apple health. Among them, there were 630 images of apple scab, 621 images of apple black rot, 275 images of apple cedar rust, and 1645 images of apple health. The label distribution of data is shown in Figure 15.

The same model training in the full dataset is applied to apple dataset alone and the number of iterations is also set to seventy. The accuracy and loss rate curve of apple data subset is shown in Figure 16. The blue curve in the figure represents the training set and the red curve represents the verification set. It is obvious from the accuracy image that the accuracy is getting higher and higher with the increase of training iterations. The loss curve shows that the loss value decreases faster with the increase of training iterations. The curve of verification set fluctuates greatly and tends to be stable after 53 training iterations. Finally, the model modified in this paper achieves 97.3% accuracy in the verification set of apple data subset.

In order to observe the performance of the model more clearly, the proposed method adopts an index commonly used in classification named confusion matrix shown as Figure 17. This is a special contingency table with two dimensions including real value and predicted value; and these two dimensions have the same set of categories. After training the apple data subset, this paper uses the conflict under the machine learning library_Matrix method further drawing apple's confusion matrix. The row column joint table represents the category of apple diseases. Here, 0–3 respectively correspond to four categories: apple health, cedar apple rust, apple black rot, and apple scab. The color depth of each grid represents the recognition accuracy of the corresponding category. It is obvious from the figure that the corresponding recognition accuracy of each category is 88%, 93%, 99%, and 100%, respectively. The experimental results finally show that the model has a good accuracy for each category in the apple dataset.

The above experiments show that the model has good training results for a single apple dataset. In order to verify that the model also has better training results for other data subsets, this paper further applies the model to different data subsets of PlantVillage. The dataset includes grape, corn, potato, and tomato datasets, which have different numbers of diseases, and the accuracy rate of different subset is shown as Table 3. There are 4 species in grape dataset, 4 species in corn dataset, 3 species in potato dataset, and 10 species in tomato dataset. The specific experimental steps are consistent with the operation on the apple dataset. First, the dataset is preprocessed, and then the model is trained with 70 iterations to analyze and compare the experimental results. When curves are stable, both accuracy and loss have good results. The final model obtained 96.7% recognition accuracy on grape dataset, 95.1% recognition accuracy on corn dataset, 94.6% recognition accuracy on potato dataset, and 95.8% recognition accuracy on tomato dataset. Experiments show that the model has good accuracy for a single dataset. The following table shows the recognition accuracy of diagnosis on the dataset using the proposed model.

Taking apple dataset as an example, the recognition accuracy of each category under the dataset can be clearly obtained through the confusion matrix. This paper further presents the confusion matrix on the above four datasets: grape, corn, potato, and tomato datasets and then analyzes the recognition effect of the model on each category under a single dataset. This paper uses the conflict under the machine learning library_Matrix method drawing a confusion matrix similar to the above apple dataset. The row column table represents the real and predicted labels, and the color depth of each grid under the matrix represents the recognition accuracy. Through the confusion matrix, it can be clearly understood that under the grape dataset, the recognition accuracy of grape black rot is 100%, that of Grape Black measures is 96%, that of grape leaf spot is 98%, and that of grape health is 84%. Under the corn dataset, the accuracy of Corn Gray Leaf Spot category is 98%, the accuracy of corn common trust is 100%, the accuracy of corn health is 89%, and the accuracy of corn northern leaf bright is 100%. Under the potato dataset, the recognition accuracy of potato early blight category is 100%, that of potato late blight is 82%, and that of potato health is 97%. Under the tomato dataset, the recognition accuracy of tomato bacterial spot is 78%, that of tomato early blue is 85%, that of tomato health is 88%, that of tomato late blue is 92%, that of tomato leaf mold is 94%, that of tomato Septoria leaf spot is 79%, and that of two spotted tomato spider mites is 93%. The recognition accuracy of tomato target spot, tomato mosaic virus, and tomato yellow leaf curl virus is 100%. Experiments show that it also has good performance for different categories under different data subsets. The confusion matrix of grapes, potatoes, corn, and tomatoes is shown in Figure 18.

Deconvolution is of great significance when adjusting parameters of models. Deconvolution, also known as transformed convolution, is the inverse operation of convolution. It is difficult to know the features learned by each layer of convolution in convolution operation of CNN and neural networks are usually considered as black box process. To improve the recognition effect of the model, it is necessary to understand the feature information obtained by the network model after deconvolution of each layer. Through the feature map, the super parameters of CNN can be adjusted accurately. Figure 19 display the visualization results of the feature map obtained by deconvolution of the leaves of apple scab. It can be observed that with the deepening of the number of convolution layers, the crop leaf information learned by the network becomes more specific.

Finally, in order to verify the effectiveness of the proposed model, in the experiment, some network models including vgg-19, AlexNet and ResNet-50, which are commonly used in image recognition, are applied to diagnose the crop disease of this dataset for comparison. The results show that vgg19 model has an accuracy of 92.4% for
the training set, 89.9% for the verification set, and 87.8% for the test set; AlexNet model has 86.5% accuracy for the training set, 85.8% accuracy for the verification set, and 82.6% accuracy for the test set; Resnet50 model has 88.6% accuracy for the training set, 86.8% accuracy for the verification set, and 83.2% accuracy for the test set. The improved Xception model, Tiny_Xception model, has 96.3% accuracy in the training set, 95.7% accuracy in the verification set, and 94.3 accuracy in the test set. Through analysis, the improved Xception model in this paper is superior to
other similar models in training set, verification set, and test set. The experimental results show that the improved Xception model proposed in this paper has better performance than the traditional image recognition model, which verifies the effectiveness of this model again. The overall results are shown in Table 4; we could see that our improved Xception model achieves the best performance in all the dataset including training set, verification set, and test set.

Table 3: Accuracy rates of different subsets.

| Plant name     | Number of types | Accurate rate (%) |
|----------------|-----------------|-------------------|
| Apple          | 4               | 97.3              |
| Corn           | 4               | 95.1              |
| Potato         | 3               | 94.6              |
| Grape          | 4               | 96.7              |
| Tomato         | 10              | 95.8              |

Figure 18: Confusion matrix of different subsets: (a) confusion matrix of grape disease, (b) confusion matrix of corn disease, (c) confusion matrix of potato disease, and (d) confusion matrix of tomato disease.
5. Conclusion

In this paper, an improved CNN is proposed for crop disease diagnosis. The proposed model applies computer image processing to crop disease recognition and uses deep learning to extract more intelligent and robust features. Specifically, convolution neural network and deep separable convolution are used to build a deep convolution neural network to diagnose crop disease, and CNN is used to train models based on crop datasets. The optimizer is used to speed up the parameter learning speed and the perceptron is inactivated randomly to prevent overfitting. Furthermore, image enhancement is used to improve the generalization ability of the model, and the dataset preprocessing is used to speed up the network training process. After 70 rounds of training in PlantVillage dataset, the convolutional neural network has achieved a recognition effect of 95.85% and has achieved a recognition rate of about 96% in apple, corn, grape, and other single classes. Experiments show that the classification accuracy of the improved model is significantly higher than that of other network models for Plantsville dataset. However, due to the limitation of dataset, there are few recognition types, and because the number of partial category data is small, the neural network cannot learn enough features, so the recognition accuracy is not ideal. We will continue to look for datasets with more abundant crop categories and more data, constantly enrich the crop categories that can be identified by the model, and improve the accuracy of identification.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Science and Technology Key Project of Henan Province under grant no. 212102210520.

References

[1] Y. Yang and Y. He, “Early prediction of antioxidant enzyme value of rice blast based on hyper-spectral image,” Transactions of the Chinese Society of Agricultural Engineering, vol. 29, no. 20, pp. 135–141, 2013.
[2] Y. R. Zhao, X. Li, K. Q. Yu, F. Cheng, and Y. He, “Hyper-spectral imaging for determining pigment contents in cucumber leaves in response to angular leaf spot disease,” Scientific Reports, vol. 6, no. 1, pp. 1–9, 2016.
[3] G. Prince, J. P. Clarkson, and N. M. Rajpoot, “Automatic detection of diseased tomato plants using thermal and stereo visible light images,” PloS One, vol. 10, no. 4, pp. 1–20, 2015.
[4] P. Zhong, Z. Gong, S. Li, and C.-B. Schonlieb, “Learning to diversify deep belief networks for hyperspectral image classification,” IEEE Transactions on Geoscience and Remote Sensing, vol. 55, no. 6, pp. 3516–3530, 2017.
[5] S. E. A. P. Raza, J. P. Clarkson, and N. M. Rajpoot, “Automatic detection of diseased tomato plants using thermal and stereo visible light images,” PloS One, vol. 10, no. 4, Article ID e0123262, 2015.
[6] M. Islam, A. Dinh, K. Wahid, and P. Bhowmik, “Detection of potato diseases using image segmentation and multiclass support vector machine,” in Proceedings of the 2017 IEEE 30th canadian conference on electrical and computer engineering (CCECE), pp. 1–4, IEEE, Windsor, ON, Canada, May 2017.
[7] Y. Tian, C. Zhang, and C. Li, “Application of support vector machine in plant lesion shape recognition,” Journal of agricultural engineering, vol. 20, no. 3, pp. 134–136, 2004.
[8] S. Sun, Classification of Apple Leaf Diseases Based on Image Processing and Support Vector Machine [D], Xi’an University of science and technology, Xian, China, 2017.
[9] S. Huang, C. Sun, and L. Qi, “Detection method of rice ear blast based on deep convolution neural network,” Journal of agricultural engineering, no. 20, pp. 169–176, 2017.

[10] Y. Zhou, Extraction of Potato Spatial Distribution Based on BP Neural Network [D], Chinese Academy of Agricultural Sciences, Beijing, China, 2018.

[11] R. Zhao, C. Qi, and L. Duan, “Rice leaf roll recognition based on BP neural network,” Journal of southern agriculture, vol. 49, no. 10, 2018.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proceedings of the 25th International Conference on Neural Information Processing Systems. Lake Tahoe, vol. 60, no. 6, pp. 1097–1105, ACM, 2012.

[14] Y. Ren, X. Zhang, Y. Ma et al., “Full convolutional neural network based on multi-scale feature fusion for the class imbalance remote sensing image classification,” Remote Sensing, vol. 12, no. 21, p. 3547, 2020.

[15] S. Anand, P. Mathikshara, and A. Johnson, “Attention embedded residual CNN for disease detection in tomato leaves,” Applied Soft Computing, vol. 86, Article ID 105933, 2020.

[16] Y. Li, J. Nie, and X. Chao, “Do we really need deep CNN for plant diseases identification?” Computers and Electronics in Agriculture, vol. 178, Article ID 105803, 2020.

[17] J. Ma, K. Du, F. Zheng, L. Zhang, Z. Gong, and Z. Sun, “A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network,” Computers and Electronics in Agriculture, vol. 154, pp. 18–24, 2018.

[18] A. K. Rangarajan, R. Purushothaman, and A. Ramesh, “Tomato crop disease classification using pre-trained deep learning algorithm,” Procedia Computer Science, vol. 133, pp. 1040–1047, 2018.

[19] D. Oppenheim, G. Shani, O. Erlich, and L. Tsror, “Using deep learning for image-based potato tuber disease detection,” Phytopathology, vol. 109, no. 6, pp. 1083–1087, 2019.

[20] A. Martin, A. B. Ashish, and B. C. Eugene, 2015.

[21] J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, and M. Kudlur, “TensorFlow: a system for large-scale machine learning,” in Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation, A. Martin, B. Paul, C. Jianmin, C. Zhifeng, and D. Andy, Eds., Savannah GA USA, November 2014.

[22] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in Proceedings of the 13th European conference on computer vision Zurich: ECCV, Zurich, Switzerland, September 2014.

[23] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” Neural Computation, vol. 18, no. 7, pp. 1527–1554, 2006.

[24] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” Science, vol. 313, no. 5786, pp. 504–507, 2006.

[25] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, Vol. 1, MIT press, Cambridge, USA, 2016.

[26] J. Gu, Z. Wang, J. Kuen et al., “Recent advances in convolutional neural networks,” 2015.

[27] K. Fukushima, “Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” Biological Cybernetics, vol. 36, no. 4, pp. 193–202, 1980.