Automatic Detection of Tomato Diseases and Pests Based on Leaf Images

Jia Shijie
Electronic&information college
Dalian Jiaotong University
Dalian, China
jsj@djtu.edu.cn

Jia Peiyi
School of Mechatronics&
Information Engineering
Shandong University（Weihai）
Weihai, China
pp_luck@163.com

Hu Siping
Electronic&information college
Dalian Jiaotong University
Dalian, China
jsj@djtu.edu.cn

Liu Haibo
Electronic&information college
Dalian Jiaotong University
Dalian, China
jsj@djtu.edu.cn

***Abstract*—There are many species of tomato diseases and pests, and the pathology of which is complex. It is difficult and error-prone to simply rely on manual identification. For the ten most common tomato diseases and pests in China, This paper explores the detection algorithms on leaf images and constructs the convolution neural network model to detect tomato pests and diseases based on VGG16[8] and transfer learning. The detection model is trained with Keras/TensorFlow deep learning framework and achieves an average classification accuracy of 89%.**

Keywords—Tomato diseases and pests; CNN; Leaf images; Transfer learning；

I. INTRODUCTION

There have been a variety of tomato pests and diseases which seriously affects the tomato yield, such as a). Tomato bacterial spot, b) Tomato early blight, c) Tomato late blight, d) Tomato leaf mold e) Tomato septoria leaf spot, f) Tomato two spotted spider mite, g) Tomato target spot h) Tomato mosaic virus , i) Tomato yellow leaf curl virus, j) Tomato gray spot, just as illustrated as Fig 1.(a~j corresponds to each row from top to bottom). In order to effectively control the pests and diseases, it is important to make accurate identifications. However, the process of identification usually relies on experiences and manual identification, which is laborious, time consuming and error-prone.

With the continuous development of pattern recognition and machine learning, the automatic recognition of plant disease types has been carried out by image processing technology [1] [2] [3] [4] [5]. Some of these methods detect plant disease with leaf surface images[4] [5]. In 2015, Sharada, David Hughes and Marcel [6] established a plant disease prevention institution CrowdAI[7]. They trained a deep convolution neural network to identify 26 diseases of 14 crops. When testing the model on a set of images collected from online sources, the model still achieves an accuracy of 31.4%, which is much higher than the one based on random selection (2.6%), a more diverse set of training data is needed to improve the general accuracy.

This paper constructs the convolution neural network model to detect tomato pests and diseases based on VGG16[8] and transfer learning. The VGG16 deep model is released by the University of Oxford in the 2014 ILSVRC competition and has achieved the championship for the year's image recognition group. As the original task, VGG16 is trained with 1.26 million images to classify 1000-category images, while the target task is to classify 10- category leaf images of tomato pest and diseases., As a transfer learning technolgy [9], fine-tuning is employed to transfer the original task and data domain to the target task and data domain. This paper builds an image dataset of tomato diseases and pests, which contains leaf images of ten common tomato diseases in China as well as some normal ones. The dataset has a total of 11 categories, 7040 images, each category has 640 images.

II. MTHODS

Two algorithms are employed, respectively. The first(*VGG16+SVM*)) is to employ VGG16 as image feature extractor, and combine SVM classifier to detect tomato pests and diseases on leaf images; the second (*Fine-tuning*) is to use fine-tuning to construct an end-to-end classification model based on the original VGG16 model.

A. VGG16+SVM

The input image is passed through a stack of convolution layers. The input to the convnet is a fixed-size 224 × 224 RGB image, which is preprocessed to subtract the mean RGB value computed on the training set. The filters with 3*3 receptive field, and the convolution stride is fixed to 1 pixel; the spatial padding is 1 pixel to preserve the spatial resolution after convolution. Spatial pooling is carried out by five max-pooling layers over a 2 × 2 pixel window. A stack of convolutional layers is followed by three Fully-Connected (FC) layers: the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). All hidden layers are equipped with the rectification(ReLU) non-linearity. In this scheme, the outputs of the penultimate fully connected layer are extracted as the feature descriptors. These descriptors of training images and their corresponding labels are employed to train SVM classifier.

SVM (Support Vector Machine)[10] is a classification algorithm which can improve the generalization ability of learning machine by finding structural risk minimization, and realize the minimization of experience risk and confidence

This work is supported by Liaoning Provincial Natural Science Foundation(201602118).

interval. Computing the (soft-margin) SVM classifier amounts to minimizing an expression of the form

Fig 1. Leaf images of tomato pests and diseases

$$\frac{1}{n}\sum_{i=1}^{n}\max(0,1-y_i(\omega * x_i + b)) + \lambda\left\|\omega^2\right\| \qquad (1)$$

Where x_i is the ith sample in the training dataset *and* y_i indicats the class label to the point x_i.

SVMs can efficiently perform a non-linear classification using the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

B. *Fine-tuning*

The original VGG16 model has 3 fully-connected layers which output 4096, 4096 and 1000 nodes, respectively. In our scheme, 3 fully-connected layers are replaces with two fully-connection layers, the nodes of which are 2048 and 11 (corresponding to the ten categories of pests and diseases and the normal one, respectively.), the number of parameters for each layer of the migration learning model is shown in Table 4.1.

Table 1 Number of parameters for each layer

Layer	output shape	#parameters
convolution1_1	(64, 224, 224)	1792
convolution1_2	(64, 224, 224)	36928
maxpooling_1	(64, 112, 112)	0
convolution2_1	(128, 112, 112)	73856
convolution2_2	(128, 112, 112)	147584
maxpooling_2	(128, 56, 56)	0
convolution3_1	(256, 56, 56)	295168
convolution3_2	(256, 56, 56)	590080
convolution3_3	(256, 56, 56)	590080
maxpooling_3	(256, 28, 28)	0
convolution4_1	(512, 28, 28)	1180160
convolution4_2	(512, 28, 28)	2359808
convolution4_3	(512, 28, 28)	2359808
maxpooling_4	(512, 14, 14)	0
convolution5_1	(512, 14, 14)	2359808
convolution5_2	(512, 14, 14)	2359808
convolution5_3	(512, 14, 14)	2359808
maxpooling_5	(512, 7, 7)	0
flatten_1	(28088)	0
dense_1	(2048)	51380224
dropout_1	(2048)	0
dense_2	(11)	22528

III. EXPERIMENT

1 Experiment set

All the experiments are tested on the computer equipped with 2.40GHz CPU, 16G memory, and one NVIDIA GeForce GT 1080. The operating system is Linux Ubuntu 14.04. TensorFlow 1.0& Keras 1.0.3 is employed as the deep learning framework. The dataset is divided into three independent data sets: the training set, the validation set and the test set. The training set accounts for 62.5% of the total dataset (400 images per category), the validation set accounts for 25%(160 images per category), and the test set accounts for 12.5 %(80 images per category).

The super-parameters are set as follows：

Learning rate 0.01

Batchsize：40

Epoch：180

Decay：1e^{-6}

Momentum：0.9

Optimizer：SGD

Four data enhancement methods are employed to expand the dataset: rotating, flipping, inverting, scaling and translating. Figure 2 illustrates the original image, the 30^0 rotated image, the horizontally flipped image, the zoomed image and the left-shifted image, respectively.

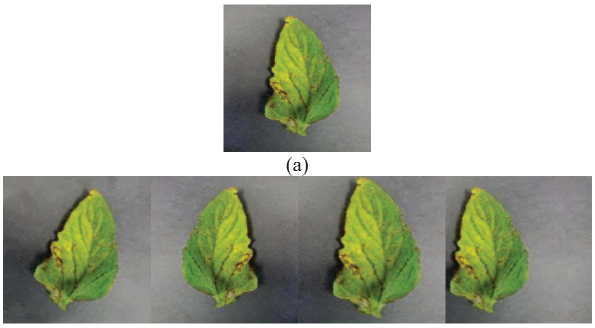

Fig. 2 Image data enhancement

(a)The original image (b) The rotated image (c) The horizontally flipped image (d) The zoomed image (e)The image shifted to the left.

2. Experiment result

2.1 VGG16+SVM:

The average accuracy of the training set reaches 100%, while that of the test set degrades to 88%. The training process costs 3h to converge, and the testing time is 0.08s/image.The test result of each category and the confusion matrix are illustrated as Table 2 and Table 3, respectively.

2.2 *Fine-tuning*

The average accuracy of the training set reaches 98%, while that of the test set degrades to 89%. The training process costs 4.5h to converge, and the testing time is 0.12s/image. The test result of each category and the confusion matrix are illustrated as Table 4 and Table 5, respectively.

The following conclusions can be drawn from the above result:

(1) Both the two algorithms attain the test accuracies range from 84% ~100%. The yellow leaf curl virus is the most easily detectable pest and disease (100%) and the tomato late bligh is the hardest one(84%), which is consistent for the two algorithms.

(2) The fine-tuning model performs slightly better than the VGG16+SVM model. Both the two algorithms can achieve real-time test performance. However, the training process of the fine-tuning model is much more time-consuming.

Table 2 The test results of each category (VGG16+SVM)

category	Total test number	Correct number	Accuracy
Normal	80	72	90%
Tomato bacterial spot	80	70	88%
Tomato early blight	80	69	86%
Tomato late blight	80	67	84%
Tomato leaf mold	80	72	90%
Tomato septoria leaf spot	80	69	86%
Tomato two spotted spider mite	80	64	80%
Tomato target spot	80	69	86%
Tomato mosaic virus	80	66	82%
Yellow leaf curl virus	80	80	100%
Tomato gray spot	80	75	94%
Total	880	774	88%

Table 3 Confusion matrix(VGG16+SVM)

	a	b	c	d	e	f	g	h	i	j	k
a	72	0	0	1	0	1	4	1	0	0	1
b	0	70	0	2	0	2	0	1	2	0	3
c	0	0	69	3	2	2	2	0	1	0	1
d	1	2	3	67	0	2	1	2	0	1	1
e	0	0	2	0	72	2	4	1	0	0	1
f	1	0	0	2	0	69	0	0	4	0	4
g	8	0	2	2	0	2	64	0	2	0	0
h	1	1	0	2	1	4	0	69	4	0	3
i	0	0	2	0	2	4	2	4	66	0	0
j	0	0	0	0	0	0	0	0	0	80	0
k	0	0	2	1	0	0	0	1	1	0	75

Table 4 The test results of each category (Fine-tuning)

Category	Total test number	Correct number	Accuracy
Normal	80	72	90%
Tomato bacterial spot	80	70	88%
Tomato early blight	80	69	86%
Tomato late blight	80	69	86%
Tomato leaf mold	80	72	90%
Tomato septoria leaf spot	80	69	86%
Tomato two spotted spider mite	80	67	84%
Tomato target spot	80	69	86%
Tomato mosaic virus	80	70	88%
Yellow leaf curl virus	80	80	100%
Tomato gray spot	80	75	94%
Total	580	782	90%

Table 5 Confusion matrix(Fine-tuning)

	a	b	c	d	e	f	g	h	i	j	k
a	72	0	0	1	0	1	2	1	2	0	1
b	0	70	0	1	3	2	1	1	2	0	0
c	0	0	69	3	2	2	2	0	2	0	0
d	1	0	3	69	0	2	1	2	0	1	1
e	0	0	2	0	72	2	4	0	0	0	0
f	1	0	2	2	0	69	0	0	4	0	0
g	6	0	2	2	1	2	67	0	0	0	0
h	0	1	0	2	1	0	0	69	4	0	3
i	1	2	3	0	2	0	2	0	70	0	0
j	0	0	0	0	0	0	0	0	0	80	0
k	1	0	1	1	0	0	0	1	1	0	75

(3) Compared with our manual designed 5-layer CNN network(the average test accuracy is 84%), the proposed VGG16-based models are more effective, up to 88% and 89%, respectively.

IV. CONCLUSION

This paper mainly studies the automatic detection of tomato pests and diseases based on leaf surface. The detection models are trained to classify the tomato diseases and pests by transfer learning technology, which achieves an average classification accuracy of 89%. However, the overall high performance rely on relative high-quality test images(i.e. simple background, object-centred, positive close-up shooting), future research will focus on the complicate algrithms to detect tomato pests and diseases based on relative low-quality leaf images.

REFERENCES

[1] C. Mix, F. X. Pico and N. J. Ouborg. "A Comparison of Stereomicroscope And Image Analysis For Quantifying Fruit Traits [J]. SEED Technology, 2003.25(1):9.

[2] Mokhtarian F, Abbasi S. Matching shapes with self-intersections: APPlication to leaf classification[C]. IEEE Trans on Image Processing, 2004.13(5): 653-661.

[3] Casanova D, de Mesquita Sa Junior J J, Bruno O M. Plant leaf identification using neural networks[J]. International Journal Of Imaging Systems And Technology, 2009. 19(3): 236-243.

[4] Ehsanirad A, Sharath Kumar Y H. Leaf recognition for plant classification using GLCM and PCA methods[J]. Oriental Journal of Computer Science and Technology, 2010. 3(3): 31-36.

[5] Rossatto DR, Casanova D, Kolb RM, et al. Fractal analysis of leaf-texture properties as a tool for taxonomic and identification purposes:a case study with species from Neotropical Melastomataceae (Miconieae tribe) [J]. Plant Systematics and Evolution, 2011. (291): 103-116.

[6] Sharada Prasanna Mohanty David Hughes, and Marcel Salath. Using Deep Learning for Image-Based Plant Disease Detection[J]. Digital Epidemiology Lab, EPFL, Switzerland, 2015. (3):42-48.

[7] CrowdAI. https://www.crowdai.org/challenges/1/dataset_files

[8] K. Simonyan and A. Zisserman. Very deep convolutional networksfor large-scale image recognition[J]. In ICLR, 2015.(3):53-66.

[9] Durjoy Sen Maitra; Ujjwal Bhattacharya; S.K. Parui.CNN based common approach to handwritten character recognition of multiple scripts[J]. In Document Analysis and Recognition (ICDAR), 2015.(6):1021-1025.

[10] Ben-Hur, Asa, Horn, David, Siegelmann, Hava, and Vapnik, Vladimir; Support vector clustering[J] Journal of Machine Learning Research, 2001.(2): 125–137.