Effective data augmentation and training techniques for improving deep learning in plant leaf disease recognition

Prem Enkvetchakul
Ph.D. Student, Multi-agent Intelligent Simulation Laboratory (MISL), Department of Information Technology, Faculty of Informatics, Mahasarakham University, 44150, Thailand

Olarik Surinta*
Multi-agent Intelligent Simulation Laboratory (MISL), Department of Information Technology, Faculty of Informatics, Mahasarakham University, 44150, Thailand

* Corresponding author. E-mail: olarik.s@msu.ac.th (O. Surinta)

Abstract
Plant disease is the most common problem in agriculture. Usually, the symptoms appear on leaves of the plants which allow farmers to diagnose and prevent the disease from spreading to other areas. An accurate and consistent plant disease recognition system can help prevent the spread of diseases and save maintenance costs. In this research, we present a plant leaf disease recognition system using two deep convolutional neural networks (CNNs); MobileNetV2 and NasNetMobile. These CNN architectures are designed to be suitable for smartphones due to the models being small. We have experimented on training techniques; online, offline, and mixed training techniques on two plant leaf diseases. As for data augmentation techniques, we found that the combination of rotation, shift, and zoom techniques significantly increases the performance of the CNN architectures. The experimental results show that the most accurate algorithm for plant leaf disease recognition is NASNetMobile architecture using transfer learning. Additionally, the most accurate result is obtained when combining the offline training technique with data augmentation techniques.

Keywords: Plant leaf disease recognition, Deep learning, Convolutional neural networks, Transfer learning, Data augmentation

1 Introduction

Deep learning is currently combined with computer vision and artificial intelligence to help detect and recognize images and videos, as well as to help solve problems in different areas. For example, in medicine, deep learning is used in medical image classification [1], magnetic resonance imaging (MRI) [2], retinal image quality [3], brain abnormality classification [4], and sperm morphology analysis [5]. In the industrial arena, the deep belief network (DBN) is used in the process monitoring process employing industrial process images [6] and concrete pore structure [7].

In agriculture, deep learning is proposed for use in conjunction with the internet of things (IoT) technology and unmanned aerial vehicles (UAV) [8] to develop intelligent agriculture systems, such as agricultural environment prediction with long short-term memory (LSTM) and gated recurrent unit (GRU) to analyze data for temperature, soil moisture, pollution index, wind pressure, wind speed, and wind direction [9]. Deep learning and IoT used in agriculture result in higher quality agricultural products and also a reduction in the cost of farming.

The main problem that directly affects agricultural products is abnormalities caused by plant diseases and insect pests. Farmers must have knowledge and expertise to diagnose or solve problems in order to prevent and resolve them quickly and to avoid the spread of disease to a wider area. In this study, plant diseases that show leaf symptoms were divided into two main characteristics as follows: 1) The stage of disease formation may be the initial stage or the stage where a disease is widely spread and 2) some plant diseases have similar symptoms. If farmers lack the knowledge and fail to diagnose plant diseases, yields may be damaged. Therefore, many researchers have developed plant disease identification based on the leaves of plants such as rice, tomato, cucumber, apple, grape, and cassava [10]–[13]. Furthermore, most plant diseases can be identified by leaf.

Contribution: This research studies deep learning that can be used in plant leaf disease recognition system.

1. Studying the architecture of convolutional neural networks (CNNs) to create smaller models, including MobileNetV2 and NASNetMobile, and perform scratch and transfer learning for training speed and recognition accuracy with the aim of having an efficient and small model for use in applications on a smartphone.

2. The performance of the deep learning method is improved when combining data augmentation techniques and training techniques. In this paper, the image manipulation techniques consisting of width and height shift, rotation, zoom, brightness, cutout [14], and mixup [15] are used. We also test on three training techniques, including offline, online, and mixed methods.

3. We examine the proposed deep learning method on two sets of plant leaf disease data: the leaf disease and iCassava 2019 datasets.We found that the NASNetMobile architecture outperforms the MobileNetV2 architecture on the two plant leaf disease datasets when applying offline training technique and data augmentation, including rotation, shift, and zoom.

Outline of the paper: This paper is organized in the following way. Section II, we present a review of related work. Section III describes the background theory of two deep learning architectures, MobileNetV2 and NASNetMobile. The datasets which are used in the experiments are called plant leaf disease and iCassava 2019 datasets and are explained in Section IV. The experimental results and conclusion are presented in Section V and Section VI, respectively.

2 Related Work

2.1 *Deep Learning Architectures for Plant Leaf Disease Recognition*

Deep learning architecture is proposed for plant recognition, which can categorize characters of the leaf and fruit. Pawara et al. [16] proposed to use deep convolutional neural networks (CNNs), including AlexNet and GoogLeNet architectures. The accuracy performance of these CNN architectures provided more than 97% when using the transfer learning method. However, it obtained an accuracy of approximately 89% when training from scratch. It was reported that the transfer learning technique is more efficient in recognition and also reduces training time. Additionally, CNN architectures are used to recognize the plant disease, for example in rice [10], cassava [13], tomato, and cucumber leaf diseases.

Ramcharan et al. [13] experimented on the cassava disease dataset using Inception v3. This CNN architecture obtained an accuracy of 93%. Lu et al. [11] presented a new architecture of deep CNN architecture consisting of a convolutional layer and stochastic pooling layer. The softmax regression was proposed as the softmax layer. It was found that the deep CNN architecture achieved 95% accuracy, while Zhang et al. [10] designed three channels CNN for RGB color values, called TCCNN architecture. Each color channel was separated to calculate in the specific CNN of each channel: CNN1, 2, and 3. The final layers of CNN1, 2, and 3 were concatenated and delivered to the fully-connected layer for training and recognition. The recognition performance with this method was 91.15% on the tomato leaf disease dataset and 91.16% on the cucumber leaf disease dataset.

Sun et al. [17] presented the BJFU100 dataset, a plant dataset taken from a natural environment, with 10,000 images from 100 plants (ornamental plant species) in the Beijing Forestry University campus. The ResNet26 architecture was selected to test the number of layers consisting of 18, 26, 34, and 50 Layers. The experiment found that the ResNet26 architecture using SGD optimizer was fast in training with an accuracy of 91.78% on the BJFU100 dataset and accuracy of 99.65% on the Flavia dataset.

2.2 *Data Augmentation Techniques to Improve Deep Learning Performance*

Deep learning needs much information to create effective models and to avoid overfitting problems. However, lack of data may become a big issue in the case of models [18], [19]. Hence, the idea of generating new data based on existing data, which is called data augmentation, was proposed. Taylor and Nitschke [18] divided data augmentation into two techniques consisting of 1) geometric techniques: flipping, rotating and cropping, and 2) image metric techniques: color jittering, edge enhancement, and fancy principal component analysis. According to an experiment on the Caltech101 dataset, it was found that recognition of the CNN architecture was only 48.13% accurate, but when adding data using data augmentation with cropping, it has increased recognition accuracy to 61.95%. Shorten and

Khoshgoftaar [19] described that data augmentation is divided into two main categories consisting of 1) basic image manipulations: kernel filters, geometric transformations, random erasing, mixing images, and color space transformations and 2) deep learning approaches: adversarial training, neural style transfer and generative adversarial networks (GAN).

Mikołajczyk and Grochowski [20] compared two techniques of creating new datasets, consisting of 1) traditional transformation: shear, zoom in, reflection, rotation, contrast, histogram equalization, white balance and sharpen, called data augmentation and 2) GAN, which is commonly called data synthesis. GAN has the distinctive feature of style transfer, which means creating a synthetic image by learning from the original content combined with the new style. Therefore, it can create unlimited data in new styles, and the newly created synthetic image will look more realistic than the traditional transformation.

Using data augmentation in plant recognition, Pawara et al. [21] presented 7 data augmentation techniques including flip, rotation, blur, contrast, scaling, illumination, projective for experimented on the AgrilPlant, Folio, and Swedish datasets. The experiment found that data augmentation helped to make the CNN techniques more accurate. The new images are increasing 9-25 times and also directly increasing learning time. When using new images created by rotation and contrast techniques, the CNN techniques obtained 98.6% accuracy compared to 98.33% without data augmentation. The image data increased 17 times when data augmentation techniques were applied. The data used in training increased from 2,100 images to 35,700 images. For the Folio dataset, it reported that the accuracy result obtained 99.42% when applied illumination technique and compared to 97.63% without using data augmentation. The data increased from 445 images to 4,005 images. Therefore, it can be concluded that data augmentation can increase the efficiency of CNN techniques.

3 Convolutional Neural Network Architectures

Convolutional neural network (CNN) architectures are part of deep learning. The distinctive feature of CNN architecture is the convolution operation and the number of layers in the architecture. For example, the layer of the VGGNet [22] was designed to have 16 and 19 layer. The layer of the ResNet [23] is 18, 34, 50, 101, and 152 layers. Also, the layer of the DenseNet [24] is extended up to 264 layers. Importantly, the increase in the number of the layer is effected to increased network efficiency. However, the number of parameters is also increased. These architectures require devices that can be computed at high speed, such as the graphics processing unit (GPU), which is not suitable for smartphones and embedded devices [25].

This research aims to study the CNN architectures that can create a small and efficient model suitable for smartphones comprising MobileNetV2 [26] and NASNetMobile [27].

3.1 *MobileNetV2 Architecture*

Howard et al. [28] designed MobileNets architecture, also known as MobileNetV1, that is suitable for smartphones and embedded devices. Depthwise separable convolutions were proposed, which consisted of depthwise convolution and pointwise convolution to reduce the dimension of the number of layers and reduce the size of the parameter. Then, add the batch normalization (BN) layer and the rectified linear unit (ReLU) after depthwise separable convolutions in every step, as shown in Figure 1.

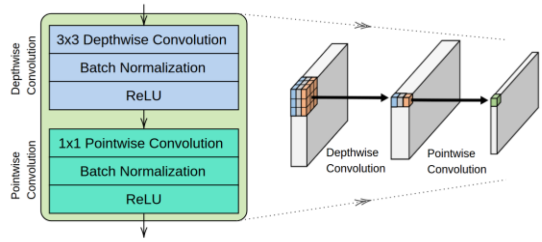


Figure 1: MobileNets with the depthwise separable convolutions process, which consists of depthwise convolution and pointwise convolution. The batch normalization layer and the rectified linear unit are added at the end of every convolutional layer [28], [29].

When using MobileNets to test on the ImageNet dataset, MobileNetV1 had 4.2M parameters, while popular architectures GoogLeNet and VGG16 architectures had 6.8M and 138M, respectively. The experiments of the MobileNetV1 on the ImageNet dataset obtained the accuracy of 70.6% [28] while the GoogLeNet obtained the accuracy of 69.8%

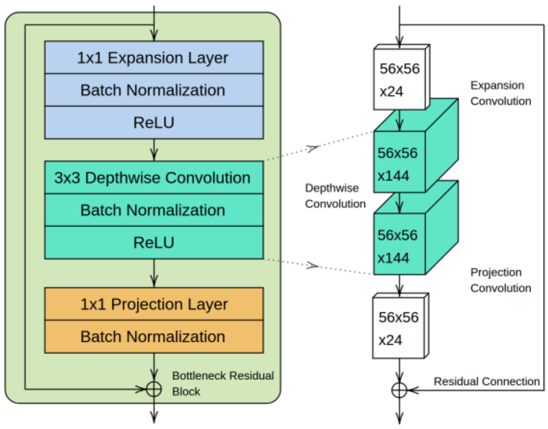


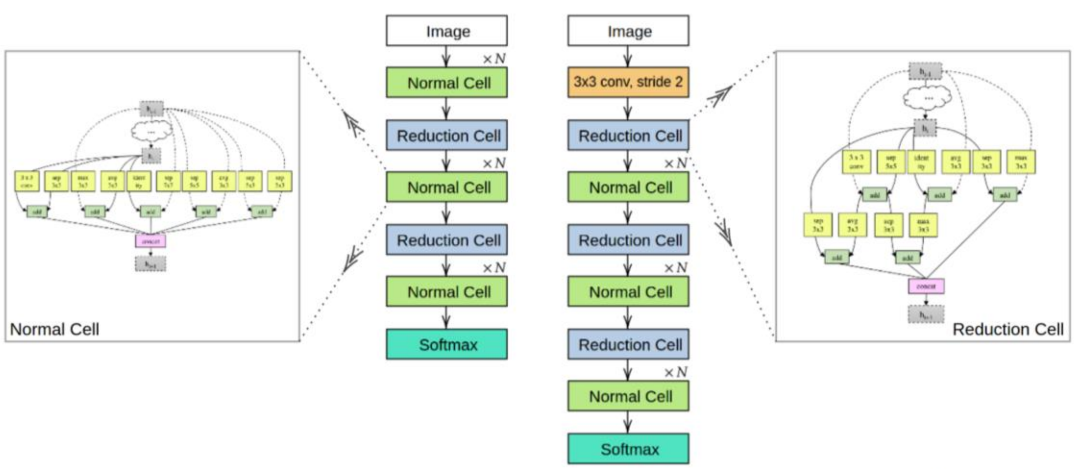
Figure 2: MobileNetV2 with inverted residuals. Process for making linear bottlenecks with the increase in feature map from 24 maps to 144 maps and the reduction of feature map from 144 maps to 24 maps [26].

Sandler et al. [26] introduced MobileNetV2 by increasing invert residuals, a short connection. Inverted residuals were designed to manage memory problems by reducing the amount of tensor stored on memory while processing. Inverted residuals are shown in Figure 2. The linear bottlenecks, which is an increase in the number of the feature map, such as ResNet [23] increases a feature map from 64 to 128, 256, and 512, respectively. Figure 2 shows the Linear Bottlenecks process, which begins with 24 maps and expanding it to 144 maps and 144 maps, respectively, then reducing the number of feature maps to only 24 maps before sending it to the next block. Also, the example shows that the feature map has changed up to 6 times.

MobileNetV2 architecture can decrease the number of parameters and faster in computation time than MobileNetV1. The experiments with MobileNetV2 obtained an accuracy of 72.0%, which was higher than with MobileNetV1, ShuffleNet, and NASNet [26].

3.2 *NASNetMobile Architecture*

Zoph and Le [27] designed a neural architecture search network, called NASNet architecture, using a recurrent neural network (RNN) and reinforcement learning to train to obtain the most accurate parameters from generated architecture. Creating a CNN architecture requires a lot of computation time if the content is large, such as the ImageNet dataset. Zoph et al. [30] designed the CNN architecture that can search the best architecture from a small dataset and transferred the best architecture to use to train on the large data, this architecture called learning transferable architectures. NASNet architecture can be scaled according to the amount of data. Figure 3 shows the scalability by increasing the number of normal cells and reduction cells, which can increase normal cells as required (N time), and normal and reduction cells can be obtained through a search process using the RNN method.



(a) (b) (c) (d)

Figure 3: Scalability of NASNet designed for use with (b) CIFAR10 dataset and (c) ImageNet dataset and examples of (a) normal cell and (d) reduction cell [27].

Figure 3 shows an examples of the normal and reduction cells obtained by searching with the controller RNN for the appropriate architecture from operation as follows:

- Identity
- 1 x 7 then 7 x 1 convolution
- 3 x 3 average pooling
- 5 x 5 max pooling
- 1 x 1 convolution
- 3 x 3 depthwise-separable convolution
- 7 x 7 depthwise-separable convolution
- 1 x 3 then 3 x 1 convolution
- 3 x 3 dilated convolution
- 3 x 3 max pooling
- 7 x 7 max pooling
- 3 x 3 convolution
- 5 x 5 depthwise-separable convolution

Controller RNN combines two hidden states to forward to the next hidden layer, as shown in Figure 4.

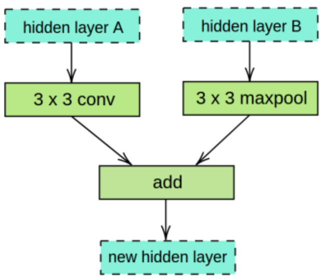


Figure 4: Block of convolution cell obtained from searching with RNN [27].

4 Example of Dataset

In this research, the accuracy of deep learning was experimented on two datasets of leaf diseases, consisting of the leaf disease dataset and iCassava 2019 dataset.

4.1 *Leaf Disease Dataset*

The leaf disease dataset is a collection of images of plant diseases, taking into account only the leaves of plants. Some images were collected from websites, while others were collected using a smartphone to take images of diseased leaves. As some plant diseases have similar symptoms, e.g. Whitefly-Transmitted (Figure 5(k)) and woolly aphid (Figure 5(l)) infestation the disearse may be wrongly identified, by inexperienced examinors. Then, all the leaf images in the dataset were screened by plant disease experts. From the screening process, a total of 608 plant leaf images were used, divided into 13 classes, as detailed in Table 1. The plant leaf images were cropped to show only affected areas and adjusted to be 224 x 224 pixels, as shown in Figure 5.

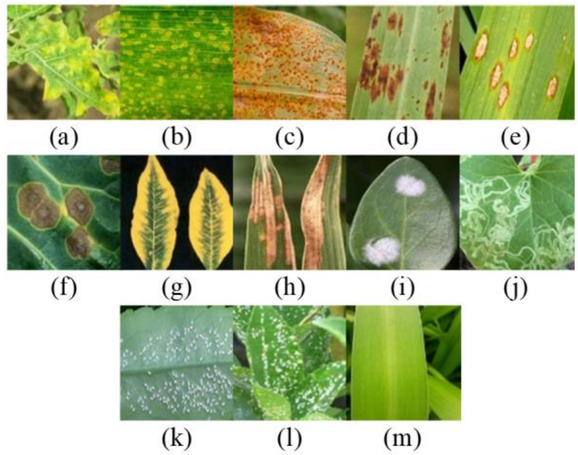


Figure 5: Sample images from leaf disease dataset, which consists of 13 classes consisting of (a) mosaic disease, (b) yellow leaf spot disease, (c) rust diseases, (d) narrow brown spot disease, (e) brown spot disease, (f) ringspot disease, (g) plant nutrient deficiencies, (h) leaf scald disease, (i) powdery mildew disease, (j) leaf miner, (k) whitefly-transmitted, (l) woolly aphid, and (m) healthy.

Table 1: Details of the leaf disease dataset (consists of 13 types; 12 types of plant diseases and one type of healthy) and the number of images of leaf diseases as each type of plant disease.

Types of Plants	No.	Types of Plants	No.
Mosaic Disease	44	Leaf Scald Disease	40
Yellow Leaf Spot Disease	40	Powdery Mildew Disease	47
Rust Disease	64	Leaf Miner	43
Narrow Brown Spot Disease	45	Whitefly-Transmitted	51
Brown Spot Disease	42	Woolly Aphid	49
Ringspot Disease	43	Healthy	42
Plant Nutrient Deficiencies	58		

4.2 *iCassava 2019 Dataset*

The iCassava 2019 dataset was presented at the sixth workshop on fine-grained visual-categorization (FGVC6 workshop) at the conference on computer vision and pattern recognition (CVPR 2019). This dataset contained images of 5 different diseases of cassava leaves, comprising 4 types of diseased cassava leaves and one type of normal leaf collected from Uganda. Farmers took images and sent them to The National Crops Resources Research Institute (NaCRRI) and AI lab in Makerere University, Kampala [31] for experts to sort the cassava leaves. The iCassava 2019 dataset includes 9,436 annotated images and 12,595 unlabeled images. In this research, however, we selected 5,656 annotated images published on the Kaggle website that contained four disease types and one healthy type, as shown in Table 2, and five types of cassava leaf images are shown in Figure 6.

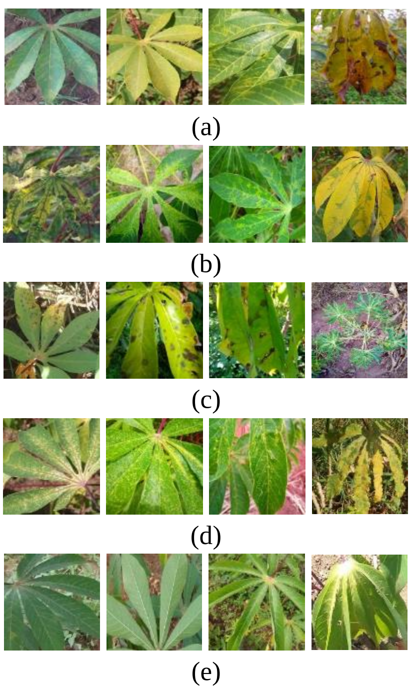


Figure 6: Examples of five types of iCassava 2019 dataset used in the experiment, consisting of (a) cassava brown streak disease, (b) cassava mosaic disease, (c) cassava bacterial blight, (d) cassava green mite, and (e) Healthy.

Table 2: Details of the iCassava 2019 dataset (consists of 5 types; 4 types of plant diseases and one healthy type) and the number of plant leaf images of each type.

Types of Plants	No. of Images
Cassava Brown Streak Disease (CBSD)	1,443
Cassava Mosaic Disease (CMD)	2,658
Cassava Bacteria Blight (CBB)	466
Cassava Green Mite (CGM)	773
Healthy	316

5 Experimental Result

This research studied two small convolutional neural network (CNN) architectures, consisting of MobileNetV2 and NASNetMobile, with the aim of identifying the best models to develop into smartphone applications. Data augmentation, which includes brightness, shift, rotation, zoom, cutout, and mixup was experimented with two datasets: 1) leaf disease dataset with a total of 608 images of diseased plant leaves, divided into 13 classes and 2) iCassava 2019 dataset with a total of 5,656 images, divided into five classes. In the experiment, the images were resized to 224x224 pixels before training with CNNs using TensorFlow's platform. The experiment was running on the Linux operating system with an Intel (R) Core-i5 computer, 2320 CPU @ 3.00GHz, 12GB RAM, GeForce GTX 1070Ti GPU.

5.1 *Experiments on Training Technique and Data Augmentation*

To test the hypothesis that training technique and data augmentation allowed CNN architecture to learn from limited data and increase the accuracy of recognition. First, we selected MobileNetV2 and trained the architecture using the fine-tuning technique [32]. Second, to demonstrate the performance of the training technique, we experimented with three training techniques; online, offline, and mixed training. Finally, the data augmentation, called rotation technique, was chosen with a random parameter between 0-170. Three training and data augmentation methods are as follows:

1) Offline training and data augmentation; This method generates new images in the pre-processing data scheme. The original image can create unlimited number of new images [19]. For example, from 100

original images, each of them can generate three new images. In total, the number of new images will increase to 400 images ((100 x 3) + 100). Therefore, the disadvantage of offline training technique is an increasing training time.

2) Online training and data augmentation; In this method, we combine online training and data augmentation to generate a new image in every training epoch. Therefore, this method can reduce training time. For example, if there are 100 input images to be trained by CNN architecture with 200 epochs, it is equivalent to sending 20,000 images (200x100) for training.

3) Mixed training and data augmentation; This method is a mixture of offline and online training techniques. First, in the pre-processing, we use a data augmentation technique to generate new images. So, this method increases the number of training images. Second, to allow the CNN architecture to learn more diverse data, new images are regenerated in every epoch during training CNN architecture to create the best model.

In this experiment, we evaluate the MobileNetV2 architecture on the leaf disease dataset. Data training was carried out using data augmentation, called the rotation technique, with a random parameter. The leaf disease dataset has 13 classes and contains 608 images, including 487 (80%) training images and 121 (20%) test images.

Table 3 shows the results of different training techniques and data augmentation on the leaf disease dataset. The results show that offline training and data augmentation method when randomly generating 15 new images from one original image significantly outperforms the other training techniques. The accuracy obtained from the offline training technique and data augmentation is 76.15%. However, it generated 7,792 training images in the pre-processing data scheme and took 15h 17min in training. The worst performance was obtained while training the CNN architecture without data augmentation, and the accuracy decreased to 63.08%.

As can be seen from the result in Table 3, it can be concluded that data augmentation has a direct effect on increasing recognition accuracy. Hence, we choose the offline training and data augmentation (15-image) technique in the following experiments.

Table 3: Results from three training techniques and data augmentation using the rotation technique. The results are computed using MobilenetV2 architecture on leaf disease dataset.

Training and Data Augmentation Techniques	Training Time	Training Samples	Accuracies
Offline Training + without Data Augmentation	1h 3 min	487	63.08
Online Training + Data Augmentation	1h 31min	487	74.62
Offline Training + Data Augmentation (3-image)	3h 54min	1,948	70.00
Offline Training + Data Augmentation (5-image)	5h 48min	2,922	72.31
Offline Training + Data Augmentation (7-image)	7h 46min	3,896	72.31
Offline Training + Data Augmentation (9-image)	13h 26min	4,870	74.62
Offline Training + Data Augmentation (15-image)	**15h 17min**	**7,792**	**76.15**
Mixed Training + Data Augmentation (15-image)	21h 33min	7,792	74.62

5.2 *Experiments on Leaf Disease Dataset*

In this section, to compare the performance of CNN architectures on leaf disease recognition, using MobileNetV2 and NASNetMobile architectures on the leaf disease dataset. The objective was to compare these two learning methods and show that transfer learning shows a better result than training data from scratch on the leaf disease dataset. Moreover, for testing the performance of data augmentation, we selected the basic image manipulations, which consist of seven techniques: rotation, brightness, width shift, height shift, zoom, cutout, and mixup. The new images are then generated according to the random parameters, as shown in Table 4. The example of the images obtained from data augmentation is shown in Figure 7.

(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)
(k)

Figure 7: Examples of the (a) leaf disease images and samples of data augmentation images using (b) rotation, (c) brightness, (d) shift, (e) zoom, (f) rotation+shift, (g) rotation+zoom, (h) shift+zoom, (i) rotation+shift+zoom, (j) cutout, and (k) mixup techniques.

Table 4: Data augmentation techniques and parameters used in the experiment.

Data Augmentation Techniques	Parameters
Rotation	[-170,170]
Brightness	[1, 5]
Width shift	[-0.2, +0.2]
Height shift	[-0.2, +0.2]
Zoom	[0.5, 1.5]
Fill mode	Reflect
Cutout	0.5
Mixup	0.4

Table 5: MobileNetV2 and NASNetMobile architectures on the leaf disease dataset using different data augmentation techniques.

Data Augmentation methods	MobileNetV2			NASNetMobile		
	Time	Scratch	Fine-Tuning	Time	Scratch	Fine-Tuning
Original image	2h 12m	63.08	93.08	4h 50m	68.08	92.31
Brightness	20h 15m	65.39	90.77	1d 11h 30m	66.92	89.23
Shift		74.62	90.77		75.39	93.08
Rotation		77.69	94.62		83.08	93.85
Zoom		77.69	**95.39**		64.62	93.01
Shift + Zoom		82.31	93.08		84.62	92.31
Rotation + Zoom		79.23	93.85		76.92	93.08
Rotation + Shift		79.23	**95.39**		77.69	**96.15**
Rotation + Shift + Zoom		77.69	90.77		81.54	95.39
Cutout		64.06	93.75		77.34	93.75
Mixup		61.71	89.84		67.18	92.18

Table 5 presents accuracy results and execution times for recognition using the leaf disease dataset. The results show that using the fine-tuning method always performs better than training from scratch (around 15-30%). Additionally, we examine the individual effect of each data augmentation Technique. The results of these comparisons show that the zoom technique is the best data augmentation, followed by the rotation technique. The highest recognition accuracy of 96.15% is obtained when combining the rotation and the shift techniques as the data augmentation and training with NASNetMobile architecture. On the other hand, it can be concluded that the brightness technique is an inappropriate data augmentation on the leaf disease dataset because this technique eliminates important information from an image. When comparing model size between two CNN architectures, the size of the model obtained by training with MobileNetV2 was 18MB, while NASNetMobile doubled the model size to 36MB.

5.3 *Experiments on iCassava 2019 Dataset*

In this experiment, we used 10-fold cross-validation in the training scheme. The standard deviation and accuracy were reported. We selected the data augmentation techniques; zoom, rotation+shift, and rotation+shift+zoom based on high accuracy results according to the experimental results from Table 5. The examples of the images generated from data augmentation techniques are shown in Figure 8. We performed two CNN architectures; MobileNetV2 and NASNetMobile, using the fine-tuning model with specific parameters; Epoch = 2000, Batch Size = 64, Learning Rate = 0.001, and Optimizer = Stochastic Gradient Descent (SGD) algorithm.

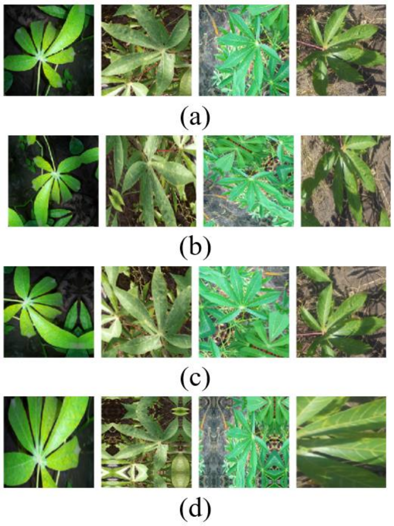


Figure 8: Examples of the iCassava 2019 dataset and samples of data augmentation images. (a) Original, (b) zoom, (c) rotation+shift, and (d) rotation+shift+zoom images.

In Table 6 we show the experimented results with the MobileNetV2 and NASNetMobile on the iCassava 2019 dataset. It can be seen from Table 6 that NASNetMobile architecture with combining rotation, shift, and zoom techniques is the best CNN architecture on the test set. The NASNetMobile outperforms the MobileNetV2 with around 1%. On the other hand, the MobileNetV2 obtained a slightly better result of around 0.9% than the NASNetMobile when testing on 10-fold cross-validation.
As for the computation time, it was found that the MobileNetV2 architecture was 2.25 times faster than the NASNetMobile architecture. Also, the model size of the MobileNetV2 is smaller than the NASNetMobile.

The average confusion matrices on 10-fold cross-validation are shown in Figure 10. The data augmentation technique is decreased misclassified. For recognition performance, the incorrect classification from CGM to CMD class is decreased from 19 to 11 images. Furthermore, the CMD class classifies as the CGM class decreased from 13 images to only 4 images. The results of the incorrect classification images are shown in Figure 9.

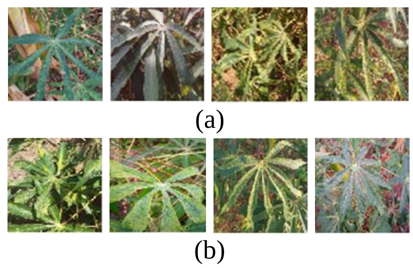


Figure 9: Examples of incorrect classification on the iCassava 2019 dataset. (a) The images of the CMD class that are classified as CGM class. (b) The images of the CGM class that are classified as CMD class.

Table 6: A Comparison of the performance of the MobileNetV2 and NASNetMobile architectures on the iCassava 2019 dataset.

Data Augmentation methods	MobileNetV2					NASNetMobile				
	Model Size	**Model Parameters**	**Time**	**10-cv**	**Test**	**Model Size**	**Model Parameters**	**Time**	**10-cv**	**Test**
Original	18 MB	2.26 m	12h 28 min	84.98 ± 1.75	81.33	36 MB	4.27 m	23h 26m	78.09 ± 2.75	74.65
Zoom			4d 20h	87.35 ± 0.14	80.11			9d 22h	86.95 ± 0.14	79.75
Rotation+Shift				88.55 ± 1.83	83.27				87.65 ± 0.56	83.98
Rotation+Shift+Zoom				88.94 ± 2.39	**83.62**				88.05 ± 1.12	**84.51**

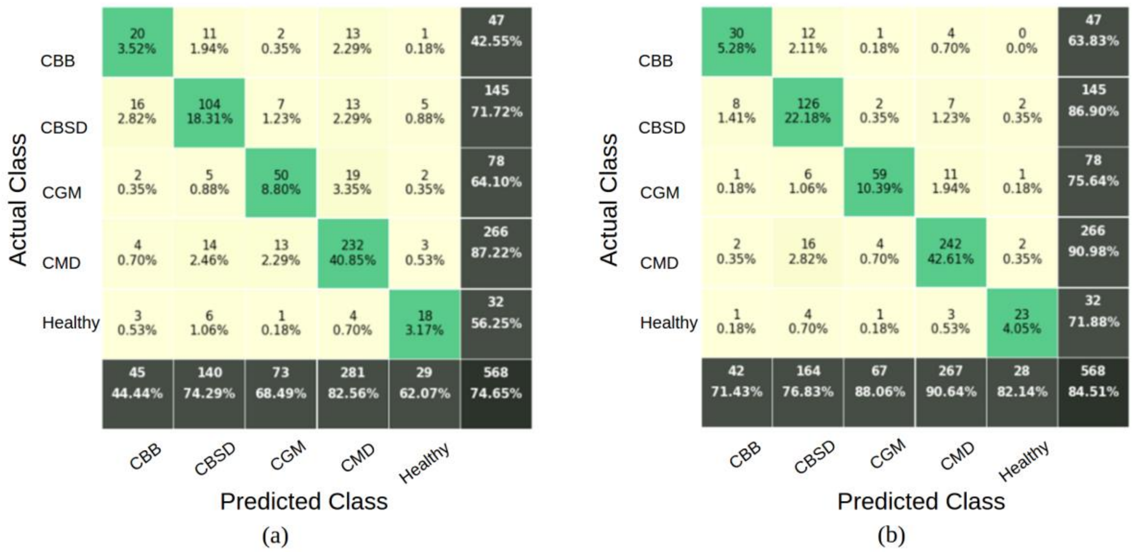


Figure 10: Confusion matrix of NASNetMobile architecture on the iCassava 2019 dataset. (a) The result of original data (b), and data augmentation using rotation, shift, and zoom techniques.

6 Conclusion

This research studied two deep convolutional neural networks (CNNs) proposed to create an efficient architecture and a small model that are suitable for smartphones and embedded devices and can be applied in a plant disease recognition system. In the experiment, we performed the CNN architectures on two plant disease datasets, consisting of the leaf disease and iCassava 2019 datasets. First, to find the best framework, we experimented with training techniques that allow CNN architectures to learn new data from various augmentation techniques. We evaluated the performance of the CNN architectures using several parameters. The best framework was the combination of the offline training technique and data augmentation techniques: rotation, shift, and zoom. On the contrary, the brightness technique that generated a plant leaf image by adding high-intensity values affected the plant leaf disease images by changing the white spots and the disease spots on the plant leaves. Hence, it is inappropriate for plant leaf disease recognition. Second, we propose to use two CNN architectures, called MobileNetV2 and NasNetMobile architectures, for plant leaf disease recognition. We are interested in a training scheme: fine-tuning and training from scratch, which obtains high recognition and requires less computation time. As a result, we found that the fine-tuning obtained better accuracy than training from scratch and decreased computation time. Consequently, MobileNetV2 architecture obtains a better result when the data augmentation technique is not applied. On the other hand, the NasNetMobile outperforms the MobileNetV2 when applied data augmentation.

In future work, we will concentrate on improving the performance of plant leaf disease recognition. We will study and apply other data augmentation techniques such as AutoAugment [33] and neural style transfer [34].

References

[1] J. Zhang, Y. Xie, Q. Wu, and Y. Xia, “Medical image classification using synergic deep learning,” *Med. Image Anal.*, vol. 54, pp. 10–19, May 2019, doi: 10.1016/J.MEDIA.2019.02.010.

[2] A. S. Lundervold and A. Lundervold, “An overview of deep learning in medical imaging focusing on MRI,” *Z. Med. Phys.*, vol. 29, no. 2, pp. 102–127, May 2019, doi: 10.1016/J.ZEMEDI.2018.11.002.

[3] R. J. Chalakkal, W. H. Abdulla, and S. S. Thulaseedharan, “Quality and content analysis of fundus images using deep learning,”

Comput. Biol. Med., vol. 108, pp. 317–331, May 2019, doi: 10.1016/j.compbiomed.2019.03.019.

[4] M. Talo, U. B. Baloglu, Ö. Yıldırım, and U. Rajendra Acharya, “Application of deep transfer learning for automated brain abnormality classification using MR images,” *Cogn. Syst. Res.*, vol. 54, pp. 176–188, May 2019, doi: 10.1016/J.COGSYS.2018.12.007.

[5] S. Javadi and S. A. Mirroshandel, “A novel deep learning method for automatic assessment of human sperm images,” *Comput. Biol. Med.*, vol. 109, pp. 182–194, Jun. 2019, doi: 10.1016/j.compbiomed.2019.04.030.

[6] Y. Lyu, J. Chen, and Z. Song, “Image-based process monitoring using deep learning framework,” *Chemom. Intell. Lab. Syst.*, vol. 189, pp. 8–17, Jun. 2019, doi: 10.1016/J.CHEMOLAB.2019.03.008.

[7] S. Zhou *et al.*, “Quick image analysis of concrete pore structure based on deep learning,” *Constr. Build. Mater.*, vol. 208, pp. 144–157, May 2019, doi: 10.1016/J.CONBUILDMAT.2019.03.006.

[8] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” *Comput. Electron. Agric.*, vol. 147, pp. 70–90, Apr. 2018, doi: 10.1016/J.COMPAG.2018.02.016.

[9] S. Chen, B. Li, J. Cao, and B. Mao, “Research on agricultural environment prediction based on deep learning,” *Procedia Comput. Sci.*, vol. 139, pp. 33–40, Jan. 2018, doi: 10.1016/j.procs.2018.10.214.

[10] S. Zhang, W. Huang, and C. Zhang, “Three-channel convolutional neural networks for vegetable leaf disease recognition,” *Cogn. Syst. Res.*, vol. 53, pp. 31–41, Jan. 2019, doi: 10.1016/J.COGSYS.2018.04.006.

[11] 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, Dec. 2017, doi: 10.1016/J.NEUCOM.2017.06.023.

[12] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, “Deep neural networks based recognition of plant diseases by leaf image classification,” *Comput. Intell. Neurosci.*, pp. 1–11, 2016, doi: 10.1155/2016/3289801.

[13] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, “Deep learning for image-based cassava disease detection,” *Front. Plant Sci.*, vol. 8, no. 2, pp. 1–7, Oct. 2017, doi: 10.3389/fpls.2017.01852.

[14] T. DeVries and G. W. Taylor, “Improved regularization of convolutional neural networks with cutout,” *arXiv Prepr. arXiv1708.04552.*, pp. 1–8, 2017, [Online]. Available: http://arxiv.org/abs/1708.04552.

[15] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “Mixup: Beyond empirical risk minimization,” Oct. 2017, [Online]. Available: http://arxiv.org/abs/1710.09412.

[16] P. Pawara, E. Okafor, O. Surinta, L. Schomaker, and M. Wiering, “Comparing local descriptors and bags of visualwords to deep convolutional neural networks for plant recognition,” in *6th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, 2017, pp. 479–486, doi: 10.5220/0006196204790486.

[17] Y. Sun, Y. Liu, G. Wang, and H. Zhang, “Deep learning for plant identification in natural environment,” *Comput. Intell. Neurosci.*, pp. 1–6, May 2017, doi: 10.1155/2017/7361042.

[18] L. Taylor and G. Nitschke, “Improving deep learning using generic data augmentation,” in *IEEE Conference on Symposium Series on Computational Intelligence (SSCI)*, Aug. 2018, pp. 1542–1547, [Online]. Available: http://arxiv.org/abs/1708.06020.

[19] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *J. Big Data*, vol. 6, no. 60, pp. 1–48, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[20] A. Mikołajczyk and M. Grochowski, “Data augmentation for improving deep learning in image classification problem,” in *International Interdisciplinary PhD Workshop (IIPhDW)*, 2018, pp. 117–122, doi: 10.1109/IIPHDW.2018.8388338.

[21] P. Pawara, E. Okafor, L. Schomaker, and M. Wiering, “Data augmentation for plant classification,” in *International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, 2017, vol. 10617 LNCS, pp. 615–626, doi: 10.1007/978-3-319-70353-4_52.

[22] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *3rd International Conference*

on Learning Representations (ICLR), Sep. 2015, pp. 1–14.
[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Dec. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
[24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.
[25] H. Y. Chen and C. Y. Su, "An enhanced hybrid MobileNet," in *International Conference on Awareness Science and Technology (iCAST)*, 2018, pp. 308–312, doi: 10.1109/ICAwST.2018.8517177.
[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Dec. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.
[27] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8697–8710, doi: 10.1109/CVPR.2018.00907.
[28] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv Prepr. arXiv1704.04861.*, pp. 1–9, 2017, [Online]. Available: http://arxiv.org/abs/1704.04861.
[29] K. C. Kamal, Z. Yin, M. Wu, and Z. Wu, "Depthwise separable convolution architectures for plant disease classification," *Comput. Electron. Agric.*, vol. 165, p. 104948, Oct. 2019, doi: 10.1016/j.compag.2019.104948.
[30] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv Prepr. arXiv1611.01578*, pp. 1–16, Nov. 2017, Accessed: Mar. 07, 2020. [Online]. Available: http://arxiv.org/abs/1611.01578.
[31] E. Mwebaze, T. Gebru, A. Frome, S. Nsumba, and J. Tusubira, "iCassava 2019 fine-grained visual categorization challenge," *arXiv Prepr. arXiv1908.02900*, pp. 4321–4326, 2019, [Online]. Available: http://arxiv.org/abs/1908.02900.
[32] A. K. Reyes, J. C. Caicedo, and J. E. Camargo, "Fine-tuning deep convolutional networks for plant recognition," in *Working notes of CLEF*, 2015, vol. 1391, pp. 1–9.
[33] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation strategies from data," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, May 2019, pp. 113–123, doi: 10.1109/CVPR.2019.00020.
[34] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: A review," in *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, Jun. 2019, pp. 1–25, doi: 10.1109/tvcg.2019.2921336.