International Journal of Engineering & Technology, 7 (4.11) (2018) 106-111

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

An End-To-End Practical Plant Disease Diagnosis System for Wide-Angle Cucumber Images

Q. H. Cap[1], K. Suwa[1], E. Fujita[1], S. Kagiwada[2], H. Uga[3], H. Iyatomi[1]*

[1]Applied Informatics, Graduate School of Science and Engineering, Hosei University, Japan
[2]Clinical Plant Science, Faculty of Bioscience and Applied Chemistry, Hosei University, Japan
[3]Saitama Agricultural Technology Research Center, Japan
**Corresponding author E-mail: iyatomi@hosei.ac.jp*

Abstract

With the breakthrough of deep learning techniques, many leaf-based automated plant diagnosis methodologies have been proposed. To the best of our knowledge, most conventional methodologies only accept narrow range images, typically one or quite a limited number of targets are in their input. This is because the appearance of leaves is diverse and leaves usually heavily overlap each other in practical situations. In this paper, we propose a basic and practical end-to-end plant disease diagnosis system for wide-angle images. Our system is principally composed of two specially designed types of convolutional neural networks. The system achieves leaf detection performance of 73.9% in F1-score, overall (detection and diagnosis) performance of 68.1% in recall and 65.8% in precision at around 3 seconds/image on 500 wide-angle on-site images which have 6,860 healthy and 6,741 infected leaves (13,601 in total).

Keywords: *deep learning; convolutional neural networks; cucumber plant disease; automated disease diagnosis; object detection*.

1. Introduction

Loss of crop yield due to plant diseases is one of the most serious and longstanding problems in the development of worldwide agriculture. Early detection and appropriate treatment are crucial steps in efforts to increase crop production and productivity. There are several ways to analyze plant diseases including visual inspection by experts, biological examination, or automatic computer-based diagnosis systems. The problem with visual inspection by experts and biological examinations are that those analyses are often time-consuming, expensive and fail to identify diseases in timely manner. In this context, many automatic computer-based diagnosis methodologies which are capable of identifying diseases in a rapid and reliable way have been recently proposed.

In [1] used conventional image segmentation techniques and a support vector machine (SVM) [2] to classify four types of alfalfa diseases in leaves. The SVM classifier achieved an accuracy of 94.7% on images in a laboratory environment. A work in [3] proposed a fast method to identify four sugar beet diseases using smartphones. They extracted multiple hand-crafted features on sugar beet leaf images and trained a classifier using with SVM. Their system achieved 82.0% classification accuracy. A smartphone-based system to diagnose five classes (four types of disease and a healthy diagnosis) in cassava was built [4]. The system used three classifiers (Linear SVM, K-Nearest neighbor and extremely randomized trees) with a hand-crafted feature called Color and Oriented FAST and Rotated BRIEF (ORB) [5] and achieved over 99% accuracy. These methods successfully established preferable performance for their own target task. However, since they are designed based on conventional pattern recognition techniques, i.e. sequential process of (1) pre-processing including segmentation, detection of the regions of interests (ROIs), etc., (2) development of hand-crafted features specially designed for that specific task, and (3) classification, they usually have constraints on their usage.

In the recent past, convolutional neural networks (CNN) have demonstrated tremendous success in object recognition and image classification tasks. Since the breakthrough of AlexNet [6] in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC 2012) [7], many automatic plant diagnosis systems have used CNN for identifying plant diseases. In [8] designed their own CNN model to classify four types of apple diseases by combining AlexNet and GoogLeNet [9]. Under a controlled environment, their model had considerably smaller parameters than AlexNet while they achieved an overall accuracy of 97.6%. In [10] applied CNN-based transfer learning. More specifically, they used pre-trained AlexNet and GoogLeNet to classify 26 diseases from 14 kinds of plants on the PlantVillage dataset [11] and attained 99.3% classification accuracy. With open access and large number of disease images, the PlantVillage dataset has also been widely used for plant disease classification systems. Another work from [12] applied transfer learning with VGG-Net [13] and classified the apple black rot disease on the same PlantVillage dataset, showing an accuracy of 90.4%.

Tomato diseases have also received attention from researchers. The researchers from [14] investigated transfer learning on AlexNet and SqueezeNet [15] to classify ten classes (nine diseases, one healthy) of tomato leaves from the PlantVillage dataset. They showed a competitive accuracy of 94.3% on SqueezeNet compared to AlexNet with 95.6%. Thus, the small size of SqueezeNet is well suited for mobile applications. In another work in the same tomato dataset, in [16] showed a test accuracy of 99.9% while classifying ten tomato diseases with deep residual learning [17].

Although the above methods achieved remarkable results, they have a major drawback in that most of the photographic images are taken in a laboratory setting (i.e. each leaf is manually cropped and placed on a uniform background), not under real conditions in

the cultivation field. In fact, in [10] also noted in their manuscript that the accuracy dropped to around 31% (from a performance of 99%) under different settings from the training images. Note here that we found a significant number of inappropriate training labels assigned and currently this dataset is not available to the public.

On the other hand, several other diagnosis systems have also proven the reliable of CNN-based methods under practical conditions. In [18] collected on-site banana images from the PlantVillage dataset and re-trained the LeNet model [19] to differentiate two banana diseases from a healthy state with an accuracy of 99.7%. A study from [20] trained their own CNN on an on-site cucumber leaf dataset (including seven types of diseases and one healthy state) and showed an average of 82.3% accuracy with various backgrounds and photographic conditions. In [21] used CNN and got an overall accuracy of 96.3%, while identifying 13 types of disease in five crops using images downloaded from the Internet. Research work in [22] attained a 99.5% success rate in identifying the corresponding [plant, disease (or healthy)] combination in 58 distinct classes using a dataset taken from both laboratory settings and cultivation fields. In [23] investigated on-site cassava leaves and reported an overall accuracy of 93%, while classifying six classes (five diseases and a healthy state) using transfer learning and deployed a real-time mobile application. Also, a transfer learning CNN-based model was proposed to diagnose healthy, diseased, and snail-infested states on wild rice images [24] and achieved a 91.2% accuracy classification performance.

In the meantime, CNNs also demonstrated brilliant performance in the simultaneous processing of object detection and localization. Recently, many state-of-the-art methodologies have been proposed [25-28] which are designed jointly with or implemented on CNNs. Inspired by that work, some interesting diagnosis systems are not only detecting the diseases but also localizing their involved areas. In [29] used three CNN-based systems (i.e. Faster R-CNN [25], R-FCN [26] and SSD [27]) which performed object localization and diagnosis processes simultaneously. Their system achieved 86.0% mean average precision on annotated tomato leaf images. In [30] designed a framework to do both localization and diagnosis for wheat diseases with a fully convolutional network. Their system achieved 98.0% mean recognition accuracy on a wheat disease database (WDD2017) and can be deployed for mobile applications.

The above sophisticated systems have achieved excellent performance on a wide variety of images, but all of their targets are narrow range images, i.e. the ROIs are generally located in the center of the input. Fig. 1 is a comparison between the input images of the abovementioned systems and the wide-angle images taken in practical situations. The images from (a-f) in [18, 20, 21, 23, 29, 30] with various backgrounds have a narrow range compared to the wide-angle images (g, h). Furthermore, as we experienced, practical wide-angle images are extremely complex with multiple leaves overlapping each other along with a wide variety of backgrounds, lighting conditions, distance between camera and each leaf, etc. Furthermore, plant symptoms are highly diverse. Thus, even though the abovementioned simultaneous localization and identification works well on narrow range images, simultaneous processing for wide-angle images is quite difficult. Developing a diagnosis system for wide-angle images (e.g. taken by surveillance cameras) is however necessary in practical situations. In this case, we believe the location of each leaf should be extracted and analyzed independently with other followed by diagnosis schema.

In a study of leaf detection from wide-range images, in [31] attained a result of 87.7% on pepper leaves in wide-range images using modified active shape models. Their method is applied in greenhouse conditions with a clear background and can only detect leaves with a roughly similar shape. They admitted that their method has difficulties in application to outdoor environments with overlapped leaves. A study from [32] proposed a CNN-based method for wide-angle cucumber leaf detection. Their method achieved 78.0% of F1-score at 2.0 fps with complex backgrounds and overlapped in-field wide-angle cucumber images. They showed a promising result, but their system only focused on detection, and the experiments were conducted only on images having healthy leaves.

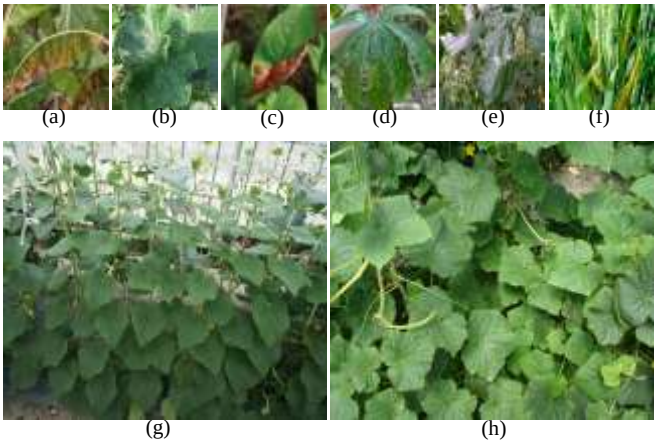

Fig. 1: The comparison between the narrow range images (a, b, c, d, e, f) and wide-angle images (g, h). Wide-angle images are often complex due to the heavy overlap of multiple leaves. Also, symptoms are scattered in different leaves (h)

To the best of our knowledge, there is no literature providing end-to-end plant diagnosis solutions for wide-angle images.

We should note here that when considering practical usage, we need to find the location of infected plants in the image, but it is generally not necessary to detect and diagnose all of the leaves since the infection of plant diseases often spreads from nearby on the site.

In this paper, we first propose a basic end-to-end plant disease diagnosis system for cucumber wide-angle images. Our system localizes a sufficient number of healthy and diseased leaves from wide-angle images and diagnoses them. Note that the reason we handle only two classes of diagnosis (healthy or diseased) is mainly due to the difficulty of quantitative evaluation. It is quite difficult to assign an appropriate gold standard even by experts for wide-angle images. Once suspicious leaf areas are detected, we can apply a conventional but sophisticated leaf diagnosis system mainly designed to diagnose one leaf on those locations.

The remainder of this paper is organized as follows. Section 2 describes the whole schematic of our proposed end-to-end plant diagnosis system including the leaf detection part and the leaf diagnosis part for identifying cucumber diseases. Section 3 analyses the experimental results of the leaf detection part, the leaf diagnosis part, and the overall result of the end-to-end system on 500 wide-angle images. Finally, Section 4 and Section 5 contain the discussion and conclusion, respectively.

2. The End-To-End Practical Cucumber Plant Diagnosis System

Our end-to-end plant diagnosis system consists of two main parts: (1) the leaf detection part, and (2) the leaf diagnosis part. Fig. 2 shows the schematics of our system. When the wide-angle input image is given, the leaf detection part works to localize and extract all the cucumber leaves of an appropriate size with the whole part almost visible as "full leaf" regions. Then, each of these detected regions is classified as healthy or diseased by the leaf diagnosis part. The system then finally synthesizes and displays the corresponding combination (location of leaf, disease/healthy) from the wide-angle input image.

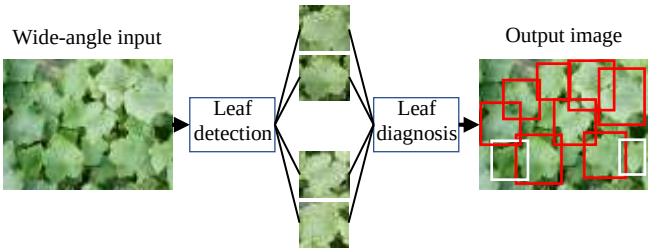

Fig. 2: Schematic of our end-to-end plant diagnosis system. Given the wide-angle input image, the leaf detection part will localize and extract full leaves to be diagnosed by the leaf diagnosis part. The result will be synthesized and displayed. Red boxes indicate disease leaves while white boxes indicate healthy leaves

We collected a total of 60,000 unique cucumber leaf images from Saitama Agricultural Technology Research Center, Japan. Each image contains a single cucumber leaf roughly in the center and surrounded with various backgrounds. We refer to these as "original images". Note here that all the leaf images have confirmed diagnoses. There are 15,379 healthy leaf images and the rest are cases of some infection.

2.1. The Leaf Detection Part

The objective of the leaf detection part is to localize each leaf from the wide-angle input image.
This part consists of a refinement of our previous work [32] and it discriminates input image patches cropped by an exhaustive search as one of "full leaf", "not full leaf" or "no leaf" with a specially designed CNN classifier - "CNNdetect". The architecture of CNNdetect is described in Fig. 3. First, given a wide-angle image, our method reduces input to the size of 200x150 and applies a sliding window with eight searching window sizes S×S (S = 20, 25, 30, 35, 40, 45, 50, 55) to extract numerous candidate boxes that may contain full leaf regions as the target. Second, CNNdetect classifies those windows to find the locations of leaves that should be detected in the boundary box format, i.e. select boxes identified as "full leaf". Finally, the non-maximum suppression (NMS) is used to remove the overlapping bounding boxes from the detected results.
Our former work left room for improvement on robustness under various lighting conditions. In practical conditions, the brightness is distributed unevenly across the entire image. In this work, we made an improvement on this by introducing a brightness adjustment method.

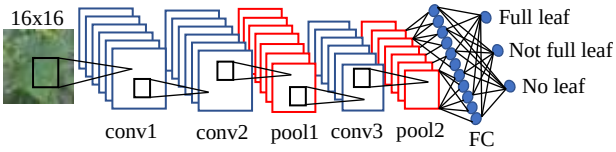

Fig. 3: The architecture of our CNNdetect for the leaf detection part. It accepts a single cucumber leaf image of 16x16 and generates three outputs. This CNNdetect is used to classify the "full leaf" among all extracted boxes from an exhaustive search on the wide-angle image

2.1.1. Training Dataset for CNNdetect

To train our CNNdetect, we built a training dataset including "full leaf", "not full leaf", and "no leaf" classes from our original images and the ImageNet dataset [7]. For the "full leaf" class, we performed data augmentation (For each image in the original images, performing a rotation on it and its cropped center with the incremental step of 90 degree; yielded eight times larger) on those images and thus formed 480,000 images. For "not full leaf", we divided all leaf images in the cropped and original images into quarters and formed 480,000 images. For the "no leaf" class, 480,000 images were collected randomly from the ImageNet dataset. In summary, training data for CNNdetect was constituted of 1.44 million image patches. For clarity, we refer to this dataset as the "detection dataset". Please see our previous paper [32] in detail.

2.1.2. Brightness Adjustment for Leaf Candidate Extraction

We found that CNNdetect provides better performance if it receives an input with an appropriate brightness range. We introduce a brightness adjustment on every single extracted image patch to adjust the brightness. In our experiment, we calculated the perceived brightness (PB) as in (1). Note that there are several ways to calculate this value.

$$\text{perceived brightness (PB)} = \sqrt{0.241\mathrm{R}_{\mathrm{RMS}} + 0.691\mathrm{G}_{\mathrm{RMS}} + 0.068\mathrm{B}_{\mathrm{RMS}}} \quad (1)$$

Here, $\mathrm{R}_{\mathrm{RMS}}$, $\mathrm{G}_{\mathrm{RMS}}$, $\mathrm{B}_{\mathrm{RMS}}$ are the root-mean-square of each color band (Red, Green and Blue) of image, respectively.
Then, we adjusted the pixel value of that image by multiplying the factor of $\alpha = \frac{\xi}{\mathrm{PB}}$ pixel by pixel for all color channels. Here, ξ is the predefined hyper parameter and we set $\xi = 130$: the average perceived brightness that calculated over the entire detection dataset by (1). Notice that the factor $\alpha < 1.0$ will produce a darker image, while $\alpha > 1.0$ will produce a brighter one. If the adjusted value exceeds the acceptable range of the color channel (255), we set it to 255.

2.2. The Leaf Diagnosis Part

In this part, all the extracted leaf areas identified as "full leaf" with CNNdetect, were classified as "healthy" or "diseased" with our classifier "CNNdiag". CNNdiag was built by transfer learning from the pre-trained VGGNet (VGG-16) [13].

2.2.1. Training Dataset for CNNdiag

We built a "diagnosis dataset" for training CNNdiag. It consists of a total of 30,758 images from the original 60,000 leaf images. Since the number of healthy leaf images was the largest, 15,379, we randomly selected the same number for the diseased case and formed the diagnosis dataset from those components. The diagnosis dataset also was augmented by 24 times to provide efficient training for CNNdiag. For more detail, we first augmented an image eight times larger by cropping its center and rotating it as in the detection dataset creation. Then, the gamma correction was applied to change the brightness of those images with γ = 1.5 and 0.67 respectively to provide the different brightness conditions. Finally, the image was resized to 224x224 pixels to fit CNNdiag's input. Following this procedure, the diagnosis dataset had a total of 30,758x24 = 738,192 images.

2.2.2. The Architecture of CNNdiag

Fig. 4 shows the overview of CNNdiag. This network is basically composed of a VGG-16 network [13]. CNNdiag accepts a color image with a size of 224x224 pixels and consists of several convolution blocks (each block has two or three convolution layers), pooling layers (after each convolution block) and two final fully-connected layers. In this work, we froze the first three convolution blocks, namely Conv_1, Conv_2, Conv_3 and trained other layers after Conv_3. We used batch normalization [33] for two convolution blocks, Conv_4 and Conv_5. The last two fully-connected layers were modified to have 2048 and 2 units respectively (FC-2048 and FC-2). The dropout technique [34] with a ratio of 50% was applied for the FC-4096 and FC-2048 layer. Finally, the softmax function was used as the output layer.

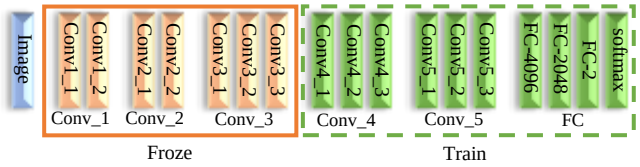

Fig. 4: The architecture of our proposed CNNdiag accepts a color image of 224x224. We froze the weights in the first three convolution blocks (surrounded by orange box) and trained Conv_4, Conv_5 and FC blocks (surrounded by dotted green box). The final layer is the softmax layer with two units as output

3. Results and Analysis

3.1. Training of End-To-End System

The detection dataset was divided into 60% training set (864,000 images) and 40% testing set (576,000 images). CNNdetect was trained using mini-batch gradient descent with batch-size of 500 and Adam optimizer [35]. After 20 training epochs, the accuracy of CNNdetect on the training dataset was 97.9% whereas it was 97.0% for the testing dataset.

For CNNdiag, we divided the diagnosis dataset with a high ratio of training set to testing set to train the CNNdiag. Namely, 78% was allocated to the training set (576,000 images), and 22% was allocated to the testing set (162,192 images). Also, CNNdiag was trained using mini-batch gradient descent with a batch size of 280 and Momentum optimizer (momentum was set to 0.9) [36]. After 30 training epochs, our model attained an accuracy of 98.0% on the training dataset and 97.4% on the testing dataset.

3.2. Experimental Setting

3.2.1. Target Wide-Angle Images and the Ground-Truth Definition

To evaluate our end-to-end system, 500 wide-angle cucumber images were taken in farms using various digital cameras. The size of these images is at least 2048x1536 pixels. Of which 300 images include all healthy leaves (visually examined by experts), while the remaining 200 include many infected disease leaves and several healthy leaves within each image. We refer these dataset as "healthy dataset" and "disease dataset" respectively. Since the healthy dataset is composed of healthy leaves only, the total of 6,860 full leaves were annotated as healthy by non-biologists. On the other hand, we asked experts to annotate the disease dataset. As it turned out, a total of 6,741 full leaves included 2,656 healthy and 4,085 diseased leaves. Fig. 5 shows an example of ground-truth on wide-angle image from healthy and disease dataset. The red and white boxes indicate the disease and healthy leaves respectively.

Fig. 5: Example of wide-angle ground-truth images. Images from the healthy dataset (left) were labeled by non-biologists; images from the disease dataset (right) were labeled by biological experts. The red boxes indicate disease leaves, while white boxes indicate healthy leaves

3.2.2. Evaluation Criteria

We used the F1-score criteria in both of the leaf detection and diagnosis parts. Given the result bounding boxes and ground-truth boxes on each image, the F1-score is calculated by the following equation:

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

where:

$$\text{precision} = \frac{\text{Number of correctly detected boxes}}{\text{Number of detected boxes}}$$

$$\text{recall} = \frac{\text{Number of correctly detected boxes}}{\text{Number of ground-truth boxes}}$$

A detected box is considered as a correct where the intersection over union (IoU) of that box and the corresponding ground-truth box is equal or larger than 0.5.

3.3. Experiment Results

3.3.1. Results of Leaf Detection

Table 1 and Fig. 6 show the leaf detection performance and examples with and without our proposed brightness adjustment. Thanks to the brightness adjustment, our new leaf detection schema achieved an average of 73.9% in F1-score on 500 wide-angle images with the processing speed of 1.5 seconds/image.

Table 1: Comparison of leaf detection performance with and without proposed brightness adjustment

	Precision	Recall	F1-Score
w/o brightness adjustment	63.0%	75.1%	68.4%
with brightness adjustment	**70.9%**	**77.2%**	**73.9%**

without Brightness Adjustment | with **Brightness Adjustment**

Fig. 6: The improvement of leaf detection in our system with brightness adjustment (right) compared to our previous result without brightness adjustment (left)

3.3.2. Results of Overall Performance

In our experiments, diseased leaves are only in the disease dataset and healthy leaves constitute a minority of that dataset. In the healthy dataset, all the images are healthy. Therefore, in our evaluation, overall performance (detection and diagnosis) for healthy and diseased leaves were performed on the healthy and the disease datasets, separately. Table 2 shows the evaluation performance of our end-to-end system. The running time of the whole system is around 3 seconds/input image.

Table 2: Performance result of leaf diagnosis part on healthy and disease dataset

	Healthy Dataset	Disease Dataset	Overall
precision (%)	70.0	61.5	**65.8**
recall (%)	80.3	55.9	**68.1**

Example on final results of our system can be found in Fig. 7. The left column represents the output of our system, while the right column is the ground-truth images. All the red and white boxes indicate the healthy and disease leaves respectively.

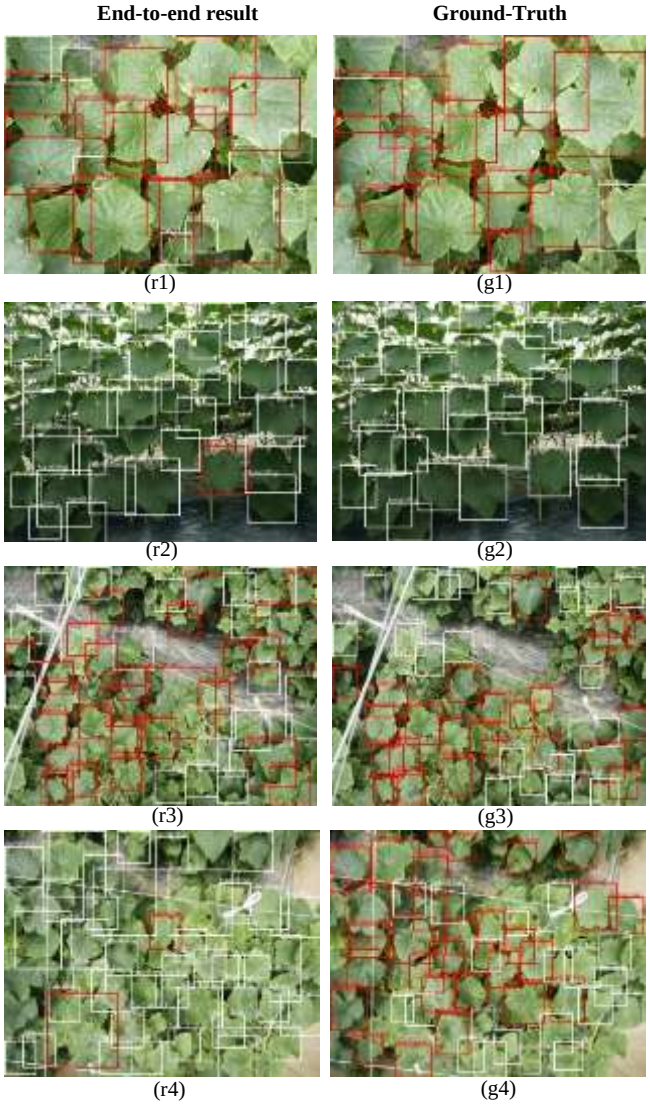

Fig. 7: The result of our end-to-end system on some wide-angle images (left) compared to the ground-truth images (right). The red and white boxes indicate healthy and diseased leaves respectively

4. Discussion

To the best of our knowledge, this paper is the first investigation on an end-to-end practical plant disease diagnosis system from wide-angle images. The simple but efficient proposed brightness adjustment provides a performance improvement. From a practical perspective, it is not necessary to localize all full leaves in images, but we should keep a certain level of precision to avoid passing wrong areas (i.e. the areas do not contain full leaf) to the diagnosis system. Considering the difficulty of this setting as mentioned earlier, we think our leaf detection performance (F1-score = 73.9%) is sufficiently reasonable in practical conditions and can be combined with the diagnosis system. The end-to-end diagnosis performance, i.e. detection and diagnosis, for healthy leaves (70.0% of precision and 80.3% of recall) is considered consistent with the prior leaf detection result. But, it decreases to 61.5% and 55.9% respectively for diseased leaves. This is caused by many difficulty factors in the practical conditions.

Considering the condition of wide-angle images on the disease dataset as in Fig. 7, the third and fourth result show the difficulties of diagnosing them. There are various leaf sizes and heavy overlapping (r3, r4) as well as low-quality images (blur, poor camera focus) (r4). They are more complicated compared to the top two results (r1, r2) where all the leaf sizes are nearly uniform, and overlap is not extensive. Therefore, these factors directly affect the performance of leaf detection part and lead to deterioration of the quality of the leaf diagnosis part.

For further investigation on this issue, we did an additional experiment by classifying only correctly detected leaves. In this setting, our CNNdiag achieved an accuracy of 75.0% in classifying diseased leaves on the disease dataset. This performance is still lower compared with current automated systems utilizing single leaf images in the literature, but there is a further gap between this result and the end-to-end result. We think the primary reason is the small intra-observer agreement on assigning the gold standard by experts. More concretely, a significant number of labeled leaves in the gold standard have inconsistency among images. This indicates making appropriate ground-truth using only the pictures that were already taken from wide-angle images is quite difficult even with specialists. In addition, our CNNdiag was trained with the diagnosis dataset only (all the images in this dataset had confirmed diagnosis) and the ground-truth leaves from the wide-angle dataset were used for testing. Note that when we built the CNN classifier based on this dataset, we observed that the classification performance will reach around 80-90% for multiple disease classification tasks. This difference between training and testing data generates a significant number of mis detected leaves and decreases our performance. We believe our system still has room for improvement by addressing these problems, and we intend to investigate this in future work. Although those factors make the diagnosis difficult, we believe our results are promising. In practical situations, infected leaves often spread disease to nearby areas. Once suspicious leaf areas are detected, we can apply further investigation to those nearby locations.

5. Conclusion

This paper investigated the first end-to-end practical plant diagnosis system on cucumber wide-angle images. We achieved a promising diagnosis performance under practical conditions by combining simple but efficient leaf detection with a robust diagnosis system using transfer learning. We are continuing to develop our end-to-end system and expect our model will be applied to practical automated plant diseases diagnosis applications in the near future.

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] Qin F, Liu D, Sun B, Ruan L, Ma Z, Wang H. Identification of alfalfa leaf diseases using image recognition technology. *Plos One*, 11(12), 2016, 1-26.
[2] Cortes C, Vapnik V. Support-vector networks. *Machine Learning*, 20(3), 1995, 273-297.
[3] Hallau L, Neumann M, Klatt B, Kleinhenz B, Klein T, Kuhn C, Röhrig M, Bauckhage C, Kersting K, Mahlein A K, Steiner U. Automated identification of sugar beet diseases using smartphones. *Plant Pathology*, 67(2), 2018, 399-410.
[4] Mwebaze E, Owomugisha G. Machine learning for plant disease incidence and severity measurements from leaf images. *Proceedings of the 15th IEEE International Conference on on Machine Learning and Applications*, 2016, 158-163.
[5] Rublee E, Rabaud V, Konolige K, Bradski G. ORB: An efficient alternative to SIFT or SURF. *Proceedings of the IEEE international conference on Computer Vision*, 2011, 2564-2571.
[6] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. *Proceedings of the Advances in Neural Information Processing Systems*, 2012, 1097-1105.
[7] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg A C. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3), 2015, 211-252.
[8] Liu B, Zhang Y, He D, Li Y. Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry*, 10(1), 2017, 11-26.
[9] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, 1-9.
[10] Mohanty S P, Hughes D P, Salathé M. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7(1419), 2016, 1-10.
[11] Hughes D, Salathé M. An open access repository of images on plant health to enable the development of mobile disease diagnostics. 2015, *CoRR*, abs/1511.08060.
[12] Wang G, Sun Y, Wang J. Automatic image-based plant disease severity estimation using deep learning. *Computational Intelligence and Neuroscience*, 2017, 1-8.
[13] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014, *CoRR*, abs/1409.1556.
[14] Durmuş H, Güneş E O, Kırcı M. Disease detection on the leaves of the tomato plants by using deep learning. *Proceedings of the IEEE 6th International Conference on Agro-Geoinformatics*, 2017, 1-5.
[15] Iandola F N, Han S, Moskewicz M W, Ashraf K, Dally W J, Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. 2016, *CoRR*, abs/1602.07360.
[16] Atabay H A. Deep residual learning for tomato plant leaf disease identification. *Journal of Theoretical and Applied Information Technology*, 95(24), 2017, 6800-6808.
[17] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Vision and Pattern Recognition*, 2016, 770-778.
[18] Amara J, Bouaziz B, Algergawy A. A deep learning-based approach for banana leaf diseases classification. *Lecture Notes in Informatics*, 2017, 79-88.
[19] LeCun Y, Boser B, Denker J S, Henderson D, Howard R E, Hubbard W, Jackel L D. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4), 1989, 541-551.
[20] Fujita E, Kawasaki Y, Uga H, Kagiwada S, Iyatomi H. Basic investigation on a robust and practical plant diagnostic system. *Proceedings of the IEEE Machine Learning and Applications*, 2016, 989-992.
[21] Sladojevic S, Arsenovic M, Anderla A, Culibrk D, Stefanovic D. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*, 2016, 1-11.
[22] Ferentinos K P. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, 2018, 311-318.
[23] Ramcharan A, Baranowski K, McCloskey P, Ahamed B, Legg J, Hughes D. Deep learning for image-based cassava disease detection. *Frontiers in Plant Science*, 8, 2017, 1-7.
[24] Atole R R, Park D. A multiclass deep convolutional neural network classifier for detection of common rice plant anomalies. *International Journal of Advanced Computer Science and Applications*, 9(1), 2018, 67-70.
[25] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 2017, 1137-1149.
[26] Dai J, Li Y, He K, Sun J. R-FCN: Object detection via region-based fully convolutional networks. *Proceedings of the Advances in Neural Information Processing Systems*, 2016, 379-387.
[27] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C Y, Berg A C. SSD: Single shot multibox detector. *Proceedings of the European Conference on Computer Vision*, 2016, 21-37.
[28] Dai J, Qi H, Xiong Y, Li Y, Zhang G, Hu H, Wei Y. Deformable convolutional networks. 2017, *CoRR*, abs/1703.06211.
[29] Fuentes A, Yoon S, Kim S C, Park D S. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, 17(9), 2017, 1-21.
[30] Lu J, Hu J, Zhao G, Mei F, Zhang C. An in-field automatic wheat disease diagnosis system. *Computers and Electronics in Agriculture*, 142, 2017, 369-379.
[31] Xia C, Lee J M, Li Y, Song Y H, Chung B K, Chon T S. Plant leaf detection using modified active shape models. *Biosystems Engineering*, 116(1), 2013, 23-35.
[32] Cap H Q, Suwa K, Fujita E, Kagiwada S, Uga H, Iyatomi H. A deep learning approach for on-site plant leaf detection. *Proceedings of the 14th IEEE International Colloquium on Signal Processing and its Applications*, 2018, 120-124.
[33] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015, *CoRR*, abs/1502.03167.
[34] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 2014, 1929-1958.
[35] Kingma D P, Ba J. Adam: A method for stochastic optimization. *Proceedings of the International Conference on Learning Representations*, 2014, 1-15.
[36] Qian N. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1), 1999, 145-151.