Jurnal Elektronika dan Telekomunikasi (JET), Vol. 20, No. 1, August 2020, pp. 29- 35
Accredited by RISTEKDIKTI, Decree No: **32a/E/KPT/2017**
doi: 10.14203/jet.v20.29-35

Data Augmentation using Adversarial Networks for Tea Diseases Detection

R. Sandra Yuwana [a,*], Fani Fauziah [b], Ana Heryana [a], Dikdik Krisnandi [a], R. Budiarianto Suryo Kusumo [a], Hilman F. Pardede [a]

[a]Research Center for Informatics
Indonesian Institute of Sciences
Bandung, Indonesia
[b]Research Institute for Tea and Cinchona
Indonesian Agency for Agricultural Research and Development
Gambung, Indonesia

Abstract

Deep learning technology has a better result when trained using an abundant amount of data. However, collecting such data is expensive and time consuming. On the other hand, limited data often be the inevitable condition. To increase the number of data, data augmentation is usually implemented. By using it, the original data are transformed, by rotating, shifting, or both, to generate new data artificially. In this paper, generative adversarial networks (GAN) and deep convolutional GAN (DCGAN) are used for data augmentation. Both approaches are applied for diseases detection. The performance of the tea diseases detection on the augmented data is evaluated using various deep convolutional neural network (DCNN) including AlexNet, DenseNet, ResNet, and Xception. The experimental results indicate that the highest GAN accuracy is obtained by DenseNet architecture, which is 88.84%, baselines accuracy on the same architecture is 86.30%. The results of DCGAN accuracy on the use of the same architecture show a similar trend, which is 88.86%.

Keywords: Tea diseases detection, augmentation data, GAN, DCGAN

I. INTRODUCTION

Tea (*Camellia sinensis)* is one of main agriculture product from Indonesia. Based on data from FAO, Indonesia is one of the biggest tea producing countries in the world [1]. The Research Institute for Tea and Cinchona (RITC) in Gambung West Java is a research and development institute of Tea and Quinine commodities in Indonesia. The two superior varieties of RITC are the Gambung Assamica (GMB) series, which comprise of eleven types of clones, and Gambung Sinensis (GMBS), which comprise of five types of clones.

The crop production could significantly decrease when the tea plants are suffered from diseases. Tea diseases may also affect the quality of the produced tea. Several common diseases that are found in Indonesia are blister blight, caterpillar looper (Hyposidra talaca), and leafhoppers (Empoasca sp.) [2]. These diseases are usually triggered by climate conditions in Indonesia. For instance, for blister disease in GMB series, the trigger factors are the temperature and humidity. Usually, GMB clones get blister attack in the rainy season [3], but it would rarely be found during dry season.

Tea plant experts can detect diseases in tea leaves by looking at the physical characteristics that appear on the leaves, such as its colour, size, bone structure, and texture [4]. However, there would be inadequate number of experts available to provide correct diagnosis of the diseases. Therefore, the need of an automatic devices to identify the tea diseases is very important. The detection of tea leaves disease by utilizing deep learning technology is one of the best solutions. Several studies on the discovery of diseases in plants by using this technology have been carried out in [2], [4]-[8]. In these works, deep convolutional neural networks (DCNN) are usually employed. Works on [2] and [4] use the original image in the dataset as input for the experiment. Attention Cropping (AC) used in [5], transfer learning and deep feature extraction are used in [6], rotation transformation technique and mirror symmetry are used in [7], while [8] uses unsupervised feature learning. Only [8] uses DCGAN.

In deep learning, one of the main challenge is to avoid overfitting in the training process. According to Ying et al. [9], overfitting occurs due to several things, including the limited size of training data, the presence of noise, and the complexity of classifying data. DCNN usually requires large amount of data to train. The lack of data may cause the networks to easily overfit. Thus, practising with more data is one way to avoid overfitting. The availability of enough amount of data with variety of conditions is significant for deep neural

* Corresponding Author.
Email: rade014@lipi.go.id
Received: June 03, 2020 ; Revised: July 20, 2020
Accepted: August 05, 2020 ; Published: August 31, 2020

networks to produce a robust model. However, building a large dataset may be costly and time consuming. This is because a particular disease may only attack the plants on a limited time of a year. Producing data with great amount also requires human resources to collect them. One way to overcome the problem of limited data is by augmenting the data. With data augmentation, variations of new data for training the deep learning algorithms, making it less prone to overfit.

Data augmentation is the process of generating more diverse data either by collecting new data or generate new data artificially. Traditionally, In-depth learning requires more data when compared to machine learning. For this reason, optimal performance is a need. If the original image data is still inadequate to conduct training, then we can do data augmentation techniques on the original image data. This technique will manipulate data without removing important information from the data. There are many techniques for data augmentation, such as scaling, translation, rotation, cropping, bluring, flipping, transposinge, random erasing, and adding noise addition are techniques for data augmentation , and etc.

First approach is by transforming data. In these approach, various image processing and transformation such as translation, rotation, and shifting, are performed to the existing data to increase the number of data. In this approach, the resulted data, in a sense, are not really new and mostly effective to adapt the DCNN models to environmental variations.

Recently, there is an increasing interest to develop deep learning architectures for generating artificial data. In addition to traditional methods, other methods use generative adversarial networks (GAN). GAN is proposed to get data augmentation. A Generative Adversarial Networks (GAN) discovered by Goodfellow et al. [10] in 2014. GAN are generative models, which can produce trained images to learn underlying distribution of the real data without even see the real data.

Figure 1 depicts the architecture of GAN. GAN are generative models, which can produce images. There are two neural networks involved in the GAN architecture, namely generators and discriminators, which are involved in two “adversarial” networks. The generator tries to make a realistic picture to trick the differentiator. Whereas the discriminator, on the other hand, tries to correctly recognize whether the image shown is from the original image or generator (fake image), as shown in Figure 1. These two neural networks compete with each other to produce new synthetic data that can increase the amount of variation in real data. The exciting part of the GAN is that the discriminator cannot know which images are from the original dataset and which ones are from the generator. That is because the generator can make noise as random input and map the noise into an image. One variant of GAN is Deep Convolutional GAN (DCGAN) [11]. DCGAN has the same concept as GAN except it replace multilayer perceptron in GAN into convolutional layers. The DCGAN structure consists of several convolution layers without maximum unification or fully connected layers, as shown in Figure 2. In DCGAN, pooling layers with stride convolutions are used for discriminators, and fractal-stride convolutions are used for generators. Batch normalization is applied at the discriminator except at the input layer, and at the generator except at the output layer. The generator uses Rectified Linear Unit (ReLU) activation for all layers except for the output, while the discriminator uses Leaky ReLU activation for all layers.

In this paper, we wanted to improve data augmentation by using GAN and DCGAN for tea disease detection. Since GAN and DCGAN are unsupervised learning, we apply both on each class of our data. After the training process is complete, we use the generator to generate artificial data for each class, which could be augmented to the real data and then are used in the training. The performance of plant diseases detection is evaluated on popular DCNN architectures, such as Alexnet, DenseNet, ResNet, and Xception. Our experimental results show that the use of GAN and DCGAN for data augmentation, the performance is improved. It is better than only using original data for training.

The main contributions of this experiment are as follows:

1. The GAN and DCGAN approaches are implemented in data augmentation to produce new images that are more varied based on the original image reference.
2. Reducing cost, time and human resources in taking data of original tea leaves in tea plantations.
3. Advanced technology in generating new data on the tea leaf domain.

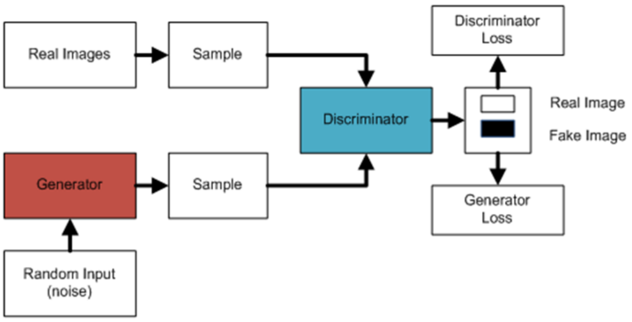


Figure 1. General Architecture of GAN

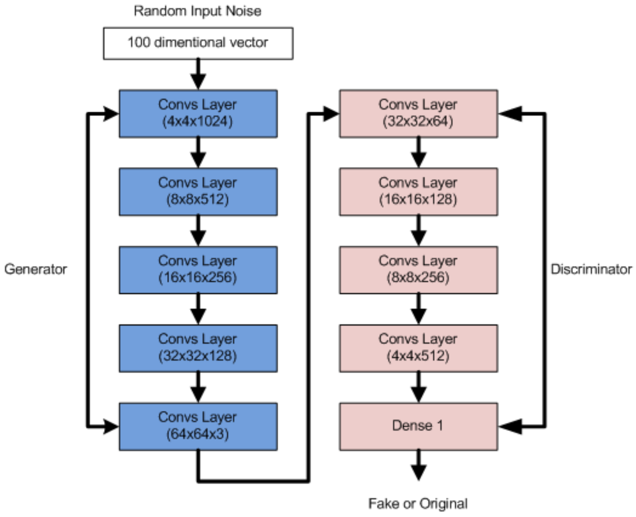


Figure 2. General Architecture of DCGAN

This paper has structure as follows: Section II describes our proposed method. In Section III, the experimental setup is presented. Section IV describes the result and discussion. Finally, Section V explains the conclusions.

II. Proposed Method

In this section, we first describe briefly, several DCNN architectures that are used in plant diseases detection. Then, we present the architecture of our GAN and DCGAN.

A. AlexNet

Krizhevsky et al. [12] introduce AlexNet and it achieves the best performance on ImageNet Large Scale Visual Recognition Challenge in 2012 AlexNet consists of eight convolution layers, max-pooling, and three are fully connected layers, as shown in Figure 3. Softmax is used as activation for the fully connected layers. This architectural image is redrawn from [13].

B. DenseNet

Densely Connected Networks (DenseNet) is proposed in [14], the source of Figure 4 [14]. This architecture maximizes the effect of shortcut connections, and all layers are connected directly. All layers receive additional input from all previous layers, and pass feature-maps to all subsequent layers, as shown in Figure 4. As a result, the network becomes compact and thinner, i.e., the number of channels become less. The number of channels shows the number of additional channels for each layer.

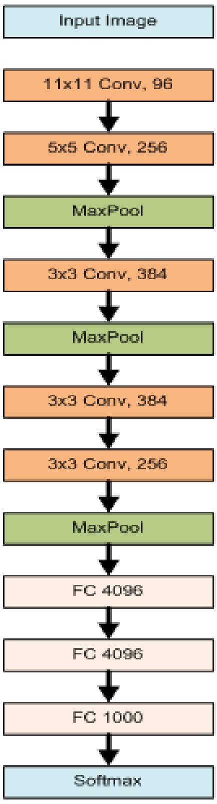


Figure 3. AlexNet Architecture

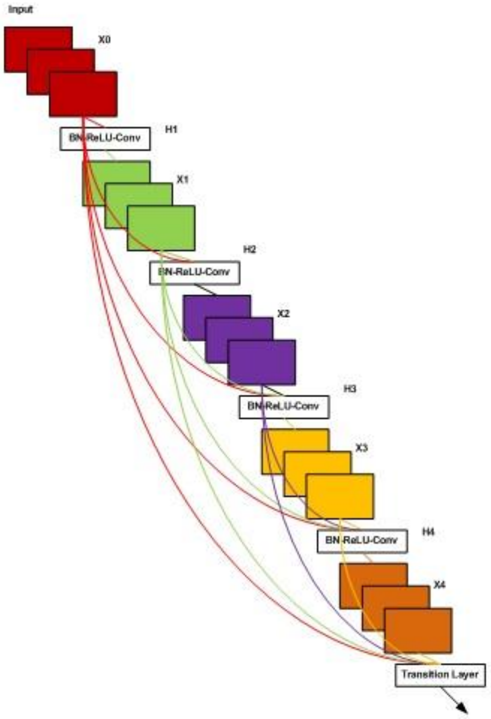


Figure 4. A 5-layer dense block with a growth rate of k = 4, Each layer takes all preceding feature-maps as input

C. ResNet

ResNet was introduced by He et al. in [15] to solve a problem of having very deep networks. Very deep networks, i.e. networks that have great number of hidden layers are prone to vanishing gradient problems. The effect is networks may fail to converge, and the solutions may stuck at local minimum and the performance may stuck at very low performance.

ResNet has the main identity uses "shortcut connection" that can jumppass-over one or more layers in the network, as shown in Figure 5 [15]. That is the core of ResNet, then various variants of ResNet are developing layers of the idea. ResNet applies residuals as the shortcut connection to the layer stack. The advantage of using a residual block is that it maximizes accuracy if it is used together with a conventional layer. If the dimensions of the input data in the model are the same as the exit data dimensions, then it will do a residual block. Here, the neural network will retain what it has learned. The number of hidden layers can increase as much as needed with the help of a residual block, and this method will not cause the gradient to explode/disappear.

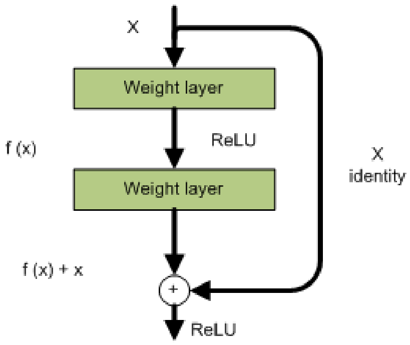


Figure 5. A residual block of ResNet

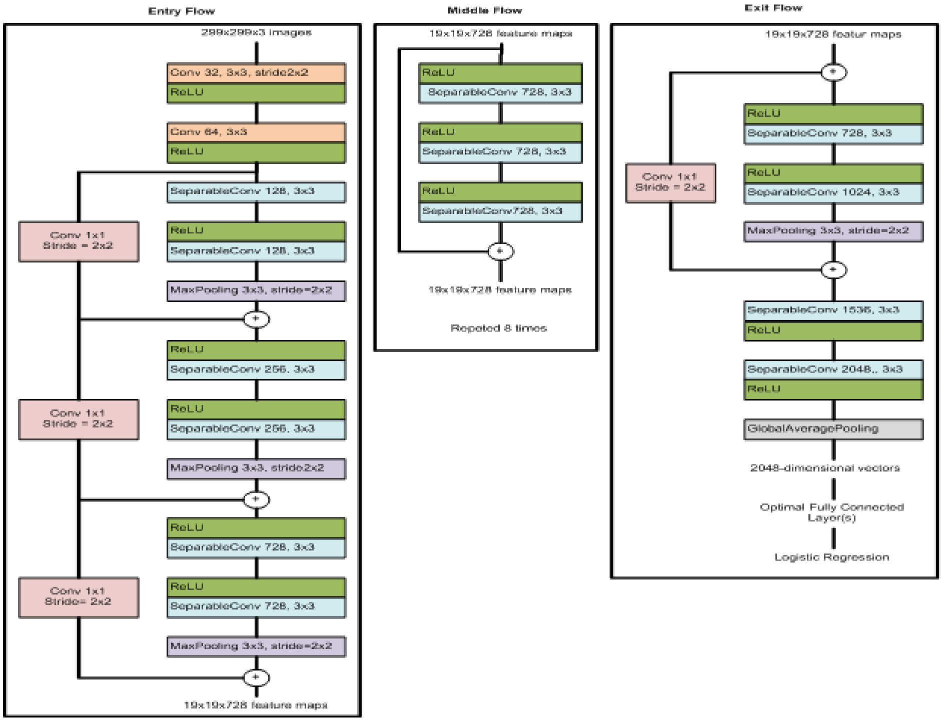


Figure 6. Xception architecture

D. Xception

Xception (extreme inception) is proposed [16]. Xception fixes Inception in terms of accuracy and speed. This architecture is built based on depthwise separable convolution layers and has three main flows, namely entry flow, middle flow, and exit flow. Xception architecture is illustrated in Figure 6 [16]. The data flow in Xception is as follows: first, the data enters through the entry flow. Then the data enters the middle flow, which is repeated eight times and exits through the exit flow. At each flow, separable convolution layers and all convolution layers are followed by batch normalization.

E. Traditional Data Augmentation

According to Shorten et al. [17], data augmentation is a series of techniques that can improve the quality of the training datasets and the number of datasets to get better deep learning models. The use of data augmentation can improve the performance of models with satisfying accuracy and increase data collection.

According to Perez et al. [18], some simple data augmentation techniques include rotation, zooming in/out, flipping, shifting, and shading with a hue, distorted. But, these traditional approaches are basically produce a “duplicate” images.

F. GAN and DCGAN for Data Augmentation

Figure 7 shows the role of the data augmentation process in the classification task. GAN, as a generative model, receives random input (noise) through the generator as shown in Figure 1. The generator generates an image based on the distribution of the real images that it learned. Discriminator is trained to determine that the new image is fake. This adversarial runs until the generator is able to trick the discriminator that the generated image is the original data. The generated images created during GAN training process are then added to the original dataset. Therefore the dataset obtains some new data variations.

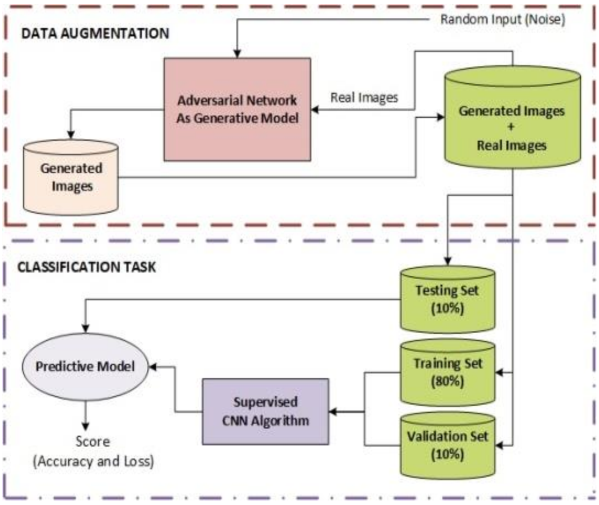


Figure 7. Illustration augmentation data for classification task in our proposed method

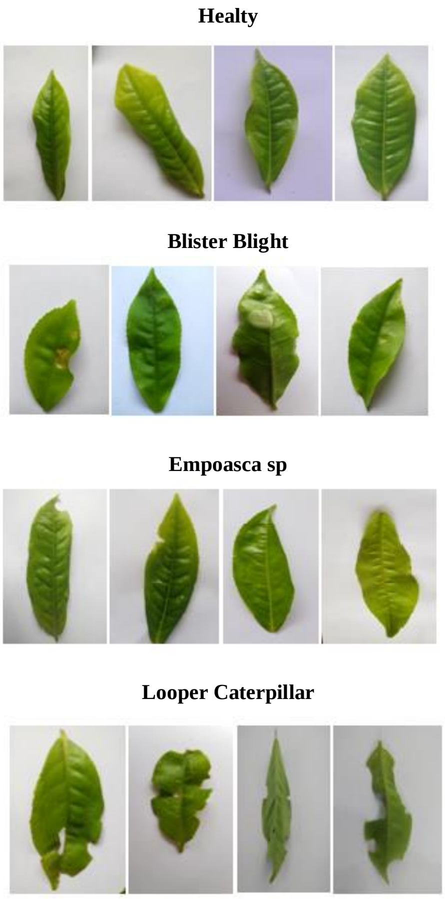
The adversarial networks that we use for data augmentation are GAN (Figure 1) and DCGAN (Figure 2). In general, GAN and DCGAN have the same architectures. The difference is, in GAN, multilayer perceptron is used whereas convolutional neural networks is used in DCGAN. The other difference is, DCGAN uses the Rectified Linear Unit (ReLU) activation function on the generator, while GAN applies LeakyReLU to the generator. Since GAN and DCGAN learn in unsupervised manner, we apply GAN and DCGAN for each class of our data. After GAN and DCGAN are trained for each class, generators are used to generate a number of new data for each class, which are then used for data training.

After the augmentation process, the dataset is divided into three parts: 80% training set, 10% validation set, and 10% testing set. Then the data is used to train several CNN architectures such as AlexNet, Densenet, Resnet and Xception, as supervised CNN algorithms. The results of the classification task are expected to provide satisfying accuracy.

III. Experimental Setup

In this experiment, we used healthy and diseased tea leaves from The Research Institute for Tea and Cinchona (RITC) in Gambung, West Java – Indonesia. The types of tea found in RITC, according to Prawira Atmaja et al. [19], include the variety of Camellia sinensis var. Assamica and Camellia sinensis var Sinensis. We collected images using a DSLR camera and a smartphone camera. Room conditions for shooting rely on room lighting and indirect sunlight, with a plain white background. The data of Tea leaf image consists of four classes, namely one class of healthy Tea leaves and three diseased Tea leaf classes.

Tea leaf diseases include, among others, blister blight, leafhoppers (Empoasca sp.), and caterpillar looper (Hyposidra talaca). Some examples of healthy and diseased tea leave as shown in Figure 8. The amount of original image data used in this experiment amounted to 4,727 data, consisting of 1,248 image data for healthy Tea leaves, 842 data for blister blight, 1,728 Empoasca sp. image data and 909 Looper Caterpillar data.



Healty

Blister Blight

Empoasca sp

Looper Caterpillar

Figure 8. Sample pictures of tea leaves

For experiments, the original dataset is divided into three parts: 80% training set, 10% validation set, and 10% testing set. In addition to images of original Tea leaves from RITC, we also use images of Tea leaves from data augmentation from GAN and DCGAN, for training. We generate 1000 and 2000 data, notated GAN100 and GAN2000, for each classes for these experiments so now there are 7,933 for GAN1000, and 11,933 samples data for GAN2000. The original training data are 3,933 sample data. Then the data are used to train several CNN architectures: Alexnet, Densenet, Resnet and Xception.

This experiment uses TensorFlow (TF), Python, Keras, and machine learning nodes. All of these are High-Performance Computing (HPC) facilities owned by LIPI. This experiment uses parameters for the network as follows, image size 64x64 pixels, number of epoch 100, Adam as optimizers, learning rate = 1e-5.

IV. Result and Discussion

Table 1 shows the accuracy of the data augmentation results for the baseline, GAN 1000, DCGAN 1000, GAN 2000, and DCGAN 2000. In the baseline, DenseNet architecture achieves the best accuracy with 86.30%, followed by AlexNet, Xception, and ResNet. This is interesting since DenseNet is the architecture with the least number of parameters (See Table 2). But, with limited number of data, it is preferreable to have architectures with small number of parameters than those with large number of parameters as the amount of needed data are usually proportional to the number of parameters. Using data augmentation clearly improves the accuracy for all CNN architectures. Augmenting 1000 data those are artificially generated with GAN (GAN1000) to training data improves the performance for all architectures while DCGAN1000 improves AlexNet and Densenet while the performance slightly worse for ResNet and Xception. Using more augmented data (GAN2000 and DCGAN2000) further improves the performance of ResNet and Xception, but the performance on AlexNet and DenseNet tend to be worse that GAN1000 and DCGAN1000. So generally, adding more data improves the performance of our systems as expected. As we aim to optimize millions of parameters, having more data are expected to improves the performance.

Figure 9 shows samples of generated data using GAN and DCGAN. All of the pictures of tea leaves in this paper are the result of our experiment. We observe that the quality of the generated data may affected the performance since data are generated artificially.

Tabel 1
The performance of different CNN architecture models for diseasses detection of tea leaves

Architecture	Accuracy (%)				
	Baseline	GAN 1000	DCGAN 1000	GAN 2000	DCGAN 2000
AlexNet	77.02	80.00	81.89	79.81	81.80
DenseNet	86.30	88.84	88.86	86.66	87.11
ResNet	71.53	72.16	70.63	73.78	73.51
Xception	72.25	73.06	71.98	73.06	64.59

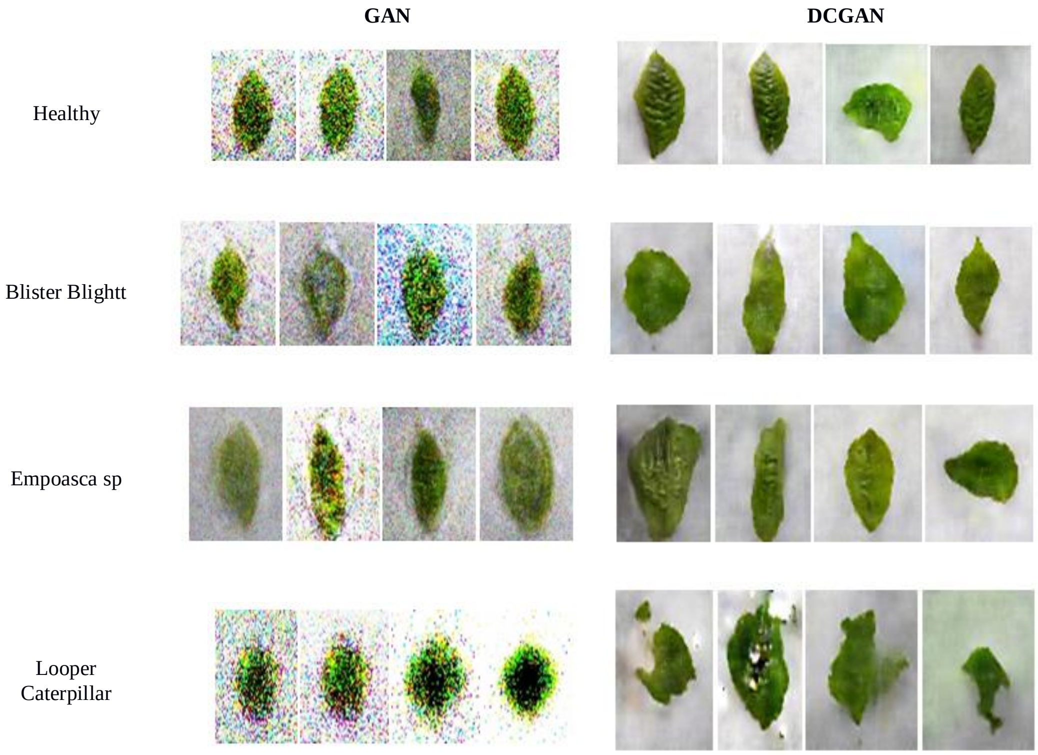


Figure 9. Example of the results of image augmentation data on DCGAN 1000

TABEL 2
NUMBER OF PARAMETERS FOR EACH CNN ARCHITECTURES

Architectures			
AlexNet	**DenseNet**	**ResNet**	**Xception**
29,750,532	1,075,276	23,580,548	20,869,676

Based on visual observation, in general, DCGAN generates more similar data to the original than GAN as GAN generates more “noisy” data and the distinctions between classes are less distinctive. But, DCGAN may produce images with similar looks (for instance empoasca and blister looks the same). So, adding more data does not necessarily improves the performance.

The image produced by the DCGAN has drawbacks, for example, for the human eye, the details of leaf diseases are not yet apparent. In the future, the use of regularization labels might be an interesting direction to take because it is very clear that GAN and DCGAN cannot produce high quality images.

CONCLUSION

In this paper, we propose the use of augmentation data to detect disease in tea leaves. New data samples were obtained using generative GAN and DCGAN models. An increase in the number and variation of sample images has an impact on increasing the accuracy of disease detection in tea leaves. The results show that the image quality produced by DCGAN is better than GAN and in generally better performance can be achieved when using DCGAN.

ACKNOWLEDGMENT

This work is supported by The Ministry of Research, Technology and Higher Education of the Republic of Indonesia, and Research Center for Informatics, Indonesian Institute of Sciences (LIPI). The experiment on this research is conducted on High Performance Computing (HPC) facilities in Research Center for Informatics, Indonesian Institute of Sciences (LIPI).

REFERENCES

[1] http://www.fao.org/3/y5143e/y5143e0z.htm, Accessed: Augst 5th, 2020

[2] D. Krisnandi, H.F. Pardede, R.S. Yuwana, V. Zilvan, A. Heryana, F. Fauziah, and V.P. Rahadi, “Diseases classification for tea plant using concatenated convolution neural network,” *CommIT (Communication and Information Technology) Journal*, vol. 13, no. 2, pp. 67-77, 2019.

[3] H. Lehmann-Danzinger, “Diseases and pests of tea: overview and possibilities of integrated pest and disease management,” *Journal of Agriculture in the Tropics and Subtropics*, vol. 101, no. 1, pp. 13-38, 2000.

[4] A. Ramdan, E. Suryawati, R.B.S. Kusumo, H.F. Pardede, O. Mahendra, R. Dahlan, F. Fauziah, and H. Syahrian, “Deep CNNBased detection for tea clone identification,” *Jurnal Elektronika dan Telekomunikasi*, vol. 19, no. 2, pp. 45-50, 2019.

[5] Q. Xiao, G. Li, L. Xie, and Q. Chen, “Real-world plant species identification based on deep convolutional neural networks and visual attention,” *Ecological Informatics*, vol. 48, pp. 117-124, 2018.

[6] M. Turkoğlu, D. Hanbay, “Plant disease and pest detection using deep learning-based features,” *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 27, no. 3, pp. 1636-1651, 2019.

[7] B. Liu, Y. Zhang, D.J. He, Y. Li, “Identification of apple leaf diseases based on deep convolutional neural networks,” *Symmetry*, vol. 10, no. 1, 2018.

[8] E. Suryawati, V. Zilvan, R.S. Yuwana, A. Heryana, D. Rohdiana, H.F. Pardede, “Deep convolutional adversarial network-based feature learning for tea clones identifications,” in *proc. 3rd International Conference on Informatics and Computational Sciences (ICICoS)*, 2019, pp. 1-5.

[9] X. Ying, “An overview of overfitting and its solutions,” *Journal of Physics: Conference Series*, vol. 1168, no. 2, p. 022022. IOP Publishing, 2019.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, “Generative adversarial nets,” *Advances in Neural Information Processing Systems*, pp. 2672-2680, 2014.

[11] A. Radford, L. Metz, S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434* (2015).

[12] A. Krizhevsky, I. Sutskever, G.E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, pp. 1097-1105, 2012.

[13] A. Lumini, L. Nanni, “Ocean ecosystems plankton classification,” *Recent Advances in Computer Vision*, pp. 261-280, Springer, Cham, 2019.

[14] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, “Densely connected convolutional networks,” in *Proc. of The IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700-4708.

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

[16] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. of The IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1251-1258.

[17] C. Shorten, T.M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, 6(1):60, 2019.

[18] L. Perez, J. Wang, “The effectiveness of data augmentation in image classification using deep learning,” *arXiv preprint arXiv:1712.04621* (2017).

[19] M. Prawira-Atmaja, H. Khomaini, H. Maulana, S. Harianto, D. Rohdiana, et al., “Changes in chlorophyll and polyphenols content in camellia sinensis var. sinensis at different stage of leaf maturity,” in *IOP Conference Series: Earth and Environmental Science*, vol. 131, p. 012010, IOP Publishing, 2018.