

# Transfer Learning-Based Convolutional Neural Network Image Recognition Method for Plant Leaves

Yue Zhao, Yili Zheng, Honglei Shi and Lu Zhang

**Abstract**—To improve the accuracy of plant leaf image recognition with a small dataset of plant leaves, a convolution neural network (CNN) plant leaf image recognition method based on transfer learning is proposed. First, a plant leaf image database was expanded by pre-processing the original plant leaf images through random horizontal and vertical rotation and random zooming. The expanded dataset was then processed by mean removal and divided into training and testing sets at a ratio of 4:1. Second, transfer learning training was performed on the plant leaf dataset using existing models (AlexNet and InceptionV3) that were pre-trained on a large dataset. To ensure these models can be adapted to image recognition for plant leaves, the original parameters of the last fully connected layer were replaced, whereas those of all other convolution layers were retained. Finally, the method proposed in this paper was compared to support vector machine, deep belief network, and CNN through testing on the ICL database. A Tensorflow training network model was used in the comparison test, and the results were visualized by Tensorboard. The testing results showed a considerable improvement in recognition accuracy when using the pre-trained AlexNet and InceptionV3 models, where the training dataset accuracies were 95.31% and 95.4%, respectively.

**Keywords**—convolutional neural network, leaf recognition, transfer learning

## I. INTRODUCTION

PLANT species recognition plays a major role in botanical research [1]. As plant leaves contain abundant information about the plant species and are easy to collect, plant leaf recognition has become the most direct and effective method for plant species recognition. However, the traditional leaf recognition methods mainly rely on manual identifying the shapes of leaves. This type of method involves heavy workloads, is highly inefficient, and is subjective in nature. To date, many effective plant recognition methods have been

developed. Ingrouille and Laird [2] sorted different oaks using principal component analysis based on 27 leaf features. Liu and Kan [3] obtained leaf texture features by combining a local binary pattern (LBP), a gray-scale co-concurrent matrix, and Gabor filtration. They then compared these texture features with profile features such as the Hu invariant moment and Fourier operator and achieved a high recognition rate using a deep belief network (DBN) classifier. Zheng et al. [4] also achieved a high recognition rate using a support vector machine (SVM) classifier based on multiple leaf geometries including the Hu invariant moment, gray-scale co-concurrent matrix, and fractal dimension. Fu et al. [5] recognized and classified different leaves using a k-nearest neighbor method by combining LBP features with the shape features of leaves. After the AlexNet [6] model won the ImageNet [7] Large Scale Visual Recognition Competition (ILSVRC), many deep convolutional neural networks (CNNs) such as GoogleNet [8], VGGNet [9], and ResNet [10] began to emerge. These models have since been applied to an increasing number of research studies for specific image recognition tasks. Jeon and Rhee [11] sorted plant leaves by extracting leaf features automatically using a CNN. An 8-layer CNN was used by Zhang and Huai [12] to train and recognize leaf images included in the PI@antNet leaf dataset as well as their own expanded datasets. The use of CNN+SVM and CNN+SoftMax classification helped to achieve recognition rates of 91.11% and 90.90%, respectively.

CNN model training requires millions of parameters, therefore, many labeled samples are required. Thus, the CNN can perform extremely well only with a complicated network model and a sufficiently large sample size [13]. Missing training samples are likely to result in problems such as over-fitting or trapping in a locally optimized result [6]. To solve this problem, a transfer learning method is proposed. Transfer learning can effectively resolve problems which caused by missing samples, therefore, has been widely used in the field of image recognition.

Considering the limitations of traditional plant leaf image recognition methods, in order to improve the recognition accuracy of plant leaf images, we studied the plant leaf recognition method based on the transfer learning with convolutional neural network.

This work is supported by the Fundamental Research Funds for the Central Universities (2015ZCQ-GX-04) and National Nature Science Foundation of China (Grant No. 31670719).

Yue Zhao is with Beijing Forestry University, Beijing, 100083 China. (e-mail: zhaoyue0609@126.com).

Yili Zheng is with the Beijing Forestry University, Beijing, 100083 China (corresponding author to provide phone: 86-10-62336922; fax: 86-10-62336922; e-mail: zhengyili@bjfu.edu.cn).

Honglei Shi is with Beijing Forestry University, Beijing, 100083 China. (e-mail: Shi\_long\_lei@hotmail.com).

Lu Zhang is with Beijing Forestry University, Beijing, 100083 China. (e-mail: zhanglu@bjfu.edu.cn).

## II. THEORY

### A. Convolutional Neural Networks

Please submit your manuscript electronically for review as e-mail attachments.

A convolutional neural network (CNN) is a deep learning model, the structure of which is shown in Fig.1.

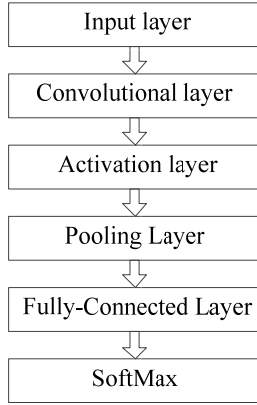


Fig. 1 Architecture of convolutional neural network.

A convolutional layer extracts the local features of an image through convolutional operations. After the feature graph from the previous layer is analyzed by a convolutional kernel, another feature graph from the new layer is obtained through an activation function. The functionality of the convolutional layer is given by the following equation [14]:

$$x_j^l = f \left( \sum_{i \in M_j} x_i^{l-1} \times k_{ij}^l + b_j^l \right) \quad (1)$$

where  $M_j$  is the set of input feature mappings,  $k_{ij}^l$  is the convolutional kernel,  $b_j^l$  is the bias term, and  $f(\cdot)$  is the non-linear activation function. A rectified linear unit (ReLU) function is often used as the activation function, which is defined as

$$f(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases} \quad (2)$$

The pooling layer, which is usually behind the convolutional layer, can reduce the dimensions of the feature maps and increase the robustness of feature extraction [15]. The typical pooling operations are max pooling [16] and average pooling [17]. The pooling process is described by the following equation:

$$x_j^l = \text{down}(x_j^{l-1}) \quad (3)$$

where  $\text{down}(\cdot)$  refers to down sampling.

After the alternate concatenation of several convolutional and pooling layers, further dimension reduction is performed on the remaining fully connected layer in order to extract features. These features are input into a SoftMax classifier to realize classification.

The objective of CNN training is to minimize the loss function by using the stochastic gradient descent (SGD)

method. The loss function is given by

$$L(W, b) = - \sum_{i=1}^N \sum_{j=1}^C I \{ \hat{y}_i = j \} \log p_i^j \quad (4)$$

where  $\hat{y}_i$  is the expectation value of the  $i$  th sample,  $p_i^j$  is the  $j$  th class of the prediction probability of the  $i$  th training sample,  $C$  is the number of classes in the training samples, and  $N$  is the total number of training samples,  $I$  is the indication function, when  $\hat{y}_i = j$ ,  $I$  is 1, otherwise  $I$  is 0.

The iteration process of the SGD weight is expressed in the following two equations as follows:

$$W_i = W_i - \eta \frac{\partial L(W, b)}{\partial W_i} \quad (5)$$

$$b_i = b_i - \eta \frac{\partial L(W, b)}{\partial b_i} \quad (6)$$

where  $\eta$  is the learning rate used for controlling the strength of back propagation.

### B. Transfer Learning

Transfer learning is a machine learning method that focuses on storing gained knowledge while solving one problem and applying it to different but related problems. The objective is to transfer “knowledge” from one domain to related domains. For a CNN, transfer learning involves successfully applying knowledge obtained from a specific dataset to other new domains [18]. As long as there is a relationship between the source domain and the target domain, transfer learning can realize the reuse and transfer of learned knowledge in similar or related fields by means of the knowledge extracted from the source domain data and features, so that the traditional from zero beginning learning becomes accumulative learning, which not only reduces the cost of model training, but also significantly improves the learning effect [19].

Transfer learning can be divided into instance-transfer approach, feature-representation transfer, and parameter-transfer approach according to specific implementation approaches. When the data of the source domain and the target domain are very similar, the instance-transfer approach can effectively solve the problem of insufficient target domain samples; the feature-representation transfer approach finds the potential feature space shared by the source domain and the target domain by reconstructing features to minimize the difference between the domains; parameter-transfer approach which assumes that the source tasks and the target tasks share some parameters or prior distributions of the hyper-parameters of the models. The transferred knowledge is encoded into the shared parameters or priors. Thus, by discovering the shared parameters or priors, knowledge can be transferred across tasks [20].

Donahue et al. [21] used features extracted from a deep CNN model trained on the ImageNet dataset as general visual features. They then applied those features to different tasks such as scene recognition and bird classification and achieved excellent classification results.

### III. PLANT LEAF IMAGE RECOGNITION BASED ON TRANSFER LEARNING

#### A. Architecture of the Model

Both the AlexNet and InceptionV3 [22] pre-trained models were used in this paper. The AlexNet architecture consists of 5 convolution layers, followed by 3 fully connected layers. The first two convolution layers are each followed by a normalization layers. Max-pooling layers follow both normalization layers as well as the fifth convolutional layer. The ReLU non-linearity is applied to the output of every convolutional and fully-connected layer. The InceptionV3 architecture is a much deeper and wider architecture with 46 layers. It consists of 11 inception modules which uses parallel  $1 \times 1$ ,  $3 \times 3$ , and  $5 \times 5$  convolutions along with a max-pooling layer in parallel [23]. The architecture of the InceptionV3 model is shown in Fig.2. First, all the parameters of the convolutional layers in the trained InceptionV3 model were frozen except for those in the last fully connected layer, which were replaced by our trained 42 fully connected SoftMax. The network layers before the last fully connected layer are known as bottleneck layers. Once a plant leaf image is inputted in the network, a 2048-dimensional feature vector is eventually generated in the bottleneck layer. This feature vector is stored in a .txt file and used to train the SoftMax classification.

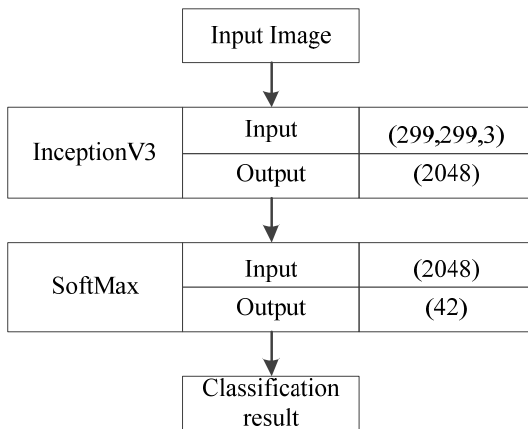


Fig. 2 Architecture of InceptionV3

#### B. SoftMax Classifier.

The SoftMax classifier is a generalization of the logistics regression model on a multi-classification problem. In a multi-classification problem, each class  $y$  can have  $r$  different values. For training set  $\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$ , the corresponding class labels are  $y^{(n)} \in \{1, 2, \dots, r\}$ . For a given input  $x^{(n)}$ , the probability value  $p(y^{(n)} = 1 | x^{(n)}, k = 1, \dots, r$  can be evaluated for each of the  $k$  classes using the assumption function  $h_{\lambda}(x^{(n)})$ . Here,  $h_{\lambda}(x^{(n)})$  is a column vector containing  $r$  rows of entries that can be summed to 1. Each of the entries denotes the probability that a specific classification result will be derived. The assumption function  $h_{\lambda}(x^{(n)})$  can be expressed as [24]:

$$h_{\lambda}(v_i) = \begin{bmatrix} p(y^{(n)} = 1 | x^{(n)}; \lambda) \\ p(y^{(n)} = 2 | x^{(n)}; \lambda) \\ \vdots \\ p(y^{(n)} = r | x^{(n)}; \lambda) \end{bmatrix} = \frac{1}{\sum_{k=1}^r e^{\lambda_k^T x^{(n)}}} \begin{bmatrix} e^{\lambda_1^T x^{(n)}} \\ e^{\lambda_2^T x^{(n)}} \\ \vdots \\ e^{\lambda_r^T x^{(n)}} \end{bmatrix} \quad (7)$$

where  $\lambda_1, \lambda_2, \dots, \lambda_r$  are the parameters of the model. The probability of labeling  $x^{(n)}$  as the class  $k$  is given by

$$p(y^{(n)} = k | x^{(n)}; \lambda) = e^{\lambda_k^T x^{(n)}} / \sum_{l=1}^r e^{\lambda_l^T x^{(n)}} \quad (8)$$

The class  $k$  associated with the maximum value of  $p(y^{(n)} = k | x^{(n)}; \lambda)$  is regarded as the classification result of the current sample. This result will be compared to the real label. The classification is correct if they are the same, and vice versa.

### IV. EXPERIMENTAL DESIGN AND RESULTS ANALYSIS

The experiments were performed on a Jupyter notebook with an Anaconda3 distribution using the open source deep learning framework, Tensorflow, as the developing environment. We used Graphics Processing Unit (GPU) for acceleration, and the GPU used in the experiment is GTX 1080. The loss and accuracy curves in the experimental results are drawn by the data visualized by, TensorBoard [25], which were used to analyze the convergence of the convolutional neural network.

#### A. Experimental Data and Preprocessing

Plant leaf images were obtained from the ICL database [26], which was established by the Intelligent Computing Lab at Hefei Mechanical Research Institute at the Chinese Academy of Sciences. ICL contains leaf samples from approximately 220 types of plants. The number of samples for each type ranges from 26 to 1078. All images were scanned to have a monochrome background and size of  $200 \times 500$  pixels. Forty-two types of plants, with each type containing more than 50 samples, were selected for this experiment. There were 4771 leaf images in total. Some of the images from the dataset are shown in Fig.3.

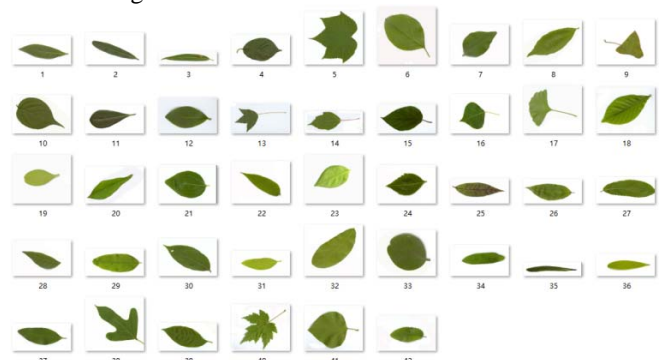
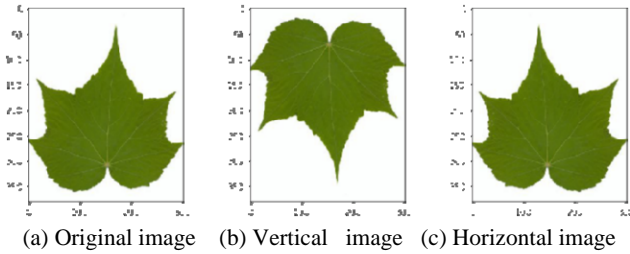


Fig. 3 ICL Leaf Image Dataset.

To eliminate over-fitting to a certain degree, the dataset was expanded by random horizontal and vertical rotation of the original images using Python script. The total number of images in the expanded dataset was thus increased to 10285. The expanded dataset was called ICL-new. It was divided into

training and testing sets at a 4:1 ratio. All images were modified to a uniform size of  $224 \times 224$  pixels and  $299 \times 299$  pixels for fitting in the AlexNet and InceptionV3 models, respectively. Fig.4 shows the series of images obtained after the pre-processing.



(a) Original image (b) Vertical image (c) Horizontal image

Fig. 4 Series of images obtained after preprocessing.

A mean value removal [14] was performed on the samples before processing the images with CNN, as shown by the following equation:

$$x' = x - \mu \quad (9)$$

where  $\mu$  is the mean value of the sample in the training set.

### B. Experimental Setup.

The model used in the experiment was initialized using the weight and bias of the AlexNet model pre-trained on ImageNet. The last fully connected layer of this model was replaced by a 42 fully connected SoftMax classifier. The modified model was eventually used to train the plant leaf image dataset. The learning rate, number of iterations, Batch-size, and dropout were set to 0.001, 50, 64, and 0.5, respectively. For the convenience of description, this method is called "Fine\_Alex."

The feature vectors were obtained by training the plant leaf image dataset with the InceptionV3 model pre-trained on ImageNet. These feature vectors were then used as input to train a 42 fully-connected SoftMax layer. During this procedure, the learning rate and number of iterations of the model were set to 0.01 and 4000, respectively. For the convenience of description, this method is called "Fine\_IncepV3."

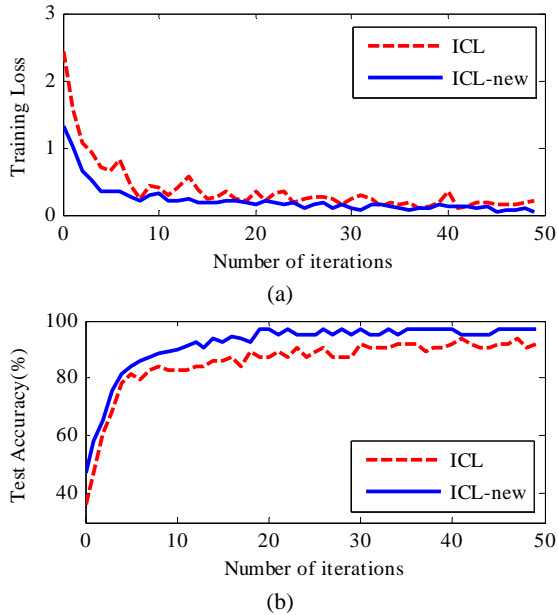


Fig. 5 Training Loss and Test Accuracy of the AlexNet Model in ICL and ICL-new

### C. Experiment Results and Analysis

To verify the effect of data enhancement, we compared the trend of training loss and test accuracy with the number of iterations of our methods on the original ICL dataset and the expanded ICL dataset (ICL-new). In Fig.5 and Fig.6, with the increase of the number of iterations, the training loss of the ICL and ICL-new all decreases gradually, while test accuracy of the ICL and ICL-new all increases. We can also obtain the training loss of the ICL-new is smaller than the ICL, while the test accuracy is higher. The training loss and test accuracy in the model of AlexNet and InceptionV3 reveal that the results on the ICL-new are better than ICL.

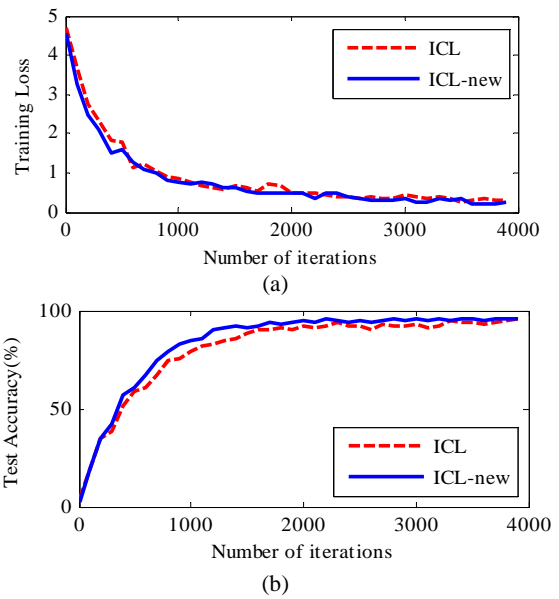
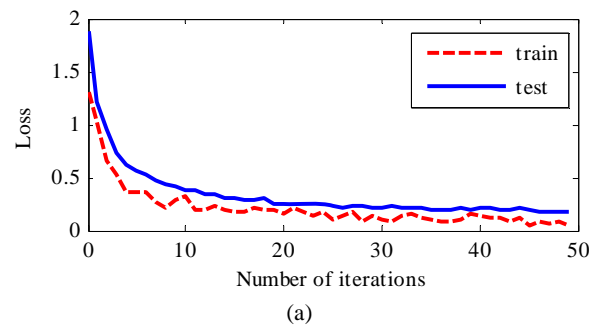


Fig. 6 Training Loss and Test Accuracy of the InceptionV3 Model in ICL and ICL-new.

Figure 7 shows the accuracy and loss of our Fine\_Alex method in ICL-new. Here, the red and blue lines represent the performance of the training and testing datasets. In Fig.7(a), we can see when the number of iterations is 30, the loss of the training set is still changing, while the loss of testing dataset has been stabilized at approximately 0.2. There was over-fitting.



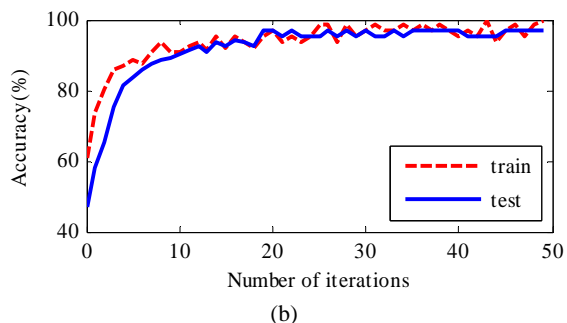


Fig. 7 Accuracy and Loss Curves of the AlexNet Model.

Figure 8 shows the accuracy and loss curves when the Fine\_IncepV3 method was used on the training and testing datasets, respectively. These curves shared a consistent trend in which both the accuracy and loss values basically stabilized when the number of iterations reached 4000. No over-fitting was observed for this model.

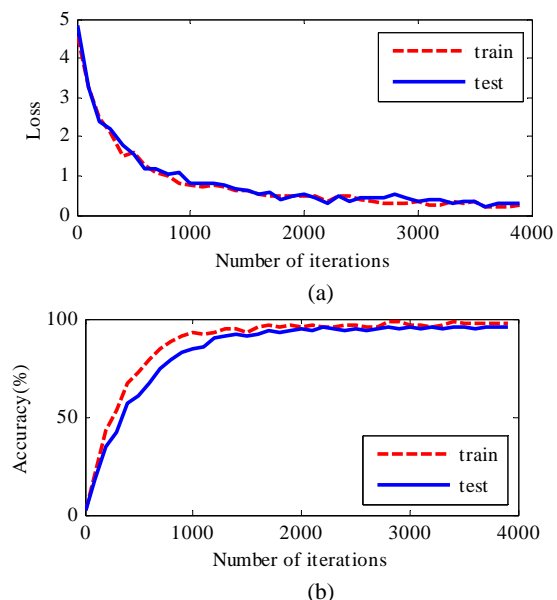


Fig. 8 Accuracy and Loss Curves of the InceptionV3 Model.

In order to evaluate the impact of different feature extraction methods on the accuracy of testing dataset, we reproduced the approach proposed by Zheng *et al.*, [4] and the approach proposed by Liu and Kan [3]. They all extracted the shape and texture features of the plant leaf images. Zheng *et al.*, [4] used the SVM classifier, Liu and Kan [3] uses the DBN classifier. The methods are referred to as SVM, DBN.

To evaluate the performance of our method, a BP neural network model, is also compared. In this part, a two hidden layers and one output layer BP neural network model build with Matlab Neural Network Toolbox[27] is applied, as shown in Fig.9. The BP method uses the features of the literature 4. The numbers of neurons in the hidden layers are 60 and 120, the learning rate is 0.05, the momentum factor is 0.9, and the maximum number of training steps is 10000, the target error is 0.003.

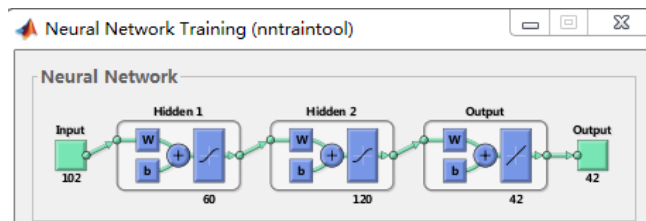


Fig. 9 The Implement BP Model.

The results of contrasting experiments for different methods performed on the ICL-new dataset are listed in Table 1. Among the methods, the CNN method extracted features from the plant leaf image dataset by directly assigning random initial weight parameters to each layer and bias values to the AlexNet model. These features were then classified using the SoftMax classifier with identical parameters set with the Fine\_Alex method. The term “time” referenced in the table represents the total time required to complete training and testing.

Table 1 show that SVM and BP methods were advantageous in terms of time duration but suffered from low accuracy.

Table. 1 Comparison of evaluation index on different methods

Methods	Time [s]	Accuracy [%]
SVM	23.1	89.97
BP	42.2	89.91
DBN	5045	93.94
CNN	8650	41.9
Fine_Alex	2882	95.31
Fine_IncepV3	601	95.4

The reason for this is that SVM and BP methods relied on manual extraction in order to obtain multi-features, a process that requires a certain level of expertise. Compared to SVM and BP methods, DBN method had better accuracy because of the use of a complicated DBN classifier in the classification model. However, because DBN method also used features from manual extraction, it still depended highly on the dataset. In other words, a large volume of data was still required to train its model. Compared with the SVM method, BP method, and DBN method, our methods have improved the accuracy, mainly because the methods of automatically extracting features from the depth model are superior to the traditional artificial feature extraction method.

For a deep learning model, a satisfying result can only be obtained with extensive data. Because the size of the dataset used in this experiment was small, the accuracy of the CNN method remained very low. In addition, the CNN method trained the model from scratch, which was thus time-consuming. The Fine\_Alex method had advantages over the CNN method in terms of time and accuracy. This means that fine-tuning a model pre-trained on a large dataset and applying it to other specific fields is a more time-saving and accurate approach when compared to training from scratch. In other words, applying transfer learning on a small dataset is feasible. Furthermore, we found that the Fine\_IncepV3 method



required less time and was more accurate than the Fine\_Alex method because the former contained more layers and had better convergence compared to the latter.

### V. CONCLUSION

A transfer-learning-based CNN plant leaf image recognition method was proposed in this paper. Pre-trained AlexNet and InceptionV3 models were used to perform transfer training on a plant leaf image dataset. Experimental results showed that these two methods yielded an accuracy of 95.31% and 95.4% on a testing dataset, respectively. Compared to the AlexNet model, the InceptionV3 model contained more layers, which yielded faster convergence without overfitting between the training and testing datasets. We also used a CNN to extract the features of the plant leaf image dataset by directly assigning random initial weight parameters to each layer and bias values to the AlexNet model. We compared this method to the Fine\_Alex method and determined that a well-trained model from a large dataset could be fine-tuned and applied to related fields to achieve a better recognition rate and in less time.

Compared to similar studies, the deep learning model proposed in this paper required fewer pre-processing operations on a testing dataset and extracted the features of plant leaf images automatically and with better accuracy.

The limitation is that all sample images trained and tested in this paper are single background, but in reality, most of the acquired image of the plant leaf has a more complex background. There is no in-depth study of the sample images in the natural environment or in a complex background, so the main goal for the future work is to capture images from complex background and use them to improve the automatic recognition of plant leaf in practice.

### ACKNOWLEDGMENT

This work is supported by the Fundamental Research Funds for the Central Universities (No. 2015ZCQ-GX-04) and National Nature Science Foundation of China (Grant No. 31670719).

### REFERENCES

[1] G. Hou, Z. Weng, H. Li, and W. Li, "An image processing based system for leaf area measurement," *J. Wuhan Botanical Research*, vol. 23, no. 4, pp. 369-372, Aug. 2005.

[2] J. M. Ingrouille and M. S. Laird, "A quantitative approach to oak variability in some north London woodlands," *The London Naturalist*, vol. 65, pp. 35-46, 1986.

[3] N. Liu and J. Kan, "Plant leaf identification based on the multi-feature fusion and deep belief networks method," *J. Beijing Forestry University*, vol. 38, no. 3, pp. 110-119, Mar. 2016.

[4] Y. Zheng, G. Zhong, Q. Wang, Y. Zhao, and Y. Zhao, "Method of leaf identification based on multi-feature dimension reduction," *Chinese Society for Agricultural Machinery*, vol. 48, no. 3, pp. 30-37, Feb. 2017.

[5] B. Fu, Z. Yang, X. Zhao, and Z. Dan, "Plant leaves recognition method based on dimension reduction local binary pattern and shape features of leaves," *Computer Engineering and Applications*, vol. 54, no. 2, pp. 173-176+187, Feb. 2018.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in NIPS, 2012, pp. 1097-1105.

[7] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Computer Vis.*, vol. 3, no.115, pp. 211-252, Dec. 2015.

[8] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Vis. Pattern Recognit.*, Jun. 2015, pp. 1-9.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computer Vision and Pattern Recognition*, Sep. 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Vis. Pattern Recognit.*, 2016, pp. 770-778.

[11] W. Jeon and S. Rhee, "Plant leaf recognition using a convolution neural network," *Int. J. Fuzzy Logic and Intelligent Systems*, vol. 17, no. 1, pp. 26-34, Mar. 2017.

[12] S. Zhang and Y. Huai, "Leaf image recognition based on layered convolutions neural network deep learning," *J. Beijing Forestry University*, vol. 38, no. 9, pp. 108-115, Sep. 2016.

[13] L. Tian, C. Fan, Y. Ming, and Y. Jin, "Stacked PCA Network (SPCANet): An effective deep learning for face recognition," in *Proc. Int. Conf. Digital Signal Processing*, 2015, pp. 1039 - 1043.

[14] Z. Zhong, L. Jin, and Z. Xie, "High performance offline handwritten chinese character recognition using GoogLeNet and directional feature maps," in *Proc. Int. Conf. Document Analysis and Recognition*, 2015, pp. 846-850.

[15] T. Guo, J. Dong, H. Li, and Y. Gao, "Simple convolutional neural network on image classification," in *Proc. IEEE 2nd Int. Conf. Big Data Analysis*. Mar. 2017, pp.721-724.

[16] Y. Boureau, J. Ponce, and Y. Le Cun, "A theoretical analysis of feature pooling in visual recognition," in *Proc. Int. Conf. Machine Learning*, 2010, pp. 111-118.

[17] T. Wang, D. Wu, A. Coates, and A. Ng, "End-to-end text recognition with Convolutional neural networks," in *Proc. Int. Conf. Pattern Recognit.* Nov, 2012, pp. 3304-3308.

[18] Y. Li, Z. Hao, and L. Hang. "Survey of convolutional neural network," *J. Computer Application*, vol. 36, no. 9, pp. 2508-2515, Jan. 2016.

[19] H. Wang. "Review of transfer learning research," *Computer Knowledge and technology*. vol. 13, no. 32, pp.203-205, Nov. 2017.

[20] S. J. Pan and Q. Yang, "A survry on transfer learning," *IEEE Trans. Knowl. Data Eng*, vol. 22, no. 10, pp. 1345-1359, Oct. 2010.

[21] J. Donahue et al., "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Machine Learning*, vol. 32, 2014.

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE. Conf. Vis Pattern Recognit.*, 2016, pp. 2818-2826.

[23] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.* vol. 7, pp.1-10, Sep. 2016.

[24] Q. Song, Z. Tian, W. Sun, X. Wu and Y. An. "Combined dimension reduction method for isolated digital speech recognition," *J. Xi'an Jiaotong University*, vol. 50, no. 6, pp. 42-46, Apr.2016.

[25] E. Gibson et al., "NiftyNet: a deep-learning platform for medical imaging," *Computer Methods and Programs in Biomedicine*, vol. 158, pp. 113-122, May. 2018.



**YUE ZHAO** received her B.S. degree in Measuring and Control Technology and Instrumentation from Harbin Engineering University in 2009. And received her Ph.D. degree in control science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2014. She joined the faculty of the Department of Automation, Beijing Forestry University, in 2014, where she has been an Associate Professor since 2018. Her research interests include deep learning, image processing and data processing.



**Yili Zheng**, born in 1981, received the B.S degree in Automation from National University of Defense Technology Changsha, China, in 2004. In 2009, he received the Ph.D. degree in Mechatronics Engineering from Beijing University of Posts and Telecommunications, Beijing, China. He is currently an associate professor of forestry electrification and automation at Beijing Forestry University, Beijing, China. His research interests include intelligent forestry and forestry informatization.



**Honglei Shi**, born in 1998, received the B.S degree in Automation from Qingdao Agricultural University, China, in 2019. She is currently a master student of control theory and control engineering at Beijing Forestry University, Beijing, China. Her search interests include deep learning and digital image processing.



**Lu Zhang**, born in 1995, received the B.S degree in electrical engineering and automation from Beijing Forestry University, China, in 2016. She is currently a master student of control theory and control engineering at Beijing Forestry University, Beijing, China. Her search interests include deep learning and digital image processing.