ASEAN Engineering
Journal

Full Paper

SOYBEAN PEST IDENTIFICATION BASED ON CONVOLUTIONAL NEURAL NETWORK AND TRANSFER LEARNING

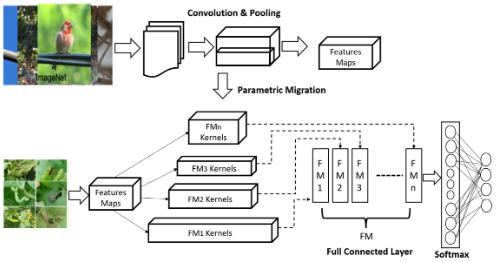
Xin MingYuan*, Ang Ling Weay

School of Information Technology, Malaysia University of Science & Technology, MUST, 47810, Petaling Jaya, Selangor, Malaysia

Article history
Received
29 April 2022
Received in revised form
18 July 2022
Accepted
22 July 2022
Published online
28 February 2023

*Corresponding author
xin.mingyuan@phd.must.edu.my

Graphical abstract



Abstract

Despite the fact that the ensemble classifier improved classification precision by integrating texture and colour. However, the significant image preparation process is a laborious and time-consuming. Manually depicting constrained feature extraction can result in a semantic void in a picture. These limits cause inaccuracy of agricultural disease identification. Thus, this study proposes a soybean pest detection method based on a hybrid of Transfer learning and pyramidal convolutional neural networks that can identify soybean pests quickly and accurately on small sample sets. Bean borer, soybean poison moth, mite, stink bug, and pod borer photographs are first pre-processed using standard data improvement methods, and then manually categorized into six groups based on pest characteristics. The weight parameters from the VGG16 model trained on the ImageNet image dataset were then transferred to the recognition of soybean pests using the transfer learning method, and the VGG16 convolutional and pooling layers were used as feature extraction layers, while the top layer was redesigned as a pyramidal convolutional layer, an average pooling layer, and a SoftMax output layer, with some of the convolutional layers frozen during training. According to the testing statistics, the model's average test accuracy is 98.23%, and the model size is only 95.4 M. For bean borer, soybean poison moth, mite, skewed night moth, stink bug, and bean pod moth, the model's recognition accuracy is 96.4, 97.78, 98.12, 98.4, 99.56, and 99.16, respectively. The results of the experiments show that the method has a high identification efficiency and a good recognition effect.

Keywords: CNN, Transfer learning, Pyramidal convolution, VGG16 model, Features Extraction

1.0 INTRODUCTION

Plant leaf disease is one of the most significant components controlling plant growth. In current agriculture, fast and effective identification of crop leaf diseases, as well as focused management measures, are crucial for crop output and quality stability. Plant leaf disease species are generally detected by consulting specialists or experienced farmers.

However, such knowledge acquisition has certain drawbacks, such as high cost, low timeliness, and unpredictable accuracy. Therefore, automated detection of plant disease species by means of machines has gained more and more popularity in current agricultural management. The aim of this study is to enhance the standard convolutional neural network and combine it with transfer learning to accomplish quick and accurate detection of soybean pests. By adopting a pyramid coiling layer to substitute for the standard convolution layer, multi-scale picture information may be acquired without incurring additional computational expense. Transfer learning approach is used in model training to compensate for limited data sample size and extended training duration. Compared with new learning, transfer learning is more helpful in increasing model performance and expediting network convergence.

In recent years, significant progress has been achieved in the use of computer vision technology and machine learning algorithms to improve crop disease image recognition. Crop disease detection and recognition using machine learning will provide detailed information for early disease diagnosis and treatment. Support vector machine (SVM) and K-Nearest Neighbor are the most commonly used categorization algorithms (KNN). For instance, [1] categorised 100 photos of various corn diseases after segmenting the lesion region and extracting features. [2] classified insect data sets using a number of machines learning techniques, including the base classifier and the ensemble classifier, and evaluated the classification results using majority voting. The results indicated that using the majority voting approach, the ensemble classifier enhanced classification accuracy by incorporating texture, colour, shape, and other variables. However, they have some limitations. Firstly, performing extensive image preparation is tedious and time-consuming. Second, manually representing restricted feature extraction can easily result in an image semantic gap. These restrictions directly contribute to the inaccuracy and slowness with which agricultural disease identification is used.

Nowadays, with the rapid improvement of device computing capability and the continuous development of big data technology, more and more researchers turn their attention to the field of deep learning. Traditional deep learning algorithms rely on large-scale and class-balanced data for training, and perform well in multiple tasks in computer vision, natural language processing, speech, and other fields, and even outperform humans in some tasks. For example, [3] used a deep convolutional neural network for the detection and diagnosis of plant leaf disease, and the recognition power could reach 99.53%. The model training data came from the public database and contained 87848 images. [4] used the structure of a dense convolutional neural network to identify maize leaf disease species and obtained 12332 images through manual collection for model training, with the highest accuracy reaching 98.06%. [5] makes use of the AlexNet model and GoogleNet model in the pre-training model trained by the ImageNet data set and then applies it to the PlantVillage data set task through partial adjustment of the model. This dataset yields satisfactory results for the 26 diseases covered by 14 crops. The deep network structure of deep learning corresponds to the huge network model parameters and usually requires a large amount of sample data for training to obtain better results. If the training data is insufficient, the network model is prone to overfitting. However, data collection is extremely difficult in a variety of practical applications for a variety of reasons, some of which are related to the high cost of human and material resources, and others of which are related to events with a low probability that cannot be obtained in a short period of time. However, human learning does not require a large amount of data in order to generate superior results. For example, when a youngster is introduced to a new animal, a picture frequently teaches them how to identify it.

To enable computers to learn in the same way as humans, transfer learning can be used to instruct the neural network trained on the large image dataset to complete the target recognition task by making small adjustments to the small target dataset in a specific way [6] [7]. This will help to effectively compensating for the lack of data and reducing calculation time. AlexNet made a significant breakthrough in 2012 when it was used to the ImageNet classification challenge for large-scale picture datasets [8]. Deep convolutional neural networks with a variety of different topologies have been suggested and applied to the job of image categorization [5,65,66]. The top 2014 summit is on artificial intelligence. [9] shown that the depth of the neural network demonstrates that mobility grinds down. They proposed numerous learning approaches based on depth migration algorithms, and they developed substantial learning algorithms that outperform those based on the depth migration effect in the field of image classification.

2.0 METHODOLOGY

Figure 1 shows the main steps of the soybean pest identification algorithm described in this paper, which is based on convolutional networks and transfer learning. Firstly, the collected soybean disease image dataset is enhanced and preprocessed; the feature parameters are extracted using the convolutional layer of VGG16. The weight parameters obtained through pretraining on ImageNetare used as the transfer learning technique's initial parameters [10]. The training data and input are combined. Then, using the training data and the input pyramidal convolutional layer, we modified the structure of the fully connected layer and the output layer of the model to correspond to the disease category, retrained and fine-tuned the model parameters. The dynamic learning rate adjustment strategy is used to retrain and fine-tune the parameters of the convolutional and fully connected layers during the training process. The learning rate of the fully connected layer was increased. Finally, the processed images were fed into the network for training, and the results were compared to those obtained using more established models such as AlexNet, VGG19, and ResNet18.

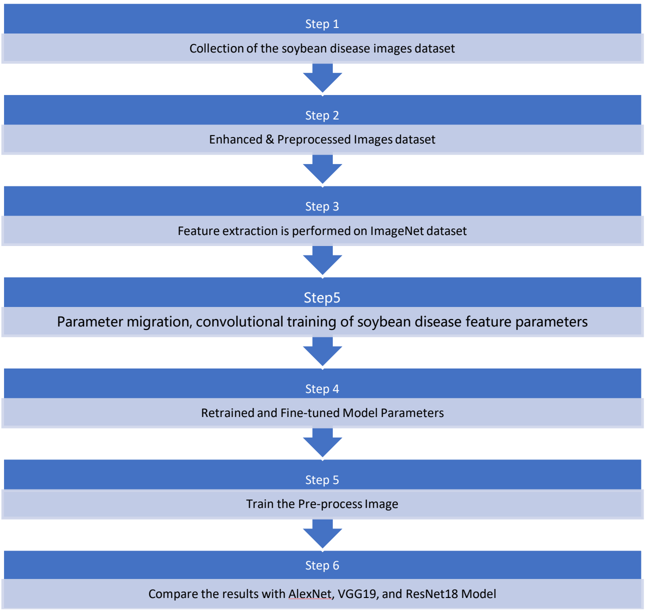

Step 1
Collection of the soybean disease images dataset
Step 2
Enhanced & Preprocessed Images dataset
Step 3
Feature extraction is performed on ImageNet dataset
Step5
Parameter migration, convolutional training of soybean disease feature parameters
Step 4
Retrained and Fine-tuned Model Parameters
Step 5
Train the Pre-process Image
Step 6
Compare the results with AlexNet, VGG19, and ResNet18 Model

Figure 1. Research Flow

Transfer Learning

Transfer learning is the process of applying information or experience gained on one activity or problem to other but related tasks or problems. Transfer learning is the process of establishing a correlation between two domains and adapting the information and experience gained in the previous sectors to the new field by the discovery of relationships between data, tasks, or models. Machine learning aims to derive a problem model from a large amount of available data and apply it to the required task, whereas transfer learning focuses on establishing a connection between the source domain task and the target domain task, as well as fine-tuning the knowledge and model acquired in the source domain and applying it to the target domain [11].

Formal definition of Transfer learning: Assume that there is a source domain D_s={x_i,y_i}(i= 1,...,n) and a target field D_t={x_j}(j = n + 1,...,m). The sample probability distributions P(X_S) and P(X_t) for these two domains are generally different. The purpose of transfer learning is to transfer the knowledge learned in the source domain to the target domain. D is data. (Source domain) is the domain with a large amount of available data that can be learned, represented by D_s; target domain is the domain that requires us to learn knowledge, represented by D_t. A sample of a learning process, which we usually describe in terms of x, is also a description of a tensor. In general, X_i is the first i data or sample Information. A task is a target to be learned, including prediction labels and learning objective functions.

Pyramidal convolution

Pyramidal convolution structure contains n convolution layers of different types of convolution kernels [12], without any increase in computational cost and number of parameters under the premise of different dimension convolution kernels are the characteristics of the input figure, different types with different spatial resolution and depth, in order to capture more detail information, a smaller field of kernel can focus on details, Increasing the kernel size provides more reliable information between contexts [13]. With the increase of space size, the depth of the convolution kernel decreases from the first level to the N level, and finally forms multiple convolutional branches. Feature mapping group convolution through different convolutional branches, and finally cascade the features of different branches to form a fusion feature. The Pyramidal convolution structure is shown in Figure 2.

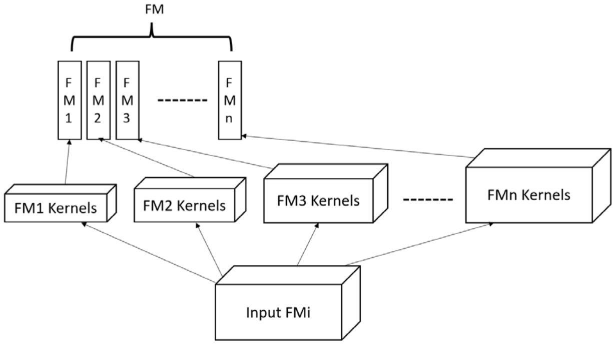


Figure 2. Pyramidal convolution structure

The convolution kernel is equivalent to the filtering operator in digital image processing. For input feature mapping FM_i, convolution kernels of different space sizes are applied to different pyramid convolution layers {k_1^2, k_2^2, k_3^2,..., k_n^2} and different core depths {FM_i, FM_i /(k_2^2 /k_1^2), FM_i /(k_3^2 /k_1^2)... , FM_i /(k_n^2 /k_1^2)}, and output the corresponding output mapping $\{FM_1, FM_2, FM_3 \ldots , FM_n\}$, so the parameters formula of pyramid convolution can be obtained as follows:

$$parameters = k_n^2 \cdot \frac{FM_i}{\left(\frac{k_n^2}{k_1^2}\right)} \cdot FM_n + \cdots + k_2^2 \cdot \frac{FM_i}{\left(\frac{k_2^2}{k_1^2}\right)} \cdot FM_2 + k_1^2 \cdot FM_i \cdot FM_1 \quad (1)$$

Each row in the formula represents the number of parameters at a level in the pyramid convolution layer. If the pyramid convolution layers of each layer output the same number of feature graphs, then the number of parameters of the pyramid convolution layer is evenly distributed along each convolution layer. According to the above formula, regardless of the number of pyramid convolution layers, the number of parameters is similar to the standard convolution of single-core k_1^2 when the size of the nucleus space increases from k_1^2 to k_n^2 .

Compared with standard convolution, the pyramid convolution layer can expand the acceptance domain of the kernel at no extra cost, and it is also suitable for different types of kernels in parallel, with different spatial resolutions and depths. Therefore, the pyramid convolution layer dissolves the input at multiple scales to capture more detailed information. These different types of pyramid convolution layer kernels bring complementary information and help to improve the recognition performance of the network. Kernels with smaller acceptance fields can focus on details, capturing information about smaller objects and/or parts of objects, while increasing the size of the kernel provides more reliable details.

Construction of soybean pest image recognition model

This experiment employs the VGG16 model for transfer learning due to the simplicity of the VGG16 network design [14]. Figure 3 shows the architecture of the model based on transfer learning and pyramidal convolution. VGG16 is a model of a traditional convolutional neural network frequently utilized in the field of image categorization [15]. To extract fine details from the image, VGG16 is stacked with a mixture of 33 filters. With 3.2 million images, ImageNet is an excellent image classification resource that is widely utilized in the fields of target recognition and image classification [16]. In this experiment, the parameters learned from VGG16 training on ImageNet were transferred to a soybean pest recognition model, which was capable of performing small data set classification for soybean pests.

VGG16's convolutional layer offers a high capacity for feature extraction. The convolutional layer's front end is capable of extracting the image's general characteristics [17]. With increasing the number of convolutional layers, the derived features become more precise and are capable of extracting subtle information in the image. To increase disease detection accuracy while conserving limited computing resources, this study modified the VGG16 network structure by reserving the original convolutional layer for disease feature extraction and then removing the original fully connected layer and replacing it with a new pyramid convolution. Its structure is constructed of pyramidal convolution layers at various kernel levels (9, 77, 55, 33), each with a variable kernel depth. To adapt to the task of identifying soybean diseases in a variety of sizes. The convolution layer's parameters are initialised by pre-training them on ImageNet, a large image classification dataset, while the complete connection layer's parameters are initialised randomly. The pyramid convolution block (PyConvBlock) begins with four kernel size levels and gradually decreases the convolution depth of each level as the level feature map's space diminishes. Finally, the network is reduced to a single pyramid convolution block, which forms the standard convolution layer. Between each convolutional layer, there are several Bach Normalization (BN) and rectified Linear Unit (ReLU) activation functions. Normalization speeds up network training by lowering the mean square deviation and facilitating gradient propagation. It aids in parameter updating during back propagation and aids the network in overcoming the gradient disappearance problem. The dynamic learning rate adjustment technique successfully improves the detection accuracy of the model by fine-tuning the convolutional layer and full-connection layer parameters with varied learning rates.

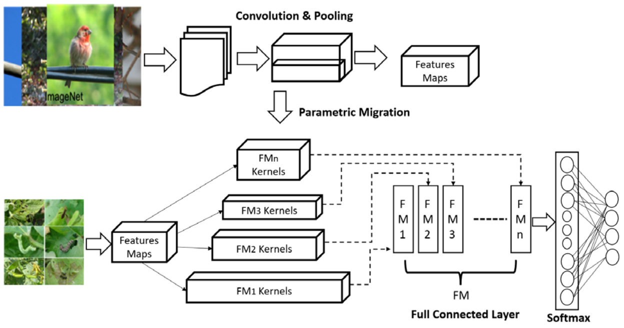


Figure 3. Architecture of the model based on transfer learning and pyramidal convolution

Since the soybean disease data set was manually constructed and the composition of data samples was defective, the Dropout layer was added after the full connection layer [18], and some units in the neural layer were randomly selected and temporarily hidden in a cycle, and then the neural network training and optimization process was carried out in this cycle. In the next cycle, additional neurons are hidden until the training is over. It can effectively alleviate the occurrence of overfitting and achieve regularization effect to a certain extent.

3.0 RESULTS AND DISCUSSION

Data Preprocessing

The soybean pest dataset collected in this experiment is small, and it is difficult for the soybean pest identification model to understand the real distribution of the dataset, which is prone to overfitting phenomenon. Therefore, the problem of small amount of data should be solved by data expansion methods, and data augmentation is one of the commonly used data expansion methods in deep learning. In this paper, we use six data augmentation methods, including shift, flip, rotation, scale, noise, and brightness, to expand the soybean pest dataset, expanding each pest image by 10 times and increasing the sample images from 617 to 6170. The number of sample images increased from 617 to 6170, as shown in Table 1. In addition, since part of the dataset in this paper originated from the network, the size of the image dataset before and after expansion was uneven. In order to meet the input requirements of the VGG16 network, the soybean pest images were uniformly adjusted to 224 pixels × 224 pixels and the image formats were all jpg format. In order to meet the input requirements of the VGG16 network, the soybean pest images were uniformly resized to 224 pixels × 224 pixels and the images were in jpg format. 80% of the images were used as the training data set and the remaining 20% as the test dataset.

Table 1 Soya Bean Pest Dataset

Pest species	**Number of images**	
	The original	**After data enhancement**
Soybean Green Worm	103	1030
Soybean moth insect	101	1010
Mites	100	1000
Moth	100	1000
Stink bug	102	1020
Bean pod bug	101	1010

Experimental Setup Environment

The soybean pest identification model was trained and tested using a Windows 10 64-bit operating system. Table 2 below shows the hardware environment for the experiments.

Table 2 Hardware & Software Environment

Hardware / Software	**Specifications**
The operating system	Windows 10
CPU	3.8GHz
GPU	GTX3090
Memory	32G
Deep Learning Framework	Pytorch

SGD (stochastic gradient descent method), batch size of 32, initial learning rate of 5 × 10^{-3}, impulse (Momentum) of 0.9, decay (Decay) of 5 × 10^{-4}, cross-entropy loss function.

Evaluation

This experiment uses accuracy rate as an evaluation index. The accuracy rate is a response to the result of a model's judgment of the actual sample data compared with the sample labels. The higher the accuracy rate, the more effective the model is. The general formula of accuracy rate is:

$$ACC = \frac{TN+TP}{TN+FN+FP+TP} \qquad (2)$$

where ACC is the number of accuracy rate ,TP is the number of correctly detected bridge surface disease pictures, TN is the number of correctly detected non-disease pictures, FP is the number of non-disease pictures that are considered as disease pictures number, and FN is the number of diseased pictures that are considered as non-diseased pictures. In addition, the loss of the experiment is another aspect to judge the performance of the model. the faster the value of the loss decreases, the faster the model converges; the smaller the value of the loss, the more robust the model is and the better the performance.

Dynamic Learning Rate Adjustment Strategy

The learning rate is generally referred to as a crucial hyperparameter in deep learning and supervised learning since it impacts whether or not the model can converge to the optimal state and how quickly it can converge. When the learning rate is set too low, the model will converge slowly and may fall into a local optimal point, making it difficult to reach the model objective function's optimal point; when the learning rate is set too high, the model will converge quickly but may oscillate repeatedly before reaching the optimal point, preventing it from reaching the optimal point. As a result, selecting an adequate learning rate is critical for a model's convergence.

The current work employs a dynamic learning rate adjustment technique that gradually decreases the learning rate as the model's iterations grow [19]. This method enables the model to converge rapidly during the first training phase by skipping some local optimal points, and as the number of iterations increases, the learning rate slows, the model's learnt features increase, and the model's recognition accuracy increases. The following is the strategy for dynamically adjusting the learning rate:

$$LR_{schedule} = \frac{LR}{\left(1+\frac{10\times epoch}{n\_epoch}\right)0.75} \qquad (3)$$

Where LR is the learning rate size, $epoc$h is the number of current training rounds, and n_epoch is the total number of rounds set for the experiment. During the training, the initial learning rate LR and the total number of rounds n_epoch are constant, and the current number of training rounds $ee\,poc$h will gradually increase, resulting in a training learning rate LR_sce $dual$ decreases gradually and slowly as the number of training rounds increases. After dynamic learning rate adjustment, the accuracy of the model changes as shown in Table 3, where VGG16_py_da and ResNet50_py_da denote the VGG16 network model based on dynamic learning rate adjustment. From the experimental results shown in Table 3, it can be found that the accuracy of the models with the dynamic learning rate adjustment strategy is higher than the accuracy without the strategy adjustment.

Table 3 Model accuracy under the dynamic learning rate adjustment strategy

Model	**Accuracy**	**Various types of Soybean pests identification accuracy/%**					
		Green Worm	**moth insect**	**Mites**	**Moth**	**Stink bug**	**Bean pod bug**
VGG16	83.04	80.33	81.34	83.6	80.23	85.67	87.1
VGG16_py_da	96.06	94.4	96.78	97.56	93.7	96.8	97.17
Resnet50	93.2	90.6	92.2	93.21	92.78	95.52	94.89
Resnet50_py_da	95.6	95.8	95.9	96.9	93.6	95.3	96.67

Fine-Tuning

Fine-tuning is a common technique used in deep learning-based transfer learning approaches. [20] Typically, the fine-tuning technique refers to a model trained in one domain that can perform excellently in a new domain by updating and reusing that model parameters and model architecture. In this experiment, all convolutional layers of the VGG16 network model are migrated and used as feature extractors, and the initial learning rate of the convolutional layer is set to 5 × 10^{-3} and the initial learning rate of the fully connected layer is set to 10 times the learning rate of the convolutional layer. This setting allows the network to update the parameters of the pyramidal convolutional layer by back propagation during training, so that the pyramidal convolutional layer can be adapted to new feature extraction tasks; the fully-connected layer uses a larger learning rate, so that the fully-connected layer can converge quickly. Finally, a dynamic learning rate adjustment strategy is used to make the learning rate decrease gradually with the increase of training rounds, so that the model learns more features during the training process and improves the model performance.

In this paper, two sets of experiments are conducted to verify the experimental performance of fine-tuning. The first one is to freeze the convolutional part of the pre-trained VGG16 network and train only the fully connected layer with 10 times the initial learning rate; the second one is to train the network according to the above experimental setup, and both sets of experiments use the dynamically adjusted learning rate at the same time, and the experimental results are shown in Table 4. The results are shown in Table 4. As can be seen from the table, the accuracy of the model after fine-tuning the pyramidal convolutional layer is 1.16% higher than that of the VGG16 model without the fine-tuning technique, indicating that the fine-tuned model has a more powerful learning capability and can adapt to new tasks and learn more image features.

Table 4 Comparison of experimental results of model before and after tuning

Method	**VGG16 pre-trained model parameters**	**Training rounds**	**soybean pests identification accuracy (%)**
After fine-tuning	Effective	200	97.22
Without fine-tuning	Effective	200	96.06

Effect Of Freezing All Convolutional Layers On Experiments

The improved model, denoted as VGG16_P, is further explored by unfreezing some of the convolutional layers on top of VGG16_M. Since the introduction of global averaging pooling can significantly reduce the training time and the size of the model when all the convolutional layers are frozen, but there is undertraining, we can unfreeze some of the convolutional layers based on this top-level design and observe the experimental results. The experimental results for VGG16_M and VGG16_P are shown in Table 5.

Table 5 Influence of freezing all convolutional layers on the experiment

Comparison of Experimental Results **Model**	**Whether to unfreeze the Convolution layer**	**Accuracy (%)**	**Loss**	**Model size (MB)**	**Speed (MS)**
VGG16_M	Yes	97.67	0.048	175	67
	No	95.22	0.1	157	63
VGG16_P	Yes	98.78	0.027	90.2	51
	No	91.13	0.41	75.4	34

The method in this paper is based on transfer learning and pyramid convolution model architecture, using dynamic learning rate for fine-tuning, which includes VGG16_py_da and VGG16_P. Based on the division of the dataset, the trained network model is combined with other mainstream models to conduct comparative experimental validation on the validation set, and the results are shown in Table 6 & 7. As can be seen from the tables 6 & 7, experimental results are better than those of all mainstream models, including a 15.19% improvement in accuracy over the standard VGG16 model and a 5.03% improvement in accuracy over the standard ResNet50 model. In addition, with a sufficient number of validation sets, our model achieves higher recognition accuracy than other models on each disease category with high accuracy. In our experiments, the 99% recognition accuracy of the slash moth pest is achieved because of the powerful feature extraction capability of the migrated VGG16 convolutional layer, coupled with the single simple disease feature of the slash moth dataset compared to the other two disease features. The accuracy of the standard VGG19 decreases compared with that of the standard VGG16, indicating that the deeper the number of layers of the unoptimized standard model network, the more noise the model learns and the more likely it is to be overfitted, which affects the final judgment of the model. The validation accuracies of the standard ResNet18 and ResNet50 models reached 95.6% and 93.2%, indicating that the use of transfer learning can effectively extract the characteristics of soybean pests. The accuracy of the VGG16 model using the dynamic learning rate adjustment strategy improved by 13.02% compared with the original VGG16 model, which indicates that the dynamic learning rate adjustment used in this study enables the network to learn more features and effectively improves the accuracy of the model.

Table 6 Comparison of fine-tuning results

Model	**Accuracy**	**Various types of soybean pests identification accuracy (%)**					
		Green Worm	**moth insect**	**Mites**	**Moth**	**Stink bug**	**Bean pod bug**
VGG16_py_da	96.06	94.4	96.78	97.56	93.7	96.8	97.17
Method of this article	98.23	96.4	97.78	98.12	98.4	99.56	99.16

Table 6 Experimental result of mainstream models on validation sets

Model	**Accuracy**	**Various types of soybean pests identification accuracy (%)**					
		Green Worm	**moth insect**	**Mites**	**Moth**	**Stink bug**	**Bean pod bug**
VGG16	83.04	80.33	81.34	83.6	80.23	85.67	87.1
VGG19	82.2	78.9	82.45	81.3	79.89	84.65	86.23
VGG16_py_da	96.06	94.4	96.78	97.56	93.7	96.8	97.17
Method of this article	98.23	96.4	97.78	98.12	98.4	99.56	99.16
Resnet50	93.2	90.6	92.2	93.21	92.78	95.52	94.89
Resnet18	95.6	95.8	95.9	96.9	93.6	95.3	96.67

It is known from the above comparison experiments that the fine-tuned parameters of the pyramidal convolutional layer and the dynamic learning rate adjustment used in this paper can effectively suppress the overfitting of the model and make the model adaptable to new tasks and can learn soybean pest characteristics with high accuracy.

4.0 CONCLUSION

The purpose of this study is to develop a method for identifying soybean pests using convolutional neural networks (CNN) and transfer learning. The transfer learning approach is implemented by extracting features from the convolutional layer of the VGG16 model. The migration technique was used to transfer the parameters to the golden pose convolution for adjustment, initialize the VGG16 model with pre-trained parameters, and apply the dynamic adjustment strategy for learning rate. The parameters of the convolutional and fully connected layers in the training phase were fine-tuned to make them appropriate to the soybean pest detection challenge.

When all convolutional layers are frozen, the recognition effect of the model with transfer learning is related to the top layer's distinct structure. As the number of neurons in the model rises, the recognition accuracy improves and the loss value lowers. However, as the number of neurons increases, the model's size increases as well, resulting in decreased mobility. By unfreezing some of the convolutional layers, the problem of insufficient training due to too few neurons in the top layer can be resolved, and the average recognition accuracy on the soybean pest dataset is 98.23 with a model size of only 75.4M, which is more portable and takes less time than the original model.

As a conclusion, the suggested model outperforms other popular models in terms of accuracy, and the validation set outperforms the original VGG16 model by 15.19%. The model is quite accurate in identifying soybean pests.

Acknowledgement

This work was supported by Heilongjiang Provincial Natural Science Foundation of China: LH2020F039

References

[1] Kasinathan, T., & Uyyala, S. R. 2021. Machine learning ensemble with image processing for pest identification and classification in field crops. *Neural Computing and Applications*, *33*(13): 7491–7504. https://doi.org/10.1007/s00521-020-05497-z

[2] Zhang, S. W., Shang, Y. J., & Wang, L. (2015). Plant disease recognition based on plant leaf image. *Journal of Animal and Plant Sciences*, *25*(3): 42–45.

[3] Ferentinos, K. P. 2018. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, *145*(January): 311–318. https://doi.org/10.1016/j.compag.2018.01.009

[4] Waheed, A., Goyal, M., Gupta, D., Khanna, A., Hassanien, A. E., & Pandey, H. M. 2020. An optimized dense convolutional neural network model for disease recognition and classification in corn leaf. *Computers and Electronics in Agriculture*, *175*: 105456. DOI: https://doi.org/10.1016/j.compag.2020.105456

[5] Mohanty, S. P., Hughes, D. P., & Salathé, M. 2016. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, *7*(September): 1–10. https://doi.org/10.3389/fpls.2016.01419

[6] Liu, Y. Z., Shi, K. M., Li, Z. X., Ding, G. F., & Zou, Y. S. 2021. Transfer learning method for bearing fault diagnosis based on fully convolutional conditional Wasserstein adversarial Networks. *Measurement: Journal of the International Measurement Confederation*, *180*(May): 109553. https://doi.org/10.1016/j.measurement.2021.109553

[7] Krishnamoorthy, N., Narasimha Prasad, L. V., Pavan Kumar, C. S., Subedi, B., Abraha, H. B., & Sathishkumar, V. E. 2021. Rice leaf diseases prediction using deep neural networks with transfer learning. *Environmental Research*, *198*(May): 111275. https://doi.org/10.1016/j.envres.2021.111275

[8] Krizhevsky, B. A., Sutskever, I., & Hinton, G. E. 2012. CNN Practical Training. *Communications of the ACM*, *60*(6): 84–90.

[9] Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. 2014. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, *4*(January): 3320–3328.

[10] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2010). ImageNet: Constructing a large-scale image database. *Journal of Vision*, *9*(8): 1037–1037. https://doi.org/10.1167/9.8.1037

[11] Guo, L., Lei, Y., Xing, S., Yan, T., & Li, N. 2019. Deep Convolutional Transfer Learning Network: A New Method for Intelligent Fault Diagnosis of Machines with Unlabeled Data. *IEEE Transactions on Industrial Electronics*, *66*(9): 7316–7325. https://doi.org/10.1109/TIE.2018.2877090

[12] Lyu, Q., Xia, D., Liu, Y., Yang, X., & Li, R. 2021. Pyramidal convolution attention generative adversarial network with data augmentation for image denoising. *Soft Computing*, *25*(14): 9273–9284. https://doi.org/10.1007/s00500-021-05870-7

[13] Liang, Z., Shao, J., Zhang, D., & Gao, L. 2020. Traffic sign detection and recognition based on pyramidal convolutional networks. *Neural Computing and Applications*, *32*(11): 6533–6543. https://doi.org/10.1007/s00521-019-04086-z

[14] Xiao, Y., Huang, X., & Liu, K. 2021. Model Transferability from ImageNet to Lithography Hotspot Detection. *Journal of Electronic Testing: Theory and Applications (JETTA)*, *37*(1): 141–149. https://doi.org/10.1007/s10836-021-05925-5

[15] Pardede Jasman, Benhard, S., Saiful, A., & KhodraMasayu Leylia. 202). Implementation of Transfer Learning Using VGG16 on Fruit Ripeness Detection. *International Journal of Intelligent Systems and Applications*, *13*(2): 52–61. https://doi.org/10.5815/ijisa.2021.02.04

[16] Moon, J., Hossain, M. B., & Chon, K. H. 2021. AR and ARMA model order selection for time-series modeling with ImageNet classification. *Signal Processing*, *183*: 108026. https://doi.org/10.1016/j.sigpro.2021.108026

[17] Moraes, G. S. de O., Ferreira, M. de A., Guim, A., Tabosa, J. N., Chagas, J. C. C., & Almeira, M. de P. 2019. A transfer learning method with deep residual network for pediatric pneumonia diagnosis. *Livestock Science*, *221*: 133–138.

[18] Srivastava, N., Hinton, G., Krizhevsky, A., & Ruslan, I. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Physics Letters B*, *299*(3–4): 345–350. https://doi.org/10.1016/0370-2693(93)90272-J

[19] Ganin, Y., Ajakan, H., Hugo, L., Laviolette, F., & Lempitsky, V. 2016. Domain-adversarial training of neural networks. *Otolaryngology - Head and Neck Surgery*, *133*(4): 562–568. https://doi.org/10.1016/j.otohns.2005.05.012

[20] Chebet ET, Li Y, Sam N, Liu Y. 2019. A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture, 161: 272–9.