


Article

Deep Convolutional Neural Networks for Tea Tree Pest Recognition and Diagnosis

Jing Chen ^{1,2} , Qi Liu ^{1,*} and Lingwang Gao ^{2,*}

¹ Department of Plant Pathology, College of Agronomy, Xinjiang Agricultural University, Urumqi 830052, China; cj-yx2004@163.com

² Department of Plant Biosafety, College of Plant Protection, China Agricultural University, Beijing 100193, China

* Correspondence: liuqi@xjau.edu.cn (Q.L.); lwgao@cau.edu.cn (L.G.)

Abstract: Due to the benefits of convolutional neural networks (CNNs) in image classification, they have been extensively used in the computerized classification and focus of crop pests. The intention of the current find out about is to advance a deep convolutional neural network to mechanically identify 14 species of tea pests that possess symmetry properties. (1) As there are not enough tea pests images in the network to train the deep convolutional neural network, we proposes to classify tea pests images by fine-tuning the VGGNET-16 deep convolutional neural network. (2) Through comparison with traditional machine learning algorithms Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP), the performance of our method is evaluated (3) The three methods can identify tea tree pests well: the proposed convolutional neural network classification has accuracy up to 97.75%, while MLP and SVM have accuracies of 76.07% and 68.81%, respectively. Our proposed method performs the best of the assessed recognition algorithms. The experimental results also show that the fine-tuning method is a very powerful and efficient tool for small datasets in practical problems.



Citation: Chen, J.; Liu, Q.; Gao, L. Deep Convolutional Neural Networks for Tea Tree Pest Recognition and Diagnosis. *Symmetry* **2021**, *13*, 2140. <https://doi.org/10.3390/sym13112140>

Academic Editor: José Carlos R. Alcantud

Received: 5 October 2021

Accepted: 27 October 2021

Published: 10 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: CNN; fine-tune; SVM; MLP; tea pests classification

1. Introduction

In 2016, China is the largest producer of tea in the world, with a total output of tea plant production exceeding 2.4 billion tons [1]. Tea trees (*Camellia sinensis*) are perennial evergreen plants, and they are widely planted in south China, where the warm and humid climate is particularly suitable for the growth of tea trees. Tea trees are especially suitable for insects life because of their clumpy shrubbery, large-scale planting, and climates within which they grow. Tea pests can cause 11–55% loss in yield if left unchecked [2]. However, the abuse of insecticides can not only lead to resistance in some pests but also pollute the environment. Some pesticides can also concentrate and accumulate in the food chain, endangering human health. Timely diagnosis of insect pests and the adoption of efficient, targeted, environmentally friendly prevention and control measures have become the main challenges tea gardens are facing today.

Usually, tea pests diagnosis is based on appearance characteristics of insects. Due to lack of pests biology and identification, farmers are unable to timely diagnose insect pest types, thus causing serious damage to the production process. Furthermore, the availability of agricultural experts is limited [3]. Therefore, an automatic identification system for tea pests can ensure that agricultural producers accurately assess tea pests and thus determine the most economical and effective control methods to employ. This allows farmers to reduce labor costs and losses while ensuring the quality of tea. With the rapid development of agricultural information technology, computer image processing, and pattern recognition technology, image-based classification has been used to identify crop pests [4]. At present, the general methods of image-based crop pest identification are image acquisition, feature extraction, and classifier selection. Insect pest images can be

obtained by photographing insect specimens. There are many online datasets containing such images, such as ImageNet [5] and Plant Village [6]. The extracted features are mainly based on global features (color, shape, and texture) and local features (scale-invariant feature transform (SIFT), speeded-up robust features (SURF) and histogram of oriented gradient (HOG). These features are easy to extract and require little computation and therefore are most widely used. Finally, the extracted features are led to the classifier (such as SVM, MLP, and KNN) for classification and recognition. The main disadvantage of the above method is that it requires manual extraction in advance, and the selection of features directly affects the final recognition accuracy.

In recent years, with the continuous expansion and improvement of the Internet and hardware performance, deep learning theory has developed rapidly. Due to its superior performance in feature extraction, model generalization, and fitting, it has attracted considerable attention from researchers. CNN is widely used in image recognition. The main characteristic of CNN is that it uses a stacked layer of non-linear processing units to extract the features of input images autonomously, resulting in end-to-end feature extraction and classification [7]. More and more agricultural researchers have begun to use CNN in research and applications [8–11], which has further enhanced the accuracy of classification.

AlexNet [12] gained the ImageNet Large Scale Visual Recognition Challenge in 2012, and the cnn has been broadly used for the reason that then. Currently, representative CNN models include VGGNet [13], GoogleNet [14], and ResNet [15]. The CNN is composed of repeatedly stacked convolutional layers and a pooling layer, as well as a full connection layer for classification. Local receptive fields, shared weights, and pooling are the main characteristics, and the network performs significantly higher than usual shallow machine learning algorithms with handcrafted features.

Deep CNN require a lot of training data because of the big quantity of parameters to be learned. The size of the training dataset determines the identification accuracy of the network, which is liable to overfitting when a small training dataset is used. It also affects the variance and bias during modeling. Over-fitting is that the model cannot be well generalized from the original data set to a new data set, which is model error in statistics. Although there are many online datasets with over a million images, like ImageNet, but there are insufficient data in some specific fields, such as medicine and agriculture. Even if there is a large amount of data, there are not enough labeled training data. Huge manpower and financial resources are needed to create a training dataset from such a large amount of data. Furthermore, in many machine learning methods, the training data and the test data must conform to the assumption of independent and identically distributed [16]. Thankfully, transfer learning solves the small dataset problem [17,18]. Therefore, Transfer learning is a machine learning method using the learned better features in the source domain to train the related task or other domain; in other words, transfer learning uses the parameters of the trained networks to improve the predictive function of the target networks [19]. Meanwhile, the target networks do not need to be trained from scratch. Instead, we can use the trained parameters of the source networks to train the target networks, which can significantly reduce training time. Yosinski [20] used transfer learning on CNN and proved the validity of this method; the results indicated that LeNet, AlexNet, VGGNET, Inception, and ResNet are suitable for transfer learning. Many researchers have studied transfer learning [21–25]. There are two main ways to implement transfer learning in convolutional neural networks.

1. Take CNN (also recognized as the pre-training model) as the feature extractor, remove the last layer of the network, take the rest of CNN as the feature extractor to get the vectors, and then use these feature vectors to train a classifier (such as an SVM).
2. Fine-tune the weight parameters of the target CNN on the new dataset. In the fine-tuning process, all the layer parameters of the model can be adjusted, or the parameters of the first several layers of the model can be fixed. Then we train the parameters of the last layers or just the softmax layer. We fine-tune the last layer because the features extracted from the first few layers of the CNN are universal

features, while the features extracted from the last few layers of the model are related to datasets and classification tasks. Therefore, only the latter layers can be adjusted in the new datasets, and the training time can be greatly shortened.

In this study, we adopt a classical CNN (VGGNET-16) to extract features and classify tea pests images. The network is pre-trained on the ImageNet database and fine-tuned with our tea pests images database. We compare our method against Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP) algorithms. The dense scale-invariant feature transform (DSIFT)-based bag of visual words (BoVW) model is used to obtain image features for the two classifiers. Liblinear classifier is used in order to get better accuracy for large-scale data classification. And the one vs. rest implementation method is adopted, which the advantage is the algorithm runs more efficient than the SVM. MLP classifier is a kind of artificial neural network. This article mainly uses a three-layer perceptron structure. As the extracted features are 1000-dimensional vectors, the input layer contains 1000 nodes, the hidden layer contains 100 nodes, and the output layer contains seven nodes, which are the number of tea leaf diseases, and 14 nodes are the number of pests.

2. Materials and Methods

2.1. Insect Image Dataset

Because there are not enough tea pests images on the Internet, we collected our own pest photos in a field tea garden; this also made the network model to be more easily applied to actual agricultural environment. Images of tea pests were taken 20 cm directly above the pest and captured in the auto-focus mode at a resolution of 4000×3000 pixels using a Cannon PowerShot G12 camera in real tea plantation environment in the Hubei province of China. A total of 2562 tea pests images were obtained: these included 14 different insects which possess symmetry property, as identified by an entomologist (Figure 1). All the images in this manuscript were resized to 256×256 pixels, and the 224 pixels sub-blocks of the image were randomly extracted as CNN input. In order to enhance the classifier's generalization potential a, we enlarged the size of the dataset. Three methods have been used to convert the image and improve classification (Figure 2). Finally, there have been 13,487 images in the database.

The data are split into training dataset, validation dataset, and test dataset, with percentage ratios of 80%, 10%, and 10%, respectively. The validation dataset determines if the model is overfitting, and the test dataset is used for prediction and evaluation of the models. Table 1 suggests the number of images for classes used in every datasets for the pests classification.



Figure 1. Cont.



Figure 1. Example tea tree pest images. (1) *Homona coffearia* (Meyrick); (2) *Eterusia aedea* (Linnaeus); (3) *Euproctis pseudoconspersa* (Strand); (4) *Arctornis alba* (Bremer); (5) *Amata germana* (Felder); (6) *Rikiosatoa vandervoordeni* (Prout); (7) *Culcula panterinaria* (Bremer et Gray); (8) *Scopula subpunctaria* (Herrich-Schaeffer); (9) *Ricania sublimbata* (Jacobi); (10) *Ricania speculum* (Walker); (11) *Euricania ocellus* (Walker); (12) *Spilosoma menthastri* (Esper); (13) *Spilarctia subcarnea* (Walker); and (14) *Ceratonoros transiens* (Walker).

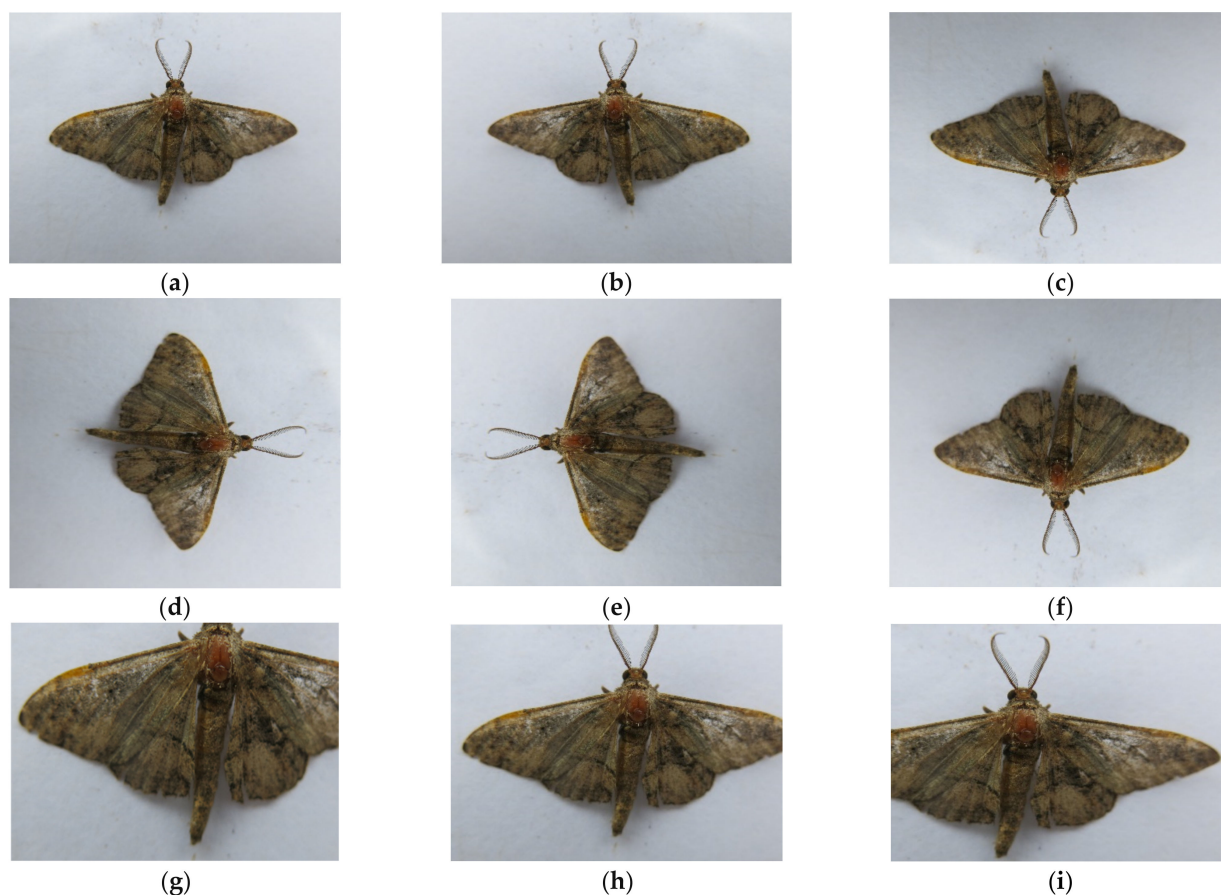


Figure 2. Examples of data augmentation used for *R. vandervoordeni* insect images. (a) Initial; (b) flip horizontal; (c) flip vertical; (d) right-rotated 90°; (e) left-rotated 90°; (f) rotated 180°; and (g–i) randomly cropped.

Table 1. Dataset of 14 insects used in this study.

Class	Number of Images from the Dataset Used for Train	Number of Images from the Dataset Used for Validate	Number of Images from the Dataset Used for Test
(1) <i>Ricania speculum</i> (Walker)	768	96	96
(2) <i>Euricania ocellus</i> (Walker)	730	91	91
(3) <i>Ricania sublimbata</i> (Jacobi)	722	90	90
(4) <i>Ceratonoros transiens</i> (Walker)	768	96	96
(5) <i>Spilarctia subcarnea</i> (Walker)	804	101	100
(6) <i>Homona coffearia</i> (Meyrick)	710	89	89
(7) <i>Eterusia aedea</i> (Linnaeus)	774	97	97
(8) <i>Culcula panterinaria</i> (Bremer et Gray)	770	96	96
(9) <i>Euproctis pseudoconspersa</i> (Strand)	802	100	100
(10) <i>Arctornis alba</i> (Bremer)	760	95	95
(11) <i>Rikiosatoa vandervoordeni</i> (Prout)	843	105	105
(12) <i>Scopula subpunctaria</i> (Herrich-Schaeffer)	794	99	99
(13) <i>Amata germana</i> (Felder)	790	99	99
(14) <i>Spilosoma menthastri</i> (Esper)	756	95	94
Total	10,791	1349	1347

2.2. Performance Measurements

Accuracy and mean class accuracy (MCA) indices were used to evaluate algorithm performances. CCR_k is first defined as the correct classification rate for category k , as decided with Equation (1):

$$CCR_k = \frac{C_k}{N_k} \quad (1)$$

where C_k is the number of correct identifications for category k and N_k is the total number of elements in category k . Classification accuracy is t decided with Equation (2):

$$Accuracy = \frac{\sum_k CCR_k \cdot N_k}{\sum_k N_k} \quad (2)$$

MCA is decided with Equation (3):

$$MCA = \frac{1}{k} \sum_k CCR_k \quad (3)$$

where k is the total number of classes.

2.3. Pre-Trained Convolutional Neural Networks

We use the VGGNet architecture, which is developed by the Visual Geometry Group of the University of Oxford and researchers from Google DeepMind. This architecture won second place in the classification project and first place in the positioning project of the 2014 ILSVRC competition held by ImageNet. VGGNet constructs six kinds of CNN of different structures: two networks with 11 layers, one network with 13 layers, two networks with 16 layers, and one network with 19 layers. These are realized by repeatedly stacking small convolution kernels of 3×3 and maximum pooling layer of 2×2 . The purpose of using small convolution kernels is not only to ensure the visual field of perception but also to reduce the parameters of the convolutional layer. Although the number of network layers increases from 11 to 19, increasing the network depth will not bring obvious parameter expansion; instead, higher accuracy will be achieved. Compared with the previous CNN structure, the error rate of VGGNet is reduced. The network architecture is shown in Figure 3. Layer parameters for the VGGNet-16 are shown in Table 2.

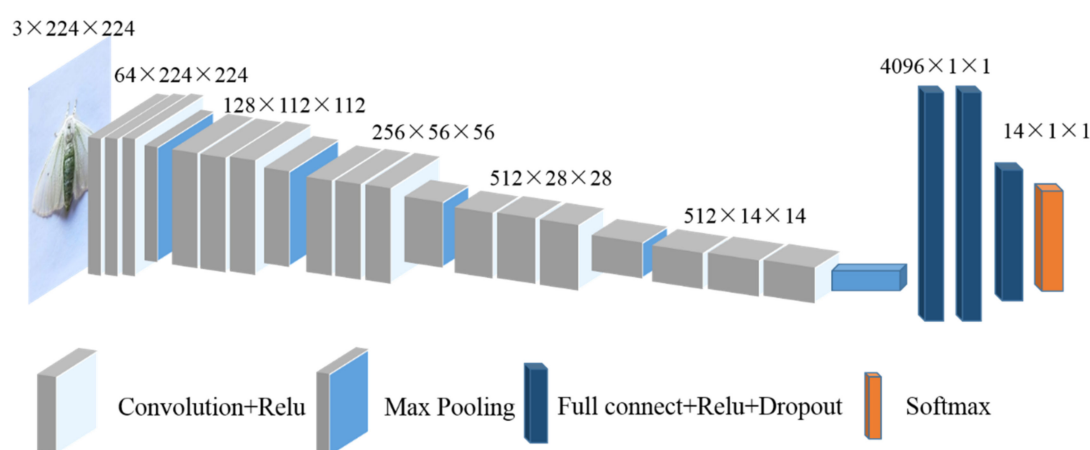


Figure 3. Architecture of the VGGNet-16 network.

Table 2. Layer parameters for VGGNet-16.

Layer	Parameter	Output
Input	$224 \times 224 \times 3$	-
Conv1-1	64 convolution filters (3×3), 1 stride, 1 pad	$224 \times 224 \times 64$
Conv1-2	64 convolution filters (3×3), 1 stride, 1 pad	$224 \times 224 \times 64$
Max pool 1	Max pooling (2×2), 2 stride	$112 \times 112 \times 64$
Conv2-1	128 convolution filters (3×3), 1 stride, 1 pad	$112 \times 112 \times 128$
Conv2-2	128 convolution filters (3×3), 1 stride, 1 pad	$112 \times 112 \times 128$
Max pool 2	Max pooling (2×2), 2 stride	$56 \times 56 \times 128$
Conv3-1	256 convolution filters (3×3), 1 stride, 1 pad	$56 \times 56 \times 256$
Conv3-2	256 convolution filters (3×3), 1 stride, 1 pad	$56 \times 56 \times 256$
Conv3-3	256 convolution filters (3×3), 1 stride, 1 pad	$56 \times 56 \times 256$
Max pool 3	Max pooling (2×2), 2 stride	$28 \times 28 \times 256$
Conv4-1	512 convolution filters (3×3), 1 stride, 1 pad	$28 \times 28 \times 512$
Conv4-2	512 convolution filters (3×3), 1 stride, 1 pad	$28 \times 28 \times 512$
Conv4-3	512 convolution filters (3×3), 1 stride, 1 pad	$28 \times 28 \times 512$
Max pool 4	Max pooling (2×2), 2 stride	$14 \times 14 \times 512$
Conv5-1	512 convolution filters (3×3), 1 stride, 1 pad	$14 \times 14 \times 512$
Conv5-2	512 convolution filters (3×3), 1 stride, 1 pad	$14 \times 14 \times 512$
Conv5-3	512 convolution filters (3×3), 1 stride, 1 pad	$14 \times 14 \times 512$
Max pool 5	Max pooling (2×2), 2 stride	$7 \times 7 \times 512$
Full Connect-6	$4096 \times 1 \times 1$, 1 stride	4069
Full Connect-7	$4096 \times 1 \times 1$, 1 stride	4069
Full Connect-8	$14 \times 1 \times 1$, 1 stride	14
Output	-	1

Caffe's Model Zoo provides a large number of pre-training models that can be used for transfer learning [26]. We downloaded the VGGNet-16 model from Caffe's Model Zoo and pre-trained it on the ILSVRC 2012 (ImageNet) dataset [5].

2.4. Fine-Tune the CNN Model

Our target dataset is not very different from the ImageNet dataset, as we only re-trained the last fully connected layers. If we retrained the whole network end-to-end, then our training dataset would be prone to overfitting. The number of neurons in the last softmax layer is replaced with the new class number (i.e., 14 neurons). As we only trained the softmax layer of the network, the learning rate of the top layer's parameters (the convolutional layer) is set to 0, that is, the parameters were frozen, and there is no need to train those layers. The parameters of the pre-trained VGGNet-16 were used as initialization parameters for the top layers of the target networks, and the parameters of the softmax layer were initialized with random values. The fine-tuning network is trained us-

ing stochastic gradient descent with a batch size of 32. As the parameters of the pre-trained model were randomly initialized and trained from scratch, the learning rate, step size, and iterations were relatively large. Moreover, the parameters of the pre-trained model were relatively good and did not need to be updated very quickly during the fine-tuning. Therefore, the learning rate during fine-tuning is reduced accordingly. We changed the basic learning rate from 0.01 to 0.001. The momentum is 0.9, and the weight decay is 0.0005. Deep learning library Caffe is used to implement our experiments.

2.5. Dense SIFT-Based BoVW Model

The Bag of Word (BoW) model is initially mainly used for text classification and retrieval technology. Due to the profitable utility of the BoW model in text classification and retrieval and the robustness of local features of images, Csurka [27] introduced the BoW model into the field of computer vision in 2004. The core idea of the BoVW is that the image is handled as a document, with the characteristics of the image (usually the local characteristics) being treated as words; however, the difference is that the images are not ready-made words that need to be independent of each other. Similar characteristics can be seen as a word, and the image can be described as an unordered set of local characteristics (visual words). Although local features (such as SIFT descriptor) can describe an image, each SIFT descriptor is a vector of 128 dimensions, and an image contains thousands of SIFT descriptor. The computation is extremely heavy, and thus the vector data will be clustered; a cluster can represent a word.

Image classification with the BoVW algorithm mainly includes the following steps.

1. Extraction and description the image features: Detect interest points through random sampling and obtain local features of the image. Common local features descriptors include the SIFT, SURF, and HOG. In our study, we employ the DSIFT descriptor. The main difference between the DSIFT and SIFT is their different sampling methods. SIFT descriptor detects and screens the feature point by building scale space. Conversely, the DSIFT descriptor divides the image into rectangular areas of the same size and then uses a fixed-size window to sample from left to right and from top to bottom with a certain step length to extract the SIFT features. Finally, each feature is represented by a 128-dimension vector. The features extracted by this method are evenly distributed with the same specifications, and have stable illumination, translation, and rotation.
2. Construction the visual vocabulary: K-means algorithm is used to cluster the local feature vectors of all sample images. The size of the visual vocabulary is N , if there are N cluster centers (defined as a visual word). In this manuscript, the size of the visual vocabulary is defined as 1000.
3. Image representation: We measure distance each local features to the visual word of the vocabulary and map the local features to the nearest visual word. We then compute the occurrences of each visual word in the image, which becomes a N -dimensional numerical vector. In this paper, each tea pests image represented by a 1000-dimensional numerical vector.
4. Training classifier: In our study, we apply the SVM and MLP to classify and identify the 1000-dimensional numerical vector and the label of the input image is determined by the classifier. The workflow based on the BoVW model is shown in Figure 4.

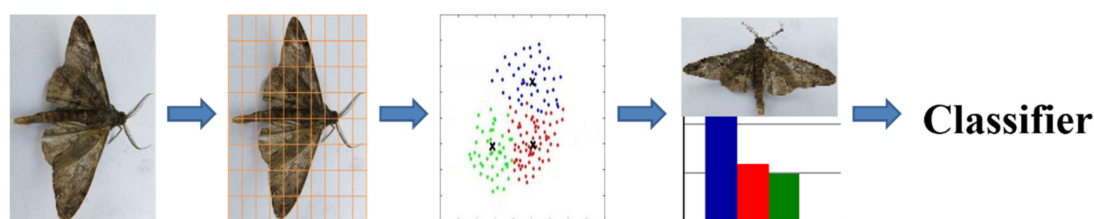


Figure 4. Workflow based on the BoVW model used for insect recognition.

3. Results

In this study, we evaluated the identification accuracy of tea tree pest species by implementing a convolutional neural network and traditional machine learning algorithm. We trained and fine-tuned the VGGNET-16 deep learning framework structure. The experimental results are shown in Figure 5. The error matrices of the three methods are shown in Figures 6–8.

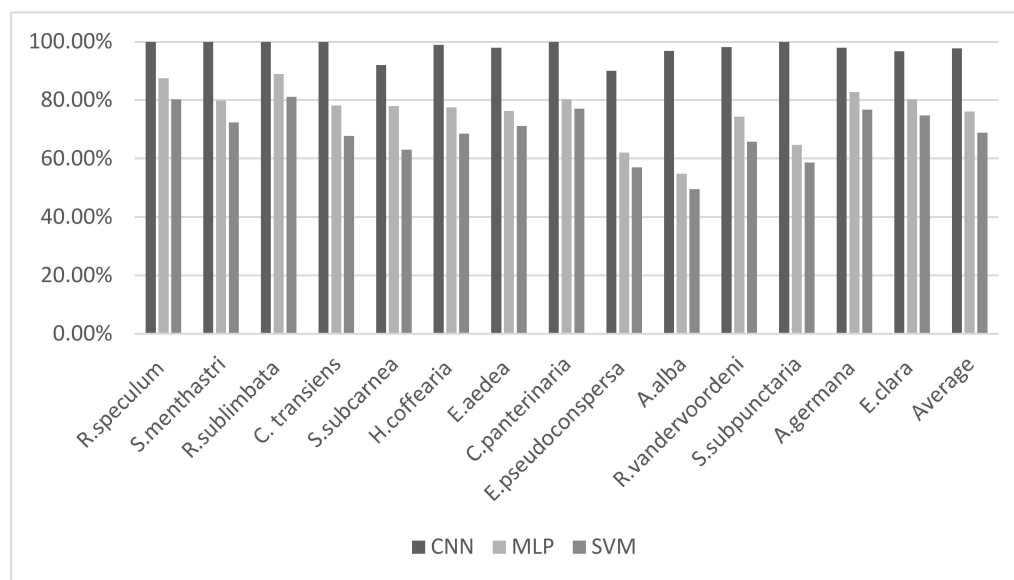


Figure 5. Accuracy of the three classification models in identifying 14 tea pests.

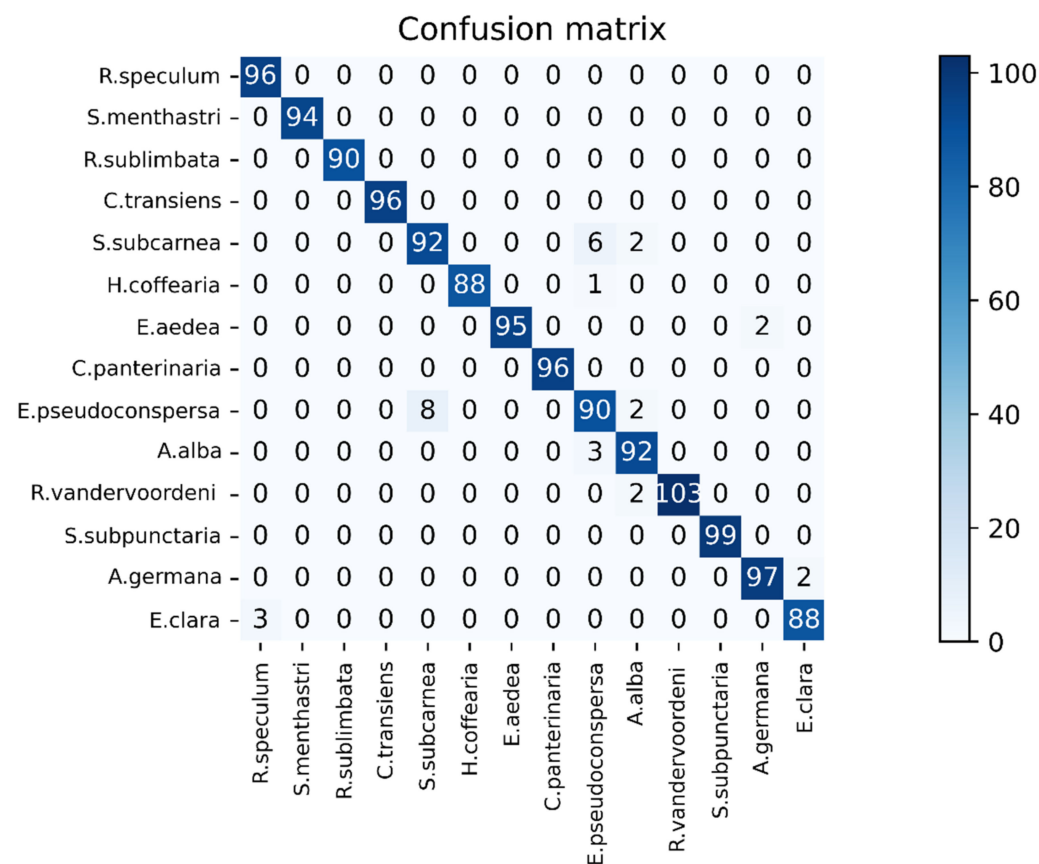


Figure 6. Error matrix showing the classification accuracy of the CNN algorithm.

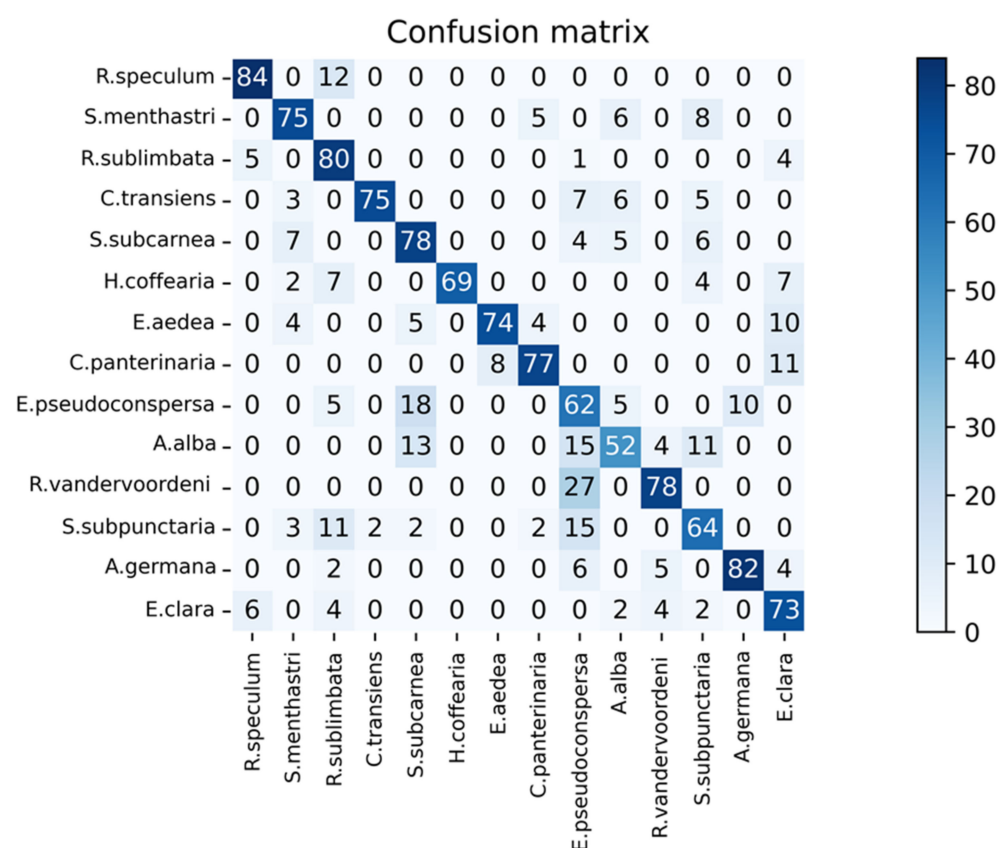


Figure 7. Error matrix showing the classification accuracy of the MLP algorithm.

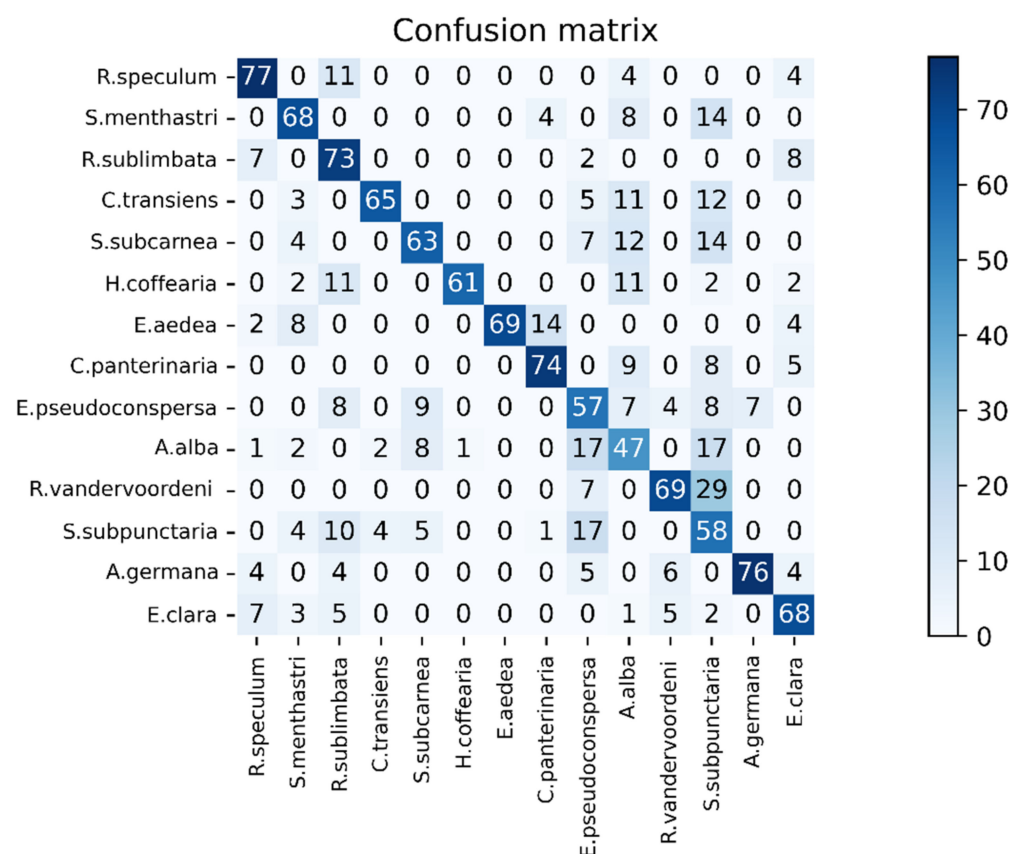


Figure 8. Error matrix showing the classification accuracy of the SVM algorithm.

After fine-tuning the architectures, the identification accuracy of *Ricania speculum* (Walker), *Spilosoma menthastri* (Esper), *Ricania sublimbata* Jacobi, *Ceratonoros transiens* (Walker), *Culcula panterinaria* (Bremer et Gray), and *Scopula subpunctaria* (Herrich-Schaeffer) were up to 100%. The identification accuracy of other pests is between 90% and 98.88%, and the overall MCA is 97.75%.

When using MLP, *Ricania sublimbata* Jacobi had the highest recognition accuracy of 88.89%, followed by *Ricania speculum* (Walker) (accuracy of 87.5%). *Arctornis alba* (Bremer) had the lowest recognition accuracy (only 54.74%). The identification accuracy of other pests ranged from 62% to 82.83%. The MCA is 76.07%.

When utilizing the SVM, *Ricania sublimbata* Jacobi had the highest recognition accuracy of 81.11%, followed by *Ricania speculum* (Walker) (accuracy of 80.21%). *Arctornis alba* (Bremer) had the lowest recognition accuracy (49.47%). Other pests were identified with an accuracy ranging from 57% to 67%. The MCA is 68.81%.

In addition, the performance of our method is compared against the methods in literature [18]. The accuracy of our method is slightly higher than the aforementioned algorithms (Table 3). Although the species of pests in Literature 29 are few, they have a complicated shooting background, which is the main reason for their low accuracy. Consequently, the method proposed here to classify tea pests is slightly superior to the previously described algorithms.

Table 3. Comparison of our proposed algorithms and previously published classification algorithms.

Method	Pest Category	Accuracy (%)
VGGNet-16	14	97.75%
VGGNet-16	9	92.13%
VGGNet-19	9	97.39%

In general, the recognition accuracy of the convolutional neural network is better than those of the traditional machine learning algorithms. By fine-tuning the pre-training network on our dataset, the network could easily achieve state-of-the-art accuracy while also saving a lot of training time. Therefore, transfer learning is an efficient deep learning algorithm in image classification. It does not need to train the network from scratch and has low dataset size requirements.

4. Discussion

Convolutional neural networks have been widely used in agricultural fields, such as in plant disease identification, and with favorable performance. In this study, we showed that image-based tea tree pest recognition with convolutional neural networks is efficient. This method avoids the artificial extraction and filtering of features from images in advance, as is the case in the SVM and MLP, thus saving a lot of time. Moreover, feature selection is a subjective activity, and its quality directly affects the recognition accuracy of the algorithm. However, the convolutional neural network can automatically carry out feature extraction and recognition without human intervention.

Traditional machine learning involves learning knowledge from a large number of annotated data and then using this model to classify new data. However, due to the small amount of data available on tea pests, we applied the transfer learning method to realize a deep neural network for tea pests identification. We could thus use a small amount of training data and training time, and our experimental results showed that our network had the highest prediction accuracies compared with two traditional classifiers. Therefore, we showed that transfer learning can realize the identification of tea tree pests under natural light conditions by fine-tuning the VGGNET model with a small size of data. The overall identification accuracy is satisfactory.

In our paper, the process of sample data augmentation is time-consuming, but with the non-stop boom of network data resources, the range of sample images of tea pests

will proceed to increase and become easier to obtain. Although the current convolutional neural network has a better identification accuracy than traditional classifiers for tea tree pests, we still need to improve the operation time of the network. This is a direction for future study. At present, the CNN has been applied to the identification of the grape pests. However, it needs to be in addition investigated according to the actual situation of the grape pests, so as to verify this model is highly optimized for general use. Meanwhile, our next step is to apply transfer learning to other network structures to identify more species of tea pests.

Author Contributions: Conceptualization, J.C. and L.G.; methodology, J.C. and L.G.; software, J.C. and Q.L.; validation, Q.L. and J.C.; writing—original draft preparation, J.C.; writing—review and editing, L.G., Q.L. and J.C.; project administration, L.G., Q.L. and J.C.; funding acquisition, L.G., Q.L. and J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by National Natural Science Foundation of China (31860477) National Natural Science Foundation of China (M2042001) and China agriculture research system (CARS-29-bc-3).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, N.; Yuan, M.F.; Wang, P.; Zhang, R.B.; Sun, J.; Mao, H.P. Tea Diseases Detection Based on Fast Infrared Thermal Image Processing Technology. *J. Sci. Food Agric.* **2019**, *99*, 3459–3466. [[CrossRef](#)] [[PubMed](#)]
2. Hazarika, L.K.; Bhuyan, M.; Hazarika, B.N. Insect Pests of Tea and Their Management. *Annu. Rev. Entomol.* **2009**, *54*, 267–284. [[CrossRef](#)] [[PubMed](#)]
3. Martineau, M.; Conte, D.; Raveaux, R.; Arnault, I.; Munier, D.; Venturini, G. A survey on image-based insect classification. *Pattern Recognit.* **2017**, *65*, 273–284. [[CrossRef](#)]
4. Liu, Z.; Gao, J.; Yang, G.; Zhang, H.; He, Y. Localization and Classification of Paddy Field Pests using a Saliency Map and Deep Convolutional Neural Network. *Sci. Rep.* **2016**, *6*, 20410. [[CrossRef](#)] [[PubMed](#)]
5. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
6. Hughes, D.; Salathe, M. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv* **2015**, arXiv:1511.08060v2.
7. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [[CrossRef](#)] [[PubMed](#)]
8. Pichayoot, O. Corn Disease identification from leaf images using convolutional neural networks. In Proceedings of the 2017 21st International Computer Science and Engineering Conference (ICSEC), Bangkok, Thailand, 15–18 November 2017; pp. 233–238.
9. Ma, J.; Du, K.; Zheng, F.; Zhang, L.; Gong, Z.; Sun, Z. A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. *Comput. Electron. Agric.* **2018**, *154*, 18–24. [[CrossRef](#)]
10. Chen, J.; Liu, Q.; Gao, L. Visual Tea Leaf Disease Recognition Using a Convolutional Neural Network Model. *Symmetry* **2019**, *11*, 343. [[CrossRef](#)]
11. Zhang, X.; Qiao, Y.; Meng, F.; Fan, C.; Zhang, M. Identification of Maize Leaf Diseases Using Improved Deep Convolutional Neural Networks. *IEEE Access* **2018**, *6*, 30370–30377. [[CrossRef](#)]
12. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
13. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2015**, arXiv:1409.1556.
14. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. *arXiv* **2015**, arXiv:1409.4842.
15. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
16. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A Survey on deep transfer learning. In Proceedings of the 27th International Conference on Artificial Neural Networks, Rhodes, Greece, 4–7 October 2018; pp. 270–279. [[CrossRef](#)]
17. Souza, W.; Alves, A.N.; Borges, D.L. A deep learning model for recognition of pest insects in maize plantations. In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, 6–9 October 2019.
18. Cao, X.; Wei, Z.; Gao, Y.; Huo, Y. Recognition of Common Insect in Field Based on Deep Learning. *J. Phys. Conf. Ser.* **2020**, *1634*, 012034. [[CrossRef](#)]

19. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [[CrossRef](#)]
20. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 3320–3328.
21. Sladojevic, S.; Arsenovic, M.; Anderla, A.; Culibrk, D.; Stefanovic, D. Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. *Comput. Intell. Neurosci.* **2016**, *2016*, 3289801. [[CrossRef](#)] [[PubMed](#)]
22. Liu, B.; Zhang, Y.; He, D.; Li, Y. Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks. *Symmetry* **2017**, *10*, 11. [[CrossRef](#)]
23. Ferentinos, K. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318. [[CrossRef](#)]
24. Ramcharan, A.; Baranowski, K.; McCloskey, P.; Ahmed, B.; Legg, J.; Hughes, D. Using Transfer Learning for Image-Based Cassava Disease Detection. *arXiv* **2017**, arXiv:1707.03717v2.
25. Korzh, O.; Cook, G.; Andersen, T.; Serra, E. Stacking approach for CNN transfer learning ensemble for remote sensing imagery. In Proceedings of the 2017 Intelligent Systems Conference, London, UK, 7–8 September 2017; pp. 599–608.
26. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 675–678.
27. Csurka, G.; Dance, C.R.; Fan, L.X.; Willamowski, J.; Bray, C. Visual categorization with bags of keypoints. In Proceedings of the 2004 Workshop on Statistical Learning in Computer Vision, Prague, Czech Republic, 10–16 May 2004; pp. 1–2.