agriculture

MDPI

Article

VGNet: A Lightweight Intelligent Learning Method for Corn Diseases Recognition

Xiangpeng Fan [1,2] and Zhibin Guan [1,2,*]

[1] Agricultural Information Institute, Chinese Academy of Agricultural Sciences, Beijing 100081, China; fanxiangpeng@caas.cn
[2] National Nanfan Research Institute (Sanya), Chinese Academy of Agricultural Sciences, Sanya 572024, China
* Correspondence: guanzhibin@caas.cn

Abstract: The automatic recognition of crop diseases based on visual perception algorithms is one of the important research directions in the current prevention and control of crop diseases. However, there are two issues to be addressed in corn disease identification: (1) A lack of multicategory corn disease image datasets that can be used for disease recognition model training. (2) The existing methods for identifying corn diseases have difficulty satisfying the dual requirements of disease recognition speed and accuracy in actual corn planting scenarios. Therefore, a corn diseases recognition system based on pretrained VGG16 is investigated and devised, termed as VGNet, which consists of batch normalization (BN), global average pooling (GAP) and L2 normalization. The performance of the proposed method is improved by using transfer learning for the task of corn disease classification. Experiment results show that the Adam optimizer is more suitable for crop disease recognition than the stochastic gradient descent (SGD) algorithm. When the learning rate is 0.001, the model performance reaches a highest accuracy of 98.3% and a lowest loss of 0.035. After data augmentation, the precision of nine corn diseases is between 98.1% and 100%, and the recall value ranges from 98.6% to 100%. What is more, the designed lightweight VGNet only occupies 79.5 MB of space, and the testing time for 230 images is 75.21 s, which demonstrates better transferability and accuracy in crop disease image recognition.

Keywords: VGNet; corn diseases; leaf detection; lightweight; transfer learning; agriculture

check for updates

Citation: Fan, X.; Guan, Z. VGNet: A Lightweight Intelligent Learning Method for Corn Diseases Recognition. *Agriculture* **2023**, *13*, 1606. https://doi.org/10.3390/agriculture13081606

Academic Editor: Roberto Alves Braga Júnior

Received: 19 June 2023
Revised: 8 August 2023
Accepted: 9 August 2023
Published: 14 August 2023

Copyright: © 2023 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/).

1. Introduction

Crop diseases can cause irreversible damage to crop growth and are considered one of the main limiting factors for crop cultivation, and spraying pesticides is the main measure to address crop diseases. Appropriate pesticide category selection and dosage regulation can ensure effective crop disease resolution and avoid pesticide residues' ecological impact. Therefore, accurately identifying the types and degrees of crop diseases is a prerequisite for achieving precise agricultural spraying [1–8]. In traditional methods, professionals mainly detect and identify crop diseases based on their naked eyes and experience, but it is time-consuming, laborious, and subjective. With the development of deep learning (DL) and visual perception technology, visual feature learning methods based on deep learning have become the mainstream of crop disease recognition, which realizes automatic recognition of crop diseases by extracting and learning the pest and disease features of crop images [9,10].

Deep learning is a branch of machine learning that mainly utilizes deep artificial neural networks to extract multilayer visual features and fuse multigranularity features of input images, thereby achieving high-level semantic learning of images [11]. Unlike traditional machine learning methods, deep learning methods require significant computational resources, because deep artificial neural network models optimize model parameters through a large number of parameter calculations in the high-level semantic learning of images.

With the rapid development of high-performance computing and image processing units, deep learning methods have been successfully applied in various fields, which has turned out to be very excellent in discovering intricate structures in high-dimensional data and is therefore applicable to many domains of science, engineering [12–14], industries[15–17], bioinformatics [18–20], and agriculture [21–26]. Concretely, deep learning has provided many significant works in the field of plant stress phenotyping and image analysis for detection [27–30], recognition [31–34], classification [35–38], quantification [39], and prediction [40] in agriculture to tackle the challenges of agricultural production [41]. And the convolutional neural network (CNN)-based approaches are arguably the most commonly used [42].

Ferentinos developed a plant diseases detection model with a best performance of 99.5% using 87,848 images under controlled conditions [43]. Liang et al. designed a deep plant diseases diagnosis and severity estimation network (PD2-SE-Net) model to identify plant species, diseases, and their severities with a final accuracy of 99% [44]. They utilized the artificial intelligence (AI) Challenger [45] images for experiment data. The approach they proposed reached an accuracy of 99.4%. Zhong et al. proposed an apple diseases classification method based on dense networks with 121 layers (DenseNet-121) and 2462 apple leaf images from AI Challenger, which achieved an accuracy of 93.71% [46]. He et al. proposed an approach to detect oilseed rape pests based on SSD with an Inception module, which was helpful for integrated pest management [47]. Zeng et al. introduced a self-attention mechanism to a convolutional neural network, and the accuracy of the proposed model reached 98% using 9244 diseased cucumber images [48].

Deep convolutional neural networks have a strong ability for feature learning and expression. The above crop disease recognition methods based on CNNs have achieved good accuracies or success rates. However, the accuracy and robustness of deep learning models require training on a large amount of image data. There are two issues that need to be addressed in crop disease identification. On the one hand, there is a lack of diverse maize disease training datasets, as most of the crop disease images used in the existing methods are created under controlled or laboratory conditions. On the other hand, the complexity of existing corp disease models is high, making it difficult to meet the actual detection needs of field scenarios, and their performance in identifying fine-grained corn diseases is insufficient. Therefore, we introduced transfer learning and designed VGNet to solve the above problems. Specifically, we first collected corn disease image data from real field scenarios, covering nine types of corn diseases, which can be used for parameter optimization of fine-grained corn disease recognition models. Afterwards, we designed a relatively simple VGNet model based on the VGG16 model but with relatively high accuracy in identifying crop diseases, which can meet the disease detection needs of actual corn planting scenarios.

The reason why the VGG16 model is selected as the backbone network is that the VGG network is a straight cylinder network structure, and its computing resource consumption is significantly less than the residual network structure, which can satisfy the dual needs of speed and accuracy in real-time crop disease detection. In the VGNet method, the structure of VGG16 is modified by adding the BN, replacing two hidden fully connected layers with a GAP layer, and adding L2 normalization. Through the comparative experiment of different training methods, parameters, and datasets, the redesigned VGNet after fine-tuning achieves an accuracy of 98.3%, which can achieve a 66.8% reduction in testing time compared with the original VGG16 model. The following summary provides the main contributions of this paper:

- A lightweight intelligent learning method, termed as VGNet, is proposed for multiple categories of corn disease detection.
- Fine-grained corn disease images are collected and can be used for the parameter optimization of corn disease recognition models.
- Evaluation results show that the accuracy of the proposed method in disease detection reaches 98.3%, which can satisfy the detection requirements of practical scenarios.

The remainder of this paper is organized as follows. Section 2 describes the materials and methods. The experiment results of VGNet are detailed in Section 3. In Section 4, the discussion of VGNet for fine-grained corn disease recognition is given. Finally, the conclusions are drawn in Section 5. Further research directions are also proposed.

2. Materials and Methods

2.1. Image Samples

2.1.1. Images for Pretraining

In the field of crop disease recognition, many crop disease datasets have appeared, among which the most commonly used ones are PlantVillage [49] and AI Challenger datasets. PlantVillage contains open and free datasets with 54,306 annotated images and 26 diseases for 14 crop plants, and it was created by Mohanty et al. under controlled conditions [50]. AI Challenger is provided by the Shanghai Science and Technology Innovation Center as a new guest competition crop leaf image datasets, with 45,285 marked images, containing 10 kinds of plants (apple, cherry, grape, orange, peach, strawberry, tomatoes, peppers, corn, and potato), 27 kinds of diseases, and a total 61 categories. Both of these datasets are open-source image datasets containing healthy plant leaves and diseased leaves and have great similarity with the target disease image dataset in this research area. ImageNet dataset [51] contains a large number of images from all aspects of life, and the initial training of VGG16 was obtained through the ImageNet dataset, which has achieved excellent results. These three different large open datasets were used for pretraining the selected CNN structure. The properties of the three pretrained experimented datasets are shown in Table 1.

Table 1. Properties of the pretrained experimented datasets.

Dataset	Classes	Samples	Features Type	Image Type
ImageNet	1000	14,197,122	coarse-grained	RGB
PlantVillage	38	54,306	fine-grained	RGB
AI Challenger	61	45,285	fine-grained	RGB

2.1.2. Images for Parameter Optimization

In this experiment, the images used for recognition and fine-tuning training were composed of symptom pictures of nine corn diseases caused by fungus. They were Anthracnose (ANTH), Tropical Rust (TR), Southern Corn Rust (SCR), Common Rust (CR), Southern Leaf Blight (SLB), Phaeosphaeria Leaf Blight (PHLB), Diplodia Leaf Streak (DLS), Physoderma Brown Spot (PHBS), and Northern Leaf Blight (NLB) of corn. The images were captured using a digital camera (Nikon D750) under natural field conditions at the Western Corn Farm of Urumqi, Xinjiang, China. In order to make the collected images be more representative, symptom images were obtained, respectively, in sunny, cloudy, and windy weather conditions from different times in the morning, noon, and evening with multiangle shooting. The shooting background was complicated, containing corn stalks, soil, weeds, and blades covering each other, etc., to reflect the practical growth situation of corn. There is a total of 1150 images obtained in a 3096 $\times$ 3096 pixel spatial resolution. The sample numbers of various diseases are kept balanced relatively. The quantity distribution of corn disease images is shown in Figure 1. Some image examples are shown in Figure 2.

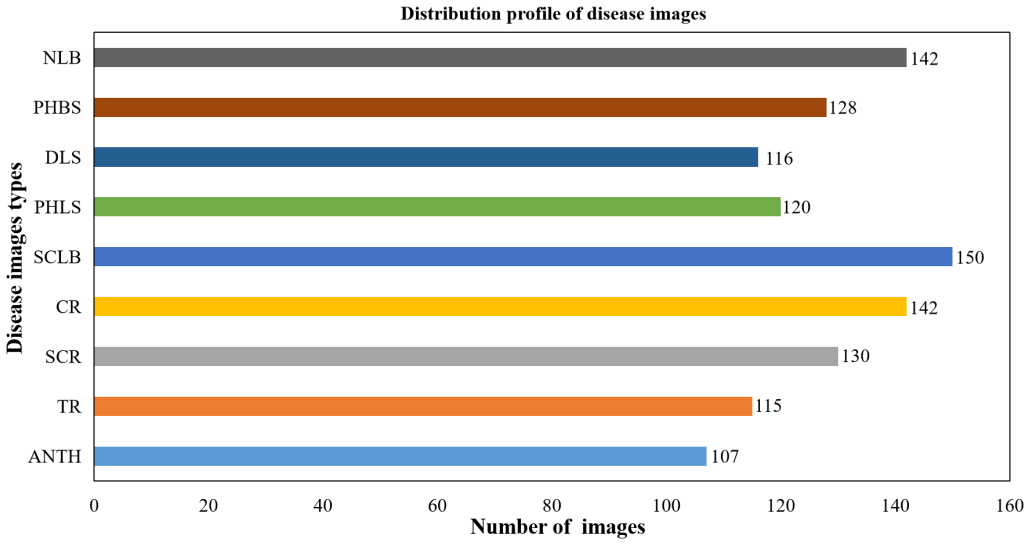


Figure 1. The quantity distribution of maize disease images with complex background.

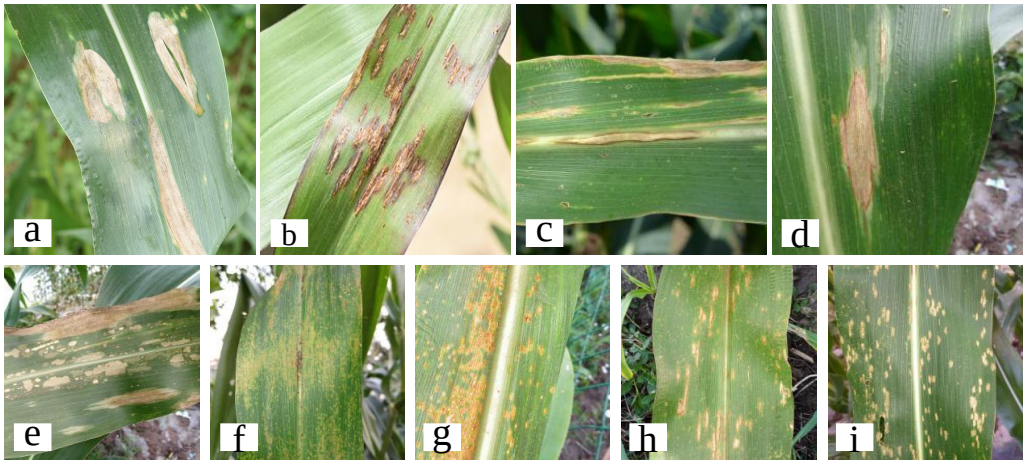


Figure 2. Some examples of corn disease images with complicated backgrounds from a field: (**a**) Northern leaf blight. (**b**) Common rust. (**c**) Anthracnose. (**d**) Diplodia leaf streak. (**e**) Phaeosphaeria spot. (**f**) Physoderma brown spot. (**g**) Sourthern corn rust. (**h**) Sourthern corn leaf blight. (**i**) Tropical rust.

2.1.3. Data Preprocessing

Data preprocessing includes annotation, cropping, or zooming. Firstly, the CNN model needs supervised training and learning; so, it is necessary to manually annotate the disease images acquired in the field. After the images were confirmed by corn pathologists, the LabelMe tool was used for annotation, and the annotated images were saved as PASCAL VOC2007 format. Secondly, because the images from the corn field and public dataset websites have different resolution and sizes, the size of each image is uniformly cropped and resized to $(224, 224, 3)$ channels.

2.2. Backbone Network

2.2.1. CNN and VGG16 Network

The CNN is one of the classical network algorithms of deep learning. A CNN consists of input layers, convolutional layer, activation function, pooling layers (sampling layer), fully connected layers, and classification layers. Several baseline architectures of CNN have been developed for image recognition, including AlexNet, GoogLeNet, VGGNet, XceptionNet, and ResNet et al. [52]. VGG Net was first devised by Simonyan and Zisserman (2015) for the ILSVRC-2014 challenge. It has been proven to have excellent performance

for image classification. The most significant superiority of VGG Net is the utilization of a smaller convolution kernel and pooling window in the feature extractor, which can extract fine-grained features from the input data. Figure 3 shows the basic structure diagram of VGG16. VGG16 contains thirteen convolutional layers and three fully connected layers with 4096, 4096, and 1000 dimensions, respectively. There are five maximum pooling layers between the convolutional layers. During training, the input to VGG16 is a fixed $(224, 224, 3)$-channel RGB image. Large receptive fields in VGG16 were substituted with consecutive layers of 3×3 convolution filters. The convolutional stride was fixed to 1 pixel. The padding of the convolution layer input was maintained as 1 pixel and max-pooling was performed with a stride of 2 over a 2×2 pixel pooling window. The neuron activation function used in VGG16 is the rectified linear unit (ReLU) function.

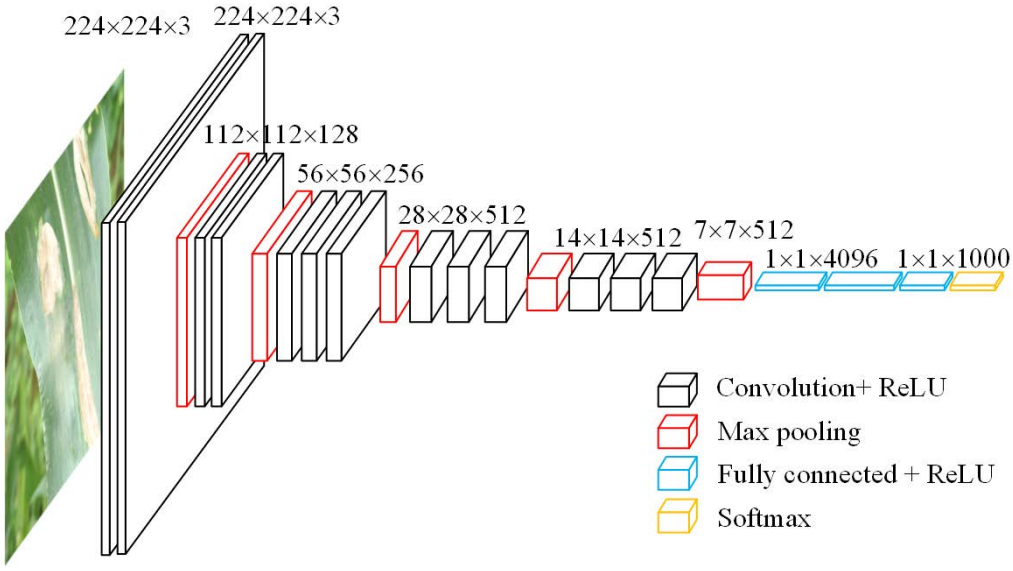


Figure 3. Structure diagram of original VGG16 convolutional neural network.

2.2.2. Proposed Approach and Processes

Figure 4 describes the main process of the VGNet with transfer learning for corn disease recognition. The whole recognition process includes three parts. Part one is the pretraining and parameters transfer process of original VGG16 using three different large datasets, the aim of transfer learning is to shift the general knowledge of image classification acquired by VGG16 from a large image dataset to the new corn leaf disease recognition model. Part two is the establishment of VGNet, the remaining part is fine-tuning the updated VGNet with a new image dataset. After acquiring the new images, they were preprocessed and divided into training set and test set. The modification of the VGG16 network included adding a batch normalization layer to speed up fine-tuning training, replacing the two hidden dense layers by a global average pooling layer to reduce feature dimension, and integrating the L2 regularization algorithm to improve the ability of the model to extract effective features from complex backgrounds. The last layer of the VGG Net was changed by a 9-tag softmax classifier instead of the original softmax classifier with 1000 tags. Three large open datasets were used to obtain the model parameters and feature extraction abilities in the pretraining process, and different training tactics in the parameter tuning were utilized to optimize the VGNet model. After pretraining, the convolutional layers and pooling layers remained unchanged. Their parameters were loaded to the newly designed VGG16 Net and then they were frozen. The VGNet was fine-tuned through the iteration of loss function to reoptimize the parameters of the remaining fully connected layer and softmax function. Finally, the test process was executed by the designed model.

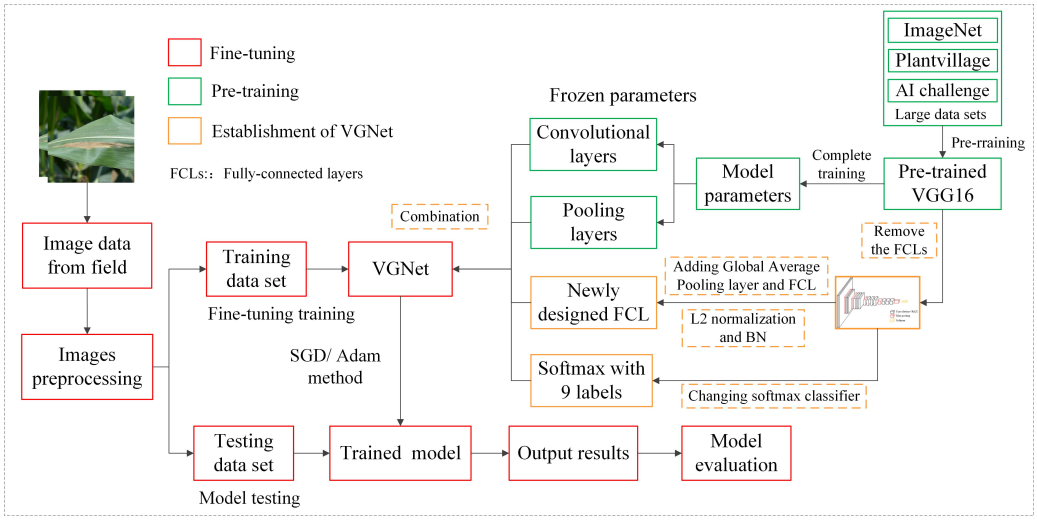


Figure 4. Flowchart of corn disease image recognition method based on transfer learning and VGNet.

2.3. VGNet

As described in Section 2.2.1, the original VGG16 network has 13 convolutional layers, 5 pooling layers, and 3 fully connected layers, and it has 138 million parameters and large amounts of computation, leading to the consumption of both memory and time. The model will easily fall into an overfitting state and lower convergence. Thus, we redesigned VGNet to improve the accuracy and real-time performance of the VGG-based network. Normalization strategies were also adopted, including adding batch normalization (BN) processing and the L2 normalization algorithm. The number of our class labels in the softmax layer of VGNet is 9.

2.3.1. Batch Normalization

For the convolutional neural network, the normalization of datasets is required in the gradient descent process, which can prevent gradient explosion and accelerate the convergence of the network. Thus, batch normalization (BN) processing was applied to normalize the feature map of each sample after the convolutional layers. The mean (μ) and variance (σ) of the total number of pixels in the feature graph were obtained firstly; then, the normalization equation was utilized to calculate the sample normalization values, and the optimal value search data are converted into the standard normal distribution. The BN layer can effectively solve the problem of the data distribution changes in the middle layer during the training process of the model. BN can also accelerate convergence, improve accuracy, and reduce the overfitting phenomenon. The calculation equations of mean (μ) and variance (σ) of the feature maps are described as Equations (1) and (2).

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{1}$$

$$\sigma = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2. \tag{2}$$

where x_i represent the value of the ith pixel in the image sample. n represents the total number of pixels in the sample. The normalization equation is shown in Formula (3).

$$\bar{x} = \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}}. \quad (3)$$

where x represents the normalized pixel value of the ith pixel of the sample. ε is a small constant value greater than 0 to ensure that the denominator in Equation (3) is greater than 0. According to the batch normalization algorithm in the training process, the average value and variance of the data estimated based on each batch will be used to replace the actual average value and variance, and the data will be converted to the standard normal distribution according to the estimated average value and variance. The data of the standard normal distribution will be restored by constantly updating the values of x_i and u during the training process. And then they are output by the model.

2.3.2. Replacing Fully Connected Layers by GAP Layer

Although the original VGG16 network structure has 16 weight layers, there is a large number of parameters in the fully connected layer, which leads to excessive computation in the training and testing process. Thus, we decided to compress its weight matrix using a global average pooling (GAP) layer after the last convolutional layer, which outputs a series of feature maps with a depth the same as the number of classes in the classification problems. A GAP layer could enhance the relationship between feature map and category. It has been proven that GAP layers can replace fully connected layers in a conventional structure and thus reduce the storage required by the large weight matrices of the fully connected layers [53]. Performing GAP on a feature map involves computing the average value of all the elements in the feature map.

The principle of GAP is to shrink the parameter space to avoid overfitting and enable precise adjustment of the dropout ratio, which can be treated as the process of dimension reduction in a feature matrix. As shown in Figure 5, the output feature maps from C_I, which is the last convolutional layer, are downsampled into fm_{GAP}, which has a size of $1 \times 1 \times size_{fm}$ after global average pooling. In GAP, the weight matrices of f_1, W can be adjusted as Equation (4) as follows:

$$W' = \sum_{l=(j-1)size_{fm}^2+1}^{j*size_{fm}^2} W_{i,j}. \quad (4)$$

where $size_{fm}$ is the size of the input feature map, i, j is the index of the output neurons and input feature maps, and W' is the modified weight matrix. As shown in Figure 5, the corresponding weights of each feature map are summed up, and each matrix in W is modified and reduced to a column vector composed of $1 \times 1 \times$ depth of fm_{GAP}. Thus, the dimension reduction in the feature matrix is realized. Instead of adding fully connected layers on top of the feature maps, we take the average of each feature map, and the resulting vector is fed directly into the softmax layer. One advantage of the GAP layer over the fully connected layers is that it is more native to the convolution structure by enforcing correspondences between feature maps and categories. Another advantage is that there is no parameter to optimize in the GAP layer, thus overfitting is avoided at this layer. Furthermore, the GAP layer sums out the spatial information, thus it is more robust to spatial translations of the input.

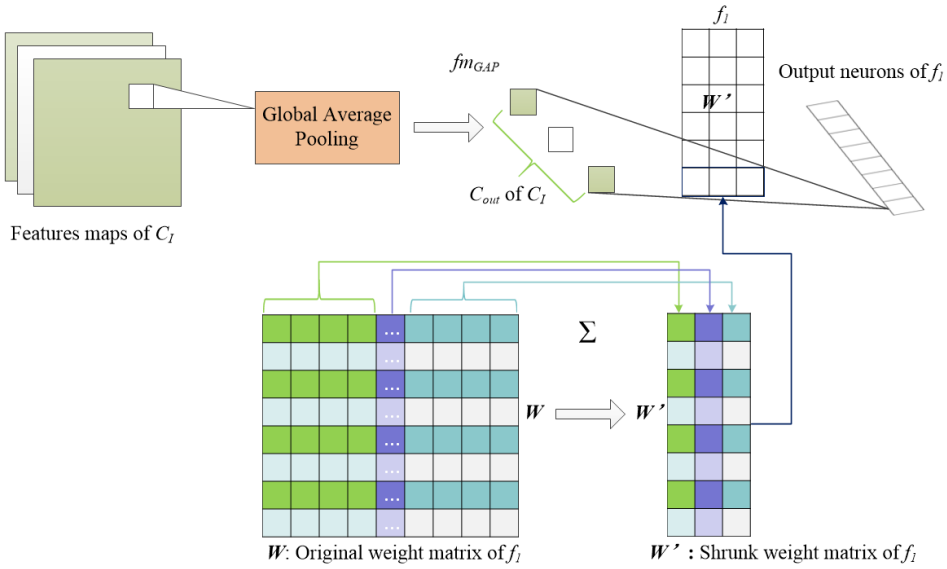


Figure 5. Flowchart of matrix dimension reduction by GAP layer feature.

2.3.3. L2 Normalization

The idea of L2 normalization is to add the regularization term (penalty term) to the loss function, which prevents the model from arbitrarily fitting the complex background and other noise information in the training set by restricting the most weight value ω in the model. Suppose the original loss function in the training process is $J_0(\omega, b)$, the utilization of L2 normalization is to optimize $J_0(\omega, b) + c\lambda R(\omega)$, and $R(\omega)$ is the regularization term or penalty term, which describes the complexity of the model. Relative equations above are illustrated in Equations (5)–(7).

$$J_0(\omega, b) = \frac{1}{m}\sum_{i=1}^{m} L(y'^{(i)}, y^{(i)}). \tag{5}$$

$$R(\omega) = \|W\|^2 = \sum_{j=1}^{l} \omega_j^2. \tag{6}$$

$$J(\omega, b) = \frac{1}{m}\sum_{i=1}^{m} L(y'^{(i)}, y^{(i)}) + \frac{\lambda}{2m}\sum_{j=1}^{l} \omega_j^2. \tag{7}$$

where, $J_0(\omega, b)$ is the original loss function; ω is the weight in the neuronal transmission process; relatively, ωj stands for the weight of the jth neuron and b represents the bias of neuronal transmission process; m represents the size of the sample dataset; $y'(i)$ represents the actual output value; $y(i)$ represents the expected output of a neuron; l is the number of dense; k is the number of neurons; $J_0(\omega, b)$ represents the new updated loss function; and λ is the parameter of L2 normalization. From Equation (9), it can be seen that the realization of L2 normalization is adding the sum of squares of the weight coefficients to the original loss function. In this experiment, the λ parameter was set to 0.12.

2.4. Transfer Learning and Fine-Tuning

In the field of deep learning, it is often necessary to train the model with a large number of datasets. However, in practical application, it is often difficult to obtain a large-scale dataset in the target field. Therefore, the idea of transfer learning can be adopted, and the image classification and recognition ability acquired by the deep convolutional neural network model trained on a large dataset after full training can be used to transfer the useful knowledge from the source domain to the new target domain. This makes the utility and inference scope from learned models much wider than an isolated model specific to individual plant species. Transfer learning also enables rapid progress and improved performance in modeling subsequent tasks by fine-tuning training. The most

commonly used transfer learning approach is parameter-based transfer learning, which uses a model but, after fine-tuning, the partial parameters are based on the new dataset. This process is often referred to as domain adaption. Thus, in the experiment, VGG16 was pretrained, and the parameters of the convolutional layers and pooling layers were transferred to the newly designed VGNet. The internal weights of the newly designed model are automatically updated by fine-tuning training. To obtain a preferable model for this research, external factors containing training methods, regularization techniques, and the value of the hyperparameters are considered in the fine-tuning process.

2.4.1. Parameter Fine-Tuning

In deep learning networks, making each network parameter learn automatically and effectively with the input of training data is the key procedure to let the network training converge towards the required direction. The learning rate defines the learning progress of the proposed model and updates the weight parameters to reduce the loss function of the network. Thus, learning rate is an important parameter in the training algorithm. Some optimization strategies for network training parameters have been put forward [54], such as SGD, AdaGrad, AdaDelta, RMSProp, Adam [55], etc. The SGD and Adam optimizer are the most commonly used in image classification applications. In this experiment, we compared performance with the fine-tuning training algorithm involving the SGD and Adam optimizer to obtain better performance of the VGNet model.

2.4.2. Experimental Environment

All of the experiments were performed on Windows 7 (64-bit) operation system. The RAM of the computer is 16 GB, with Intel(R) Xeon(R) CPU E5-2630 v4 @2.20GHz CPU. The program platform was Anaconda 3.5.0, CUDA 8.0. CuDNN was the library for CUDA, developed by NVIDIA, which provided highly tuned implementations of primitives for deep neural networks. Python 3.5.6 was applied based on TensorFlow environment. The image dataset of the fine-tuning process was divided into two parts: 80% of image data were for training and the remaining 20% were for testing. Table 2 presents the hyperparameters of the fine-tuning training process of VGNet.

Table 2. Specification of hyperparameters in the experiment.

Parameters	Setting Values
Initial learning rate (SGD, Adam)	0.001, 0.005, 0.01
Momentum (SGD)	0.9
Small constant τ (Adam)	10^{-8}
Weight decay (SGD, Adam)	0.00005
L2 normalization parameter λ	0.12
Iteration	5000

2.5. Evaluation of Proposed Method

The performances are graphically depicted for each model with accuracy and loss. An overall loss score and accuracy based on the test dataset are computed and used to determine the performance of the models. The accuracy is calculated on the testing dataset in a regular interval with validation frequency of 25 iterations, and it is given as Equation (8).

$$Acc = \frac{\text{Predicted samples}}{\text{Disease samples}}. \tag{8}$$

Meanwhile, categorical cross-entropy is used as the loss function, which has softmax activations in the output layer, which is illustrated as Equation (9)

$$Loss = \sum_{i=1}^{N} \sum_{j=1}^{K} t_{ij} \ln y_{ij}. \tag{9}$$

where N represents the number of corn disease images, K is the number of diseases classes, t_{ij} indicates that the ith disease image belongs to the jth disease class, and y_{ij} stands for the output for sample i for disease class j. To evaluate the results of the disease recognition and classification experiment in the confusion matrix intuitively, P_{re} (Precision) and R_{ec} (Recall) are calculated after testing the samples. They are used to measure how accurately the results for each category are with respect to the corresponding ground-truth data. A comprehensive evaluation index, the *F1* score, is used as the evaluation value of P_{re} and R_{ec}. Equations for P_{re}, R_{ec}, and *F1* score are as follows in Equations (10)–(12).

$$P_{re} = \frac{TP}{TP + FP}. \tag{10}$$

$$R_{ec} = \frac{TP}{TP + FN}. \tag{11}$$

$$F1 = \frac{2P_{re}R_{ec}}{P_{re} + R_{ec}}. \tag{12}$$

where, the *TP* (true positive) is the amount of positive data that are correctly predicted as positive. The *FP* (false positive) represents the amount of negative data points that are wrongly predicted as positive. The *FN* (false negative) is the amount of negative data that are misclassified as negative. *Pre* (Precision) is used to find the proportion of positive identifications that are true. *Rec* is used to determine the proportion of actual positives that were correctly identified. The *F1* score reflects the number of instances that are correctly classified by the learning models.

3. Results

In this study, an assessment of the appropriateness of VGNet with transfer learning and fine-tuning training for the task of crop disease recognition was carried out. Our focus was to pretrain the VGG 16 Network with different public datasets and to fine-tune the newly designed VGNet model with different a training mechanism and parameters. Large open datasets like ImageNet, PlantVillage, and AI Challenger were utilized to pretrain the model; then, the weights and parameters of the convolutional layers and pooling layers were transferred to the new model and frozen. After updating the structure of VGNet, the parameters of the GAP layer, the remaining fully connected layers, and the softmax layer were retrained and fine-tuned by the new dataset obtained from corn fields. The performance of the proposed method was analyzed after five-fold cross-validation experiments to acquire convincing results. K-fold cross-validation is a common method used to test the accuracy of DL algorithms. To perform K-fold cross-validation on the overall data, the image dataset C is divided into K parts for disjoint subsets. In order to prevent data leakage, suppose the number of training samples in dataset C is M; then, the number of samples in each subset is M/K. When training the network model, one subset is selected each time as the verification set, and the other (K-1) subsets are selected as the training set, and the classification accuracy of the network model on the selected verification set can be obtained. After repeating the above process for K times, the average of classification ac-curacy is obtained as the true classification accuracy of the model. In our research, the K is set as 5, since the results of 5-fold validation and 10-fold validation are the same in the previous experimental experience.

3.1. Effects of Fine-Turning Training Mechanism

The following sections analyze the effects on model performance with a different training mechanism in the fine-tuning VGNet process, including different training methods and initial learning rates. Table 3 shows the testing loss and accuracy of the different training mechanism in the fine-tuning process.

Table 3. Testing loss and accuracy of the method based on SGD or Adam with different learning ranges.

Optimizer	Initial Learning Rate	Loss	Accuracy (%)
SGD	0.01	0.103	85.6
SGD	0.005	0.089	89.1
SGD	0.001	0.061	93.0
Adam	0.01	0.074	91.3
Adam	0.005	0.058	94.4
Adam	0.001	0.035	98.3

From Table 3, it can be seen that six different experiments were carried out; their final loss values and accuracies of testing vary with the training methods and initial learning rate. Figures 6 and 7 show the loss and accuracy curves of two training methods with initial learning rates of 0.01 and 0.001, respectively. As seen in Figures 6 and 7 and Table 3, training methods and initial learning rate have great influence on the performance of the model. By comparing experiment 1, 2, and 3 using the SGD method, it can be found that the loss value decreases as the learning rate declines, while the accuracy increases with the fall in learning rate. When the learning rate is set to 0.01, the loss value of the model test is 0.103, and the accuracy is only 85.65%. In this process, the performance is unstable, and the loss and accuracy shake violently, which can be seen by the green curves in Figure 6. When the initial learning rate drops to 0.001, the loss value of the model test decreases to 0.061, and the accuracy is improved to 93.04%. At this time, the testing process has fewer shocks, and the model can converge at about 4500 iterations, which is described by green curves in Figure 7. Rows 4, 5, and 6 in Table 3 were fine-tuning-trained with the Adam optimizer. Their variation in loss value and accuracy are consistent with former experiments 1, 2, and 3. The reason is that with the aid of transfer learning, all the front layers of the network obtained good training, and the weight parameters at the initial time of training are close to the optimal state. If the initial learning rate is not set properly, the training process will shock and even diverge. If a higher learning rate (0.01) is used in the fine-tuning training phase, the model is likely to skip the optimal solution, resulting in larger loss, lower accuracy, or severe oscillation. When the initial learning rate is 0.001, the model is more stable, and its performances are much better. Therefore, when the transfer learning mechanism is applied to the training of a convolutional neural network, the initial learning rate in the fine-tuning training stage needs to be lower than that of the model trained from scratch.

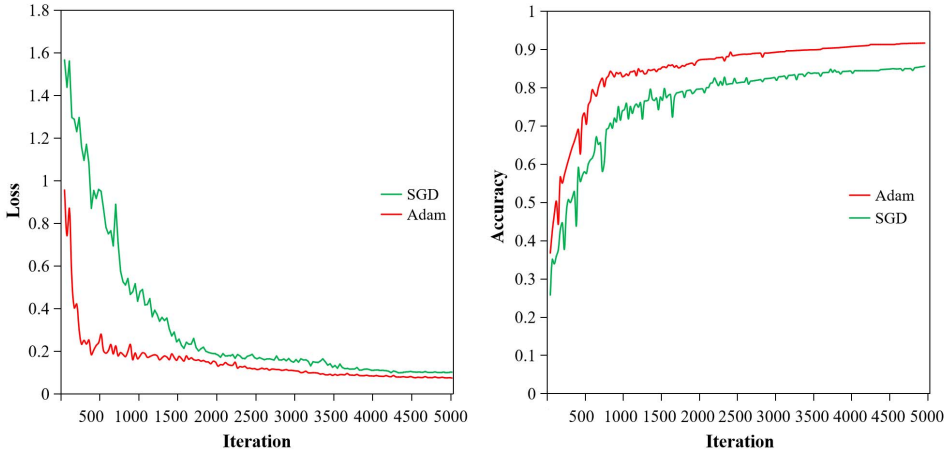


Figure 6. Comparison of loss and accuracy of two learning methods when the learning rate is 0.01.

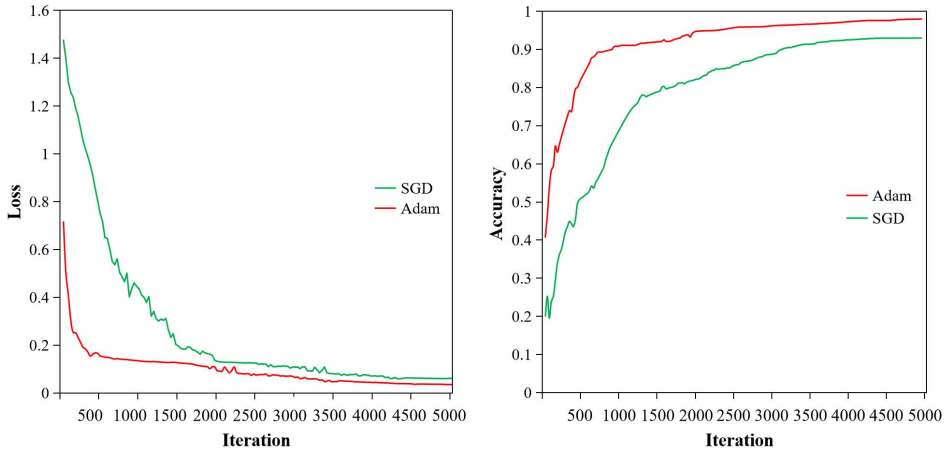


Figure 7. Comparison of loss and accuracy of two learning methods when the learning rate is 0.001.

Compare experiment 3 with experiment 6 in Table 3, where the initial learning rate was set as 0.001 with the SGD algorithm and Adam optimizer, respectively. At this point, the final performance of the model was different due to the different training methods. The loss value of the model trained by the Adam optimizer is lower than that of the model trained by SGD algorithm. Furthermore, the model trained by the Adam optimizer reaches convergence first and becomes stable after 3500 iterations, which is illustrated by the red curve in Figure 7. However, the model trained by the SGD method converges slowly, and the final loss value after convergence is 0.061, which is higher than the model trained by the Adam optimizer. Moreover, since the SGD training algorithm adjusts the weight for each data point, the network performance fluctuates up and down a lot more than the Adam optimizer during the learning process. The right part of Figure 7 shows the variation in the accuracy of the two training methods. It can be found that the model retrained by the Adam optimizer reached an accuracy of 98.26%, while the model retrained by the SGD algorithm did not perform as well. Apparently, when the model is fine-tuned by the SGD algorithm, it is always lower than when trained by the Adam optimizer. In general, the Adam optimizer algorithm has the advantage of faster model convergence than the SGD training algorithm and is more stable in the testing process. Therefore, the Adam optimizer in the fine-tuning training stage of the model is more in line with the corn disease recognition model.

3.2. Effects of Transfer Learning on Multiple Datasets

To explore the impact of training mechanisms and different datasets in the pretraining process, four completely selfsame VGNet models were utilized in the form of learning from scratch and transfer learning, respectively. The scratched learning model only adopted the image obtained from corn fields without pretraining. The other three models utilized three different large open datasets for pretraining and parameter transfer learning. The experimental results of applying four different learning types and datasets are listed in Table 4. From Table 4, it can be seen that the accuracy of learning from scratch is the lowest, reaching an accuracy of 69.57%. Under the condition of transfer learning and fine-tuning learning, the model pretrained using the PlantVillage dataset has the best performance, with an accuracy of 98.26%. Since training the VGNet model from scratch needs more images and time to optimize network parameters, and the training dataset only has 920 images, it is not enough for a deep convolutional neural network. This leads to the nonideal classification effect. Pretraining and transfer learning make the VGNet model acquire the ability of feature extraction and the knowledge of classification; thus, it is easier to achieve higher accuracy than with the scratched learning model. Therefore, transfer learning seems to be a better approach than learning from scratch when the dataset is not big enough. Though the original VGG16 Net is a model with excellent performance trained on ImageNet, a large public dataset, in general, the filter at the bottom of the model can acquire different local edge and texture information through training, which has good

universality for any image. However, the feature gaps between the ImageNet dataset from source area and the corn disease images in this new area are too large, while the other two datasets have much more similar features in color, texture, and shape to the corn disease images. Thus, the accuracies of the models pretrained with PlantVillage and AI Challenger are higher than the model pretrained with ImageNet. Images from PlantVillage are very similar to those from AI Challenger, but the number of PlantVillage is bigger than that of AI Challenger. Thus, the model pretrained with PlantVillage obtains a better learning effect, and PlantVillage is more suitable for the pretraining in this research. This indicates that in transfer learning, the source domain and target domain should have a high fitting degree for better performance.

Table 4. Experiment results of different learning types and datasets for pretraining and fine-tuning.

Learning Types	Pretrained Images	Accuracy on Original Images (%)	Accuracy on Augmented Images (%)
Learning from Scratch	—	69.6	89.5
Transfer learning	ImageNet	93.5	94.6
Transfer learning	PlantVillage	**98.3**	**99.4**
Transfer learning	AI Challenger	97.3	91.3

3.3. Effects of Augmentation

Data augmentation was applied here based on image transformations, such as geometric transformation, color changing, and noise adding, to generate new training images from the original ones by applying such random image transformations. The size of the dataset was enlarged from 1150 to 11,500. The ratio of the training dataset and testing dataset was also 8:2. The effects of image augmentation for fine-tuning learning are also illustrated in Table 4. It can be concluded that the effects of image data augmentation on different training models are different. In the mode of learning from scratch, data augmentation improves the accuracy by nearly 20%. Because the original dataset is too small, and the structure of the network structure is deep, the overfitting phenomenon reduces the performance of the network. When the image data are enlarged by data augmentation, the number and diversity of the data are increased. Thus, data augmentation has a larger role in avoiding overfitting and increasing accuracy when the model is learning from scratch. In the transfer learning mode, the accuracy of the fine-tuned model trained with augmentation is at least 2% higher than that of the model fine-tune-trained by original image data. This is because the pretraining model has learned a lot of knowledge from the large image dataset, which weakens the role of data augmentation. Hence, enlarging data plays a slight role in improving the performance of model classification in transfer learning.

4. Discussion

4.1. Obfuscation Matrix Analysis and Quantitative Statistics

To clearly show the recognition precision and classification results based on the fine-tuning training of the designed VGNet with augmented datasets, the confusion matrix drawn on the basis of the model classification results is shown in Figure 8. ANTH, TR, SCR, CR, SLB, PHLS, DLS, PHBS, and NLB, respectively, represent the abbreviations of nine types of corn diseases. The values in darker diagonal lines in Figure 8 (left) illustrate the number of correct classifications for each disease category, while the results of darker diagonal lines in Figure 8 (right) represent the recognition accuracies of correct classifications. It can be found that the recognition accuracies of nine corn diseases present some differences. Relatively, the accuracy of ANTH (Anthracnose) is lower than others; this probably because the sample number is fewer than other types. And the accuracy of SCR (Southern corn rust) reaches 100%. On the whole, the accuracies are kept in the range of 98.6% and 100%, which can be treated as a balanced result.

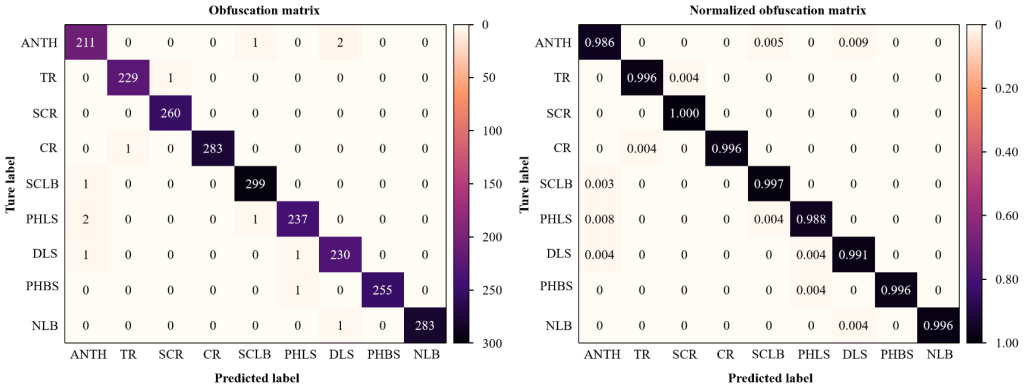


Figure 8. Obfuscation matrix analysis of classification based on transfer learning and data augmentation. The left is the obfuscation matrix, and the right is the normalized obfuscation matrix.

After the analysis and statistics of the confounding matrix, each parameter reflecting the model performance is obtained, as shown in Table 5, which describes the more detailed original and testing classification information of the proposed VGNet. It can be found in Table 5 that the precision and recall values of each disease type are different, which is related to the characteristic types and image numbers of each disease. The precision value in Table 5 is between 98.1% and 100%. The recall value ranges from 98.6% to 100%. The F1 value ranges from 98.4% to 99.8%, with an average accuracy of 99.4%. This indicates that the proposed method performs well in the established dataset after transfer learning and fine-tuning training, which could be applied to the actual detection of crop diseases in the field environment.

Table 5. Obfuscation matrix statistics for nine types of corn diseases with transfer learning and augmentation.

Types	ANTH	TR	SCR	CR	SLB	PHLS	DLS	PHBS	NLB
Samples	1070	1150	1300	1420	1500	1200	1160	1280	1420
Positive	214	230	260	284	300	240	232	256	284
Negative	2086	2070	2040	2016	2000	2060	2068	2044	2016
TP	211	229	260	283	299	237	230	255	283
FN	3	1	0	1	1	3	2	1	1
TN	2076	2058	2027	2004	1988	2050	2057	2032	2004
FP	4	1	1	0	2	2	3	0	0
Pre (%)	98.1	99.6	99.6	100.0	99.3	99.2	98.7	100.0	100.0
Rec (%)	98.6	99.6	100.0	99.7	99.7	98.8	99.1	99.6	99.7
F1 (%)	98.4	99.6	99.8	99.8	99.5	99.0	98.9	99.8	99.8
Acc (%)					99.4				

4.2. Comparison with State-of-the-Art Methods

To further validate the effect of our method based on fine-tuning training and VGNet, we compared the proposed method with the traditional machine learning classifiers and state-of-the-art models (deep learning methods), respectively, under the same experiment conditions as well as the same dataset. The total number of images was 1150. Traditional machine learning methods include random forest (RF) classification algorithm, support vector machine (SVM), and BP neural network. AlexNet, ResNet50, Inception v3, and the original VGG16 Net are the selected deep convolutional neural networks for the comparative experiment. For conventional machine learning methods, we preprocessed the corn disease images, including image enhancement, segmentation, and feature extraction. After removing background information, the disease spots with clear boundaries were obtained. Then color histogram feature in HSV color space and the matrix characteristics in RGB color space were extracted, respectively. The gray-level co-occurrence matrix was used for texture features and a seven-hue invariant matrix was used for shape feature extraction. Then, the extracted features were fused as input vectors of the BP, SVM, and RF classifiers. The

learning experiments of AlexNet, ResNet50, Inception v3, the original VGG16, and VGNet models adopt the method of transfer learning and fine-tuning mechanism. The experiment parameters were consistent with the proposed method. After training, the models were test tested and identification results were output. The accuracies obtained from different traditional machine learning classifiers and deep learning methods are shown in Figure 9. It can be seen in Figure 9 that the accuracies of traditional methods are generally lower than 87%. In addition, conventional classifiers often require tedious preprocesses involving image enhancement, segmentation, and extraction of features manually. In deep learning methods, the accuracies are greater than 92%, and they vary because of the different deep structures and abilities of feature extraction. The accuracy of AlexNet is the lowest among the five deep architectures, because the structure of AlexNet is shallower than others, which leads to the insufficient ability to extract the features of corn disease images. The accuracy of the original VGG16 Net is 94.78%, the ResNet50 is 95.22%, and Inception v3 achieves an accuracy of 96.96%. Experimental results indicate that deep learning methods are superior to conventional machine learning. It can also be seen that our model reaches a highest accuracy of 98.26%, which is improved by 3.48% compared with the original VGG16 Net. The addition of BN, a GAP layer, and L2 normalization makes the VGG16 Net more robust with higher accuracy. The improvement of our method based on the classical VGG16 Net has the capability to learn more complex features, as more convolutional layers are in the stack with smaller filter sizes compared with other deep learning models.

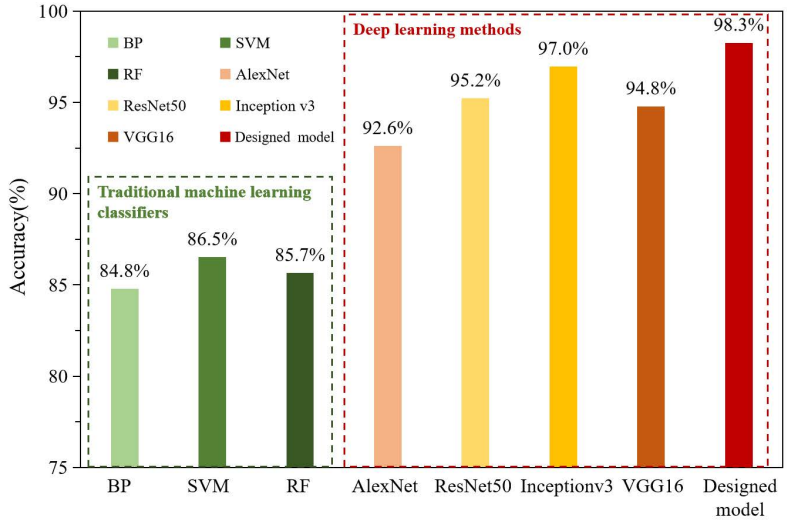


Figure 9. Comparison of accuracy between different models based on the same dataset.

Table 6 shows the comparative parameters and testing time of different deep learning methods. From Table 6, we can see that the original VGG16 Net has the most parameters and the longest testing time. AlexNet has eight weight layers and 58.3 million parameters; the testing time of AlexNet is the shortest, only 50.14 s for 230 images. However, the accuracy of AlexNet is the lowest (Figure 9). The parameters and testing time of ResNet50 and Inception v3 are slightly different. Our VGNet has 14 weight layers and 22.9 million parameters after replacing huge hidden fully connected layers by a GAP layer, and it only occupies 79.5 MB of memory space. The testing time of our model is only 75.21 s for 230 images, which improves by 151.11 s compared with the original VGG16 Net. In addition, the loss value of the designed VGNet is only 0.035, which is significantly smaller than other models, such as VGG16 and ResNet50. The proposed method can achieve real-time detection of corn diseases. In general, our proposed method has the best recognition effect after transfer learning and fine-tuning. The utilization of the GAP layer realized the feature dimension reduction. The parameters of the network were greatly reduced, as well as the calculation amount. This means the network regularization in the structure to prevent overfitting. The connections between each category in the feature map are more

intuitive (compared with the fully connected layers), and it is easier for the feature map to be converted into classification probability. Thus, the proposed VGNet is lightweight and robust, which could obtain the best performance among the state-of-the-art models.

Table 6. Comparison of the classic convolutional neural networks and corresponding parameters.

Methods	Network Layers	Parameters (Millions)	Weights (MB)	Times (s)	Loss Value
AlexNet	8	60.9	224	50.14	0.912
ResNet50	50	25.5	102	88.78	0.587
InceptionV3	46	24.7	96	86.02	0.271
VGG16	16	138	533	226.32	0.196
VGNet	14	22.9	79.5	75.21	0.035

Actually, our method utilizes 1150 corn disease images from field conditions, and the recognition accuracy reaches 98.3%, which is better than the models learning from scratch. After data augmentation, the accuracy of the model improves slightly by 1.2%. The dataset in this research is small compared with many deep convolutional models. Actually, Ferentinos et al. collected 87,848 images of plant diseases to train a convolutional neural network model, whose performance finally reached 99.5% accuracy [43]. In our experiment, when the dataset is enlarged to 11,500, the accuracy of VGNet increases to 99.4%. Compared with the study of Ferentinos, our success rate is only 0.1% lower than that of the model using 86,000 images. Thus, transfer learning seems to be an ideal method for the CNN model to achieve better performance. With the aid of the parameters transfer of the pretrained model, a more accurate model can be generated when fine-tuning several layers for disease image classification.

Three types of open large datasets, including ImageNet, PlantVillage, and AI Challenger, were used, and the results show that the models pretrained with PlantVillage or AI Challenger were better than that pretrained ones with ImageNet. The similarity of the training data to the experimental data results in easier transferability. The SGD algorithm and Adam optimizer are compared and analyzed in the fine-tuning phase. The experiments prove that the Adam optimizer for training the VGG16 Net is more accurate and more stable than the SGD algorithm. The initial learning rate is also an important parameter in model training. In regard to the pretrained model, smaller learning rates for convolutional nets are common, as network parameters should not be changed dramatically.

4.3. Feature Visualization

The ability of automatic feature extraction is an important factor to reflect the performance of the model. To examine the effect of feature extraction on the proposed model, feature map visualization was carried out. Figure 10 illustrates the original input image and the feature maps derived from the pooling layer of the model. From the right of Figure 10, we find that the disease spots were abstracted high-dimensional features; the VGNet obviously had high-quality feature extraction, which was beneficial for recognition and classification.

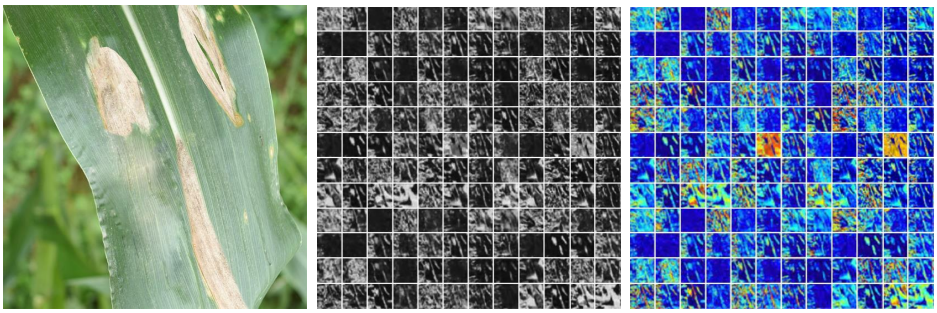


Figure 10. Obfuscation matrix analysis of classification based on transfer learning and data augmentation. The left is the original image; the middle is the grey feature map; and the right is the color feature map.

5. Conclusions

Data diversity and representativeness are the key elements to ensure the generalization of the model. In this paper, we devised a VGNet which takes VGG16 as the backbone and adds batch normalization, as well as replacing two fully connected layers with a GPA layer and adding L2 normalization. The parameters of the convolutional layers and pooling layers are transferred to the newly designed VGNet; then, the fine-tuning learning for VGNet is studied to enhance the ability of recognizing corn disease images from real field conditions.

Data augmentation has greater promotion of model learning from scratch than on pretrained model, because the parameters of pretrained models are trained enough by open large datasets. Compared with traditional machine learning methods and state-of-the-art deep learning methods, the proposed VGNet has a stronger ability to identify a hierarchy of features of corn diseases. The accuracy of VGNet is improved by 3.5% compared with the original VGG16 Net, and the testing time for 230 images is reduced by 66.8%, with balanced precision, recall, and F1 indexes. The parameters and memory occupation of the proposed VGNet are reduced by 83.4% and 85.1%, respectively. The comparative experiments and performance analysis illustrated the wide adaptability of the proposed method. In addition, the proposed method could provide baseline architecture for other types of phenotypic information recognition or interpretation with much fewer parameters and computation time. In future work, we will focus on collecting multiple crop disease images from real scenes and developing fine-grained disease detection methods that can be used for multiple categories of crops.

Author Contributions: Conceptualization, Z.G.; methodology, X.F. and Z.G.; software, X.F.; validation, X.F.; visualization, Z.G.; writing—original draft, X.F.; writing—review and editing, Z.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Major Science and Technology Projects under grant number 2022YFD2022205; in part by the Nanfan special project, CAAS under grant number YBXM10; in part by the Science and Technology Innovation Program of the Chinese Academy of Agricultural Sciences under grant number CAAS-ZDRW202107 and CAAS-ASTIP-2023-AII.

Data Availability Statement: The data presented in this study are available on request from the corresponding author or the first author.

Acknowledgments: The authors are grateful for the constructive advice on the revision of this manuscript from anonymous reviewers.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Xu, Y.; Wang, X.; Zhai, Y.; Li, C.; Gao, Z. Precise variable spraying system based on improved genetic proportional-integral-derivative control algorithm. *Trans. Inst. Meas. Control* **2021**, *43*, 3255–3266. [CrossRef]
2. Roshan, S.H.; Kazemitabar, S.J.; Kheradmandian, G. Artificial Intelligence Aided Agricultural Sensors for Plant Frostbite Protection. *Appl. Artif. Intell.* **2022**, *36*, 2031814. [CrossRef]
3. Xu, Y.; Xue, X.; Sun, Z.; Gu, W.; Cui, L.; Jin, Y.; Lan, Y. Joint path planning and scheduling for vehicle-assisted multiple Unmanned Aerial Systems plant protection operation. *Comput. Electron. Agric.* **2022**, *200*, 107221. [CrossRef]
4. Godara, S.; Toshniwal, D.; Bana, R.S.; Singh, D.; Bedi, J.; Parsad, R.; Dabas, J.P.S.; Jhajhria, A.; Godara, S.; Kumar, R.; et al. AgrIntel: Spatio-temporal profiling of nationwide plant-protection problems using helpline data. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105555. [CrossRef]
5. Li, Z.; Wang, W.; Zhang, C.; Zheng, Q.; Liu, L. Fault-tolerant control based on fractional sliding mode: Crawler plant protection robot. *Comput. Electr. Eng.* **2023**, *105*, 108527. [CrossRef]
6. Tang, Y.; Fu, Y.; Guo, Q.; Huang, H.; Tan, Z.; Luo, S. Numerical simulation of the spatial and temporal distributions of the downwash airflow and spray field of a co-axial eight-rotor plant protection UAV in hover. *Comput. Electron. Agric.* **2023**, *206*, 107634. [CrossRef]
7. Liu, Y.; Gao, G.; Zhang, Z. Crop Disease Recognition Based on Modified Light-Weight CNN With Attention Mechanism. *IEEE Access* **2022**, *10*, 112066–112075. [CrossRef]
8. Haque, M.A.; Marwaha, S.; Deb, C.K.; Nigam, S.; Arora, A. Recognition of diseases of maize crop using deep learning models. *Neural Comput. Appl.* **2023**, *35*, 7407–7421. [CrossRef]

9. Hua, J.; Zhu, T.; Liu, J. Leaf Classification for Crop Pests and Diseases in the Compressed Domain. *Sensors* **2023**, *23*, 48. [CrossRef]
10. Kurmi, Y.; Gangwar, S.; Chaurasia, V.; Goel, A. Leaf images classification for the crops diseases detection. *Multimed. Tools Appl.* **2022**, *81*, 8155–8178. [CrossRef]
11. Youk, G.; Kim, M. Transformer-Based Synthetic-to-Measured SAR Image Translation via Learning of Representational Features. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–18. [CrossRef]
12. Naseem, A.; Rehman, M.A.; Qureshi, S.; Ide, N. Graphical and Numerical Study of a Newly Developed Root-Finding Algorithm and Its Engineering Applications. *IEEE Access* **2023**, *11*, 2375–2383. [CrossRef]
13. Chakraborty, S.K.; Chandel, N.S.; Jat, D.; Tiwari, M.K.; Rajwade, Y.A.; Subeesh, A. Deep learning approaches and interventions for futuristic engineering in agriculture. *Neural Comput. Appl.* **2022**, *34*, 20539–20573. [CrossRef]
14. Liang, D.; Tang, W.; Fu, Y. Sustainable Modern Agricultural Technology Assessment by a Multistakeholder Transdisciplinary Approach. *IEEE Trans. Eng. Manag.* **2023**, *70*, 1061–1075. [CrossRef]
15. Tu, J.; Aznoli, F.; Navimipour, N.J.; Yalçin, S. A new service recommendation method for agricultural industries in the fog-based Internet of Things environment using a hybrid meta-heuristic algorithm. *Comput. Ind. Eng.* **2022**, *172*, 108605. [CrossRef]
16. Almadani, B.; Mostafa, S.M. IIoT Based Multimodal Communication Model for Agriculture and Agro-Industries. *IEEE Access* **2021**, *9*, 10070–10088. [CrossRef]
17. Wang, F.; Yang, J.; Wang, X.; Li, J.; Han, Q. Chat with ChatGPT on Industry 5.0: Learning and Decision-Making for Intelligent Industries. *IEEE CAA J. Autom. Sinica* **2023**, *10*, 831–834. [CrossRef]
18. Jia, M.; Li, J.; Zhang, J.; Wei, N.; Yin, Y.; Chen, H.; Yan, S.; Wang, Y. Identification and validation of cuproptosis related genes and signature markers in bronchopulmonary dysplasia disease using bioinformatics analysis and machine learning. *BMC Med. Inform. Decis. Mak.* **2023**, *23*, 69. [CrossRef]
19. Liu, C.; Zhou, Y.; Zhou, Y.; Tang, X.; Tang, L.; Wang, J. Identification of crucial genes for predicting the risk of atherosclerosis with system lupus erythematosus based on comprehensive bioinformatics analysis and machine learning. *Comput. Biol. Med.* **2023**, *152*, 106388. [CrossRef]
20. Bacon, W.; Holinski, A.; Pujol, M.; Wilmott, M.; Morgan, S.L. Correction: Ten simple rules for leveraging virtual interaction to build higher-level learning into bioinformatics short courses. *PLoS Comput. Biol.* **2023**, *19*, e1010964. [CrossRef]
21. Ramana, K.; Aluvala, R.; Kumar, M.R.; Nagaraja, G.; Krishna, A.V.; Nagendra, P. Leaf Disease Classification in Smart Agriculture Using Deep Neural Network Architecture and IoT. *J. Circuits Syst. Comput.* **2022**, *31*, 2240004:1–2240004:27. [CrossRef]
22. Bajpai, C.; Sahu, R.; Naik, K.J. Deep learning model for plant-leaf disease detection in precision agriculture. *Int. J. Intell. Syst. Technol. Appl.* **2023**, *21*, 72–91. [CrossRef]
23. Pal, A.; Kumar, V. AgriDet: Plant Leaf Disease severity classification using agriculture detection framework. *Eng. Appl. Artif. Intell.* **2023**, *119*, 105754. [CrossRef]
24. Kwaghtyo, D.K.; Eke, C.I. Smart farming prediction models for precision agriculture: A comprehensive survey. *Artif. Intell. Rev.* **2023**, *56*, 5729–5772. [CrossRef]
25. Surampudi, S.; Kumar, V. Flood Depth Estimation in Agricultural Lands From L and C-Band Synthetic Aperture Radar Images and Digital Elevation Model. *IEEE Access* **2023**, *11*, 3241–3256. [CrossRef]
26. Tong, K.; Wu, Y.; Zhou, F. Recent advances in small object detection based on deep learning: A review. *Image Vis. Comput.* **2020**, *97*, 103910. [CrossRef]
27. Zhao, X.; Zhang, J.; Huang, Y.; Tian, Y.; Yuan, L. Detection and discrimination of disease and insect stress of tea plants using hyperspectral imaging combined with wavelet analysis. *Comput. Electron. Agric.* **2022**, *193*, 106717. [CrossRef]
28. Vishnoi, V.K.; Kumar, K.; Kumar, B.; Mohan, S.; Khan, A.A. Detection of Apple Plant Diseases Using Leaf Images Through Convolutional Neural Network. *IEEE Access* **2023**, *11*, 6594–6609. [CrossRef]
29. Prabhakar, M.L.C.; Merina, R.D.; Mani, V. IoT Based Air Quality Monitoring and Plant Disease Detection for Agriculture. *Autom. Control Comput. Sci.* **2023**, *57*, 115–122. [CrossRef]
30. Amrani, A.; Sohel, F.; Diepeveen, D.; Murray, D.; Jones, M.G.K. Deep learning-based detection of aphid colonies on plants from a reconstructed *Brassica* image dataset. *Comput. Electron. Agric.* **2023**, *205*, 107587. [CrossRef]
31. Quach, B.M.; Cuong, D.V.; Pham, N.; Huynh, D.; Nguyen, B.T. Leaf recognition using convolutional neural networks based features. *Multimed. Tools Appl.* **2023**, *82*, 777–801. [CrossRef]
32. Lv, Z.; Zhang, Z. Research on plant leaf recognition method based on multi-feature fusion in different partition blocks. *Digit. Signal Process.* **2023**, *134*, 103907. [CrossRef]
33. Jin, H.; Li, Y.; Qi, J.; Feng, J.; Tian, D.; Mu, W. GrapeGAN: Unsupervised image enhancement for improved grape leaf disease recognition. *Comput. Electron. Agric.* **2022**, *198*, 107055. [CrossRef]
34. Laxmi, S.; Gupta, S.K. Multi-category intuitionistic fuzzy twin support vector machines with an application to plant leaf recognition. *Eng. Appl. Artif. Intell.* **2022**, *110*, 104687. [CrossRef]
35. Reddy, S.R.G.; Varma, G.P.S.; Davuluri, R.L. Resnet-based modified red deer optimization with DLCNN classifier for plant disease identification and classification. *Comput. Electr. Eng.* **2023**, *105*, 108492. [CrossRef]
36. Janani, M.; Jebakumar, R. Detection and classification of groundnut leaf nutrient level extraction in RGB images. *Adv. Eng. Softw.* **2023**, *175*, 103320. [CrossRef]
37. Kumar, R.R.; Athimoolam, J.; Appathurai, A.; Rajendiran, S. Novel segmentation and classification algorithm for detection of tomato leaf disease. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7674. [CrossRef]

38. Cui, S.; Su, Y.L.; Duan, K.; Liu, Y. Maize leaf disease classification using CBAM and lightweight Autoencoder network. *J. Ambient Intell. Humaniz. Comput.* **2023**, *14*, 7297–7307. [CrossRef]
39. Charak, A.S.; Sinha, A.; Jain, T. Novel approach for quantification for severity estimation of blight diseases on leaves of tomato plant. *Expert Syst. J. Knowl. Eng.* **2023**, *40*, e13174. [CrossRef]
40. Ashwini, C.; Sellam, V. EOS-3D-DCNN: Ebola optimization search-based 3D-dense convolutional neural network for corn leaf disease prediction. *Neural Comput. Appl.* **2023**, *35*, 11125–11139. [CrossRef]
41. Cen, H.; Zhu, Y.; Sun, D.; Zhai, L.; He, Y. Current status and future perspective of the application of deep learning in plant phenotype research. *Trans. Chin. Soc. Agric. Eng.* **2020**, *36*, 1–16.
42. Zhang, J.; Rao, Y.; Man, C.; Jiang, Z.; Li, S. Identification of cucumber leaf diseases using deep learning and small sample size for agricultural Internet of Things. *Int. J. Distrib. Sens. Netw.* **2021**, *17*, 155014772110074. [CrossRef]
43. Ferentinos, K.P. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318. [CrossRef]
44. Liang, Q.; Xiang, S.; Hu, Y.; Coppola, G.; Zhang, D.; Sun, W. PD^{2}SE-Net: Computer-assisted plant disease diagnosis and severity estimation network. *Comput. Electron. Agric.* **2019**, *157*, 518–529. [CrossRef]
45. AI Challenger 2018. Available online: https://github.com/AIChallenger/AI_Challenger_2018 (accessed on 6 December 2022)
46. Zhong, Y.; Zhao, M. Research on deep learning in apple leaf disease recognition. *Comput. Electron. Agric.* **2020**, *168*. [CrossRef]
47. He, Y.; Zeng, H.; Fan, Y.; Ji, S.; Wu, J. Application of Deep Learning in Integrated Pest Management: A Real-Time System for Detection and Diagnosis of Oilseed Rape Pests. *Mob. Inf. Syst.* **2019**, *2019*, 4570808:1–4570808:14. [CrossRef]
48. Zeng, W.; Li, M. Crop leaf disease recognition based on Self-Attention convolutional neural network. *Comput. Electron. Agric.* **2020**, *172*, 105341. [CrossRef]
49. Hughes, D.P.; Salathé, M. An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing. *arXiv* **2015**, arXiv:1511.08060.
50. Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using Deep Learning for Image-Based Plant Disease Detection. *Front Plant Sci.* **2016**, *7*, 1419. [CrossRef]
51. Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
52. Waheed, A.; Goyal, M.; Gupta, D.; Khanna, A.; Hassanien, A.E.; Pandey, H.M. An optimized dense convolutional neural network model for disease recognition and classification in corn leaf. *Comput. Electron. Agric.* **2020**, *175*, 105456. [CrossRef]
53. Hsiao, T.; Chang, Y.; Chou, H.; Chiu, C. Filter-based deep-compression with global average pooling for convolutional networks. *J. Syst. Archit.* **2019**, *95*, 9–18. [CrossRef]
54. Kaya, A.; Keçeli, A.S.; Catal, C.; Yalic, H.Y.; Temuçin, H.; Tekinerdogan, B. Analysis of transfer learning for deep neural network based plant classification models. *Comput. Electron. Agric.* **2019**, *158*, 20–29. [CrossRef]
55. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.