

Research Article

Can Deep Learning Identify Tomato Leaf Disease?

Keke Zhang ¹, Qiufeng Wu ², Anwang Liu ¹, and Xiangyan Meng ²

¹College of Engineering, Northeast Agricultural University, Harbin 150030, China

²College of Science, Northeast Agricultural University, Harbin 150030, China

Correspondence should be addressed to Qiufeng Wu; qfwu@neau.edu.cn

Received 9 June 2018; Accepted 30 August 2018; Published 26 September 2018

Academic Editor: Alexander Loui

Copyright © 2018 Keke Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper applies deep convolutional neural network (CNN) to identify tomato leaf disease by transfer learning. AlexNet, GoogLeNet, and ResNet were used as backbone of the CNN. The best combined model was utilized to change the structure, aiming at exploring the performance of full training and fine-tuning of CNN. The highest accuracy of 97.28% for identifying tomato leaf disease is achieved by the optimal model ResNet with stochastic gradient descent (SGD), the number of batch size of 16, the number of iterations of 4992, and the training layers from the 37 layer to the fully connected layer (denote as “fc”). The experimental results show that the proposed technique is effective in identifying tomato leaf disease and could be generalized to identify other plant diseases.

1. Introduction

Tomato is a widely cultivated crop throughout the world, which contains rich nutrition, unique taste, and health effects, so it plays an important role in the agricultural production and trade around the world. Given the importance of tomato in the economic context, it is necessary to maximize productivity and product quality by using techniques. *Corynespora* leaf spot disease, early blight, late blight, leaf mold disease, septoria leaf spot, two-spotted spider mite, virus disease, and yellow leaf curl disease are 8 common diseases in tomato [1–8]; thus, a real time and precise recognition technology is essential.

Recently, since CNN has the self-learned mechanism, that is, extracting features and classifying images in the one procedure [9], CNN has been successfully applied in various applications, such as writer identification [10], salient object detection [11, 12], scene text detection [13, 14], truncated inference learning [15], road crack detection [16, 17], biomedical image analysis [18], predicting face attributes from web images [19], and pedestrian detection [20], and achieved the better performance. In addition, CNN is able to extract more robust and discriminative features with considering the global context information of regions [10], and CNN is scarcely affected by the shadow, distortion, and brightness

of the natural images. With the rapid development of CNN, many powerful architectures of CNN emerged, such as AlexNet [21], GoogLeNet [22], VGGNet [23], Inception-V3 [24], Inception-V4 [25], ResNet [26], and DenseNets [27].

Training deep neural networks from scratch needs amounts of data and expensive computational resources. Meanwhile, we sometimes have a classification task in one domain, but we only have enough data in other domains. Fortunately, transfer learning can improve the performance of deep neural networks by avoiding complex data mining and data-labeling efforts [28]. In practice, transfer learning consists of two ways [29]. One option is to fine-tune the networks weights by using our data as input; it is worth nothing that the new data must be resized to the input size of the pretrained network. Another way is to obtain the learned weights from the pretrained network and apply the weights to the target network.

In this work, first, we compared the performance between SGD [30] and Adaptive Moment Estimation (Adam) [30, 31] in identifying tomato leaf disease. These optimization methods are based on the pretrained networks AlexNet [21], GoogLeNet [22], and ResNet [26]. Then, the network architecture with the highest performance was selected and experiments on effect of two hyperparameters (i.e., batch size and number of iterations) on accuracy were carried out. Next,

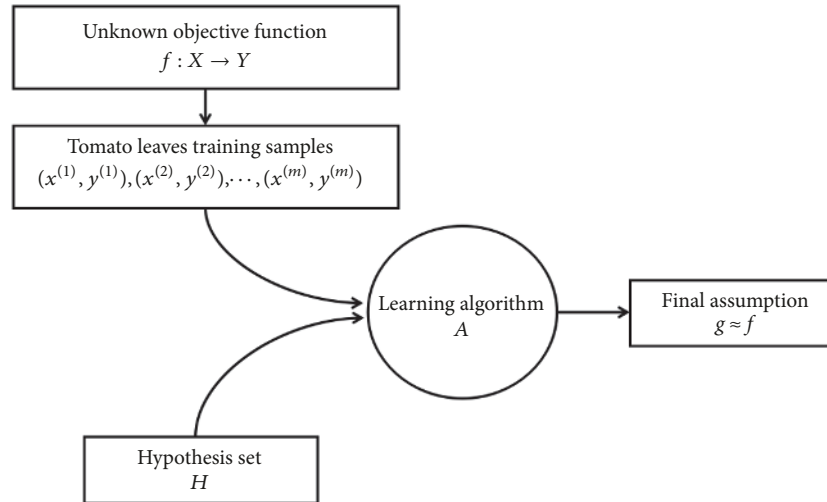


FIGURE 1: Proposed workflow diagram.

we utilized the network with the suitable hyperparameters, which was obtained from the previous experiments, to discuss the impact of different network structures on recognition tasks. We believe this makes sense for researchers who choose to fine-tune pretrained systems for other similar issues.

The rest of this paper is organized as follows. Section 2 displays an overview of related works. Section 3 introduces the dataset and three deep convolutional neural networks, i.e., AlexNet, GoogLeNet, and ResNet. Section 4 presents the experiments and results in this work. Section 5 concludes the paper.

2. Related Work

The research of agricultural disease identification based on computer vision has been a hot topic. In the early years, the traditional machine learning methods and shallow networks were extensively adopted in the agricultural field.

Sannakki et al. [32] proposed to use k-means based clustering performed on each image pixel to isolate the infected spot. They obtained the result that the Grading System they built by machine vision and fuzzy logic is very useful for grading the plant disease. Samanta et al. [33] proposed a novel histogram based scab diseases detection of potato and applied color image segmentation technique to exact intensity pattern. They got the best classification accuracy of 97.5%. Pedro et al. [34] applied fuzzy decision-making to identify weed shape, with fuzzy multicriteria decision-making strategy; they achieved the best accuracy of 92.9%. Cheng and Matson [35] adopted Decision Tree, Support Vector Machine (SVM), and Neural Network to identify weed and rice; the best accuracy they achieved is 98.2% by using Decision Tree. Sankaran and Ehsani [36] used quadratic discriminant analysis (QDA) and k-nearest neighbour (kNN) to classify citrus leaves infected with canker and Huanglongbing (HLB) from healthy citrus leaves; they got the highest overall accuracy of 99.9% by kNN.

Recently, deep learning methods have been applied in identifying plant disease widely. Cheng et al. [37] used

ResNet and AlexNet to identify agricultural pests. At the same time, they carried out comparative experiments with SVM and BP neural networks; finally, they got the best accuracy of 98.67% by ResNet-101. Ferreira et al. [38] utilized ConvNets to perform weed detection in soybean crop images and classify these weeds among grass and broadleaf. The best accuracy they achieved is 99.5%. Sladojevic et al. [39] built a deep convolutional neural network to automatically classify and detect 15 categories of plant leaf diseases. Meanwhile, their model was able to distinguish plants from their surroundings. They got an average accuracy of 96.3%. Mohanty et al. [40] trained a deep convolutional neural network based on the pretrained AlexNet and GoogLeNet to identify 14 crop species and 26 diseases. They achieved an accuracy of 99.35% on a held-out test set. Sa et al. [41] proposed a novel approach to fruit detection by using deep convolutional neural networks. They adapted Faster Region-based CNN (Faster R-CNN) model, through transfer learning. They got the F1 score with 0.83 in a field farm dataset.

3. Materials and Methods

This paper concentrates on identifying tomato leaf disease by deep learning. In this section, the abstract mathematical model about identifying tomato leaf disease is displayed at first. Meanwhile, the process of typical CNN is described with formulas. Then, the dataset and data augmentation are presented. Finally, we introduced three powerful deep neural networks adopted in this paper, i.e., AlexNet, GoogLeNet, and ResNet.

The main process of tomato leaf disease identification in this work can be abstracted as a mathematical model (see Figure 1). First, we assume the mapping function from tomato leaves to diseases is $f : X \rightarrow Y$ and then send the training samples to the optimization method. The hypothesis set H means possible objective functions with different parameters; through a series of parameters update, we can get the final assumption $g \approx f$.



FIGURE 2: Raw tomato leaf images.

The typical CNN process can be represented with following formulas. Firstly, send the training samples (i.e., training tomato leaf images) to the classifier (i.e., AlexNet, GoogLeNet, and ResNet). Then, convolution operation is carried out; that is, a number of filters slide over the feature map of the previous layer, and the weight matrices do dot product.

$$M_j^l = f \left(\sum_{i \in N_j} M_i^{l-1} * w_j^l + b_j^l \right) \quad (1)$$

where $f(\cdot)$ is activation function, typically a Rectifier Linear Unit (ReLU) [42] function:

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

N_j is the number of kernels of the certain layer, M_i^{l-1} represents the feature map of the previous layer, w_j^l is the weight matrix, and b_j^l is the bias term.

Max-pooling or average pooling is conducted after the convolution operation. Furthermore, the learned features are sent to the fully connected layer. The softmax regression always follows the final fully connected layer, an input x will get the probability of belonging to class i .

$$p(y = i | x; \theta) = \frac{e^{\theta_i^T x}}{\sum_{j=1}^k e^{\theta_j^T x}} \quad (3)$$

where y is the response variable (i.e., predict label), k is the number of categories, and θ is the parameters of our model.

3.1. Raw Dataset. The raw tomato leaf dataset utilized in this work comes from an open access repository of images, which focus on plant health [43]. Health and other 8 diseases categories are included (see Table 1, Figure 2), i.e., early blight (pathogen: *Alternaria solani*) [1], yellow leaf curl disease (pathogen: Tomato Yellow Leaf Curl Virus (TYLCV), Family Geminiviridae, Genus Begomovirus) [2], corynespora leaf spot disease (pathogen: *Corynespora cassiicola*) [3], leaf mold disease (pathogen: *Fulvia fulva*) [4], virus disease (pathogen: Tomato Mosaic Virus) [5], late blight (pathogen: *Phytophthora infestans*) [6], septoria leaf spot (pathogen: *Septoria lycopersici*) [7], and two-spotted spider mite (pathogen: *Tetranychus urticae*) [8]. The total dataset is 5550.

3.2. Data Augmentation. Deep convolutional neural networks contain millions of parameters; thus, massive amounts of data is required. Otherwise, the deep neural network may be overfitting or not robust. The most common method to reduce overfitting on image dataset is to enlarge the dataset manually and conduct label-preserving transformations [21, 44].

In this work, at first, the raw image dataset was divided into 80% training samples and 20% testing samples, and then the data augmentation procedure was conducted: (1) flipping

TABLE 1: The raw tomato leaf dataset.

Label	Category	Number	Leaf symptoms	Illustration
1	Corynespora leaf spot disease	547	Small brown spots appear, leaf spots have yellow halo.	See Figure 1 first row No.1-No.5
2	Early blight	405	Black or brown spots appear, leaf spots often have yellow or green concentric ring pattern.	See Figure 1 first row No.6-No.10
4	Late blight	726	Water-soaked area appears and rapidly enlarges to form purple-brown, oily-appearing blotches.	See Figure 1 second row No.1-No.5
5	Leaf mold disease	480	Irregular yellow or green area appears.	See Figure 1 second row No.6-No.10
6	Septoria leaf spot	734	Round spots, marginal brown, chlorotic yellow, appear.	See Figure 1 third row No.1-No.5
7	Two-spotted spider mite	720	Show white or yellow spots, blade back netting.	See Figure 1 third row No.6-No.10
8	Virus disease	481	Develop yellow or green, slightly shrinking.	See Figure 1 fourth row No.1-No.5
9	Yellow leaf curl disease	814	Develop small and curl upward, crumpling, and marginal yellowing, bushy appearance.	See Figure 1 fourth row No.6-No.10
3	Health	643		See Figure 1 fifth row
Total		5550		

the image from left to right; (2) flipping the image from top to bottom; (3) flipping the image diagonally; (4) adjusting the brightness of image, setting the max delta to 0.4; (5) adjusting the contrast of image, setting the ratio from 0.2 to 1.5; (6) adjusting the hue of image, setting the max delta to 0.5; (7) adjusting the saturation of image, setting the ratio from 0.2 to 1.5; (8) rotating the image by 90° and 270° , respectively. The final dataset is shown in Table 2, and the label in the first row represents the disease categories which are given in Table 1.

3.3. Deep Learning Models

3.3.1. AlexNet. AlexNet is the winner of ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) 2012, a deep convolutional neural network, which has 60 million parameters and 650,000 neurons [21]. The architecture of AlexNet utilized in this paper is displayed in Figure 3. The AlexNet architecture consists of five convolutional layers (i.e., conv1, conv2, and so on), some of which are followed by max-pooling layers (i.e., pool1, pool2, and pool5), three fully connected layers (i.e., fc6, fc7, and fc8), and a linear layer with softmax activation in output. In order to reduce overfitting in the fully connected layers, a regularization method called “dropout” is used (i.e., drop6, drop7) [21]. The ReLU activation function is applied to each of the first seven layers (i.e., relu1, relu2, and so on) [45]. In Figure 3, the notation $m \times m \times n$ in each convolutional layer represents the size of the feature

map for each layer, 4096 represents the number of neurons of the first two fully connected layers. The number of neurons of the final fully connected layer was modified to 9, since the classification problem in this work has 9 categories. In addition, the size of input images must be shaped to 227×227 , which meets the input pixel size requirement of AlexNet.

3.3.2. GoogLeNet. GoogLeNet is an inception architecture [22], which is the winner of ILSVRC 2014 and owns roughly 6.8 million parameters. The architecture of GoogLeNet is presented in Figure 4. The inception module is inspired by the network in network [46] and uses a parallel combination of 1×1 , 3×3 , and 5×5 convolutional layer along with 3×3 max-pooling layer [45]; the 1×1 convolutional layer before 3×3 and 5×5 convolutional layer reduces the spatial dimension and limits the size of GoogLeNet. The whole architecture of GoogLeNet is stacked by inception module on top of each other (See Figure 4), which has nine inception modules, two convolutional layers, four max-pooling layers, one average pooling layer, one fully connected layer, and a linear layer with softmax function in the output. GoogLeNet uses dropout regularization in the fully connected layer and applies the ReLU activation function in all of the convolutional layers [29]. In this work, the last three layers of GoogLeNet were replaced by a fully connected layer, a softmax layer, and a classification layer; the fully connected layer was modified to 9 neurons, which is equal to the

TABLE 2: The final tomato leaf dataset.

Labels	Label1	Label2	Label3	Label4	Label5	Label6	Label7	Label8	Label9	Total
Training set	3933	2916	4626	5229	3456	5283	5184	3465	5859	39951
Testing set	110	81	129	145	96	147	144	161	163	1176

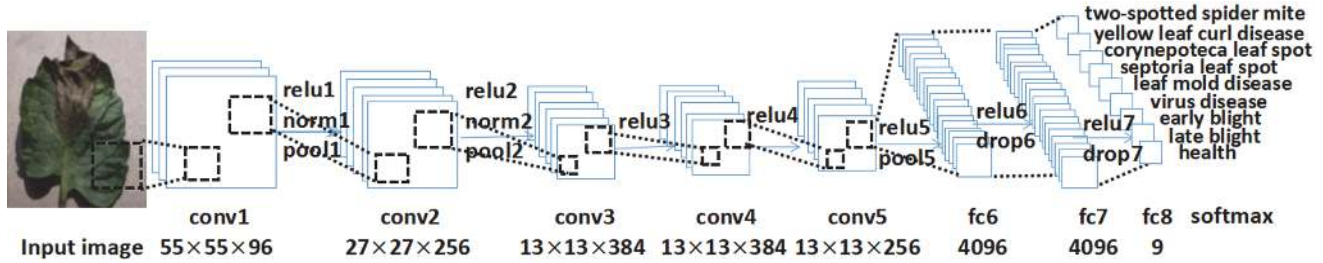


FIGURE 3: The architecture of AlexNet in this work.

categories in the tomato leaf disease identification problem. The size requested of input image of GoogLeNet is 224×224 .

3.3.3. ResNet. The deep residual learning framework is proposed for addressing the degradation problem. ResNet consists of many stacked residual units, which won the first place in ILSVRC 2015 and COCO 2015 classification challenge with error rate of 3.57% [26]. Each unit can be expressed in the following formulas [47]:

$$y_l = h(x_l) + F(x_l, W_l) \quad (4)$$

$$x_{l+1} = f(y_l) \quad (5)$$

where x_l and x_{l+1} are input and output of the l -th unit, and F is a residual function. In [26] $h(x_l) = x_l$ is an identity mapping and f is a ReLU function [42]. A “bottleneck” building block is designed for ResNet (See Figure 5) and comprises two 1×1 convolutions with a 3×3 convolution in between and a direct skip connection bypassing input and output. The 1×1 layers are responsible for changing in dimensions. ResNet model has three types of layers with 50, 101, and 152. For saving computing resources and training time, we choose the ResNet50, which also has high performance. In this work, at first, the last three layers of ResNet were modified by a fully connected layer, a softmax layer, and a classification layer, the fully connected layer was replaced to 9 neurons, which is equal to the categories of the tomato leaf disease. We changed the structure of ResNet subsequently. The size of input image of ResNet should satisfy 224×224 .

4. Experiments and Results

In this section, we reveal the experiments and discuss the experimental results. All the experiments were implemented in Matlab under Windows 10, using the GPU NVIDIA GTX1050 with 4G video memory or NVIDIA GTX1080Ti with 11G video memory. In this paper, overall accuracy was regarded as the evaluation metric in every experiment on

tomato leaf disease detection, which means the percentage of samples that are correctly classified:

$$\text{accuracy} = \frac{\text{true positive} + \text{true negative}}{\text{positive} + \text{negative}} \quad (6)$$

where “true positive” is the number of instances that are positive and classified as positive, “true negative” is the number of instances that are negative and classified as negative, and the denominator represents the total number of samples. In addition, the training time was regarded as an additional performance metric of the network structure experiment.

4.1. Experiments on Optimization Methods. The first experiment is designed for seeking the suitable optimization method between SGD [30] and Adam [30, 31] in identifying tomato leaf diseases, combining with the pretrained network AlexNet, GoogLeNet, and ResNet, respectively. In this experiment, the hyperparameters were set as follows for each network: the batch size was set to 32, the initial learning rate was set to 0.001 and dropped by a factor of 0.5 every 2 epochs, and the max epoch was set to 5; i.e., the number of iterations is 6240. So far as SGD optimization method, the momentum was set to 0.9. For Adam, the gradient decay rate β_1 was set to 0.9, the squared gradient decay rate β_2 was set to 0.999, and the denominator offset ϵ was set to 10^{-8} [31]. The accuracy of different networks is displayed in Table 3. In addition, we choose the better results in each deep neural network to show the training loss against number of iterations during the fine-tuning process (See Figure 6). The words inside parenthesis indicate the corresponding optimization method.

In Table 3, the ResNet with SGD optimization method gets the highest test accuracy 96.51%. In identifying tomato leaf diseases, the performance of Adam optimization method is inferior to the SGD optimization method, especially in combining with AlexNet. In the following paper, AlexNet (SGD), GoogLeNet (SGD), and ResNet (SGD) are referred to as AlexNet, GoogLeNet, and ResNet, respectively.

As it can be seen in Figure 6, the training loss of ResNet drops rapidly in the earlier iterations and tends to stable after 3000 iterations. Consistent with Table 3, the performance of

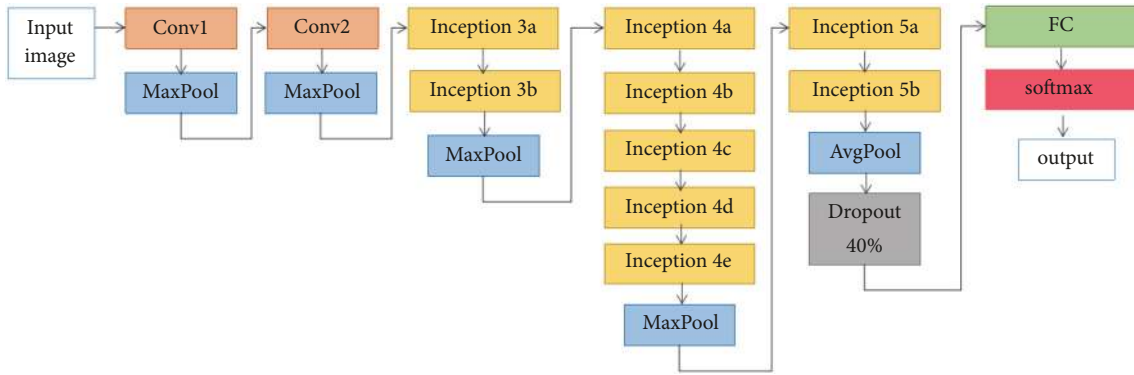


FIGURE 4: The architecture of GoogLeNet [22, 45].

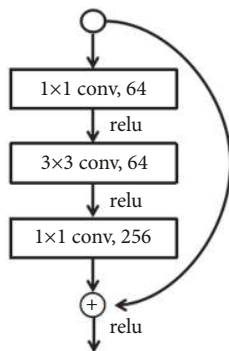


FIGURE 5: ResNet bottleneck residual building block [26].

TABLE 3: Model recognition accuracy.

Model	Accuracy
AlexNet (SGD)	95.83%
AlexNet (Adam)	13.86%
GoogLeNet (SGD)	95.66%
GoogLeNet (Adam)	94.06%
ResNet (SGD)	96.51%
ResNet (Adam)	94.39%

AlexNet and GoogLeNet is similar and both inferior to the ResNet.

4.2. Experiments on Batch Size and Number of Iterations. From the experiment on optimization methods, the ResNet obtains the highest classification accuracy. Next, we evaluated the effects of batch size and the number of iterations on the performance of the ResNet. The batch size was set to 16, 32, and 64, respectively. Meanwhile, the number of iterations was set to 2496, 4992, and 9984. The classification accuracy of different training scenarios is given in Table 4. At the same time, the classification accuracy of each label's representative leaf disease category (See Table 1) is given. In this experiment, the initial learning rate was set to 0.001 and dropped by a factor of 0.5 every 2496 iterations.

In Table 4, the best overall classification accuracy 97.19% is got by the ResNet combining with batch size 16 and

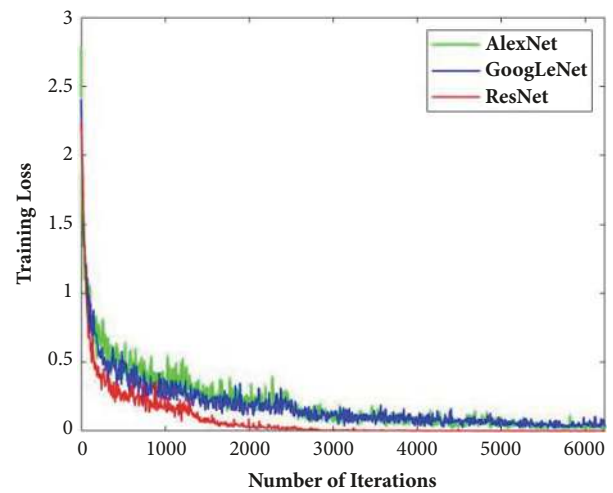


FIGURE 6: The training loss during the fine-tuning process.

iterations 4992. As shown in Table 4, whether increasing the number of iterations or batch size, the performance of corresponding models has not been improved significantly in identifying tomato leaf disease. A small batch size with a medium number of iterations is quite effective in this work. Moreover, a larger batch size and number of iterations increases the training duration. We have not tried higher or lower values for the attempted parameters, since different classification task may have various suitable parameters, and it is hard to give a certain rule in setting hyperparameters.

4.3. Experiments on Full Training and Fine-Tuning of ResNet. This section is designed for exploring the performance of CNN by changing the structure of the models. In practical, a deep CNN always owns a large size which means a large number of parameters. Thus, full training of a deep CNN requires extensive computational resources and is time-consuming. In addition, full training of a deep CNN may led to overfitting when the training data is limited. So we compared the performance of the pretrained CNN through full training and fine-tuning their structures.

We changed the structure of ResNet, and combination of the best parameters from the front experiments was utilized.

TABLE 4: Classification accuracies with different parameters during fine-tuning of the ResNet. The numbers inside parenthesis indicate batch size and number of iterations.

Networks	label1	label2	label3	label4	label5	label6	label7	label8	label9	overall
ResNet (16,2496)	90.91%	88.89%	100%	100%	96.88%	100%	90.28%	88.20%	97.55%	94.98%
ResNet (16,4992)	98.18%	98.77%	100%	98.62%	96.88%	100%	96.53%	88.82%	98.77%	97.19%
ResNet (16,9984)	98.18%	97.53%	100%	98.62%	96.88%	100%	97.22%	86.96%	98.77%	96.94%
ResNet (32,2496)	97.27%	95.06%	100%	97.93%	96.88%	100%	95.14%	86.34%	99.39%	96.34%
ResNet (32,4992)	97.27%	95.06%	100%	97.24%	96.88%	100%	96.53%	86.96%	99.39%	96.51%
ResNet (32,9984)	96.36%	96.30%	100%	99.31%	96.88%	100%	94.44%	88.20%	98.77%	96.60%
ResNet (64,2496)	93.64%	93.83%	100%	97.24%	96.88%	100%	94.44%	87.58%	99.39%	95.92%
ResNet (64,4992)	94.55%	95.29%	100%	96.55%	95.83%	100%	95.83%	86.96%	99.39%	95.83%
ResNet (64,9984)	95.45%	93.83%	100%	97.93%	96.88%	100%	94.44%	87.58%	99.39%	96.17%

ResNet50 has 177 layers if the layers for each building block and connection are calculated. In this experiment, the last three layers of ResNet were modified to a fully connected layer (denoted as “fc”), a softmax layer, and a classification layer, and the fully connected layer owns 9 neurons. The structure was changed by freezing the weights of a certain number of layers in the network by setting the learning rate in those layers to zero. During training, the parameters of the frozen layers are not updated. Full training and fine-tuning are defined by the number of training layers, i.e., full training (1-“fc”), fine-tuning (37-“fc”, 79-“fc”, 111-“fc”, 141-“fc”, 163-“fc”). The accuracy and training time of different network structure are presented in Table 5. At first, the batch size and 4992 iterations were combined, the initial learning rate was set to 0.001 and dropped by a factor of 0.1 every 2496 iterations. In order to get more convincing conclusions, ResNet (16, 9984), which gets the second place in Table 4, was also used to execute the experiments.

In Table 5, the accuracy and training time of different network structures are presented. In two cases, i.e., the 4992 iterations and 9984 iterations of ResNet, the accuracy of the model from the 37 layer fine-tuning structure are higher than that of the full training model. In the case where the number of iterations is 4992, the accuracy of the model from the 79 layer fine-tuning structure is equal to that of the full training model. The final column of the Table 5 represents the training time of the corresponding network, and it is clear that the training time of the fine-tuning models is greatly lowered than the full training model. Because the gradients of the frozen layers do not need to be computed, freezing the weights of initial layers can speed up network training. We observe that the moderate fine-tuning models (37-“fc”, 79-“fc”, 111-“fc”) always led to a performance superior or approximately equal to the full training models. Thus, we suggest that, for practical application, the moderate fine-tuning models may be a good choice. Especially for the researcher who holds massive data, the fine-tuning models may achieve good performance while saving computational resources and time.

Moreover, the features of the final fully connected layer of ResNet (16, 4992, 37-“fc”) were examined by utilizing the t-distributed Stochastic Neighbour Embedding (t-SNE) algorithm (see Figure 7) [48]. 1176 test images were used to

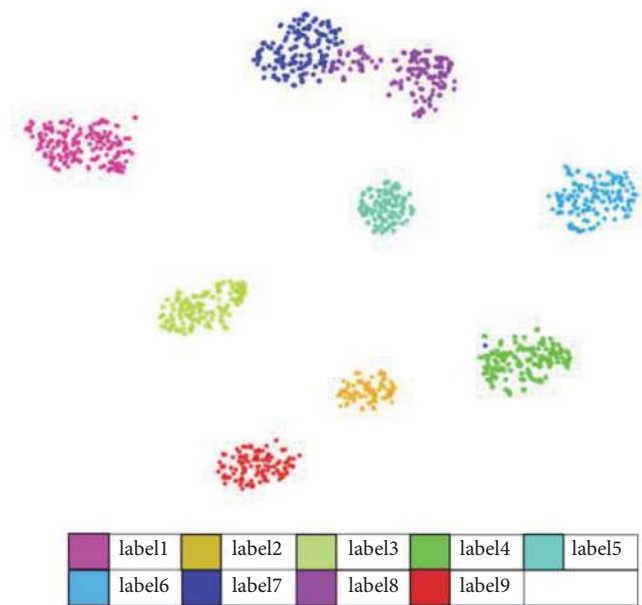


FIGURE 7: Two-dimensional scatter plot of high-dimensional features generated with t-SNE.

extract the features. In Figure 7, different colors represent different labels; the corresponding disease categories of the labels were listed in Table 1. As shown in Figure 7, 9 different color points are clearly separated, which indicates that the features learned from the ResNet with the optimal structure can be used to classify the tomato leaf disease precisely.

5. Conclusion

This paper concentrates on identifying tomato leaf disease using deep convolutional neural networks by transfer learning. The utilized networks are based on the pretrained deep learning models of AlexNet, GoogLeNet, and ResNet. First we compared the relative performance of these networks by using SGD and Adam optimization method, revealing that the ResNet with SGD optimization method obtains the highest result with the best accuracy, 96.51%. Then, the performance evaluation of batch size and number of

TABLE 5: Accuracies and training time in different network structures. The values inside parenthesis denote batch size, number of iterations, and training layers.

Network topology	Accuracy	Time (min:sec)
ResNet (16, 4992, 1-“fc”)	96.43%	59min 30sec
ResNet (16, 4992, 37-“fc”)	97.28%	44min 13sec
ResNet (16, 4992, 79-“fc”)	96.43%	37min 27sec
ResNet (16, 4992, 111-“fc”)	95.75%	30min 6sec
ResNet (16, 4992, 141-“fc”)	95.32%	24min 15sec
ResNet (16, 4992, 163-“fc”)	92.69%	19min 31sec
ResNet (16, 9984, 1-“fc”)	96.94%	118min 32sec
ResNet (16, 9984, 37-“fc”)	97.02%	92min 53sec
ResNet (16, 9984, 79-“fc”)	96.77%	72min 23sec
ResNet (16, 9984, 111-“fc”)	96.26%	58min 40sec
ResNet (16, 9984, 141-“fc”)	95.75%	47min 22sec
ResNet (16, 9984, 163-“fc”)	93.96%	39min 32sec

iterations affecting the transfer learning of the ResNet was conducted. A small batch size of 16 combining a moderate number of iterations of 4992 is the optimal choice in this work. Our findings suggest that, for a particular task, neither large batch size nor large number of iterations may improve the accuracy of the target model. The setting of batch size and number of iterations depends on your data set and the utilized network. Next, the best combined model was used to fine-tune the structure. Fine-tuning ResNet layers from 37 to “fc” obtained the highest accuracy 97.28% in identifying tomato leaf disease. Based on the amount of available data, layer-wise fine-tuning may provide a practical way to achieve the best performance of the application at hand. We believe that the results obtained in this work will bring some inspiration to other similar visual recognition problems, and the practical study of this work can be easily extended to other plant leaf disease identification problems.

Data Availability

The tomato leaf data supporting this work are from previously reported studies, which have been cited. The processed data are available from the corresponding author request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

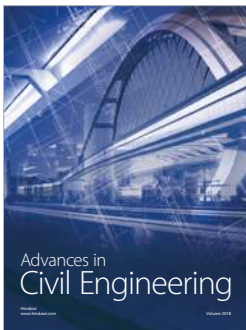
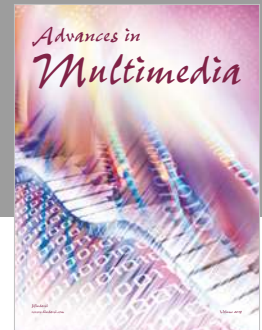
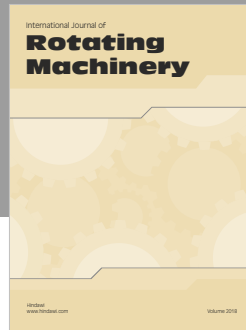
This study was supported by the National Science and technology support program (2014BAD12B01-1-3), Public Welfare Industry (Agriculture) Research Projects Level-2 (201503116-04-06), Postdoctoral Foundation of Heilongjiang Province (LBHZ15020), Harbin Applied Technology Research and Development Program (2017RAQXJ096), and Economic Decision Making and Early Warning of Soybean Industry in Technology Collaborative Innovation System of Soybean Industry in Heilongjiang Province (20170401).

References

- [1] R. Chaerani and R. E. Voorrips, “Tomato early blight (*Alternaria solani*): The pathogen, genetics, and breeding for resistance,” *Journal of General Plant Pathology*, vol. 72, no. 6, pp. 335–347, 2006.
- [2] A. M. Dickey, L. S. Osborne, and C. L. Mckenzie, “Papaya (*Carica papaya*, Brassicales: Caricaceae) is not a host plant of tomato yellow leaf curl virus (TYLCV; family Geminiviridae, genus Begomovirus),” *Florida Entomologist*, vol. 95, no. 1, pp. 211–213, 2012.
- [3] G. Wei, L. Baoju, S. Yanxia, and X. Xuewen, “Studies on pathogenicity differentiation of corynespora cassiicola isolates, against cucumber, tomato and eggplant,” *Acta Horticulturae Sinica*, vol. 38, no. 3, pp. 465–470, 2011.
- [4] P. Lindhout, W. Korta, M. Cislík, I. Vos, and T. Gerlagh, “Further identification of races of *Cladosporium fulvum* (Fulvia fulva) on tomato originating from the Netherlands France and Poland,” *Netherlands Journal of Plant Pathology*, vol. 95, no. 3, pp. 143–148, 1989.
- [5] K. Kubota, S. Tsuda, A. Tamai, and T. Meshi, “Tomato mosaic virus replication protein suppresses virus-targeted posttranscriptional gene silencing,” *Journal of Virology*, vol. 77, no. 20, pp. 11016–11026, 2003.
- [6] M. Tian, B. Benedetti, and S. Kamoun, “A second Kazal-like protease inhibitor from *Phytophthora infestans* inhibits and interacts with the apoplastic pathogenesis-related protease P69B of tomato,” *Plant Physiology*, vol. 138, no. 3, pp. 1785–1793, 2005.
- [7] L. E. Blum, “Reduction of incidence and severity of *Septoria lycopersici* leaf spot of tomato with bacteria and yeasts,” *Ciência Rural*, vol. 30, no. 5, pp. 761–765, 2000.
- [8] E. A. Chatzivasilieiadis and M. W. Sabelis, “Toxicity of methyl ketones from tomato trichomes to *Tetranychus urticae* Koch,” *Experimental and Applied Acarology*, vol. 21, no. 6-7, pp. 473–484, 1997.
- [9] M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe, and S. Mougiakakou, “Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network,” *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1207–1216, 2016.
- [10] Y. Tang and X. Wu, “Text-independent writer identification via CNN features and joint Bayesian,” in *Proceedings of the 15th*

- International Conference on Frontiers in Handwriting Recognition, ICFHR 2016*, pp. 566–571, Shenzhen, China, October 2016.
- [11] Y. Tang and X. Wu, “Saliency Detection via Combining Region-Level and Pixel-Level Predictions with CNNs,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 1608–1618, 2016.
 - [12] Y. Tang and X. Wu, “Salient object detection with chained multi-scale fully convolutional network,” *ACM Multimedia (ACMMM)*, pp. 618–626, 2017.
 - [13] Y. Tang and X. Wu, “Scene text detection and segmentation based on cascaded convolution neural networks,” *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1509–1520, 2017.
 - [14] Y. Tang and X. Wu, “Scene Text Detection using Superpixel based Stroke Feature Transform and Deep Learning based Region Classification,” *IEEE Transactions on Multimedia*, vol. 20, no. 9, pp. 2276–2288, 2018.
 - [15] Y. Yao, X. Wu, Z. Lei, S. Shan, and W. Zuo, “Joint Representation and Truncated Inference Learning for Correlation Filter based Tracking,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 1–14, 2018.
 - [16] L. Zhang, F. Yang, Y. Daniel Zhang, and Y. J. Zhu, “Road crack detection using deep convolutional neural network,” in *Proceedings of the 23rd IEEE International Conference on Image Processing, ICIP 2016*, pp. 3708–3712, Phoenix, AZ, USA, September 2016.
 - [17] D. Xie, L. Zhang, and L. Bai, “Deep learning in visual computing and signal processing,” *Applied Computational Intelligence and Soft Computing*, vol. 2017, Article ID 1320780, 13 pages, 2017.
 - [18] Z. Zhou, J. Shin, L. Zhang, S. Gurudu, M. Gotway, and J. Liang, “Fine-tuning convolutional neural networks for biomedical image analysis: Actively and incrementally,” in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 4761–4772, USA, July 2017.
 - [19] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 3730–3738, Santiago, Chile, 2015.
 - [20] W. Ouyang and X. Wang, “Joint deep learning for pedestrian detection,” in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 2056–2063, Sydney, Australia, December 2013.
 - [21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, Lake Tahoe, Nev, USA, December 2012.
 - [22] C. Szegedy, W. Liu, Y. Jia et al., “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 1–9, IEEE, Boston, Mass, USA, June 2015.
 - [23] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” <https://arxiv.org/abs/1409.1556>, 2015.
 - [24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” <https://arxiv.org/abs/1512.00567>, 2015.
 - [25] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-ResNet and the impact of residual connections on learning,” <https://arxiv.org/abs/1602.07261>, 2016.
 - [26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 770–778, July 2016.
 - [27] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, “Densely connected convolutional networks,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2016.
 - [28] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
 - [29] M. Mehdipour Ghazi, B. Yanikoglu, and E. Aptoula, “Plant identification using deep neural networks via optimization of transfer learning parameters,” *Neurocomputing*, vol. 235, pp. 228–235, 2017.
 - [30] S. Ruder, “An overview of gradient descent optimization algorithms,” <https://arxiv.org/abs/1609.04747>, 2017.
 - [31] D. P. Kingma and J. L. Ba, “Adam: a method for stochastic optimization,” <https://arxiv.org/abs/1412.6980>, 2017.
 - [32] S. S. Sannakki, V. S. Rajpurohit, V. B. Nargund, R. Arun Kumar, and S. Prema Yallur, “Leaf disease grading by machine vision and fuzzy logic,” *International Journal of Computer Technology and Applications*, vol. 2, no. 5, pp. 1709–1716, 2011.
 - [33] D. Samanta, P. P. Chaudhury, and A. Ghosh, “Scab diseases detection of potato using image processing,” *International Journal of Computer Trends and Technology*, vol. 3, pp. 109–113, 2012.
 - [34] P. J. Herrera, J. Dorado, and Á. Ribeiro, “A novel approach for weed type classification based on shape descriptors and a fuzzy decision-making method,” *Sensors*, vol. 14, no. 8, pp. 15304–15324, 2014.
 - [35] B. Cheng and E. T. Matson, “A feature-based machine learning agent for automatic rice and weed discrimination,” *International Conference on Artificial Intelligence and Soft Computing*, pp. 517–527, 2015.
 - [36] S. Sankaran and R. Ehsani, “Comparison of visible-near infrared and mid-infrared spectroscopy for classification of Huanglongbing and citrus canker infected leaves,” *Agricultural Engineering International: CIGR Journal*, vol. 15, no. 3, pp. 75–79, 2013.
 - [37] X. Cheng, Y. Zhang, Y. Chen, Y. Wu, and Y. Yue, “Pest identification via deep residual learning in complex background,” *Computers and Electronics in Agriculture*, vol. 141, pp. 351–356, 2017.
 - [38] A. dos Santos Ferreira, D. Matte Freitas, G. Gonçalves da Silva, H. Pistori, and M. Theophilo Folhes, “Weed detection in soybean crops using ConvNets,” *Computers and Electronics in Agriculture*, vol. 143, pp. 314–324, 2017.
 - [39] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, “Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification,” *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 3289801, 11 pages, 2016.
 - [40] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” *Frontiers in Plant Science*, vol. 7, article no. 1419, 2016.
 - [41] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, “Deepfruits: A fruit detection system using deep neural networks,” *Sensors*, vol. 16, article no. 1222, no. 8, 2016.
 - [42] V. Nair and G. E. Hinton, “Rectified linear units improve Restricted Boltzmann machines,” in *Proceedings of the 27th International Conference on Machine Learning (ICML '10)*, pp. 807–814, Haifa, Israel, June 2010.

- [43] D. P. Hughes and M. Salathe, “An open access repository of images on plant health to enable the development of mobile disease diagnostics,” <https://arxiv.org/abs/1511.08060>, 2016.
- [44] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, “A committee of neural networks for traffic sign classification,” in *Proceedings of the 2011 International Joint Conference on Neural Networks (IJCNN 2011 - San Jose)*, San Jose, CA, USA, July 2011.
- [45] P. Pawara, E. Okafor, O. Surinta, L. Schomaker, and M. Wiering, “Comparing Local Descriptors and Bags of Visual Words to Deep Convolutional Neural Networks for Plant Recognition,” in *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods*, pp. 479–486, Porto, Portugal, February 2017.
- [46] M. Lin, “Network in Nnetwork,” <https://arxiv.org/abs/1312.4400>, 2014.
- [47] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *Proceedings of the European Conference on Computer Vision*, pp. 630–645, 2016.
- [48] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2625, 2008.



Hindawi

Submit your manuscripts at
www.hindawi.com

