Fruit Leaf Pests and Diseases Identification Based on Data Enhancement and Transfer Learning

Yun Deng*
College of Information Science and Engineering
Guilin University of Technology
13978321602, Guilin, Guangxi, 541004
woshidengyun@sina.com

Zhaojun Chen
College of Information Science and Engineering
Guilin University of Technology
17809163959, Guilin, Guangxi, 541004
1120304227@qq.com

Shouxue Chen
College of Information Science and Engineering
Guilin University of Technology
13978326212, Guilin, Guangxi, 541004
7666817@qq.com

ABSTRACT
There are many types of fruit leaf diseases and insect pests, and the pathology is complicated. It is difficult to rely on manual identification alone, and the error rate is high. Aiming at the most common fruit leaf diseases and insect pests in China, this article collected a large number of harmful and healthy fruit leaf surface pictures. A new fruit disease and insect pest library was constructed using data enhancement techniques such as blurring, drying, sharpening, and rotation. Subsequently, transfer learning was introduced to migrate the Inception V3 model to its own fruit leaf data set. A new fully connected layer and an output layer are defined, and the model is trained through parameter adjustment and iteration. The experimental results show that the accuracy rate of fruit leaf pest and disease recognition model obtained by transfer learning can reach 97.4%, which can improve by up to 35.1 percentage points compared with the model based on LeNet-5. In the end, the model uses a leaflet brown grape When the blade was tested, the model was successfully identified with a probability of 99%, which proved that the method has high accuracy and generalization.

CCS Concepts
•Computing methodologies →Artificial intelligence →Computer vision→Computer vision problems →Object identification

Keywords
Pest and Disease Identification; Data Enhancement; Migration Learning; Confusion Matrix

*Corresponding author

ICIIP'2019, November 16–17, 2019, China, China

ACM ISBN 978-1-4503-6191-0/19/11…$15.00
https://doi.org/10.1145/3378065.3378146

1. INTRODUCTION
China has a vast territory and a wide variety of fruits. It is a country with abundant fruit production. Fruit leaf pests and diseases are numerous and pathologically complex, which seriously affects the fruit yield and quality and causes economic damage to fruit farmers[1]. However, it is difficult to rely solely on agricultural experts to manually identify diseases and insect pests, with high error rate and high cost. In recent years, many researchers at home and abroad have done a lot of research on the application of image processing technology based on machine vision in the detection of fruit plant leaves diseases and insect pests, and made great progress[2-3]. It is difficult to extract some key features when using image processing technology to detect diseases and insect pests on fruit leaves, which will affect the accuracy of detection of diseases and insect pests.

Convolutional neural network (CNN) learns directly from the features of the data itself, and has a strong ability to express features. CNN has been successfully applied to research fields such as handwritten digit recognition, face recognition, etc., and some progress has been made in the recognition and classification of crops, but there are few studies on its pest and disease in fruit leaves. In this study, the diseases and insect pests of apple, cherry and grape were studied[4], and a large number of pictures of leaf and insect pests were collected. Through experimental data enhancement technology and transfer learning, a more stable model was proposed to solve the detection problem of fruit leaf diseases and insect pests, which provided valuable experience for later transplantation of the model into Android and agricultural production.

2. NETWORK MODEL ESTABLISHMENT
Generally speaking, transfer learning is to use existing knowledge to learn new knowledge. The core is to find the similarity between existing knowledge and new knowledge[5-6]. Because data labels are difficult to obtain, building models from scratch is complex and time-consuming. In transfer learning, our existing knowledge is called the source domain, and the new knowledge to be learned is called the target domain. The source and target domains are different but related to each other. We need to reduce the source and target domains. The distribution of domains is different, and knowledge is transferred to achieve data calibration.

For machine learning models, rapid construction and strong generalization are required. For data, most data have no labels. But o collecting labeled data and building a model from scratch are costly, and you need to reuse the model and labeled data[7]. Traditional machine learning: Assume that the data follow the same distribution, but we want to quickly build models for data from different distributions and implement data labeling.

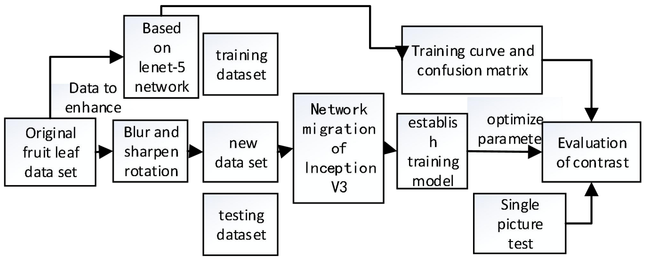

Figure 1. Network Model Design Flow Chart

This paper establishes a network model of fruit leaf pest and disease recognition based on data augmentation and transfer learning (Figure 1). First, pictures of fruit diseases and insect pests were obtained from the Internet. Due to limited pictures and in order to enhance the generalization performance of the net-work model, data augmentation technology was used to expand the picture library. The original fruit leaf data set is subjected to operations such as blurring, drying, sharpening, flipping, shearing, etc. to expand into a new data set, and the data set is divided into a training set and a test set, and the model and the evaluation model are trained respectively. With the inceptionv3 network as the migration model, the last layer of the classification layer is removed, a layer of full connectivity is redefined, and the bottleneck layer is used as the output of the fully connected layer to establish a new network structure. Finally, in the experiment, the performance of the analysis model is evaluated according to the training set's challenge rate and confusion matrix, and a simple comparison is made with other models.

3. BASED ON DATA ENHANCEMENT BLADE PEST IDENTIFICATION AND TRANSFER LEARNING OF NETWORK ALGORITHM DE-SIGN

3.1 Data Enhancement

This paper collected pictures of leaf diseases and insect pests of apple, cherry and grape. They were healthy apple, apple scab, gray leaf spot of apple, cedar apple rust, healthy cherries, cherry powdery mil-dew, Jian Kang grapes, grape black rot, leaf spot round grapes, grape leaf spot a total of ten categories of data, each class 1 00 different pictures. Some of the data are shown in Table 1 below:

Table 1. Part of The Pest and Disease Leaf Image

fruit leaf					
disease	Apple Scab	Apple gray spot	Cherry Powdery Mildew	Grape Black Measles Fungus	Grape Leaf Blight Fungus

Data enhancement is generally divided into two ways: one is offline enhancement, and the other is online enhancement. The so-called offline enhancement is to directly process the data set, and the data volume increases geometrically. One type is online enhancement: This enhanced method is used to obtain the batch data and then enhance the batch data, such as rotation, translation, fold and other corresponding changes. Because some data sets cannot accept linear level growth This method is often used for large data sets. This article takes the offline enhancement method. The general offline enhancement includes blurring, sharpening, drying, flipping, cropping and other operations.

3.1.1 Blurring Operation

Blur operation is one of the most common and simple operations in image expansion, and the reason for using it is that it reduces image noise. The reason for reducing the noise is to make the picture features more easily captured by the convolution kernel. The general fuzzy operation calculation formula is as follows:

$$g(i,j)=\sum_{k,l} f(i+k,j+l)*h(k,l) \tag{1}$$

Among them, the $h(k,l)$ is filter coefficient can also be simply written as $g=f\otimes h$

Gaussian blur, in fact, it is a data smoothing technique. It is understood that each pixel takes the average value of surrounding pixels. It is a kind of blurring effect in the form of image expression, which erases many details. Therefore, the larger the average value of the pixels, the stronger the blurring effect. The Gaussian fuzzy algorithm uses the normal distribution function as a weight function, that is, the closer the center point is, the larger the weight is, the farther from the center, the smaller the weight. The one-dimensional form of the gaussian is shown in formula (2), where μ is the mean of x and σ is the variance of x .Because when you calculate the average, the center is the origin, so μ is equal to 0.

$$f(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-(x-u)^2/2\sigma^2} \tag{2}$$

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{(x^2+y^2)}{2\sigma^2}} \tag{3}$$

According to the one-dimensional gaussian function, a two-dimensional gaussian function can be de-rived (formula 3).

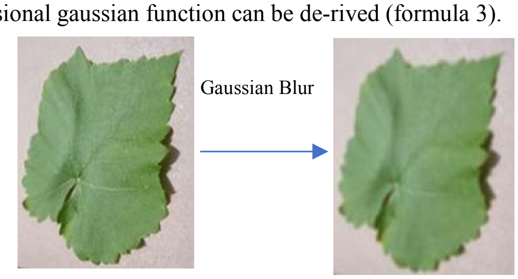

Figure 2. Gaussian Blurred Grape Leaves

Uniform blur, also called uniform smoothing, is a linear smoothing method, as shown in Figure 2 Grape leaves after uniform blur. Uniform blur is based on the current pixel point, and takes the average value of all pixel values in a window of size $m*n$ as the output.

$$f(x,y)=\frac{1}{mn}\sum_{(s,t)\in s_{xy}} g(s,t) \tag{4}$$

After several iterations of this process, the effect of image smoothing is enhanced, while eliminating image noise, while maintaining the edge sharpness calculation formula (4).

3.1.2 Sharpening Operation

The purpose of image sharpening in this paper is to enhance the image edge. Although sharpening operation has no impact on the noise of the image, it makes the edge contour of the image obvious and the image color prominent, improves the quality of the image, produces more easily recognized images, and lays a foundation for the subsequent convolution operation.

General gradient sharpening is to sharpen the image using a differential algorithm. Using the inverse operation of the integral, the differential operation can sharpen the edge contours in the image. In the operation of the image, the first order differential is expressed by the gradient method. For an image represented $f(x,f)$ by a function, the gradient defined at the point (x,f) is a vector, defined as:

$$G[f(x,y)] = [\frac{\partial f}{\partial x}\frac{\partial f}{\partial y}] \tag{5}$$

For digital images, differentials $\partial f / \partial x$ and $\partial f / \partial y$ can be approximated by difference, and the ap-proximate gradient expression is:

$$G[f(x,y)] = \sqrt{[f(x,y)-f(x+1,y)]^2 + [f(x,y)-f(x,y+1)]^2} \tag{6}$$

When the gradient of a pixel cannot be calculated in the last row or the last column of the image, it is generally replaced by the gradient value of the previous row or the previous column.

3.1.3 Adding Noise

In layman's terms, noise is all kinds of interference. Because the reality picture is a process with a lot of noise, adding noise to the data set can be closer to the real image recognition process, which can effectively suppress the over-fitting problem. Probability density function of gaussian noise (formula 7)

$$g(t) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(t-E)^2}{2\sigma^2}} \tag{7}$$

E is the sample mean. σ is the standard deviation of t . In this paper, gaussian noise with a mean of 0 and a standard deviation of 0.05 is added to the image.

3.1.4 Rotation

Taking the center of the image as the origin and rotating at any angle θ , that is, rotating all pixels on the image by the same angle. The rotation transformation of an image can also be represented by a matrix transformation.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \tag{8}$$

Set point $F_0(x_0, y_0)$ to rotate angle θ anticlockwise and the corresponding point is $F(x,y)$ .Then, the coordinates of points $F_0(x_0, y_0)$ and $F(x,y)$ before and after rotation are respectively as shown in formula (8).

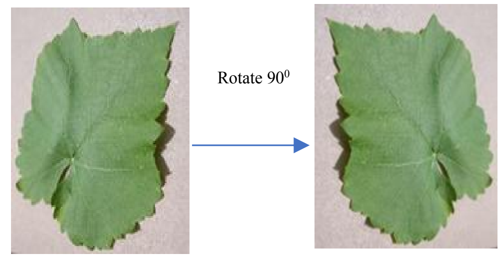

Figure 3. Rotate the grape leaves 90 degrees

3.2 Based Inception V3 of Migration Network Model Design

Migration learning extracts and migrates knowledge from existing data to complete new learning tasks. Specifically[8-9], it can be formally defined as: source domain D_s , source task T_s , target domain D_t , domain target task T_t . Domain D is defined as a binary pair $\{x, p(x)\}$, where x is the feature space, $p(x)$ is the edge distribution of x , $x = \{x_1, x_2, \ldots x_n\}$ .Task T is also a binary pair $\{y, f(x)\}$, y is the label space, and $y = f(x)$ is the objective function learned from the training samples $\{x_x, y_i\}$ .The task of this paper is to migrate from the source domain ImageNet big data set to the target domain fruit leaf.

Before the emergence of inception network, in order to better extract and fine features, people only considered to make the network deeper and wider, but this also brought many problems. The author of perception network directly starts from the network structure, changes the convolution layer and full connection layer into sparse connection, and finds the approximate optimal sparse matrix. The final result is that perception can not only keep the sparsity of network structure, reduce the computational complexity, but also improve the training efficiency by using the high computational performance of dense matrix[10]. One of the most important improvements of perception V3 is factorization, which decomposes 7x7 into two one-dimensional convolutions (1x7,7x1). 3x3 is the same (1x3,3x1). This advantage can not only accelerate the calculation (redundant computing power can be used to deepen the network), but also decompose one conv into two conv, further increasing the network depth and increasing the network nonlinearity. The schematic diagram of perception-v3 model is as follows:

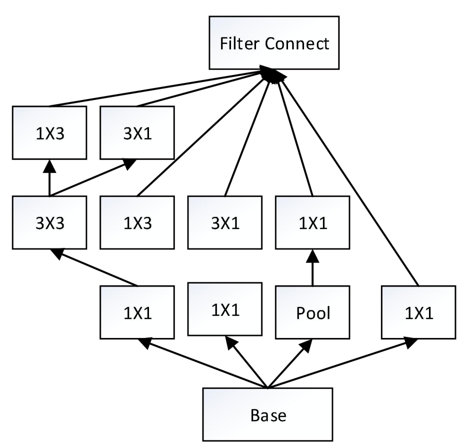

Figure 4. Inception-v3 convolution structure

Article selecting Inception V3 after algorithm steps are primarily as follows: (1) to the data enhancement fruit blade data set as input (2) so that with Inception-v3 network as a transport model , which removed the last layer of the classification, with the bottleneck layer of The output as the extracted feature (3) uses the bottleneck layer as the output of the fully connected layer to and from the layer as the output of the classification .

$$P(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{k} e^{zk}} \tag{9}$$

This article is a 10-class recognition problem, so K = 10 in formula (9). The features of the first few layers of the Inception-v3 network are migrated to the fruit leaf domain (Figure 5).

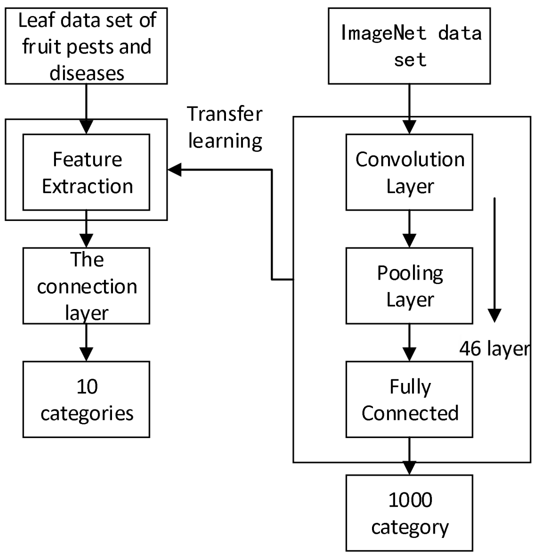

Figure 5. migration model flowchart

The globality of the deep features of the migration may lead to poor migration performance of the new model, and the fruit leaf image information is captured by designing fully connected layers (Figure 7).

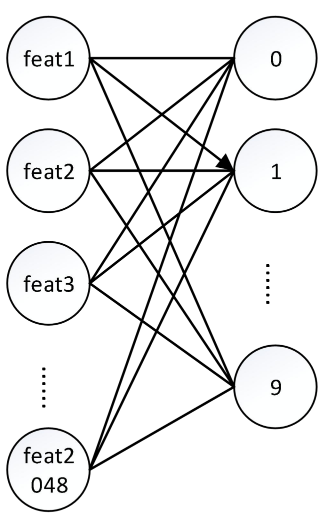

Figure 6. full connection

The last full connection layer F_{X1} of the new network consists of 1024 neurons, the output layer F_{X2} is the classification layer, the number of categories of the classification task of the output target domain is 10, the output vector X_i of the upper layer is input to the F_{X1} layer, and the feature vector T_{X1} is input to the F_{X2} layer, and finally the classification output results are obtained. The calculation formulas of the two network layers are as follows:

$$T_{X1} = f(X_I * K_{FX1} + b_{FX1}) \tag{10}$$

$$T_{X2} = \sigma(T_{X1}) \tag{11}$$

In formula (6) (7), activation function σ is R e lu function; f is $soft\max$ probability function $K_{F_{X1}}$ and $b_{F_{X1}}$ are convolution kernel parameters and offsets of formula (10) (11), respectively. The reason why $relu$ is chosen as the activation function is that it has ideal properties[11]. They do not need input normalization to prevent them from saturating. If at least some training examples provide positive input value for $relu$, then learning will happen in that neuron. If formula (12) is used, it is helpful for local generalization. $X_{a,b}^{i}$ is the activation value calculated by kernel i at position (a,b) , and then the normalized activation value $Y_{a,b}^{i}$ of the response is obtained by nonlinear change of $relu$:

$$Y_{a,b}^{i} = X_{a,b}^{i} \Big/ \left(\lambda \sum_{j=\max(0,i-\frac{n}{2})}^{\min(N-1,i+\frac{n}{2})} (X_{a,b}^{j})^2 + c\right)^{\alpha} \tag{12}$$

N is the mapping of "adjacent" cores in the same location, and N is the number of cores in that layer. The sorting of kernel mapping is arbitrary and can be determined before training. This response normalization reflects the lateral inhibition[12] of neurons stimulated by some stimuli in real life, which results in neurons output many different activation values and use different cores. Constants λ , n , α , and c are super parameters whose values are determined by the validation set.

In this paper, it is assumed that there are m samples in the target domain training set, a single input sample is (x^i, y^i) . x^i is n-dimensional vector, y^i is the real label of the sample, and the overall cost function of the network model is:

$$J(\mathrm{w},b) = [\frac{1}{m}\sum_{i=1}^{m} J(\mathrm{w},b,x^i,y^i)] + \frac{\lambda}{2}\sum_{l-1}^{n_{l-1}}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{i+1}}(w_{ij}{}^{l})^2 \tag{13}$$

Among them, λ is the weight attenuation coefficient, n_l is the total number of layers in the network, s_l is the number of nodes in the l layer of the network, and $h_{w,b}(x^i)$ is the output value obtained when the network is propagating forward. The batch gradient descent method is used to adjust the parameters $w_{i,j}^{l+1}$, and the parameters $w_{i,j}^{l+1}$ and $b_{i,j}^{l+1}$ of each layer are updated.

In this paper, stochastic gradient descent is used to train the parameters[13]. The initial training batch sample is set to 30 and adjusted appropriately according to the training effect. Let the increment be Δ and the weight decay factor be δ, where the weight decay factor δ is conducive to regularization and reduces model errors. The update rule of weight w is formula (9) (8):

$$v_{i+1} = \Delta v_i - \varepsilon\left(\left\langle \frac{\partial L}{\partial W}\Big|_{w_i} \right\rangle_{D_i} + \delta w_i\right) \qquad (14)$$

$$w_{i+1} = w_i + v_{i+1} \qquad (15)$$

In the above formula, i is the number of iterations, v is the increment, and ε is the learning rate. After multiple iterations, the training on the target data set is completed when the cost function value is the smallest.

4. EXPERIMENT

4.1 Experimental Environment

The experimental environment of this paper is desktop computer, the processor is Intel (R) Xeon (R) CPU e5-1620 @ 3.50ghz, windows 10 system, 8g memory. The programming tool is pycham2019, using Python language, python version is 3.7.4, and anaconda3 + tensorflow-cpu1.14.01 is configured

Table 2. Basic parameter settings of the network

Training parameter	parameter settings
Learning rate	0. 1
The maximum number of iterations	3000
Activation function	*relu*
optimization	Stochastic gradient descent

4.2 Results and Discussion

This paper begins in the original data set trained based LeNet-5 model, since a simple network coupled with less data than the last recognition accuracy in the training set only 82 % or so (Fig.7, len_tr_acc), is not very high. The recognition accuracy of the verification set is only about 75 % (Fig.7, len_va_acc). It is obvious that the accuracy of the new model is higher than that of the original model.

Then this paper uses data enhancement and migration learning to build a migration network model based on Inception V3. After data initialization, parameter adjustment and training model, we get the convergence results of the accuracy iteration on the training set and the verification set (Figure 7). The curves "new_tr_acc" and "new_va_acc" represent the rate of achievement in the training set and the verification set, respectively. When the number of iterations reaches 2000 when the second accuracy rate converge, the value is no longer a big change. It can be clearly seen that the accuracy of the new model is higher than the original model.

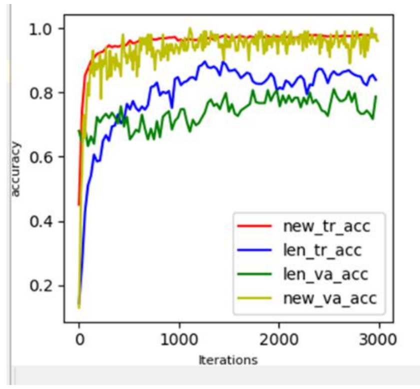

Figure 7. Comparison of the new model with the LeNet-5 based model

According to the confusion matrix (Figure 8, Figure 9), this method of transfer learning is analyzed. In the confusion matrix, blue represents the recognition accuracy. The darker the color, the more accurate the recognition. The x-axis represents the test data set, the y-axis represents the training data set, and 0-9 represents 10 fruit leaf categories. Obviously, the accuracy of Figure 8 is higher than that of Figure 9, and most of the pictures are correctly recognized. Although there are some image recognition errors, the reason is that there are similarities between different leaves, for example, the texture and shape are similar, the comprehensive analysis results are still relatively ideal.

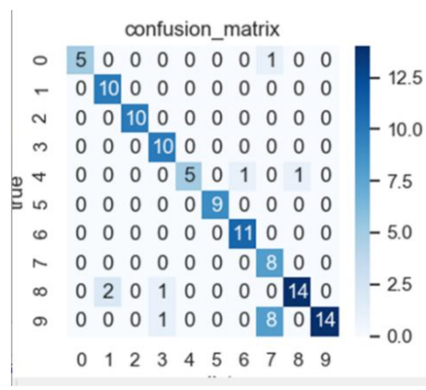

Figure 8. Confusion matrix of the new model

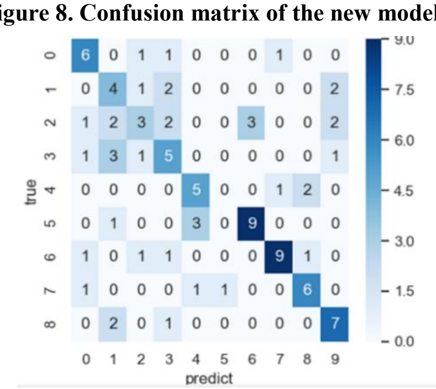

Figure 9. Confusion matrix based on LeNet-5 model

Using a single grapevine melasma test (Figure 11), the network shows in probability which of the 10 categories the images may belong to. It is obvious that the ninth category, vitiligo, is the highest, so the prediction of this model is also accurate.

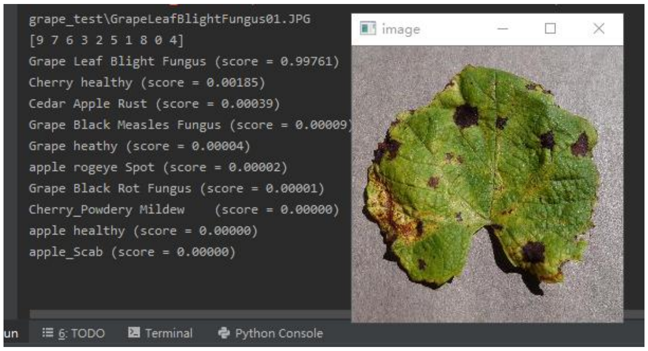

Figure 11. Single picture test

At the end of the paper, a simple comparison is made between the various models of the experiment (Table3 network model comparison). Obviously, the network constructed in this paper is better than the other two network models. Although the accuracy of the network model of vgg16 migration learning is also good, it takes a little longer. However, the network model without migration learning has been over fitted.

Table 3. Comparison of various network models

Network	Based on LeNet-5	Migration based on Inception V3	Migration based on Vgg16
Number of iterations	3000	3000	3000
raining set correct rate	82%	96.1%	92.5%
Test set correct rate	60%	94.2%	91.04%
Training time	1745.34s	786.45s	1056.86 s

5. CONCLUSIONS

The important advantage of migration learning is that it can make good use of the successful feature expression obtained from the existing large-scale data set training, at the same time, combined with its own special data set properties for effective retraining, which not only effectively saves time, but also maintains a relatively high accuracy. In this paper, a small network model which is fully trained on a large-scale data set is introduced into the leaves of fruit diseases and insect pests after data expansion, and a method of vegetable identification based on small samples is proposed. Experiments show that the model trained by this method has high accuracy and generalization, and has great advantages over other networks. The data set constructed by the experiment needs to be optimized and strengthened. Although more than 10000 pictures have been collected, the representativeness and universality of the data set are still insufficient. In the next step, we will continue to consider further increasing the data set to improve the generalization ability of the model. It will lay the foundation for transplanting the model to Android and playing a role in the agricultural greenhouse in the future.

6. REFERENCES

[1] Johnson R, Zhang T. Effective use of word order for text categorization with convolutional neural networks[J]. arXiv preprint arXiv:1412.1058, 2014.

[2] Zhou Jinfeng, ye Shiren, Wang Hui. Text emotion classification based on deep convolution neural network model [J]. Computer Engineering, 2019,45 (3): 300-308

[3] Liao S, Wang J, Yu R, et al. CNN for situations understanding based on sentiment analysis of twitter data[J]. Procedia computer science, 2017, 111(1): 376-381.

[4] Gambäck B, Sikdar U K. Using convolutional neural networks to classify hate-speech[C]//Proceedings of the first workshop on abusive language online. 2017: 85-90.

[5] Wang Jili, Peng Dunlu, Chen Zhang, et al. Am-cnn: a text classification model of convolutional neural network based on attention [J]. Minicomputer system, 2019, 40 (4): 710-714

[6] Zheng yuanpan, Li Guangyang, Li Ye. Review of the application of deep learning in image recognition [J]. Computer engineering and application, 2019,55 (12): 20-36

[7] Wang Liwei, Li Jiming, Zhou Guomin, et al. Application of deep transfer learning in hyperspectral image classification [J]. Computer engineering and application, 2019, 55 (5): 181-186

[8] Uguroglu S, Carbonel [J]. Feature selection for transfer learning[C]//Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Berlin, Heidelberg, 2011: 430-442.

[9] Zhuang Fuzhen, Luo Ping, He Qing, et al. Research progress on transfer learning [J]. Journal of Software, 2015, 26 (1): 26-39.

[10] Duan Meng, Wang Gongpeng, Niu Changyong. Small sample image recognition method based on convolutional neural network [J]. Computer Engineering and Design, 2018, 39 (1): 224-229.

[11] Zeng Weiliang, Lin Zhixian, Chen Yongsa. Research on Fruit and Vegetable Image Recognition in Smart Refrigerators Based on Convolutional Neural Networks [J]. Microcomputer & Application, 2017, 36 (8): 56-59.

[12] Oquab M, Bottou L, Laptev I, et al. Learning and transferring mid-level image representations using convolutional neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition, 2014: 1717-1724.

[13] Zhang Zhi. A transfer learning method based on deep neural network [D]. Harbin Institute of technology, 2018.