Vegetable Pest Image Recognition Method Based on Improved VGG Convolution Neural Network

Haijian Ye, Hang Han, Linna Zhu, Qingling Duan
College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China

Abstract. Vegetables are one of the main crops in China, and pests are one of the important factors affecting the quality of vegetables. In order to improve the recognition accuracy of vegetable pest images, a vegetable pest image recognition method based on improved VGG convolution neural network is proposed. Based on the VGG16 and VGG19 models, the method optimizes the number of full connection layers, replaces the original SoftMax classifier in VGGNet with the three-label SoftMax classifier, optimizes the structure and parameters of the model, and uses the weight parameters of convolution layer and pooling layer in the pre-training model in transfer learning. Experiments were carried out on the self-expanding data set of vegetable pest images, and the performance of the method was tested. Tensorflow was used to train the network model. The experimental results showed that the pre-trained models (VGG16, VGG19, Inception V3, ResNet50) were trained on the vegetable pest image data set to adapt to the recognition of vegetable pest images. The experimental results also showed that compared with Inception V3 and ResNet50, the recognition accuracy of the pre-trained models using VGG16 and VGG19 were higher, and the test accuracy of the two models were 99.90% and 99.99% respectively. Finally, the methods were compared with the traditional VGG method in self-expanding data sets. The results showed that the accuracy of VGG16 model and VGG19 model were improved from 85.90% and 86.21% to 100% and 100% respectively; the classification accuracy of VGG16 model was improved from 64.02% to 99.90%, and the classification accuracy of VGG19 model was improved from 85.83% to 99.99%, which effectively improved the recognition accuracy.

1. Introduction
Vegetable pest species identification is of great significance and play an important role in the prevention and control of vegetable pests. Traditional vegetable pest identification methods are mainly based on the images’ color, shape, texture and other characteristics of pests, which have some limitations, such as heavy workload, low efficiency, strong subjectivity and large delay. There are many effective methods of vegetable pest identification until now. In recent years, convolutional neural network (CNN)[1] has become the main method of various computer vision tasks[2-4]. Natalia et al. developed an automatic fruit flies classification system, which can classify four kinds of fruit flies, namely, Calineeuria, Doroneuria, Hesperoperla and Yoraperla, with 82% accuracy[5]; Gassoumi H et al. developed a classification and recognition algorithm based on fuzzy neural network, which can distinguish twelve kinds of common insects in cotton field. And eleven of kinds of them classification accuracy reached more than 90%. As AlexNet[3], GoogleNet[7], VGGNet[6], ResNet[8] and other deep convolution neural network models constantly emerged, more and more researchers use these models to study specific image recognition. Nicati Casmu et al. used VGGNet and ResNet models to classify oasis plant...
communities in the hinterland of desert automatically[9].

The training of CNN model requires millions of parameters, so a large number of labeled samples are needed in CNN training. Only when the network depth is enough and the training samples are enough, can CNN show better performance[10]. Conversely, if samples’ size is small, it is easy to be over-fit or fall into local optimal solution. Transfer learning is proposed to solve the above problems, by insufficient number of samples. Therefore, transfer learning has a wide range of applications in the field of image recognition[11].

This paper presents a vegetable pest image method based on improved VGG convolution neural network. By optimizing the number of full connection layers and replacing the original SoftMax classifier in VGGNNet with a three-label SoftMax classifier, the structure and parameters of the model are optimized, combining the weighting parameters of the convolutional layer and the pooling layer in the pre-training model to train the improved model. Trained convolutional neural network model is used to extract the features of pest images. The extracted feature vectors are trained by classifiers, and the recognition results are obtained finally.

2. Materials and methods

2.1. Convolutional Neural Network

Convolutional neural network (CNN) is often used to process large image datasets. Convolutional neural network is trained by back propagation algorithm. Compared with traditional methods, convolutional neural network has an important advantage in that it can directly extract feature vectors from images to identify images with minimal preprocessing. Its network structure is shown in Figure 1.

![Architecture of convolutional neural network](image)

Fig.1 Architecture of convolutional neural network

(1) Convolution Layer

Convolution operation is taken on the convolution layer, extracts the filtering features of the image, and combines the convolution results into a feature map, which serves as the input data of the next layer. The working process of convolution layer is as follows:

\[
y_{conv} = f \left( \sum_{j=0}^{M} \sum_{i=0}^{N} x_{m+i, n+j} \cdot \omega_{ij} + b \right) \quad (0 \leq m \leq M, \ 0 \leq n \leq N)
\]

In the formula, \(x\) represents the two-dimensional vector of receiving region (M, N); \(\omega\) represents the convolution core whose length and width are j and i respectively; \(b\) represents the bias term added to each output feature map, which is the result of the complete output; \(M\) represents the length of the two-dimensional vector and \(N\) represents the width of the two-dimensional vector; and \(f\) represents the non-linear activation function. In this paper, the non-linear activation function uses Rectified Linear Units (RELU) function. The expression of the RELU function is:

\[
f_{relu}(x) = \begin{cases} 
0, & x < 0 \\
x, & x \geq 0
\end{cases}
\]

In the formula, \(f_{relu}(x)\) represents the result of RELU function; \(x\) represents the vector of the acceptance region.

(2) Pooling Layer

Generally, the pooling layer is behind the convolution layer, and its main function is to reduce the dimension of the feature map, which is composed of the convolution results of the upper layer, while maintaining the local invariance of the feature data[16]. Take the maximum pooling function with scale size two and step size two as an example.

\[
f_{pool} = \text{Max} \left( x_{m, n}, x'_{m+1, n}, x_{m, n+1}, x'_{m+1, n+1} \right) \\
(0 \leq m \leq M, \ 0 \leq n \leq N)
\]
In the formula, $f_{pool}$ represents the maximum pooling result of feature graph.

(3) Full Connection Layer

The full connection layer is behind the continuous stacked convolution layer and pooling layer. The full connection layer takes the dimension reduction process of the global features and integrates them into the SoftMax layer for classification. The function of the SoftMax layer in the convolution neural network is to classify the data according to the probability.

SoftMax layer is a classifier for solving multi-classification problems, and it is an evolutionary extension of logistic regression model for multi-classification problems. In multi-classification problem, classification label $y$, training set $T=\{(x(1), y(1)), (x(2), y(2)), \ldots, (x(n), y(n))\}$, classification label $y(i) \in \{1, 2, \ldots, n\}$, $x(i)$ represent the input sample sets. For each specific class $j$, the probability of classifying it as class $n$ label is estimated by using hypothesis function $h_j(x(i))$.

$$P(y=j | x) \begin{cases} \text{for } j = 1, 2, \ldots, n \end{cases}$$

(4)

Formula of Hypothesis function $h_j(x(i))$ as following:

$$h_j(x(i)|\gamma) = \begin{bmatrix} p(y(i) = 1 | x(i), \gamma) \\ p(y(i) = 2 | x(i), \gamma) \\ \vdots \\ p(y(i) = n | x(i), \gamma) \end{bmatrix} = \frac{1}{\sum_{j=1}^{n} e^{\gamma_j^T x(i)}} \begin{bmatrix} e^{\gamma_1^T x(i)} \\ e^{\gamma_2^T x(i)} \\ \vdots \\ e^{\gamma_n^T x(i)} \end{bmatrix}$$

(5)

Among them: $\gamma_1, \gamma_2, \ldots, \gamma_n$ are learning parameters for the model.

The probability that $x(i)$ divided into $n$ class is recorded as:

$$P(y(i) = j | x(i); \gamma) = \frac{e^{\gamma_j^T x(i)}}{\sum_{j=1}^{n} e^{\gamma_j^T x(i)}}$$

(6)

The classification result of the current sample is the category $n$ corresponding to the maximum value $P(y(i) = j | x(i); \gamma)$, which is compared with the real label of the sample itself. If it is consistent, the classification result is correct, otherwise the classification result is wrong.

2.2 Transfer learning

Transfer learning is a machine learning method, which uses existing knowledge to solve different but related domain problems. It aims to complete the transformation of different knowledge among related domains[12]. For convolutional neural networks, transfer learning is the successful application of "knowledge" trained on specific data sets to new fields[13-14].

3. Vegetable pest image recognition based on improved VGG convolution neural network

VGGNet network[6] is a network model with excellent classification performance in convolutional neural networks. It is developed on the basis of AlexNet network. Its essence lies in using 3*3 small convolution core to design and build the network. The network depth of VGGNet is 16-19 layers. VGGNet network not only has good classification effect on large-scale data sets, but also has excellent expansion ability on other data sets.
3.1. Experimental data

3.1.1 Data Sources

In this paper, vegetable pest images were collected in Greenhouse of West Campus of China Agricultural University from 2017 to 2018. The pest images of rape, common cabbage, Chinese cabbage and radish were collected in greenhouse environment. Considering the influence of weather conditions on image quality, we choose sunny and cloudy days for image acquisition. In order to ensure the image quality, the camera is 20-40 cm away from the worms, and the shooting angles are 90 and 60 respectively. The acquisition device is Canon SLR digital camera, model 650D. The focal length is set to autofocus. The resolution of the acquisition image is 5184 pixels * 3456 pixels. A total of 2814 images were collected to form the vegetable pest image data collection, including 802 aphids, 1046 cotton bollworms and 966 diamondback moths. The image of vegetable pests is shown in Figure 3.

![Vegetable pest image](image)

Fig.3 Vegetable pest image

3.1.2 Data Preprocessing

The image size is normalized to 224 pixels*224 pixels. The image data sets of three pests were divided into training set and test set in a ratio of 6:1. In order to minimize over-fitting, matlab script statement is used to expand the data set by random horizontal, horizontal, vertical and color change operations on training set image and test set image respectively. The total number of self-expanding data set samples is 14070. Fig. 4 is a partial image of a selected image after preprocessing. Before using the network to process the image, the sample is de-averaged[15]. The formula is:

\[ x' = x - \theta \]  

(7)

In the formula, \( x' \) represents the sample after removing the mean, \( x \) represents the sample, and the
sample mean on the training set is the sample. $\theta$ represents the sample mean on the training set.

(a) original image  
(b) horizontal flip  
(c) vertical flip  
(d) random horizontal flip  
(e) color change

Fig. 4 Partial image obtained by preprocessing

3.2 Vegetable Pest Image Recognition Model Based on Improved VGG Convolutional Neural Network

Firstly, the VGG19 and VGG16 models are improved. Combined with the research in this paper, two full-connection layers are proposed to replace the original three full-connection layers. The first full-connection layer is 4096 nodes, the second full-connection layer is 3 nodes, and the three-label SoftMax classifier is used to replace the SoftMax classification layers of each model for the classification of the research objects in this paper. Then, the parameters and weights of the pre-training model are added to the ImageNet database of 1 million images for the two models. Finally, experiments are carried out on the self-expanding data set. Taking VGG19 convolutional neural network as an example, the research method in this paper is introduced. On the basis of retaining all convolution layer parameters in the pre-trained VGG19 model, the training set is input into the improved network for training, and the optimal model parameters of the whole network are obtained[17]. Based on VGG19 convolutional neural network and the research object of this paper, a model for identifying three kinds of vegetable pest images was built (Fig. 5).

4. Design of experiments and analysis of results

The experimental running environment of this paper is Spider in Anaconda3, which uses the open source deep learning framework Tensorflow as the development environment. In this paper, the Loss curve and
the accuracy curve of the experimental results are drawn by using MATLAB R2014b to analyze the convergence of convolutional neural network.

4.1 Experimental Settings

The VGG16 model is used to train the self-expanding training set from scratch. The learning rate of the model is 0.0001, the number of iterations is 50, the Batch_size is 32, and the Momentum is 0.9. It is named VGG16Net to describe this method.

The VGG19 model is used to train the self-expanding training set from scratch. In order to reduce over-fitting, a regularization term is added to the network, and the self-expanding training set is trained from scratch using the modified model. The initial learning rate of the model is 0.001, the number of iterations is 50, the Batch_size is 8, and the Momentum is 0.9. It is named VGG19Net to describe this method.

The weights and deviations of the VGG16 model trained on ImageNet are used to initialize the model. The parameters of the pre-training model are used to optimize the model parameters of the convolution layer. At the same time, the full connection layer of the model is optimized. Finally, the self-expanding data set is trained. The learning rate of the model is 0.0001, the number of iterations is 50, the Batch_size is 32, and the Momentum is 0.9. It is named Fine_VGG16 to describe this method.

The weight and deviation of the VGG19 model trained on ImageNet are used to initialize the model. The parameters of the pre-training model are used to optimize the model parameters of the convolution layer, and the full connection layer of the model is optimized. Finally, the self-expanding data set is trained. The learning rate of the model is 0.0001, the number of iterations is 50, the Batch_size is 32, and the Momentum is 0.9. It is named Fine_VGG19 to describe this method.

The Inception V3 model trained on ImageNet is used to train the self-expanding data set and obtain the feature map. The feature map is input into the full connection layer to get the classification results. The learning rate of the model is 0.001, the number of iterations is 100, the Batch_size is 32, the Momentum is 0.9. It is named Fine_Inception V3 to describe this method.

The parameters of ResNet50 model trained on ImageNet are used to optimize the convolution layer parameters, and then the self-expanding data set is trained. The learning rate of the model is 0.001, the number of iterations is 100, the Batch_size is 32, and the Momentum is 0.9. It is named Fine_ResNet50 to describe this method.

4.2 Experimental results and analysis

Fig. 6 shows the accuracy of VGG16Net method and the curve of Loss value varying with the number of iterations. As can be seen from Fig. 6a, the final accuracy of modeling is 85.90%. As can be seen from Fig. 6b, the Loss value of the training set is finally close to 0.000015. The accuracy of the test set was 64.06%. The final loss value is 0.000014. Because the data set used in this paper belongs to small data set, there exists over-fitting phenomenon.
Fig. 7 shows the accuracy of VGG19Net method and the curve of Loss value varying with the number of iterations. As can be seen from Figure 7a, the final accuracy of modeling is 86.21%. As can be seen from Figure 7b, the Loss value of the training set is finally close to 7.6994. The accuracy of the test set was 85.83%. The final Loss value was 7.6993. Compared with VGG16Net method, VGG19Net method has higher accuracy on test set.

In Fine_VGG16 method, the accuracy of training set is increased to 100% in 16 iterations, and the accuracy of test set is basically stable at 99.90%. The Loss value of training set is close to 0.0100. The Loss value of the test set is about 0.0160.

With Fine_VGG19 method, the accuracy of training set is increased to 100% in 20 iterations, and the accuracy of test set is basically stable at 99.99%. The Loss value of training set is close to 0.0100. The Loss value of the test set is about 0.0130.

Fig. 8 shows the curve of the accuracy of Fine_Inception V3 method and the Loss value varying with the number of iterations. As can be seen from Fig. 8a, the accuracy of training set modeling is stable at about 96.50%, and the accuracy of testing set is basically stable at 39.51%. As can be seen from Fig. 8b, the Loss value of training set is close to 0.1967 and that of test set is 4.1753. There is a big gap between the LOS value and accuracy of training set and test set. It can be seen that the generalization ability of this model is weak for the object of this study, and there is an over-fitting phenomenon.

In Fine_ResNet50 method, the accuracy of training set modeling is stable at about 99.90%, and the accuracy of testing set is basically stable at 93.17%. From Fig. 9b, we can see that the
Loss value of training set is close to 0.2500 and that of testing set is about 0.2450. Although their trends are basically the same on the test set and training set, there are some gaps between the Loss values of the training set and the test set, and the final accuracy of the test set is slightly lower than that of the modeling, and there is a certain over-fitting phenomenon.

![Accuracy and Loss Function Curves](image)

The experimental results of these methods on the self-expanding data set in this paper are shown in Table 1, where time is the time spent when training and testing are completed.

| Methods            | Time/s | Training accuracy/% | Testing accuracy/% |
|--------------------|--------|---------------------|--------------------|
| VGG16Net           | 12112  | 85.90%              | 64.06%             |
| VGG19Net           | 26276  | 86.21%              | 85.83%             |
| Fine_InceptionV3   | 1718   | 96.50%              | 39.51%             |
| Fine_ResNet50      | 2902   | 99.90%              | 93.17%             |
| Fine_VGG16         | 10598  | 100%                | 99.90%             |
| Fine_VGG19         | 12529  | 100%                | 99.99%             |

From Table 1, we can see that the accuracy of VGG16Net and VGG19Net is similar, but because the depth of VGG16 network model is not as deep as VGG19, the generalization ability of the model is poor, and the accuracy of the test is lower than that of the latter; in terms of efficiency, because both need to start training from scratch, it takes a long time. Compared with VGG16Net and VGG19Net, Fine_VGG16 and Fine_VGG19 improved the accuracy, testing accuracy and time efficiency of the model. It shows that model training on large data sets and fine-tuning the trained model to specific fields will take less time and have a higher accuracy than training from the beginning. It is feasible to use the method of transfer learning in small data sets.

Combining the migration learning method, Fine_Inception V3 and Fine_ResNet50 have great advantages in time. However, because of the large structure of Inception V3 and ResNet50 network models, the depth model needs a lot of training data to obtain ideal experimental results. The data set in this experiment belongs to small data sets, so the accuracy of both methods is low. This phenomenon is especially evident in Fine_Inception V3 method. Fine_VGG16 method and Fine_VGG19 method have better accuracy, and because the structure of VGGNet network is relatively simple, the training effect is better on small data sets, which shows that these two methods are more suitable for this data set.
Fine_VGG16 method takes less time than Fine_VGG19 method, and the accuracy of the two methods is similar, but for this data set, the convergence speed of the former is faster.

Overall, Fine_VGG16 method and Fine_VGG19 method have higher training and recognition accuracy than Fine_Inception V3 method and Fine_ResNet50 method for this data set. Compared with traditional VGGNet method, Fine_VGG16 method has higher accuracy and time efficiency.

5. Conclusion
On the basis of VGG convolution neural network, two full-connection layers are used to replace the three original full-connection layers, and three-label SoftMax classifier is used to replace the SoftMax classifier in VGGNet. The training and recognition experiments of three vegetable pest images such as aphid, cotton bollworm and diamondback moth are carried out. The following conclusions are drawn:

1. A vegetable pest image recognition method based on improved VGG convolution neural network is proposed. In this paper, two methods are studied to compare Inception V3 and ResNet50 models combined with transfer learning method for transfer training on self-expanding vegetable pest image data set. The experimental results show that compared with Inception V3 and ResNet50, VGG16 and VGG19 have higher accuracy on test set, reaching 99.90% and 99.99% respectively, and have better classification performance on vegetable pest image.

2. Comparing the traditional VGG16Net and VGG19Net methods with the Fine_VGG16 and Fine_VGG19 methods studied in this paper, the results show that the optimization of the traditional VGGNet model combined with transfer learning can not only reduce the time-consuming, but also a better result in image recognition.

3. The research methods in this paper are trained and tested on 14070 images of three self-expanding vegetable pest image datasets. In the future, we will continue to collect vegetable pest images, expand the number and types of vegetable pest image datasets, increase training samples, and further improve the performance of vegetable pest identification.

4. In this paper, we improve the whole connection layer of convolutional neural network. In the later stage, we need to optimize the structure of other layers of the model, further optimize the network structure of the model, and strengthen the feature extraction of the model for vegetable pests.

Acknowledgments
This work was financially supported by the Tianjin Agricultural Science and Technology Achievements Transformation and Promotion Project (201704070).

Reference:
[1] Lecun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553):436.
[2] Oneata D, Revaud J, Verbeek J, et al. Spatio-temporal Object Detection Proposals[J]. 2014.
[3] Krizhevsky A, Sutskever I, Hinton G. ImageNet Classification with Deep Convolutional Neural Networks[C]// NIPS. Curran Associates Inc. 2012.
[4] Long J, Shelhamer E, Darrell T. Fully Convolutional Networks for Semantic Segmentation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 39(4):640-651.
[5] Larios N, Deng H, Zhang W, et al. Automated Insect Identification through Concatenated Histograms of Local Appearance Features[C]// IEEE Workshop on Applications of Computer Vision. IEEE, 2007.
[6] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.
[7] Szegedy C, Liu W, Jia Y, et al. Going Deeper with Convolutions[J]. 2014.
[8] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. 2015.
[9] Nicati Casmu, Shiqingdong, Liu Suhong, Billari Yiming, Li Hao. An automatic classification method of plant communities in desert hinterland Oasis Based on VGGNet and ResNet models [J/OL]. Journal of Agricultural Machinery: 1-19 [2019-01-25](in Chinese).
[10] Tian L, Fan C, Ming Y, et al. Stacked PCA Network (SPCANet): An effective deep learning
for face recognition[C]/ Digital Signal Processing (DSP), 2015 IEEE International Conference on. IEEE, 2015.

[11] Russakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge[J]. International Journal of Computer Vision, 2014, 115(3):211-252.

[12] Day O, Khoshgoftaar T M. A survey on heterogeneous transfer learning[J]. Journal of Big Data, 2017, 4(1):29.

[13] Wang Wenpeng, Mao Wentao, He Jianliang, et al. Smoke Recognition Method Based on Deep Migration Learning [J]. Computer Applications, 2017 (11): 144-149+161 (in Chinese).

[14] Wang Liwei, Li Jiming, Zhou Guomin, Yang Dongyong. Application of depth transfer learning in hyperspectral image classification[J/OL]. Computer engineering and application: 1-8 [2019-01-25] (in Chinese).

[15] Planas, Santiago, et al. "Performance of an ultrasonic ranging sensor in apple tree canopies." Sensors 11.3 (2011): 2459-2477.

[16] Yandong L I, Zongbo H, Hang L. Survey of convolutional neural network[J]. Journal of Computer Applications, 2016.

[17] Xie J, Ding C, Li W, et al. Audio-only Bird Species Automated Identification Method with Limited Training Data Based on Multi-Channel Deep Convolutional Neural Networks[J]. 2018.