



Crop Disease Image Classification Based on Transfer Learning with DCNNs

Yuan Yuan¹, Sisi Fang^{1,2(✉)}, and Lei Chen^{1(✉)}

¹ Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei, China
fss135721@163.com, chenlei@iim.ac.cn

² University of Science and Technology of China, Hefei, China

Abstract. Machine learning has been widely used in the crop disease image classification. Traditional methods relying on the extraction of hand-crafted low-level image features are difficult to get satisfactory results. Deep convolutional neural network can deal with this problem because of automatically learning the feature representations from raw image data, but require enough labeled data to obtain a good generalization performance. However, in the field of agriculture, the available labeled data in target task is limited. In order to solve this problem, this paper proposes a method which combines transfer learning with two popular deep learning architectures (i.e., AlexNet and VGGNet) to classify eight kinds of crop diseases images. First, during the training procedure, the batch normalization and DisturbLabel techniques are introduced into these two networks to reduce the number of training iterations and overfitting. Then, after training the pre-trained model by using the open source dataset PlantVillage. Finally, we fine-tune this model with our relatively small dataset preprocessed by a proposed strategy. The experimental results reveal that our approach can achieve an average accuracy of 95.93% compared to state-of-the-art method for our relatively small dataset, demonstrating the feasibility and robustness of this approach.

Keywords: Transfer learning · Deep learning · Image classification
DCNN · Crop diseases

1 Introduction

Crop disease is one of the important factors that affect food security [9, 15]. It is reported that 50% of the yield losses are caused by crop diseases and pests [3]. Due to the wide variety of diseases, it is easy to misdiagnose only by artificial observation and experience judgment.

In the past few years, there has been a great progress in the area of crop disease image recognition since computer vision and machine learning was used. The most commonly used classification methods include support vector machine (SVM) [10, 16], K-nearest neighbors (KNN) [8, 19] and discriminant analysis [17]. For example, Tian, et al. [16] extracted the color and texture features of the

lesion leaves and then used SVM with different kernel functions to identify 60 images including cucumber downy mildew and powdery mildew. Zhang, et al. [19] classified 100 images of five different corn diseases with KNN after the lesion area segmentation and feature extraction. Wang, et al. [17] used discriminant analysis to identify three different cucumber diseases of 240 images by combining color, shape and texture feature of leaf spots with environmental information. These previous studies have two principal problems. First, the number of samples in datasets is small (between 60 and 240 images). Second, it is necessary to segment the lesion area firstly and extract some specific features, which are always not easy for some kinds of crop diseases, such as cucumber powdery mildew, rice flax spot, etc. Meanwhile, the information of crop diseases cannot be represented entirely with the specific features.

Fortunately, deep convolutional neural network (DCNN), which can extract the deep feature of images by multiple convolution layer and pool layer, was adopted to deal with the above problems in recent years. In 2012, a large DCNN achieved a top-5 error of 16.4% for the classification of images into 1000 possible categories [6]. In the next few years, some DCNN architectures such as AlexNet, GoogLeNet [14] and VGGNet [11] were widely applied in the task of plant disease image recognition. Mohanty, et al. [7] trained a CNN to identify 14 crop species and 26 diseases of PlantVillage dataset, which demonstrated that the feasibility of the approach for disease classification based on the pre-trained model. Srdjan, et al. [12] and Brahimi, et al. [1] classified plant leaf disease images by fine-tuning CaffeNet and AlexNet, which obtained good results. We can see that the above methods are mainly based on the PlantVillage dataset with a large number of images and simple background.

Different from the above works, our crop disease dataset, including five kinds of rice diseases and three kinds of cucumber diseases, has two key issues, a relatively small number of images (less than 10,000 images) and complex background. Therefore, this paper proposes a novel method that employs the PlantVillage dataset to assist our crop disease dataset for classification based on one pre-processing strategy for our dataset and two networks, which are optimized by using the batch normalization and DisturbLabel technique during training.

2 Materials and Methods

2.1 Image Preprocessing

Two datasets are used in this paper. First, in order to obtain the pre-trained model, we use an auxiliary dataset that is collected from the open dataset PlantVillage [4], which contains 54306 images with simple background in 38 classes. Another one is the target dataset with complex background that is collected on sunny days, using the digital single lens reflex camera Canon EOS 6D. The original target dataset, which consists of 2 crop species with 8 different kinds of diseases, contains 2430 images with the inconsistent size. Figure 1 shows some examples from the original target dataset.

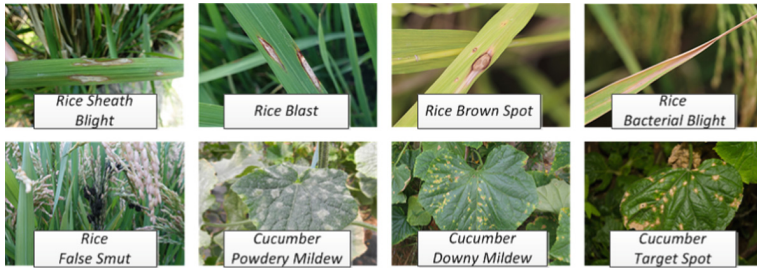


Fig. 1. Example of leaf images from original target dataset

Two pre-processing strategies of the target dataset called center crop and corner crop are used in this work. In center crop, we crop a 300×300 square region from the center of each image. Thus, most complex background can be removed and the image quantity is unchanged. In corner crop, we firstly crop center area to 512×512 resolution which keeps most complex background. And then we divide the image into four pieces with 256×256 resolution. Finally, we resize these images into two different sizes (227×227 pixels for AlexNet and 224×224 pixels for VGGNet) using bi-linear interpolation respectively. The pre-processing procedures are shown in Fig. 2. After conducting the above operations on each image and filtering the images with no lesion area, the original target dataset is eventually augmented to 9592 images.

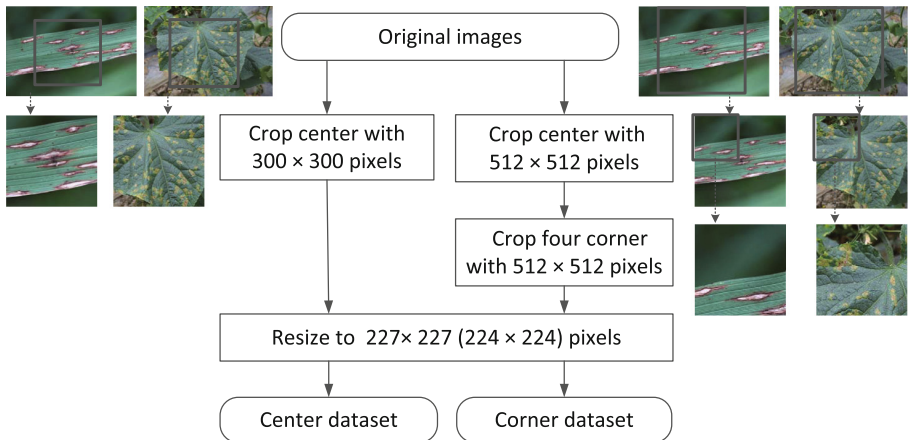


Fig. 2. Two strategies for image pre-processing

2.2 Batch Normalization

Batch normalization ensures that the inputs of layers always fall in the same range even though the earlier layers are updated and always leads to an obvious reduction in the number of training iterations and regularizes the model [5]. We calculate the mean and variance of $x_1 \sim x_n$ for each batch of n samples according to formulas (1) and (2):

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad (1)$$

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

where μ and σ are the mean and variance of the data of current batch respectively. After normalized according to formulas (3), the parameter \hat{x}_i is obtained whose mean is 0 and variance is 1:

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

where ε is a small constant that is added to the variance to avoid zero-division. To avoid the change of feature distribution by data normalization, the reconstitution is needed to restore the original feature distribution.

$$y_i = \gamma_i \hat{x}_i + \beta_i \quad (4)$$

$$\gamma_i = \sqrt{\text{Var}[x_i]} \quad (5)$$

$$\beta_i = E[x_i] \quad (6)$$

where γ_i and β_i are trainable parameters, Var the variance function and E is the mean function. It can be found that the original data can be restored when γ_i and β_i are set in accordance with formulas (5) and (6).

In fact, the above parameters are vectors whose dimensions are the same as the size of the input image.

2.3 Transfer Learning with DCNNs

In this paper, we compare performances for crop disease classification between two network architectures. In order to optimize the result, batch normalization and DisturbLabel algorithm are introduced into different layers of the network.

DisturbLabel can be interpreted as a regularization method on the loss layer, which works by randomly choosing a small subset of training data, and intentionally setting their ground-truth labels to be incorrect [18]. So it can improve the network training process by preventing it from over-fitting. We assume that there are N samples in C classes in each batch given as $(x_n, y_n)_{n=1}^N$, where y_n is a corresponding label for a sample. When a sample x_n is determined to be disturbed with a certain probability γ which is a noise rate, its label y_n will be

set to a new label \tilde{y}_n that is randomly chosen from $\{1, 2, \dots, C\}$ according to formulas (7) and (8),

$$p_t = 1 - \gamma \cdot \frac{C - 1}{C} \quad (7)$$

$$p_i = \gamma \cdot \frac{1}{C} \quad (8)$$

where t is the ground-truth label, $i \neq t$ and the range of γ which can be set according to different datasets and different networks is 0 to 1.

The first network we use is AlexNet, which is a DCNN successfully trained on roughly 1.2 million labeled images of 1,000 different categories from the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset. It consists of five convolution layers, followed by three fully connected layers (fc6 to fc8) and a softmax classifier. The first two convolution layers are each followed by a Local Response Normalization (LRN) and a max-pooling layer, and the last convolution layer is followed by a single max-pooling layer. Moreover, it uses the dropout regularization method [13] to reduce over-fitting in the fully connected layers and applies Rectified Linear Units (ReLUs) [2] for the activation of those and the convolutional layers. The second network we use is a modified version of the 16-layer model from the VGG team in the ILSVRC 2014 trained on the ImageNet dataset. In our paper, we denote it as VGGNet. The network consists of thirteen convolution layers followed by three fully connected layers (fc6 to fc8) and a softmax classifier. There is an obvious improvement on VGGNet with depth increase and very small convolution filters (3×3). The width of convolution layers is rather small, starting from 64 in the first layer and increasing by a factor of 2, until it reaches 512.

It will take a long time to converge when training a model for disease image classification by analyzing the structures of two networks. Thus, we consider adding batch normalization in final fully connected layers to reduce the number of iterations. To train a transfer learning model where the final fully connected layers (fc8) of two networks are replaced with a layer with 38 outputs corresponding to the 38 image categories of the PlantVillage dataset. The training is carried out by using mini-batch gradient descent where the batch size is set to 64. And the dropout ratio for the first two fully-connected layers is set to 0.5. The learning rate is initially set to 10^{-2} , and then decreased by a factor of 0.98. For random initialization, the weights are initialized from a normal distribution with the zero mean and 0.01 variance and the biases are initialized with zero. However, during the procedure of transfer learning, we take the following three measures.

1. The output of the final fully connected layer (fc8) is set to 8 to satisfy target dataset.
2. DisturbLabel algorithm are employed in the loss layer to improve the network training process by preventing it from over-fitting. Here, the batch size is set to 128.
3. And the weights of the final fully connected layer (fc8) for two networks is re-initialized.

The improved network architecture based on fine-tuning the pre-trained model with the PlantVillage dataset is shown in Fig. 3.

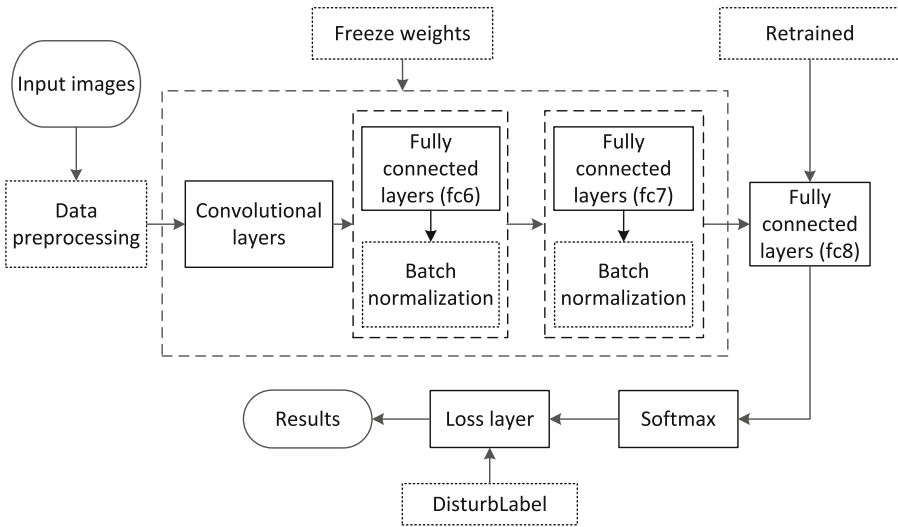


Fig. 3. The improved network architecture based on fine-tuning the pre-trained model

3 Experimental Results and Discussion

3.1 Experimental Setup

All the experiments are conducted on TensorFlow framework, which is a fast open source framework for deep learning. On a system equipped with three NVIDIA 1080Ti GPUs and a 64 G of memory, training a model on the PlantVillage dataset takes approximately fifteen hours depending on the architecture. For our approach, we make use of the PlantVillage dataset as an auxiliary dataset to train the pre-trained model for our target dataset containing eight kinds of crop diseases of 2430 original images.

We use the average accuracy as the evaluation index of the experiment result and calculate it according to formula (9):

$$Accuracy = \frac{1}{n_c} \sum_{i=1}^{n_c} \frac{n_{ai}}{n_i} \times 100\% \tag{9}$$

where n_c is the training number of each epoch, n_{ai} is the number of the sample predicted accuracy of each training and n_i is the number of the sample of each training.

3.2 The Pre-trained Model

During training the pre-trained models, for comparison, we train the models by using the PlantVillage dataset on two different network architectures. The dataset is split into two sets, namely training set (80% of the dataset) and validation set (20% of the dataset). Since the learning always converges well within 100 epochs based on the empirical observation, each of these experiments runs for 100 epochs, where one epoch is defined as the number of training iterations in which the neural network has completed a full pass of the whole training set. As Fig. 4(a) shows, between the AlexNet and VGGNet architectures, we can see that the classification results on the PlantVillage dataset of AlexNet is better than VGGNet. Meanwhile, Fig. 4(b) shows that there is no divergence between the validation loss and the training loss of these two network architectures, confirming that the over-fitting problem is not a contributor to the obtained results.

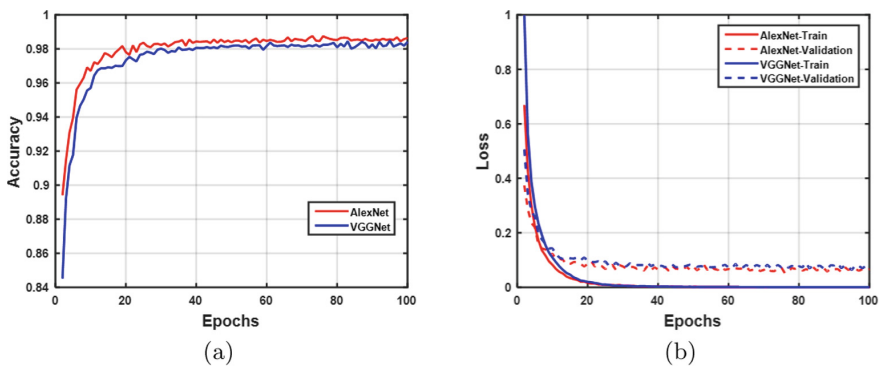


Fig. 4. (a) Comparison of validation accuracy by training on AlexNet and VGGNet with the PlantVillage dataset; (b) Comparison of train-loss and validation-loss by training on AlexNet and VGGNet with the PlantVillage dataset.

3.3 Transfer Learning

After obtaining the pre-trained model with the PlantVillage dataset, we carry out transfer learning based on this model. During fine-tuning, each target dataset (shortly written as Corner dataset and Center dataset) is also split into two sets, training set (80% of the dataset) and validation set (20% of the dataset). Based on the empirical observation, the number of iterations is set to 300 epochs. Then we compare the results of two networks by training models on the Center dataset and Corner dataset.

The Effect of γ on Accuracy. Because each target dataset has only a few thousands of images, we use DisturbLabel algorithm on the loss layer to reduce the over-fitting problem. Dropout rate is fixed to 0.5, since it has been proved that DisturbLabel cooperates well with dropout when dropout rate takes this

value. On the one hand, we carry out the experiments on two datasets when noise rate γ is set to different values from 0.08 to 0.2 according to previous works. The results are shown in Table 1. For Center dataset, when γ takes 0.15, the validation accuracy of two networks can reach 94.97% and 95.14%, respectively. For Corner dataset, when γ takes 0.08, two networks achieve the highest accuracies of 95.93% and 95.42% respectively. Besides, we can see that overall experimental results of Corner dataset are better than Center dataset. On the other hand, we compare the results on two networks when γ is set to different values, showing that AlexNet performs better than VGGNet on the Corner dataset.

Table 1. Validation accuracies of different γ on different networks and datasets

Model	Dataset	$\gamma = 0.08$	$\gamma = 0.1$	$\gamma = 0.15$	$\gamma = 0.2$
AlexNet	Center dataset	0.9414	0.9181	0.9497	0.9239
AlexNet	Corner dataset	0.9593	0.9566	0.9542	0.9484
VGGNet	Center dataset	0.9179	0.9505	0.9514	0.9215
VGGNet	Corner dataset	0.9542	0.9497	0.9489	0.9457

Fine-Tuning vs Training from Scratch. These two original networks mainly use dropout layer, data augmentation and L2 regularization to optimize models. In addition, we propose two kinds of strategies to process the target dataset and combine DisturbLabel algorithm with batch normalization to improve the final results. As shown in Table 2, comparing our method with two original networks for fine-tuning the pre-trained model, our method can achieve better results than the two original networks on both Corner dataset and Center dataset.

Table 2. Validation accuracies of different methods on different networks and datasets

Model	Corner dataset	Center dataset
AlexNet (Our method) fine-tuning	0.9593 ($\gamma = 0.08$)	0.9497 ($\gamma = 0.15$)
AlexNet fine-tuning	0.9492	0.9100
AlexNet from scratch	0.9473	0.8924
VGGNet (Our method) fine-tuning	0.9542 ($\gamma = 0.08$)	0.9514 ($\gamma = 0.15$)
VGGNet fine-tuning	0.8613	0.8260
VGGNet from scratch	0.8520	0.8448

Furthermore, in order to ensure the availability of our method, transfer learning and training from scratch are compared, showing that transfer learning always yields better results. From Fig. 5(a) and (b), we can see that our method

converge well within 300 epochs for two networks. And the performance of the training on the Corner dataset is more stable than on the other one. We think the reason is that the images in Corner dataset after data preprocessing is more than Center dataset.

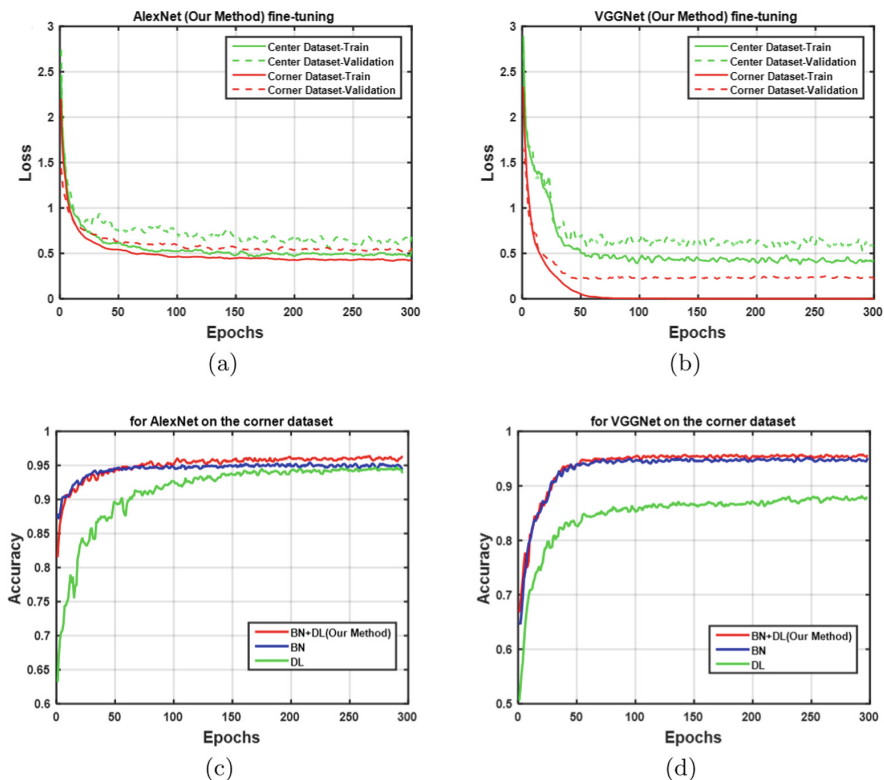


Fig. 5. (a) Comparison of loss on the two dataset for AlexNet; (b) Comparison of loss on the two dataset for VGGNet; (c) Comparison of validation accuracy for AlexNet with our method, BN and DL; (d) Comparison of validation accuracy for VGGNet with our method, BN and DL. (BN: batch normalization; DL: DisturbLabel)

The Effects of Batch Normalization and DisturbLabel. In order to know the effect of batch normalization or DisturbLabel on results, we show the performance of our method in Fig. 5(c) and (d), including the method only with the batch normalization and the method only with DisturbLabel algorithm for two networks on the Corner dataset. As we expect, there is a faster convergence by adding the batch normalization to the fully connected layers than only with DisturbLabel algorithm. Because batch normalization always results in a significant reduction in the required number of training iterations. Although our method lead to a slightly decrease of accuracy than the method only with the

batch normalization at first, there is still an advantage on our method than two other methods after 80 epochs. Meanwhile, the method only with DisturbLabel algorithm which causes a decrease of accuracy and a slower convergence reveals the worst results.

The Proposed Method vs Traditional Method. To show the effectiveness of our approach, we compare the result of our approach with traditional method. In the segmentation stage, the background is removed and replaced with a black color. During features extraction, color (color moment), texture (GLCM) and shape features such as discrete index and circularity are extracted. Then a classifier SVM whose overall performance is good is employed. We notice that the best accuracy of our method is 95.93% against 93.15% in traditional method as shown in Fig. 6. The result reveals the power of DCNN in learning features without human intervention.

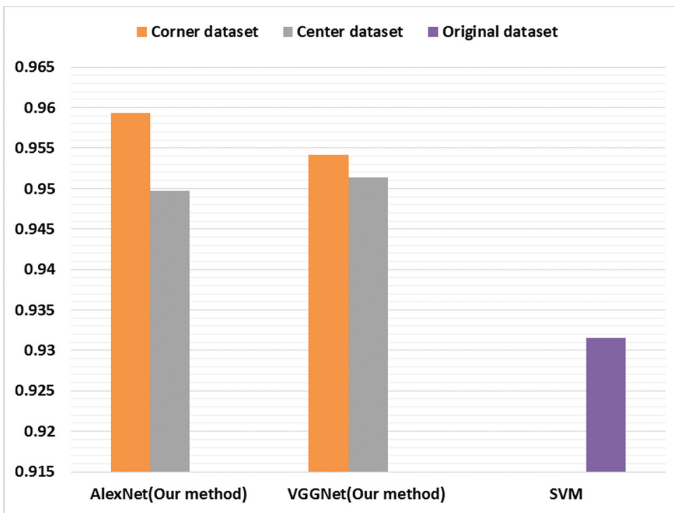


Fig. 6. Comparison of accuracy between our method and traditional method

4 Conclusion

The paper proposes a method which uses the open source dataset PlantVillage to combine transfer learning with two popular deep learning architectures AlexNet and VGGNet to classify eight kinds of crop diseases images, including five kinds of rice diseases and three kinds of cucumber diseases. First, the strategy of target dataset preprocessing can obviously augment the original target dataset. Second, the method combining batch normalization with DisturbLabel algorithm can better optimize these two networks. Comparing with original networks, the proposed method is able to achieve an average accuracy of 95.93%.

The experiment results reveal that using PlantVillage dataset to assist our target dataset for classification is feasible and AlexNet always performs better than VGGNet for target dataset with our method. Meanwhile, the proposed method provides one possibility for classification of relatively small disease dataset based on DCNNs and avoid the problem of spot segmentation. This work can provide the theoretical basis for the development of automatic identification system for crop diseases.

The work in this paper is still preliminary. In next work, how to select the more suitable auxiliary training dataset and obtain the more appropriate features will be studied. And comparisons with more deep learning architectures will also be considered.

Acknowledgments. The authors would like to thank the anonymous reviewers for their helpful reviews. The work is supported by National Natural Science Foundation of China (Grant No. 31871521), the Open Project of Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture, China (2017AIOT-01) and the 13th Five-year Informatization Plan of Chinese Academy of Sciences (Grant No. XXH13505-03-104).

References

1. Brahim, M., Boukhalfa, K., Moussaoui, A.: Deep learning for tomato diseases: classification and symptoms visualization. *Appl. Artif. Intell.* **314**, 1–17 (2017)
2. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: *International Conference on Artificial Intelligence and Statistics*, pp. 315–323 (2011)
3. Harvey, C.A., et al.: Extreme vulnerability of smallholder farmers to agricultural risks and climate change in Madagascar. *Philos. Trans. R. Soc. Lond.* **369**(1639), 20130089 (2014)
4. Hughes, D.P., Salathe, M.: An open access repository of images on plant health to enable the development of mobile disease diagnostics. *Comput. Sci.* (2015)
5. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift, pp. 448–456 (2015)
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: *International Conference on Neural Information Processing Systems*, pp. 1097–1105 (2012)
7. Mohanty, S.P., Hughes, D.P., Salath, M.: Using deep learning for image-based plant disease detection. *Front. Plant Sci.* **7**, 1419 (2016)
8. Prasad, S., Peddoju, S.K., Ghosh, D.: Multi-resolution mobile vision system for plant leaf disease diagnosis. *Sig. Image Video Process.* **10**(2), 379–388 (2016)
9. Sanchez, P.A., Swaminathan, M.S.: Cutting world hunger in half. *Science* **307**(5708), 357–359 (2005)
10. Sema, N.A., Tharwat, A., Elhariri, E., Hassanien, A.E.: Fruit-based tomato grading system using features fusion and support vector machine. In: Filev, D., et al. (eds.) *Intelligent Systems 2014. AISC*, vol. 323, pp. 401–410. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-11310-4_35
11. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *Computer Science* (2014)

12. Srdjan, S., Marko, A., Andras, A., Dubravko, C., Darko, S.: Deep neural networks based recognition of plant diseases by leaf image classification. *Comput. Intell. Neurosci.* **2016**(6), 1–11 (2016)
13. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
14. Szegedy, C., et al.: Going deeper with convolutions, pp. 1–9 (2014)
15. Tai, A.P.K., Martin, M.V., Heald, C.L.: Threat to future global food security from climate change and ozone air pollution. *Nat. Clim. Change* **4**(9), 817–821 (2014)
16. Tian, Y.W., Li, T.L., Zhang, L., Wang, X.J.: Diagnosis method of cucumber disease with hyperspectral imaging in greenhouse. *Trans. Chin. Soc. Agric. Eng.* **26**(5), 202–206 (2010)
17. Wang, X., Zhang, S., Wang, Z., Zhang, Q.: Recognition of cucumber diseases based on leaf image and environmental information. *Trans. Chin. Soc. Agric. Eng.* **30**(14), 148–153 (2014)
18. Xie, L., Wang, J., Wei, Z., Wang, M., Tian, Q.: Disturblabel: regularizing CNN on the loss layer. In: *Computer Vision and Pattern Recognition*, pp. 4753–4762 (2016)
19. Zhang, S.W., Shang, Y.J., Wang, L.: Plant disease recognition based on plant leaf image. *J. Anim. Plant Sci.* **25**(3), 42–45 (2015)