Deep Learning Approach for Practical Plant Disease Diagnosis

HUU QUAN, Cap

(開始ページ / Start Page)
1
(終了ページ / End Page)
82
(発行年 / Year)
2021-09-15
(学位授与番号 / Degree Number)
32675甲第525号
(学位授与年月日 / Date of Granted)
2021-09-15
(学位名 / Degree Name)
博士(工学)
(学位授与機関 / Degree Grantor)
法政大学 (Hosei University)
(URL)
https://doi.org/10.15002/00024532

Doctoral Dissertation Reviewed by
Hosei University

Deep Learning Approach for Practical Plant Disease Diagnosis

by

Huu Quan Cap - 18R9802

Supervisor: Prof. Hitoshi Iyatomi

Graduate School of Science and Engineering
Hosei University, Tokyo, Japan
April 2021

Abstract

With the breakthrough of deep learning techniques, many excellent applications for the automated diagnosis of plant disease have been proposed. However, there are several open issues for developing practical plant disease diagnosis systems in real cultivation. *Firstly*, most conventional methodologies only accept narrow range images, typically one or quite a limited number of targets are in their inputs. Applying these models to wide-angle images in large farms would be very time-consuming, since many targets (e.g., leaves) need to be diagnosed. In this work, we propose a two-stage system which has independent leaf detection and leaf diagnosis stages for wide-angle disease diagnosis. We show that our proposal attains a promising disease diagnostic performance that is more than six times higher than end-to-end systems (state-of-the-art detection methods like Faster R-CNN or SSD) with F1-score of 33.4 – 38.9% compared to 4.4 - 6.2% on an unseen target dataset.

Secondly, the lack of image resolution (i.e., diagnosing from low-quality input images such as low-resolution, blur, poor camera focus, etc.) could significantly reduce the diagnostic performance in practice. Also, high-resolution data is very difficult to obtain and are not always available in practice. Deep learning-based techniques, and particularly generative adversarial networks (GANs), can be applied to generate high-quality super-resolution images, but these methods often produce unexpected artifacts that can lower the diagnostic performance. In this paper, we propose a novel artifact-suppression super-resolution method that is specifically designed for diagnosing leaf disease, called LASSR. Our LASSR can detect and suppress artifacts to a considerable extent. Thus, generating much more pleasing, high-quality images from low-resolution inputs. Experiments show that training with data generated by our proposal significantly boosts the performance on an unseen test dataset by over 21% compared with the baseline.

Thirdly, collecting and labeling training disease data for these diagnosis systems

requires solid biological knowledge and is very labor-intensive. Limited amount of disease training data leads to the fourth problem of model overfitting. The performance of disease diagnostic models are drastically decreased when used on test data sets from new environments. Meanwhile, we observe that healthy images are easier to collect. Based on this, we propose LeafGAN, a novel image-to-image translation system. LeafGAN generates countless diverse and high-quality diseased data via transformation from healthy images, as a data augmentation tool for improving the performance of plant disease diagnosis. Our model can transform only relevant areas from images with a variety of backgrounds, thus enriching the versatility of the training images. Experiments show that data augmentation with LeafGAN help to improve the generalization, boosting the diagnostic performance on unseen data by 7.4% from baseline.

In summary, we show that our approaches significantly improve the diagnostic performance under practical settings, confirming to be efficient and reliable methods for real cultivation scenarios.

Acknowledgements

My Ph.D. studies have been a wonderful and meaningful time in my life. This thesis would not have been completed without the help of people for whom I am very grateful.

First and foremost, I would like to express my sincere gratitude to my supervisor, professor Hitoshi Iyatomi for his generosity in sharing his thoughts, motivation, enthusiasm, and knowledge with me. His office door was always open whenever I ran into trouble on my research and he is always the right person who steered me in the right direction. Outside of research time, I am grateful to him for always telling me about many great Japanese cultures. Every time walking with him is a new experience for me. Iyatomi-sensei has always been the one who inspires me and whom I have always admired. In addition, the trips to international conferences in many parts of the world with him, his son, and friends in the laboratory were amazing memories.

I was very lucky to be a member of Iyatomi's lab. I would like to thank Shimada, Erika, Panama, Pink, and Joe for being my tutors. I want to give a special thanks to Panama (Kasumasa Suwa), Pink (Shunsuke Kitada), and Joe (Odagiri Kaito), who were really kind friends since I came here. They literally helped and assisted me countless times during my student life in Japan. In addition, I cherished the time spent studying, working, and participating in activities with Chasen, Palloc, Yawara, Sai, Yuma, Daif, Pei, Cocoa, Tetsu, and all other members of Iyatomi's lab. Thanks to them, I have learned a lot of Japanese and research skills. I really felt like this lab was my second home. These are warm and wonderful memories for me that I will never forget.

As an IIST student, I wish to have a big thanks to professor Kazuo Yana, who is the founder of the IIST program and also the person who inspired me to come to Japan. I still vividly remember his enthusiasm and kindness the first days I arrived in Koganei. His great supports in my academic life at Hosei University is invaluable. I wish him a meaningful retirement. I want to thank professor Jinjia Zhou, who taught me many

useful soft skills in research. The knowledge from her is very rewarding on my career path. Also, many thanks to all IIST and foreign students for being with me, especially Chi, Man, and Peter. Life outside Hosei would have been tasteless without them.

One very important thing, I must express my very profound gratitude to my beloved parents, ba Quý and mẹ Thản, to my lovely little sister Diệu, and to my amazing girlfriend Nhi for providing me with unfailing support and continuous encouragement throughout my years of study despite the geographical distance. They are wonderful in every way. Without them, I could not be myself today!

Lastly, I am very grateful and would like to have a special thanks to the Watanuki International Scholarship Foundation for giving me scholarships for the last three years. Thanks to the Japan Student Services Organization (JASSO) and Hosei University for supporting my finances during my study as well. This work was also supported by the Ministry of Agriculture, Forestry and Fisheries (MAFF) Japan Commissioned project study on "Development of pest diagnosis technology using AI" (JP17935051), and 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).

Huu Quan Cap

List of Publications

Journal papers

[1] **Quan Huu Cap**, Hiroki Tani, Satoshi Kagiwada, Hiroyuki Uga, Hitoshi Iyatomi, "LASSR: Effective Super-Resolution Method for Plant Disease Diagnosis," Computers and Electronics in Agriculture, vol. 187, pp. 106271, June 2021.

[2] **Quan Huu Cap**, Hiroyuki Uga, Satoshi Kagiwada, and Hitoshi Iyatomi, "LeafGAN: An Effective Data Augmentation Method for Practical Plant Disease Diagnosis," IEEE Transactions on Automation Science and Engineering, pp. 1-10, 2020. (early access)

[3] **Quan Huu Cap**, Katsumasa Suwa, Erika Fujita, Satoshi Kagiwada, Hiroyuki Uga, and Hitoshi 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, October 2018.

Conference papers

[1] **Quan Huu Cap**, Hitoshi Iyatomi, and Atsushi Fukuda, "MIINet: An Image Quality Improvement Framework for Supporting Medical Diagnosis," in Proceedings of the International Conference on Pattern Recognition Workshops, January 2021, pp. 254–265.

[2] **Quan Huu Cap**, Hiroki Tani, Hiroyuki Uga, Satoshi Kagiwada, and Hitoshi Iyatomi, "Super-resolution for practical automated plant disease diagnosis system," in Proceedings of the Annual Conference on Information Sciences and Systems, March 2019, pp. 1–6.

[3] **Quan Huu Cap**, Katsumasa Suwa, Erika Fujita, Satoshi Kagiwada, Hiroyuki Uga,

and Hitoshi Iyatomi, "A deep learning approach for on-site plant leaf detection," in Proceedings of the 14th IEEE International Colloquium on Signal Processing & Its Applications, March 2018, pp. 118–122.

[4] Kasumi Obi, **Quan Huu Cap**, Noriko Umegaki-Arao, Masaru Tanaka, and Hitoshi Iyatomi, "Bulk Production Augmentation Towards Explainable Melanoma Diagnosis," in Proceedings of the IEEE EMBS Conference on Biomedical Engineering and Sciences, March 2021, pp. 454–459. (*Student best paper award*)

[5] Satoi Kanno, Shunta Nagasawa, **Quan Huu Cap**, Shogo Shibuya, Hiroyuki Uga, Satoshi Kagiwada, and Hitoshi Iyatomi, "PPIG: Productive and Pathogenic Image Generation for Plant Disease Diagnosis," in Proceedings of the IEEE EMBS Conference on Biomedical Engineering and Sciences, March 2021, pp. 554–559.

[6] Hideaki Okamoto, **Quan Huu Cap**, Takakiyo Nomura, Hitoshi Iyatomi, and Jun Hashimoto, "Stochastic Gastric Image Augmentation for Cancer Detection from X-ray Images," in Proceedings of the IEEE International Conference on Big Data Workshops, December 2019, pp. 4858–4863.

[7] Katsumasa Suwa, **Quan Huu Cap**, Ryunosuke Kotani, Hiroyuki Uga, Satoshi Kagiwada, and Hitoshi Iyatomi, "A comparable study: Intrinsic difficulties of practical plant diagnosis from wide-angle images," in Proceedings of the IEEE International Conference on Big Data Workshops, December 2019, pp. 5195–5201.

[8] Takumi Saikawa, **Quan Huu Cap**, Satoshi Kagiwada, Hiroyuki Uga, and Hitoshi Iyatomi, "AOP: An anti-overfitting pretreatment for practical image-based plant diagnosis," in Proceedings of the IEEE International Conference on Big Data Workshops, December 2019, pp. 5177–5182.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Loss of crop yield due to plant diseases is one of the most serious and longstanding problems in the development of agriculture worldwide. Early detection and appropriate treatment are crucial steps to increase crop productivity. This is also essential in ensuring global food security and the sustainability of agroecosystems [1, 2]. 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 a 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.

Qin et al. [3] used conventional image segmentation techniques and a support vector machine (SVM) [4] to classify four types of alfalfa diseases in leaves. The SVM classifier achieved an accuracy of 94.7% on images in a laboratory environment. Hallau et al. [5] 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 an SVM classifier. Their system achieved 82.0% classification accuracy. Mwebaze et al. [6] built a smartphone-based system to diagnose five classes (four types of disease and a healthy diagnosis) in cassava. The system used three classifiers (Linear SVM, K-Nearest neighbor (KNN), and extremely randomized trees (ERT)) with a hand-crafted feature called Color and Oriented FAST and Rotated BRIEF (ORB) [7] and achieved over 99% accuracy. Es-Saady et al. [8] designed a system based on a serial combination of two SVM classifiers to recognize six classes of disease. Image features such as colour, shape, texture were fed to the two SVM classifiers to make decisions. Their system

attained an overall recognition rate of 87.8%.

These methods successfully established preferable performance for their own target task. However, since they were 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. For example, the selection of the pre-processing methods, the hand-crafted features or classification algorithms is a tedious process and it is difficult to find the combinations that yield the best results. Moreover, such methods often fail when diagnosing on complex real-world plant images [9].

In recent years, convolutional neural networks (CNNs) have demonstrated tremendous success in object recognition and image classification tasks. Since the breakthrough of AlexNet [10] in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC 2012) [11], many automatic plant diagnosis systems have used CNNs for identifying plant diseases. Liu et al. [12] designed their own CNN model to classify four types of apple diseases by combining AlexNet with Inception modules [13]. Under a controlled environment, their model had considerably smaller parameters than AlexNet while they achieved an overall accuracy of 97.6%. Mohanty et al. [14] applied CNN-based transfer learning. More specifically, they used pre-trained AlexNet and GoogLeNet to to identify 14 crop species and 26 diseases on the PlantVillage dataset [15] and attained a classification accuracy of 99.3%. With open access and large number of disease images, the PlantVillage dataset has also been widely used for plant disease classification systems. Wang et al. [16] applied transfer learning with VGG-Net [17] and classified different stages of the apple black rot disease on the same PlantVillage dataset, showing an accuracy of 90.4%. Durmuş et al. [18] investigated transfer learning on AlexNet and SqueezeNet [19] to classify ten classes (nine diseases, one healthy) of tomato leaves from the PlantVillage dataset. They showed a com-

petitive accuracy of 94.3% on SqueezeNet compared to AlexNet with 95.6%. Thus, the small size of SqueezeNet is well suited for mobile applications. Elhassouny and Smarandache [20] also built a mobile application for classifying those ten tomato leaf diseases using Mobilenet [21]. Their model attained 90.3% of recognition accuracy while running on smartphones. In another work in the same tomato dataset, Atabay [22] showed a test accuracy of 99.9% while classifying ten tomato diseases with deep residual learning [23]. Barbedo [24] designed a system to identify disease from individual lesions and spots images on 14 types of crops. Multiple GoogLeNet [13] models were fine-tuned and achieved overall accuracy of 82.0% among multiple crops. The advantages of the recent state-of-the-art EfficientNet model [25] was also exploited to classify all classes of the PlantVillage dataset [26]. Their EfficientNet model reached the highest classification accuracy of 99.97%. 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.

On the other hand, several other diagnosis systems have also proven the reliability of CNN-based methods under practical conditions. In pioneering work, Kawasaki et al. [27] trained a three-layer CNN to diagnose three classes of cucumber diseases (two classes of diseased and one of healthy) on images from a real farm, in which the target objects appear with complex backgrounds. Their model achieved an average accuracy of 94.9%. Similar studies on cucumber [28–30] have also conducted. Sladojevic et al. [31] designed a customized CNN model and got an overall accuracy of 96.3% while identifying 13 types of disease in five crops using images downloaded from the Internet. Ramcharan et al. [32] 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. DeChant et al. [33] proposed an automated system to identify northern leaf blight lesions on field-

acquired images of corn plants and achieved 96.7% accuracy on test set. Ferentinos [34] 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. On studies of diagnosing natural rice images, Lu et al. [35] trained their system to identify ten common rice diseases. Under the 10-fold cross-validation strategy, the proposed CNN model achieves an accuracy of 95.48%. Chen et al. [36] combined the VGGNet [13] and Inception [37] to form the INC-VGGN model for rice and maize disease classifications. Their proposal attained the average accuracy of 92.0% on five-class in-field rice diseases and 80.4% on four-class maize classification. Picon et al. [38] proposed several systems that incorporate contextual non-image metadata such as crop information and high-level feature extracted from a CNN for multi-crop disease classification. Their model classified total of 23 disease classes from five crops namely winter wheat, rice, corn, rapeseed, and winter barley. They obtained the best performance result of balanced accuracy of 98.0% on their model.

In the meantime, CNNs also demonstrated brilliant performance in the simultaneous processing of object detection and localization. Many state-of-the-art object detections methods have been proposed [46–56]. Inspired by that work, some interesting diagnosis systems are not only detecting the diseases but also localizing their involved areas. Fuentes et al. [39] used three CNN-based systems (i.e., Faster R-CNN [48], R-FCN [49] and SSD [51]) which performed object localization and diagnosis processes simultaneously. Their system achieved 86.0% mean average precision (mAP) on annotated tomato leaf images. Lu et al. [40] 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. Wang et al. [41] developed two different models, Faster R-CNN [48] and Mask R-CNN [57] for identifying the types of tomato

Table 1.1: Summary of recent works on leaf disease recognition

Article	Crop	# of class	Dataset condition	Type	Method	Performance (%)
Qin et al. [3]	Alfalfa	4	In-lab	Classification	SVM	94.7
Hallau et al. [5]	Sugar beet	4	In-field (background removed)	Classification	SVM	82.0
Mwebaze et al. [6]	Cassava	5	In-field	Classification	SVM	99.0
Es-Saady et al. [8]	Multiple	6	In-lab	Classification	SVM	87.8
Liu et al. [12]	Apple	4	In-lab	Classification	AlexNet with Inception	97.6
Mohanty et al. [14]	Multiple	38	In-lab	Classification	GoogLeNet	99.4
Wang et al. [16]	Apple	4	In-lab	Classification	VGG	90.4
Durmuş et al. [18]	Tomato	10	In-lab	Classification	SqueezeNet	94.3
Elhassouny et al. [20]	Tomato	10	In-lab	Classification	MobileNet	90.3
Atabay [22]	Tomato	10	In-lab	Classification	Customized CNN	99.9
Barbedo [24]	Multiple	79	In-field (background removed)	Classification	GoogLeNet	82.0
Atila et al. [26]	Multiple	39	In-lab	Classification	EfficientNet	99.9
Kawasaki et al. [27]	Cucumber	3	In-field	Classification	Customized CNN	94.9
Fujita et al. [29]	Cucumber	9	In-field	Classification	VGG	93.6
Sladojevic et al. [31]	Multiple	15	In-lab & in-field	Classification	Customized CNN	96.3
Ramcharan et al. [32]	Cassava	6	In-field	Classification	Inception V3	93.0
DeChant et al. [33]	Corn	1	In-field	Classification	Multiple CNNs	96.7
Ferentinos [34]	Multiple	58	In-lab & in-field	Classification	VGG	99.5
Chen et al. [36]	Rice and corn	9	In-field	Classification	VGG with Inception	92.0 on rice 80.4 on corn
Picon et al. [38]	Multiple	23	In-field	Classification	ResNet50 with meta-data	98.0
Fuentes et al. [39]	Tomato	10	In-field	Detection & Classification	R-FCN	mAP: 86.0
Lu et al. [40]	Wheat	7	In-field	Detection & Classification	Customized CNN	98.0
Wang et al. [41]	Tomato	11	In-field	Detection & Classification	Mask R-CNN	mAP: 99.6
Ozguven et al. [42]	Sugar beet	4	In-field	Detection & Classification	Updated Faster R-CNN	95.5
Bhatt et al. [43]	Tea	2	In-field	Detection & Classification	YOLOv3	mAP: 86.0
Jiang et al. [44]	Apple	5	In-lab & in-field	Detection & Classification	SSD with Inception	mAP: 78.8
Xie et al. [45]	Grape	4	In-lab	Detection & Classification	Improved Faster R-CNN	mAP: 81.1

diseases and segmenting the locations and shapes of the infected areas. Their Faster R-CNN and Mask R-CNN models showed the highest results of mAP of 88.5% and 99.6%, respectively. Ozguven and Adem [42] designed an updated version of the Faster R-CNN for automatic detection of leaf spot disease in sugar beet. Their overall correct classification rate was 95.5%. Bhatt et al. [43] proposed a diseases and pests detection

from in-field tea images based on YOLOv3 model [54]. They reported that detection using their YOLOv3 model achieved mAP of 86.0% while making the system usable in real time. Jiang et al. [44] built a real-time detection system of apple leaf diseases by combining SSD and Inception modules. Their model showed a detection performance of 78.8% mAP on five disease classes. Xie et al. [45] proposed an improved version of Faster R-CNN model for in-lab grape disease detection. The result of 81.1% mAP on four grape leaf diseases was reported. Table 1.1 provides a summary of recent studies on leaf disease recognition. Systematic reviews of automated leaf disease recognition can be found in [58, 9, 59].

1.1 Motivations

Although the above diagnosis systems have achieved excellent performance on a wide variety of in-field images, they are still far away from being practical and there are several problems remained as follows:

Diagnosing from wide-angle images

The inputs of all abovementioned systems are narrow range images, which contain few targets for diagnosis (i.e., the ROIs are generally located in the center of the input). Applying these models to wide-angle images in large farms would be very time-consuming, since many targets (e.g., leaves) need to be diagnosed. In practice, 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. In this study, we define a narrow range image to be close-up to the camera which contains several targets for diagnosis. On the other hand, a wide-angle image is far-off from the camera and consists of dozen of targets to be diagnosed. Fig. 1.1 shows a comparison between the narrow range input images of the abovementioned systems and the wide-angle

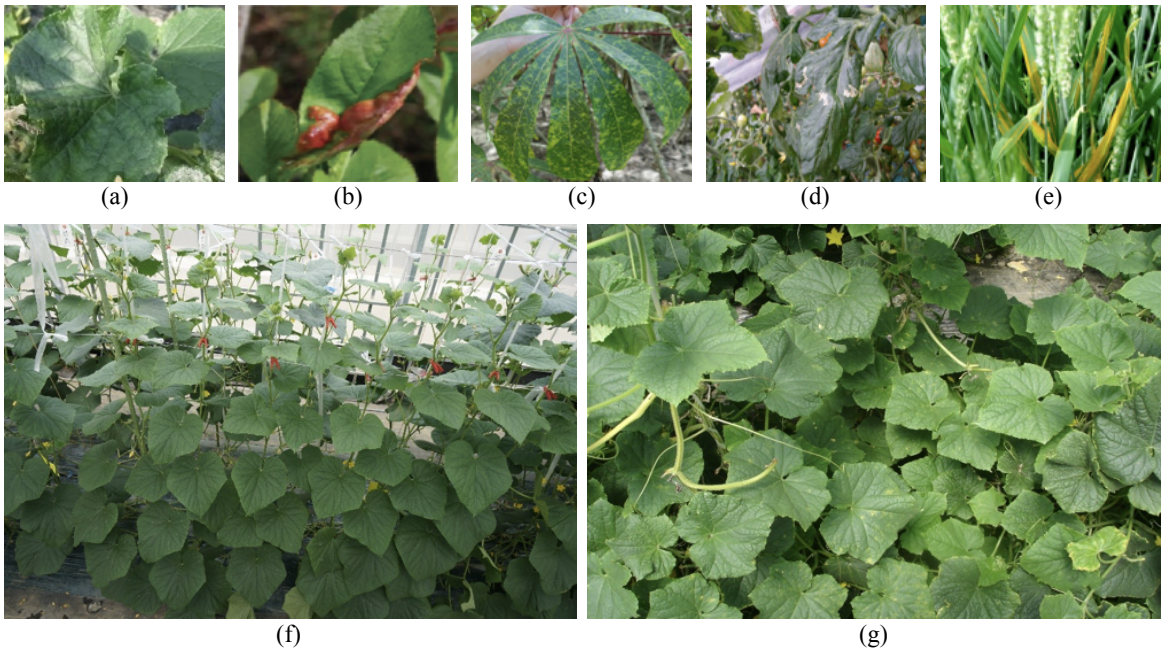


Figure 1.1: The comparison between the narrowed range images (a, b, c, d, e) and wide-angle images (f, g). Wide-angle images are often complex due to the heavily overlapped of multiple leaves. Also, symptoms are scattered in different leaves (g).

images taken in practical situations. The images from (a-e) in [28, 31, 32, 39, 40] with various backgrounds have a narrow range compared to the wide-angle images (f, g). Thus, even though the simultaneous localization and identification systems worked well on narrow range images [39–45], simultaneous processing for wide-angle images is quite difficult. Developing diagnosis systems for wide-angle images (e.g., taken by surveillance cameras) is, however, necessary in practical situations.

In this work, we propose a system that performs leaf detection and leaf diagnosis from wide-angle input images. Fig. 1.2 illustrates an overview of recent studies on plant leaf disease recognition including our proposal (the wide-angle diagnosis system). To the best of our knowledge, this system was the first investigation on plant disease diagnosis from wide-angle images under practical settings (described in detail in Chapter 2).

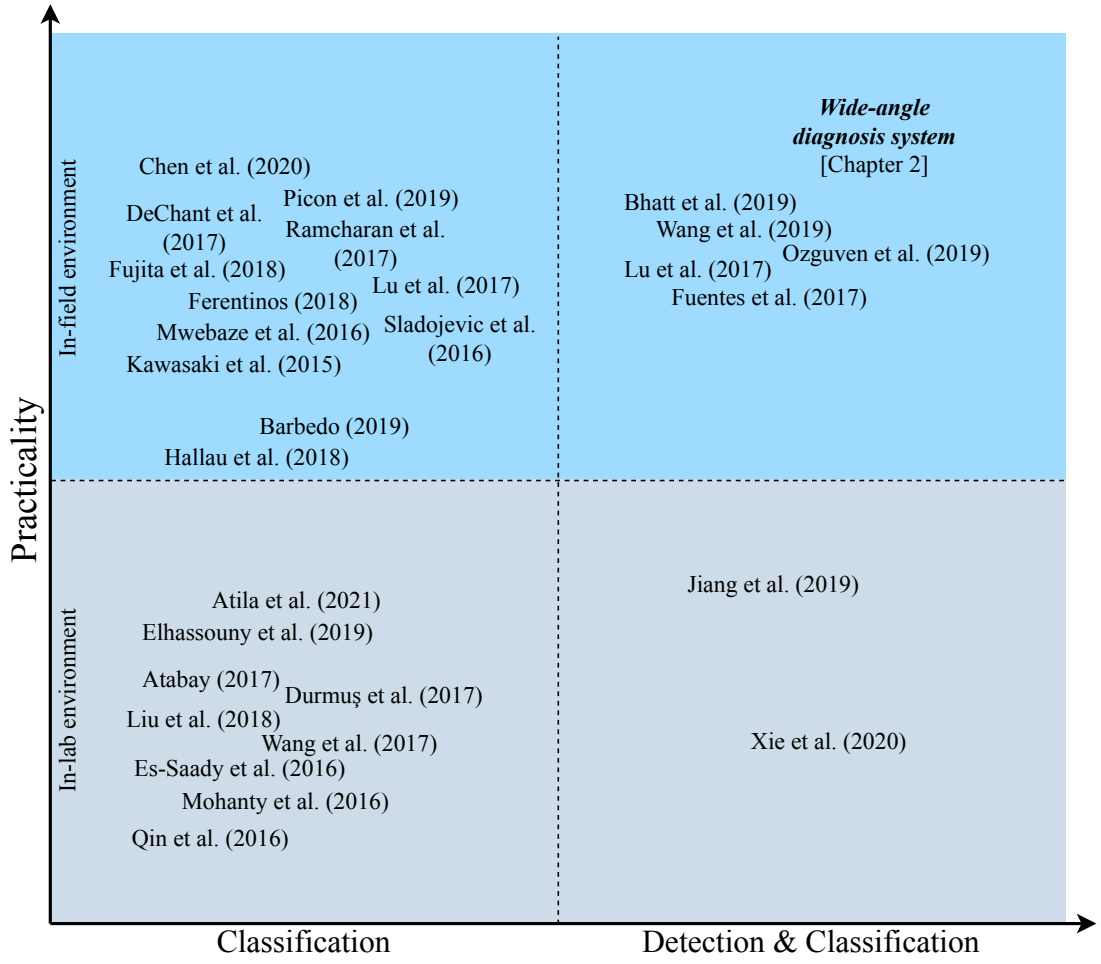


Figure 1.2: An overview of recent studies on plant leaf disease recognition. Our proposal was the first wide-angle plant disease diagnosis system under practical settings.

The lack of high-resolution resources

Several agricultural studies have pointed out that the lack of high-resolution of targets in wide-angle images is the main reason for the relatively low performance. Sa et al. [60] and Bresilla et al. [61] designed systems for the real-time counting of fruits on trees, in order to support robotic harvesting. However, they reported that low-quality test images could cause their detection systems to miss these fruits. Tian et al. [62] developed an algorithm for in-farm apple fruit detection, but their scheme required images with high resolution (HR) and a high-level of detail for accurate detection. In the case of diagnosing disease from wide-angle images, we have experienced that small leaf

Figure 1.3: The comparison between the low-resolution (first row, 4× up-scaled using Bicubic) and the original high-resolution images (second row). The loss of symptom information from low-resolution images will largely reduce the disease diagnostic performance.

sizes and low-quality input images (i.e., low-resolution, blur, poor camera focus, etc.) could significantly reduce the diagnostic performance of their disease detection scheme [63]. Fig. 1.3 shows a visual comparison between the low-resolution (4× up-scaled using Bicubic) and the original high-resolution images. Since practical wide-angle images contain many small leaves, simply enlarging those leaves using conventional techniques is not sufficient enough to recover the loss of disease symptom and will largely degrade the diagnostic performance. We believe that recovering the high-frequency components of images by applying super-resolution (SR) methods offers a promising solution for addressing the abovementioned issue for practical agricultural applications.

Motivated by this, we propose a specially designed SR method namely leaf artifact-suppression super-resolution (LASSR) to improve the performance of plant leaf disease diagnosis from low-resolution (LR) data. Fig. 1.4 shows the difference of the LASSR

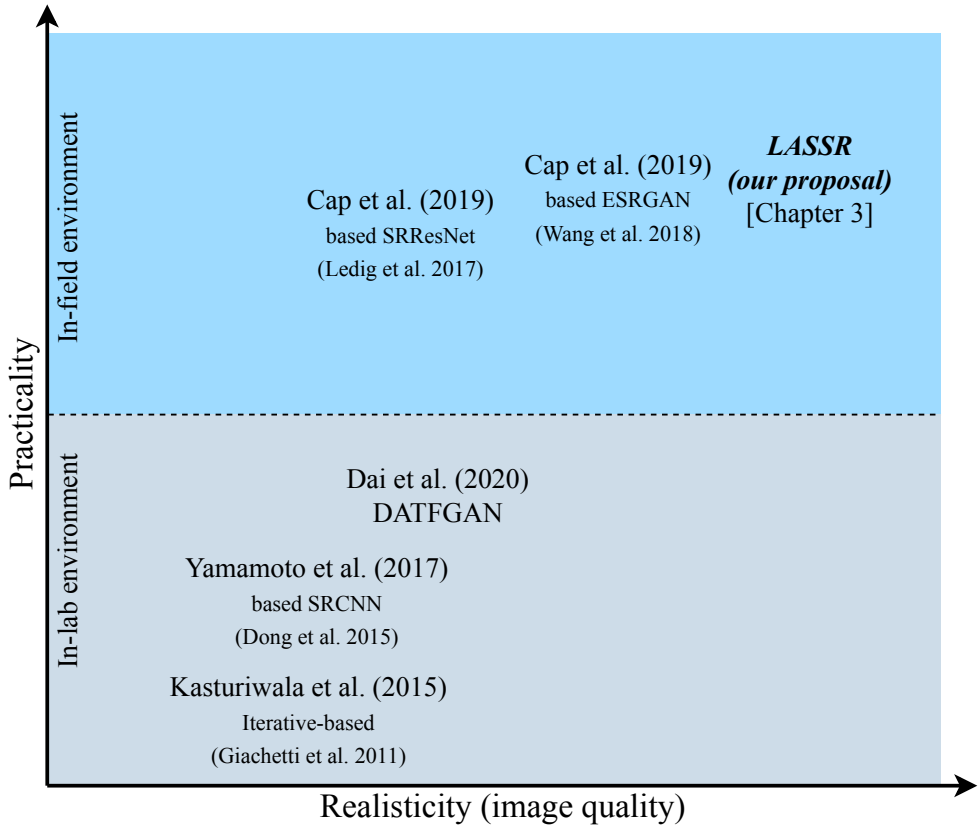


Figure 1.4: An overview of SR studies for plant disease diagnosis.

among other studies of SR for plant disease recognition. Our LASSR is capable of generating high-quality and reliable SR images for practical usages. More details of LASSR will be described in Chapter 3.

The overfitting problem and the difficulty of collecting data

The overfitting problem is particularly long lasting and serious in plant diagnosis tasks, since the image features that provide diagnostic clues (i.e., evidence for classification) are typically much smaller than in general object recognition problems. In general, a deep classifier such as a CNN tends to capture the image characteristics (i.e., brightness, color) of a large area, rather than a faint feature that may indicate disease. In addition, when evaluating a classifier using a dataset divided into training, validation, and test sets (where cross-validation is applied), the "latent similarities" within the dataset

Table 1.2: Decreased discrimination performance on unseen data

Article	Crop	# of class	Dataset condition	Type	Performance on validation data (%)	Performance on unseen data (%)
Mohanty et al. [14]	Multiple	38	In-lab	Classification	99.3	31
Ferentinos [34]	Multiple	58	In-lab & in-field	Classification	99.5	25-35
Cap et al. [63]	Cucumber	2	In-field	Detection & Localization	97.4	68.1
Saikawa et al. [64]	Cucumber	8	In-field	Classification	97.4	40.3
Suwa et al. [65]	Cucumber	2	In-field	Detection & Localization	F1-score: 86.0	F1-score: 19.5

(such as the background, brightness and/or distance between target and camera etc.) works as a positive bias, and generally improves only the superficial diagnostic accuracy, while the accuracy when evaluated on other unknown environments becomes very low [14, 34, 63–65] (see Table 1.2). The evidence confirming the overfitting of models in plant diagnosis tasks has been shown in our previous studies [29, 64] by using Grad-CAM [66] to visualize the key regions of diagnostic evidence. Although these models provided a high diagnosis accuracy, the backgrounds were sometimes considered as diagnostic regions. The most plausible reason for this is that when collecting a dataset, the foreground objects in each image class tend to be incidentally correlated with similar backgrounds. A lack of background diversity could be a distractor, meaning that the model sometimes responds to the background rather than discriminative targets (i.e., leaf regions).

On the other hand, unlike other general computer vision tasks, collecting and labeling disease datasets requires solid biological knowledge. In order to collect gold standard datasets with a wide-variety of diseases, the plants must be grown in a strictly controlled and isolated environment to avoid contamination, which is generally labor-intensive and very expensive. Disease development is also strongly influenced by ambient conditions such as weather, temperature and vector-borne insects. Therefore, several diseases are difficult to collect. It should be noted that recently, several methods based on the generative adversarial networks (GANs) [67] have been proposed to synthesize more artificial images for disease recognition tasks [68–72]. Fig 1.5 shows

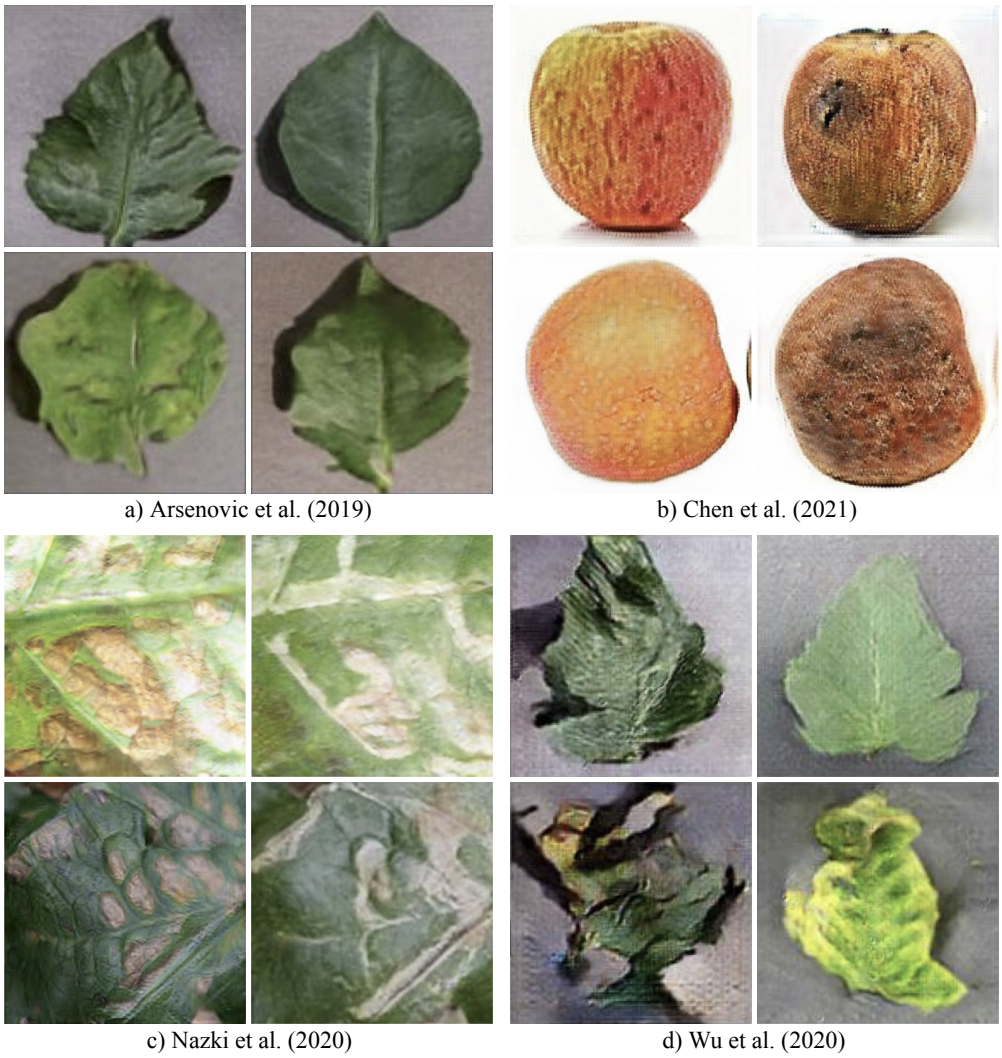


Figure 1.5: Synthesized images by recent GAN-based methods for plant recognition tasks. Images partly captured from above studies. They either used in-lab or close-up images with simple backgrounds.

several examples of synthesized images generated by GAN-based methods for supporting disease recognition. However, these techniques either used in-lab or close-up images with blank backgrounds. Thus, they are impractical and cannot increase the background diversity nor the variety of images. We believe that a method that can realistically and effectively synthesize disease image data under practical settings could greatly reduce the labor-intensive for experts on collecting datasets.

In this work, we propose a novel image-to-image translation method so-called Leaf-GAN that effectively synthesizes a wide variety of high-fidelity training images as well

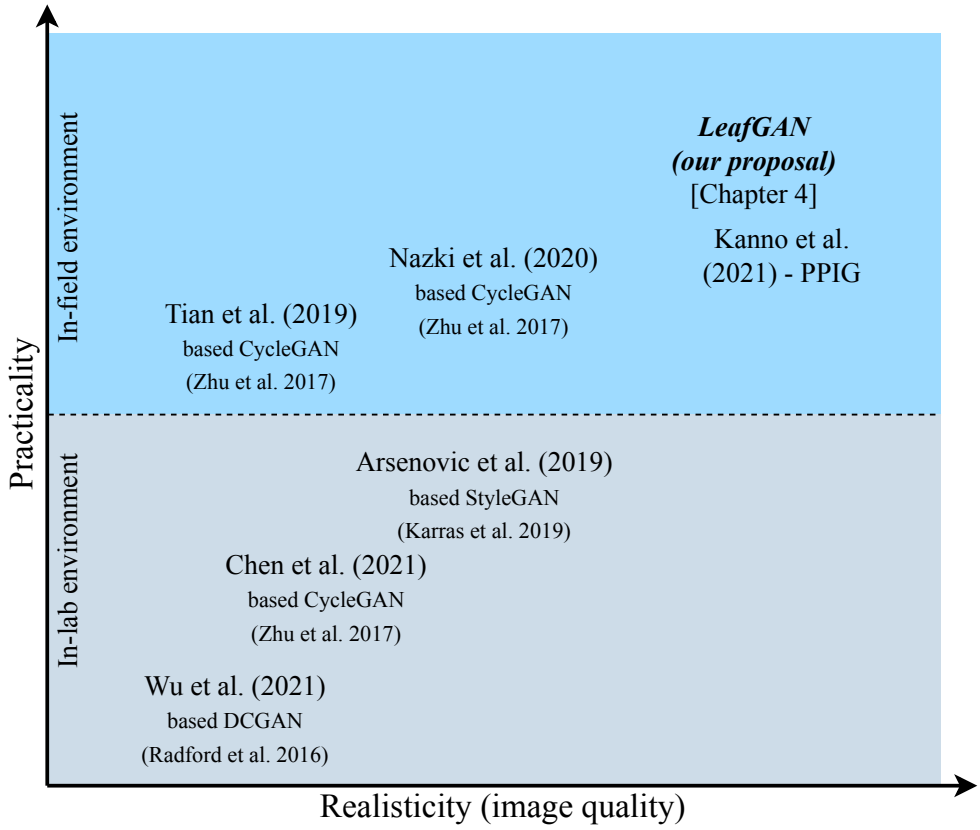


Figure 1.6: An overview of GAN-based image generation studies for plant disease diagnosis.

as reduces the burden of disease data collecting and the overfitting problem. Fig. 1.6 provides an overview of GAN-based image generation methods (abovementioned studies) for plant disease diagnosis including our proposal. Our LeafGAN generates countless diverse, high-quality images as an efficient data augmentation for the diagnosis classifiers. Such generated images can be used as useful resources for improving the performance of disease diagnosis systems. We should note that recently, we have proposed the PPIG method [73] (see Fig. 1.6) for generating disease images from noise vectors. The PPIG is an update and a successor to LeafGAN. Therefore, we decide to only discuss the LeafGAN method in this work. The details of LeafGAN will be described in Chapter 4.

1.2 Thesis structure

The remainder of this thesis is organized as follows:

Chapter 2 describes and discusses the whole schematic of our proposed wide-angle plant diagnosis system for cucumber diseases.

Chapter 3 describes and discusses our proposed LASSR method for supporting disease diagnosis from LR images.

Chapter 4 describes and discusses our novel image-to-image translation LeafGAN model that realistically generate high-quality training data.

Chapter 5 contains the conclusion section.

Chapter 2

The wide-angle plant disease diagnosis system

As mentioned earlier, most of the above systems were designed to diagnose a limited number of targets (e.g., up to a dozen), meaning that there are still limitations when applying these methods to large-scale farm environments. A system that can accurately detect and diagnose diseases from wide-angle images is very important in order to support agricultural practice. However, based on our experiences, it is not easy to develop a practical plant disease diagnosis system for wide-angle images. There are two major problems that need special attention.

The first problem is the overfitting that arises due to the latent similarities between the training and test images, even though they were exclusive to each other. Where the diagnostic performance on real unseen data is usually significantly reduced. This problem is predicted to be more serious when wide-angle images are used, because the same or similar objects may appear in different images.

The second problem is the labor cost and the required accuracy of the gold standard assignment. When end-to-end diagnosis systems (i.e., simultaneous disease localization and identification) are built, numerous training images with a huge number of bounding boxes are required, along with the appropriate disease labels. Moreover, there are innumerable objects in the images such as overlapping leaves, and their resolution is often insufficient, making it very difficult to label each object with an appropriate ground truth.

In this work, we propose a two-stage diagnosis system that performs leaf detection and leaf diagnosis independently. We believe that two-stage diagnosis systems have

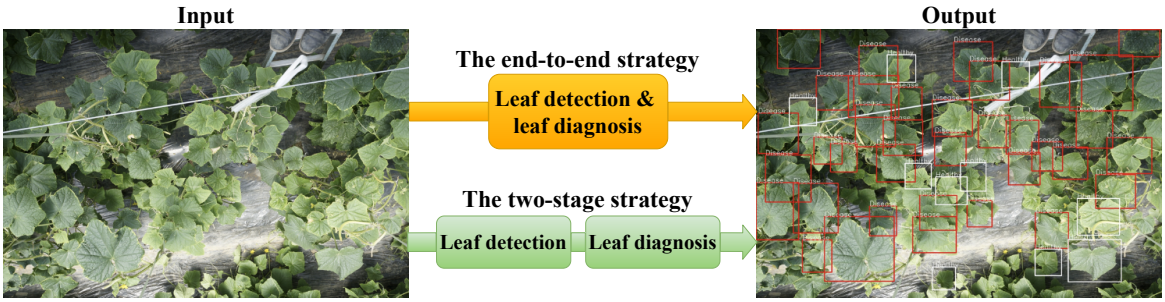


Figure 2.1: Overview of the end-to-end and two-stage strategies (red and white boxes indicate disease and healthy leaves, respectively).

several advantages over the end-to-end systems and that they can overcome the issues on developing practical plant disease diagnosis system for wide-angle images. Firstly, two-stage systems have the detection stage and the diagnosis stage separately; thus, labeling of the bounding boxes of the objects (i.e., leaves, fruits) to be detected is easy, since it does not require disease-specific knowledge and can be done by non-experts. Secondly, labeling a single object or collecting labeled single-object images is much simpler than for wide-angle images, as mentioned above. The diagnosis stage therefore could be trained with a wider variety of data, boosting the robustness of the two-stage system when new types of data are encountered. To this end, we examine and compare two types of diagnosis strategy (i.e., end-to-end versus two-stage) for practical wide-angle cucumber images in terms of disease diagnostic performance under different evaluation environments.

2.1 Materials and methods

Fig. 2.1 shows an overview of the two different types of diagnosis strategy. The first approach is an end-to-end strategy, which simultaneously performs leaf detection and leaf diagnosis based on a sophisticated object detection framework such as the single shot multibox detector (SSD) [51] or Faster R-CNN [48]. The second approach is a two-stage strategy that performs these functions separately. In this study, we carry out

diagnosis using these strategies in order to estimate whether each leaf in a wide-angle image is healthy or diseased. The reason for using only two diagnostic classes is that it is difficult to assign gold standard labeling to wide-angle images, as described earlier. For both systems, we compare the final diagnostic performance on a test dataset from the same farm and a dataset from different farms. This comparison is to examine the effect of the latent similarities between the training and test datasets on the final diagnostic performance. We then discuss which approach is more suitable for real cultivation conditions.

2.1.1 Object detection methods

Recently, deep learning methods have achieved remarkable performance in object detection. In general, there are two types of frameworks among deep learning-based object detection models. The first framework is region proposal-based which consists of a region proposal module to output a set of rectangular object proposals from input image, and a classifier module to predict the final classes from those object proposals. Models like R-CNN [46], Fast R-CNN [47], Faster R-CNN [48] are of this type. These methods are high accuracy but relatively low in terms of inference speed. The second framework is regression-based which does not depend on the region proposals but straightly maps from image pixels to bounding box coordinates and class probabilities for these boxes. Models like YOLO V1-V3 [50, 53, 54], SSD [51] are of this type. These types of methods are extremely fast but at a cost of decreasing accuracy.

To study the model selection, in this work, we select one method from each object detection type namely Faster R-CNN and SSD for our experiments.

Faster R-CNN

Faster R-CNN consists of two parts: a fully convolutional region proposal network (RPN) that generates ROI proposals from input images, followed by a downstream

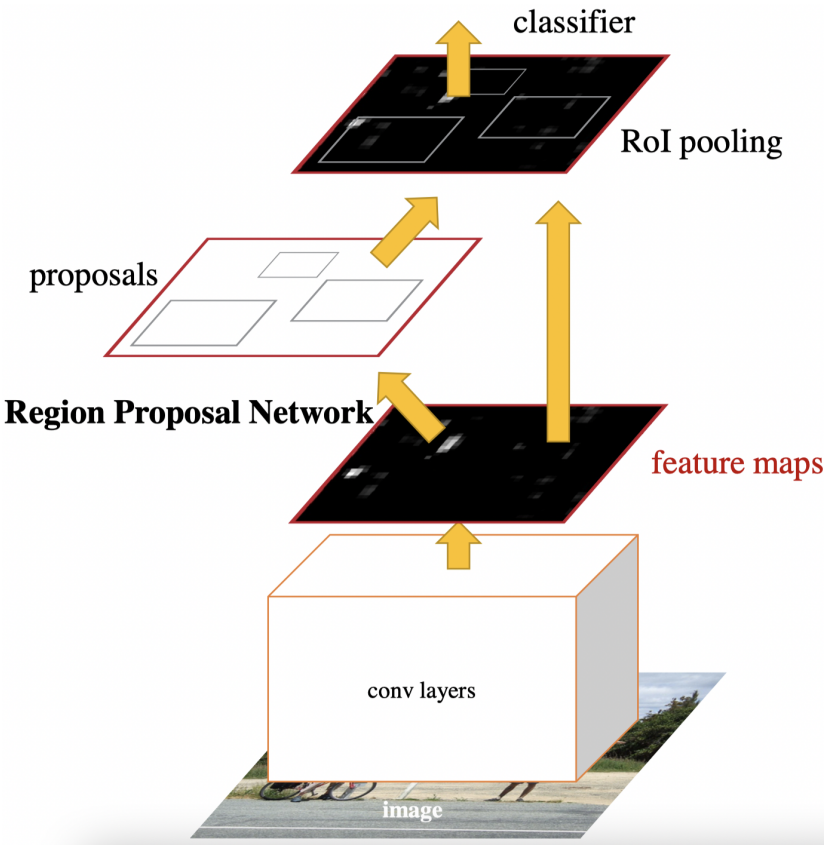


Figure 2.2: Architecture of the Faster R-CNN object detection method (figure captured from [48]).

Fast R-CNN classifier for proposal classification and bounding boxes regression. Fig. 2.2 illustrates the overview of the Faster R-CNN method. In RPN, region proposals are generated by sliding a small network over the feature maps from the last shared convolutional layer of an ImageNet pre-trained model. This small network takes as input a 3×3 spatial window of the convolutional feature map. At each sliding-window location, RPN simultaneously predicts multiple rectangular boxes with predefined ratios and scales, where the number of boxes is denoted as k. Those boxes are called anchors. Anchor coordinate offsets and objectness scores that estimate probability of object or not object for each anchor are learned from the feature maps. Proposals are generated by adjusting anchors with coordinate offsets. In the Faster R-CNN literature, number of anchors is $k = 9$. For a convolutional feature map of a size $W \times H$, there will be $W \times H \times k$ generated anchors in total. Fig. 2.3 shows the RPN architecture for

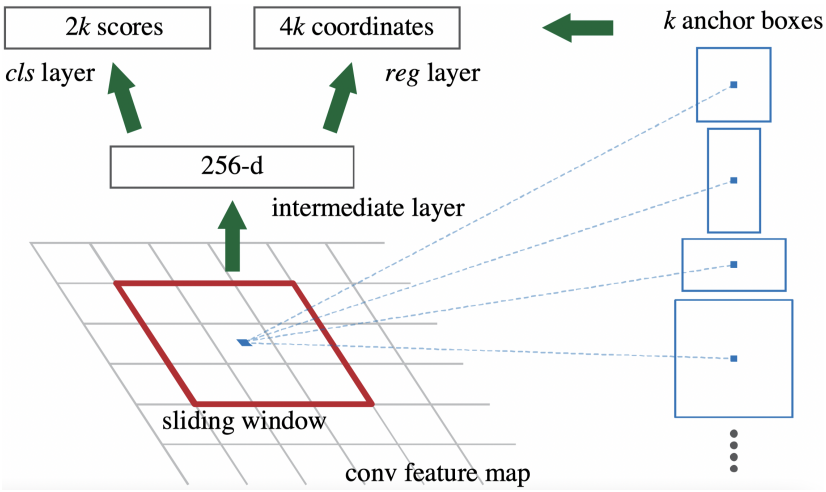


Figure 2.3: The region proposal network (RPN). At each sliding-window location, k anchors with predefined ratios and scales are predicted. k anchors correspond with $4k$ coordinates and $2k$ scores (two-class of object *vs.* not object). (figure captured from [48]).

generating region proposals.

For training RPN, an anchor which has the Intersection-over-Union (IoU) overlap higher than 0.7 with any ground-truth box will be considered as positive. Anchors with the IoU ratio is lower than 0.3 for all ground-truth boxes are considered as negative. Anchors that are neither positive nor negative do not contribute to the training objective. The loss functions for RPN will be the log loss over two classes (an anchor is an object or not) and the smooth L_1 [47] for bounding box coordinate regression.

In the downstream Fast R-CNN classifier, the box proposals previously generated by RPN are used to predict the class probability and bounding box for each region proposal. Details of training this Fast R-CNN classifier can be found in [47].

Training Faster R-CNN consists of four steps. First step, the RPN is trained with a backbone of an ImageNet pre-trained model for the region proposal prediction. Second step, the separated Fast R-CNN classifier is trained using the proposals generated by the RPN in the first step. The backbone of this network is also an ImageNet pre-trained model but not the same as in step one. Third step, the fixed backbone network from the second step is used to initialize RPN training. This time, the backbone network

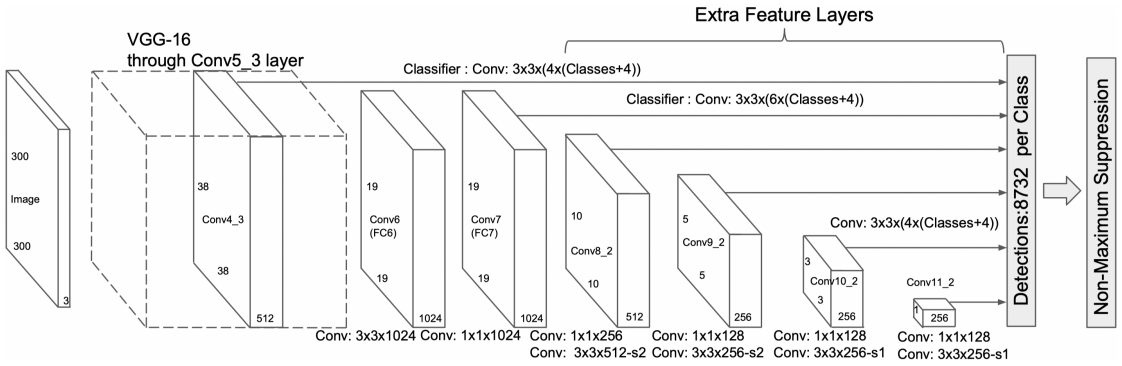
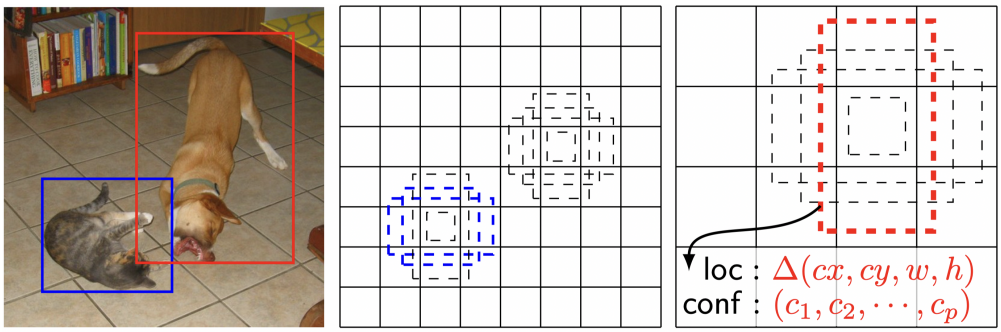


Figure 2.4: Architecture of the SSD object detection method. (figure captured from [51]).

is fixed (not trainable) and only fine-tune the new RPN. Finally, the backbone and the RPN networks from the third step are fixed and only fine-tune the Fast R-CNN classifier. From this step, the whole network is trained jointly end-to-end. The final detection results are formed by a non-maximum suppression step. More details of training and implementation can be found in [48].

Single shot multibox detector

The single shot multibox detector (SSD) approach directly predicts a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes using a feed-forward convolutional network. SSD combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes. Different from Faster R-CNN, SSD does not depend on the region proposals and encapsulates all computation in a single network. Fig. 2.4 illustrates the architecture of the SSD method. In the early network layers, SSD uses an ImageNet pre-trained model (e.g., VGG-16 [17]) to obtain the meaningful image features. Based on these features, several convolutional layers are added to allow predictions of detections at multiple scales. Similar to Faster R-CNN, SSD slides a 3×3 kernel over those feature maps. At each feature map cell, SSD predicts k anchors of different aspect ratios. For each anchor box, c class scores that indicate the presence of a class instance, and the 4



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Figure 2.5: Predicting anchors at different scales in SSD. At each feature map cell, coordinate offsets and class scores of four anchors are computed. (figure captured from [51]).

coordinate offsets are computed. For a $m \times n$ feature map, this yields $(c+4)k \times m \times n$ outputs. Fig. 2.5 shows an example of anchors prediction at different feature scales.

An anchor box which has the IoU overlap higher than 0.5 with any ground-truth box will be considered as positive. Otherwise, they are considered as negative samples. Since most of these boxes are negative, SSD introduces the hard negative mining at training time to keep the ratio between the negatives and positives is at most 3:1. Loss functions of SSD consists of a confidence loss and a regression loss similar to Faster R-CNN. The confidence loss is the softmax loss over multiple classes confidences, and the regression loss is the smooth L_1 as in [47]. At inference time, a non-maximum suppression step is used to produce the final detection results. More details of training and implementation can be found in [51].

2.1.2 Datasets

In this work, we use the following two datasets to explore suitable configurations for automatic wide-angle diagnosis of plant disease. Table 2.1 shows the summary of datasets used in our study.

Table 2.1: Summary of datasets used in this study (wide-angle dataset and single-leaf dataset)

	Wide-angle dataset			**Single-leaf dataset**	
	Train	Test	Unseen	Cropped	All
# of images	867	96	51	22,196	50,000
# of bounding boxes	22,196	2,369	1,829		

The wide-angle dataset

A total of 963 wide-angle images of cucumbers were acquired from several farms, using various digital cameras. Each wide-angle image contained numerous cucumber leaves that overlapped each other and was taken under different light conditions (see Fig. 2.1, Fig. 2.6–2.7 for sample images). The images contained a total of 24,565 leaves, of which 16,924 were healthy and 7,641 were diseased. All of the wide-angle images were annotated by experts, and bounding boxes were created for each leaf. We randomly divided the images, using 90% (867 wide-angle images containing 15,369 healthy and 6,827 diseased leaves) for training, and the rest for testing (96 wide-angle images containing 1,555 healthy and 814 diseased leaves). We refer to these sets of images as the wide-angle$_{\text{train}}$ and the wide-angle$_{\text{test}}$ datasets, respectively.

In order to evaluate end-to-end systems and two-stage systems equally, we prepared 51 wide-angle cucumber images taken from completely different farms. A total of 1,829 single leaves (of which 820 are healthy and 1,009 diseased) were also annotated by experts. We used this dataset only for the final diagnostic test and refer to it here as the wide-angle$_{\text{unseen}}$ dataset.

Our wide-angle images mainly had two aspect ratios, 2:3 and 3:4, and the typical resolution of these images was between 12 and 20 megapixels. They were resized to either 512×512, 600×900 or 600×800 pixels, depending on the architecture of the end-to-end models (as described in more detail in the experimental section).

The single-leaf dataset

The single-leaf dataset was used for training the diagnosis stage of the two-stage systems. From the 867 images in the wide-angle$_{\text{train}}$ dataset, we cropped all the gold standard bounding boxes (a total of 22,196 images, 15,369 healthy and 6,827 diseased), each containing one cucumber leaf, to form the dataset. We refer this dataset as the single-leaf$_{\text{cropped}}$ dataset. In addition, we combined these images with another set of single-leaf images collected from Saitama Agricultural Technology Research Center, Japan. Note that these images were not included in the abovementioned wide-angle dataset. This formed the single-leaf$_{\text{all}}$ dataset, which contains 50,000 images of single cucumber leaves (25,000 healthy and 25,000 diseased).

The reason for building this larger single-leaf$_{\text{all}}$ dataset is to verify the advantages of the two-stage systems, as hypothesized earlier in the introduction. The end-to-end systems only accept annotated leaf regions in wide-angle images in the training set, while the two-stage systems could include additional single-leaf images in the training of the diagnosis stage. Note again that the acquisition of labeled single-leaf images is much easier than from wide-angle images. We believe that adding a variety of single-leaf images to the dataset can improve the robustness of the diagnostic model. The resolution of the single-leaf dataset was normalized to 224×224 pixels.

2.1.3 Wide-angle plant diagnosis systems

End-to-end systems

We first built our end-to-end systems using the SSD512 and Faster R-CNN models. The input image size was resized to 512×512 pixels for the SSD model, while for the Faster R-CNN, we resized the input images to sizes 600×900 or 600×800 pixels, corresponding to images with aspect ratios of 2:3 or 3:4. The backbone of these models was basically the VGG-16 [17] model pre-trained with the ImageNet dataset [11], and they were

fine-tuned with the wide-angle$_{\text{train}}$ dataset. The diagnostic performance of the end-to-end systems was evaluated and compared on the wide-angle$_{\text{test}}$ and wide-angle$_{\text{unseen}}$ datasets.

Two-stage systems

A two-stage system is a combination of a leaf detection stage and a leaf diagnosis stage. In the leaf detection stage, we used the above end-to-end systems (i.e., SSD512 or Faster R-CNN) as the leaf detectors to enable an unbiased comparison. In the subsequent leaf diagnosis stage, the detected leaves were diagnosed using an additional CNN model called DiagNet. This classifier was also fine-tuned based on the pre-trained VGG-16 network with two outputs, i.e., healthy or diseased. Our DiagNet model accepts a color image with a size of 224×224 pixels. In this work, we froze the first ten convolutional layers and fine-tuned the last six layers (three convolutional and three fully-connected layers).

For experimental purposes, we trained two versions of the DiagNet model for performance comparison. The first model, named DiagNet$_{\text{cropped}}$, was trained only on the single-leaf$_{\text{cropped}}$ dataset (22,196 images), while the other, called DiagNet$_{\text{all}}$, was trained on the single-leaf$_{\text{all}}$ dataset (50,000 images). The diagnostic performance of the two-stage systems was also evaluated and compared using the wide-angle$_{\text{test}}$ and wide-angle$_{\text{unseen}}$ datasets.

Training wide-angle plant diagnosis systems

To train the end-to-end systems, the Faster R-CNN and SSD512 models were fine-tuned using the wide-angle$_{\text{train}}$ dataset. We followed the training strategy used in the original Faster R-CNN and SSD papers, fine-tuning the models using stochastic gradient descent (SGD) with momentum [74] with an initial learning rate of 10^{-3} a momentum of 0.9, and a weight decay of 0.0005. The mini-batch size was set to one

to train the Faster R-CNN and 16 to train the SSD512. The training was terminated after 50,000 iterations.

For the two-stage systems, the $\text{DiagNet}_{\text{cropped}}$ and $\text{DiagNet}_{\text{all}}$ were trained on the $\text{single-leaf}_{\text{cropped}}$ and $\text{single-leaf}_{\text{all}}$ datasets, respectively. During the training, we applied augmentation on the fly, using horizontal and vertical flipping, and random 90 degrees rotations. We used the SGD momentum optimizer with the same hyper-parameters when training both the end-to-end systems and our two-stage models. The minibatch size was set to 256, and we terminated the training process after 30 epochs.

2.2 Experimental results

We compare the diagnostic performance of the two different diagnosis strategies for the wide-angle pictures taken on the same farm and those from different farms. Again, it should be noted here that the purpose of this experiment is to find a suitable configuration for practical systems based on this comparison. More specifically, we clarify the effect of the latent similarities in the dataset, and propose a suitable solution to this problem. In this experiment, diagnosis bounding boxes with an IoU ≥ 0.5 which correspond to the ground-truth label are regarded as correct detection results. We use the evaluation criteria of precision, recall and F1-score for both healthy and diseased cases, and calculate the average diagnostic F1-score by averaging the F1-scores of the healthy and diseased leaves as an indicator of the overall diagnostic performance.

Experiment 1: Diagnosing the $\text{wide-angle}_{\text{test}}$ dataset

Table 2.2 shows a comparison of the performance in terms of leaf detection and leaf diagnosis on the $\text{wide-angle}_{\text{test}}$ dataset (96 images, containing 1,555 healthy and 814 diseased leaves). These results show that the best leaf detection performance is achieved by SSD512 with an F1-score of 91.5%, which is slightly better than the Faster R-CNN with 90.4%. The diagnostic results show that the end-to-end systems give better

Table 2.2: Performance comparison between end-to-end and two-stage systems on the wide-angle$_{\text{test}}$ dataset

		Leaf detector performance			Leaf disease diagnostic performance						
					Healthy			Disease			Average
		F1-score [%]	Precision [%]	Recall [%]	F1-score [%]	Precision [%]	Recall [%]	F1-score [%]	Precision [%]	Recall [%]	F1-score [%]
SSD512 [51]	End-to-end	**91.5**	89.8	93.3	**87.8**	88.1	87.5	**84.1**	81.4	86.9	**86.0**
	Two-stage (DiagNet$_{\text{all}}$)				82.6	88.0	77.9	73.2	62.8	87.6	77.9
	Two-stage (DiagNet$_{\text{cropped}}$)				**84.6**	86.0	83.3	**79.1**	75.0	83.7	**81.9**
Faster R-CNN [48]	End-to-end	90.4	86.7	94.4	85.2	82.8	87.8	81.5	78.7	84.6	83.4
	Two-stage (DiagNet$_{\text{all}}$)				80.8	81.5	80.1	75.1	68.3	83.5	78.0
	Two-stage (DiagNet$_{\text{cropped}}$)				82.8	81.6	84.1	77.7	73.2	82.9	80.3

The **red** and **blue** colors indicate the best performance of the end-to-end and two-stage systems on the wide-angle$_{\text{test}}$ dataset, respectively.

performance on diagnosing diseased leaves compared to the two-stage systems. The best average diagnostic F1-score is 86.0% for the SSD512, while the best result for the two-stage systems is 81.9% for the DiagNet$_{\text{cropped}}$ using SSD512 as the leaf detector. The overall ranking indicates that of the end-to-end systems, the SSD512 performed slightly better than the Faster R-CNN. For the two-stage systems, the DiagNet$_{\text{cropped}}$ achieved higher results than the DiagNet$_{\text{all}}$ using both SSD512 and Faster R-CNN as the leaf detectors. We should note here that the DiagNet$_{\text{all}}$ was trained with a much larger leaf image dataset (roughly 2.3 times larger than the DiagNet$_{\text{cropped}}$), but the performance was consistently lower.

Fig. 2.6 shows some examples from this experiment. The red and white boxes indicate diseased and healthy leaves, respectively. Based on the results, it can be seen that although there is a slight difference in the performance for the two types of system, both the end-to-end and two-stage systems can correctly diagnose almost all leaf locations, giving reasonable diagnostic performance.

Experiment 2: Diagnosing the wide-angle$_{\text{unseen}}$ dataset

Table 2.3 shows a comparison of the performance of leaf detection and diagnosis for the wide-angle$_{\text{unseen}}$ dataset, which contains 51 images (820 healthy and 1,009 diseased leaves). Again, these images were taken in a completely different environment from the above wide-angle$_{\text{test}}$ dataset.

The leaf detection performance for these unseen images was significantly reduced with respect to the recall, but both SSD and Faster R-CNN maintained a very high

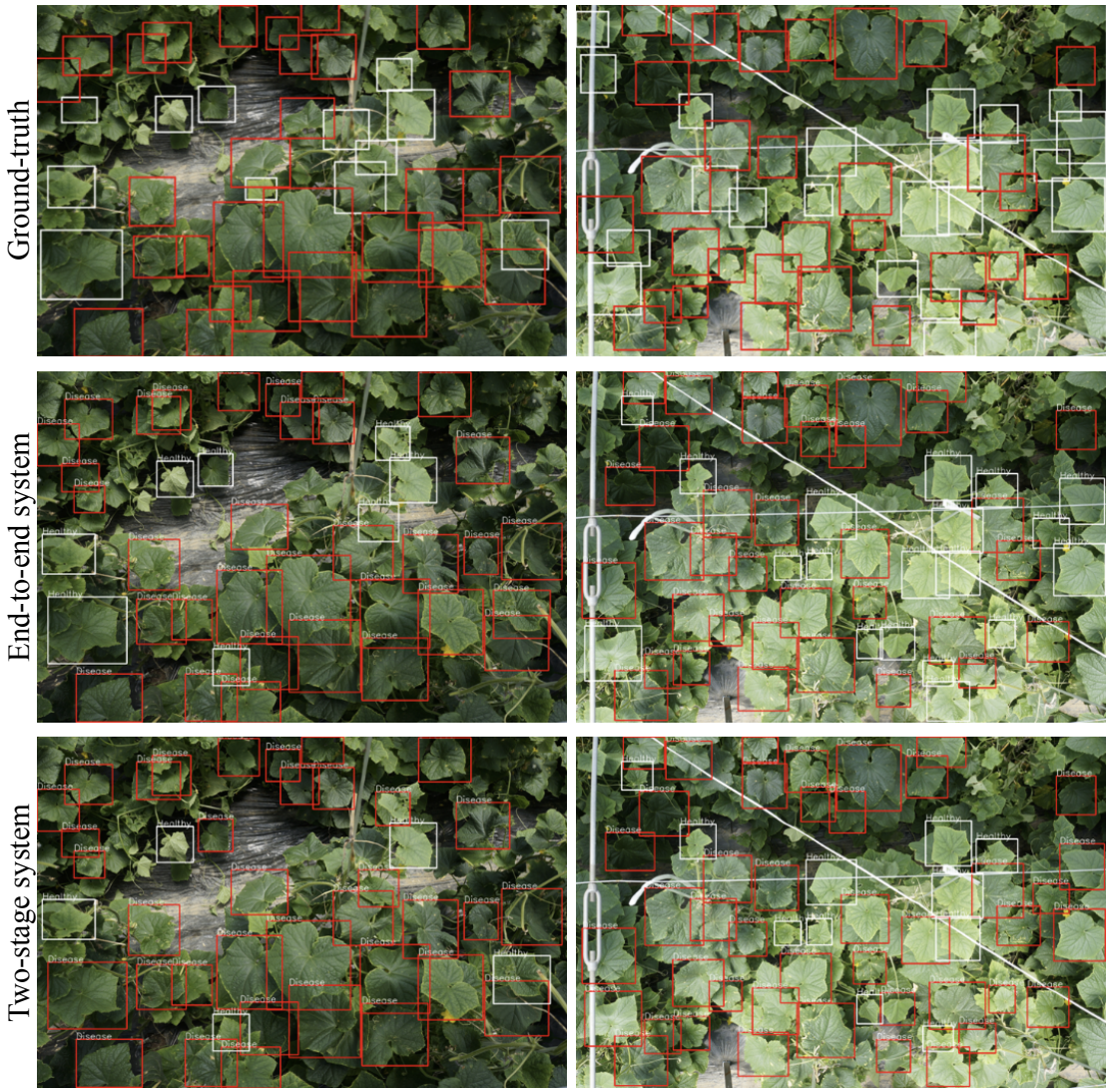


Figure 2.6: Final diagnostic results of two diagnosis strategies on the wide-angle$_{\text{test}}$ dataset. The first row represents the ground-truth images, the second and third rows indicate the results of the end-to-end SSD512 system and the two-stage system with DiagNet$_{\text{all}}$, respectively. Note that the red and white boxes show diseased and healthy cases, respectively.

value of precision (94.4 – 96.1%). From a practical point of view, this can be considered reasonable, since we still can detect most leaves precisely. The best leaf detection performance in this case is achieved by the Faster R-CNN model, with an F1-score of 54.2% as compared to the SSD512 with 51.8%. The final diagnostic performance was totally dissimilar from the previous experiment, as showed in Table 2.2. Although all systems showed a considerably reduced diagnostic performance, the two-stage systems

Table 2.3: Performance comparison on the wide-angle$_{\text{unseen}}$ dataset

		Leaf detector performance			Leaf disease diagnostic performance						
					Healthy			Disease			Average
		F1-score [%]	Precision [%]	Recall [%]	F1-score [%]	Precision [%]	Recall [%]	F1-score [%]	Precision [%]	Recall [%]	F1-score [%]
SSD512 [51]	End-to-end	51.8	96.1	35.5	34.5	39.4	30.7	4.4	66.7	2.3	19.5
	Two-stage (DiagNet$_{\text{all}}$)				36.2	53.9	27.2	33.4	81.2	21.0	34.8
	Two-stage (DiagNet$_{\text{cropped}}$)				35.4	44.1	29.6	17.4	80.6	9.9	26.4
Faster R-CNN [48]	End-to-end	**54.2**	94.4	38.0	**35.1**	38.2	32.4	**6.2**	84.6	3.2	**20.7**
	Two-stage (DiagNet$_{\text{all}}$)				**35.6**	53.2	26.7	**38.9**	79.9	25.7	**37.3**
	Two-stage (DiagNet$_{\text{cropped}}$)				34.7	40.4	30.4	15.9	75.0	8.9	25.3

The **red** and **blue** colors indicate the best performance of end-to-end and two-stage systems on the wide-angle$_{\text{unseen}}$ dataset, respectively.

outperformed the end-to-end systems. The best average diagnostic F1-score for the two-stage systems is 37.3% for the DiagNet$_{\text{all}}$, while the best end-to-end system is the Faster R-CNN diagnostic system with only 20.7%. It is notable that both the SSD512 and Faster R-CNN end-to-end systems were almost unable to detect the locations of diseased leaves, with a very low F1-score of 4.4 – 6.2%.

In contrast, the two-stage system (DiagNet$_{\text{all}}$) achieved much higher recall and precision for the diseased cases, attaining F1-score of 33.4 – 38.9%. The diagnostic performance of DiagNet$_{\text{all}}$ was also well balanced between the healthy and diseased cases. Along with that, the DiagNet$_{\text{cropped}}$ still attained a desirable result in terms of diagnosing disease, even with a smaller set of training data. The overall performance ranking is opposite to that in the previous experiment, with the best result achieved by DiagNet$_{\text{all}}$, the second best by DiagNet$_{\text{cropped}}$, and the lowest by the end-to-end systems.

Fig. 2.7 shows typical examples of the final diagnostic results for the wide-angle$_{\text{unseen}}$ dataset. As mentioned above, the end-to-end systems typically failed to diagnose the positions of diseased leaves, while the two-stage systems could correctly identify the important locations of diseased leaves for an unseen dataset, outperforming the end-to-end systems.

2.3 Discussion

We investigated changes in diagnostic performance by experimenting with different practical scenarios, and have shown that the final diagnostic performance varies greatly

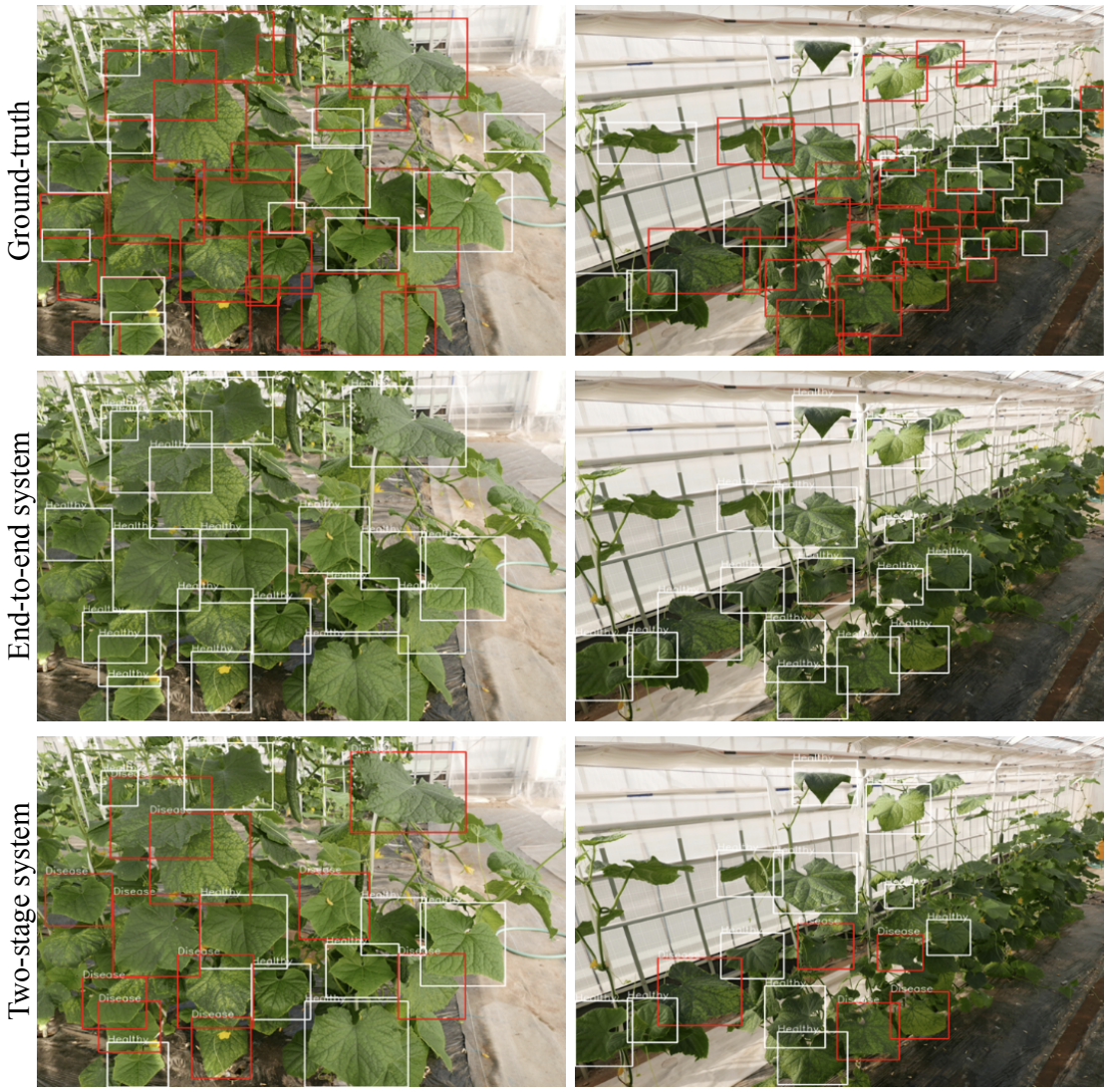


Figure 2.7: Final diagnostic results of two strategies on the wide-angle$_{\text{unseen}}$ dataset. The first row shows the ground-truth images, while the second and third rows indicate the results of the end-to-end Faster R-CNN system and the two-stage system with DiagNet$_{\text{all}}$, respectively. The end-to-end system completely failed to detect the diseased leaves, while the two-stage system correctly diagnosed the important leaf locations in the unseen dataset.

depending on whether the test data form part of the whole dataset (Table 2.2; Experiment 1) or a completely different dataset (Table 2.3; Experiment 2). The results of both experiments indicated that the end-to-end systems were overfitted to the wide-angle$_{\text{train}}$ dataset. The end-to-end Faster R-CNN and SSD512 models showed very high performance for disease diagnosis on the wide-angle$_{\text{test}}$ dataset (F1-score 81.5 – 84.1%) but

extremely poor performance on the wide-angle$_{\text{unseen}}$ dataset (F1-score 4.4 – 6.2%). The primary reason for this huge gap is the large latent similarities between the training and test data (i.e., there is a high possibility that the same or a similar object appears in the wide-angle images in the same field). In addition, collecting a sufficiently large and reliable wide-angle training dataset is difficult even for experts, because the leaf objects that need to be labeled are often small, with unclear appearance. This limits the scalability of the system, leading to non-generalization to the unseen dataset. In this case, the end-to-end systems are not the best choice for practical automated disease diagnosis.

In contrast, although the two-stage systems attained a slightly lower F1-score than the end-to-end systems in Experiment 1, they showed superior performance in diagnosing disease cases from the wide-angle$_{\text{unseen}}$ dataset, which represented a more practical scenario in Experiment 2 (with an F1-score of 33.4 – 38.9% compared to 4.4 – 6.2% for the end-to-end systems). We also showed that even when we used only the cropped single-leaf images from the same training dataset (i.e., single-leaf$_{\text{cropped}}$ dataset is cropped from the wide-angle$_{\text{train}}$ dataset) as in the end-to-end systems, DiagNet$_{\text{cropped}}$ still achieved better results for diagnosing an unseen disease, with an F1-score ranging from 15.9% to 17.4%, thus confirming the effectiveness of the two-stage strategy in real situations. It should be noted that the performance of the leaf diagnosis stage of the two-stage systems greatly depends on the performance from the leaf detection stage. Designing a good leaf detector could further increase the performance of the overall system and we tend to improve it in the future.

We achieved these results thanks to the advantages of the training method in the two-stage strategy. First, the leaf diagnosis stage in a two-stage system accepts single-leaf images as input, and can be trained with a wide variety of data. Second, the collection of single-leaf images for the disease classifier is much more straightforward than for end-to-end systems (i.e., wide-angle images). These two properties therefore

contribute to improving the generalization of the two-stage systems and increasing their scalability.

In general, although the detection of the location of a diseased leaf is very important, it is unnecessary to accurately detect and diagnose all the leaves. Once the areas of diseased leaves are detected, further inspection can be applied to the nearby locations, since plant diseases often spread outwards from a given area. In this context, a two-stage system is a suitable choice for the diagnosis of plant diseases from wide-angle images in practical situations.

Chapter 3

Effective super-resolution method for plant disease diagnosis

It has been known that the lack of high-resolution (i.e., low-quality input images, blur, poor camera focus, etc.) could significantly reduce the diagnostic/detection performance of targets in wide-angle images [60, 61, 63]. One possible solution would be to use HR camera devices to obtain high-quality images, but this is generally expensive to deploy in practice.

In the main field of computer vision, a certain degree of resolution is necessary to achieve high discriminative power. Most CNNs using ImageNet datasets [11] have long used a resolution of 224×224, following the achievements of AlexNet [10], but recently, higher resolution models have been more successful. AmoebaNet [75] and GPipe [76] have achieved state-of-the-art levels of accuracy for ImageNet classification with resolutions of 331×331 and 480×480, respectively. Tan and Le [25] proposed a scaling method called the compound coefficient that balances the depth and width (number of filters) of the network and the resolution of the input image. They demonstrated that their EfficientNet achieved state-of-the-art results with significantly lower computational requirements. As these results show, image resolution is an important factor in achieving high-accuracy recognition.

In practical agricultural applications, we believe that recovering the high-frequency components of images by applying SR methods offers a promising solution for addressing the abovementioned issue. SR techniques can be divided into two broad types called "registration-type" and "learning-type". Registration-type SR techniques utilize a large number of images in order to increase the pixel density of the image. In

this way, the true high-frequency components of the image can be estimated using an appropriate reconstruction algorithm. Typical registration-type SR techniques utilize maximum likelihood (ML) [77], maximum a posteriori (MAP) method [78], or projection onto convex sets (POCS) [79] as a reconstruction algorithm. However, since these classical methods require a relatively high number of observed images, precise correction of the positional deviations between images using sub-pixel image registration is necessary for the successive reconstruction process. SR methods using multi-camera devices [80, 81] also fall into this category; although these techniques have been commercialized in recent years with the spread of high-end mobile phone cameras, they were originally expensive.

Learning-type SR techniques usually utilize only one base image from the observed set and predict unknown details. Before deep learning techniques were developed, they often applied pre-trained database and/or estimators [82] or an interpolation approach based on signal processing techniques [83]. The quality and resolution of SR images generated by these methods were usually lower than the other and required appropriate settings. However, thanks to the modeling power of CNNs, recent SR methods based on a single image, known as single image super-resolution (SISR) techniques, have shown excellent performance [84–86]. Dong et al. [84] first proposed the super-resolution convolutional neural network (SRCNN), which provided end-to-end training between low-resolution (LR) and HR images. Ledig et al. [85] then proposed SRGAN as the first SR method to adopt a GAN [67] algorithm, resulting in indistinguishable super-resolved images from high-resolution images. Recently, an improved version of SRGAN called ESRGAN [86] with the proposed residual-in-residual dense block (RRDB) and the relativistic average GAN (RaGAN) [87] loss, outperformed SRGAN in terms of perceptual quality.

SR techniques have been widely used in various fields, however, only limited applications in the agricultural sector have been reported thus far [88–91]. Kasturiwala

and Aladhake [88] applied an iterative curvature-based interpolation method [92] to increase the resolution of diseased leaf images. They claimed this approach could support pathologists with better visual quality of the infected leaves, but have not yet tested their method on any disease diagnosis tasks. Yamamoto et al. [89] and Dai et al. [91] improved disease diagnostic performance by applying an SRCNN and a GAN-based SR model called DATFGAN to tomatoes and other types of crops, respectively.

Although these methods showed promising results, they are not very realistic, since they were applied to the impractical PlantVillage dataset [15] in which each leaf image was taken under ideal conditions (e.g., manually cropped and placed on a uniform background). Several reports have shown that the diagnostic performance of systems trained on these images is significantly reduced when applied to real on-site images [14, 34]. Hence, we cannot conclude from their results that their diagnostic schemes can be used in practice, or that SR contributes to improving diagnostic accuracy in practical situations.

In terms of the effectiveness of SR in practice, our previous GAN-based SR model [90] with perceptual loss [93] dramatically helped to improve the diagnostic performance under in-field LR cucumber images. The diagnostic result from our SR model was 20.7% better than the baseline which was the state-of-the-art SRResNet model [85] at that time. However, we experienced that SR images generated using GAN often contain artifacts like "rubber stamps", especially in the leaf region (see Fig. 3.1). Leaves have been most frequently targeted in the study of automatic diagnosis of plants, since they exhibit more disease characteristics than other parts of the plant. We are particularly mindful of the problem of artifacts occurring in the leaf region, as this could cause difficulty in diagnosing some types of disease in practical situations. To address this problem, we propose an effective artifact-suppression SR method specifically designed for leaves, called leaf artifact-suppression super-resolution (LASSR).

One further aspect of this proposal should be emphasized. In the field of automated

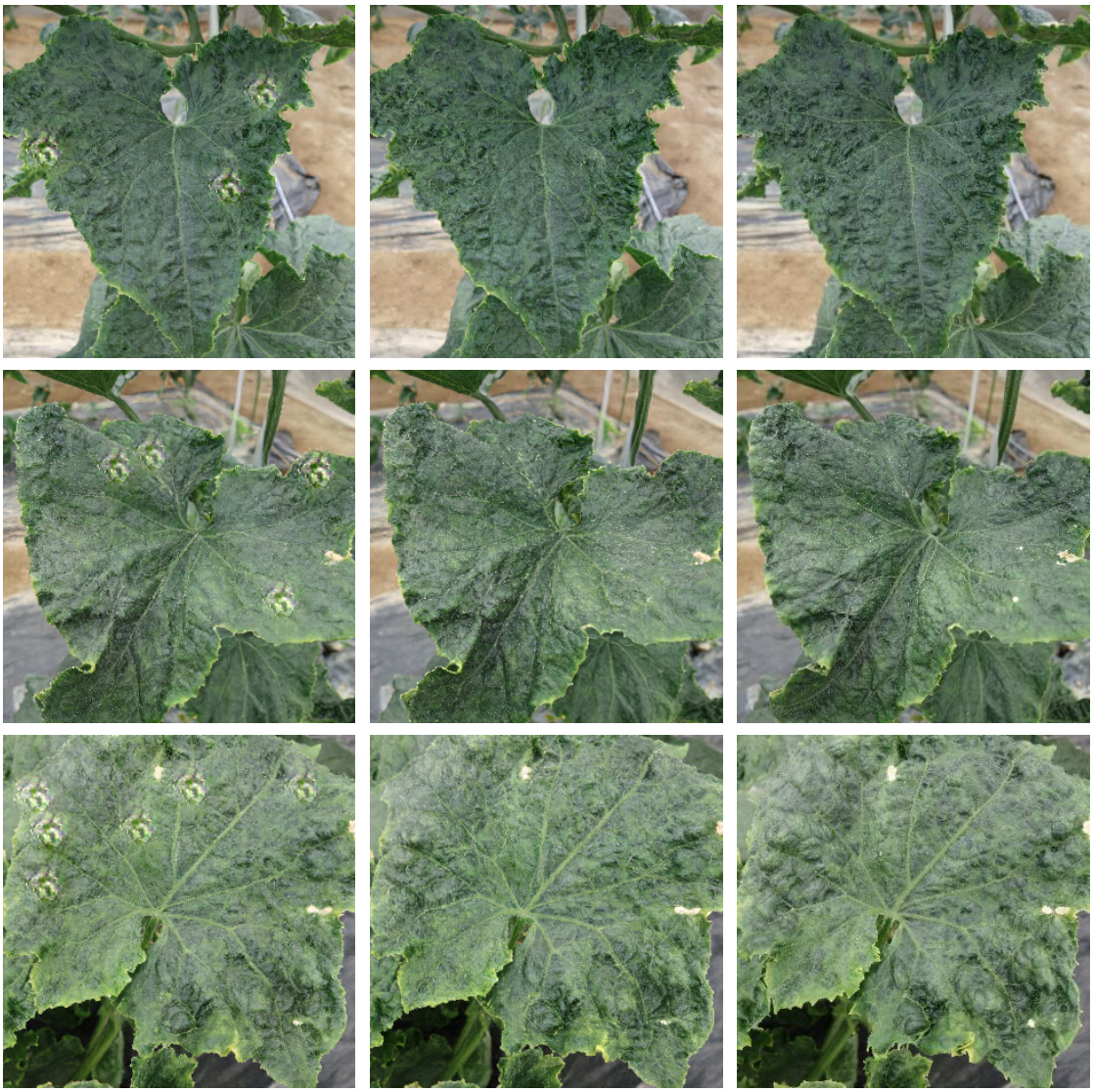


Figure 3.1: Comparison of SR methods for leaf regions (4× up-sampling): (a) ESRGAN; (b) the proposed LASSR; and (c) the original HR image. The artifacts in ESRGAN could obscure the symptoms of disease and make it difficult to detect certain types of disease.

plant diagnosis, evaluation data are in most cases generated from a portion of the training dataset. Recently, it has been noted that the reported diagnostic accuracy is likely to have been superficially overstated due to latent similarities within the dataset (i.e., similarities in image conditions such as background and light conditions) [14, 34, 63–65]. We therefore evaluate the performance of our LASSR method based on the degree of improvement in diagnostic accuracy for a completely exclusive dataset, in

addition to the image quality.

In this study, we found that the larger the size of the image input to the model, the better the diagnostic accuracy achieved on an *unseen dataset*. However, HR training resources are not always available in practice. We therefore believe that SR methods can be used to generate high-quality training resources and can help to improve the robustness of disease diagnosis systems on unknown test data to allow for more practical use.

In summary, the contributions of this work are as follows:

- We propose LASSR as a specially designed SR method to improve the performance of plant leaf disease diagnosis, with a novel artifact removal module (ARM) that dynamically suppresses artifacts on-the-fly during training.
- LASSR provides visually pleasing images by effectively suppressing artifacts and gives a better Fréchet inception distance (FID) [94] than the ESRGAN method.
- LASSR significantly improves the accuracy of plant diagnosis in unseen images by over 21% from the baseline. This is more than 2% better than a model trained on images generated by ESRGAN.

3.1 Materials and methods

3.1.1 Image datasets

In this work, we used two datasets, Dataset-A and Dataset-B. Dataset-A was used to train and evaluate both LASSR and the ESRGAN model as a comparison. Dataset-B was used to train and evaluate the classifiers, in order to assess the effectiveness of SR in disease diagnosis. These datasets are independent of each other.

Table 3.1: Statistics of Dataset-B

Class	Dataset-B_{Train}	Dataset-B_{Val}	Dataset-B_{Test}
Healthy	13,089	4,394	1,276
Brown spot	5,142	1,668	2,786
CCYV	4,356	1,438	2,096
MYSV	10,451	3,512	1,550
Downy mildew	2,514	893	2,219
Total	35,552	11,905	9,927

Dataset-A for SR models

We used a cucumber dataset previously reported in the literature [30, 90] as Dataset-A. This is a multiple infection dataset with 25 classes containing a total of 48,311 cucumber leaf images, of which 38,821 show single infections, 1,814 show multiple infections, and 7,676 contain healthy leaves. Each image had a size of 316×316 pixels (cucumber leaf size is roughly 20-25cm, so pixel size is 0.63-0.79 mm/pixel in this case). We divided this dataset into separate training and testing datasets. The training set contained 36,233 images (roughly 75% of the dataset, referred to here as Dataset-A_{Train}), while the testing set contained 12,078 images (roughly 25% of the dataset, referred to as Dataset-A_{Test}).

Dataset-B for disease classifiers

Dataset-B was another cucumber leaf dataset collected from multiple locations in Japan, taken during the period 2015–2019. Table 3.1 summarizes the statistics for this dataset. It contains four classes of disease (*Cucurbit chlorotic yellows virus* (CCYV), *Melon yellow spot virus* (MYSV), *Brown spot*, *Downy mildew*) and healthy. Each image in this dataset had a size of 512×512 pixels (cucumber leaf size is roughly 20-25cm, so pixel size is 0.39-0.49 mm/pixel in this case). We divided this dataset into two sets, Dataset-$B_{Train/Val}$ and Dataset-B_{Test}. Images in Dataset-B_{Test} were taken at different times and in different locations from those in Dataset-$B_{Train/Val}$ in order to avoid the problem of latent similarities among datasets, as mentioned earlier. Note

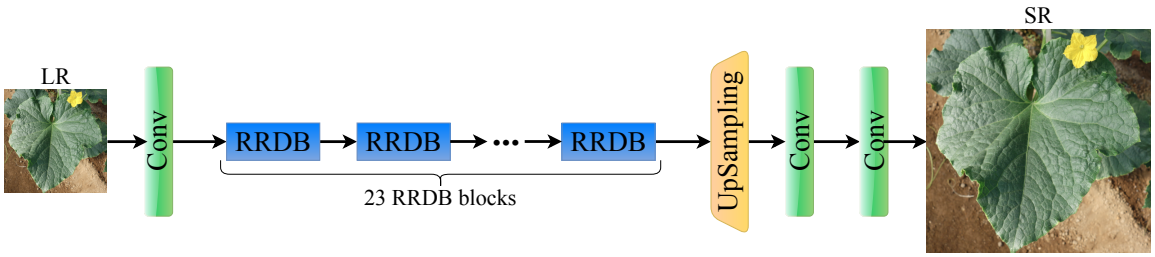


Figure 3.2: The generator G consists of 23 RRDB blocks, followed by 4× upsampling and convolutional layers to generate the SR images.

that the appearances of the images in these sets varied widely, due to the differences in the circumstances (i.e., photographic conditions and background) in which they were taken.

3.1.2 Proposed method - LASSR

The proposed leaf artifact-suppression super-resolution (LASSR) is a SISR framework that is specifically designed to solve the problem of artifacts in SR images and hence to improve the performance of plant disease diagnosis. LASSR is inherited from our previous GAN-based network [90]. It is basically built on ESRGAN [86] with the proposed artifact removal module (ARM) to guide the network in suppressing artifacts on-the-fly during training. Fig. 3.1 shows examples of the artifacts generated by the ESRGAN model; our proposed LASSR resolves this problem, resulting in natural and convincing generated images.

LASSR is composed of two CNN models: the generator G, which generates SR images, and the discriminator D, which distinguishes SR images from HR images. The networks are trained together to solve an adversarial min-max problem.

The generator

Our scheme uses the architecture of a generator G in the same way as in ESRGAN [86]. G is composed of 23 residual-in-residual dense (RRDB) blocks, resulting in a total of 345 convolutional layers. Our network G up-scales 4× from the input LR image.

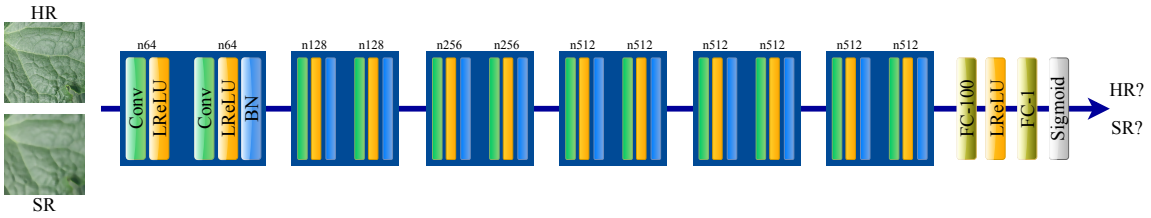


Figure 3.3: Discriminator D, consisting of six *conv_blocks* with corresponding number of feature maps (n).

Fig. 3.2 illustrates the architecture of the generator G used in our experiments. More technical details can be found in the ESRGAN paper.

The discriminator

Our discriminator D is designed in the same way as in our previous model [90]. It is deeper than the discriminator used in ESRGAN, and has a larger input size of 192×192. The architecture of our discriminator D is illustrated in Fig. 3.3. We define a convolution block as a block of two convolutional (Conv) layers. Following a Conv, we use either a leaky rectified linear function (LReLU) [95] or a combination of LReLU and batch normalization (BN) [96]. We use LReLU with $\alpha = 0.2$ as the activation function for all layers except for the last. More details are described in our previous paper [90].

Loss functions of LASSR

The objective functions in LASSR are extended from ESRGAN. To train our generator G, we minimize the loss function $\mathcal{L}_{\mathrm{G}}$ as follows:

$$\mathcal{L}_{\mathrm{G}} = \lambda \mathcal{L}_{\mathrm{G}}^{\mathrm{adv}} + \mathcal{L}_{\mathrm{percep}} + \eta |I_{\mathrm{HR}} - I_{\mathrm{SR}}|_1 + \beta \mathcal{L}_{\mathrm{ARM}}, \tag{3.1}$$

where $\mathcal{L}_{\mathrm{G}}^{\mathrm{adv}}$, $\mathcal{L}_{\mathrm{percep}}$, and $|I_{\mathrm{HR}} - I_{\mathrm{SR}}|_1$ appear in the original loss function in ESRGAN. Here, $\mathcal{L}_{\mathrm{G}}^{\mathrm{adv}}$ is the adversarial loss for the generator G, and $\mathcal{L}_{\mathrm{percep}}$ is the perceptual loss [93], which minimizes the similarity between the HR and SR images in the feature

space of the VGG-19 model [17] pre-trained with the ImageNet dataset [11]. I_{HR} and I_{SR} are the HR and SR images, respectively. $\mathcal{L}_{\mathrm{ARM}}$ is our proposed novel loss term for calibrating the artifact effects, and is formed based on our proposed ARM (described in detail in the next section). λ, η, and β are coefficients used to balance the different loss terms.

To train our discriminator D, we use the same adversarial loss $\mathcal{L}_{\mathrm{D}}$ as in ESRGAN:

$$\mathcal{L}_{\mathrm{D}} = -\mathbb{E}_{I_{\mathrm{HR}}}[\log(D(I_{\mathrm{HR}}, I_{\mathrm{SR}}))] - \mathbb{E}_{I_{\mathrm{SR}}}[\log(1 - D(I_{\mathrm{SR}}, I_{\mathrm{HR}}))]. \tag{3.2}$$

Finally, the G and D networks are trained together to solve an adversarial min-max problem [67]. More details of the loss functions can be found in the ESRGAN literature [86].

The artifact removal module

In this work, we propose a novel artifact removal module (ARM) that detects and suppress artifacts on-the-fly during the training of our SR model. The ARM acts like a dynamic training strategy, and provides guidance allowing LASSR to suppress the occurrence of artifacts. The key idea of the ARM is to detect artifacts and then to minimize the differences between the areas of these artifacts and the corresponding areas of the ground-truth HR images. In most cases, artifacts appear in the form of similar specks or "rubber stamps", as shown in Fig. 3.1. We refer to these artifact regions as blobs, and apply the difference of Gaussians (DoG) to detect them. DoG is an algorithm that finds scale-space maxima by subtracting different blurred versions of an original image. These blurred images are obtained by convolving the original images with Gaussian kernels with differing standard deviations. After obtaining the scale-space maxima, the areas of blobs are defined by the local maxima points and their corresponding Gaussian kernels.

Fig. 3.4 illustrates the steps in the process of detecting artifact areas (blobs) in

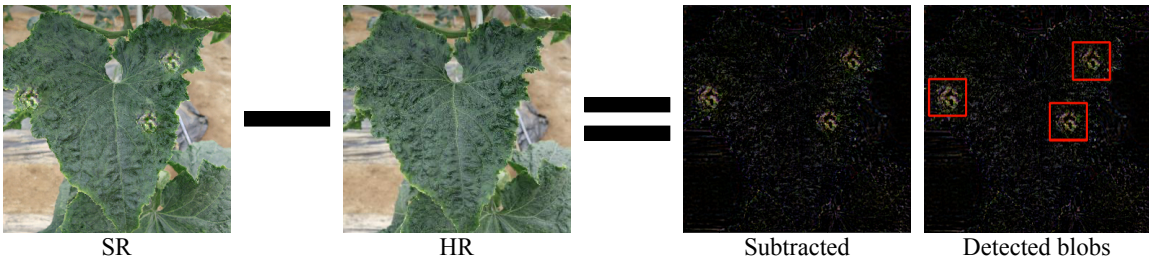


Figure 3.4: SR and ground-truth HR images of a leaf. After obtaining the subtracted image, the blobs can be detected by applying DoG.

SR images. Given a SR image and a ground-truth HR image, we first subtract two images to obtain the subtracted version. DoG is then applied to detect the locations of artifacts (blobs). Based on our preliminary experiments, artifact areas can be detected effectively using only two Gaussian kernels with corresponding values of $\sigma_1 = 0.078 \times N$ and $\sigma_2 = 0.104 \times N$, where $N \times N$ is the size of the cropped image used for training (i.e., 192×192 in our study).

Once the artifacts have been detected, we add a new loss term $\mathcal{L}_{\text{ARM}}$ in Eq. 3.1 to LASSR to allow the network to suppress the artifact. $\mathcal{L}_{\text{ARM}}$ is defined as:

$$\mathcal{L}_{\text{ARM}} = \sum_{b_i \in \mathfrak{B}} b_i, \tag{3.3}$$

where b_i is the sum of the pixel values of a blob detected in the subtracted image, and $\mathfrak{B} = \{b_1, b_2, ..., b_{|\mathfrak{B}|}\}$ is the set of detected blobs.

3.2 Experiments and results

We conduct two experiments. The first experiment is for comparing our LASSR with ESRGAN by evaluating the quality of SR images. The second experiment evaluates the improvement in diagnostic performance among disease classifiers which used images obtained by SR techniques as training data. This evaluation is used due to the difficulty in obtaining HR images, which usually provides high diagnostic accuracy, as mentioned

earlier.

3.2.1 Implementation details

Training SR models

We trained our LASSR and ESRGAN using Dataset-A_{Train}. During training, HR images were obtained by random cropping from training images with the size of 192×192. LR images were then created by $1/4\times$ down-sampling from HR images, using bicubic interpolation. Both LR and HR images were augmented with random horizontal flips and random 90 degrees rotations on-the-fly. It should be noted that these are the same conditions that were used in the original training of ESRGAN.

The ESRGAN model was trained using the loss functions from the original paper, while our LASSR was trained using the loss functions in Eqs. 3.1 and 3.2, with $\lambda = \beta = 5 \times 10^{-3}$ and $\eta = 10^{-2}$. We set the mini-batch size to 32 images and used the Adam optimizer [97] with the learning rate of 10^{-3} for both G and D models. The training process was completed after 400 epochs.

Training disease classifiers

In this experiment, we trained similar plant disease classifiers using LR, HR and SR generated images from both LASSR and ESRGAN for comparison. Specifically, we trained the following five classifiers based on the pre-trained EfficientNet-B4 [25] model:

1. model_LR: The classifier was trained on a $1/4\times$ down-sampled Dataset-B_{Train} (i.e., input size 128×128) using bicubic interpolation.

2. model_HR: The classifier was trained with Dataset-B_{Train} (i.e., input size 512×512).

3. model_Bicubic: As in (1), except with training images $4\times$ SRed (i.e., 512×512) using bicubic interpolation.

4. model_ESRGAN: As in (1), except with training images 4× SRed (i.e., 512×512) using ESRGAN.

5. model_LASSR [proposed model]: As in (1), except with training images 4× SRed (i.e., 512 × 512) using the proposed LASSR.

All five EfficientNet-B4-based classifiers were fine-tuned at all layers using the Adam optimizer [97]. The mini-batch size was set to 32 and the learning rate was 10^{-3}. To handle the class imbalance in Dataset-B_{Train}, we applied the softmax class-balanced loss [98] with $\beta = 0.9999$ to all classifiers. During the training process, we applied random horizontal and vertical flips to each image. Training was complete after 20 epochs.

3.2.2 Evaluation of image quality

Figs. 3.5 and 3.6 show the visual comparison and line profiles of the generated images and the original HR image. Our LASSR successfully suppressed the artifacts, and generated images that were more natural than the ESRGAN method. We also observed that for Dataset-A_{Test}, ESRGAN produced over 1,300 artifact images (11.39% of the dataset), while our LASSR created only 177 cases (1.47%). On Dataset-B, which was exclusive of Dataset-A, ESRGAN still produced 380 cases of artifacts (0.66%) while our LASSR generated zero artifact (artifact-free) images. To quantitatively evaluate the image quality, we used Fréchet inception distance (FID) [94] as our evaluation criteria, in the same way as in other SR methods. This is because other standard quantitative measures such as PSNR and SSIM have been reported as being unable to capture and accurately assess the perceptual image quality which is highly correlated with human perception [85, 99, 100]. Table 3.2 shows the FID scores of the images generated using the bicubic, ESRGAN and LASSR techniques versus the original HR images. In a similar way as for Dataset-A_{Test}, we calculated the FID score for the entire Dataset-B, since it is exclusive from Dataset-A_{Train}. Note that the test images were 1/4× down-

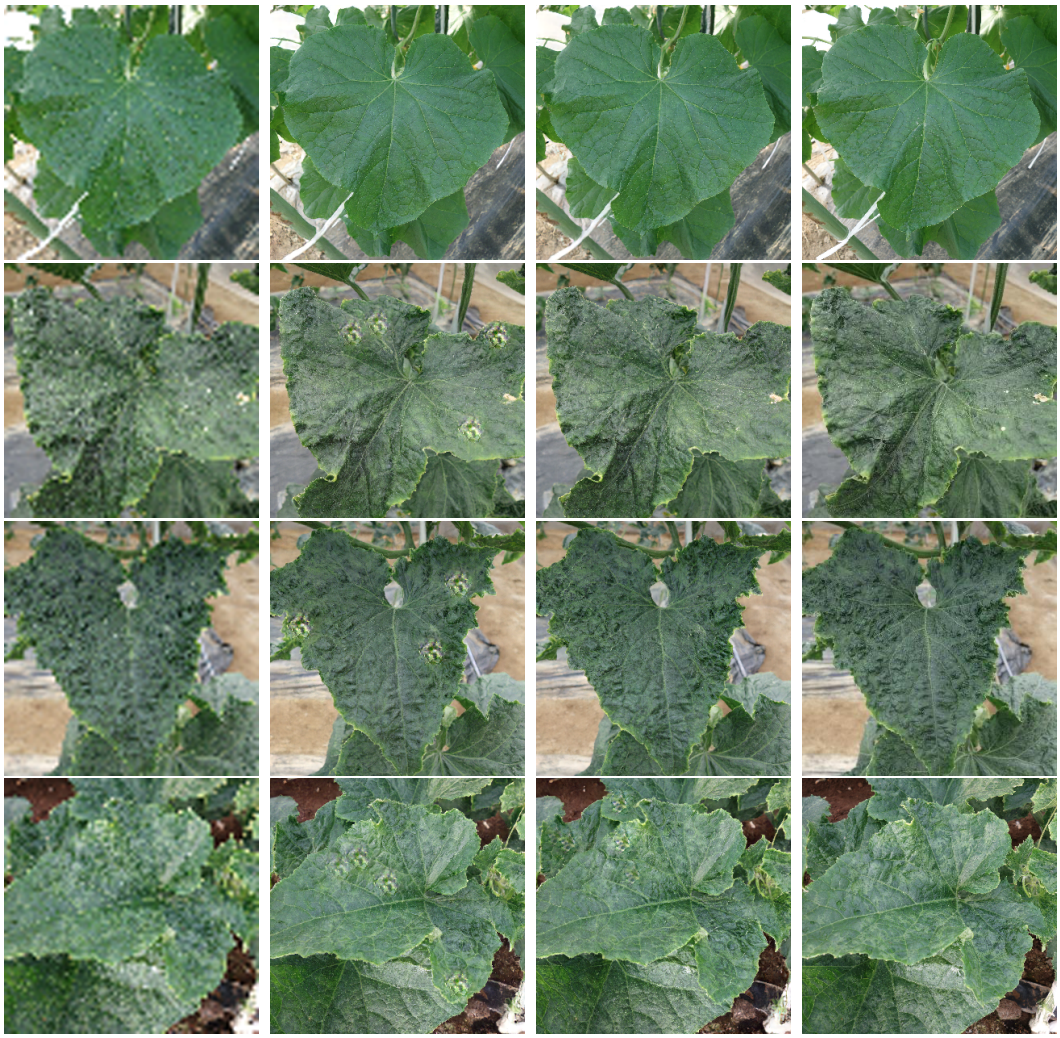


Figure 3.5: Comparison between the generated SR results and the original HR images. LASSR generates more a natural image with very suppressed artifacts compared to the existing ESRGAN method.

Table 3.2: FID [94] scores for bicubic, ESRGAN, LASSR, and HR (lower is better)

Dataset	(Bicubic, HR)	(ESRGAN, HR)	(LASSR, HR)
Dataset-A^{*}_{Test}	104.49	2.98	**2.90**
Dataset-B**	45.36	2.42	**2.38**

*,** : Calculated for images of size 316×316 and 512×512, respectively

sampled before being fed to the SR models. Our LASSR achieved better (lower) FID scores than ESRGAN on both datasets.

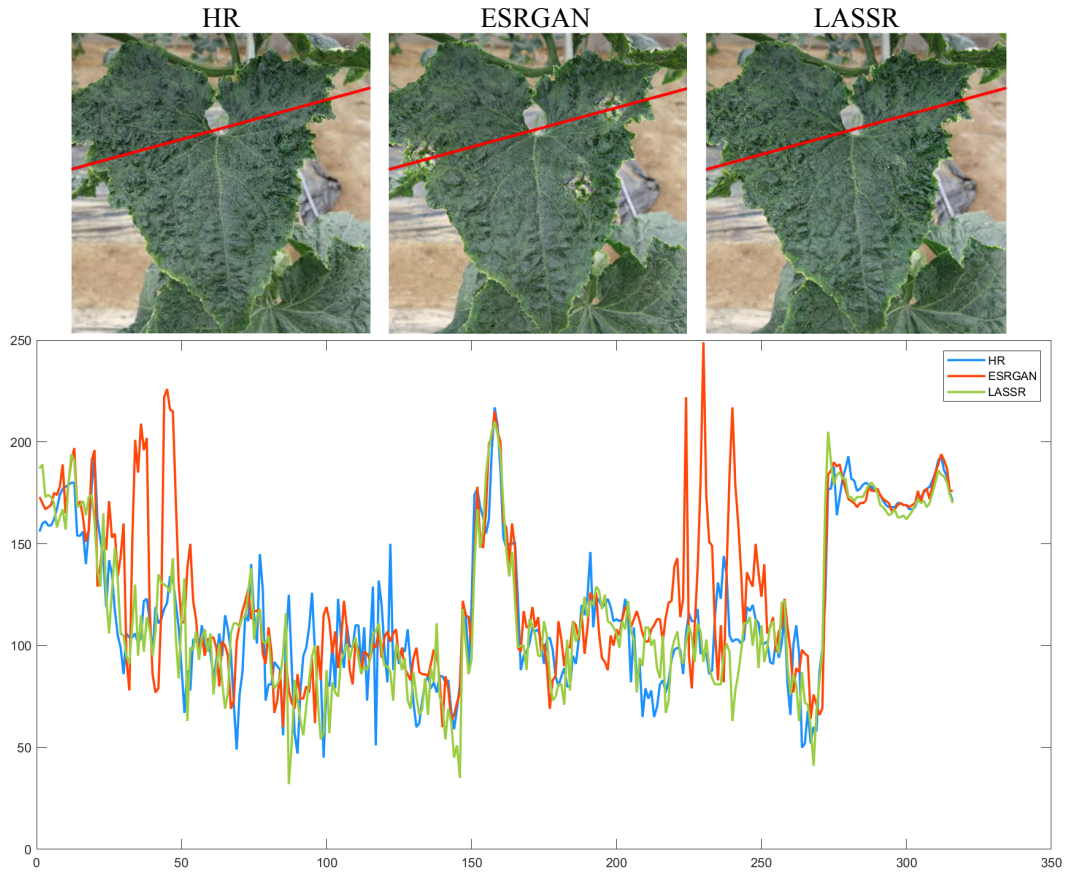


Figure 3.6: Line profiles of the SR and the original HR images. We can see similar profiles of LASSR and HR, while that of ESRGAN shows significant dissimilarities due to the presence of artifacts.

3.2.3 Comparison of diagnostic performance on an unseen dataset

Table 3.3 presents a comparison of the diagnostic performance of disease classifiers from the second experiment. We can see that there is a large gap between the micro-average accuracy of the Dataset-B_{Val} and Dataset-B_{Test}, since the two sets are inherently different in nature due to differences in the photographed locations of the images and other factors as mentioned earlier. However, our LASSR model significantly helped boosting the performance on the unseen dataset from the baseline model, achieving a competitive result.

Table 3.3: Summary of results on disease classification with different training images

a) Classification performance (in micro-average accuracy) on Dataset-B

Model	Dataset-B_{Train}	Dataset-B_{Val}	Dataset-B_{Test}
model_LR	95.20	93.77	64.94
model_Bicubic	98.27	97.35	72.81
model_ESRGAN	96.36	95.95	83.53
model_LASSR	98.32	97.42	86.00
model_HR	**99.77**	**99.17**	**91.71**

b) Classification performance (in accuracy) on Dataset-B_{Test}

Model	**Healthy**	**Brown spot**	**CCYV**	**MYSV**	**Downy mildew**	**Macro-average**
model_LR	94.44	85.28	41.79	59.16	48.36	65.81
model_Bicubic	**97.26**	85.18	67.70	63.29	54.71	73.63
model_ESRGAN	94.12	**92.53**	79.25	84.58	69.45	83.99
model_LASSR	89.42	**93.72**	**84.88**	**87.81**	**74.13**	**85.99**
model_HR	**97.57**	90.09	**94.13**	**89.03**	**89.95**	**92.16**

* The input size of model_LR is 128×128, and all other models is 512×512. **Red** indicates the best performance and **blue** indicates the second-best performance

3.3 Discussion

We proposed a SR method for addressing the artifact problem and explored its potential for improving the performance of an automated plant disease diagnosis system on unseen data.

3.3.1 Improvement in image quality

Thanks to the introduction of the novel ARM, our LASSR effectively addressed the problem of artifacts and produces more natural SR images compared to the results from ESRGAN (see Figs. 3.1, 3.5 and 3.6), and achieves better FID scores (Table 3.2). Although some artifact cases remained in the images generated by LASSR on Dataset-A_{Test}, it should be noted that our LASSR produced much weaker artifacts than ESRGAN (see Fig. 3.5, last row). Moreover, LASSR generated artifact-free SR images on Dataset-B.

The incidence of ESRGAN artifacts was 11.4% for Dataset-A_{Test}, which consisted of images taken in the same field as the training data. The rate was far lower for Dataset-B at 0.66%, on the completely unknown dataset. In addition, the artifacts generated by ESRGAN were almost all of the same size (where each side of the boundary box was about 1/10th the size of the image) and had circle-like shapes, as if the image had been stamped with a rubber stamp. This may be due to the particular combination of the input image and kernels in a relatively forward convolutional layer corresponding to a specific receptive field size. However, we have not been able to identify the reason for this. Although our ARM effectively suppressed and reduced these artifact effects, it seems likely that further investigation of the behavior of the convolutional layers could be a way to be toward the artifact-free SR, and we aim to address this in the near future.

3.3.2 Improvement in disease diagnosis performance

From the results in Table 3.3, although the diagnostic accuracy of all of the classifiers on the two subsets Dataset-$B_{Train/Val}$ was similar and very high, the performance was significantly reduced on Dataset-B_{Test}. As discussed above, this is due to the problem of latent similarities between data [14, 34, 63–65]. As frequently reported in the literature, the performance was higher for data from the same imaging environment (Dataset-$B_{Train/Val}$) due to overfitting, but lower for data from a different environment (Dataset-B_{Test}). This reduction in performance is generally known as a covariate shift.

The model_LR and model_Bicubic models showed poor performance on Dataset-B_{Test} (with micro-average accuracies of around 65% and 73%, respectively, as shown in Table 3.3.a). This was because the size of the LR input images was not sufficient for the diagnosis of plant diseases. Our LASSR scheme successfully reconstructed the information from LR images and helped the model_LASSR model to achieve high diagnostic performance, significantly outperforming the model trained on LR images (model_LR)

by over 21% in terms of micro-average accuracy (Table 3.3.a). Moreover, model_LASSR achieved performance that was 2.47% and 2% better than model_ESRGAN in terms of micro- and macro-average accuracy, respectively (Table 3.3).

For the healthy plants, model_LASSR (89.42%) was less numerically accurate than all other classifiers. However, in all other disease cases, our model_LASSR performed significantly better than model_ESRGAN, achieving the closest result to the model using the original HR images. To explain this, our LASSR successfully helped recover the HR components of symptoms and largely increased the performance for diseases, while it slightly increases false positives (i.e., slightly decreased performance on healthy case). Despite this fact, the effects of eliminating false negatives of LASSR is much larger than a little increment of false positives.

Above results reinforce our argument for the benefits of using the SR method. Since high-quality training data are not always available in practice, SR techniques are confirmed to be effective in generating reliable training resources and improving the robustness of diagnosis systems. Note that we also trained another classifier with HR images of size 224×224 (as commonly used in many disease classifiers) and recorded a diagnostic of around 80% in terms of micro-accuracy. This implies that the quality of the training image is important.

Although LASSR achieved promising results, we note that the selection of hyper-parameters for the ARM is still done manually, as it depends on the input size. We believe that the development of an ARM that dynamically adapts the training set will further improve the effectiveness of LASSR.

Chapter 4

Effective data augmentation method for plant disease diagnosis

In recent years, deep learning has revolutionized the field of computer vision, and is now becoming a standard tool for many applications. Many deep learning-based techniques for the automated diagnosis of plant disease have been developed with the aim of supporting farmers and reducing losses in terms of plant productivity [27, 14, 39, 32, 12, 34, 24, 38, 26, 36].

Despite the success of the above methods, several essential problems still remain. *Firstly*, deep learning-based systems need a huge number of training images. Unlike other general computer vision tasks, labeling disease datasets requires solid biological knowledge. Moreover, in order to collect gold standard datasets of diseases, the plants must be grown in a strictly controlled and isolated environment to avoid contamination, which is generally labor-intensive and very expensive. *Secondly*, practical plant disease datasets are often imbalanced. Although the target plants are grown in a tightly controlled environment as described above, disease development is also strongly influenced by ambient conditions such as weather, temperature and vector-borne insects. Therefore, several diseases are difficult to collect, and the obtained datasets often have imbalanced amount on each class. Although several techniques have been proposed to address this data imbalance problem [52, 98], disease classification models are generally biased toward classes with more samples and higher variation [101]. *Thirdly*, the overfitting problem is particularly serious in plant diagnosis tasks, since the image features that provide diagnostic clues (i.e., evidence for classification) are typically much smaller than in general object recognition problems. Particularly in early-stage cases,

the clues for diagnosis may consist only of a tiny dot or faint wrinkles in the image. This is the main problem that is going to be addressed in this study. Image-based plant diagnosis is a particularly difficult task due to the fine-grained object recognition required. In general, a deep classifier such as a CNN tends to capture the image characteristics (i.e., brightness, color) of a large area, rather than a faint feature that may indicate disease. In addition, when evaluating a classifier using a dataset divided into training, validation, and test sets (where cross-validation is applied), the latent similarities within the dataset (such as the background, brightness and/or distance between target and camera etc.) works as a positive bias, and generally improves only the superficial diagnostic accuracy, while the accuracy when evaluated on other unknown environments becomes very low [14, 34, 63–65]. For example, in the cucumber disease diagnosis from wide-angle images, the diagnostic performance on the same farm showed 86.0% in F1-score, but it dropped to 20.7% on a different farm [63]. Other evidence confirming the overfitting of models in plant diagnosis tasks has been shown in our previous studies [29, 64] by using Grad-CAM [66] to visualize the key regions of diagnostic evidence. Although these models provided a high diagnosis accuracy of over 90% on this dataset, the backgrounds were sometimes considered as diagnostic regions.

The most plausible reason for this is that when collecting a dataset, the foreground objects in each image class tend to be incidentally correlated with similar backgrounds. A lack of background diversity could be a distractor, meaning that the model sometimes responds to the background rather than discriminative targets (i.e., leaf regions). One possible solution for this is to remove the background from the RoI as in our proposal anti-overfitting pretreatment (AOP) network [64]. The network segments the leaf areas before training disease classifiers, in order to reduce the negative impact of the background in terms of causing overfitting. We confirmed that our AOP significantly improved the classification performance in a practical setting. However, this approach

requires a large amount of expensive masking data and may eliminate surrounding information that is important for diagnosis (e.g., the lighting conditions of the picture, indicators of infection). Furthermore, we believe that the latent similarities within the dataset such as brightness, lighting, and/or distance between target and camera etc. still remains even on the segmented images and could cause difficulties for diagnosing on unseen data.

In general, the background diversity of disease images tends to be limited, especially when plants are grown in a controlled environment to ensure the quality of training labels. However, collecting healthy images is relatively easy. In these situations, we can assume that if we could transform the wide variety of healthy images (including backgrounds) into disease cases, we could build a more divergent and reliable disease dataset. As a result, we expect to both improve the performance of diagnosis and to reduce the cost of labeling.

Recently, an excellent image-to-image translation method called CycleGAN [102] has been shown to have outstanding performance and has become a standard method of generating appealing images. CycleGAN removes the need for paired label training data by introducing the cycle-consistency loss, based on the assumption that the image generated from the source domain should be able to be transformed back to its original form.

Based on the superiority of CycleGAN, several methods have been developed for application in the field of plant science. Tian et al. [69] applied CycleGAN as a data augmentation method to generate more data on diseased apples to train their apple lesion detection system. However, since CycleGAN generates images that are close to the distribution of the original training data, the effect of adding these generated images to the training set was limited. In addition, because the original CycleGAN itself has no explicit attention mechanism, it tends to transform the entire image from the source to the target domain, rather than transforming the specific objects (i.e., the

apple in this case). As a result, a significant number of the generated images are of low quality.

Nazki et al. [70] improved CycleGAN by introducing an additional perceptual loss [93] in order to generate more natural images. Their model so-called AR-GAN transformed healthy tomato leaves into six different kinds of disease, and they claimed that their proposal could significantly improve disease classification performance compared to other classical data augmentation techniques. However, AR-GAN was trained on tomato images which have no complex backgrounds (i.e., almost entire area of each image is tomato leaves) and based on our preliminary experiments, it mostly failed to transform the symptoms on the images which include practical backgrounds like ours. Moreover, their disease classifier was tested on a dataset that was split from the same population as the training dataset, the results must be biased due to the latent similarities among the datasets as mentioned earlier. Therefore, no essential results have been confirmed.

In order to overcome these limitations and achieve a practical method of image augmentation, we propose an image-to-image translation system named LeafGAN for generating images of leaves from diseased plants. LeafGAN determines the area of the image that is relevant for diagnosis, and translates only that area from the source to the target domain. The key idea is to develop a segmentation module that segments the area of interest (i.e., the leaf region) from the background, and which can help in guiding our LeafGAN model to pay attention to the RoIs. Similar to our study, there have been studies to improve CycleGAN by introducing the attention mechanism [103–105]. All of those studies added an attention network to each generator in CycleGAN and produce attention maps to guide the generator transforming the most discriminative regions between the source and target domains only. The attention networks are then trained simultaneously with CycleGAN model. Different from their works where those attention networks are sensitive to initialization and require careful care in training,

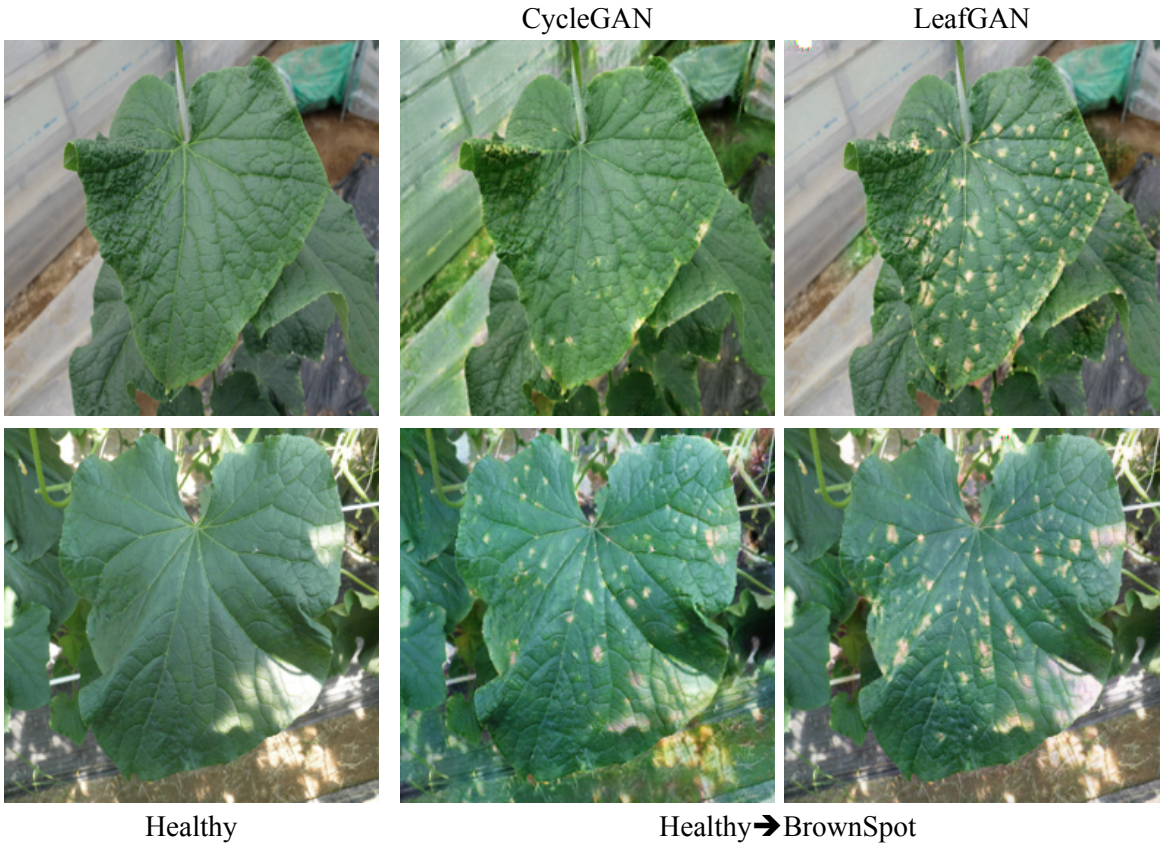


Figure 4.1: Comparison of the original CycleGAN and our LeafGAN to transform images of healthy plants to diseased ones (brown spot in this case). The CycleGAN model transforms not only the leaf regions but also the background, and as a result, the generated images have an unrealistic quality compared to the proposed LeafGAN method.

our segmentation module can be trained very quickly, and easily to achieve effective segmentation results. Moreover, our segmentation module is trained separately, and we use only one segmentation network for both generators in our LeafGAN.

We observe that LeafGAN not only generates high-quality images compared to CycleGAN, but also makes disease diagnosis systems more robust against unseen data by adding these generated images as training resources. Our contributions can be summarized as follows:

- We propose the LeafGAN model for practical plant disease diagnosis. This is an effective and easy-to-implement data augmentation tool that generates natural, high-quality disease images from healthy images while preserving a wide variety

of backgrounds.

- As a key module of LeafGAN, we introduce a novel label-free leaf segmentation module called LFLSeg, composed of a weakly supervised segmentation network that learns how to segment the leaf region without the need of expensive masking data. LFLSeg provides guidance during training that helps the network to focus attention on the leaf regions for image-to-image translation in LeafGAN.

- We demonstrate the effectiveness of LeafGAN in terms of improving the generalizability of diagnostic systems. Training with the augmented data generated by our system improves the average diagnostic performance by 7.4% on different unseen images taken from other farms while generated images from CycleGAN only help improve by 0.7%. The code of LeafGAN is available publicly at `https://github.com/IyatomiLab/LeafGAN`.

4.1 Materials and methods

4.1.1 Cucumber diseases dataset

In this work, we train our LeafGAN models to generate new images of cucumber disease. We collected cucumber leaf images from multiple locations in Japan, taken during the period 2015–2019. Each image contains a single cucumber leaf, roughly in the center and against various backgrounds. These images are of healthy (H) leaves or leaves infected with one of three diseases: Melon yellow spot virus (MYSV) (M), brown spot (B), or powdery mildew (P). Table 4.1 summarizes the datasets used in our study. We divided these images into Datasets A and B. Images in these datasets were exclusive, and were taken on different farms. Dataset A was used for training and validation, and Dataset B was used to test performance. Note that the appearance of the images in those two sets varied due to the differences in the circumstances (e.g.,

Table 4.1: Details of cucumber datasets (Datasets A and B)

Class	**Dataset A**		**Dataset B**
	Training	**Validation**	**Testing**
Healthy (H)	4,000	717	1,046
MYSV (H)	4,000	745	2,034
Brown Spot (B)	2,000	784	1,220
Powdery Mildew (P)	2,000	796	89
Total	12,000	3,042	4,389

photographic conditions and background) in which they were taken.

4.1.2 Proposed method - LeafGAN

LeafGAN is an image generation network that is specially designed to mitigate the serious overfitting problem in image-based plant diagnosis tasks via the effective generation of high-quality and widely varying pseudo training images. LeafGAN is built on CycleGAN and our proposed label-free leaf segmentation module (LFLSeg) to guide the network in transforming the relevant regions (i.e., leaf areas) while preserving the backgrounds. Fig. 4.1 shows the limitations of the vanilla CycleGAN compared to LeafGAN; while CycleGAN transforms the entire image along with the background, LeafGAN focuses only on the leaf regions, resulting in natural and convincing generated images.

Similar to CycleGAN, LeafGAN has two mapping functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$ corresponding to two data domains X and Y. The training of G requires a discriminator D_Y to discriminate the generated image $G(x)$ from the real samples $y_i \in Y$. The mapping F and the corresponding discriminator D_X, which discriminates the generated image $F(y)$ from the real samples $x_i \in X$, are also trained simultaneously. We assume here that X and Y are the sets of healthy and arbitrary target disease images, respectively.

Fig. 4.2a shows an overview of the framework for LeafGAN. For the transformation $X \rightarrow Y$ (Fig. 4.2b), the proposed LFLSeg module first produces two binary masking images S_x and S_y, which represent the leaf areas from input images $x \in X$ and $y \in$

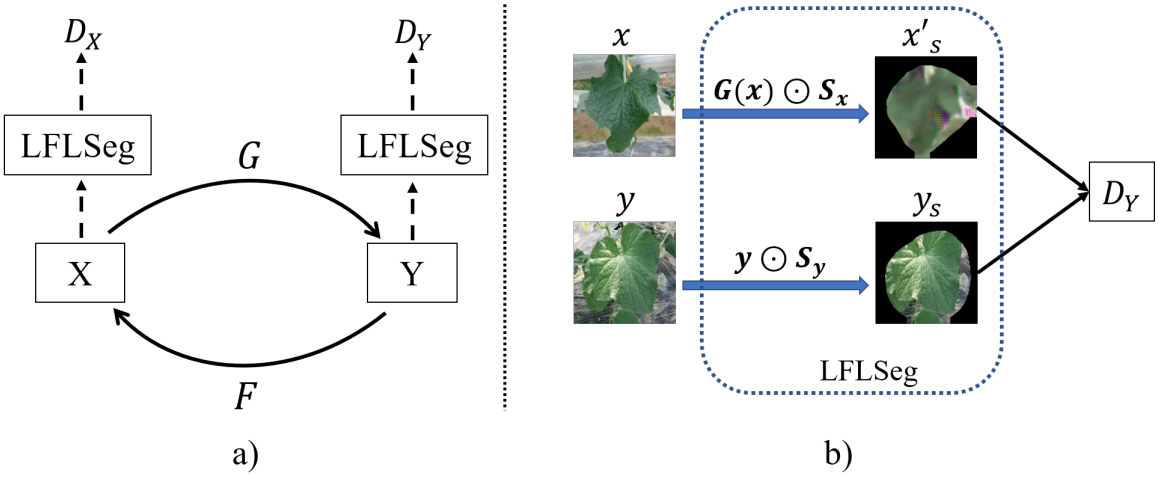


Figure 4.2: a) Overview of the proposed LeafGAN scheme; b) Dataflow when transforming the sample $x \in X$ to the domain Y. Note that the dataflow from domain Y to X is the reverse of that from X to Y. We use the same LFLSeg network in both transformations.

Y, respectively, where $S_x = \text{LFLSeg}(x)$ and $S_y = \text{LFLSeg}(y)$. After generating the image $x' = G(x)$, we obtain the masked leaf images $x'_s = S_x \odot x'$ and $y_s = S_y \odot y$, where $\odot$ denotes the element-wise product. These images x'_s and y_s are then fed into the discriminator D_Y rather than feeding x' and y. In this way, the discriminator is guided to discriminate only in terms of the leaf areas, instead of the backgrounds. Consequently, due to the adversarial training scheme for the GANs [67], the generator G is also forced to minimize its losses by paying attention to the leaf regions when generating (i.e., transforming) the images.

Note that the dataflow for the transformation $Y \rightarrow X$ is the reverse of that for $X \rightarrow Y$, since they are symmetric.

Label-free leaf segmentation module (LFLSeg)

In practice, the segmentation of in-field leaf images using conventional techniques such as thresholding, clustering, edge detection, etc. is inefficient due to the complex appearance of the leaf and the diversity of the backgrounds as well as lighting conditions. A better option involves using the power of modern deep learning-based supervised

[106–109] or weakly supervised segmentation techniques [110–113]. However, the former approach usually requires pixel-level annotation datasets in order to get a reliable result, and is therefore labor-intensive. As mentioned previously, our AOP model achieved an F1-score of 98.1% for cucumber leaf segmentation, although this score was established using 8,000 masked images for training [64]. The latter approach extracts the segmentation information from feature maps produced by a deep network trained for image classification. Although the advantage of these weakly supervised models can be trained without extra labeling data, the models are often complex and require a lot of implementation.

In this work, we propose a simple but effective weakly supervised label-free leaf segmentation module (LFLSeg) that helps the classification model to learn the dense and interior leaf regions implicitly. From an architecture point of view, the backbone of LFLSeg is a simple CNN, and is designed to discriminate between "full leaf", "partial leaf", and "non-leaf" objects. Specifically, "full leaf" objects are images that contain a single full leaf, while "partial leaf" objects are images that contain part of a "full leaf", and "non-leaf" objects do not contain any part of a leaf.

The segmented leaf region is obtained using a heatmap with respect to the "full leaf" class by applying the Grad-CAM [66] technique. This heatmap is a probability map representing the contribution of each pixel to the final decision of the "full leaf" class, and thus can be used as a binary mask after thresholding with a specific threshold value δ.

The key idea underlying LFLSeg is the introduction of the "partial leaf" class for training. As mentioned in [112], a heatmap of a classifier that is trained to discriminate between an object and its background will only cover small and most discriminative regions of the object of interest. Hence, if we train our LFLSeg only to classify "full leaf" and "non-leaf" objects, the network will not be able to cover the "full leaf" area. The introduction of a "partial leaf" class leverages the model to seek a larger leaf-

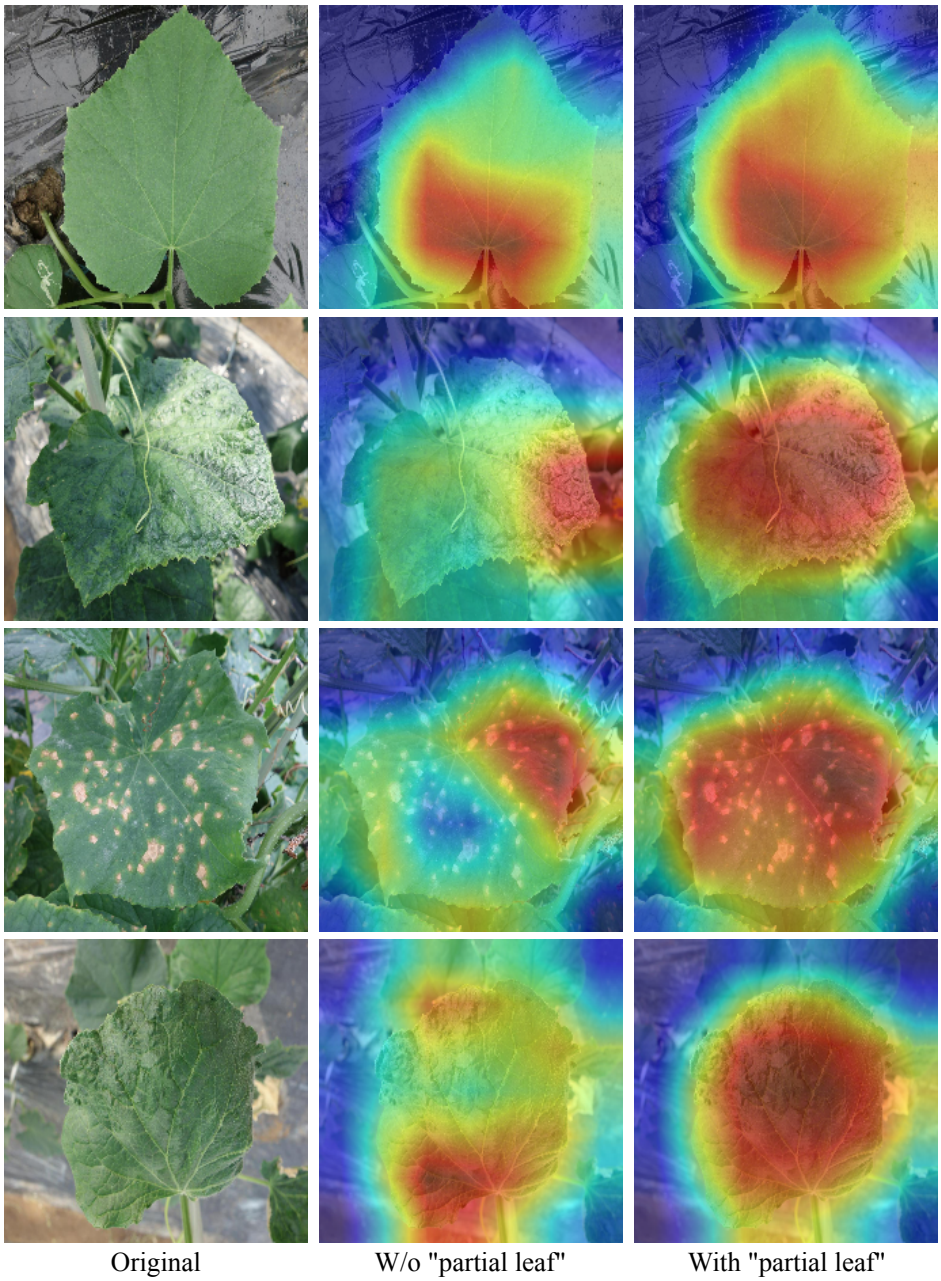


Figure 4.3: The heatmaps comparison between different classifiers trained with and without the "partial leaf" images. The warmer color region, the more it contributes to the final decision for a class (i.e., "full leaf" in this case).

shaped region in order to classify the "full leaf" image correctly.

Fig. 4.3 shows a comparison of the heatmaps between classification models with and without the "partial leaf" training data. The warmer the color of a region, the more it contributes to the final decision for the "full leaf" class. These heatmaps show

that our network, which was trained with the strategy described above, is able to focus on the whole shape of "full leaf" images, while the other model without the "partial leaf" class (i.e., which only classifies "full leaf" or "non-leaf" images) focuses on small, scattered leaf regions.

Loss functions for LeafGAN

We design the loss functions for LeafGAN with reference to CycleGAN. The adversarial losses for the two mapping functions $G : X \to Y$ and $F : Y \to X$ are expressed as:

$$\mathcal{L}_{\text{adv}}(G, D_Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[(D_Y(y_s) - 1)^2] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[(D_Y(x'_s))^2]. \tag{4.1}$$

Note again that $y_s = S_y \odot y$ is the masked version of the image $y \in Y$, where $S_y = \text{LFLSeg}(y)$ is the masking which represents the leaf area after feeding image y to the LFLSeg module. Likewise, the adversarial loss $\mathcal{L}_{\text{adv}}(F, D_X)$ for the mapping $F : Y \to X$ is defined as follows:

$$\mathcal{L}_{\text{adv}}(F, D_X) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[(D_X(x_s) - 1)^2] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[(D_X(y'_s))^2]. \tag{4.2}$$

Please note that we use the same LFLSeg to segment the inputs from both domains X and Y. The cycle consistency loss is as follows:

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[|F(G(x)) - x|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[|G(F(y)) - y|_1]. \tag{4.3}$$

Since the purpose of our study is to enrich the backgrounds of images of diseased leaves, we need to prevent generating similar backgrounds as the images in the target domain and keep the generated backgrounds as close to the original input images as possible. To meet this requirement, we introduce a new loss term called *background similarity loss* ($\mathcal{L}_{\text{bs}}$). The objective of $\mathcal{L}_{\text{bs}}$ is to minimize the L_1 distance between the

background of the generated image and the original source image. The background can be easily obtained by calculating the element-wise product between the inverted version of the mask image S (i.e., $1 - S$) and the input leaf image. Therefore,

$$\mathcal{L}_{\text{bs}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[|(1 - S_x) \odot (G(x) - x)|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[|(1 - S_y) \odot (F(y) - y)|_1]. \tag{4.4}$$

Our final objective function is:

$$\mathcal{L}(G, F, D_X, D_X) = \mathcal{L}_{\text{adv}}(G, D_Y) + \mathcal{L}_{\text{adv}}(F, D_X) + \lambda[\mathcal{L}_{\text{cyc}}(G, F) + \mathcal{L}_{\text{bs}}(G, F)], \tag{4.5}$$

where λ is a coefficient that controls the balance of different loss terms.

4.2 Experiments

4.2.1 Training the LFLSeg module

We used the fine-tuned ResNet-101 model [23] as the backbone of LFLSeg, and replaced the last layer of the network with a three-node layer. Using a deeper model (i.e., ResNet-152) yielded slightly better results, but we decided to use the ResNet-101 model for cost reasons. To train the LFLSeg module, we built datasets corresponding to the "full leaf", "partial leaf", and "non-leaf" classes. For the "full leaf" class, we used all 12,000 single leaf training images from Dataset A. During training of the network, we used a rotation with a step increment of 90 degrees and horizontal and vertical flips for data augmentation, giving a resulting dataset that was six times larger than the original one (i.e., 72,000 images).

For the "partial leaf" class, we randomly selected 8,000 images from Dataset A (the training set) and divided each image into nine equally overlapping patches (i.e., 72,000 images). Given a "full leaf" image of size N×N, we used a sliding window with size N/2×N/2 to crop a training sample for "partial leaf" class with a step size of N/4×N/4

from both the vertical and horizontal directions. In our preliminary experiments, we found that this setting showed the best performance.

For the "non-leaf" class, 72,000 images were collected randomly from the ImageNet dataset [11]. In total, the training data for LFLSeg module consisted of 216,000 images. The dataset was divided randomly, with 70% allocated as the training set and 30% as the testing set. Our LFLSeg module was fine-tuned using momentum optimization [74] with a mini-batch size of 128. The training process was terminated after 30 epochs.

4.2.2 Training the disease translation models

We used LeafGAN to build three types of healthy↔diseased translation models: (i) healthy↔MYSV (H↔M); (ii) healthy↔brownspot (H↔B); and (iii) healthy↔powderymildew (H↔P). For comparison purposes, we also built three more corresponding disease translation models using CycleGAN.

Since there were only 2,000 training samples for each of the brown spot (B) and powdery mildew (P) classes, we randomly selected 2,000 images from 4,000 images of healthy leaves (H) to train the (H↔B) and (H↔P) models (i.e., we used a total of 2,000 healthy images in this case). Note that we only used one-way translation from healthy→diseased at test time, since our target was to generate more data for diseased leaves.

We applied the same parameters as described in [102] to train both the CycleGAN and LeafGAN models. For the LeafGAN model, we set the segmentation threshold value for LFLSeg to $\delta = 0.35$. Training of both the LeafGAN and CycleGAN models was terminated after 200 epochs. Please refer to the CycleGAN article for more details of the training process.

At test time, we generated new three types of disease images from healthy images from the validation set of Dataset A (i.e., 717 images for each disease type in our experiments). These images were then used as augmented data for further training of

the disease classifiers.

4.2.3 Training the disease classification models

To carry out a qualitative evaluation of the effectiveness of LeafGAN in terms of improving the generality of disease diagnosis performance on an unseen dataset, we trained the disease diagnosis models with and without images newly generated by LeafGAN, and compared the performance in each case. Specifically, we trained the following classifiers:

- The first classifier was trained using only the training images from Dataset A. We refer to this as our baseline model.

- The second classifier was based on the above baseline model but was trained with additional disease images generated by the CycleGAN models. We refer to this as baseline+CycleGAN.

- The third classifier was similar to the second classifier, but was trained with additional disease images generated by the LeafGAN models. We refer to this as the baseline+LeafGAN. Note that this is the proposed model.

All classifiers were fine-tuned from the pre-trained ResNet-101 model, and we applied horizontal and vertical flip augmentation on the fly during training. The SGD momentum optimizer with a minibatch size of 128 was used to train these models. The training process was terminated after 30 epochs.

4.3 Results

4.3.1 Segmentation performance of LFLSeg

Our LFLseg module achieved an accuracy of 99.8% in classifying the three classes ("full leaf", "partial leaf", "non-leaf") on the validation set from Dataset A. Fig. 4.4

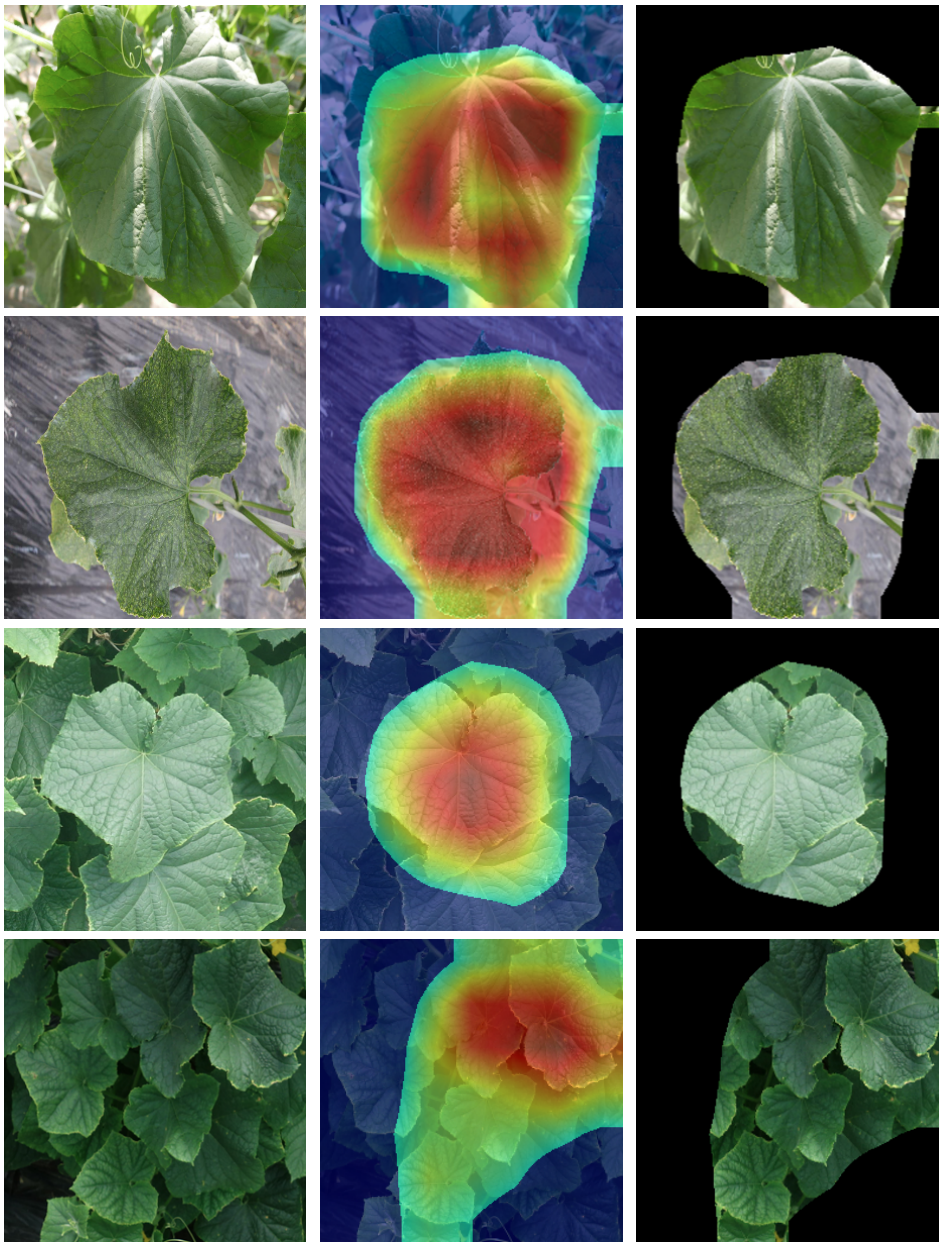


Figure 4.4: Leaf segmentation results of the LFLSeg module. The heatmaps from our network can be used as the useful segmentation masks without the need of pixel-label data.

shows several examples of leaf segmentation using our proposed LFLSeg module, with heatmaps for the "full leaf" class and their corresponding segmented results.

We confirmed that LFLSeg works well on different in-field images with complex backgrounds. However, when the images contain multiple and overlapping leaves, the

Table 4.2: Performance comparison (in accuracy) of the three classifiers in disease diagnosis on the unseen Dataset B

Class	**# of test images**	**Baseline (%)**	**Baseline+ CycleGAN (%)**	**Baseline+ LeafGAN (%)**
Healthy (H)	1,046	**85.1**	84.7	84.6
MYSV (M)	2,034	75.4	76.0	**83.3**
Brown Spot (B)	1,220	62.8	65.4	**75.9**
Powdery Mildew (P)	89	61.8	61.8	**70.8**
Average		71.3	72.0	**78.7**

LFLSeg fails to correctly segment the leaf area (Fig. 4.4 last row). Despite this fact, we do not expect the input which contains multiple leaves to be the case since we assume the input of the disease classifier is a single leaf image in this study.

We also compared the segmentation performance of our LFLSeg module with the previous AOP network [64] using 1,000 full leaf images. Our module achieved an F1-score of 83.9% while the AOP network achieved 98.1%. Even though LFLSeg showed poorer performance than the AOP network with pixel-level labeled training images, our network which requires no masking training data still achieved a reliable result that was sufficient for our task.

4.3.2 Results from disease translation models

Examples of diseased images generated by the CycleGAN and LeafGAN models are shown in Fig. 4.5. Without the explicit attention mechanism, CycleGAN tended to transform the whole area of the image, including the background, giving implausible results. In contrast, our LeafGAN models learned to pay attention to the leaf regions rather than the backgrounds, which gives more realistic disease images.

4.3.3 Improving the generality of disease diagnosis systems

The baseline, baseline+CycleGAN, and baseline+LeafGAN models obtained average accuracies of 97.2%, 97.7%, and 97.9%, respectively, on the validation images from

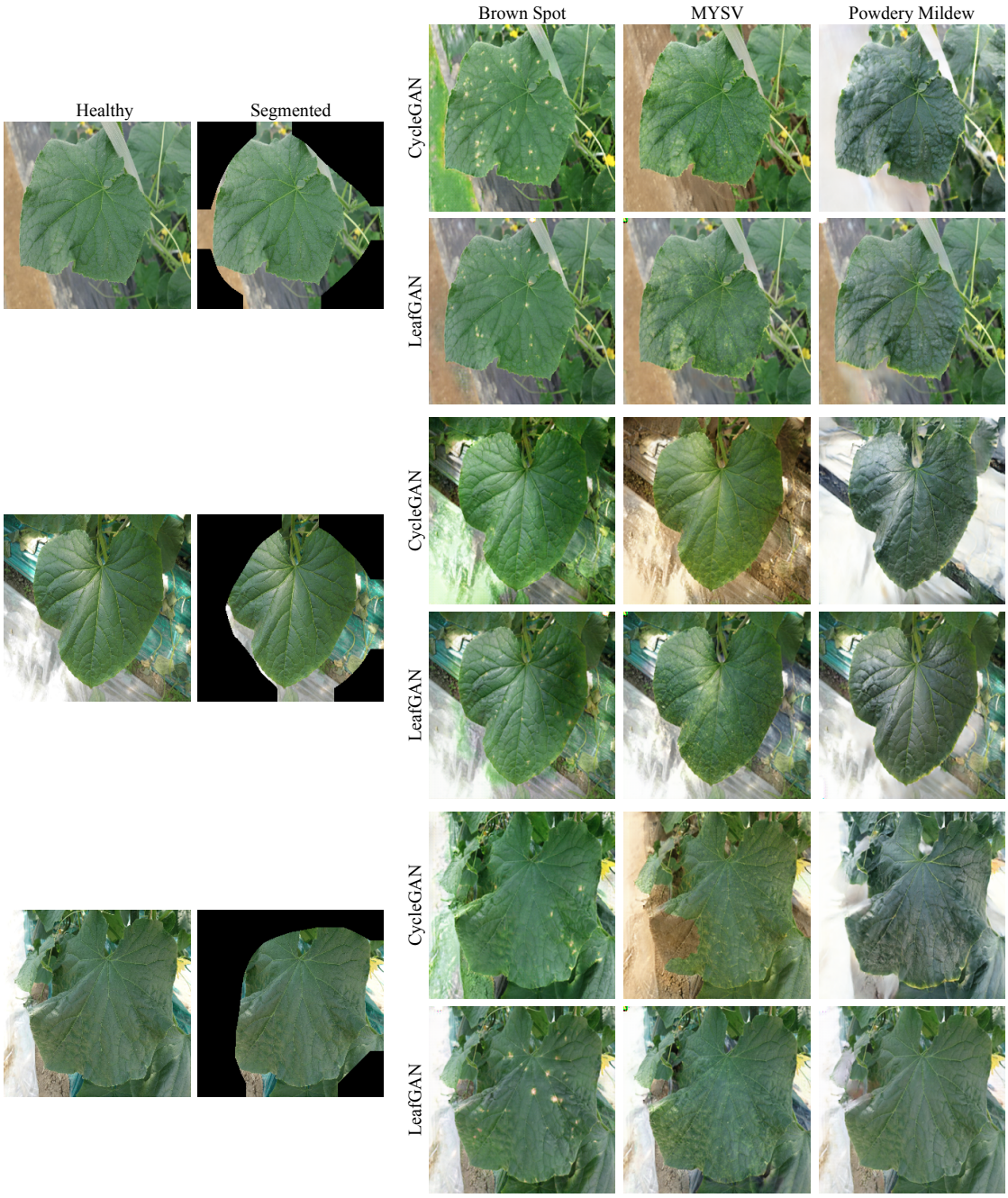


Figure 4.5: Comparison of the images generated by CycleGAN and LeafGAN. The segmented leaves are the outputs of our LFLSeg module. LeafGAN preserves the background from the original, meaning that the generated images are more realistic than those of CycleGAN.

Dataset A. We then used the trained classifiers to test Dataset B. Table 4.2 presents a comparison of the diagnostic performance of the three classifiers on the unseen Dataset B.

We can see that there is a large gap between the average accuracy for the Dataset A validation set and Dataset B, since the two sets are completely different. Although the baseline model was trained on 2,000 images per class, the average diagnostic performance only reached 71.3%. The LeafGAN models helped to boost the performance, and achieved the best result of the three classifiers with an average accuracy of classification of 78.7%.

4.4 Discussion

In this study, we investigated the effectiveness of using image-to-image translation models as a data augmentation tool to improve the performance of an automated diagnosis system for cucumber plant disease. In this experiment, the baseline model was overfitted to its training dataset and did not generalize well to the unseen samples. We also observed a large performance gap on training and testing datasets as noted in former studies [14, 34, 63–65]. The visual results in Fig. 4.5 demonstrated that our LeafGAN model could generate more persuasive and realistic images than the original CycleGAN. Since CycleGAN learns to transform the whole content of the training images, the backgrounds of the generated results appear closer to the samples from the target domain. Specifically, the backgrounds of the healthy images are transformed to be as close as possible to the images from the real disease datasets (see Fig. 4.5).

Using the proposed LFLSeg module, our LeafGAN is guided to focus on transforming only the leaf area, and can generate more compelling results. Although LFLSeg does not perfectly segment the leaf region, it is sufficiently effective to guide the LeafGAN models, thanks to the introduction of the "partial leaf" class. The results in Table 4.2 showd that using LeafGAN as a data augmentation tool could improve the diagnostic performance by 7.4% on the unseen Dataset B. This is because the probability distribution of the generated images is significantly different from that of the original training data, due to the integration of images by the segmentation mask.

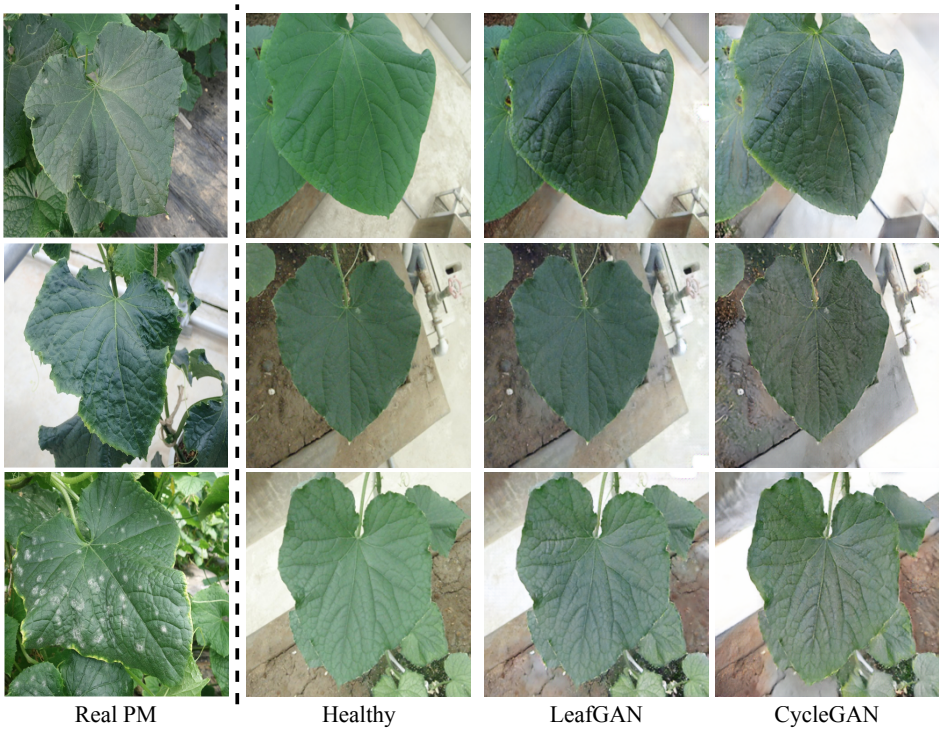


Figure 4.6: Symptoms of the PM disease in different stages (left column) and the failure cases of the H→P models from both LeafGAN and CycleGAN when translating from healthy images (second to last column).

The intrinsic variety of the training data therefore increases from a stochastic point of view. In addition, symptoms appear only in the relevant region, and we believe this is an advantage in boosting the classification performance. We believe that improving the performance of LFLSeg or combining LeafGAN with a sophisticated segmentation system such as AOP could improve the quality of the generated images.

The results from the baseline+CycleGAN model showed that if we simply trained the disease classifier with the generated images from the CycleGAN models, which have no attention mechanism, the diagnostic performance improved only slightly (+0.7%) compared to the baseline model. This is because the disease images generated by CycleGAN are intended to have a close probabilistic distribution to the training disease data, and thus the variety of training data is increased only a little. This is also discussed in the literature [114–117]. From a visual assessment, it can be more intuitively seen that because CycleGAN tries to generate overall images that are probabilistically

similar to the training disease images, the symptoms are often generated in the surrounding areas, meaning that the disease classifier may use background areas as the discriminative regions.

Although our system achieved promising results, we still observed two remain limitations. First, the proposed LFLSeg may incorrectly detect "partial leaf" as "full leaf" if the "partial leaf" image has a different shooting distance than images in our training dataset. Even though we rarely encounter this extreme case, applying data augmentation techniques such as random resize/scale is expected to increase the robustness of this module and thus, boosting the performance of our system for future usage. Second, due to the complex characteristics of the training dataset, both LeafGAN and CycleGAN sometimes transformed the color rather than the disease symptom. Fig. 4.6 shows the characteristics of the powdery mildew (PM) disease in different stages (left column) and the failure cases of the H→P models from both LeafGAN and CycleGAN when translating from healthy images (second to last column). The PM dataset contained leaf images (left column) that were mostly in early and middle stages (first two images) with many of them are in dark blue color, while the later stage of PM disease (last image) is a typical case, but there is little in our dataset. Therefore, in several cases, the models generated images with a different color and few signs of PM symptoms. We believe that there is room for improvement in our system by addressing these practical problems, and we intend to investigate this in future work.

Chapter 5

Conclusion

In this work, we proposed three approaches to overcome the remaining practical problems on in-field plant disease diagnosis. As our knowledge, we were the first to explore the difficulties of establishing practical plant disease diagnosis systems for wide-angle images, and have compared two diagnosis strategies to solve these issues. Our experiments demonstrated that even sophisticated end-to-end systems still fell into overfitting and could not achieve the desired performance for an unknown dataset. On the other hand, although they required further improvement, our two-stage systems attained promising disease diagnosis performance for the unseen target dataset. These results showed that it is preferable to use two-stage systems due to the greater ease of collecting training data and assigning ground-truth labels, and due to the performance improvement they gave. We are continuing to improve our system and expect it will be applied to practical automated plant diseases diagnosis applications in the near future.

For improving the diagnostic performance from low-quality data, we have proposed an artifact-suppression SR method called leaf artifact-suppression super-resolution (LASSR), which was specifically designed for the automatic diagnosis of plant disease. Our LASSR model with the novel ARM effectively addressed the artifact effects produced by a GAN-based network and helped to improve the performance of automated plant leaf disease diagnosis. The proposed LASSR is capable of generating high-quality images and significantly improved disease diagnostic performance on unknown images in practical settings. From this perspective, we have confirmed that LASSR can be used as an efficient and reliable SR tool for real cultivation scenarios. Further research on the application of LASSR to other food crops is currently being carried out.

Lastly, we proposed the LeafGAN method as an effective data augmentation tool

for improving the robustness of an automated plant disease diagnosis system. Our LeafGAN generates countless diverse and high-quality training diseased images via transformation from healthy images. Thanks to its own attention mechanism, our model can transform only relevant areas from images with a variety of backgrounds, thus enriching the versatility of the training images. LeafGAN demonstrated significant improvements in the quality of the generated images and boosting the overall disease diagnosis performance on practical unseen data. We believe that our LeafGAN method is a reliable data augmentation tool and will make a significant impact on the field of automated crop disease diagnosis.

Currently, the above three approaches were developed independently. We argue that combining the LASSR and LeafGAN with the two-stage system could greatly improve the overall performance of diagnosing wide-angle plant images. The LASSR and LeafGAN work with single-leaf input images, and it is a suitable combination with the two-stage system. However, we believe there is still work to be done in order to realize this idea. For example, even LASSR and LeafGAN perform well on fixed scales practical in-field images, improvements are still crucial for the more complex wide-angle images where leaves are in various sizes, different shooting angles or distances, etc. We intend to develop this idea in the future.

Bibliography

[1] R. N. Strange and P. R. Scott, "Plant disease: a threat to global food security," *Annual Review of Phytopathology*, vol. 43, pp. 83–116, August 2005.

[2] S. Savary, L. Willocquet, S. J. Pethybridge, P. Esker, N. McRoberts, and A. Nelson, "The global burden of pathogens and pests on major food crops," *Nature Ecology & Evolution*, vol. 3, no. 3, pp. 430–439, March 2019.

[3] F. Qin, D. Liu, B. Sun, L. Ruan, Z. Ma, and H. Wang, "Identification of alfalfa leaf diseases using image recognition technology," *PLoS One*, vol. 11, no. 12, p. e0168274, December 2016.

[4] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, March 1995.

[5] L. Hallau, M. Neumann, B. Klatt, B. Kleinhenz, T. Klein, C. Kuhn, M. Röhrig, C. Bauckhage, K. Kersting, A.-K. Mahlein *et al.*, "Automated identification of sugar beet diseases using smartphones," *Plant Pathology*, vol. 67, no. 2, pp. 399–410, February 2018.

[6] E. Mwebaze and G. Owomugisha, "Machine learning for plant disease incidence and severity measurements from leaf images," in *Proceedings of the 15th IEEE International Conference on Machine Learning and Applications*, December 2016, pp. 158–163.

[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proceedings of the International Conference on Computer Vision*, November 2011, pp. 2564–2571.

[8] Y. Es-saady, I. El Massi, M. El Yassa, D. Mammass, and A. Benazoun, "Automatic recognition of plant leaves diseases based on serial combination of two svm classifiers," in *Proceedings of the International Conference on Electrical and Information Technologies*, May 2016, pp. 561–566.

[9] L. C. Ngugi, M. Abelwahab, and M. Abo-Zahhad, "Recent advances in image processing techniques for automated leaf pest and disease recognition-a review," *Information Processing in Agriculture*, vol. 8, no. 1, pp. 27–51, March 2021.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 25, December 2012, pp. 1097–1105.

[11] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 248–255.

[12] B. Liu, Y. Zhang, D. He, and Y. Li, "Identification of apple leaf diseases based on deep convolutional neural networks," *Symmetry*, vol. 10, no. 1, pp. 1–16, January 2018.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2015, pp. 1–9.

[14] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, p. 1419, September 2016.

[15] D. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv:1511.08060*, November 2015. [Online]. Available: https://arxiv.org/abs/1511.08060

[16] G. Wang, Y. Sun, and J. Wang, "Automatic image-based plant disease severity estimation using deep learning," *Computational Intelligence and Neuroscience*, vol. 2017, p. 2917536, July 2017.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations*, May 2015, pp. 1–14.

[18] H. Durmuş, E. O. Güneş, and M. Kırcı, "Disease detection on the leaves of the tomato plants by using deep learning," in *Proceedings of the 6th International Conference on Agro-Geoinformatics*, August 2017, pp. 1–5.

[19] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," in *Proceedings of the International Conference on Learning Representations*, April 2017, pp. 1–13.

[20] A. Elhassouny and F. Smarandache, "Smart mobile application to recognize tomato leaf diseases using convolutional neural networks," in *Proceedings of the International Conference of Computer Science and Renewable Energies*, July 2019, pp. 1–4.

[21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv:1704.04861*, April 2017. [Online]. Available: https://arxiv.org/abs/1704.04861

[22] H. A. Atabay, "Deep residual learning for tomato plant leaf disease identification," *Journal of Theoretical & Applied Information Technology*, vol. 95, pp. 6800–6808, December 2017.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp. 770–778.

[24] J. G. A. Barbedo, "Plant disease identification from individual lesions and spots using deep learning," *Biosystems Engineering*, vol. 180, pp. 96–107, April 2019.

[25] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the International Conference on Machine Learning*, May 2019, pp. 6105–6114.

[26] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using efficientnet deep learning model," *Ecological Informatics*, vol. 61, p. 101182, March 2021.

[27] Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, "Basic study of automated diagnosis of viral plant diseases using convolutional neural networks," in *Proceedings of the International Symposium on Visual Computing*, December 2015, pp. 638–645.

[28] E. Fujita, Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, "Basic investigation on a robust and practical plant diagnostic system," in *Proceedings of the IEEE International Conference on Machine Learning and Applications*, December 2016, pp. 989–992.

[29] 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, October 2018.

[30] T. Hiroki, R. Kotani, S. Kagiwada, U. Hiroyuki, and H. Iyatomi, "Diagnosis of multiple cucumber infections with convolutional neural networks," in *Proceedings of the Applied Imagery Pattern Recognition Workshop*, October 2018, pp. 1–4.

[31] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational Intelligence and Neuroscience*, vol. 2016, p. 3289801, June 2016.

[32] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, "Deep learning for image-based cassava disease detection," *Frontiers in Plant Science*, vol. 8, p. 1852, October 2017.

[33] C. DeChant, T. Wiesner-Hanks, S. Chen, E. L. Stewart, J. Yosinski, M. A. Gore, R. J. Nelson, and H. Lipson, "Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning," *Phytopathology*, vol. 107, no. 11, pp. 1426–1432, November 2017.

[34] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, February 2018.

[35] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, pp. 378–384, December 2017.

[36] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Computers and Electronics in Agriculture*, vol. 173, p. 105393, June 2020.

[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp. 2818–2826.

[38] A. Picon, M. Seitz, A. Alvarez-Gila, P. Mohnke, A. Ortiz-Barredo, and J. Echazarra, "Crop conditional convolutional neural networks for massive multi-crop plant disease classification over cell phone acquired images taken on real field conditions," *Computers and Electronics in Agriculture*, vol. 167, p. 105093, December 2019.

[39] A. F. Fuentes, S. Yoon, S. 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, p. 2022, September 2017.

[40] 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, September 2017.

[41] Q. Wang, F. Qi, M. Sun, J. Qu, and J. Xue, "Identification of tomato disease types and detection of infected areas based on deep convolutional neural networks and object detection techniques," *Computational Intelligence and Neuroscience*, vol. 2019, p. 9142753, December 2019.

[42] M. M. Ozguven and K. Adem, "Automatic detection and classification of leaf spot disease in sugar beet using deep learning algorithms," *Physica A: Statistical Mechanics and its Applications*, vol. 535, p. 122537, December 2019.

[43] P. V. Bhatt, S. Sarangi, and S. Pappula, "Detection of diseases and pests on images captured in uncontrolled conditions from tea plantations," in *Proceedings of the Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping IV*, vol. 11008, May 2019, p. 1100808.

[44] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks," *IEEE Access*, vol. 7, pp. 59 069–59 080, May 2019.

[45] X. Xie, Y. Ma, B. Liu, J. He, S. Li, and H. Wang, "A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks," *Frontiers in plant science*, vol. 11, p. 751, June 2020.

[46] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 580–587.

[47] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, December 2015, pp. 1440–1448.

[48] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, June 2016.

[49] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Proceedings of the Advances in Neural Information Processing Systems*, December 2016, pp. 379–387.

[50] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp. 779–788.

[51] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, October 2016, pp. 21–37.

[52] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, October 2017, pp. 2980–2988.

[53] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2017, pp. 7263–7271.

[54] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv:1804.02767*, April 2018. [Online]. Available: https://arxiv.org/abs/1804.02767

[55] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2020, pp. 10 781–10 790.

[56] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-yolov4: Scaling cross stage partial network," *arXiv:2011.08036*, November 2020. [Online]. Available: https://arxiv.org/abs/2011.08036

[57] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, October 2017, pp. 2961–2969.

[58] A. K. Singh, B. Ganapathysubramanian, S. Sarkar, and A. Singh, "Deep learning for plant stress phenotyping: trends and future perspectives," *Trends in Plant Science*, vol. 23, no. 10, pp. 883–898, October 2018.

[59] J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning: a review," *Plant Methods*, vol. 17, no. 1, pp. 1–18, February 2021.

[60] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, "Deepfruits: A fruit detection system using deep neural networks," *Sensors*, vol. 16, no. 8, p. 1222, August 2016.

[61] K. Bresilla, G. D. Perulli, A. Boini, B. Morandi, L. Corelli Grappadelli, and L. Manfrini, "Single-shot convolution neural networks for real-time fruit detection within the tree," *Frontiers in Plant Science*, vol. 10, p. 611, May 2019.

[62] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, "Apple detection during different growth stages in orchards using the improved yolo-v3 model," *Computers and Electronics in Agriculture*, vol. 157, pp. 417–426, January 2019.

[63] Q. H. Cap, K. Suwa, E. Fujita, H. Uga, S. Kagiwada, 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, October 2018.

[64] T. Saikawa, Q. H. Cap, S. Kagiwada, H. Uga, and H. Iyatomi, "Aop: An anti-overfitting pretreatment for practical image-based plant diagnosis," in *Proceedings of the IEEE International Conference on Big Data Workshops*, December 2019, pp. 5177–5182.

[65] K. Suwa, Q. H. Cap, R. Kotani, H. Uga, S. Kagiwada, and H. Iyatomi, "A comparable study: Intrinsic difficulties of practical plant diagnosis from wide-angle images," in *Proceedings of the IEEE International Conference on Big Data Workshops*, December 2019, pp. 5195–5201.

[66] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, October 2017, pp. 618–626.

[67] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair *et al.*, "Generative adversarial nets," in *Proceedings of the Advances in Neural Information Processing Systems*, December 2014, pp. 2672–2680.

[68] M. Arsenovic, M. Karanovic, S. Sladojevic, A. Anderla, and D. Stefanovic, "Solving current limitations of deep learning based approaches for plant disease detection," *Symmetry*, vol. 11, no. 7, p. 939, July 2019.

[69] Y. Tian, G. Yang, Z. Wang, E. Li, and Z. Liang, "Detection of apple lesions in orchards based on deep learning methods of cyclegan and yolov3-dense," *Journal of Sensors*, vol. 2019, p. 7630926, April 2019.

[70] H. Nazki, S. Yoon, A. Fuentes, and D. S. Park, "Unsupervised image translation using adversarial networks for improved plant disease recognition," *Computers and Electronics in Agriculture*, vol. 168, p. 105117, January 2020.

[71] H. Chen, M. Guan, and H. Li, "Arcyclegan: Improved cyclegan for style transferring of fruit images," *IEEE Access*, vol. 9, pp. 46 776–46 787, March 2021.

[72] Q. Wu, Y. Chen, and J. Meng, "Dcgan-based data augmentation for tomato leaf disease identification," *IEEE Access*, vol. 8, pp. 98 716–98 728, May 2020.

[73] S. Kanno, S. Nagasawa, Q. H. Cap, S. Shibuya, H. Uga, S. Kagiwada, and H. Iyatomi, "Ppig: Productive and pathogenic image generation for plant disease diagnosis," in *Proceedings of the IEEE-EMBS Conference on Biomedical Engineering and Sciences*, March 2021, pp. 554–559.

[74] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145–151, January 1999.

[75] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, July 2019, pp. 4780–4789.

[76] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu, and Z. Chen, "Gpipe: Efficient training of giant neural networks using pipeline parallelism," in *Proceedings of the Advances in Neural Information Processing Systems*, December 2019, pp. 103–112.

[77] B. C. Tom and A. K. Katsaggelos, "Reconstruction of a high-resolution image by simultaneous registration, restoration, and interpolation of low-resolution images," in *Proceedings of the International Conference on Image Processing*, vol. 2, October 1995, pp. 539–542.

[78] R. R. Schultz and R. L. Stevenson, "Extraction of high-resolution frames from video sequences," *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 996–1011, June 1996.

[79] A. J. Patti and Y. Altunbasak, "Artifact reduction for set theoretic super resolution image reconstruction with edge adaptive constraints and higher-order interpolants," *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 179–186, January 2001.

[80] D. Hirao and H. Iyatomi, "Prototype of super-resolution camera array system," in *Proceedings of the International Symposium on Visual Computing*, December 2015, pp. 911–920.

[81] E. Quevedo, E. Delory, G. Callicó, F. Tobajas, and R. Sarmiento, "Underwater video enhancement using multi-camera super-resolution," *Optics Communications*, vol. 404, pp. 94–102, December 2017.

[82] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, August 2002.

[83] T. Komatsu, Y. Ueda, and T. Saito, "Super-resolution decoding of jpeg-compressed image data with the shrinkage in the redundant dct domain," in *Proceedings of the Picture Coding Symposium*, December 2010, pp. 114–117.

[84] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, June 2015.

[85] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, pp. 4681–4690.

[86] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision*, September 2018, pp. 1–16.

[87] A. Jolicoeur-Martineau, "The relativistic discriminator: a key element missing from standard gan," in *Proceedings of the International Conference on Learning Representations*, May 2019, pp. 1–26.

[88] S. B. Kasturiwala and S. Aladhake, "Adaptive image superresolution for agrobased application," in *Proceedings of the International Conference on Industrial Instrumentation and Control*, May 2015, pp. 650–655.

[89] 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, p. 2557, November 2017.

[90] Q. H. Cap, H. Tani, H. Uga, S. Kagiwada, and H. Iyatomi, "Super-resolution for practical automated plant disease diagnosis system," in *Proceedings of the Annual Conference on Information Sciences and Systems*, March 2019, pp. 1–6.

[91] Q. Dai, X. Cheng, Y. Qiao, and Y. Zhang, "Crop leaf disease image super-resolution and identification with dual attention and topology fusion generative adversarial network," *IEEE Access*, vol. 8, pp. 55 724–55 735, March 2020.

[92] A. Giachetti and N. Asuni, "Real-time artifact-free image upscaling," *IEEE Transactions on Image Processing*, vol. 20, no. 10, pp. 2760–2768, April 2011.

[93] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proceedings of the European Conference on Computer Vision*, October 2016, pp. 694–711.

[94] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Proceedings of the Advances in Neural Information Processing Systems*, December 2017, pp. 6626–6637.

[95] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proceedings of the International Conference on Machine Learning*, vol. 30, no. 1, June 2013, p. 3.

[96] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning*, June 2015, pp. 448–456.

[97] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, May 2015, pp. 1–15.

[98] Y. Cui, M. Jia, T. Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2019, pp. 9268–9277.

[99] G. Toderici, D. Vincent, N. Johnston, S. Jin Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, pp. 5306–5314.

[100] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 586–595.

[101] A. F. Fuentes, S. Yoon, J. Lee, and D. S. Park, "High-performance deep neural network-based tomato plant diseases and pests diagnosis system with refinement filter bank," *Frontiers in Plant Science*, vol. 9, p. 1162, August 2018.

[102] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, October 2017, pp. 2223–2232.

[103] Y. A. Mejjati, C. Richardt, J. Tompkin, D. Cosker, and K. I. Kim, "Unsupervised attention-guided image-to-image translation," in *Proceedings of the Advances in Neural Information Processing Systems*, December 2018, pp. 3693–3703.

[104] X. Chen, C. Xu, X. Yang, and D. Tao, "Attention-gan for object transfiguration in wild images," in *Proceedings of the European Conference on Computer Vision*, September 2018, pp. 164–180.

[105] C. Yang, T. Kim, R. Wang, H. Peng, and C.-C. J. Kuo, "Show, attend, and translate: Unsupervised image translation with self-regularization and attention," *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 4845–4856, May 2019.

[106] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, April 2017.

[107] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of the International Conference on Medical Image Computing and Computer-assisted Intervention*, October 2015, pp. 234–241.

[108] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, pp. 2881–2890.

[109] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *International Journal of Computer Vision*, vol. 127, no. 3, pp. 302–321, March 2019.

[110] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free?-weakly-supervised learning with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2015, pp. 685–694.

[111] K. K. Singh and Y. J. Lee, "Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization," in *Proceedings of the IEEE International Conference on Computer Vision*, October 2017, pp. 3544–3553.

[112] K. Li, Z. Wu, K.-C. Peng, J. Ernst, and Y. Fu, "Tell me where to look: Guided attention inference network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 9215–9223.

[113] J. Lee, E. Kim, S. Lee, J. Lee, and S. Yoon, "Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2019, pp. 5267–5276.

[114] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv:1712.04621*, December 2017. [Online]. Available: https://arxiv.org/abs/1712.04621

[115] C. Han, L. Rundo, R. Araki, Y. Furukawa, G. Mauri, H. Nakayama, and H. Hayashi, "Infinite brain tumor images: Can gan-based data augmentation improve tumor detection on mr images?" in *Proceedings of the Meeting on Image Recognition and Understanding*, August 2018, pp. 1–4.

[116] K. Shmelkov, C. Schmid, and K. Alahari, "How good is my gan?" in *Proceedings of the European Conference on Computer Vision*, September 2018, pp. 213–229.

[117] Y. Ma, K. Liu, Z. Guan, X. Xu, X. Qian, and H. Bao, "Background augmentation generative adversarial networks (bagans): Effective data generation based on gan-augmented 3d synthesizing," *Symmetry*, vol. 10, no. 12, p. 734, December 2018.

[118] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proceedings of the International Conference on Learning Representations*, May 2016, pp. 1–16.

[119] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2019, pp. 4401–4410.