Computers and Electronics in Agriculture 173 (2020) 105393

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Using deep transfer learning for image-based plant disease identification

Junde Chen[a], Jinxiu Chen[a], Defu Zhang[a,*], Yuandong Sun[b], Y.A. Nanehkaran[a]

[a] School of Informatics, Xiamen University, Xiamen 361005, China
[b] Spira Inc, Los Angeles, CA 90032, USA

ARTICLE INFO

Keywords:
Plant disease identification
Deep learning
Convolution neural networks
Transfer learning
Image classification

ABSTRACT

Plant diseases have a disastrous impact on the safety of food production, and they can cause a significant reduction in both the quality and quantity of agricultural products. In severe cases, plant diseases may even cause no grain harvest entirely. Thus, the automatic identification and diagnosis of plant diseases are highly desired in the field of agricultural information. Many methods have been proposed for solving this task, where deep learning is becoming the preferred method due to the impressive performance. In this work, we study transfer learning of the deep convolutional neural networks for the identification of plant leaf diseases and consider using the pre-trained model learned from the typical massive datasets, and then transfer to the specific task trained by our own data. The VGGNet pre-trained on ImageNet and Inception module are selected in our approach. Instead of starting the training from scratch by randomly initializing the weights, we initialize the weights using the pre-trained networks on the large labeled dataset, ImageNet. The proposed approach presents a substantial performance improvement with respect to other state-of-the-art methods; it achieves a validation accuracy of no less than 91.83% on the public dataset. Even under complex background conditions, the average accuracy of the proposed approach reaches 92.00% for the class prediction of rice plant images. Experimental results demonstrate the validity of the proposed approach, and it is accomplished efficiently for plant disease detection.

1. Introduction

The occurrence of plant diseases has negative effects on agricultural production, and if the plant diseases are not detected in time, there will be an increase in food insecurity (Faithpraise et al., 2013). In particular, the main crops such as rice, maize, etc., are essential for guaranteeing the food supply and agricultural production. The early warning and forecast are the basis of effective prevention and control for plant diseases. They play crucial roles in the management and decision-making for agricultural production. Until now, however, the visual observations of experienced producers are still the primary approach for plant disease detection in rural areas of developing countries; this requires continuous monitoring of experts, which might be prohibitively expensive in large farms (Al-Hiary et al., 2011; Bai et al., 2018). Besides, in some remote areas, farmers may have to go long distances to contact experts, which makes the consulting too expensive and time-consuming. Nevertheless, this approach can only be done in limited areas and cannot be well extended. Automatic recognition of plant diseases is an essential research topic, as it may prove benefits in monitoring large fields of crops, and thus automatically detect the symptoms of diseases as soon as they appear on plant leaves (Al Bashish et al., 2011; Pooja et al., 2017; Khirade and Patil, 2015). Therefore, looking for a fast, automatic, less expensive, and accurate method to perform plant disease detection is of great realistic significance.

Plenty of previous works have considered the image recognition, and a particular classifier is used which categorizes the images into healthy or diseased images. Generally, the leaves of plants are the first source of plant disease identification, and the symptoms of most diseases may start to appear on the leaves (Ebrahimi et al., 2017; García et al., 2017). In the past decades, the primary classification techniques that were popularly used for disease identification in plants include k-nearest neighbor (KNN) (Guettari et al., 2016), support vector machine (SVM) (Deepa and Umarani, 2017), fisher linear discriminant (FLD) (Ramezani and Ghaemmaghami, 2010), artificial neural network(ANN) (Sheikhan et al., 2012), random forest (RF) (Kodovsky et al., 2011) and so on. As we all know that the disease recognition rates of the classical approaches rely heavily on the lesion segmentation and hand-designed features by various algorithms, such as seven invariant moments, SIFT (Scale-invariant feature transform), Gabor transform, global–local singular value, and sparse representation (Guo et al., 2007; Zhang and Wang, 2016; Zhang et al., 2017), etc. However, the artificially-designed features require expensive works and expert knowledge, which have a

* Corresponding author.
E-mail address: dfzhang@xmu.edu.cn (D. Zhang).

https://doi.org/10.1016/j.compag.2020.105393
Received 3 November 2019; Received in revised form 27 March 2020; Accepted 29 March 2020
Available online 06 April 2020

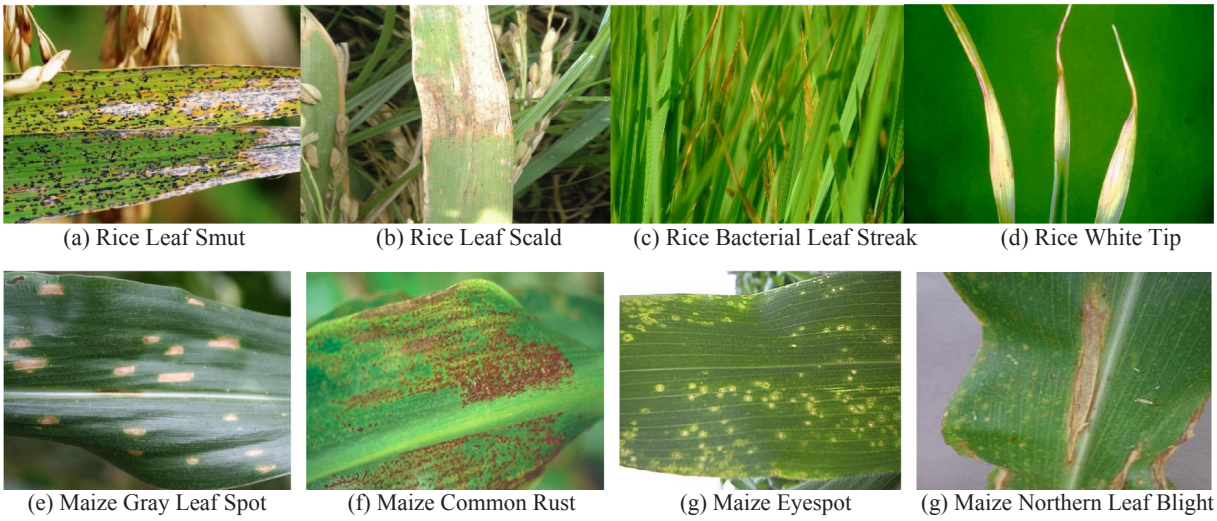


Fig. 1. Sample images of plant diseases.

certain subjectivity. Mainly, it is not easy to determine which features are optimal and robust for disease identification from the many extracted features. Besides, under the complex background conditions, most methods fail to effectively segment the leaf and corresponding lesion image from its background, which will lead to unreliable disease recognition results. Thus, the automatic recognition of plant disease images is still a challenging task due to the complexity of diseased leaf images. More recently, deep learning techniques, particularly convolutional neural networks (CNNs), are quickly becoming the preferred methods to overcome some challenges (Barbedo, 2018). CNN is the most popular classifier for image recognition in both large and small scale problems. It has shown outstanding ability in image processing and classification (Kamilaris and Prenafeta-Boldú, 2018; Kussul et al., 2017; Yalcin, 2017). For example, Mohanty, Hughes and Marcel (Mohanty et al., 2016) trained a deep learning model for recognizing 14 crop species and 26 crop diseases. Their trained model achieves an accuracy of 99.35% on a held-out test set. Ma et al. (Ma et al., 2018) used a deep CNN to conduct symptom-wise recognition of four cucumber diseases, i.e., downy mildew, anthracnose, powdery mildew, and target leaf spots. They reached the recognition accuracy of 93.4%. Kawasaki et al. (Kawasaki et al., 2015) introduced a system based on CNN to recognize cucumber leaf disease; it realizes an accuracy of 94.9%, etc. Although very good results have been reported in the literature, investigations so far have used image databases with limited diversity. The most photographic materials include images solely in experimental (laboratory) setups, not in real field wild scenarios. Indeed, images captured in cultivation field conditions include a wide diversity of background and an extensive variety of symptom characteristics (Barbedo, 2018). Additionally, there are a vast number of parameters needed to be trained for CNN and its variants, while training these CNN architectures also requires multiple labeled samples and substantial computer resources from scratch to assess their performance. Collecting a large labeled dataset is undoubtedly a challenging task. Despite the limitations, the previous investigations have successfully demonstrated the potential of deep learning algorithms. Particularly, the deep transfer learning, which alleviates the problem faced by classical deep learning methods, i.e. the solutions consisting of using a pre-trained network where only the parameters of the last classification levels need to be inferred from scratch (Kessentini et al., 2019); is naturally employed in the practical application. In this work, we study the transfer learning for the deep CNNs with the aim of enhancing the learning ability of tiny lesion symptoms along with decreasing the computational complexity. The proposed approach is generally composed of two parts: the first part is the pre-trained module, which is used as a basic feature extractor; the other is an auxiliary structure that utilizes multi-scale feature maps for detection. Specifically, the enhanced VGGNet with the Inception module is selected in our approach. The conventional VGGNet is modified by replacing its last convolutional layers with an extended convolutional layer of $3 \times 3 \times 512$, where the batch normalization is added and the Swish activation function is used to directly replace ReLu. Then, the convolutional layer is followed by two Inception modules, which are used to extract the multi-scale features of images input from the previous layer, and the fully connected layers are replaced by a global pooling layer to conduct the dimension reduction of feature maps. After that, a fully-connected Softmax layer with a practical number of categories was added as the top layer of the modified networks. In this way, the newly formed network, which we term the INC-VGGN, is used for the class prediction of plant disease images.

The remainder of this paper is organized in the following manner. Section 2 introduces the collection of the image dataset followed by an overall flow summary, and this section mainly discusses the methodology to accomplish the task of plant disease identification along with related concepts and the proposed approach. Section 3 dedicates to the algorithm experiments; multiple experiments are conducted and the experimental results are evaluated as well as comparative analysis. Finally, the paper is summarized in Section 4.

2. Materials and methods

2.1. Data acquisition

Including 500 rice images and 466 maize images, about 1000 crop leaf images were provided by the Fujian Institute of Subtropical Botany, Xiamen, China. The images were captured under non-uniform illumination intensities and clutter field background conditions. All the crop images collected in this paper are as the disease categories, also saved as the JPG format. Among them, the rice diseases include the Rice Stackburn, Rice Leaf Scald, Rice Leaf Smut, Rice White Tip, and Bacterial Leaf Streak. The maize diseases consist of Phaeosphaeria Spot, Maize Eyespot, Gray Leaf Spot, and Goss's Bacterial wilt, etc. For the subsequent calculations, these images are uniformly processed into the RGB model by Photoshop tools firstly, and then the sizes of images are adjusted to 224×224 pixels. Some of the sample images are displayed in Fig. 1.

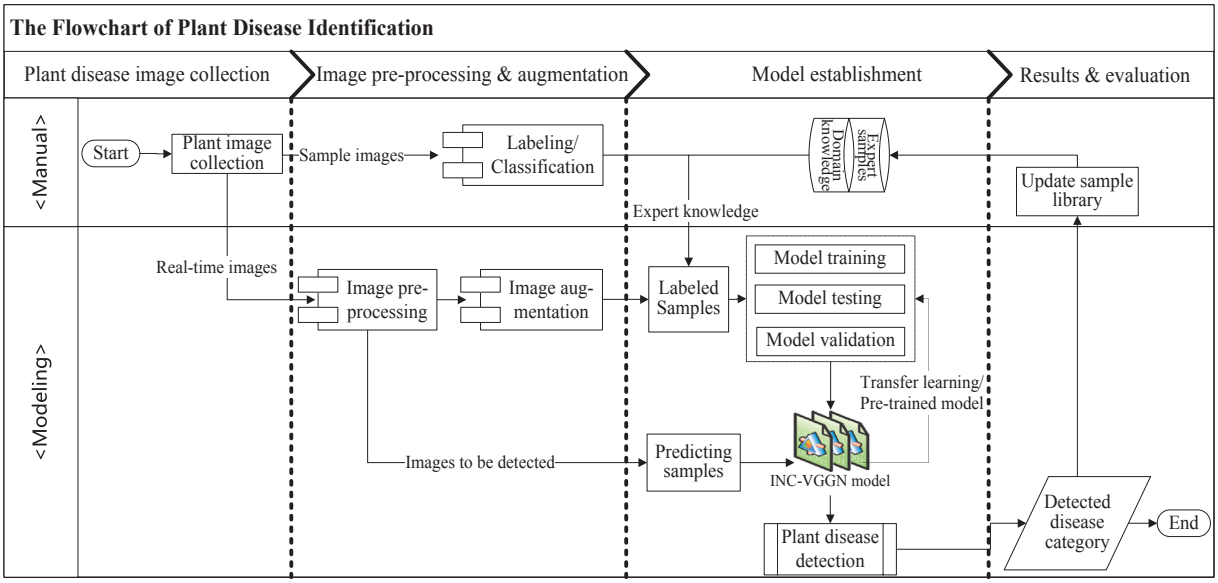


Fig. 2. The overall flow of plant disease identification.

2.2. Overview

As depicted in Fig. 2, a general overview of our method for plant disease identification is presented as follows. Firstly, the samples of plant disease images are collected and labeled based on the knowledge of experts in the field. Then, the image-processing techniques including grey transformation, image filtering, image sharpening and resizing, etc., are performed on the acquired images, and new sample images are generated to enrich the dataset using the data augmentation methods. For example, random rotation, flipping, and translation are utilized to enlarge the dataset. After that, the sample images are input to the proposed method (aliased as INC-VGGN) for model training. Thus, the trained model is applied for the class prediction of unseen images, and the results of plant disease identification are obtained eventually. The detailed descriptions of these phases are illustrated in subsequent sections.

2.3. Related works

2.3.1. Convolutional neural networks

Convolutional neural networks are a category of neural networks designed for image recognition and classification and have achieved excellent results (Huang et al., 2017; Szegedy et al., 2015; Cetinic et al., 2018). Different from the traditional approaches, CNNs can learn high-level robust features directly from the original image instead of extracting the specific features manually. In the identification of plant species and diseases, it is demonstrated that CNNs can provide better performance than the traditional feature extraction methods (Mohanty et al., 2016; Dyrmann et al., 2016). A typical CNN architecture mainly consists of convolution layers, pooling layers, and full connection layers (Lu et al., 2017), which are described as follows.

1 Convolutional layers

The convolutional layer is the crucial component of CNN, which extracts the specific features of the image by the different sizes of the convolution kernel. A set of feature maps of input images can be extracted after applying the convolutional layers several times. Let H_i represent the feature map of the i-th layer in CNN, then the H_i can be generated as follows:

$$H_i = \varphi(H_{i-1}W_i + b_i) \tag{1}$$

where H_i denotes the feature map of the current network layer, H_{i-1} is the convolution feature of the previous layer (H_0 is the original image.), W_i is the weight of the i-th layer, b_i is the offset vector of the i-th layer, and $\varphi(\cdot)$ represents the rectified linear unit (ReLU) function.

2 Pooling layers

The function of pooling layers is reducing the spatial dimension, which can reduce computational complexity and effectively control the risk of over-fitting. In the l-th pooling layer, the output feature on the j-th local receptive field can be calculated in Eq. (2).

$$x_j^l = down(x_j^{l-1}, s) \tag{2}$$

where down(·) represents the down-sampling function, x_j^{l-1} is the feature vector in the previous layer, and s is the pooling size.

3 Fully-connected layers

After convolutional and pooling layers, there are one or several fully-connected (FC) layers, and the purpose of FC layers is using the extracted features for image classification. The Softmax function is usually employed to conduct the class prediction with the features extracted from the previous layers. Mathematically, the Softmax function is written by

$$\text{softmax}(z)_j = e^{z_j} / \sum_{k=1}^{K} e^{z_k} (for\ j = 1, ..., K) \tag{3}$$

where K represents the dimension of the z vector.

2.3.2. Vggnet

VGGNet (Simonyan and Zisserman, 2014) is a kind of convolutional neural networks developed by researchers from the Visual Geometry Group of Oxford University and Google DeepMind. It consists of 16 or more convolution, pooling and fully-connected layers (Khan et al., 2019), as shown in Fig. 3. VGGNet adopts the cascaded network structure, and there is a pooling layer after 2 or 3 convolution layers using small 3 × 3 convolution filters. The VGGNet includes two typical models such as the VGG-16 and VGG-19, which are available as the pre-trained models with the 16 and 19 wt layers trained on the ImageNet (Russakovsky et al., 2015) dataset. Thus, in our experiments, the VGG-19 is considered as the base model and modified to generate the new networks, and then the labeled sample images are exploited to train and

Fig. 3. The basic architecture of VGGNet (Khan et al., 2019).

fine-tune the formed networks. The VGGNet structure is depicted as follows.

2.3.3. Transfer learning

Transfer learning is a machine learning approach in which CNNs trained for a task is reused as the starting point for a model on a second task (Lumini and Nanni, 2019). Instead of starting the training from scratch by randomly initializing the weights, we can initialize the weights using a pre-trained network on large labeled datasets, such as public image datasets, etc. In this paper, we consider using the pre-trained models learned from the massive typical dataset ImageNet, and then transfer to the specific task trained by the objective dataset. The main processes of the transfer learning approach are described as follows.

1 Determine the base networks

Determine the base networks of transfer learning and assign the network weights (W_1, W_2, …, W_n) using the pre-trained CNN model. The weights of bottom layers can be downloaded from a well-trained CNN (https://ker-as.io/applications/).

2 Establish a new neural network

Based on the bottom layers, the network structure can be modified such as replacing the layers, inserting the layers and deleting the layers from the networks, etc. In this way, a new network structure can be generated.

3 Fine-tuning the neural network.

Using our own dataset X and corresponding labels Y, the newly formed networks can be fine-tuned to minimize the loss function E, as expressed in Eq. (4).

$$E(W) = -\frac{1}{n}\sum_{x_i=1}^{n}\sum_{k=1}^{K}[y_{ik}\log P(x_i = k) + (1 - y_{ik})\log(1 - P(x_i = k))] \tag{4}$$

where W represents the weighting matrices of the convolutional and fully-connected layers, n is the number of training samples, i is the index of training samples, and k is the index of classes. $P(x_i = k)$ is the probability of input x_i belonging to the predicted k-th class.

Specifically, the stochastic gradient descent (SGD) (Ghazi et al., 2017) algorithm is often used to calculate the optimal W by minimizing the loss function E on the target dataset, as defined in Eq. (5).

$$W_k = W_{k-1} - a(\partial E(W)/\partial W) \tag{5}$$

where a is the learning rate, and k is the index of classes.

Thereby, in this work, we chose the VGGNet pre-trained on ImageNet and Inception module for transfer learning, and trained the newly formed neural networks using our own datasets. The approach combines the advantages of the VGGNet and Inception module, and the

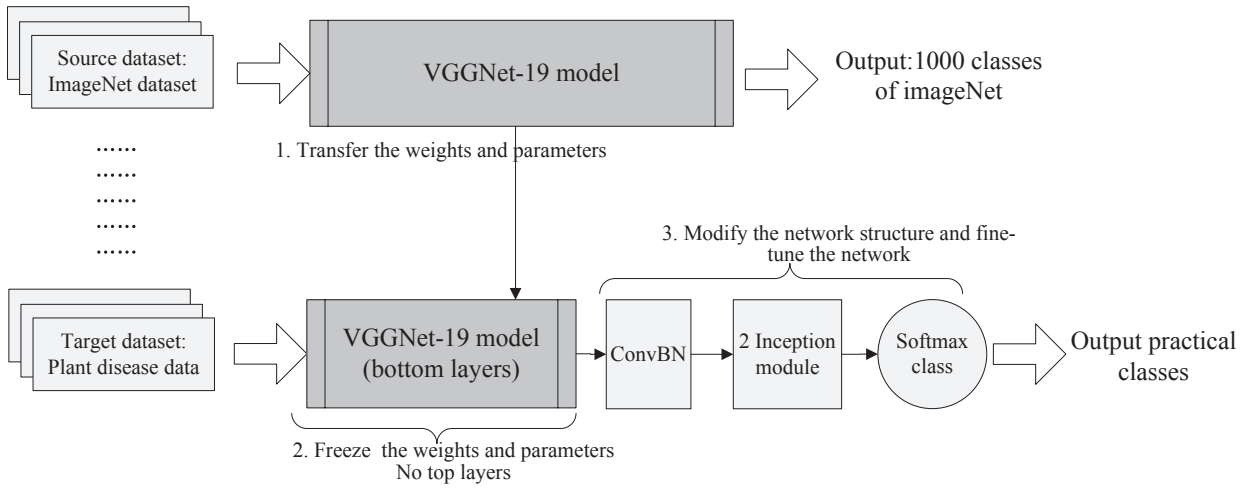


Fig. 4. The transfer learning process of VGGNet.

main processes are depicted in Fig. 4.

2.4. Proposed approach

As mentioned in Section 2.3.2, VGGNet represents the current state-of-the-art with powerful and accurate classification ability, and it is often used for transfer learning because the model is highly portable (Simonyan and Zisserman, 2014). Additionally, in ILSVRC 2014, the Inception module has shown its benefits for using in GoogleNet, which achieves impressive, record-breaking results in the competition (Chollet, 2017). Therefore, inspired by the performances, the VGGNet with the Inception module is selected in our approach. The conventional VGGNet was enhanced by replacing its last convolutional layers with an additional convolutional layer of $3 \times 3 \times 512$, then the batch normalization (BN) was added on the convolutional layer and the Swish activation function is used to directly replace ReLU (Ioffe and Szegedy, 2015; Ramachandran et al., 2017). With BN, it can get more than 2% improvement in mAP (Redmon and Farhadi, 2017), and test accuracy with Swish was found to be 0.9% higher than ReLU in ImageNet (Hung et al., 2019). Thus, the BN and Swish were both selected for being used in our networks. Furthermore, we modified the conventional VGGNet by replacing its full connection layers with a global pooling layer, and two Inception modules, as stated before, are introduced in the VGGNet to improve the feature extraction ability of the new network, namely, INC-VGGN.

More specifically, the details of this phase are illustrated as follows. The first few layers of a convolutional neural network typically extract the color and corner features (Zeiler and Fergus, 2014) and it is of little value to utilize Inception to extract these features. Therefore, the Conv1_1 to Pool3 layers of the VGGNet are preserved and the subsequent layers of VGGNet, Conv5_1 to Conv5_3, are replaced with two Inception modules and one added BN convolutional layer, in which the Swish is used as the activation function instead of the ReLU. That is, to enhance the multi-scale feature extraction ability of the network, a convolutional layer with batch normalization and Swish is added behind Pooling3, and then the two Inception modules composed of Inception and Concat are conducted. Finally, a global pooling layer instead of fully-connected layers are added after the second Inception module and followed by a Softmax classifier. Fig. 5 depicts the network structure and relevant parameters are displayed in Table 1.

Thus, the newly generated networks are generally composed of two parts: the first part is the pre-trained module, which is used as a basic feature extractor; the second part is the extended layers that extract the high-dimensional features and utilize the multi-scale feature maps for classification. On the other hand, the reason why we used a global pooling layer replacing the fully-connected layers is described below. In traditional CNN or DCNN, the ratio of parameters in all fully-connected layers is almost 80% of the whole network, which will increase the training and testing time, and lead to a large demand for computer memory (Zhang et al., 2019). Too many parameters can result in an overfitting problem as well. Particularly, the diverse images captured in real field wild scenarios contain a lot of noise such as heterogeneous background and uneven illumination, which is easy to be fitted by the complicated model and causes the over-fitting problem. Additionally, considering the number of model parameters, it is $(5 \times 5 + 1) \times C^2 = 26C^2$ generated by a 5×5 convolutional kernel in C input images, while this value is $2 \times (3 \times 3 + 1) \times C^2 = 20C^2$ for two consecutive 3×3 convolution kernel, so the model complexity can be reduced by this consecutive convolution architecture. Instead of a conventional Inception module, the enhanced Inception module is introduced in our networks. Thus, in this network architecture, the single 5×5 convolution layer is replaced with two consecutive 3×3 convolution layers, which not only maintains the range of perceptive fields but also reduces the number of parameters.

Table 1
Related parameters of the INC-VGGN model.

Layer name	Type	Kernel size	Stride	Neuron size	Maps
Conv1_1	Convolutional layer C1	3×3	1	224×224	64
Conv1_2	Convolutional layer C2	3×3	1	224×224	64
Pool1	Pooling layer P1	2×2	2	112×112	64
Conv2_1	Convolutional layer C3	3×3	1	112×112	128
Conv2_2	Convolutional layer C4	3×3	1	112×112	128
Pool2	Pooling layer P2	2×2	2	56×56	128
Conv3_1	Convolutional layer	3×3	1	56×56	256
Conv3_2	Convolutional layer	3×3	1	56×56	256
Conv3_3	Convolutional layer	3×3	1	56×56	256
Pool2	Pooling layer P2	2×2	2	28×28	256
Conv4_1	Convolutional layer	3×3	1	28×28	512
Conv4_2	Convolutional layer	3×3	1	28×28	512
Conv4_3	Convolutional layer	3×3	1	28×28	512
Pool3	Pooling layer P2	2×2	2	14×14	512
Conv2d_BN	Convolutional layer	3×3	1	14×14	512
Inception_1	Inception module	–	–	14×14	256
Inception_2	Inception module	–	–	14×14	256
Pool4	Global pooling	–	–	1×1	256
Output	Softmax regression	Classifier	–	1×1×5	5

3. Experimental results and analysis

In our experiments, some image pre-processing algorithms were conducted using Matlab, while the data augmentation and CNN were implemented using Anaconda3 (Python 3.6), Keras-GPU library (Keras-GPU. Available online: https://anaconda.org/anaconda/keras-gpu (accessed on 17 Jun 2019), and OpenCV-python3 library, etc. The deep CNN training and testing are accelerated by GPU, and the experimental

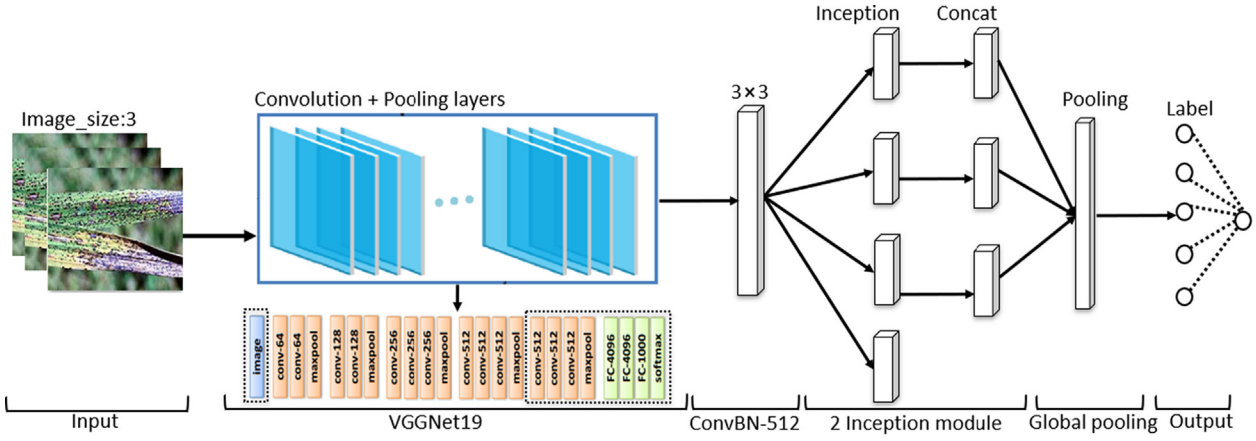


Fig. 5. Proposed INC-VGGN architecture.

hardware environment includes: Intel® Core™ i7-8750 central processing unit (CPU) at 2.20 GHz with 8-GB memory and NVIDIA GeForce GTX 1060 (CUDA 9.0 and 6.9 GB memory) graphics card (GeForce, 1060), which is used for the model training and testing.

3.1. Experiments on the public dataset

The Plantvillage database (www.plantvillage.org) is an international general database for the algorithm test of plant disease detection using machine learning (Hughes and Salathé, 2015). To assess the performance of the proposed approach, we conduct our experiments on this general database firstly, and the Maize dataset of Plantvillage is selected for the testing. There are 3852 color leaf images in this dataset, which are divided into 4 categories including 513 gray_leaf_spot images, 1192 common_rust images, 1162 healthy images, and 985 northern_leaf_blight images separately. It is noted that each image is taken in a simple background condition and the illumination intensity is relatively uniform. Some images of the same leaf are captured from different orientations. The dimension of all the images is uniform 256 × 256 pixels. Obviously, the sample distributions of this dataset are not balanced. The number of gray_leaf_spot leaf images is relatively small, and it is significantly less than that of other classes for the Maize dataset. Therefore, to ensure the balance of sample data, the data augmentation techniques are used to enrich the sample images of the gray_leaf_spot class. Through horizontal or vertical flipping, angle rotating, shearing, and size scaling, the amount of the original images in gray_leaf_spot class is enlarged using Python script. The bounded values of flipping, rotating, shearing, and scale transformation are random and evenly distributed in a specific range. For example, the rotation range is ± 40°, the scale is changed from 0.9 to 1.1, and the shear range is 0.2, etc. Thus, for the class of gray_leaf_spot, 87 additional synthetic images are selected to enrich this class, and a total of 20 original images are utilized to generate the augmented images. Fig. 6 displays the partial augmented samples.

In this way, new sample images are generated to enrich the dataset, and at least 500 images of each category are ensured for the modeling through data augmentation techniques. Then, the sample images are uniformly resized to the fixed-dimension of 224 × 224 pixels to fit the model, and the pre-processed leaf images with 500 images per class are selected to build the model. The training set and test set are divided according to the ratio of 70/30. Particularly, to know how the proposed approach will perform on new unseen data, a certain number of raw images are retained to validate the effectiveness of the model. Based on the method proposed in Section 2.4, we perform the model training and validation on the Maize dataset. Moreover, to further verify the effectiveness of the proposed approach, we considered four influential CNNs including DenseNet (Huang et al., 2017), VGGNet (Simonyan and Zisserman, 2014), Inception V3 (Szegedy et al., 2016) and ResNet (He et al., 2016) for the comparative experiments. Using the transfer learning method, the models are created and loaded with pre-trained weights from ImageNet, and the top layers are truncated by defining a new fully-connected Softmax layer with the practical number of classification. Likewise, the models of various CNNs are trained and multiple experiments are conducted on the Maize dataset. The test accuracies of different approaches are obtained in Table 2 and Figs. 7–11.

Considering the statistics of correct detections (also known as true positives), misdetections (also known as false negatives), true negatives and false positives, we can evaluate the performance of the models with the indicators including the *Accuracy*, *Sensitivity*, and *Specificity*, as expressed in Eqs. (6) and (7).

$$\text{Accuracy} = (TP + TN)/(TP + TN + FP + FN) \tag{6}$$

$$\text{Sensitivity} = TP/(TP + FN) \tag{7}$$

$$\text{Specificity} = TN/(FP + TN) \tag{8}$$

where TP (true positive) is the number of instances that actually belong to the class C and are correctly identified by the classifier, FN (false negative) is on the contrary, which is the number of instances that belong to the class C but incorrectly classified. FP (false positive) is the number of instances that do not belong to class C but mistakenly identified as this classification. TN (true negative) is the number of instances that are not in class C in reality, and they are correctly identified.

It can be seen from Table 2 that the proposed approach outperforms the other state-of-the-art methods which are experimented on the public dataset, even if the optimal classifier is adopted. The main reason is that the proposed approach transfers to the specific task using the pre-trained VGGNet on ImageNet with Inception module, and thus combines the advantages of both. The pre-trained module is employed as a basic feature extractor while the extended layers are charged with high-dimensional feature extraction as well as classification. By contrast, the other methods are individual networks. Although the models are trained with pre-trained weights rather than from scratch, the optimal results are not achieved for these models. In summary, applying the batch normalization and Swish activation function, the proposed INC-VGGN approach, which combines the advantages of the Inception module and VGGNet, achieves the top performances in the experiments. After 30 epochs of training, the validation accuracy of the proposed approach achieves 91.83% and the loss is 0.24, as displayed in Table 2. Furthermore, using the model trained by the proposed approach, the images outside modeling are selected for the class prediction of plant disease images. Fig. 12 depicted the confusion matrix of detecting results, and the corresponding evaluation indicators are calculated in Table 3.

As stated before, to validate the effectiveness of the proposed approach, the experiments were conducted on the new unseen data. From Table 3, it can be seen that the average predicting accuracy achieves no

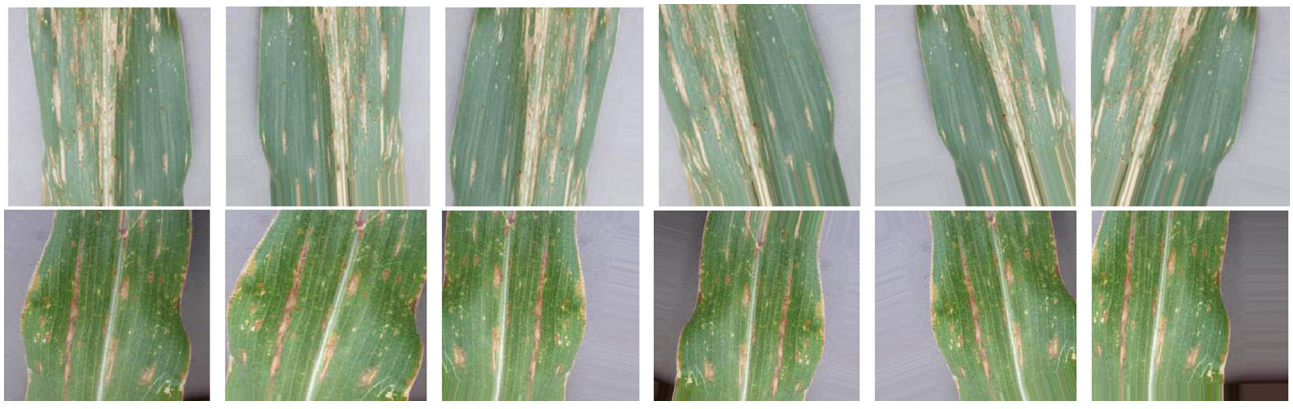


Fig. 6. Augmented images of gray leaf spot.

Table 2
Accuracy and loss of different approaches after 30 epoch training.

Pre-trained model	10 epochs			30 epochs			
	Training accuracy %	Validation accuracy %	Training loss	Training accuracy %	Validation accuracy %	Training loss	Validation loss
DenseNet-201	80.27	76.33	0.5726	84.13	79.00	0.4451	0.4987
ResNet-50	65.19	64.67	1.0028	70.41	69.67	0.8338	0.8442
Inception V3	85.56	82.33	0.4087	92.14	85.00	0.2576	0.3717
VGGNet-19	65.19	66.67	1.1640	74.20	74.83	0.9026	0.9162
Proposed method	93.78	90.17	0.2122	**97.57**	**91.83**	0.0856	**0.2409**

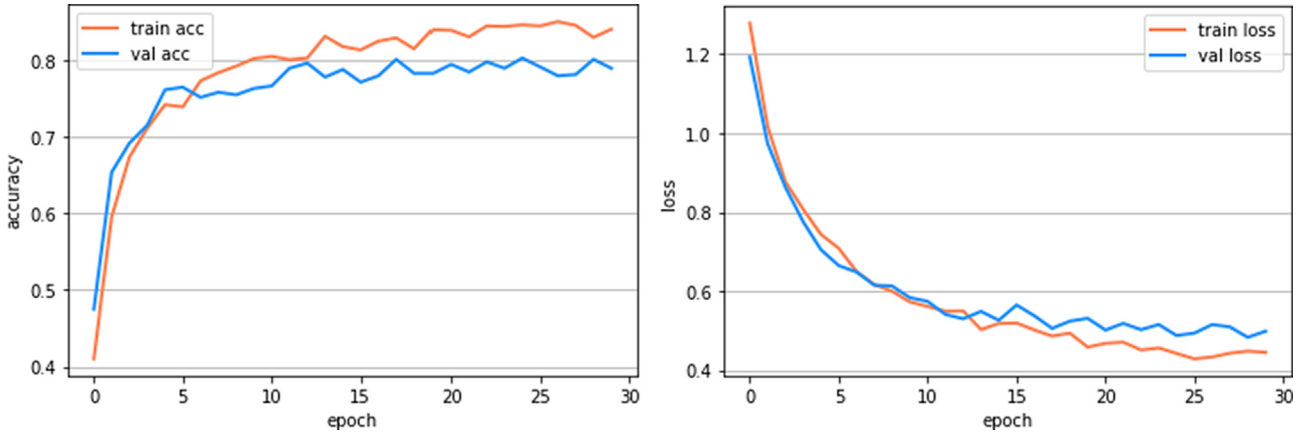


Fig. 7. DenseNet-201, left is the accuracy and right depicts the loss of the model.

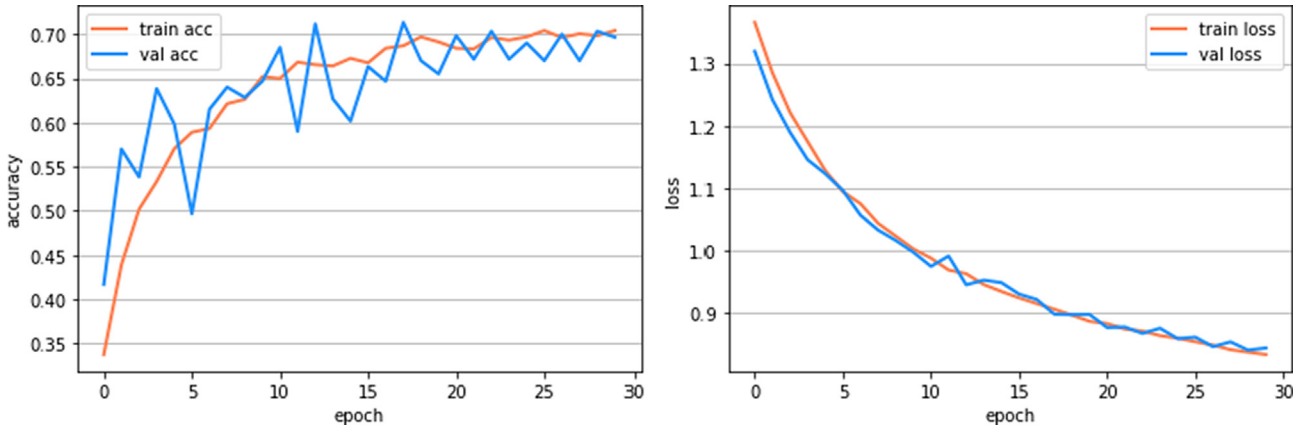


Fig. 8. ResNet-50, left is the accuracy and right depicts the loss of the model.

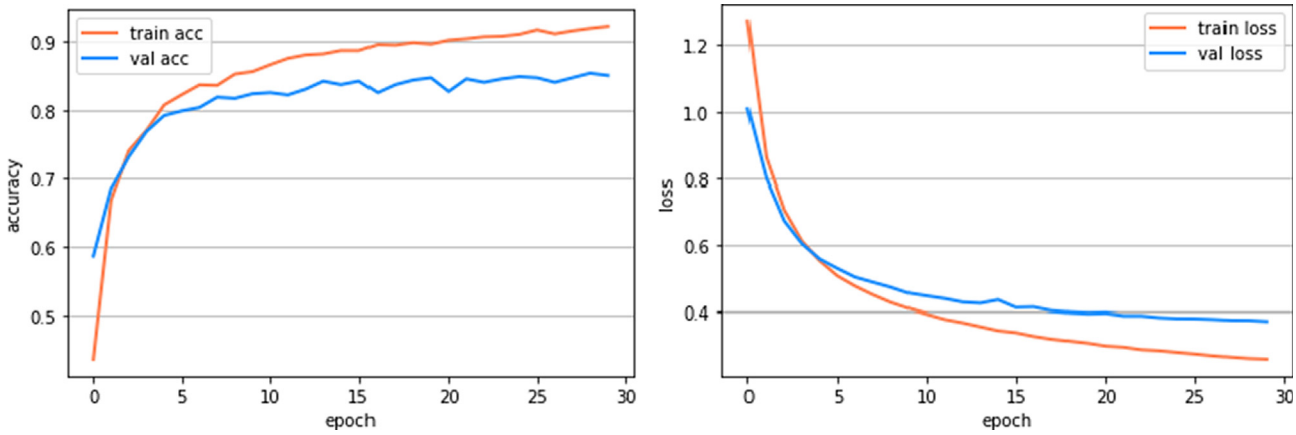


Fig. 9. Inception V3, left is the accuracy and right depicts the loss of the model.

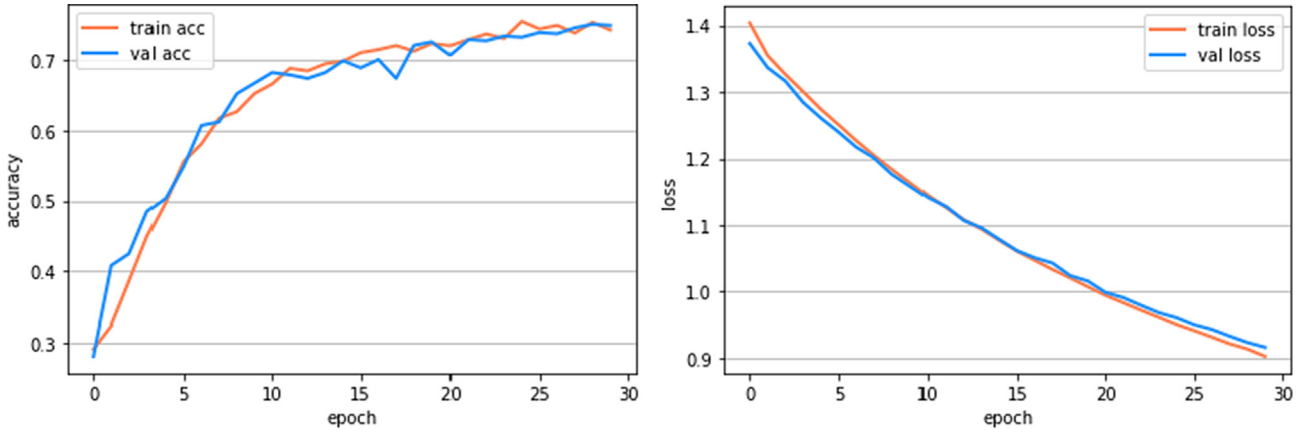


Fig. 10. VGGNet-19, left is the accuracy and right depicts the loss of the model.

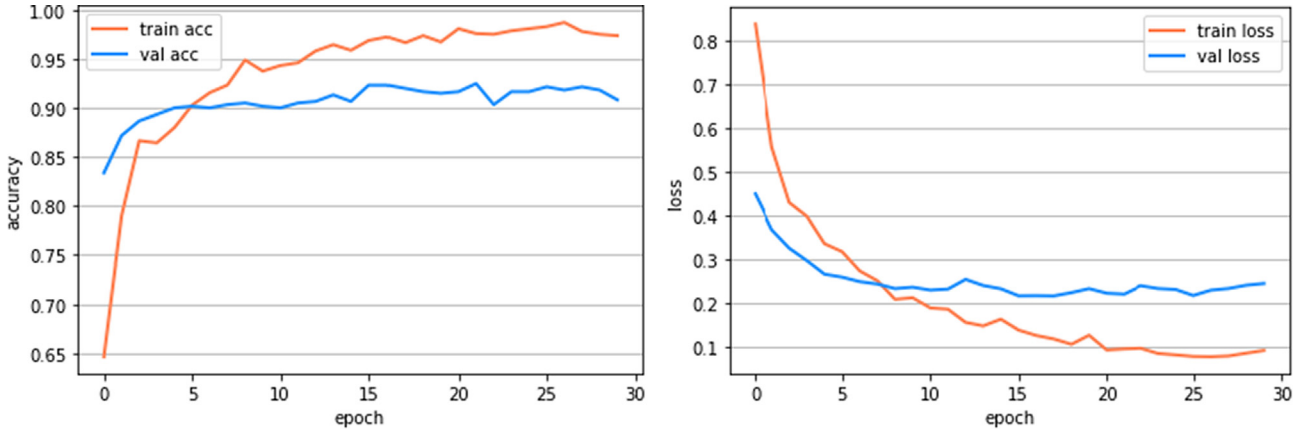


Fig. 11. The proposed approach, left is the accuracy and right depicts the loss of the model.

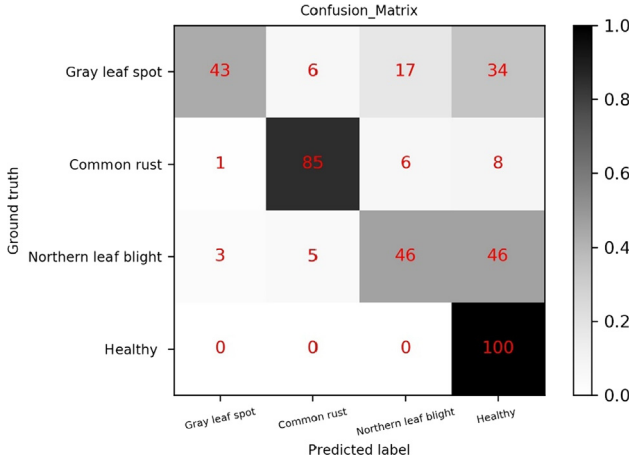


Fig. 12. Confusion matrix of maize disease detection.

Table 3
Evaluation indicators of detecting results (%).

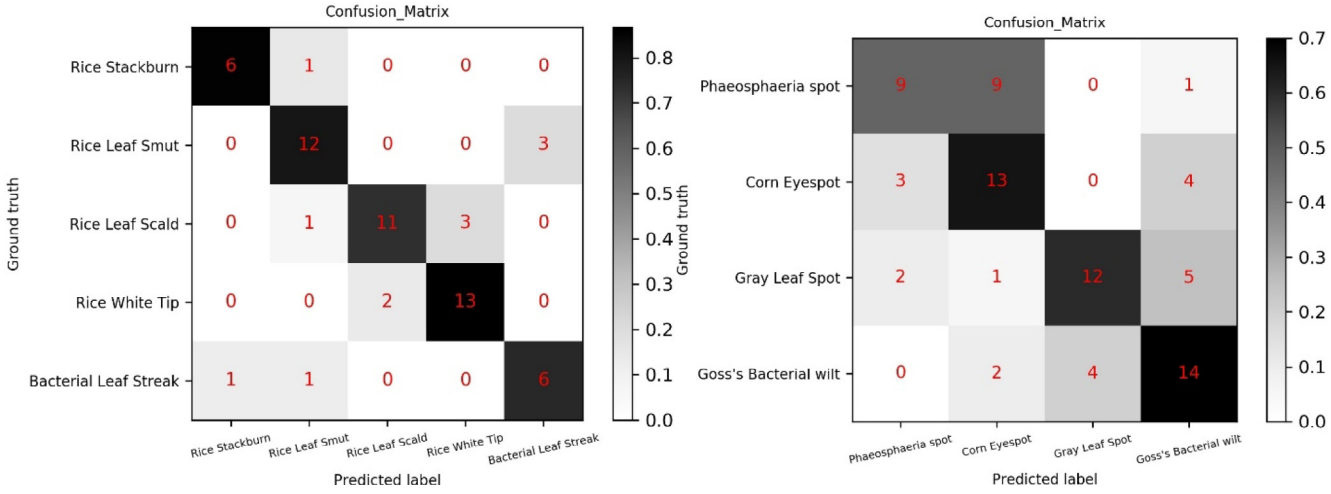
Types	Accuracy	Sensitivity	Specificity
Gray leaf spot	84.75	43.00	98.67
Common rust	93.50	85.00	96.33
Northern leaf blight	80.75	46.00	92.33
Healthy	78.00	100.00	70.67
Average	**84.25**	68.50	89.50

less than 84.25% in multiple experiments, which indicates that the proposed INC-VGGN approach has a significant capability to recognize the maize diseases. Especially, the proposed approach not only identified the healthy and diseased plants but also distinguished the specific categories of plant diseases. Thus, based on the empirical analysis, it can be concluded that the proposed method is effective for the identification of maize diseases, which can also be extended to the disease detection of other plants.

3.2. Experiments on the collected dataset

Similar to the experiments conducted in Section 3.1, the proposed approach is tested on our rice and maize image datasets, which are collected in a real-life scenario with complex background and uneven illumination intensities. The original dataset is divided into a training set and a test set with a ratio of 70/30, and a certain number of real images are retained to validate the effectiveness of the model. Particularly, to ensure the diversity of sample images and avoid the overfitting problem, the data augmentation techniques including random rotation, flipping, and scale transform are adopted to enlarge the training samples, and related pre-processing works are also performed. A brief description of the processes is presented below.

1. Change the size of the images. All the images are resized to the fixed-dimension of 224 × 224 pixels to fit the model. Using the data augmentation techniques, the disease images are augmented and at least 100 images are guaranteed for each category.
2. Image pre-processing. In order to avoid image distortion, the image pre-processing is performed to blacken the shorter sides of the



(a) Predicting results of rice disease images (b) Predicting results of maize disease images

Fig. 13. Confusion matrices of plant disease detection.

images so that they become the same proportion, thus the information of the original images is retained, and the image deformation is also prevented.

3. Dataset partition. Except for a certain number of validation images, the dataset D is divided into a training set A and testing set B with the ratio of 70/30, thus $D = A + B$.
4. Model training. Referring to the method proposed in Section 2.4, the training set A is applied to train the model. To fully verify the effectiveness, the multiple experiments are conducted with the shuffling of images.
5. Testing and validation. The testing set B is used to evaluate the model, and the new images outside modeling are applied to verify the effectiveness of the model. The output results are compared with the actual categories and the related evaluation indicators are calculated.

Thus, based on the above processes, we performed the model training on the rice disease image dataset and maize disease image dataset separately. After obtaining the trained models, the new unseen images are selected for the class prediction. Fig. 13(a,b) depicts the predicting results of rice and maize disease images, and the corresponding evaluation indicators are calculated in Tables 4,5.

For example, it can be seen from Fig. 13(a) that 13 samples are correctly predicted for the class of "Rice White Tip" except for 2 samples. Likewise, for the class of "Rice Leaf Smut", 12 samples are correctly predicted except for 3 misclassified samples. Totally, there are 48 plant leaf images that they are correctly detected by the proposed approach in 60 samples, and the average accuracy achieves 92.00% for the class prediction of all the rice disease images, as shown in Table 4. Thus, the INC-VGGN approach shows high accuracy for the identification of rice disease images, indicating the validity of the proposed approach. On the other hand, although the performance is worse than that on rice, we realize an average accuracy of 80.38% for the detection of maize disease images, which is due to the scenario that some "Phaeosphaeria Spot" and "Maize Eyespot" diseases occur in the same leaf. Additionally, the clutter field background and uneven illumination intensity can also affect the detection results. Some identified samples are displayed in Fig. 14.

Table 4
Evaluation indicators of rice disease detection (%).

Types	Accuracy	Sensitivity	Specificity
Rice Stackburn	96.67	85.71	98.11
Rice Leaf Smut	90.00	80.00	93.33
Rice Leaf Scald	90.00	73.33	95.56
Rice White Tip	91.67	86.67	93.33
Bacterial Leaf Streak	91.67	75.00	94.23
Average	**92.00**	80.00	**95.00**

As observed in Fig. 14, the top images are the original sample images, the middle images are the extracted feature maps, and the bottom images are the results of class prediction. Generally, such as Fig. 14(a-c), the predicted categories are consistent with the actual categories of these samples, and most of the rice and maize diseases are correctly detected by the proposed approach. Whereas, the serious clutter field background and uneven illumination intensity may affect the feature extraction of lesion images, and cause the individual incorrect classification, as shown in Fig. 14(d). Although the individual images are misclassified, most of the detected plant disease types are consistent with the actual categories. As seen in Tables 4 and 5, the average prediction accuracy is no less than 80.00% and the higher average specificity reaches 95.00% in multiple experiments, which indicates that the proposed INC-VGGN approach has a significant capability to recognize the plant diseases. Based on the empirical analysis, it can be concluded that the proposed approach is effective for the identification of plant disease types, which can also be extended to the application of other fields such as online fault diagnosis, target recognition, etc.

4. Conclusion

Plant diseases are the main harms to the agricultural development of the world, and they have a disastrous impact on the safety of food production. In severe cases, plant diseases may lead to no harvest completely. Therefore, the automatic identification of plant diseases is highly desired in agricultural information. Deep learning techniques, particularly CNNs, have shown promising performance in addressing most of the challenging problems associated with the classification. In this paper, the transfer learning for deep CNNs is studied with the aim of enhancing the learning ability of tiny lesion symptoms, and a novel deep learning architecture called INC-VGGN is proposed for the identification of plant disease images. The pre-trained VGGNet is modified by replacing its last layers with an additional convolutional layer, in which the batch normalization is added and the Swish activation function is used to directly replace ReLu. Then, the convolutional layer is followed by two Inception modules, and the fully connected layers are replaced by a global pooling layer to conduct the dimension reduction of feature maps. Finally, the fully-connected Softmax layer was

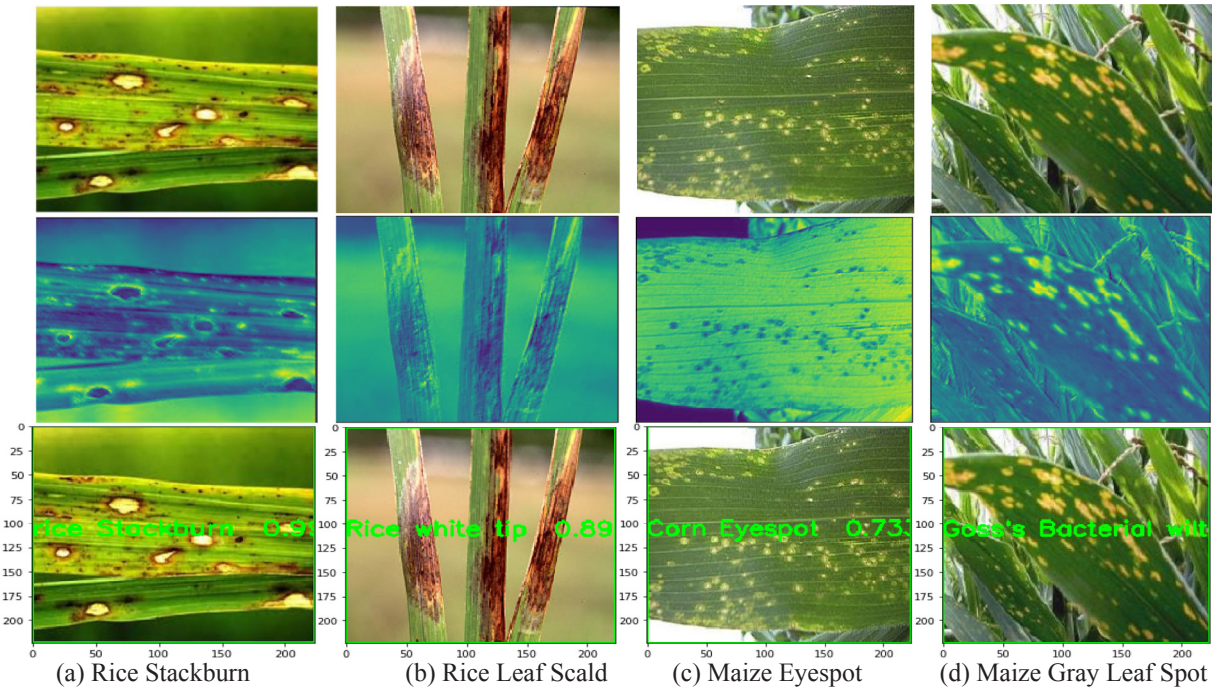


Fig. 14. The detected samples of plant disease images.

Table 5
Evaluation indicators of maize disease detection (%).

Types	Accuracy	Sensitivity	Specificity
Phaeosphaeria Spot	81.01	47.37	91.67
Maize Eyespot	75.95	65.00	79.61
Gray Leaf Spot	84.81	60.00	93.22
Goss's Bacterial wilt	79.75	70.00	83.05
Average	**80.38**	60.76	86.92

added as the top layer for the classification. Thus, the newly generated networks consist of a pre-trained module and an auxiliary structure. The former is employed as a basic feature extractor while the latter extracts the high-dimensional features and is responsible for classification. Experimental results demonstrated the model with the state-of-the-art performance on both the public dataset and our own image dataset. It achieves a validation accuracy of 91.83% on the public dataset. Even under complex background conditions, the average accuracy reaches 92.00% for the class prediction of collected rice disease images. In future development, we intend to deploy it on mobile devices to monitor and identify the broader range of plant disease information automatically. Meanwhile, we plan to apply it to more real-world applications including computer-aided diagnosis (CAD) and so on.

Declaration of Competing Interest

The authors declared that they have no conflicts of interest to this work.

We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Acknowledgment

This work is partly supported by the grants from the National Natural Science Foundation of China (Project no. 61672439) and the Fundamental Research Funds for the Central Universities (#20720181004). The authors wish to thank all the editors and anonymous reviewers for their constructive advice.

References

Faithpraise, F., Birch, P., Young, R., Obu, J., Faithpraise, B., Chatwin, C., 2013. Automatic plant pest detection and recognition using k-means clustering algorithm and correspondence filters. Int. J. Adv. Biotechnol. Res. 4 (2), 189–199.
Al-Hiary, H., Bani-Ahmad, S., Reyalat, M., Braik, M., Alrahamneh, Z., 2011. Fast and accurate detection and classification of plant diseases. Int. J. Comput. Appl. 17 (1), 31–38.
Bai, X., Cao, Z., Zhao, L., Zhang, J., Lv, C., Li, C., Xie, J., 2018. Rice heading stage automatic observation by multi-classifier cascade based rice spike detection method. Agricul. Forest Meteorol. 259, 260–270.
Al Bashish, D., Braik, M., Bani-Ahmad, S., 2011. Detection and classification of leaf diseases using K-means-based segmentation and Information Technology Journal 10 (2), 267–275.
Pooja, V., Das, R., & Kanchana, V. (2017, April). Identification of plant leaf diseases using image processing techniques. In 2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR) (pp. 130-133). IEEE.
Khirade, S. D., & Patil, A. B. (2015, February). Plant disease detection using image processing. In 2015 International conference on computing communication control and automation (pp. 768-771). IEEE.
Ebrahimi, M.A., Khoshtaghaza, M.H., Minaei, S., Jamshidi, B., 2017. Vision-based pest detection based on SVM classification method. Comput. Electron. Agricul. 137, 52–58.
García, J., Pope, C., & Altimiras, F. (2017). A distributed-means segmentation algorithm applied to lobesia botrana recognition. Complexity, 2017.
Guettari, N., Capelle-Laizé, A.S., Carré, P., 2016. Blind image steganalysis based on evidential k-nearest neighbors. IEEE, pp. 2742–2746.
Deepa, S., Umarani, R., 2017. Steganalysis on Images using SVM with Selected Hybrid Features of Gini Index Feature Selection Algorithm. Int. J. Adv. Res. Comput. Sci. 8 (5).
Ramezani, M., & Ghaemmaghami, S. (2010, January). Towards genetic feature selection in image steganalysis. In 2010 7th IEEE Consumer Communications and Networking Conference (pp. 1-4). IEEE.
Sheikhan, M., Pezhmanpour, M., Moin, M.S., 2012. Improved contourlet-based steganalysis using binary particle swarm optimization and radial basis neural networks. Neural Comput. Appl. 21 (7), 1717–1728.
Kodovsky, J., Fridrich, J., & Holub, V. (2011). Ensemble classifiers for steganalysis of digital media. IEEE Transactions on Information Forensics and Security, 7(2), 432-444.
Guo, Y., Hastie, T., Tibshirani, R., 2007. Regularized linear discriminant analysis and its application in microarrays. Biostatistics 8 (1), 86–100.
Zhang, S., Wang, Z., 2016. Cucumber disease recognition based on Global-Local Singular value decomposition. Neurocomputing 205, 341–348.
Zhang, S., Wu, X., You, Z., Zhang, L., 2017. Leaf image based cucumber disease recognition using sparse representation classification. Comput. Electron. Agricul. 134,

135–141.
Barbedo, J.G., 2018. Factors influencing the use of deep learning for plant disease recognition. Biosyst. Eng. 172, 84–91.
Kamilaris, A., Prenafeta-Boldú, F.X., 2018. Deep learning in agriculture: a survey. Comput. Electron. Agricul. 147, 70–90.
Kussul, N., Lavreniuk, M., Skakun, S., Shelestov, A., 2017. Deep learning classification of land cover and crop types using remote sensing data. IEEE Geosci. Remote Sens. Lett. 14 (5), 778–782.
Yalcin, H. (2017, August). Plant phenology recognition using deep learning: Deep-Pheno. In 2017 6th International Conference on Agro-Geoinformatics (pp. 1-5). IEEE.
Mohanty, S.P., Hughes, D.P., Salathé, M., 2016. Using deep learning for image-based plant disease detection. Front. Plant Sci. 7, 1419.
Ma, J., Du, K., Zheng, F., Zhang, L., Gong, Z., Sun, Z., 2018. A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. Comput. Electron. Agricul. 154, 18–24.
Kawasaki, Y., Uga, H., Kagiwada, S., Iyatomi, H., 2015. Basic study of automated diagnosis of viral plant diseases using convolutional neural networks. Springer, Cham, pp. 638–645.
Kessentini, Y., Besbes, M.D., Ammar, S., Chabbouh, A., 2019. A two-stage deep neural network for multi-norm license plate detection and recognition. Expert Syst. Appl. 136, 159–170.
Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognitionpp. 4700–4708.
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognitionpp. 1–9.
Cetinic, E., Lipic, T., Grgic, S., 2018. Fine-tuning convolutional neural networks for fine art classification. Expert Syst. Appl. 114, 107–118.
Dyrmann, M., Karstoft, H., Midtiby, H.S., 2016. Plant species classification using deep convolutional neural network. Biosyst. Eng. 151, 72–80.
Lu, Y., Yi, S., Zeng, N., Liu, Y., Zhang, Y., 2017. Identification of rice diseases using deep convolutional neural networks. Neurocomputing 267, 378–384.
Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: Int. Conf. Learn. Represent. pp. 1–14.
Khan, S., Islam, N., Jan, Z., Din, I.U., Rodrigues, J.J.C., 2019. A novel deep learning based framework for the detection and classification of breast cancer using transfer learning. Pattern Recog. Lett. 125, 1–6.
Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Berg, A.C., 2015. Imagenet large scale visual recognition challenge. Int. J. Comput. Vision 115 (3), 211–252.
Lumini, A., Nanni, L., 2019. Deep learning and transfer learning features for plankton classification. Ecol. Inform. 51, 33–43.
Ghazi, M.M., Yanikoglu, B., Aptoula, E., 2017. Plant identification using deep neural networks via optimization of transfer learning parametersNeurocomputing 235, 228–235.
Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognitionpp. 1251–1258.
Ioffe, S., Szegedy, C., 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on International Conference on Machine Learning, pp. 448–456. JMLR.org.
Ramachandran, P., Zoph, B., Le, Q.V., 2017. Searching for activation functions. CoRR abs/1710.05941. http://arxiv.org/abs/1710.05941.
Redmon, J., Farhadi, A., 2017. YOLO9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognitionpp. 7263–7271.
Hung, J.C., Lin, K.C., Lai, N.X., 2019. Recognizing learning emotion based on convolutional neural networks and transfer learning. Appl. Soft Comput. 84.
Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks. Springer, Cham, pp. 818–833.
Zhang, S., Zhang, S., Zhang, C., Wang, X., Shi, Y., 2019. Cucumber leaf disease identification with global pooling dilated convolutional neural network. Comput. Electron. Agricul. 162, 422–430.
Keras-GPU. Available online: https://anaconda.org/anaconda/keras-gpu (accessed on 17 Jun, 2019).
GeForce GTX 1060. Available online: https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx -1060/specifications (accessed on 17 Jun, 2019).
Hughes, D., Salathé, M., 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics. CoRR abs/1511.08060. https://arxiv.org/abs/1511.08060.
Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognitionpp. 2818–2826.
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognitionpp. 770–778.