

## A basic study on leaves detection with deep learning features

Huu Quan CAP<sup>1</sup> Erika Fujita<sup>1</sup> Katsumasa Suwa<sup>1</sup>  
Satoshi Kagiwada<sup>2</sup> Hiroyuki Uga<sup>3</sup> Hitoshi Iyatomi<sup>1</sup>

### ABSTRACT

Preventing plant diseases and their early detection are essential for reducing monetary loss. Several studies have used machine learning methods to diagnose plant diseases and achieved attractive results. As far as our best knowledge, however, most of these methodologies only accept a single leaf image as their input. Thus, they are time-consuming in practical situations. In this paper, we propose a method detecting multiple boundary boxes of leaf regions from on-site wide-angle images (e.g. images or videos from stationary surveillance camera) with deep learning approach. Our method attained 66.2% of F-score measurement.

### 1. Introduction

Plants have been faced with many dangerous diseases and their monetary loss is estimated to 50 million dollars annually [1]. Detecting and preventing plant diseases promptly is essential to help relieve this issue. The general way to diagnose plants is by experts, but they are usually expensive and time consuming. Some computer methods for automated plant diagnosis have been proposed to solve this issue. Huang et al. [2] proposed a method to diagnose diseases on orchid seedlings leaf using artificial neural networks (ANN) and attained classification accuracy over 89%. Yao et al. [3] used support vector machine (SVM) to diagnose rice diseases and it showed 89% of diagnostic accuracy. Mohanty et al. [4] used convolutional neural network (CNN) to diagnose multiple plants diseases and they attained 99% of accuracy. Although these studies have achieved good results, they still have some problems. Firstly, as shown in Fig.1, we can see that those above methods cannot diagnose the real case images. Huang, Yao, and Mohanty have achieved very high classification performance, but their systems did not use the practical images (a, b, c). While on the other hand, Fujita et al. [5] proposed practical diagnosis system with CNN accepting on-site cucumber images photographed under various conditions (d). They classified seven types of major cucumber viral diseases and normal plants. Their classification accuracy reached 83% even they included ill-conditioned images.

Secondly, the above methods only accept single leaf image as an input. Therefore, diagnosing on-site images (e.g. the images are taken with real condition by surveillance camera) is time consuming.

As far as our knowledge, there are no organized studies on leaf detection for following classification stage from wide-angle pictures. Compared with the other commonly seen detection tasks such as face detection, pedestrian detection, etc., leaf detection is more difficult. In leaf detection task, the object to be detected and its background is the same and, in addition that they have often heavily overlapped each other.

In this paper, we propose an easy and practical method to localize fully leaf in wide-angle images based on sliding windows and CNN for following diagnosis stage. Currently, our system is designed to be combined with practical diagnosis system especially such as Fujita's system. The proposed CNN architecture is efficient because it can classify the fully leaf with a small input size (32x32 pixels). Using smaller input size is not

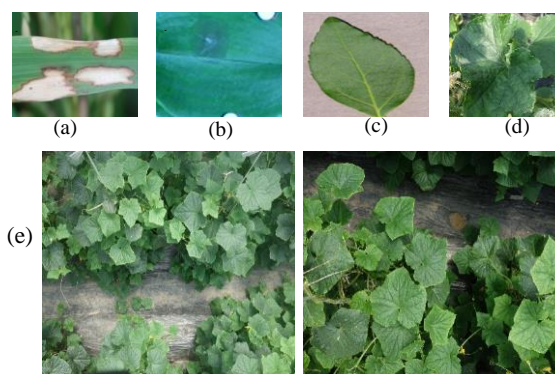


Fig 1. The comparison between single leaf input (a, b, c, d) and wide-angle images (e).

also to decrease the complexity of the CNN model, but also helps our system detect leaf faster.

### 2. Proposed leaf detection system

Fig.2 show the whole schematics of practical plant diagnosis system including proposing our leaf detection system. Our leaf detection system is designed to be combine with diagnosis system behind. Our system can be divided into 3 steps. Firstly, given a wide-angle image, sliding window with several window sizes are used to extract numerous candidate boxes that may contain fully leaf regions. Secondly, specially trained CNN classifier analyzes those boxes to find locations of fully leaf. Finally, the non-maximum suppression (NMS) is used to remove the overlapping bounding boxes. The fully leaf regions will be located after NMS stage.

#### 2.1 Exhaustive search

##### 2.1.1 The wide-angle images

For step 1 in Fig.2, we collected 170 wide-angle images. Each image contains multiple leaves with size of 2592x1944 pixels and was taken by Sony DSC-RX100 camera in daytime with different conditions. This dataset is the “wide-angle images” dataset. The examples of 2 wide-angle images are shown in Fig.1(e).

##### 2.1.2 The proposed exhaustive search based sliding windows

For detecting leaf, sliding windows based on exhaustive search strategy is used to search fully leaf regions with the different search windows and certain step sizes. Since the size of images

1 – Hosei University, Dept. Applied Informatics

2 – Hosei University, Dept. Clinical Plant Science

3 - Saitama Prefectural Agriculture and Forestry Research Center

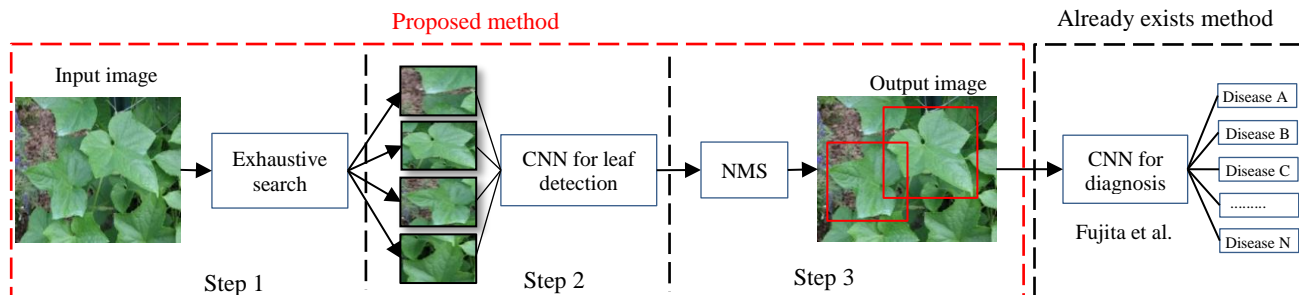


Fig 2. The Leaf Diagnosis System. Firstly, many boxes will be extracted and classified with the CNN model. After NMS, the fully leaf regions will be located base on classification result (red boxes)

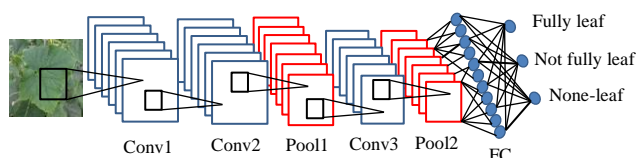


Fig 3. The proposed CNN architecture consists of 7 layers (6 hidden layers and 1 output layer).

from “wide-angle images” dataset is large (2592x1944), searching leaf location on that size costs time and inefficiency. For saving time, we first resized the original image from 2592x1944 pixels to 200x150 pixels based on our preliminary experiments and then search with different window sizes on those resized images.

Our system can detect the various size of leaf by applying the different window search sizes. Here, we detect leaf regions for the following classification system, we only search for the fully leaf which has the minimum size of [250x250] from the wide-angle images (c.f. Fujita’s system accepts input size of 224x224). It means the minimum window size to search on those resolution [200x150] images should be [20x20]. We applied 6 searching windows with [20x20], [25x25], [30x30], [35x35], [40x40], and [45x45] on those resized images. Each window has step size equal to 20% of their size.

## 2.2 The proposed CNN for leaf detection

The CNN for leaf detection in Fig.2 is three classes classifier, namely it discriminates input image patch cropped with exhaustive search as either of “fully leaf”, “not fully leaf” or “none-leaf”. The objective of this study is to determine the boundary box of “fully leaf”, which is acceptable for the following diagnosis step or not. Both locations of “not fully leaf” and “none-leaf” should not be detected, but the former include leaf and the latter do not. Since the appearance and image property of them are completely different, we left two different classes even if they should be rejected.

### 2.2.1 Dataset for training CNN

As for training of this CNN, we prepared a total of 720,000 image patches, consisting of 240,000 each for the class. For “fully leaf” class, we used a total of 60,000 cucumber leaf images with 112x112 pixels from Saitama Prefectural Agriculture and Forestry Research, Japan. Each image contains a single leaf in the center and the distance between camera and leaf is from 0.5 to 0.8 m. For these images, we augmented them with clockwise

Table I. The proposed CNN architecture.

Layer name and size [depth x width x height]	Filter before convolution (size, # filters, padding, stride)
Input – [3x32x32]	[3x3], 32, 1, 1
Conv1 – [32x32x32]	[3x3], 64, 1, 1
Conv2 – [64x32x32]	[2x2], 64, 0, 2
Pool1 – [64x16x16]	[3x3], 64, 1, 1
Conv3 – [64x16x16]	[2x2], 64, 0, 2
Pool2 – [64x8x8]	None
FC [1x100]	None

Table II. The comparison between 4 different CNN models.

Input size	16x16	28x28	32x32	56x56
Speed (s)	5.3	7.1	8.5	18.7
Accuracy (%)	86.9	90.4	98.6	98.8

rotation with the incremental step size of 90 degrees. The total number of images becomes 4 times larger with 240,000 images. For “not fully leaf” class, we divide each of abovementioned “fully leaf” image into quarters, i.e. total 60,000x4=240,000 images and accordingly each has 56x56 pixels. Here, the same leaf is included both in “fully leaf” and “not fully leaf” datasets. We expect this strategy helps to boost the discrimination performance between those classes rather than using completely different dataset under our situation those leaves are overlapped each other. For “non-leaf” class, 240,000 images were collected from ImageNet dataset [6] with the minimum size is 56x56 pixels. We use this dataset for training, testing, and validation for the CNN (We will explain the detail in the experiment part).

### 2.2.2 The proposed CNN architecture

Architecture of the proposed CNN is shown in Fig.3 and Table I. It accepts a color image with  $S \times S$  pixels as an input and consists of 6 hidden layers and 1 output layer with 3 units (“fully leaf”, “not fully leaf”, “none-leaf”). We put 100 units in FC layer fully connected with *pool2* and used dropout technique [7] with the ratio of 50%.

In order to determine appropriate input image patch size  $S$ , we performed preliminary experiments. As shown in Table II, we compared the processing speed and accuracy for 1000 test images on 4 different CNNs with different input size  $S \times S$ , namely: 56x56, 32x32, 28x28, and 16x16. The results show that the bigger input size the better accuracy in exchange of calculation cost. Because of that trade-off, we selected  $S=32$ , i.e. input size of the CNN is 32x32.

### 3. Experiments and result

#### 3.1 Training the CNN

As mentioned in section 2.2.1, totally 720,000 images will be divided into 3 datasets: Training set (432,000 images), Validation set (144,000 images), Testing set (144,000 images). Each divided dataset has the same amount from 3 classes (fully leaf, not fully leaf, none-leaf). Firstly, the training set and validation set are used during training stage. The classifier was trained with 5 epochs (1 epoch is equivalent with 1 training time). For each epoch, the validation set is used to test with the classifier. The GPU GTX 1080Ti was used to train the CNN model for nearly 4 hours. After training, the accuracy is 98.6% and the validation error rate is 4.9%. Secondly, the trained CNN model was used to classify 144,000 images from the testing set and achieved 97.8% of accuracy. Since the testing set is independent from training and validation set. With small validation error (4.9%) and high training (98.6%) and testing accuracy (97.8%), we can say that our model can control overfitting and robust for fully leaf classification.

#### 3.2 Testing the whole system

To evaluate our system, we tested and made the fully leaf ground-truth from the same “wide-angle images” dataset. Around 3500 fully leaves which have the minimum size of [250x250] were labeled with a bounding box. For each image, we used sliding windows on to extract nearly 4000 boxes. The proposed CNN takes around 30 seconds to classify all extracted boxes. Base on all scored boxes after classification, we apply NMS to reject boxes if they have an intersection-over-union (IoU) overlap with a higher scoring selected box larger than a certain threshold. In this system, we select the IoU threshold equal to 0.2 to run with NMS.

F1-score criteria is used to measure our system. Given the result bounding boxes and ground-truth, the F1-score is calculated by the following equations:

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Where:

$$\begin{aligned} \text{Precision} &= \frac{\text{Number of correct detected boxes}}{\text{Number of detected boxes}} \\ \text{Recall} &= \frac{\text{Number of correct detected boxes}}{\text{Number of ground-truth boxes}} \end{aligned}$$

We treat a detected box as correction box if the IoU of that box with respect to a ground-truth box is  $\geq 0.5$ . After detecting all “wide-angle” images, we achieved the average 66.1% of precision, 67.7% of recall, and 66.2% of F1 score.

### 4. Discussion & conclusion

The result 66.2% of F1 score is reasonable for this task. Considering the practical application of whole plant diagnosis schema in Fig.2, it is not necessary to detect exactly the whole fully leaf from the images. In the fact that we need to detect

initial infection in farm and nearby leaf probably have the same disease.

In addition, using greedy search is often not appropriate for some application due to the computational cost, but in this task the processing time of 30 sec is acceptable level. Considering this, we prefer more focus on precision to does on recall. We observed not small differences between the classification performances between unit test on CNN (97.8%) and whole test (66.2%). We think our CNN was trained with the well-focused leaf images, but some extracted boundary boxes from wide angle images did not.

In this paper, even we achieved a promising detection rate but we still do not completely satisfy current performance. Because of using the “fixed” windows sizes, we somehow miss the fully leaf regions when using sliding windows. Further develop the searching strategy is needed to solve this problem. In near future, we should improve our searching strategy and increase the CNN performant to become more suitable for practical system.

#### Acknowledgement

This research was partially supported by the Ministry of Education, Culture, Science and Technology of Japan (Grant in-Aid for Fundamental research program (C), 17K8033, 2017-2020).

#### References

- [1] K. Sastry, "Introduction to Plant Virus and Viroid Diseases in the Tropics," Plant Virus and Viroid Diseases in the Tropics, Springer Netherlands, 2013, pp.1-10.
- [2] K. Y. Huang, "Application of artificial neural network for detecting Phalaenopsis seedling diseases using color and texture features" Computers and Electronics in agriculture 57.1, pp.3-11, 2007.
- [3] Q. Yao, Z. Guan, Y. Zhou et al., "Application of Support Vector Machine for Detecting Rice Diseases Using Shape and Color Texture Features," 2009 International Conference on Engineering Computation, Hong Kong, 2009, pp. 79-83.
- [4] E. Fujita, Y. Kawasaki, H. Uga et al., "Basic Investigation on a Robust and Practical Plant Diagnostic System," 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, 2016, pp. 989-992.
- [5] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Frontiers in plant science 7, 2016.
- [6] J. Deng, W. Dong, R. Socher et al., "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, 2009, pp. 248-255.
- [7] N. Srivastava, G. E. Hinton, A. Krizhevsky et al., "Dropout: A simple way to prevent neural networks from overfitting," The Journal of Machine Learning Research 15.1, 2014, pp.1929-1958.