Super-Resolution for Practical Automated Diseases Diagnosis System

Quan Huu Cap*, Hiroki Tani*, Hiroyuki Uga†, Satoshi Kagiwada‡ and Hitoshi Iyatomi*
Email: {huu.quan.cap.78, hiroki.tani.2j}@stu.hosei.ac.jp uga.hiroyuki@pref.saitama.lg.jp {kagiwada, iyatomi}@hosei.ac.jp
∗ Applied Informatics, Graduate School of Science and Engineering, Hosei University, Tokyo, Japan
† Saitama Agricultural Technology Research Center, Saitama, Japan
‡ Clinical Plant Science, Faculty of Bioscience and Applied Chemistry, Hosei University, Tokyo, Japan

***Abstract*— Automated plant diagnosis using images taken from a distance is often insufficient in resolution and degrades diagnostic accuracy since the important external characteristics of symptoms are lost. In this paper, we first propose an effective pre-processing method for improving the performance of automated plant disease diagnosis systems using super-resolution techniques. We investigate the efficiency of two different super-resolution methods by comparing the disease diagnostic performance with the practical original high-resolution, low-resolution, and super-resolved cucumber images. Our method generates super-resolved images that look very close to natural images with 4× upscaling factors and is capable of recovering the lost detailed symptoms, largely boosting the diagnostic performance. Our model improves the disease classification accuracy by 26.9% over the bicubic interpolation method of 65.6% and shows a small gap (3% lower) between the original result of 95.5%.**

Keywords*— *super-resolution, deep learning, automated plant disease diagnosis, cucumber plant disease

I. Introduction

Diagnosing plant diseases is usually conducted by visual examination through experts. Thus, it is often time-consuming and expensive tasks. Not small number of computer-based diagnostic methods have been proposed to effectively prevent plant diseases and reduce the loss of crop yield. Those researches rapidly increased with the innovation of deep learning techniques. Early studies using these techniques were to analyze one leaf and yield the diagnosis result [1–5]. Recently, some of more sophisticated methodologies detect multiple targets of plant to be measured [6] or diagnosed [7, 8] simultaneously from relatively wide-range images (i.e. distance from camera to target is up to about 2m). In the former, they counted and measured plant stalk for supporting robot harvesting. In the latter, they investigated on-site tomato leaf images [7] and wheat images [8] taken from slightly wide angle, respectively. Despite those systems achieving excellent diagnostic performance, they still have a limitation that their target images are considerably narrower range than fixed-point observation camera images expected to be in practical applications. We have experienced that the accuracy is not sufficient when just applying these methods to wide-angle images. The wide-angle images on the actual agricultural site contain quite a lot of objects to be detected, they are basically the same kind and overlap each other. Thus, the system needs to make a diagnosis with these slight differences.

Also, key techniques used in the abovementioned systems were originally designed for general object recognition; they implicitly assumed that the physical appearance of the recognition target is somewhat different among categories. This assumption, unfortunately, does not hold onto practical plant diagnosis task with wide-range images. We believe one of the main reasons for this issue is the lack of resolution on the to-be-diagnosed targets in wide-angle images. The images taken from a distance camera often include smaller and lower resolution objects. Thus, it could decrease the performance of diagnosis systems since we should localize the diagnosis targets beforehand. The same issue is also reported in a study of the end-to-end disease diagnosis system for wide-angle cucumber images [9]. They admitted that the small leaf size and low-quality input images (low-resolution, blur, poor camera focus, etc.) could significantly reduce their disease diagnosis performance. We should note that the problem can be reduced by using a high-resolution camera device to capture the image, but it is usually expensive to deploy in the agriculture sector.

We think one solution for this is recovering the high-frequency component of images by applying the super-resolution (SR) methods. The SR methods had required multiple images to attain certain level of performance [10], but with the development of modeling power of deep learning techniques as typified by convolutional neural networks (CNN) in recent years, SR methods based on only single image, so-called single image super-resolution (SISR) have been proposed and shown excellence performance [11, 12, 13]. The pioneer work on SR was the super-resolution convolutional neural network (SRCNN) [11]. The SRCNN directly learned the mapping function between low and high-resolution images, providing an end-to-end training manner. They achieved significant improvement over conventional SR methods. The SRGAN [12] was then proposed as the first SR method using the advantage of the generative adversarial network (GAN) [14]. Furthermore, the enhanced SRGAN (ESRGAN) [13] including the residual-in-residual dense block (RRDB) and the applied relativistic average GAN (RaGAN) [15] as the key components was proposed and outperformed SRGAN in term of perceptual quality.

Although the SR techniques have been widely used in many fields, their applications to the agricultural sector have not been seen much so far [16, 17]. In [16], they proposed an adaptive based image SR method to enhance resolution of diseases leaf

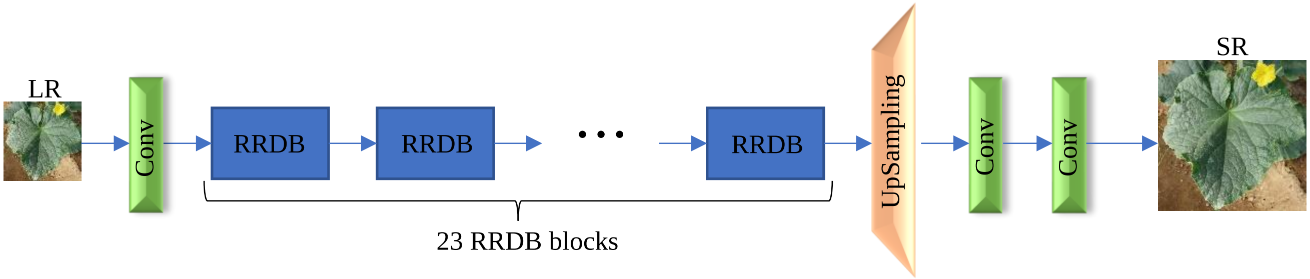

Fig.1. The generator G consists of 23 RRDB blocks followed by the up-sampling and convolutional layers to generate the SR images.

images. They claimed this could provide pathologists a better visual assessment on infected leaf, but so far, they have not evaluated the effects of their SR method for automated plant diagnosis. The use of SRCNN in the automated plant diagnosis has firstly seen in [17]. Authors compared the disease classification performance of tomato on the interpolated low resolution, the estimated SR, and the original high-resolution tomato leaf images. They demonstrated SR using SRCNN boosted the disease detection accuracy by 20%, reaching 90%. Although they showed a promising result, we cannot say from only their result that this diagnostic performance could be realized, or SR actually contributed to improving diagnostic accuracy in the practical situation. This is because the result was based on the in-lab environment since they used the PlantVillage [18] dataset. Each leaf image in that dataset is in an ideal situation; they are manually cropped and placed on a uniform background. In practical, on-site leaf images appear more complex with different symptoms, backgrounds and are affected with various photographic conditions. It is already known that the diagnosis system trained with these kinds of images showed extremely high diagnostic accuracy, but it turned into devastatingly when applied to real on-site images [2].

To the best of our knowledge, there is no literature to apply the SR methods on plant disease diagnosis in real situations. In this paper, we propose an effective pre-processing method for improving the performance of automated plant disease diagnosis systems for collective diagnosis from a wide range using super-resolution techniques.

II. Method

We introduced two SR models, M_{pix} and M_{feat}, and investigated how they improve the practical plant diagnostic accuracy. The M_{pix} is similar to SRCNN and its loss function L_{pix} is to minimize the differences between produced SR and HR images in the pixel space. While the M_{feat} is based on ESRGAN and its loss function L_{feat} is to minimize the differences between those images in the feature space instead of pixel space [19]. According to our preliminary experiments, the M_{pix} could recover the important disease features and reduced unexpected noises since it produces smoothed SR images. While the M_{feat} could recover more high-frequency details and generated the perceptually more pleasing results. However, in very few cases, M_{feat} yielded some tiny artifacts on generated leaf surface. We expect both methods could improve the diagnosis performance on plant disease images and would like to determine which method is better suited for our task.

For validation purposes, we use the multiple diseases cucumber dataset in [20] because of its practicality. Their images were taken at the actual site and consisting a wide variety of backgrounds and lighting conditions. In addition, they were planted in strictly controlled environment and therefore all of these images have reliable disease label for gold standard. We develop disease classifiers CNN_{Diag} with this dataset and compare their diagnostic performance among different resolution images, i.e. the down-sampled and interpolated low-resolution (LR) images, generated SR images, and the original high-resolution (HR) leaf images. All the interpolated and SR images will be generated with the up-scaling factor of four from LR images (i.e. the ratio of those image size is 1:16).

A. Network Architectures

The M_{pix} is composed of one “generator” CNN model for generating SR image. On the other hand, the M_{feat} is composed of two CNN models; the generator G and the discriminator D as refer in the ESRGAN. Note here that the generator G generates SR images and the discriminator D distinguishes generated SR images from HR images. We call the generator CNN model of M_{pix} as G_{pix} and the model of M_{feat} as G_{feat}, respectively. Both G_{pix} and G_{feat} have the same network architecture as the generator G of ESRGAN reported in the original literature [13]. In the M_{feat}, two networks G_{feat} and D are trained together to solve the adversarial min-max problem.

1) The Generator

The generator G is composed of a lot of residual-in-residual dense blocks (RRDB) [13]. An RRDB consists of three dense blocks [21]; each dense block has five densely connected convolutional layers. The three dense blocks are connected to the main path of the network in residual manner [22]. In this work, we use the generator G consisting of 23 RRDB blocks, i.e. a total of 115 convolutional layers. Fig. 1 describes the generator G architecture used in our experiments.

2) The Discriminator

Our discriminator D is designed under the architecture guidelines for stable deep convolutional GANs in [23]. The difference from the discriminator in ESRGAN is its input size. We designed our discriminator deeper to accept larger input size of 192×192 compared to the original 128×128. We found that the larger input size could help the network learn useful

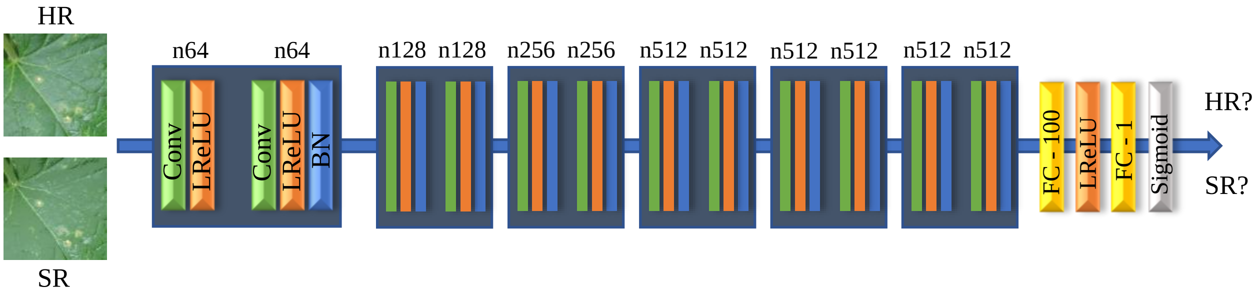

Fig.2. The discriminator D consists of six $conv\_block$ with corresponding number of feature maps (n).

information. The architecture of our discriminator D is illustrated in Fig. 2. Our discriminator consists of six convolutional blocks ($conv\_block$) followed by the two fully-connected (FC) layers and has no max-pooling layers. We define our $conv\_block$ as a block of two convolutional layers. Each convolutional layer has its kernel size (k), number of feature maps (n) and stride (s). At each $conv\_block$, we use $k_1 = 3, s_1 = 1$ for the first and $k_2 = 4, s_2 = 2$ for the second layer. The number of n is different on each $conv\_block$. The first fully-connected layer has 100 units while the last layer contains a single unit and a sigmoid activation function. We use the leaky rectified linear function (LReLU) [24] with $\alpha = 0.2$ as the activation function for all layers except for the last layer. Batch normalization (BN) [25] is applied from the second to the last convolutional layer.

3) Loss Functions

The objective of training M_{pix} is to minimize the pixel-wise difference between generated SR images I_{SR} and HR images I_{HR}. We choose the loss function of G_{pix} to be minimized as:

$$L_{pix} = |I_{HR} - I_{SR}|_1. \quad (1)$$

For the training of M_{feat}, the adversarial training between G_{feat} the discriminator D is applied. The output of our discriminator D is expressed as:

$$D(I_{HR}, I_{SR}) = \text{sig}\big(C(I_{HR}) - \mathbb{E}_{I_{SR}}[C(I_{SR})]\big) \quad (2)$$

,where $C(I)$ is the output from the $FC - 1$ layer (before the sigmoid layer) of the discriminator D (see Fig. 2); $\mathbb{E}_I[\cdot]$ is the average value of all images in a mini-batch I. Note that the I_{HR} and I_{SR} in Eq. (2) can be substituted for each other.

Here, we use the same loss functions for G_{feat} and D as used in ESRGAN. The adversarial loss for the discriminator L_D is defined as

$$L_D = -\mathbb{E}_{I_{HR}}\big[\log\big(D(I_{HR}, I_{SR})\big)\big] - \mathbb{E}_{I_{SR}}\big[\log\big(1 - D(I_{SR}, I_{HR})\big)\big]. \quad (3)$$

Therefore, the adversarial loss for generator $L_{G_{feat}}$ is in a symmetrical form:

$$L_{G_{feat}} = -\mathbb{E}_{I_{HR}}\big[\log\big(1 - D(I_{HR}, I_{SR})\big)\big] - \mathbb{E}_{I_{SR}}\big[\log\big(D(I_{SR}, I_{HR})\big)\big]. \quad (4)$$

Finally, we form the total loss L_{feat} for the generator G_{feat} as:

$$L_{feat} = L_{percep} + \lambda L_{G_{feat}} + \eta |I_{HR} - I_{SR}|_1 \quad (5)$$

,where λ, η are the coefficients to balance different loss terms. L_{percep} is the perceptual loss in the features space of HR and generated SR images represented in the VGG19 [26] model pretrained with ImageNet dataset [27]. The L_{percep} is defined as:

$$L_{percep} = \big|VGG19_{5\_3}(I_{HR}) - VGG19_{5\_3}(I_{SR})\big|_1. \quad (6)$$

Here, $VGG19_{5\_3}$ is the convolution layer before the last max-pooling layer of the VGG-19.

B. The Diseases Cucumber Dataset

All the cucumber leaf images were taken from Saitama Agricultural Technology Research Center, Japan. Each original HR image contains a single cucumber leaf roughly in the center and surrounded with various backgrounds. There are seven types of viral (CCYV, CMV, KGMMV, MYSV, PRSV, ZYMV and WMV) and four fungal (Brown spot, Downy mildew, Gray mold, and Powdery mildew) diseases (11 disease types). For multiple diseases, each class normally has two or three diseases and a total 13 combinations of multiple infections were labeled from above 11 diseases. The dataset has a total of 48,331 cucumber leaf images consisting of 38,821 single, 1,814 multiple infections, and 7,676 healthy leaves. From that dataset, we divided the training and testing set into two separate sets. Specifically, the training set has 36,253 images (roughly 75% of dataset) and the testing set contains 12,078 images (roughly 25% of dataset). We treat this task as 25-class classification task as well as in [20].

C. The CNN Architecture for Diseases Diagnosis

In order to compare the diagnostic performance between the generated SR images and the original HR images, we train our CNN$_{\text{Diag}}$ for diagnosing diseases on the training set (containing 36,253 images). The CNN$_{\text{Diag}}$ is the same as the proposed in [20] and consists of eight convolutional layers followed by two fully-connected layers. We applied ReLU as the activation function for all layers except for the last fully-connected layer and batch normalization for all convolutional layers. Two fully connected layers have 2,048 nodes each and with a dropout [28] ratio of 0.5. To classify multiple infections, a sigmoid function with tunable thresholds on each node is used in the last layer of network. The thresholds for each output value were set beforehand by applying grid-search on F1-score result domain in order to deal with the imbalance data problem.

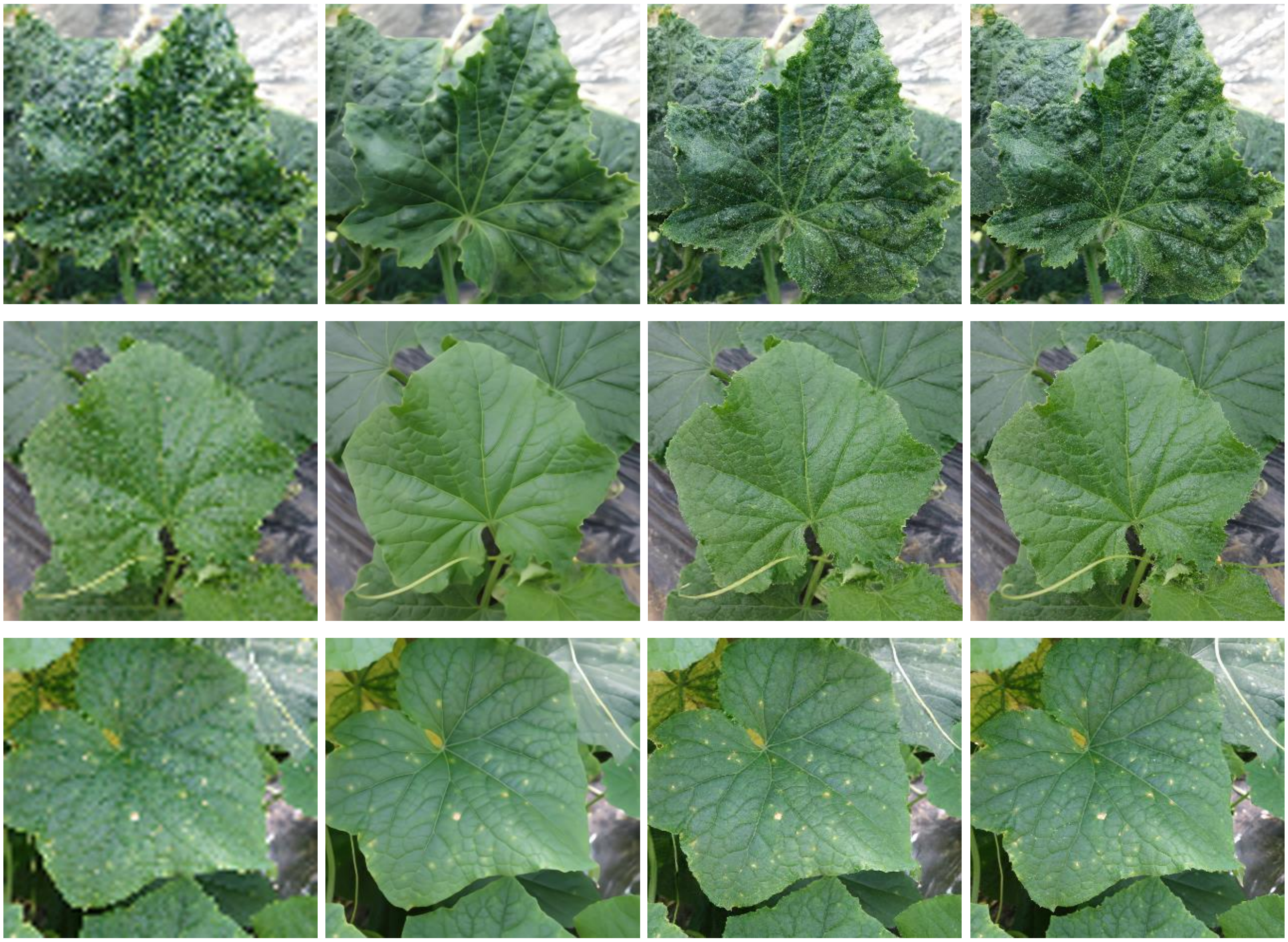

Fig.3. The comparison between the generated SR results and the original images. Super-resolved images from G_{feat} are almost indistinguishable from original ones and have more details compared to other SR methods (bicubic and G_{pix}).

III. Experiments and Results

A. Training the CNN$_{Diag}$

We applied the same data augmentation method as used in [20]. In this way, the augmented training dataset now is 36 times larger than the original. We trained our CNN$_{\text{Diag}}$ model using Adam optimization [29] with mini-batch size of 128. After 1000 epochs, we achieved the accuracy on classifying cucumber diseases of 95.5%.

B. Training the SR models

We trained our M_{pix} and M_{feat} using the same training images as used to train the CNN$_{\text{Diag}}$. During the training, the HR images were obtained by randomly cropping from the training images with a pre-defined size. The LR images are then created by down-sampling from HR images using bicubic interpolation. Both LR and HR images are augmented with random horizontally flip and random 90 degrees rotation on-the-fly.

First, we train our G_{pix} with the pixel-wise loss with Eq. (1). The HR images are randomly cropped with the size of 96×96 from training images. Based on our preliminary experiments, the G_{pix} trained with smaller HR crop size (96×96) produces better visual results. Mini-batch size is set to 64 and the training is finished after 1,000,000 iterations (roughly 1780 epochs).

Second, we train the G_{feat} with the initial weights from the pre-trained G_{pix}. Our G_{feat} is trained using the loss function in Eq. (3) with $\lambda = 5 \times 10^{-3}$ and $\eta = 10^{-2}$. We set the mini-batch size of 32 images. HR images with size of 196×196 are randomly cropped from training images. Different from the crop size of G_{pix}, the bigger HR size (196×196) used in G_{feat} could help to capture more semantic information. We completed adversarial training for G_{feat} after 400 epochs. Note that we used the Adam as the optimizer for training both G_{pix} and G_{feat}.

C. Results

To evaluate the results of our SR models, all 12,078 images from the testing set were resized to the size of 56x56 by bicubic interpolation beforehand.

TABLE I. THE QUANTITATIVE RESULTS OF DISEASE DIAGNOSIS PRODUCED BY BICUBIC (LR), G_{pix}, G_{feat}, AND ORIGINAL (HR) IMAGES

	Bicubic	G_{pix}	G_{feat}	**Original**
Accuracy (%)	65.6	71.8	**92.5**	95.50

We enlarge LR images with 4× upscaling factors by using the bicubic interpolation, and the trained models (G_{pix}, G_{feat}). The comparison between the generated results and the original image is shown in Fig. 3. Since LR images are tiny size (56x56), the generated images by bicubic interpolation have a low-quality, blurry and could not recover much details. The results from G_{pix} are over-smoothed but have much better quality than bicubic interpolated images. On the other hand, the G_{feat} produces more natural images with recovered high-frequency details. They are almost indistinguishable from original images.

Table 1 shows the accuracy of diseases diagnosis using the generated images by bicubic, G_{pix} , G_{feat} , and original HR images. These results indicate that the G_{feat} with perceptual loss performs the best among the other SR methods with 92.5% of average diagnosis accuracy on its generated images. Also, it shows a small gap between the original result (only 3% lower), outperforming the low results from bicubic and G_{pix} (65.6% and 71.8% respectively).

IV. DISCUSSION

We investigated the effectiveness of SR methods for improving the performance of automated plant disease diagnosis system on a practical cucumber images dataset. From the results in Fig. 3 and Table I, there is no surprise that the low-quality images generated by bicubic interpolation yield the lowest diagnosis performance (only 65.6%). As for the result of G_{pix}, even though they have a much better visual quality than bicubic interpolation, the diagnosis result did not increase significantly (only 6.2% higher). Since G_{pix} was trained with the pixel-wise loss, it will not be able to recover the high-frequency image components. In this case, recovering the disease symptoms which appear in a high-frequency detailed form is crucial for improving diagnosis systems. Thus, the SR methods that use pixel-wise loss are not suitable for practical in-field disease diagnosis.

On the other hand, our G_{feat} shows an outstanding diagnosis result with 92.5% mean accuracy (dramatically improved 20.7% from G_{pix}) and close to the original high-resolution result. This reinforces our inference that the practical disease symptoms usually appear in the high-frequency detailed form. Recovering the lost detailed disease symptoms could improve the diagnosis performance. Moreover, our preliminary experiment showed that within the first 100 training epochs, our G_{feat} model achieved nearly 89% of diagnostic performance. This indicates that the perceptual SR method is an effective tool for practical disease diagnosis from LR images.

In our experiments, our G_{feat} yielded some tiny artifacts on generated leaf surface for very few cases only on MYSV disease images. Although the effect of this on classification accuracy was limited, we are on the way to overcome this problem.

Although our model achieves an excellent result, we believe it is also somewhat dominated by the $\mathrm{CNN}_{\mathrm{Diag}}$ model. For improving the practicality of automated disease diagnosis system, we will continue developing our SR method to work with different magnification scales and apply it into the wide-angle images (e.g. images taken by surveillance cameras) in the future studies.

V. CONCLUSION

To the best of our knowledge, this paper is the first to propose an effective pre-processing method for automated plant disease diagnosis systems using SR methods. We achieved a promising diagnosis result under low-quality images conditions on practical multiple diseases dataset. From that result, we have confirmed that our SR method with perceptual loss is efficient and suitable for improving the practical disease diagnosis performance. We are continuing to develop our method and make it more robust in real cultivation conditions in the near future.

ACKNOWLEDGMENT

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] Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, “Basic study of automated diagnosis of viral plant diseases using convolutional neural networks,” International Symposium on Visual Computing, pp. 638–645, 2015.

[2] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” Frontiers in Plant Science, vol. 7 (1419), pp. 1–10, 2016.

[3] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahamed, J. Legg, D. Hughes, “Deep Learning for Image-Based Cassava Disease Detection,” Frontiers in Plant Science, vol. 8 (1852), pp. 1–7, 2017.

[4] K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” Computers and Electronics in Agriculture, vol. 145, pp. 311–318, 2018.

[5] E. Fujita, H. Uga, S. Kagiwada, and H. Iyatomi, “A Practical Plant Diagnosis System for Field Leaf Images and Feature Visualization,” International Journal of Engineering & Technology, vol. 7, no. 4.11, pp. 49–54, 2018.

[6] H. S. Baweja, T. Parhar, O. Mirbod, and S. Nuske, “StalkNet: A Deep Learning Pipeline for High-Throughput Measurement of Plant Stalk Count and Stalk Width,” Springer Proc. in Advanced Robotics, vol. 5, pp. 271–284, 2018.

[7] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, “A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition,” Sensors, vol. 17, no. 9, 2017.

[8] J. Lu, J. Hu, G. Zhao, F. Mei, and C. Zhang, “An in-field automatic wheat disease diagnosis system,” Computers and Electronics in Agriculture, vol. 142, pp. 369–379, 2017.

[9] Q. H. Cap, K. Suwa, E. Fujita, S. Kagiwada, H. Uga, and H. Iyatomi., “An End-To-End Practical Plant Disease Diagnosis System for Wide-Angle Cucumber Images,” International Journal of Engineering & Technology, vol. 7, no. 4.11, pp. 106–111, 2018.

[10] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, “Fast and Robust Multi-Frame Super-Resolution,” IEEE Trans. on Image Processing, vol. 13, no. 10, pp. 1327–1344, Oct. 2004.

[11] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 38, no. 2, pp. 295–307, 2016.

[12] C. Ledig, L. Theis, F. Husz´ar, J. Caballero, A. Cunningham, A. Acosta et al., “Photo-realistic single image super-resolution using a generative adversarial network,” IEEE Proc. on Computer Vision and Pattern Recognition, pp. 105–114, 2017.

[13] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong et al., “ESRGAN: Enhanced super-resolution generative adversarial networks,” CoRR, abs/ 1809.00219, 2018.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair et al., “Generative adversarial nets,” Advances in Neural Information Processing Systems, pp. 2672–2680, 2014.

[15] A. Jolicoeur-Martineau, “The relativistic discriminator: a key element missing from standard gan,” CoRR, abs/1807.00734, 2018.

[16] S. B. Kasturiwala and S. Aladhake, “Adaptive image superresolution for agrobased application,” Proc. of the 2015 International Conference on Industrial Instrumentation and Control, pp. 650–655, 2015.

[17] K. Yamamoto, T. Togami, and N. Yamaguchi, “Super-resolution of plant disease images for the acceleration of image-based phenotyping and vigor diagnosis in agriculture,” Sensors, vol. 17, no. 11, 2017.

[18] D. P. Hughes and M. Salathé, “An open access repository of images on plant health to enable the development of mobile disease diagnostics,” CoRR, abs/1511.08060, 2015.

[19] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super- resolution,” Springer Proc. on European Conference on Computer Vision, pp. 694–711, 2016.

[20] H. Tani, R. Kotani, S. Kagiwada, H. Uga, and H. Iyatomi, “Diagnosis of Multiple Cucumber Infections with Convolutional Neural Networks,” The 47th Annual IEEE AIPR 2018: Ubiquitous Imaging, 2018.

[21] G. Huang, Z. Liu, K. Q. Weinberger, and L. Van Der Maaten, “Densely connected convolutional networks,” CoRR, abs/1608.06993, 2016.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” IEEE Proc. on Conference on Computer Vision and Pattern Recognition, pp. 770–778, 2016.

[23] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” CoRR, abs/1511.06434, 2016.

[24] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” ICML Workshop on Deep Learning for Audio, Speech, and Language Processing, 2013.

[25] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” Proc. of the 32nd International Conference on Machine Learning, vol. 37, pp. 448–456, 2015.

[26] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” CoRR, abs/1409.1556, 2014.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” IEEE Proc. on Computer Vision and Pattern Recognition, pp. 248-255, 2009.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” The Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014.

[29] D. Kinga and J. B. Adam, “Adam: A method for stochastic optimization,” CoRR, abs/1412.6980, 2015.