Pepper Disease Detection Model Based on Convolutional Neural Network and Transfer Learning

To cite this article: Yuting Zeng *et al* 2021 *IOP Conf. Ser.: Earth Environ. Sci.* **792** 012001

Pepper Disease Detection Model Based on Convolutional Neural Network and Transfer Learning

Yuting Zeng[1], Yanbin Zhao[1], YingHao Yu[1], Yuan Tang[1] and Youwan Tang[2]*

[1]College of Computer Science and Cyber Security, Chengdu University of Technology, Chengdu 610059, China.

[2]Chengdu Academy of Agricultural and Forestry Sciences, Chengdu, Sichuan, 611130, China.

*Corresponding author's e-mail: tangyouwan@cdnky.com

Abstract. We used a deep learning approach based on convolutional neural networks to perform plant disease detection and diagnosis using leaves images of healthy and diseased plants. The model was trained using images from a data set of 2,478 chilies, consisting of 1,478 healthy leaves (19% from the field environment) and 1,000 infected leaves (10% from the field environment). The detection model is trained based on transfer learning, and the best performance reaches 99.55% accuracy when identifying diseased or healthy plants. The model can be applied to the early warning of pepper diseases, and the method can be further extended to support the identification of crop diseases under actual cultivation conditions.

1. Introduction

This paper studies the identification method of plant diseases and insect pests in the growth process of pepper. Intelligent recognition of pepper diseases based on images has always been a challenging research topic in precision agriculture. Lee et al. [1] proposed a CNNS system based on leaves images for the automated plants recognition. Grinblat et al. [2] developed a relatively simple, though powerful neural network for the successful identification of three different legume species based on the morphological patterns of leaves' veins. Mohanty et al. [3] compared two well-known and established architectures of CNNs in the identification of 26 plant diseases, using an open database of leaves images of 14 different plants. Their results were identified with 99.35 percent accuracy. However, a main drawback was that the entire photographic material only included solely images in laboratory setups, not in real conditions in the cultivation field. Sadojevic et al. [4] developed a similar methodology for detecting plant diseases through leaves images, which included a smaller number of diseases (13) and different plants (5), using a similar amount of data obtained on the Internet. Their model has a success rate of between 91% and 98%, according to the testing data. Fuentes et al. [5] studied CNN models for the detection of 9 different tomato diseases and pests, with satisfactory performance. However, in the natural environment, it is often unrealistic to expect the classical algorithm designed to completely eliminate the influence of scene changes on the recognition results. In the real and complex natural environment, the detection of plant diseases and insect pests is facing many challenges, such as small difference between the lesion area and the background, low contrast, large scale change of the lesion area and various types, and large noise of the lesion image.

In this paper, the transfer learning method and specific CNN architecture (MobileNet) [6] are used to study the disease detection and diagnosis method of pepper plant leaves images. The available data sets consist of images captured in laboratory scenes and field cultivation conditions. We tweaked the MobileNet architecture, retrained the top full connection layer, and carried out migration learning. The next section describes the basic principles of the model, and the third section describes the data sets used for training and testing, as well as the training process. Section 4 introduces the application results of pepper disease detection and diagnosis model. Finally, the thesis puts forward some conclusions and future research directions.

2. Method

2.1 MobileNet model

MobileNet is a mobile architecture that has very small computational complexity and model memory compared to other neural networks. MobileNet V2 is based on a streamlined architecture. Its basic unit is deep separable convolution, which is used to build a lightweight deep neural network. Its basic building block is the bottleneck deep separable convolution block based on the inversion residuals structure, and its network structure is shown in Figure 1.

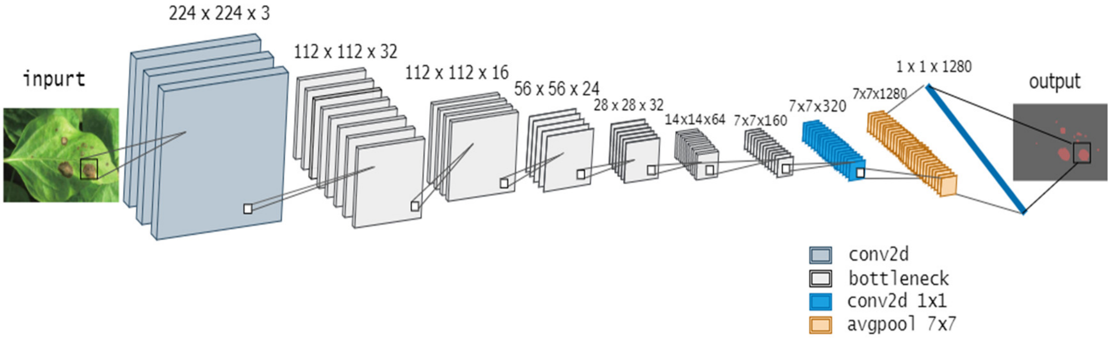


Figure 1. MobileNet V2 network structure

The core of the MobileNet V2 model is Depthwise separated Convolution (DSC), as shown in Figure 2. The difference between depth separable convolution and standard convolution is obvious.

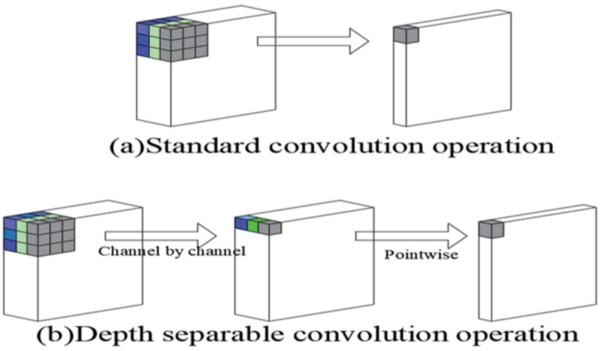

(a)Standard convolution operation
Channel by channel
Pointwise
(b)Depth separable convolution operation

Figure 2. comparison between deep separable convolution and standard convolution

It is assumed that the input feature graph size is $D_F \times D_F \times M$, and the output feature graph size is $D_F \times D_F \times N$, where $D_F$is the width and height of the feature graph. This is assuming that they are the same, and the sum refers to the number of channels or depth. It is also assumed that the width and height of the input and output feature graphs are the same. Although the size of the convolution kernel is a special case, it does not affect the generality of the following analysis. Depthwise separable convolution and standard convolution can be compared as follows:

$$\frac{D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_K \times D_K \times M \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_K^2} \qquad (1)$$

In general, when N is large, the number of parameters will decrease a lot.

The basic building block of MobileNetV2 is a bottleneck which can separate convolution and residuals. The architecture of MobileNetV2 consists of an initial full convolution layer with 32 filters, followed by the remaining 17 bottleneck layers.

2.2 Transfer Learning

Transfer learning solves the problem of how to transfer the learned knowledge from one scene to another. It is necessary to combine deep learning with transfer learning because the combination of the two can save resources while improving the accuracy of the model. When we lack enough data to complete the training, we can realize the generalization ability of the model itself through transfer learning.

Transfer learning first keeps the structure of the model convolution layer unchanged, and then loads the trained weights and parameters into the convolution layer. Then, a new full-connection layer suitable for the new task is designed, the original full-connection layer is replaced with the newly designed full-connection layer, and a new convolutional network model is formed with the previous convolution layer. Finally, the new model is trained with the new data set images.

2.3 Model Construction

We use TF-hub to load the MobileNet V2 model, use Keras Sequential Model to build the neural network, and integrate the linear classifiers into feature_ extractor_ layer and Hub models. In order to prevent over fitting and improve the generalization ability of the model, two groups of dropout layers are added to the full connection layer, with dropout rates of 0.4 and 0.2 respectively. At the same time, the activation function LeakyReLU is added as a layer after each dropout. LeakyReLU is an activation function superior to ReLU. In addition to the advantages of reducing the amount of computation, solving the problem of gradient vanishing and alleviating over fitting, it gives a non-zero slope to all negative input values, which solves the problem that the neurons do not learn when the ReLU function enters a negative interval. The parameter alpha value of the controls the gradient of the negative part of the linear function.

3. Materials and Training

3.1 Data Preprocessing

The data set contains 2,478 images of pepper leaves, including 1,478 healthy and 1,000 bacterial infected images. Table 1shows the information of the data sets, the number of available images for pepper leaves (healthy and bacterial infection), and the percentage of images taken in the laboratory environment or under actual cultivation conditions. 19% of the available images of healthy plants in Table 1 were taken under real cultivation conditions in the field. The increased complexity of images under real conditions includes multiple leaves and other parts of plants, unrelated objects (such as soil), different ground textures, shadow effects, etc.

Table 1. Data set information

Plant common name	Disease common name	Disease scientific name	Images (number)	Laboratory Conditions (%)	Field Conditions (%)
Pepper, bell（healthy）	——	——	1478	81	19

Pepper, bell (Bacterial)	Bacterial spot	Xanthomonas campestris	1000	90	10

The whole database is initially divided into two data sets, the training set and the verification set. By randomly segmentation of 2478 images, 80% of them form the training set and 20% formed the verification set. We divide the data set into two parts: train and validation, and each part is divided into two subtypes: bacterial and healthy. First of all, we preprocess the image, including reducing the size and cutting to 256 × 256 pixels, rescale and rotation_range, random horizontal flip, image displacement, zoom_range and other data enhancement operations. It is not considered to use the gray version of the image for training, because previous work shows that this method can not improve the final classification performance of the deep learning model. This is because the neural network has the ability to recognize the important and unimportant features of a group of images, and ignores the latter to some extent.

3.2 Training process

We first create training set and verification set, and then make data preprocessing. Using the sequential model of Keras build the neural network, importing MobileNet V2 model, with dropout layer and LeakyReLU activation function. The network structure is shown in Table 2.

Table 2. Improved network structure

Layer	parameter	value
Dropout	rate	0.5
LeakyReLU	alpha	0.05
Dense	units	512
	Activation	ReLU
Dropout	rate	0.2
LeakyReLU	alpha	0.05
Dense	units	train_generator.num_classes
	activation	softmax
Note: rate is dropout rate, rate = 0.5 means 50% of random nodes will be hidden in each training, units is the number of neurons, activation is the activation function.		

Using the Adam optimizer, the loss function is set as the categorization cross entropy function, and the network evaluation index is marked as accuracy.

4. Result analysis

For classification problems, Accuracy and Loss value are used as evaluation indexes of model performance. Under laboratory conditions, the accuracy and loss values of each round of training set and verification set are shown in Fig 4. When the training round reaches the 61st round, the accuracy of the verification set is gradually fitted to the accuracy of the training set, the loss value converges to the lowest value, and the accuracy rate converges to the highest value, reaching 99.55%. After getting the target value label and the output value of our model, we can compare the same rate of the two. The predicted result is expressed as Predicted, the real label is expressed as Source, and the confidence is expressed as Confidence. Fig 5 shows the classified forecast of the output.

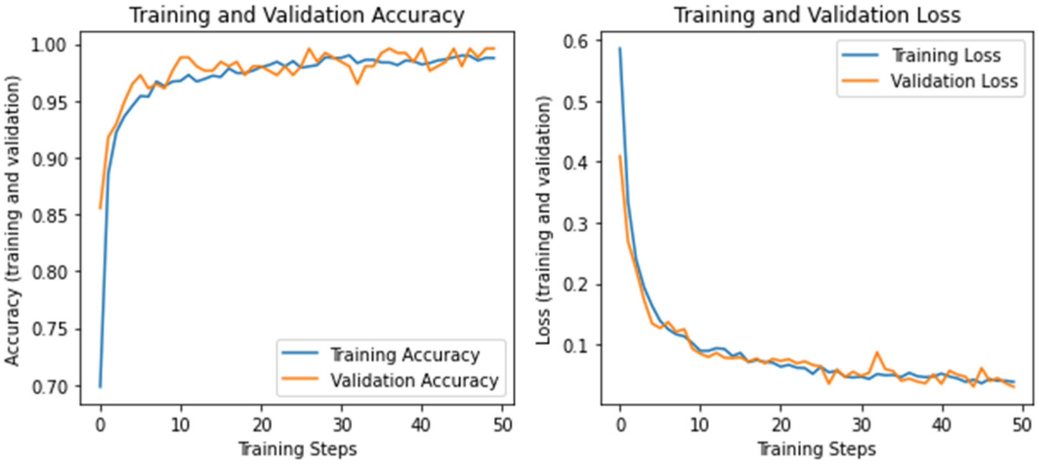
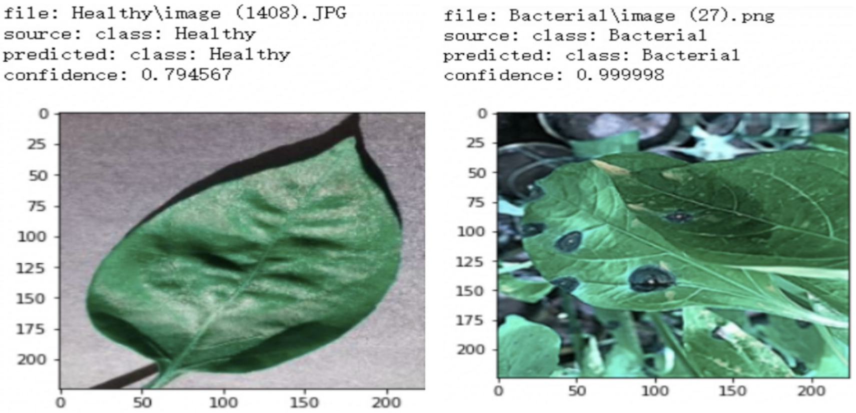


Figure 3. Accuracy and loss curves of training sets and validation sets under laboratory conditions



```
file: Healthy\image (1408).JPG
source: class: Healthy
predicted: class: Healthy
confidence: 0.794567
```

```
file: Bacterial\image (27).png
source: class: Bacterial
predicted: class: Bacterial
confidence: 0.999998
```

Figure 4. Prediction of leaf classification in the data set

It can be seen from the test results that the success rate of image recognition is significantly higher than that of images trained under real field conditions only under laboratory conditions. Meanwhile, it is found that the background of the image (shadow, ground, branches, etc.) also has an impact on the accuracy rate. The results show that image recognition in real culture conditions is more difficult and complex than in laboratory conditions, and demonstrate that the presence of images captured in real culture conditions is highly important for the development of useful and successful systems for automatic detection and diagnosis of plant diseases.

5. Conclusion

In this work, a specialized deep learning model based on the convolutional neural network structure was developed using the transfer learning method to detect disease in leaves images of healthy and disinfected plants. The model was trained using a data set of 2478 images of pepper, including 1478 healthy leaves and 1000 infected leaves. The best performance of the trained detection model is 99.55% accuracy when identifying disease or healthy plants. The results presented suggest that the presence of real conditional images (captured in the cultivation domain) in training data is very important. The trained model can be integrated into mobile applications for growers to use for disease detection in planting areas. A later stage is planned to expand the existing data set to include a wider range of crops and diseases.

Acknowledgments

This work was sponsored by Science and Technology Program of Sichuan Province, China (Grant No. 2020YFS0090, 2021YFN0117).

References

[1] Lee, S.H., Chan, C.S., Wilkin, P., Remagnino, P. 2015. Deep-plant: Plant identification with convolutional neural networks. 2015 IEEE Intl Conf. on Image Processing, pp. 452–456.

[2] Grinblat, G.L., Uzal, L.C., Larese, M.G., Granitto, P.M., 2016. Deep learning for plant identification using vein morphological patterns. Comput. Electron. Agric. 127, 418–424.

[3] Mohanty, S.P., Hughes, D.P., Salathé, M., 2016. Using deep learning for image-based plant disease detection. Front. Plant Sci. 7http://dx.doi.org/10.3389/fpls.2016. 01419. Article: 1419.

[4] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D., 2016. Deep neural networks based recognition of plant diseases by leaf image classification. Computat. Intelligence Neurosci. http://dx.doi.org/10.1155/2016/3289801. Article ID: 3289801.

[5] Fuentes, A., Yoon, S., Kim, S.C., Park, D.S., 2017. A robust deep-learning-based detector for real-time tomato plant diseases and pest recognition. Sensors 17, 2022. http://dx. doi.org/10.3390/s17092022.

[6] Sandler, M., A. Howard, Menglong Zhu, A. Zhmoginov and Liang-Chieh Chen. "MobileNetV2: Inverted Residuals and Linear Bottlenecks." 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018): 4510-4520.