IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 10, 2020, accepted May 19, 2020, date of publication May 25, 2020, date of current version June 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2997001

DCGAN-Based Data Augmentation for Tomato Leaf Disease Identification

QIUFENG WU[1], **YIPING CHEN**[2], **AND JUN MENG**[1]
[1]College of Arts and Sciences, Northeast Agricultural University, Harbin 150030, China
[2]College of Engineering, Northeast Agricultural University, Harbin 150030, China

Corresponding author: Qiufeng Wu (qfwu@neau.edu.cn)

This work was supported in part by the Public Welfare Industry (Agriculture) Research Projects Level-2 under Grant 201503116-04-06, in part by the Postdoctoral Foundation of Heilongjiang Province under Grant LBHZ15020, in part by the Harbin Applied Technology Research and Development Program under Grant 2017RAQXJ096, and in part by the National Key Application Research and Development Program in China under Grant 2018YFD0300105-2.

ABSTRACT Tomato leaf disease seriously affects the yield of tomato. It is extremely vital for agricultural economy to identify agricultural diseases. The traditional data augmentation methods, such as rotation, flip and translation, are severely limited, which cannot achieve good generalization results. To improve the recognition accuracy of tomato leaf diseases, a new method of data augmentation by generative adversarial networks (GANs) is proposed for leaf disease recognition in this work. Generated images augmented by deep convolutional generative adversarial networks (DCGAN) and original images as the input of GoogLeNet, this model can achieve a top-1 average identification accuracy of 94.33%. By adjusting the hyper-parameters, modifying the architecture of the convolutional neural networks, and selecting different generative adversarial networks, an improved model for training and testing 5 classes of tomato leaf images was obtained. Meanwhile, images generated by DCGAN not only enlarge the size of the data set, but also have the characteristics of diversity, which makes the model have a good generalization effect. We have also visually confirmed that the images generated by DCGAN have much better quality and are more convincing through the t-Distributed Stochastic Neighbor Embedding (t-SNE) and Visual Turing Test. Experiments with tomato leaf disease identification show that DCGAN can generate data that approximate to real images, which can be used to (1) provide a larger data set for the training of large neural networks, and improve the performance of the recognition model through highly discriminating image generation technology; (2) reduce the cost of data collection; (3) enhance the diversity of data and the generalization ability of the recognition models.

INDEX TERMS Tomato leaf disease, data augmentation, generative adversarial networks, generalization, recognition accuracy.

I. INTRODUCTION

Tomato is one of the most nutritive crops all over the world, whose cultivation and level of production have a crucial impact on the development of agricultural economy. Tomato not only owns plenty of nutrition but also has pharmacological effects, which keep people away from diseases such as hypertension, hepatitis, gingival bleeding and so on [1]–[6]. On account of the wide use of tomato, the demand for tomato is also rising. Statistics show that more than 80 percent of agricultural production is produced by small farmers [7], and production loss of more than 50 percent because of pests and diseases [8]. Diseases and insect pests are the key factors that affect the growth of tomato, so it is particularly significant to study the identification of crop diseases [9].

Nevertheless, traditional manual detection of pests and diseases is low efficiency and high cost [10]. With the continuous development of the Internet, image-based disease identification has seen huge adoption in computer vision applications. People use efficient image identification technology to process images, which can improve the efficiency of image recognition, reduce the cost and improve the accuracy of recognition [11].

One of the most recognized competitions in the world, ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [12], which used ImageNet to verify the model,

The associate editor coordinating the review of this manuscript and approving it for publication was Yudong Zhang.

has received a lot of attention in the field of computer vision since it was held in 2010. However, it's not going to perform as well because in real world it does not have as much data as ImageNet. Training a deep model on insufficient data usually results in overfitting, since a model of high capacity is capable of ''memorizing'' the dataset [13]. There are some methods can solve the problems of overfitting like early stopping, which means stopping iteration before the iterative convergence of the model on the training dataset, regularization, dropout, data augmentation and so on. Data augmentation aims to increase the size of the dataset [14]. It is an approach that is widely used in all fields. In this task, we chose to focus our research on data augmentation. Traditional methods of image augmentation include translation, flip, rotation, brightness adjustment, affine transformation and Gaussian noise, etc. The purpose of this method is to obtain a new image that contains the same semantic information as the original one, which cannot improve the diversity of datasets and have no generalization ability. To make up for the shortcomings of common enhancement methods, Mohamad *et al.* proposed using dropout and data augmentation to reduce model overfitting [15]. Goodfellow *et al.* [16] compared identification accuracy by using different augmentation methods such as C-DCGAN [17], rotation and translation and without augmentation. The results not only show data augmentation can improve the identification accuracy and avoid overfit of deep learning networks for tea leaf's disease identification with insufficient training set size, but also show traditional augmentation methods have been unsuccessful in generalization. Pix2pix-GAN, stacked generative adversarial networks (SGAN) are also used because of their strength for image generation [18], [19]. Hojjat Salehinejad *et al.* [20] used the GAN network generated images to train the classifier, demonstrate that augmenting the original imbalanced data sets with GAN generated images improves performance of chest pathology classification using the proposed DCNN in comparison to the same DCNN trained with the original data sets alone. Tang *et al.* [21] used an end-to-end trained generative adversarial one-class classifier for abnormal chest X-ray identification. This method can generate normal and abnormal chest X-ray images using only normal chest X-ray images, thereby reducing the workload of manual annotation. Besides, several multi-stage training generation methods in diagnostic and biomedical domains are proposed: CPG-GANs incorporate high-rough bounding box conditions in PGGANs and uses incremental training procedures from low-resolution of newly-added layers to achieve high sensitivity for generating real images [22]. In addition, Han *et al.* also combines noise-to-image with image-to-image GANs and through Visual Turing Test and t-SNE results shows this method can significantly boost tumor detection sensitivity by using PGGANs, which is used to generate realistic images and SimGAN/Multimodal UNsupervised Image-to-image Translation (MUNIT), which is used to refer images generating by PGGANs. These methods have shown good performance in the field of medical image diagnosis.

In this paper, deep learning network is applied to identify tomato leaf diseases. We provide a method, which uses DCGAN network to do data augmentation to the manually collected dataset and takes part of the generated images together with the original images as the input of deep convolutional neural network. AlexNet [23], GoogLeNet [24], VGG16 networks [25] and ResNet [26] were selected as backbone of tomato leaf disease identification model, and transfer learning was used.

The rest of the paper is organized as follows: Section 2 presents related work. Then, Section 3 presents methodology. Section 4 presents achieved results and related discussion, and finally, section 5 holds our conclusions and future directions.

II. RELATED WORK

With the wide application of deep learning, many researchers focus on the identification of diseases and pests. Many researchers use a large dataset to study the problem of disease identification in agriculture, i.e., PlantVillage [27], processing a large number of plant leaf images of different species and different disease distribution from different regions. KC *et al.* trained and tested the model on publicly available PlantVillage subsets, with a classification accuracy of 98.34% [28]. Without data augmentation, Brahimi *et al.* trained the classifier with 14,828 images of the class of tomato from the open test dataset, and achieved 99.18 percent accuracy. In the aforementioned examples, you can see that one of the benefits of using open datasets is that you don't have to worry about model overfitting due to insufficient dataset size, and it can reduce the workload. However, datasets of some categories need to be enhanced, as the number of leaves of different types varies greatly, which will affect the accuracy of identification. The importance of balancing datasets is highlighted by the fact that deep neural networks may be most valuable in the work up of rare or challenging diseases, which practitioners at a common skill level may fail to recognize or misinterpret. Besides, other researchers use self-collected data as input to neural networks to identify different plant diseases. The difficulty of this kind of problem is not only hard to collect, but also need to find experts to do manual classification. Thus, in order to prevent model overfitting, there are many methods to solve this kind of problems in previous studies.

In this work, we focus on this approach of using data augmentation. The main purpose of data augmentation is to ensure the model will not see the same picture twice during training time and expose the model to much more aspects of data and thus generalize better [15]. Nevertheless, the common data augmentation method that uses the same picture for scaling and rotation make the diversity of the dataset is not enough so that it leaves their application ad-hoc and empirical. A method recently used by many researchers – generating adversarial networks – has been successfully applied to data enhancement. Generative Adversarial Networks (GANs) [17] are a family of unsupervised neural networks most commonly used for image generation [13]. As we all know, in 2017 Arjovsky *et al.* [29] proposed an alternative method to the traditional method, Wasserstein GAN (WGAN), which proved to improve the stability of learning and solve the problem of modal collapse. Also, Han *et al.* [30] proposed an approach for medical data augmentation which used Wasserstein GAN(WGAN) to generate realistic images applying in medical diagnosis. However, in this paper we use the leaf data set. Based on the particularity of the data characteristics, the leaf images of the same class have obvious differences at different stages of diseases, and the similarity of different classes of data is high. At present, there is no related research that can prove that WGAN has better performance in this field.

Some research based on GANs used to generate images has appeared in the field of agricultural disease recognition. Tian *et al.* [31] proposed an approach that can generate more apple disease images by CycleGAN. Nevertheless, most of the images generated by CycleGAN are of poor quality and have limited diversity. LeafGAN is an image-to-image translation system, integrating an attention mechanism, which generates diseased images via transformation from healthy images, as a data augmentation tool for improving the performance of plant disease diagnosis [32]. Its purpose is to balance the data set and generate clear images by transforming healthy leaf data into leaf data for many different diseases. Another form is noise-to-image, which is what we're going to study. We use the combined method of DCGAN and CNN to conduct the identification experiment with a small amount of data, aiming to solve the generalization problem of tomato leaf disease identification, make the computer more intelligent and reduce the workload of human. These two works have in common that both GANs networks can produce images with clear disease spots, and both can solve the problem of limitations related to data diversity. The difference is that LeafGAN can transform healthy images into images of different types of diseases, which is an image-to-image transformation method, while DCGAN is a noise-to-image transformation method, which generates labeled images into the same type of images.

III. MATERIAL AND METHOD

A. DATASETS

A total of 1500 tomato leaf images, openly and freely dataset, collected from the Plant Village project [27], which are distributed in 5 different classes, were selected for this study to aim for the highest variance among classes. The images in our dataset are annotated as belonging to five different categories which are tomato healthy, tomato late blight water mold, tomato septoria leaf fungus, tomato target spot bacteria and tomato YLCV virus (See Fig. 1). The size of each image should be the same as the input size of neural network (GoogLeNet, AlexNet and ResNet as 224 × 224 pixels, VGG as 299×299 pixels), and use RGB color space and JPG format images. In deep learning, the accuracy of identification is affected if the samples are not evenly distributed [33], [34]. In addition, considering that insufficient data often occur and it is difficult to collect vast data in many practical projects, 300 images of each type were randomly selected as original dataset of this work to better solve practical problems. Under this condition, there are two problems should be considered: (1) There will be more and more parameters as the number of network layers increase. (2) The number of the manually collected tomato dataset is small. It is apt to lead to overfitting of the network under the influence of many parameters and few data sets. Data augmentation is an effective way to solve this problem. In this work, DCGAN network is proposed for the augment of tomato dataset. Based on the original dataset, 240 samples from each type of tomato leaf dataset are randomly selected as training samples, and the remaining 60 samples from each type are used as test samples (See Table 1). We increased the training set to 1000 pieces per category by means of data enhancement. This number comes from the original ImageNet classification challenge, where the dataset had 1,000 categories, with fewer than 1,000 images per class (approximately 800 images). That's enough to train early models of image classification like AlexNet, so proving about 1,000 images is enough [35]. In this case, each kind of image dataset has 800 training samples, 200 validation samples and 60 test samples.

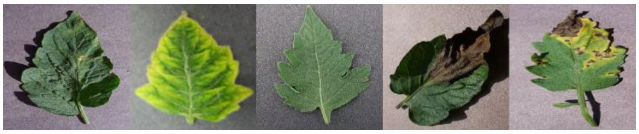


FIGURE 1. Example of tomato leaf images from the PlantVillage dataset, representing every tomato disease used. (1) tomato_target spot bacteria. (2) tomato_YLCV virus. (3) tomato_healthy. (4) tomato_late blight water mold. (5) tomato_septoria leaf spot fungus.

TABLE 1. The tomato leaf dataset.

Disease	Original dataset	Augmented dataset	Training	Validation	Test
Target spot bacteria	300	760	800	200	60
YLCV virus	300	760	800	200	60
Healthy	300	760	800	200	60
Late blight water mold	300	760	800	200	60
Septoria leaf spot fungus	300	760	800	200	60
Total	1500	3800	4000	1000	300

B. MODEL BULIDING

The main goal of this work is to generate realistic images for each of classes to solve the problem of insufficient data. In addition, we also improve the generalization ability of this model. The main process of tomato leaf diseases identification is shown in Fig.2. We designed a novel model structure to generate sample which hard to collect based on DCGAN's high stability and excellent sample generation capabilities.

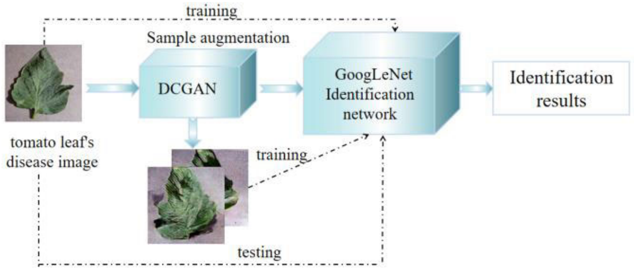


FIGURE 2. The main process of tomato leaf diseases identification.

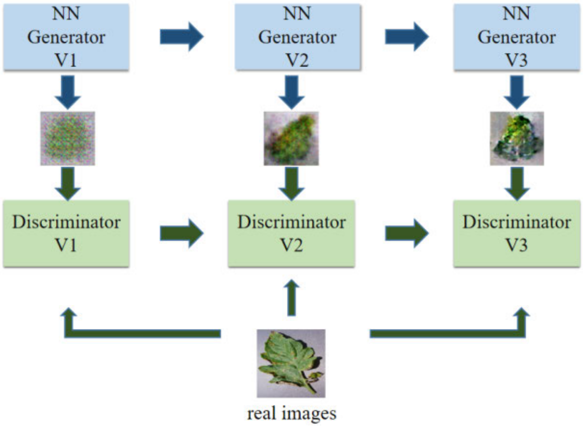


FIGURE 3. The main process of GAN.

1) BUILD DCGAN NETWORK

GAN is made up of generator network G and discriminator network D, which makes G learn the distribution of the data [17]. GAN has two different networks, among which G is a generator network, which receives a random noise z (random number) and generates images through this noise. D is a discriminator network that determines whether an image is "real" or not. Its input parameter is x, which represents a picture. Output D(x) represents the probability of real pictures. The main process of GAN is shown in Fig. 3. First, there is a generation generator that produces a very poor image. And then there's a generation of discriminators that accurately categorizes the generated images against the real ones. In short, the discriminator is a binary classifier that outputs 0 for the generated image and 1 for the real one. Next, we begin to train the second-generation generator, which can produce slightly better images and can make the first-generation generator believe that the generated images are real. It then trains a second-generation discriminator, which accurately identifies the real image with the one generated by the second-generation generator. And so on, there will be three generations, four generations…N generation of the generator and discriminator, and finally discriminator cannot distinguish between the generated image and the real image, the network fit. The objective function V (D; G) of GAN is as follows:

$$\min_{G}\max_{D} V\left(D,G\right) = E_{x\sim P_{data(x)}}\left[logD\left(x\right)\right] E_{z\sim P_{z}(z)}[\log(1 - D\left(G\left(z\right)\right))] \quad (1)$$

where x is a real sample, D(x) represents the probability of discriminating x as a real sample by discriminator networks D, G(z) is a sample generated from noise z by the generator network G, and D (G(z)) indicates the probability of discriminating G(z) as a real sample by discriminator network D. Compared with the earliest GAN, DCGAN was proposed in 2015. Convolutional neural network performs well in all tasks in supervised learning, but less in unsupervised learning. The algorithm of DCGAN, which can be considered as the application of GAN extended to the field of CNN, combines CNN in supervised learning with GAN in unsupervised learning. The advantage of GAN is that it requires no specific cost function and can learn good feature representation, but GAN is very unstable to train and often causes generators to produce meaningless output. Compared with GAN, DCGAN made some changes to the structure of convolutional neural network to improve the quality of samples and the speed of convergence. These changes include: all pooling layer is replaced by strided convolutions (discriminator) and fractional-strided convolutions (generator). Batch normalization was used on the generator and discriminator networks. The discriminator network is a convolutional neural network with the whole connection layer removed. In addition to using tanh as the activation function on the output layer, the relu activation function is used on the other layers of the generator network. All activation functions using LeakyReLu is used as a binary problem in discriminator network.

In this work, we proposed our own generator and discriminator model by referring to the DCGAN structure. The schematic diagram of DCGAN is as follows:

As shown in Fig. 4. For DCGAN based on neural network, the general learning process is as follows: As the input of G, z is a noise, which can be gaussian noise, usually uniform noise. After the generator G, a fake image is generated–G(z), and G (z) and x are taken as inputs to discriminator D. The output of the final discriminator D represents the probability that the data is real, which ranges from 0 to 1. For discriminator, batch normalization is generally not required after the first layer convolution, and the combination mode of "conv2D +BN+ LeakyReLU" is always followed. For the generator, the first layer is the full connection layer, then the combined mode of "conv2D+BN+ReLU", and the last layer of convolution is activated by tanh. Accordingly, input images are scaled to between −1 and 1 by dividing by 255 and multiplying by 2 minus 1.

2) BUILD IMAGE IDENTIFICATION NETWORKS

After aforementioned model of DCGAN enlarging training samples, we conduct a framework of image identification

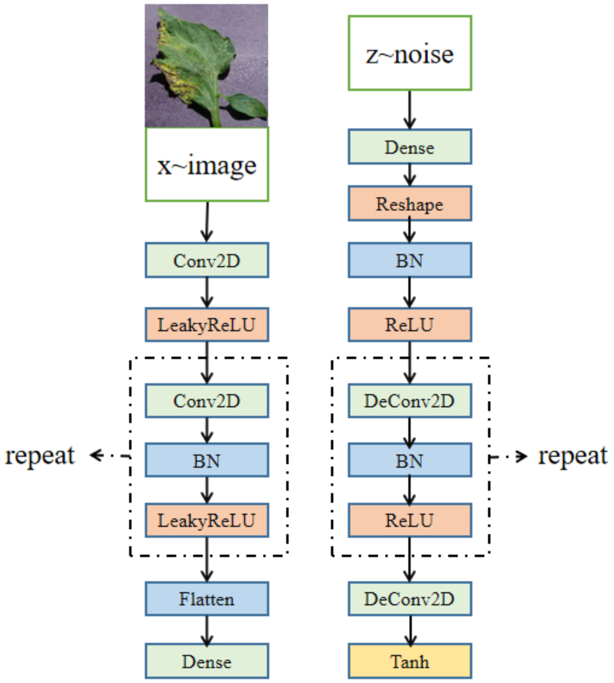


FIGURE 4. DCGAN's discriminator (left) and generator architecture (right).

based on deep learning, which performs well in recent years. Traditional neural network has a low accuracy, while machine learning like random forest has been demonstrated the unavoidable overfitting phenomenon will be exist in the problem of classification or regression with noise. Thus, we select deep neural networks. Due to the development of deep learning technology, computer vision has achieved good results in Large Scale Visual Recognition Challenge (ILSVRC) competitions for the ImageNet dataset [12], and its error rate has been lower than that of human vision. Some mainstream architectures we can see commonly are AlexNet, VGG, GoogLeNet, ResNet and so forth. In order to compare which CNN model is more suitable for the classification problem we proposed, we made corresponding experiments to verify the performance of these models. According to the characteristics of the number of the datasets, we use the method of transfer learning and choose the pre-trained model. We have made changes to some of the parameters of the architectures, which are detailed in the next chapter. Several metrics were used to evaluate the performance of our experiments (i.e. accuracy, precision, recall, f1), but for simplicity only the accuracy scores will be presented. Since the datasets are highly balanced, the rest of the metrics fall in line and consequently were considered redundant [13].

AlexNet: AlexNet was designed by Hinton and his student Alex Krizhevsky, who successfully applied ReLU, dropout and LRN in CNN for the first time, and AlexNet also used GPU for computation and acceleration. In order to avoid overfitting, dropout is used on the last fully connected layers of AlexNet, which can randomly ignore a subset of neurous while training. The AlexNet used in this work is shown in Fig. 5.

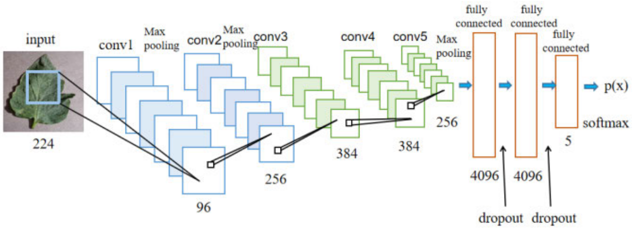


FIGURE 5. The architecture of AlexNet.

From the structure of the figure above, it can be seen that AlexNet is similar to the LeNet architecture proposed by Lecun in 1989 [36]. The network consists of eight weighted layers, the first five being the convolutional layers, and the remains are 3 fully connected layers. The first two convolutional layers are followed by the normalization and pooling layers respectively, and the last convolutional layer is followed by a single pooling layer. The third, fourth and fifth convolutional layers are connected directly. The second fully connected layer is provided to the softmax classifier with five class labels. ReLU, as the activation function of the first two fully connected layers (fc6, fc7), generates 4096 values from the results of 4096 operations. Finally, the output of the seventh layer of 4096 data is fully connected to the five neurons in the eighth layer(fc8). After training, it (fc8) outputs five floating-point values, which is the predicted result.

GoogLeNet: GoogLeNet (a.k.a, Inception V1) was the champion of the ILSVRC 2014 competition, which achieved a top-5 error of 6.67%. It proved to be it is extremely hard for human to do this well with such a low error accuracy. GoogLeNet has taken a bolder tack on networks. It implemented a newfangled section which is named an inception module, not like VGG, which inherits some of the architectures of LeNet and AlexNet. The inception module used batch normalization, RMSprop and image distortions. Although this model has a much deeper architecture with 22 layers, it drastically reduces the number of parameters, which is only 1/12 of AlexNet.

The overall structure of the GoogLeNet network is shown in the Fig. 6.

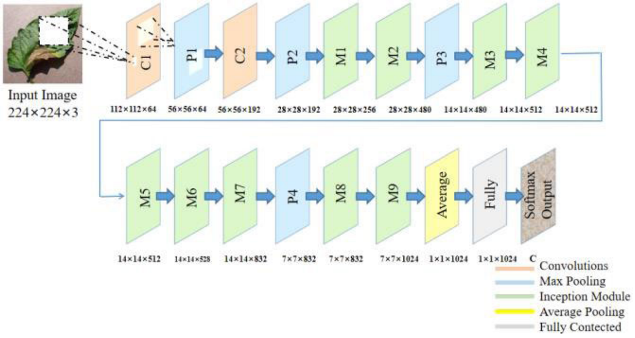


FIGURE 6. The architecture of GoogLeNet.

As it shown in Fig. 6, GoogLeNet used inception module(M1∼M9). The main idea of inception is how to find the optimal local sparse structure and cover it as an approximate dense component. It differs from traditional multi-channel convolution in that inception module used multiple convolutions (1×1, 3×3, 5×5) combined with max-pooling layer, which then associated the convolution and pooling results. Due to the large number of network parameters in the full connected layers, heavy computation and easy overfitting, GoogLeNet does not adopt the full connection structure in AlexNet, but directly used the method of averaging pooling and dropout after inception module, which not only plays a role in reducing dimension, but also prevents overfitting to some extent. In this work there are total 9 inception models in GoogLeNet architecture. A more detailed overview of this architecture can be found for reference in [24].

VGGNet: VGG, developed by Simonyan and Zisserman, was the runner-up at the ILSVRC 2014 competition. VGG has been improved on the basis of AlexNet, the entire network has used the same size of 3×3 convolution kernel size and 2×2 max pooling size, which make results of the network simple. The structure of VGG is shown in Fig. 7.

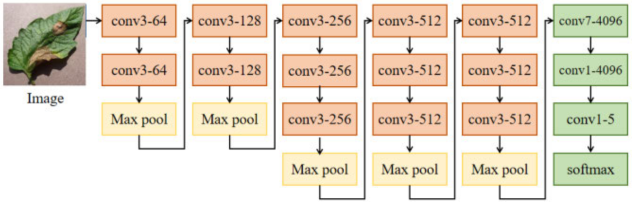


FIGURE 7. The architecture of VGGNet.

Take VGG16 network as an example, as we can see in the figure above: VGG contains 5 sets of convolution layers followed by a pooling layer. The difference is that five convolution layers contain more and more convolution layers in a cascade. In VGGNet, each convolution layer contains 2 to 4 convolution operations, the size of the convolution kernel is 3×3. It is by far the most popular option in the community for extracting features from images. However, VGGNet consists of 138 million parameters, which can be a bit challenging to handle.

ResNet: ResNet was proposed in 2015 by Kaiming He *et al.* introduced a novel architecture with "shortcut connections" and features heavy batch normalization, winning first place in the ImageNet competition classification task. As the name implies, shortcut means "choose the shortest path". There is a new structure in the ResNet, we call this "building block" (Fig.8).

We can see "a curved line", and this is the so-called shortcut connection. The whole graph is also known as "bottleneck design", which designed for ResNet-50/101/152. To be clear at a glance, it is in order to reduce the number of parameters. The first convolution of 1×1 gets the 256-dimensional channel down to 64, and then at the end it recovers by convolution of 1×1. In this work, the ResNet-50 was used, and the ResNet-50 was represented by ResNet.

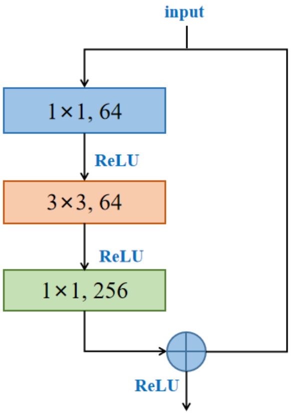


FIGURE 8. Bottleneck building block for ResNet.

TABLE 2. Workstation specifications.

Hardware (Machine Type)	Software
CPU: Intel Core i7-8700k	Windows10
RAM:32GB	CUDA9.0+CUDNN7.0
GPU: NVIDIA TITAN Xp	Keras Tensorflow

IV. EXPERIMENTS AND RESULTS

A. EXPERIMENTS DETAILS

In our implementation Keras and TensorFlow were used as deep learning framework in python to build the network model and we used dual Graphics Processing Unit (GPU) to accelerate the experimental process. The experimental setup is shown in Table 2. In this work, data augmentation scheme was performed to training dataset and validation dataset respectively. Transfer learning was applied to fine-tune the pre-trained models. In this work, 3 evaluations were selected as the quantitative evaluation indices: (a) generated image quality, (b) tomato leaf disease identification accuracy, and (c) DCGAN generalization ability.

B. COMPARISON EXPERIMENTS

To validate the performance of the proposed approach, we conducted a set of experiments using a real dataset combined with part of generated dataset by DCGAN. First and foremost, different pre-trained network models, for instance, AlexNet, GooLeNet, VGG16Net and ResNet, are used for comparative experiments to find a network with the best experimental results under the same conditions. In the case of using pre-training model used as a deep learning model for image classification can be easily reused for different problems under the condition of only slight fine-tune certain parameters [13], [37]. After that, according to the results, the best network framework is selected and its parameters are fine-tuned to compare the effects of different parameter settings on its recognition accuracy. Secondly, we chose to compare the performance of the variants of GANs such as BEGAN and DCGAN that have been successfully applied on other data sets. Two methods to evaluate network performance are selected: (a) generated image quality by human

TABLE 3. Identification accuracy (%) of different architectures.

Author	Model	Accuracy（%）
Proposed work	AlexNet	89.67
	GoogLeNet	94.33
	ResNet	83
	VGG16	79.3

evaluation, (b) Gan-train [38] and GAN-test indicators. The specific meaning will be introduced in the following part. Then select a network with the best experimental results and fine-tune its parameters. We conducted an experiment on the model of DCGAN network, comparing its performance under the conditions of different learning rate, batch size and other hyper parameters. Thirdly, a very important indicator is the prediction performance of our model on unseen data of tomato leaf diseases. Hence, in our experiments, we decided to test all the different range of train and test splits to evaluate the robustness of our proposed algorithm and its ability to avoid overfitting. The training dataset varies from 80%, 60%, 40 % to 20 % with the use of the same hyper parameters. Finally, our goal is to use transfer learning to train a good network with our own dataset. Therefore, the experiment we did was to compare the performance of whether dataset was augmented with DCGAN or not as the input of GoogLeNet. Other images were used as test datasets to verify the results and the best generalization results were obtained. Furthermore, t-Distributed Stochastic Neighbor Embedding (t-SNE) [39] is used to verify that the distribution of images generated by the proposed method is closer to the sample distribution of real images and the overlap between classes is smaller. We also evaluate the appearance of the generated images via Visual Turing Test [40] by plant experts.

C. PERFORMANCE EVALUATION

1) EXPERIMENTS ON DIFFERENT PRE-TRAINED MODELS

The identification accuracy of AlexNet, GoogLeNet, ResNet and VGG16 is shown in Table 3. In Table 3, the identification accuracy of AlexNet, GoogLeNet, ResNet and VGGNet is given in rows 2, row 3, and row 4 of the third column, respectively. We used an initial learning rate of 0.001 and then dropped by 0.5 per 512 iterations. Besides, Stochastic gradient descent (SGD) with a momentum of 0.9 was used for the optimization method. From the accuracy of identification, it can be seen that GoogLeNet is a better architecture than others under the same experimental conditions with the accuracy of 94.33.

2) EXPERIMENTS OF DIFFERENT PARAMETERS ON GoogLeNet

In order to clarify the influence of different parameters on network architecture, we rearranged the parameters of GoogLeNet in the above experiment. We adjusted the batch size and iterations, and the experimental results are shown in Table 4. According to the results, the model has the highest accuracy when the number of iterations is 2048. As the number of iterations increases (ranging from 512 to 2048), the overall accuracy increases. When the number of iterations is fixed, the accuracy of batch size which are 16 is lower than that of batch size which are 32. When the number of iterations is 2048 and the batch size is 32, the optimal result of the whole model is 94.33.

TABLE 4. Rearranged GoogLeNet results.

Iterations	Batch size	Accuracy（%）
512	16	91.00
	32	91.67
	64	92.67
1024	16	91.67
	32	92.00
	64	92.67
2048	16	93.67
	32	94.33
	64	94.00

In fact, any experiment should be analyzed according to the actual situation. Experimental results and parameter settings are largely determined by the datasets used and the performance of the computer. Batch size affects the optimization degree and convergence rate of the model, and its setting also needs to be analyzed according to the dataset actually selected. Since GAN added some generated data to the training set, it would lead to overfitting (the data in the test set is all original data, and the model learns what the original data looks like and what the generated data looks like). This identification model achieves optimal accuracy when the batch size is 32 and the number of iterations is 2048.

3) EXPERIMENTS ON DIFFERENT GAN MODELS

In this experiment, we used variants of GANs to generate images to solve the problem of few-shot learning. Due to the instability and intractability of the original GAN model, the generative adversarial networks we used is variant of GANs, DCGAN and BEGAN, which have been widely used in recent years with good effects. The experimental results are shown in Table 4. Since the number of original images in each of the 5 classes is 300, we randomly selected 240 images for data augmentation (the test set cannot be used for data augmentation) and enhanced the images in each class to 1000 by using the generative adversarial networks. In this experiment, the parameters of DCGAN and BEGAN were set to the default size, the learning rate was set to 0.0002 (the Learning rate of for Adam [0.0002]), and the magnitude of the momentum is set to 0.5(the Momentum term of Adam [0.5].) Three comparative experiments were designed: (a) training samples augmented by DCGAN, (b) training samples augmented by BEGAN, (c) training samples augmented by DCGAN and BEGAN.

We trained these five types of tomato leaf disease images respectively on GANs. To prevent all images from being read into memory at once, we used a mini-batch training method. The batch size is set to 64. The generated images and the original images are displayed as shown in Fig. 9.(a) to (j) show us the original tomato leaf images on the left and the generated tomato leaf images on the right, with images of different classes separated. There are similar features between the generated images and the original images, although the generated ones are of comparatively low resolution. The generated images convincingly show the characteristics of different disease types and can be classified for deep neural network training.

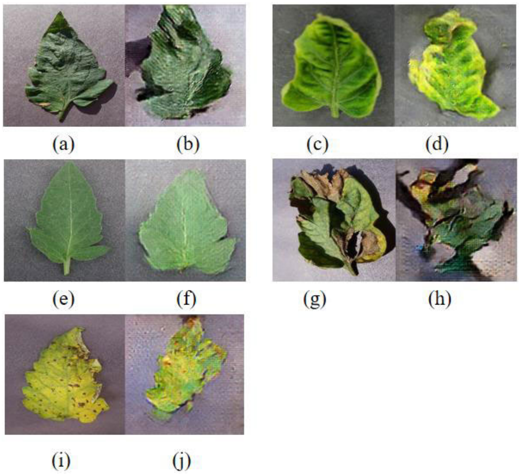


FIGURE 9. Samples of real (R) and synthesized (S) dataset: (a) Target Spot Bacteria-R; (b) Target Spot Bacteria-S; (c) YLCV Virus-R; (d) YLCV Virus-S; (e) Healthy-R; (f) Healthy-S; (g)Late Blight Water Mold-R; (h) Late Blight Water Mold-S; (i) Septoria-R; (j) Septoria-S.

TABLE 5. Accuracy (%) of different GANs.

Model	Accuracy (%)	GAN-train(%)	GAN-test(%)
DCGAN	94.33	66.00	67.00
BEGAN	88.67	21.30	49.67
DCGAN+BEGAN	91.00	55.00	60.67

GAN-train and GAN-test are accuracies given as percentage (higher is better).

The evaluation and comparison of GANs, or images generated by GANs, is a challenging task. In addition to observing the quality of generated images, we introduce three quantitative indicators based on image classification to evaluate the quality of GANs. As shown in Table 5, the second column in the table shows the identification accuracy obtained by using different networks which are used to augment the images and combining with the identification network of GoogLeNet. The indexes of the third and fourth columns are GAN-train and GAN-test respectively. The meaning of GAN-train is that a classifier is trained based on GANs generated images and tested on real images. This index evaluates the diversity and authenticity of GAN generated images. In addition, the meaning of GAN-test is that a classifier is trained according to the real images and tested on the generated images, which evaluates the authenticity of the GANs generated image. A pre-trained model trained on real data and generated data augmented by DCGAN achieves 94.33% accuracy on the test set. Images generated by DCGAN achieve a GAN-train accuracy of 66.00% and GAN-test accuracy of 67.00%, highlighting their high image quality as well as diversity.

Moreover, in order to better visualize the distribution of DCGAN-generated data, t-Distributed Stochastic Neighbor Embedding (t-SNE) was applied to this work. The t-SNE is a popular dimensionality reduction algorithm for visualizing high-dimensional datasets. It uses the distance between each individual data and all other data to weigh the correlation between each other. We randomly selected 100 images of each category, and a total of 500 images were used to test the results (see Fig. 10). In Fig. 10, different colors represent different labels; analysis from two aspects: the same class of data has a large overlap, and different classes of data are distributed far. It can be seen that the images of each class show the stripe-shaped distribution after dimensionality reduction. This phenomenon shows that DCGAN successfully captures the subtle features of real images, which are separable and can be used to train classification networks. Also, the wide distribution of data between the same category further indicates that the generated data samples are more diverse.

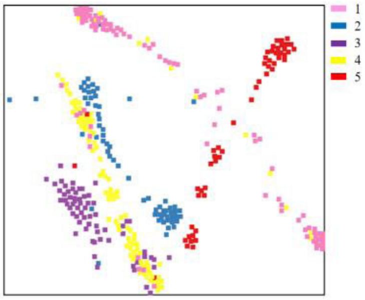


FIGURE 10. Two-dimensional scatter plot of high-dimensional data generated with t-SNE (1: tomato_target spot bacteria; 2: tomato_YLCV virus; 3: tomato_healthy; 4: tomato_late blight water mold; 5: tomato_septoria leaf spot fungus).

To visually analyze the distribution between the real images and DCGAN-based images, we randomly selected 100 real images and 100 generated images per class by using the t-SNE method. As Fig. 11 represents, the real image distributions largely overlap with the generated image distributions. This trend shows that the DCGAN-based images has a similar distribution to the real ones. As a whole, the images generated by DCGAN fill the distribution uncovered by the real ones with less overlap.

4) EFFECTS OF DIFFERENT PARAMETERS ON DCGAN

This group of experiments is similar to the second group of experiments. We rearranged the parameters of DCGAN network to achieve better generation effect and

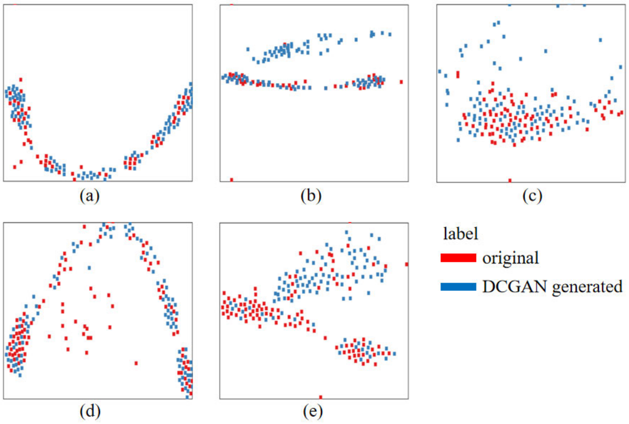


FIGURE 11. T-SNE plot with 100 tomato leaf disease images per each category: (a) tomato_target spot bacteria; (b) tomato_YLCV virus; (c) tomato_healthy; (d) tomato_late blight water mold; (e) tomato_septoria leaf spot fungus.

TABLE 6. Rearranged DCGAN results.

Learning rate	Momentum	Batch size	Accuracy (%)
0.02	0.5	16	94.33
	0.5	36	93.00
	0.9	16	86.67
	0.9	36	91.33
0.002	0.5	16	91.00
	0.5	36	92.33
	0.9	16	90.67
	0.9	36	86.00
0.0002	0.5	16	89.67
	0.5	36	94.00
	0.9	16	92.30
	0.9	36	91.00

TABLE 7. Use different proportions of the original sample.

Original samples	Generated samples	Train accuracy (%)	Test accuracy (%)
80	920	99.2	85
160	840	99.6	89.5
240	760	99.8	94.33

identification accuracy. We rearranged for the following parameters: learning rate, momentum, and batch size. The experimental results are shown in Table 6. When the fixed parameters are momentum and batch size, with the downward adjustment of learning rate value, the accuracy of recognition is lower. While when fixed parameters are learning rate and batch size, accuracy always works best when momentum is 0.5. Similarly, when fixed parameters are learning rate and momentum, we found that accuracy was not significantly affected by batch size. It can be concluded that when the learning rate is 0.02, the momentum is 0.5, and the batch size is 16, the accuracy rate is the best, and the result is 94.33.

5) EFFECTS OF DIFFERENT ORIGINAL IMAGES

In this experiment, we chose different proportions of data as the training set. In the experiment, 300 samples of each type of tomato leaf disease were collected, which were divided into training set, validation set and test set. The ratio of training set to validation set is fixed at 8:2. There are a total of 300 pieces of original data for each type of tomato leaf disease, and the rest is used for the test set. The design of the experiment is shown in Table 7. The first column of Table 7 shows the original samples, and the second column shows the samples generated using DCGAN network, both of which are used as training sets and validation sets. The third column in the table shows the accuracy of the training, while the fourth column shows the accuracy of the test. The training dataset varies from 80%, 60%, 40 % to 20 % with the use of the same hyper parameters. According to the experiment, when the training set (including the verification set) contains more information of the original data, the less information of the generated data and the higher the recognition accuracy.

TABLE 8. Use different data augmentation methods.

Model	Generalization results (100 graphs)
common	68%
DCGAN	83%

6) IDENTIFICATION ACCURACY USING TRAINING SAMPLES AUGMENTED BY DIFFERENT METHOD

In this experiment, GoogLeNet is trained with the following training samples: a. training samples augmented by DCGAN, b. training samples augmented by common augmentation method. The common augmentation method we used in this work involved operations such as move, rotation, flip and brightness enhancement. We use the original images and the generated images as the training set and validation set of the deep neural recognition network. We then used data from the non-training set and the validation set as test set to test the accuracy of recognition, which were found online. The experimental results are shown in Table 8. The average recognition accuracy using training samples augmented by DCGAN is about 15% higher than that of common augmentation method. The experimental results show that under the same conditions, generated data by DCGAN can enhance the diversity of the data set and improve the generalization ability of the model.

7) VALIDATION USING VISUAL TURING TEST

To test the quality of the generated images and the authenticity of the images, we conducted a Visual Turing Test with five botanists. The Visual Turing Test can be used to visually evaluate GAN-generated images. When the image information and background were not given, we conducted two sets of tests on 5 botanists: a total of 200 images were tested by selecting 20 real and 20 generated images from each class of tomato leaves. The first set of tests required experts to identify whether the image was generated or authentic; the second set of tests required experts to identify which class the image was. The questions for botanists are as follows:

TABLE 9. Experiment results of entire visual turing test.

Botanist Accuracy1 66% Accuracy2 90.2%	Botanist1 Accuracy1: 66% Accuracy2: 85.5%		R-R	R-G	G-R	G-G	
		Accuracy1	74%	26%	42%	58%	
			C1	C2	C3	C4	C5
		Accuracy2	77.5%	80%	85%	85%	100%
	Botanist2 Accuracy1: 57.5% Accuracy2: 91.5%		R-R	R-G	G-R	G-G	
		Accuracy1	64%	36%	49%	51%	
			C1	C2	C3	C4	C5
		Accuracy2	90%	100%	87.5%	80%	100%
	Botanist3 Accuracy1: 71.5% Accuracy2: 90%		R-R	R-G	G-R	G-G	
		Accuracy1	62%	38%	19%	81%	
			C1	C2	C3	C4	C5
		Accuracy2	80%	100%	90%	80%	100%
	Botanist4 Accuracy1: 70% Accuracy2: 92%		R-R	R-G	G-R	G-G	
		Accuracy1	71%	29%	31%	69%	
			C1	C2	C3	C4	C5
		Accuracy2	82.5%	100%	90%	87.5%	100%
	Botanist5 Accuracy1: 62.5% Accuracy2: 92%		R-R	R-G	G-R	G-G	
		Accuracy1	84%	16%	59%	41%	
			C1	C2	C3	C4	C5
		Accuracy2	90%	97.5%	92.5%	80%	100%

accuracy1 and accuracy2 denotes the successful identification rate between the real/generated images and between the different classes of images respectively.

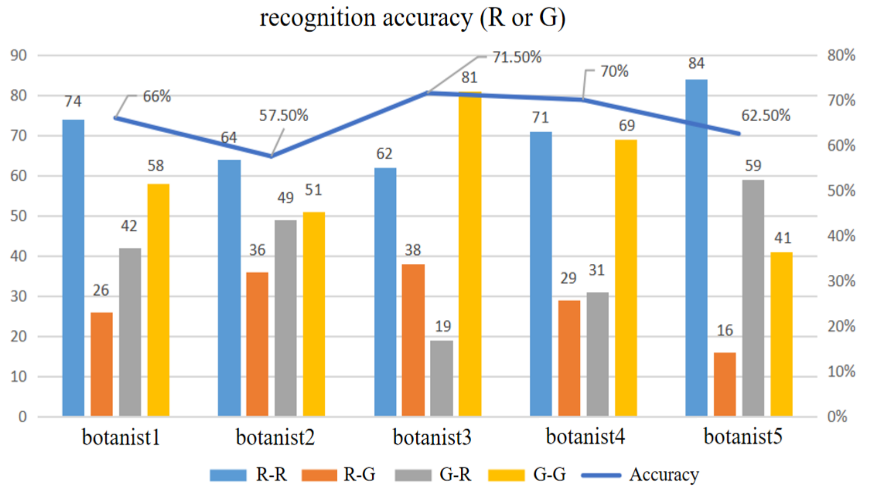


FIGURE 12. Visual Turing Test results (by botanists for classifying real(R) vs generated(G) images). R-R: a real image is recognized as a real image; R-G: a real image is recognized as a generated image; G-R: a generated image is recognized as a real image; G-G: a generated image is recognized as a generated image. Accuracy indicates the botanists' successfully classification rate of real / generated images.

(i) Identify which of the following images are real and which are generated by DCGAN; (ii) In addition, please determine which of these images are healthy and which are diseased. If the images are diseased, which category does it belong to?

In order to make the results of the Visual Turing Test reliable, we mix high-quality generated images with low-quality real images for comparison and set the size of all images to 100 × 100. In order to quantitatively measure the recognition results, we plot the test results, as shown in Fig. 12. Some images generated by DCGAN are considered real, while some real images are considered generated. The generated images successfully captured the characteristics of the real data, with an average accuracy of 66% for the five experts. Fig. 13 shows the accuracy of five experts in identifying tomato leaf classification problems. From Fig. 13, the recognition rate of some categories has reached 100%, which prove that generated leaves are more authentic. Table 9 shows the complete data results of the Turing test. Overall, botanists have a lower accuracy rate of 66% when judging whether the leaves are real or generated, and a higher accuracy rate of 90.2% when identifying tomato leaf species. The above experiments successfully proved that

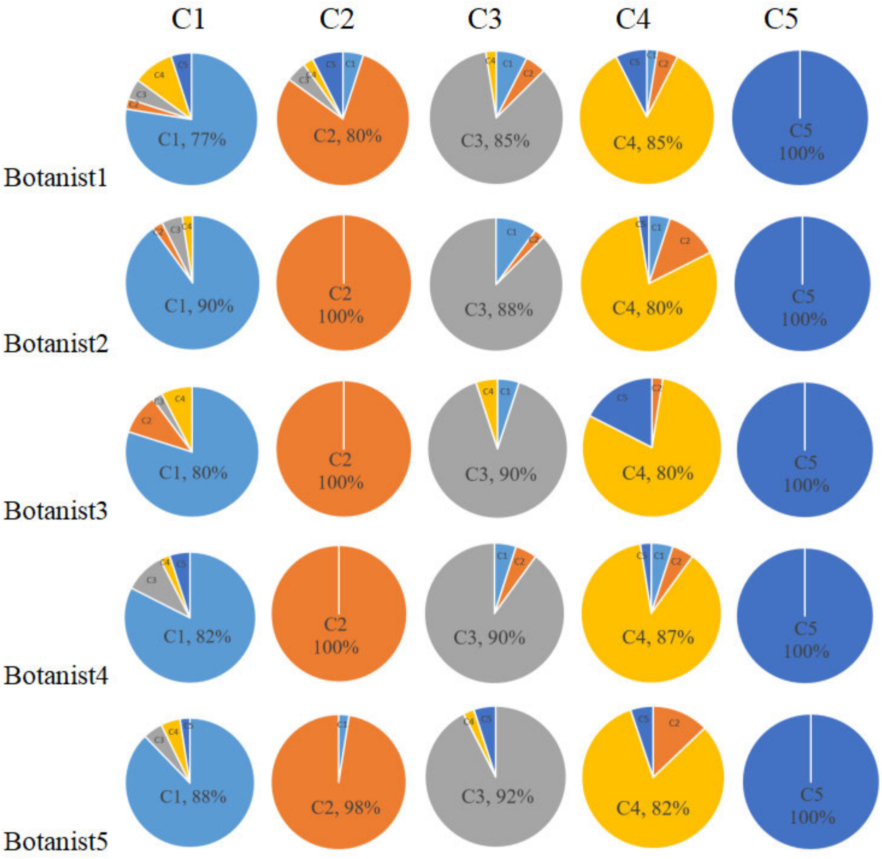


FIGURE 13. The above experiments were used to show the accuracy of 5 botanists in identifying images of different classes of tomato leaves. C1 ∼ C5 represent five classes of images. The percentage given in the pie chart indicates that the botanist identified the correct ratio (different colors indicate different categories).

TABLE 10. Studies in tomato leaf diseases identification.

Study	Classification method	Dataset	Accuracy (%)
(Raza *et al.*, 2015) [41]	SVM	71 images 2 classes	89.93
(Wang *et al.*, 2019) [42]	AlexNet+transfer learning	14529 images 10 classes	95.62
(Prasad *et al.*, 2016) [43]	KNN	287 images 5 classes	93.00
(Guo *et al.*, 2019) [44]	Multi-Scale AlexNet	5766 images 8 classes	92.70
Our method	DCGAN+ CNN	1500 images 5 classes	94.33

using DCGAN can generate realistic tomato leaf images and show good performance.

8) COMPARED WITH THE STATE-OF-ART METHODS

The state-of-art methods used to study tomato leaf diseases recognition should be mentioned here. Table 10 shows some recent works on tomato leaf diseases recognition, which separately lists the recognition methods, the number of pictures used and the accuracy rate. The first one used the traditional recognition method SVM. Although only 71 images are used for classification, its accuracy rate is only 89.93%. The second one has an accuracy of 95.62%, but 14,529 images were used to train the network. The remaining two accuracy rates are 93% and 92.7%, respectively. However, Prasad *et al.* used the KNN algorithm and the recognition process was complicated. Guo *et al.* used 5766 images to train the network. In short, a good network model with high recognition accuracy is something we need to consider, and the number of samples used should not be too much.

V. CONCLUSIONS

In this paper, we demonstrate that DCGAN can generate data that approximate to real images to provide both a larger data set for the training of large neural networks and improve the generalization ability of recognition models and enhance the diversity of data. We designed the experiments, which use the images of tomato leaf disease in the open dataset PlantVillage with the purpose of convincing people and simulate the few-shot learning problem, so as to achieve a good generalization effect. In Section IV, the experimental results show that DCGAN can generate real disease and health images of tomato leaves. Different from the traditional data augmentation methods, according to the t-SNE and Visual Turing Test results, the distributions of the images generated by GANs after dimensionality reduction were relatively clearly divided into different classes, and the generated images had more overlaps with the original images. This proves that the quality of images generated by GAN is superior to traditional data augmentation methods. In addition, we saw that the GAN-train and GAN-test values of DCGAN were higher than those of BEGAN (on the data set of tomato leaves), which proved the advantages of DCGAN's performance.

By combining DCGAN with GoogLeNet, the generated data and real data were mixed as the input of the convolutional neural network, we got the best results used to train the CNN network that we designed. In the meantime, we also solved the problem that the CNN network is difficult to converge put down to the difficulty of data collection and extremely similar features. In respect of details of identification, DCGAN can be optimized by adjusting batch size, learning rate and momentum to generate more realistic and diverse samples. By rearranging some parameters such as the batch size and learning rate of the identification network, the accuracy of the results can be improved.

In future work, we plan to find a better data augmentation method to solve the problem of tomato leaf disease recognition, so that the robustness and accuracy of the recognition can be improved. (i) Faced with the imbalance of the data set, try to use image-to-image GANs instead of noise-to-image GANs to convert healthy leaf images into disease leaf images to solve the problem of data imbalance in reality. Such an image-to-image translation system is proposed in Cap *et al.* [32]; (ii) According to the characteristics of plant leaf data, the spots of the same class of disease have obvious differences at different stages of disease, and the similarities of different classes of disease are high. A multi-scale convolutional neural network may be designed to comprehensively extract multiple features to improve the network responses with different granularity characteristics [44]. (iii) It is difficult to collect the leaves in actual works. Therefore, the problem of few-shot learning is urgent to be solved (i.e., Wang *et al.* proposed a method based on Siamese network for plant leaves classification [45]). All in all, by defining new methods to solve the problem of tomato leaf disease identification, we strive to achieve continuous improvement in performance.

REFERENCES

[1] P. Lindhout, W. Korta, M. Cislik, I. Vos, and T. Gerlagh, "Further identification of races of cladosporium fulvum (Fulvia fulva) on tomato originating from The Netherlands France and Poland," *Netherlands J. Plant Pathol.*, vol. 95, no. 3, pp. 143–148, May 1989.

[2] K. Kubota, S. Tsuda, A. Tamai, and T. Meshi, "Tomato mosaic virus replication protein suppresses virus-targeted posttranscriptional gene silencing," *J. Virol.*, vol. 77, no. 20, pp. 11016–11026, Oct. 2003.

[3] M. Tian, B. Benedetti, and S. Kamoun, "A second Kazal-like protease inhibitor from phytophthora infestans inhibits and interacts with the apoplastic pathogenesis-related protease P69B of tomato," *Plant Physiol.*, vol. 138, no. 3, pp. 1785–1793, Jul. 2005.

[4] R. Chaerani and R. E. Voorrips, "Tomato early blight (Alternaria solani): The pathogen, genetics, and breeding for resistance," *J. Gen. Plant Pathol.*, vol. 72, no. 6, pp. 335–347, Dec. 2006.

[5] W. Gao, B. Li, Y. Shi, and X. Xie, "Studies on pathogenicity differentiation of Corynespora cassiicola isolates, against cucumber, tomato and eggplant," *Acta Horticulturae Sinica*, vol. 38, no. 3, pp. 465–470, 2011.

[6] A. M. Dickey, L. S. Osborne, and C. L. Mckenzie, "Papaya (Carica papaya, Brassicales: Caricaceae) is not a host plant of tomato yellow leaf curl virus (TYLCV; FamilyGeminiviridae, GenusBegomovirus)," *Florida Entomologist*, vol. 95, no. 1, pp. 211–213, Mar. 2012.

[7] W. Matt *et al.*, *Smallholders, Food Security and the Environment*. Rome, Italy: International Fund for Agricultural Development, Mar. 2013.

[8] C. A. Harvey, Z. L. Rakotobe, N. S. Rao, R. Dave, H. Razafimahatratra, R. H. Rabarijohn, H. Rajaofara, and J. L. MacKinnon, "Extreme vulnerability of smallholder farmers to agricultural risks and climate change in madagascar," *Phil. Trans. Roy. Soc. B, Biol. Sci.*, vol. 369, no. 1639, Apr. 2014, Art. no. 20130089.

[9] L. E. B. Blum, "Reduction of incidence and severity of septoria lycopersici leaf spot of tomato with bacteria and yeasts," *Ciência Rural*, vol. 30, no. 5, pp. 761–765, Oct. 2000.

[10] L. Zhang, F. Yang, Y. Daniel Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3708–3712.

[11] S. P. Jia, H. J. Gao, and X. Hang, "Advances in image recognition technology of crop diseases and insect pests based on deep learning," *Trans. Chin. Soc. Agricult. Mach.*, vol. 50, pp. 314–317, Jul. 2019, doi: 10.6041/j.issn.1000-1298.2019.S0.048.

[12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[13] F. Konidaris, T. Tagaris, M. Sdraka, and A. Stafylopatis, "Generative adversarial networks as an advanced data augmentation technique for MRI data," in *Proc. 14th Int. Conf. Comput. Vis. Theor. Appl.*, Jan. 2019, pp. 48–59, doi: 10.5220/0007363900480059.

[14] J. Kukačka, V. Golkov, and D. Cremers, "Regularization for deep learning: A taxonomy," 2017, *arXiv:1710.10686*. [Online]. Available: http://arxiv.org/abs/1710.10686

[15] M. A. H. Abas, N. Ismail, A. I. M. Yassin, and M. N. Taib, "VGG16 for plant image classification with transfer learning and data augmentation," *Int. J. Eng. Technol.*, vol. 7, pp. 90–94, Oct. 2018.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inform. Process. Syst.*, 2014, pp. 2672–2680.

[17] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: http://arxiv.org/abs/1411.1784

[18] Y. Tang, S. Oh, J. Xiao, R. Summers, Y.-X. Tang, and R. M. Summers, "CT-realistic data augmentation using generative adversarial network for robust lymph node segmentation," *Proc. SPIE*, vol. 10950, Mar. 2019, Art. no. 109503V.

[19] Y. Tang, J. Cai, L. Lu, A. P. Harrison, K. Yan, J. Xiao, L. Yang, and R. M. Summers, "CT image enhancement using stacked generative adversarial networks and transfer learning for lesion segmentation improvement," *Mach. Learn. Med. Imag.*, vol. 11046, pp. 46–54, Sep. 2018.

[20] H. Salehinejad, S. Valaee, T. Dowdell, and J. Barfett, "Image augmentation using radial transform for training deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 3016–3020.

[21] Y.-X. Tang, Y.-B. Tang, M. Han, J. Xiao, and R. M. Summers, "Abnormal chest X-ray identification with generative adversarial one-class classifier," presented at the IEEE 16th Int. Symp. Biomed. Imag. (ISBI), Venice, Italy, Apr. 2019, pp. 1358–1361.

[22] C. Han, L. Rundo, R. Araki, Y. Nagano, Y. Furukawa, G. Mauri, H. Nakayama, and H. Hayashi, "Combining Noise-to-Image and Image-to-Image GANs: Brain MR image augmentation for tumor detection," *IEEE Access*, vol. 7, pp. 156966–156977, 2019, doi: 10.1109/ACCESS.2019.2947606.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inform. Process. Syst*, 2012, pp. 1097–1105.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

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

[28] K. Kc, Z. Yin, M. Wu, and Z. Wu, "Depthwise separable convolution architectures for plant disease classification," *Comput. Electron. Agricult.*, vol. 165, Oct. 2019, Art. no. 104948.

[29] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 214–223. [Online]. Available: http://arxiv.org/abs/1701.07875

[30] C. Han, H. Hayashi, L. Rundo, R. Araki, W. Shimoda, S. Muramatsu, Y. Furukawa, G. Mauri, and H. Nakayama, "GAN-based synthetic brain MR image generation," in *Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2018, pp. 734–738, doi: 10.1109/ISBI.2018.8363678.

[31] 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," *J. Sensors*, vol. 2019, pp. 1–13, Apr. 2019.

[32] Q. Huu Cap, H. Uga, S. Kagiwada, and H. Iyatomi, "LeafGAN: An effective data augmentation method for practical plant disease diagnosis," 2020, *arXiv:2002.10100*. [Online]. Available: http://arxiv.org/abs/2002.10100

[33] L. Bruzzone and S. B. Serpico, "Classification of imbalanced remote-sensing data by neural networks," *Pattern Recognit. Lett.*, vol. 18, nos. 11–13, pp. 1323–1328, Nov. 1997.

[34] F. J. Pulgar, A. J. Rivera, F. Charte, and M. J. del Jesus, "On the impact of imbalanced data in convolutional neural networks performance," in *Proc. Int. Conf. Hybrid Artif. Intell. Syst.*, vol. 10334. Cham, Switzerland: Springer, 2017, pp. 220–232.

[35] S. Du, Y. Wang, X. Zhai, S. Balakrishnan, R. Salakhutdinov, and A. Singh, "How many samples are needed to estimate a convolutional or recurrent neural network?" *Stat*, vol. 1050, p. 30, Jun. 2019.

[36] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Proc. Adv. Neural Inform. Process. Syst.*, 1990, pp. 396–404.

[37] E. S. Olivas, *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. Hershey, PA, USA: IGI Global, 2009.

[38] K. Shmelkov, C. Schmid, and K. Alahari, "How good is my GAN?" in *Proc. Euro. Conf. Comput. Vis. (ECCV)*, 2018, pp. 213–229, doi: 10.1007/978-3-030-01216-8_14.

[39] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[40] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 2234–2242.

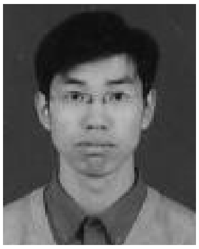
[41] S.-E.-A. Raza, G. Prince, J. P. Clarkson, and N. M. Rajpoot, "Automatic detection of diseased tomato plants using thermal and stereo visible light images," *PLoS ONE*, vol. 10, no. 4, Apr. 2015, Art. no. e0123262.

[42] Y. L. Wang *et al.*, "Image classification of tomato leaf diseases based on transfer learning," *J. China Agricult. Univ.*, vol. 24, no. 6, pp. 124–130, Jun. 2019.

[43] S. Prasad, S. K. Peddoju, and D. Ghosh, "Multi-resolution mobile vision system for plant leaf disease diagnosis," *Signal, Image Video Process.*, vol. 10, no. 2, pp. 379–388, Feb. 2016, doi: 10.1007/s11760-015-0751-y.

[44] X. Q. Guo, T. J. Fan, and X. Shu, "Tomato leaf diseases recognition based on improved Multi-Scale AlexNet," *Trans. Chin. Soc. Agricult. Eng.*, vol. 35, no. 13, pp. 162–169, Jul. 2019.

[45] B. Wang and D. Wang, "Plant leaves classification: A few-shot learning method based on siamese network," *IEEE Access*, vol. 7, pp. 151754–151763, 2019.



QIUFENG WU was born in Heilongjiang, China, in 1979. He received the Ph.D. degree in computer application technology from the Harbin Institute of Technology, Harbin, China, in January 2014. He is currently working as an Associated Professor with the College of Arts and Sciences, Northeast Agricultural University. His current research interests include machine learning, computer vision, and intelligent agriculture. He is a a member of CCF.

YIPING CHEN received the bachelor's degree in industrial engineering from Northeast Agricultural University, Harbin, China, where she is currently pursuing the M.S. degree with the College of Engineering. Her current research interests focus on machine learning, computer vision, and intelligent agriculture.

JUN MENG received the Ph.D. degree in agricultural economics and management from Northeast Agricultural University, Harbin, China, in July 2003. He is currently working as a Professor with the College of Arts and Sciences, Northeast Agricultural University. His current research interests include machine learning and agricultural economics.