Deep Convolutional Neural Network to Recognize Plant Leaf Images

Prem Enkvetchakul

A Thesis Submitted in Partial Fulfillment of Requirements for
degree of Doctor of Philosophy in Information Technology
January 2023

โครงข่ายประสาทแบบคอนโวลูชันเชิงลึกเพื่อรู้จำรูปภาพใบพืช

วิทยานิพนธ์

ของ

เปรม อิงคเวชชากุล

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

มกราคม 2566

Deep Convolutional Neural Network to Recognize Plant Leaf Images

Prem Enkvetchakul

A Thesis Submitted in Partial Fulfillment of Requirements
for Doctor of Philosophy (Information Technology)
January 2023

The examining committee has unanimously approved this Thesis, submitted by Mr. Prem Enkvetchakul , as a partial fulfillment of the requirements for the Doctor of Philosophy Information Technology at Mahasarakham University

Examining Committee

.... Chairman
(Prof. Rapeepan Pitakaso , Ph.D.)

.... Advisor
(Asst. Prof. Olarik Surinta , Ph.D.)

.... Committee
(Asst. Prof. Rapeeporn Chamchong , Ph.D.)

.... Committee
(Asst. Prof. Phatthanaphong Chompoowises , Ph.D.)

.... Committee
(Asst. Prof. Chatklaw Jareanpon , Ph.D.)

Mahasarakham University has granted approval to accept this Thesis as a partial fulfillment of the requirements for the Doctor of Philosophy Information Technology

....	....
(Assoc. Prof. Jantima Polpinij)	(Assoc. Prof. Krit Chaimoon , Ph.D.)
Dean of The Faculty of Informatics	Dean of Graduate School

TITLE	Deep Convolutional Neural Network to Recognize Plant Leaf Images		
AUTHOR	Prem Enkvetchakul		
ADVISORS	Assistant Professor Olarik Surinta , Ph.D.		
DEGREE	Doctor of Philosophy	**MAJOR**	Information Technology
UNIVERSITY	Mahasarakham University	**YEAR**	2023

ABSTRACT

This thesis focuses on three main types of research. - (1) data augmentation ensemble learning (2) learning rate schedules improved and (3) plant leaf disease recognition performance Chapter 1 provides a brief general introduction to deep learning for plant leaf disease recognition and uses deep learning techniques for detecting and diagnosing diseases in plants, followed by the research questions. The objectives of the dissertation and its contributions are described. In Chapter 2, Two deep convolutional neural networks (CNNs): MobileNetV2 and NasNetMobile are proposed to recognize plant leaf disease. I have experimented with training techniques; online, offline, and mixed training techniques on two plant leaf diseases which were the leaf disease dataset, and the iCassava2019 dataset. I have also used data augmentation techniques combining rotation, scrolling, and zooming techniques to further enhance the recognition performance. Chapter 3 presents the stacking ensemble of lightweight convolutional neural networks to improve the performance of the recognition of plant leaf disease images. I proposed a stacking ensemble of deep CNNs to evaluate three plant leaf disease datasets; PlantDoc, Crop-PlantDoc, and iCassava2019. We experimented with five classifiers that were logistic regression, support vector machine, K-nearest neighbors, random forest, and long short-term memory network. The random forest method achieved a more accurate performance. Chapter 4 proposes fusion and ensemble CNN to improve the performance of plant leaf disease recognition. The work reported in this thesis experimented with a new learning rate schedule, called equal learning rate range (ELRR) and step decay equal learning rate range (SD-ELRR), which is proposed and compared with two baseline learning rate [AP1] schedules. The proposed learning rate schedule was evaluated on two datasets: Cropped-PlantDoc and Plant Pathology. The results showed that the ELRR and SD-ELRR equations improved the efficiency of plant leaf disease recognition significantly better than the basic equations in the entire E plant disease dataset. Chapter 5 comprises two main sections: - 1) Answers to the research questions 2) Future work. This chapter briefly explains the proposed approaches and answers three main research questions in plant leaf disease recognition using deep learning techniques. Two main approaches are planned to be the focus of future work, as follows. For the data augmentation techniques, I plan to study and apply other data augmentation techniques such as AutoAugment and neural style transfer. For the ensemble learning techniques, I will focus on experiments with the other CNN

frameworks, such as snapshot ensemble CNN and 1D-CNN.

Keyword : Plant leaf disease recognition, Deep learning, Convolutional neural network (CNN), Lightweight CNN, Transfer learning, Data augmentation, Stacking ensemble learning method, Ensemble learning method, Meta-learner Method, Learning rate schedule, Deep fusion, Early stopping

ACKNOWLEDGEMENTS

This thesis was supported by the scholarships of Buriram Rajabhat University.The authors would like to thank the researchers and give special thanks to my families for the best encouragement and Asst. Prof. Dr. Olarik Surinta my advisor for his efforts toward the completion of this research.

Prem Enkvetchakul

TABLE OF CONTENTS

Table

Page

Figure

Page

Chapter 1
Introduction

Plant diseases are a significant problem affecting the quality and quantity of agricultural products for consumption, distribution, and export. If farmers cannot identify plant diseases in time, it will affect productivity and plant quality [1]. In general, farmers in underdeveloped countries may not have advanced devices to detect plant diseases and farmers rely on visual diagnosis by other experienced farmers. Diagnosis of plant leaf disease by experts may be expensive and require analysis in a laboratory. Sometimes, it takes much time to analyze, thereby allowing the plant disease to spread widely [2]. In this study, plant diseases that show leaf symptoms were divided into two main characteristics illustrated in the following figures: - Figure 1a) the stage of disease formation may be the initial stage or the stage where a disease is widely spread, and Figure 1b) some plant diseases have similar symptoms. If farmers lack the knowledge and fail to diagnose plant diseases, yields may be damaged.

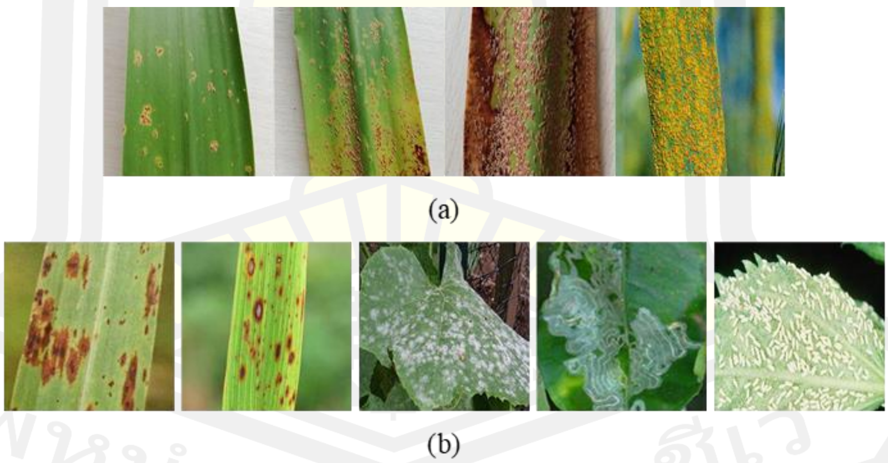


Figure 1 leaf symptoms were divided into two characteristics: (a) The stage of disease formation and (b) some plant diseases have similar symptoms.

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 [3], magnetic resonance imaging (MRI) [4], retinal image quality [5], brain abnormality classification [6], and sperm morphology analysis [7]. In the industrial arena, the deep belief network (DBN) is used in process monitoring systems that employ industrial process images [8] and also in concrete pore structure [9].

In the agricultural domain, advances in computer vision techniques have increased efficiency in monitoring and recognition [10], such as by detecting plant diseases, recognizing the types of diseases, and counting the number of plants. Further, deep learning methods are used in a large number of agricultural applications. Deep learning is proposed for use in conjunction with the internet of things (IoT) technology and unmanned aerial vehicles (UAV) [11] 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 [12]. Deep learning and IoT used in agriculture result in higher quality agricultural products and also a reduction in the cost of farming. Militante et al. [10] proposed computer vision and deep learning techniques for detecting and diagnosing diseases in plants. The proposed systems can take plant images using a camera and recognize diverse plant disease types. Zhong & Zhao [1] studied the significance of the deep learning method based on convolutional neural network (CNN) architecture to identify diseases that appeared on apple leaves.

The main objective of this thesis is to use deep convolution neural networks to recognize plant leaf images. This thesis suggests that benefit will arise from rapid leaf disease identification with high accuracy. In addition to helping to diagnose plant diseases at an early stage in order to treat, prevent or control the spread of plant diseases, it will also help farmers who lack expertise in growing crops or who lack amenities to avoid the costs and the time required for contacting specialists who by just seeing the symptoms expressed on the leaves of the plant are able to identify plant diseases immediately. The conclusions from this thesis will also help support farmers to manage plant diseases by using biological methods, replacing the use of expensive chemicals that pose a danger to users, consumers, and the environment. Knowledge of plant diseases helps with understanding the causes, allowing biological methods to more effectively prevent and protect from environmental infections, thus better preventing plant disease.

1.1 Research question

The main problems that directly affects agricultural products are abnormalities caused by plant diseases and insect pests. Farmers must have the knowledge and expertise to diagnose or solve problems in order to prevent and resolve them quickly and avoid spreading the infection to a broader area. If farmers lack the knowledge and fail to diagnose plant diseases, it may damage yields. Therefore, I aim to diagnose plant diseases early in order to treat, prevent or control the spread of plant diseases.

RQ1: Generally, convolutional neural networks take a large amount of data to learn to build an effective model. To avoid overfitting, most of the time, data collection problems are encountered. Therefore, there is an idea to create new data based on the existing data, called Data Augmentation. There are several methods employed in this such as Rotation, Brightness, Shift, Zoom, Cutout, and Mixup. Data Augmentation techniques can be divided into three processes, online, offline, and mixed. In addition, can using convolutional neural networks in combination with learning techniques and data augmentation help increase the efficiency of plant leaf image recognition performance?

RQ2: Ensemble learning is a combination of different models, and independent of several models together to increase the efficiency of the model. Models are divided into unweighted majority vote, unweighted average, and stacking ensemble. Therefore, can I use convolutional neural networks combined with ensemble learning to increase the efficiency of plant leaf image recognition? Is Stacking Ensemble suitable for improving plant leaf image recognition? Because of stacking the output probabilities of each CNN model and providing it as output to train to create the second model using the machine learning classifier, do the number of models used in collective learning and the classification method make the Stacking Ensemble method more efficient?

RQ3: The special aspect related to neural network learning is hyperparameters, For example, the learning rate, momentum, and activation function. The most critical variable is the learning rate. Therefore, can I use the learning rate schedules when training deep learning neural networks to improve plant leaf image recognition? I created a new equation to compare it with the original equation to see which gives better performance.

In order to answer all of these questions (RQ1 to RQ3), Chapter 2 to Chapter 4, describe the research done in this thesis. Finally, Chapter 5 provides concrete answers to research questions.

1.2 The objective of this dissertation

This study will focus on three detailed objectives:

1) Improve plant leaf image recognition by data augmentation and training techniques.

2) Improve plant leaf image recognition by ensemble learning.

3) Improve plant leaf image recognition by learning rate schedules.

1.3 Contribution

The main contribution was intended to be to improve the accuracy performance of the deep learning method for plant leaf disease recognition. I performed experiments on four plant leaf disease datasets consisting of iCassava 2019, PlantDoc, and Crop-PlantDoc. The contributions of the thesis are as follows.

Chapter 2. 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. The performance of the deep learning method is improved when combining data augmentation techniques and training techniques. In this thesis, the image manipulation techniques consisting of width and height shift, rotation, zoom, brightness, cutout [13], and mixup [14] are used. I also test using three training techniques; offline, online, and mixed methods. I examine the proposed deep learning method on two sets of plant leaf disease data; the leaf disease and iCassava 2019 datasets. I 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.

Enkvetchakul and Surinta [15], "Effective data augmentation and training techniques for improving deep learning in plant leaf disease recognition", *Appl. Sci. Eng. Prog.*, vol. 15, no. 3, pp. 1–12, Jul. 2022.

In chapter 3, I proposed a stacking ensemble of deep CNNs to evaluate three plant leaf disease datasets; PlantDoc, Crop-PlantDoc and iCassava2019. In the first process, I proposed to use four CNN architectures; InceptionResNetV2, NASNetMobile, MobileNetV2, and EfficientNetB1, to train on the plant leaf disease images accordingly to obtain the fittest CNN model that applies in the meta-learner process. In the second process, in the meta-learner process, I applied the output probabilities obtained from the fittest CNN models as inputs of a classifier. I employed five classifiers consisting of logistic regression (LR), support vector machine (SVM), K-nearest neighbors (KNN), random forest (RF), and long short-term memory (LSTM) network. Finally, the proposed stacking ensemble was integrated with the best CNN model from the first process and the classifier from the second process to recognize and evaluate the plant leaf disease images.

Enkvetchakul and Surinta [16], "Stacking ensemble of lightweight convolutional neural networks for plant leaf disease recognition", *ICIC Express Lett.*, vol. 16, no. 5, pp. 521–528, 2022.

In chapter 4, I present the optimization of learning rate schedules in deep learning to increase efficiency in plant leaf disease recognition, and use early stopping to determine the optimal time to use in plant leaf disease recognition. Finally, I enhanced the recognition efficiency by fusion and ensemble learning to evaluate two plant leaf disease datasets, Cropped-PlantDoc and Plant Pathology. As a first step, I used three CNN architectures, EfficientnetB1, MobileNetV2, and NASNetMobile, using a method to optimize recognition by adjusting learning rate schedules during learning with two basic equations; time-based decay and step decay; and two self-improved equations called equal learning rate range (ELRR) and step decay-equal learning rate range (SD-ELRR). In step 2: I performed the same process as the first step, but used the early stopping method to help stop learning to compare time together with the performance of the model, and then the final step selects the best CNN from two basic equations and two self-made equations for Deep Fusion by deep feature extraction from both CNNs. In addition, the CNN mentioned above was used for ensemble learning in 3 methods; unweighted majority vote, unweighted average, and weighted average. Finally, I compared the performance achieved by the fundamental equations with the newly created equations and the fusion efficiency with ensemble learning.

Chapter 2
Data augmentation and training techniques for deep learning

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 paper, I 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. I have experimented on training techniques; online, offline, and mixed training techniques on two plant leaf diseases. As for data augmentation techniques, I 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.

2.1 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 [3], magnetic resonance imaging (MRI) [4], retinal image quality [5], brain abnormality classification [6], and sperm morphology analysis [7]. In the industrial arena, the deep belief network (DBN) is used in the process monitoring process employing industrial process images [8] and concrete pore structure [9].

In agriculture, deep learning is proposed for use in conjunction with the internet of things (IoT) technology and unmanned aerial vehicles (UAV) [11] 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 [12]. 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 [17]–[20]. Furthermore, most plant diseases can be identified by leaf.

This chapter studies deep learning that can be used in plant leaf disease recognition system. The contributions of this chapter can be summarized as follows:

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 [13], and mixup [14] are used. I also test on three training techniques, including offline, online, and mixed methods.

3) I examine the proposed deep learning method on two sets of plant leaf disease data: the leaf disease and iCassava 2019 datasets.I 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.

2.2 Related Work

2.2.1 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. [21] 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 [17], cassava [20], tomato, and cucumber leaf diseases.

Ramcharan et al. [20] experimented on the cassava disease dataset using Inception v3. This CNN architecture obtained an accuracy of 93%. Lu et al. [18] 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. [17] 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. [22] 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.

2.2.2 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 [23], [24]. Hence, the idea of generating new data based on existing data, which is called data augmentation, was proposed. Taylor and Nitschke [23] 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 [24] 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 [25] 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. [26] 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.

2.3 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 [27] was designed to have 16 and 19 layer. The layer of the ResNet [28] is 18, 34, 50, 101, and 152 layers. Also, the layer of the DenseNet [29] 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 [30].

This caption aims to study the CNN architectures that can create a small and efficient model suitable for smartphones comprising MobileNetV2 [31] and NASNetMobile [32].

2.3.1 MobileNetV2 Architecture

Howard et al. [33] 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 2.

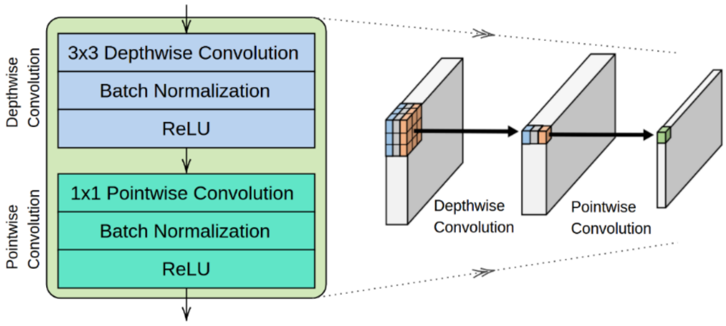


Figure 2: MobileNets with the depthwise separable convolutions process, which consists of depthwise convolution and pointwise convolution. The batch normalization layer and the rectified linear unit are added at the end of every convolutional layer [33], [34].

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% [33] while the GoogLeNet obtained the accuracy of 69.8%

Sandler et al. [31] 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 3. The linear bottlenecks, which is an increase in the number of the feature map, such as ResNet [28] increases a feature map from 64 to 128, 256, and 512, respectively. Figure 3 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.

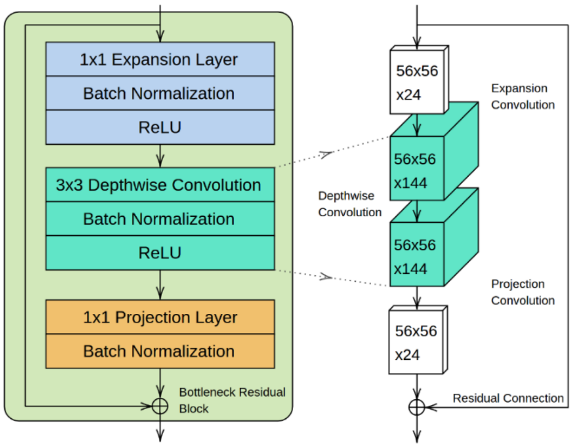
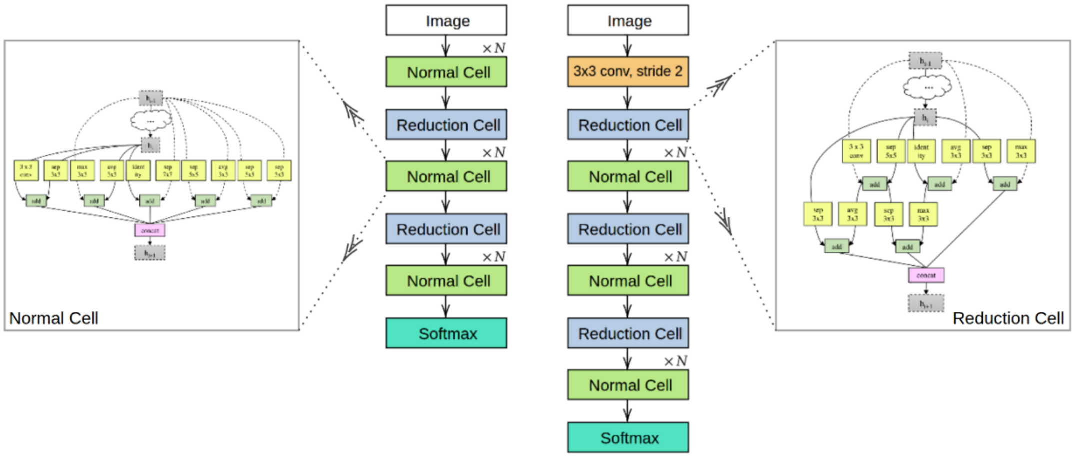


Figure 3: MobileNetV2 with inverted residuals. Process for making linear bottlenecks with the increase in feature map from 24 maps to 144 maps and the reduction of feature map from 144 maps to 24 maps [31].

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 [31].

2.3.2 NASNetMobile Architecture

Zoph and Le [32] 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. [35] 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 4 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 4 Scalability of NASNet designed for use with (b) CIFAR10 dataset and (c) ImageNet dataset and examples of (a) normal cell and (d) reduction cell [32].

Figure 4 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 5.

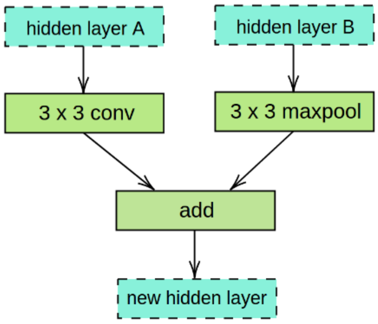


Figure 5: Block of convolution cell obtained from searching with RNN [32].

2.4 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.

2.4.1 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 6(k)) and woolly aphid (Figure 6(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 6.

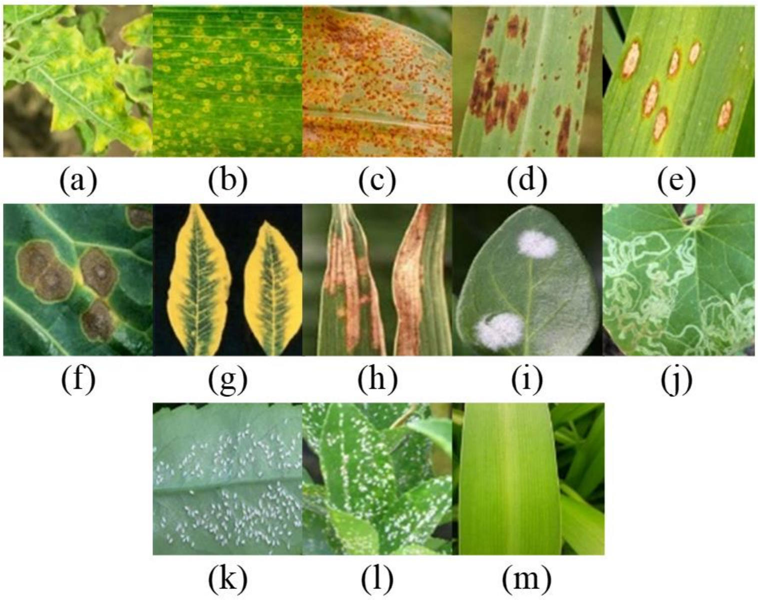


Figure 6: Sample images from leaf disease dataset, which consists of 13 classes consisting of (a) mosaic disease, (b) yellow leaf spot disease, (c) rust diseases, (d) narrow brown spot disease, (e) brown spot disease, (f) ringspot disease, (g) plant nutrient deficiencies, (h) leaf scald disease, (i) 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		

2.4.2 iCassava2019 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 [36] 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, I 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 7.

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

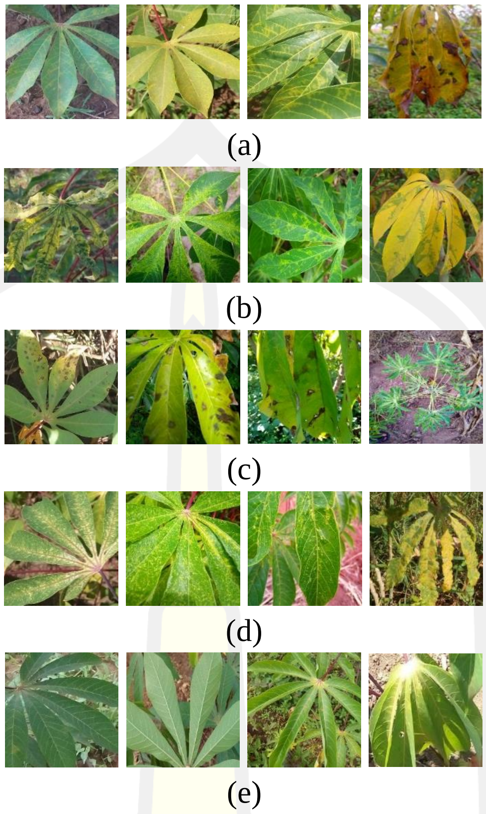


Figure 7 Examples of five types of iCassava 2019 dataset used in the experiment, consisting of (a) cassava brown streak disease, (b) cassava mosaic disease, (c) cassava bacterial blight, (d) cassava green mite, and (e) Healthy.

2.5 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.

2.5.1 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, I selected MobileNetV2 and trained the architecture using the fine-tuning technique [37]. Second, to demonstrate the performance of the training technique, I 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 [24]. 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, I 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, I 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, I 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%.

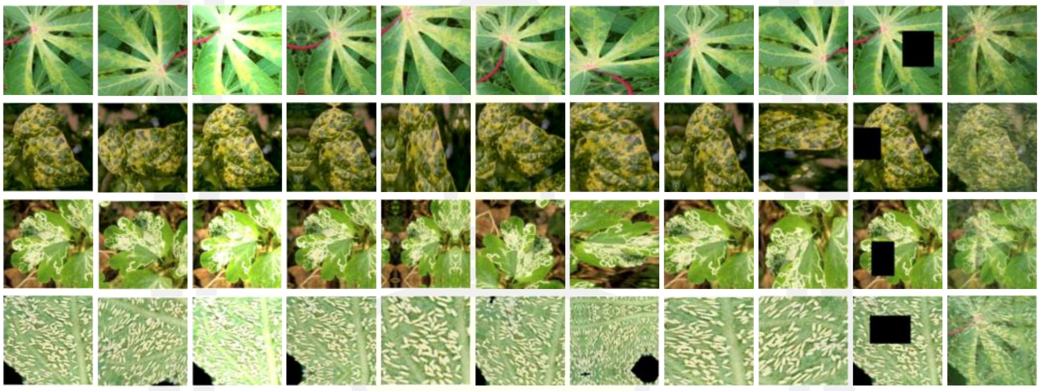
As can be seen from the result in Table 3, it can be concluded that data augmentation has a direct effect on increasing recognition accuracy. Hence, I choose the offline training and data augmentation (15-image) technique in the following experiments.

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

2.5.2 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, I 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 8.



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

Figure 8 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, I 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.

2.5.3 Experiments on iCassava 2019 Dataset

In this experiment, I used 10-fold cross-validation in the training scheme. The standard deviation and accuracy were reported. I 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 9. I 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.

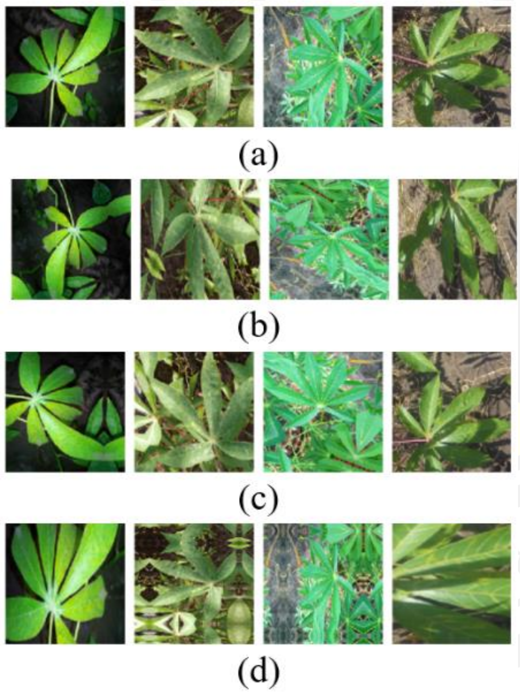


Figure 9 Examples of the iCassava 2019 dataset and samples of data augmentation images. (a) Original, (b) zoom, (c) rotation+shift, and (d) rotation+shift+zoom images.

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

<table>
<tr><th rowspan="2">Data Augmentation methods</th><th colspan="5">MobileNetV2</th><th colspan="5">NASNetMobile</th></tr>
<tr><th>Model Size</th><th>Model Parameters</th><th>Time</th><th>10-cv</th><th>Test</th><th>Model Size</th><th>Model Parameters</th><th>Time</th><th>10-cv</th><th>Test</th></tr>
<tr><td>Original</td><td rowspan="4">18 MB</td><td rowspan="4">2.26 m</td><td>12h 28 min</td><td>84.98 ± 1.75</td><td>81.33</td><td rowspan="4">36 MB</td><td rowspan="4">4.27 m</td><td>23h 26m</td><td>78.09 ± 2.75</td><td>74.65</td></tr>
<tr><td>Zoom</td><td rowspan="3">4d 20h</td><td>87.35 ± 0.14</td><td>80.11</td><td rowspan="3">9d 22h</td><td>86.95 ± 0.14</td><td>79.75</td></tr>
<tr><td>Rotation+Shift</td><td>88.55 ± 1.83</td><td>83.27</td><td>87.65 ± 0.56</td><td>83.98</td></tr>
<tr><td>Rotation+Shift+Zoom</td><td>88.94 ± 2.39</td><td>83.62</td><td>88.05 ± 1.12</td><td>84.51</td></tr>
</table>

In Table 6 I 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 11.

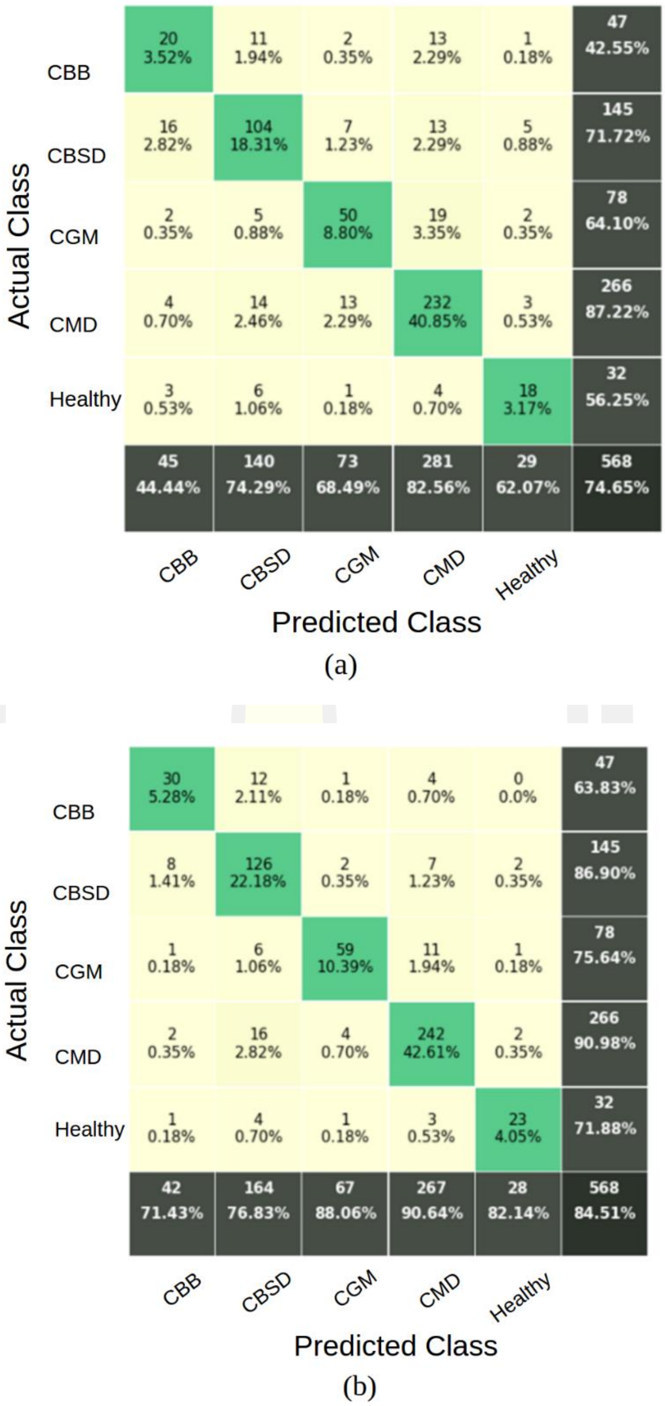
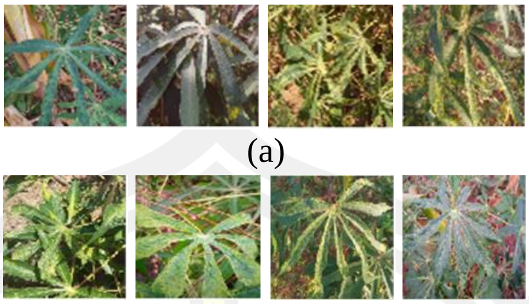


Figure 10 Confusion matrix of NASNetMobile architecture on the iCassava 2019 dataset. (a) The result of original data (b), and data augmentation using rotation, shift, and zoom techniques



(a)

(b)

Figure 11 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.

2.6 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, I performed the CNN architectures on two plant disease datasets, consisting of the leaf disease and iCassava 2019 datasets. First, to find the best framework, I experimented with training techniques that allow CNN architectures to learn new data from various augmentation techniques. I 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, I propose to use two CNN architectures, called MobileNetV2 and NasNetMobile architectures, for plant leaf disease recognition. I are interested in a training scheme: fine-tuning and training from scratch, which obtains high recognition and requires less computation time. As a result, I 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, I will concentrate on improving the performance of plant leaf disease recognition. I will study and apply other data augmentation techniques such as AutoAugment [38] and neural style transfer [39].

Chapter 3
Stacking Ensemble of Lightweight Convolutional Neural Networks

The high-grade quality of agricultural goods can be affected by diseases. Therefore, farmers need to quickly stop the spread of diseases. This study proposes a stacking ensemble of lightweight learning convolutional neural network (CNN) framework to enhance the recognition accuracy of plant leaf disease images. In the proposed framework, I first planned four lightweight CNN architectures (InceptionResNetV2, NASNetMobile, MobileNetV2, and EfficientNetB1) to train and create robust CNN models from images of plant leaf diseases. The experimental results showed that the EfficientNetB1 outperformed other CNN models. I then created the stacking ensemble learning by stacking the output probabilities of each CNN model and provided as output to train to create the second model using the machine learning classifier. In this step, I experimented with five classifiers that were logistic regression, support vector machine, K-nearest neighbors, random forest, and long short-term memory network. I found that the random forest method achieved a more accurate performance. As a result, I considered that all machine learning techniques could be involved in stacking ensemble learning.

3.1 Introduction

Plant diseases are a significant problem affecting the quality and quantity of agricultural products for consumption, distribution, and export. If the farmer cannot identify the plant disease in time, it will affect productivity and plant quality [1]. In general, farmers in underdeveloped countries may not have advanced devices to detect the plant diseases. However, the farmers rely on visually diagnosis by other experienced farmers. Diagnosis of the plant leaf disease by experts may be expensive and require analysis in a laboratory. Sometimes, it takes much time to analyze allowing the plant disease to spread all over the area [2].

More recently, the advances in computer vision techniques have increased efficiency in monitoring and recognition in the agricultural domain [10], such as by detecting the plant disease, recognizing the types of the disease, and counting the number plants. Further, deep learning methods are used in a large number of agricultural applications. These approaches allow farmers to work faster and also save energy while doing agriculture tasks. Militante et al. [10] proposed computer vision and deep learning techniques for detecting and diagnosing diseases in plants. The proposed systems can take plant images using a camera and recognize diverse plant disease types. Zhong & Zhao [1] studied the significance of the deep learning method based on convolutional neural network (CNN) architecture to identify the diseases that appeared on the apple leaves.

The deep learning methods were performed to improve the recognition of plant leaf disease images. However, using only a single deep learning model may not be sufficient to increase the accuracy performance of the plant leaf recognition systems. Furthermore, using ensemble learning with multiple deep learning models can reduce the variance of the recognition errors and improve plant leaf recognition systems [4,5]. For example, Khanramaki et al. [42] proposed the ensemble CNNs to recognize three common citrus pests; citrus leafminer, sooty mold, and Pulvinaria. For the single deep learning model, it achieved an accuracy of 96.05% with the Resnet50 architecture. As a result, the ensemble learning models provided an accuracy of 99.04%.

Contribution. This caption aims to improve the accuracy performance of the deep learning method for plant leaf disease recognition. I proposed a stacking ensemble of deep CNNs to evaluate three plant leaf disease datasets; PlantDoc, Crop-PlantDoc and iCassava2019. In the first process, I proposed to use four CNN architectures; InceptionResNetV2, NASNetMobile, MobileNetV2, and EfficientNetB1, to train on the plant leaf disease images accordingly to obtain the fittest CNN model that applies in the meta-learner process. In the second process, in the meta-learner process, I applied the output probabilities obtained from the fittest CNN models as inputs of a classifier. I employed five classifiers consisting of logistic regression (LR), support vector machine (SVM), K-nearest neighbors (KNN), random forest (RF), and long short-term memory (LSTM) network. Finally, the proposed stacking ensemble was integrated with the best CNN model from the first process and the classifier from the second process to recognize and evaluate the plant leaf disease images.

Outline of the caption. This caption is organized in the following way. Section 2 presents the proposed stacking ensemble of convolutional neural networks. The experimental settings and results are explained in Section 3. The conclusion is presented in Section 4.

3.2 Related work

Ensemble learning methods have been proposed in many applications. Mahmoud & Yaroshchak [40] proposed a bagging ensemble to classify diabetic retinopathy images containing 2,781 images. First, the training set was randomly selected for three subsets. Second, the subsets were sent to learning using three different CNN architectures. For the ensemble learning method, finally, the weighted average was used. The result showed that the bagging ensemble with three InceptionV3 models obtained an accuracy of 87.2%. In [43], the stacked CNN was proposed to diagnose COVID-19 disease from X-ray Images. Two CNN models, including the fine-tuning of VGG19 and CovNet30, were proposed to learn from the chest X-ray images. The outputs of the CNN models were stacked and logistic regression classifier was applied to classify three classes of COVID-19, consisting of COVID-19, pneumonia, and normal. It was performed with an accuracy of 92.47% on the chest X-ray images of the COVID-19 dataset.

Ju et al. [41] proposed a super learner that was based on the ensemble learning method. The super learner method achieved the best prediction accuracy on the test set of the CIFAR-10 dataset compared to other ensemble methods. In this method, the CNN models (including network in network, GoogLeNet, VGG Network, and Residual network), called base learners, were trained from the training set with mini-batch size. The outputs of each CNN network were then combined and followed by convolution with the size of 1×1. The convolution layer was trained using the validation to avoid overfitting. Further, the output of the super learner method was the score vector.

In addition, Kim et al. [44] invented an automatic defect classification on the TFT-LCD panel using the stacking ensemble method. In this method; first, the sliding window was slide through the TFT-LCD panel and then sent to the particular area to extract the deep features using distinct nine deep models. Second, the neural network was proposed as a weak learner that learns from the deep features transferred from the deep models. Third, for the stacking ensemble, the prediction scores of each learner were decided using the ensemble learning method; majority vote and score average. Finally, the outputs of the final prediction were four types of defects.

For the plant recognition using the ensemble learning method, Darwish et al. [2] proposed to use the particle swarm optimization (PSO) algorithm to optimize hyperparameters of the VGG16 and VGG19 networks. In this method, first, the optimal VGG networks were used to extract the deep features from the plant disease images. It froze the last convolution layer of the VGG networks and combined them. Second, the new convolution layers, such as flatten, dropout, batch normalization, and dense, were added to combined networks. Finally, average ensemble learning was used to predict the diseases of plant leaf images.

Chompookham & Surinta [45] invented ensemble CNN architectures to improve plant leaf classification performance. In this method, five CNN architectures were trained on the plant leaf images to create robust CNN models. After that, three best CNNs models were then combined, the output probabilities of each CNN model were assigned to classify using the ensemble methods; unweighted majority vote, unweighted average, and weighted average. The best ensemble method used in this experiment was the weighted average method. It outperformed all the ensemble methods on three plant leaf datasets; mulberry, tomato, and corn.

3.3 Proposed Stacking Ensemble of Convolutional Neural Networks

This section introduces the stacking ensemble of CNNs to recognize the plant leaf disease images, as shown in Figure 12.

In the first level. I find the baseline CNN models from various CNN models; InceptionResNetV2, NASNetMobile, MobileNetV2, and EfficientNetB1. Second, I stack CNN models and train on the plant leaf disease dataset. Subsequently, the output probabilities of each CNN model are used as the input of the machine learning technique.

In the second level. The machine learning techniques; LR, SVM, KNN, RF, and LSTM, are proposed to train from the output probability of the CNN models and obtain the final prediction, called the meta-learner method.

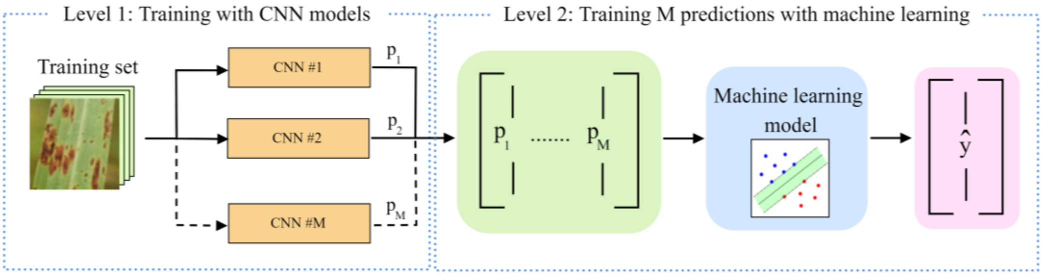


Figure 12 Illustration of the proposed stacking ensemble of lightweight CNNs

3.3.1 Convolutional Neural Network (CNN) Architectures

InceptionResNetV2. Szegedy et al. [46] proposed a new network architectures that contained the concept of Inception architecture and residual Inception blocks, called InceptionResNetV2. The Inception networks were designed as a tuning network based on InceptionV4, and they were allowed to change the number of filters in the several layers. The Inception block was designed to add the filter-expansion layer for the residual Inception blocks. Hence a 1×1 convolution layer without activation function was used for scaling up the filter dimension.

NASNetMobile. Zoph et al. [32] invented a neural architecture search network (NASNet) to address the expensive computation time while training on the large dataset. First, the NASNet architecture was proposed to search for an optimal architecture building block on a small dataset using reinforcement learning. Second, the building blocks were transferred to learn on a large dataset. The NASNet architecture consisted of two cells; a normal cell and a reduction cell. It was easy to build the NASNet because the normal and reduction cells were stacked and repeated many times. The last layer was the normal cell, followed by the softmax function. In addition, to create the NASNetMobile model, the size of the normal and reduction cells and the number of filters were decreased. The parameters of the NASNetMobile are smaller than the NASNet approximately nine times.

MobileNetV2. MobileNetV2 was designed by Sandler et al. [47] in 2018. It was the extended version of the MobileNetV1. MobileNetV2 contained three main layers; depthwise separable convolutions, linear bottlenecks, and inverted residuals. These layers performed to reduce the number of parameters and computation time when compared with MobileNetV1. In addition, MobileNetV2 was trained using the ReLU6 activation function, allowing it to learn complex patterns in the input data.

EfficientNetB1. EfficientNet was proposed by Tan & Le [48]. It involved scaling the network using four methods; width, depth, resolution, and compound scaling. It was comfortable to scale up a baseline CNN to any purpose resource limitations. Our experiment proposed using EfficientNetB1 to classify the plant leaf datasets because it had parameters with 7.8M. The parameters of EfficientNet-B1 were fewer than the DenseNet-169, Xception, Inception-v3, and even ResNet-50.

3.3.2 Meta-Learner Method

In our proposed method, the stacking ensemble of CNNs contained two levels; training with CNN models and with machine learning. The second level is called the meta-learner method. It usually trains the machine learning model using the output probabilities (p) from the first level and predicts the final output ($\hat{y}$). In our framework, the output probabilities of the CNN models were computed using the softmax function.

3.4 Dataset

In this research, the accuracy of deep learning was experimented on three datasets of leaf diseases, consisting of the PlantDoc dataset, Crop-PlantDoc dataset and iCassava 2019 dataset.

3.4.1 PlantDoc dataset

The PlantDoc dataset contained 2,567 images from 13 plant species collected from the internet. It included 27 classes of plant leaf disease and of healthy leaf [49]. Examples of the PlantDoc dataset are shown in Figure 13

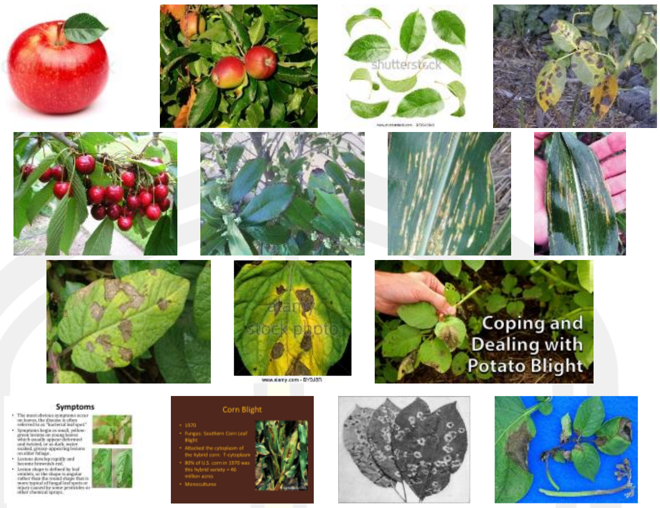


Figure 13 Illustration of PlantDoc dataset

3.4.2 Crop-PlantDoc dataset

The Crop-PlantDoc dataset is the extended version of the PlantDoc dataset. Singh et al. [49] also provided the ground truth of all images intending to crop all leaves, as shown in Figure 14. After cropping all the leaves, the Crop-PlantDoc dataset contained 8,883 images.

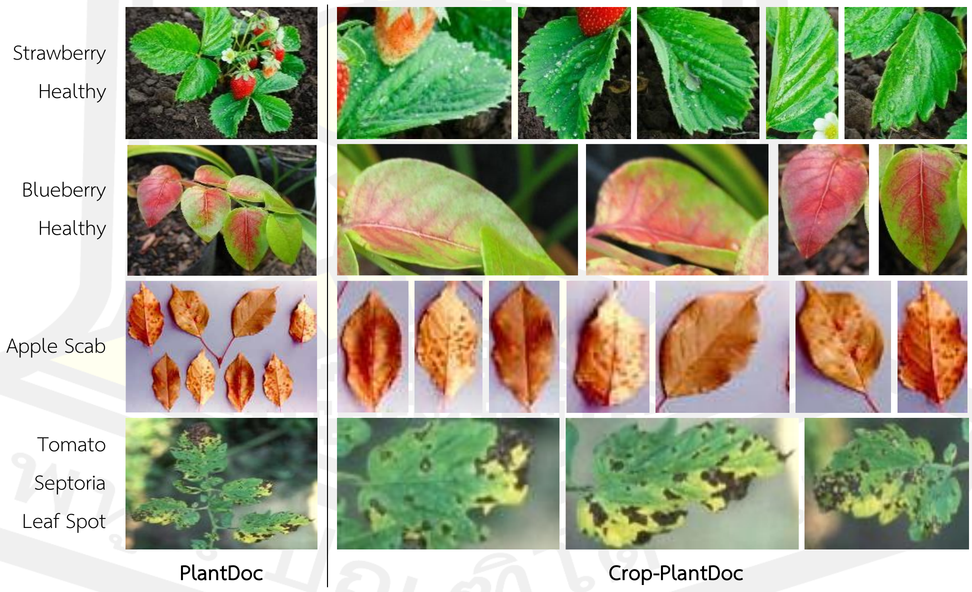


Figure 14 Illustration of Crop-PlantDoc dataset

3.4.3 iCassava2019 dataset

The iCassava2019 dataset was published on the Kaggle website. It contained 5,656 images of five cassava leaf states, including four types of disease and one healthy type, as shown in Figure 15. All cassava leaf disease images were taken from farmers in Uganda and verified by the experts of the National Crops Resources Research Institute (NaCRRI) and AI lab in Makerere University, Kampala [36].

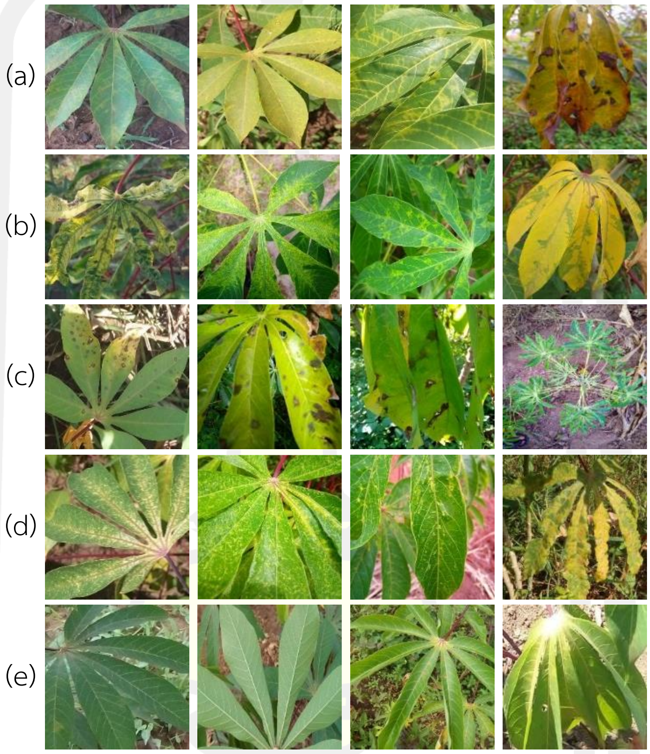


Figure 15 Illustration of iCassava2019 dataset

3.5 Experimental Setting and Results

In this paper, I divided the experiment into two parts to identify the best convolutional neural network (CNN) model and classifier to develop into the stacking ensemble of the CNN framework. In the first part, I mainly concentrated on the lightweight convolutional neural networks (CNNs). I then experimented and compared four CNN models; InceptionResNetV2, NASNetMobile, MobileNetV2, and EfficientNetB1. I applied the transfer learning method that employed the pre-trained model of four CNN models. The CNN models were trained with the following parameters; 100 epochs, batch size = 16, learning rate = 0.01, and stochastic gradient descent (SGD) optimizer.

In the second part, I examined the experiments of the ensemble learning method and the meta-learning method. For the ensemble learning method, two ensemble methods were studied; unweighted majority vote and unweighted average. In addition, in the meta-learning method, four machine learning techniques were proposed, including logistic regression (LR), support vector machine (SVM), K-nearest neighbors (KNN), random forest (RF), and long short-term memory (LSTM) network.

The plant leaf disease datasets were split into training, validation, and test. The ratio of PlantDoc and Crop-PlantDoc were 60%-20%-20% and iCassava2018 with the ratio of 80%-10%-10%.

In the experimental setting, I used the TensorFlow library as a deep learning framework running on Ubuntu operating system version 18. All experiments were evaluated with Intel(R) Core-i5, 2320 CPU @ 3.00GHz, 16GB RAM, and GPU NVIDIA GeForce GT 1060Ti.

3.5.1 Experiments on lightweight convolutional neural networks

In this experiment, the pre-trained models of four CNNs consisting of InceptionResNetV2, NASNetMobile, MobileNetV2, and EfficientNetB1, were trained on the plant leaf disease datasets. The data augmentation techniques [50], including rotation, shift, zoom, and horizontal flip, were combined in this study. In order to determine the average accuracy and standard deviation on the validation set, I randomly selected the training and validation tests and evaluated them ten times.

In Table 7, I show the experimental results with the data augmentation techniques and lightweight CNN models. The experiments indicated that the most significant performance on plant leaf disease datasets was with the EfficientNetB1 model. It outperformed on both validation and test sets. In addition, the InceptionResNetV2 model showed the second-best performance on the Crop-PlantDoc and iCassava2019 datasets. It also showed significant performance when compared with MobileNetV2 and NASNetMobile models.

Table 7 Performance evaluation of the lightweight CNNs and data augmentation techniques on plant leaf disease datasets

CNN Architectures	PlantDoc		Crop-PlantDoc		iCassava2019	
	Validation (%)	Test (%)	Validation (%)	Test (%)	Validation (%)	Test (%)
EfficientNetB1	68.33±1.95	**67.70**	85.19±0.96	**86.21**	87.01±0.82	**88.25**
InceptionResNetV2	63.15±2.74	57.98	78.20±3.55	81.37	86.40±0.83	87.28
MobileNetV2	61.19±2.25	58.56	74.51±1.08	74.96	82.49±1.43	84.45
NASNetMobile	60.33±2.29	57.59	79.02±0.69	76.48	80.72±2.57	84.10

3.5.2 Experiments on ensemble learning methods

In this experiment, to find the optimal numbers of the model, I examined the performance of ensemble CNNs from two to ten models. For the ensemble learning method, I classified the output using an unweighted majority vote and unweighted average methods.

Table 8 Performances of the ensemble learning methods and lightweight CNNs

CNN Architectures	Ensemble Methods	Evaluation Metrics	Plant Doc	Crop-PlantDoc	iCassava 2019
EfficientNetB1	Unweighted Majority Vote	Accuracy (%)	70.04	86.21	91.34
		Testing time (sec.)	0.12	0.20	0.71
		No. of CNN models	2	8	10
	Unweighted average	Accuracy (%)	**70.82**	**90.55**	**91.61**
		Testing time (sec.)	0.13	0.20	0.71
		No. of CNN models	2	8	10
MobileNetV2	Unweighted Majority Vote	Accuracy (%)	66.73	85.20	88.34
		Testing time (sec.)	0.29	0.17	0.34
		No. of CNN models	7	9	8
	Unweighted average	Accuracy (%)	67.15	85.20	88.87
		Testing time (sec.)	0.41	0.17	0.38
		No. of CNN models	10	9	9
NASNetMobile	Unweighted Majority Vote	Accuracy (%)	68.68	88.35	8781
		Testing time (sec.)	1.29	0.52	1.56
		No. of CNN models	8	9	9
	Unweighted average	Accuracy (%)	68.68	88.69	88.07
		Testing time (sec.)	1.45	0.58	1.56
		No. of CNN models	9	10	9
InceptionResNetV2	Unweighted Majority Vote	Accuracy (%)	68.87	80.19	90.28
		Testing time (sec.)	1.20	0.46	1.94
		No. of CNN models	7	2	9
	Unweighted average	Accuracy (%)	69.46	80.30	90.64
		Testing time (sec.)	1.20	0.46	1.94
		No. of CNN models	7	2	9

Table 8 provides accurate results, testing times, and numbers of CNN models for recognition on three plant leaf disease datasets. The experimental results show that the ensemble CNNs with both unweighted majority vote and unweighted average methods performed consistently better than single CNN. Furthermore, when I focused on the ensemble learning methods, the unweighted average method slightly outperformed the unweighted majority vote. Consequently, the EfficientNetB1 still significantly outperformed other CNNs on all datasets. Surprisingly, the ensemble CNNs combined with two EfficientNetB1 models achieved an accuracy of 70.82% on the PlantDoc dataset. It increased the accuracy of one EfficientNetB1 model by approximately 2%.

3.5.3 Experiments on stacking ensemble learning method

In this experiment, the stacked output probabilities of CNN models were trained using the machine learning methods; logistic regression (LR), support vector machine (SVM), K-nearest neighbors (KNN), random forest (RF), and long short-term memory (LSTM) network. First, I fine-tuned the hyperparameters of each classifier. The hyperparameters applied to each classifier were as follows. SVM, C=1, gamma=0.1, kernel=RBF; KNN, K=19, distance value=Euclidean, and weight=uniform; RF, estimators=800, max depth=30, min samples leaf = 4, min samples split=10, min features=auto, and bootstrap=true; LSTM, 1 layer with 100 neurons, batch size=64, optimizer=Adam, epochs=200. I examined the performance of each classifier with a combination of two to ten CNN models.

Table 9 Performances of the meta-learner methods trained the model using the output probabilities from the lightweight CNNs on (A) PlantDoc, (B) Crop-PlantDoc, and (C) iCassava2019 datasets.

(A)

CNN Architectures	Evaluation Metrics	Meta-learner Methods				
		LR	**SVM**	**KNN**	**RF**	**LSTM**
EfficientNetB1	Accuracy (%)	71.21	71.40	71.21	70.04	68.87
	Testing time (sec.)	0.66	0.66	0.13	0.20	0.20
	No. of models	10	10	2	3	3
MobileNetV2	Accuracy (%)	68.87	67.90	67.90	68.68	63.81
	Testing time (sec.)	0.33	0.37	0.33	0.33	0.16
	No. of models	8	9	8	8	4
NASNetMobile	Accuracy (%)	68.09	69.07	68.29	68.09	62.84
	Testing time (sec.)	0.97	0.81	1.45	0.65	1.13
	No. of models	6	5	9	4	7
InceptionResNetV2	Accuracy (%)	70.82	71.01	71.21	**72.18**	69.65
	Testing time (sec.)	1.71	1.54	1.37	1.20	1.20
	No. of models	10	9	8	7	7

(B)

CNN Architectures	Evaluation Metrics	Meta-learner Methods				
		LR	**SVM**	**KNN**	**RF**	**LSTM**
EfficientNetB1	Accuracy (%)	**90.71**	90.60	90.21	**90.71**	90.43
	Testing time (sec.)	0.17	0.22	0.20	0.17	0.15
	No. of CNN models	7	9	8	7	6
MobileNetV2	Accuracy (%)	85.37	84.97	85.93	85.37	81.77
	Testing time (sec.)	0.17	0.17	0.17	0.17	0.09
	No. of CNN models	9	9	9	9	5
NASNetMobile	Accuracy (%)	89.03	89.25	88.75	88.75	85.65
	Testing time (sec.)	0.58	0.58	0.46	0.52	0.23
	No. of CNN models	10	10	8	9	4
InceptionResNetV2	Accuracy (%)	83.12	82.72	82.95	84.36	81.54
	Testing time (sec.)	0.27	0.27	0.66	0.66	0.27
	No. of CNN models	4	4	10	10	4

(C)

CNN Architectures	Evaluation Metrics	Meta-learner Methods				
		LR	SVM	KNN	RF	LSTM
EfficientNetB1	Accuracy (%)	91.61	**91.87**	91.52	91.70	91.52
	Testing time (sec.)	0.71	0.64	0.71	0.64	0.71
	No. of CNN models	10	9	10	9	10
MobileNetV2	Accuracy (%)	89.22	88.78	88.96	88.96	88.52
	Testing time (sec.)	0.42	0.42	0.34	0.34	0.42
	No. of CNN models	10	10	8	8	10
NASNetMobile	Accuracy (%)	87.81	87.46	87.99	87.63	87.81
	Testing time (sec.)	1.56	1.73	1.39	1.56	1.56
	No. of CNN models	9	10	8	9	9
InceptionResNetV2	Accuracy (%)	90.19	90.55	90.55	90.11	90.64
	Testing time (sec.)	1.77	1.77	1.96	1.37	1.96
	No. of CNN models	9	9	10	7	10

Table *9* shows the experimental results of the stacking ensemble learning method. Notably, EfficientNetB1 could be combined with all machine learning techniques and achieved high accuracy. The experiments show that the EfficientNetB1 outperformed other CNN architectures on two plant leaf disease datasets; Crop-PlantDoc and iCassava2019. Consequently, the InceptionResNetV2, when combined with the random forest method, achieved the highest accuracy on the PlantDoc dataset. However, the MobileNetV2 was the best CNN architecture in fast prediction if the computation time is considered. Further, the EfficientNetB1 provided the second fastest prediction time.

I also compared the experimental results of the ensemble learning method with the stacking ensemble learning method. I observed that the stacking ensemble learning method slightly outperformed the ensemble learning method on all plant leaf disease datasets. However, the ensemble learning method performed faster than the stacking ensemble learning method. This was due to the stacking ensemble learning method being sent the output probabilities to predict the output with the machine learning technique, while the ensemble learning method was computed with average the output probabilities.

I compared the experimental results with the previous studies. For the plantDoc and Crop-plantDoc datasets, Singh et al. [49] achieved an accuracy of 29.73% and 70.53% on the PlantDoc and Crop-PlantDoc datasets. Our stacking ensemble of CNN performed better than Singh et al. [49] with an accuracy of 72.18% and 90.71% on the PlantDoc and Crop-PlantDoc datasets. Furthermore, for the iCassava2019 dataset, our experimental result presents greater accuracy than the accuracy obtained from Enkvetchakul & Surinta [15]. The results reported in Enkvetchakul & Surinta [15] achieved 84.51% accuracy. In comparison, our proposed method achieved an accuracy of 91.87%.

3.6 Conclusions

This paper has proposed a stacking ensemble of deep convolutional neural networks (CNNs) to recognize plant leaf disease images. First, I chose four lightweight CNNs, including InceptionResNetV2, NASNetMobile, MobileNetV2, and EfficientNetB1, to compare the accuracy results. The experiments show that the EfficeientNetB1 significantly outperforms other CNN models on all plant leaf disease datasets. I also demonstrated the impact of the ensemble learning method and the stacking ensemble learning method. Second, two types of ensemble learning, including unweighted majority vote and unweighted average methods, were proposed to recognize the output probabilities of the CNN models. Ensemble learning with the unweighted average method combined with EfficientNetB1 achieved the best accuracy performance on the three datasets. Third, I proposed to use five machine learning classifiers, consisting of logistic regression (LR), support vector machine (SVM), K-nearest neighbors (KNN), random forest (RF), and long short-term memory (LSTM) network, to create a model from the output probabilities of the CNN models. I found that EfficientNetB1 still outperformed all CNN models on Crop-PlantDoc and iCassava2019 datasets. It was the only InceptionResNetV2 that achieved better performance on the PlantDoc dataset. In the best of our experiments, the proposed stacking ensemble of the CNN framework was finally combined with EfficientNetB1, which was the lightweight model and random forest for the classifier. For the meta-learner method, all machine learning methods could further improve plant leaf disease recognition performance. However, with the increasing number of models used in ensemble learning, the time spent learning and testing increases with the number of models.

In future work to improve plant leaf disease recognition performance, I will focus on experiments with the other CNN frameworks, such as snapshot ensemble CNN and 1D-CNN. I will study other data augmentation techniques in order to increase the training data.

Chapter 4

Deep learning neural networks are trained using the stochastic gradient descent optimization algorithm. The learning rate may be the most important hyperparameter. The learning rate is a hyperparameter that controls how much the model changes in response to an estimated error every time the model's weight is updated. Choosing a learning rate can be challenging, as too small a value can result in a time-consuming training process, while extreme values can result in learning inappropriate weight sets too quickly, or the training process being unstable. Therefore this research will investigate the effects of the learning rate schedules rates on model performance. In this research, I proposed the new learning rate schedule, called equal learning rate range (ELRR) and step decay equal learning rate range (SD-ELRR), which is presented and compared with two baseline learning rate schedules: time-based decay and step decay, then using three CNNs architectures: EfficientnetB1, MobileNetV2, and NASNetMobile. The CNNs were tested on two plant leaf disease datasets; Plant Pathology and Cropped-PlantDoc Datasets. The results show that the ELRR and SD-ELRR equations improved the efficiency of plant leaf disease recognition significantly better than the basic equations in the entire plant disease dataset.

4.1 Introduction

Agriculture plays an important role in the global economy. But the major problem that affects agriculture is plant disease. Both production for consumption, distribution and export. Therefore, plant disease identification is very useful for farmers to prevent. and treat plant diseases which can help reduce economic losses. But with the disease observation by the farmers themselves takes a long time. Or seeking advice from a specialist would be quite expensive. and takes time to process thus automatically identifying plant diseases. and the diagnosis of plant diseases Therefore, it is good for agricultural productivity and help to increase productivity more as well

Nowadays, various methods are used in classification and plant leaf disease recognition, the most common of which is deep learning, which is becoming one of the essential tools for rapidly identifying images [51]. To use in-depth learning methods for detecting, diagnosing, and classifying leaf diseases, Too et al.[52] proposed deep learning architecture such as VGG 16, Inception V4, ResNet with 50, 101, and 152 layers, and DenseNets with 121 layers to recognize plant leaf diseases by learning with plantVillage datasets consisting of 38 classes of plant leaves from 14 plants, composed of both diseased and healthy plant leaves. Militante et al.[10] led a computer vision work developed by Deep Learning to detect and diagnose leaf diseases of maize, grapes, potatoes, sugarcane, and tomatoes in another study Zhong and Zhao.[1] led the Deep Model. Learning to recognize apple leaf disease By updating the DenseNet-121 module, Luna et al.[53] Using a transfer learning model, a Convolutional Neural Network (CNN) to identify tomato leaf disease. In another study, Srdjan et al.[19] used CNN to recognize 13 plant diseases, distinguishing plant leaves from the environment.

Neural network learning is essential to optimizing the model. The important thing concerning neural network learning is hyperparameters; for example, the learning rate, momentum, and activation function. The most critical variable is the learning rate [54]. If the learning rate is low, learning will proceed slowly because it adjusts the weights in the network very little. Therefore, the neural network's learning has reduced the learning rate during the learning process. which can be done by using learning rate schedules or adaptive learning rate [55]. Another problem with model learning is the number of epochs used to learn. If the number of epochs is too defined, it will take a long time to learn, and there may be overfitting problems, but if a small number of epochs occur, it may not get enough effective models (underfitting) [56]. The method used to solve the learning problem of the model is by stopping learning if the model's performance does not increase. This method is called early stopping, where early stopping checks the model's performance during learning. If the model's performance does not improve, it will stop learning immediately. This prevents overfitting problems and adopting early stopping helps stop learning before the set anniversary, thus reducing the learning time considerably [57].

In addition, deep learning has also been used to extract deep features to classify images using convolutional and pooling layers to extract features and then combine the features obtained by CNN 2 CNN (fusion) to learn the attributes. learning in this way can help improve performance [58]. In addition, there is a method that combines several models to improve the prediction performance called ensemble learning It combines deep learning models with ensemble learning to predict outcomes better.

Contribution. This research presents the optimization of learning rate schedules in deep learning to increase efficiency in plant leaf disease recognition, and uses early stopping to determine the optimal time to use in plant leaf disease recognition. Finally, I enhanced the recognition efficiency by fusion and ensemble learning to evaluate two plant leaf disease datasets, Cropped-PlantDoc and Plant Pathology. As a first step, I used three CNN architectures, EfficientnetB1, MobileNetV2, and NASNetMobile, using a method to optimize recognition by adjusting learning rate schedules during learning with two basic equations; time-based decay and step decay and two self-improved equations are called equal learning rate range (ELRR) and step decay-equal learning rate range (SD-ELRR).

Step 2. Perform the same process as the first step but use the early stopping method to help stop learning to compare time together with the performance of the model, and then the final step selects the best CNN from two basic equations and two self-made equations for Deep Fusion by deep feature extraction from both CNNs. In addition, the CNN mentioned above has been used for ensemble learning in 3 methods: unweighted majority vote, unweighted average, and weighted average. Finally, I compared the performance achieved by the fundamental equations with the equations created and the fusion efficiency with ensemble learning.

4.2 Related Work

In this study, research presented the utilization of deep learning for plant disease recognition, which developed deep learning techniques by learning rate schedule, early stopping, fusion, ensemble as follows:

4.2.1 Plant Leaf Disease Recognition/Classification

The development of deep learning has led to a growing research interest in image classification technology, which allows images to be categorized and the search for plant diseases. For example, Srdjan et al. [19] have presented an approach to using deep learning with CaffeNet to recognize 15 plant diseases, The data used was images downloaded from the Internet, cropping only the leafy parts and removing images with resolutions less than 500 pixels. Finally, a total of 33,469 images were obtained, divided into 30,880 images for training and 2,589 images for testing, with the performance of the models developed at 96.3%. In addition, Too et al.[52] presented a deep convolutional neural network for plant disease image classification, using deep learning architectures including VGG 16, Inception V4, ResNet with 50, 101, and 152 layers, and DenseNets with 121 layers. The experiment used the plantVillage dataset that contains 38 classes of leaf images from 14 different plant species, including diseased plants and healthy plant leaves. DenseNets experiments with the smallest number of parameters but more accurate than the other models at 99.75%. Atila et al. [59] have presented EfficientNet's deep learning architecture in the identification of plant leaf diseases and compared the performance of this model to other deep learning models such as AlexNet, VGG16, ResNet50, and Inception V3, testing with plantVillage datasets in deep learning model learned with transfer learning methods. According to experiments, efficientNet-B5 and EfficientNet-B4 models had the highest accuracy compared to other deep learning models at 98.42% and 99.39%, respectively.

4.2.2 Learning Rate Schedule

Current research discusses the optimization of deep neural networks, focusing on how learning rate affects the behavior of stochastic optimization. For example, Zhiyuan and Sanjeev. [60] argued that deep learning performs better at different learning rates. Research has shown results in weighting for decay and momentum. The training model uses SGD at there was momentum and an exponentially increasing learning rate schedule. Zhen et al. [55] stated that the learning rate is a crucial hyperparameter that affects the learning of the model. There are several learning rate schedules, for example; linear decay, cosine decay, exponential decay, and inverse square root decay. This study used an adaptive learning rate schedule. Later, Wangpeng et al. [61] presented the learning rate exponential decay sine wave, a technique for SGD. This method improves the learning speed of the neural network because it uses fewer epochs than the step-decay learning rate and cyclical learning rate

4.2.3 Early stopping

Early stopping is a method used to solve the learning problems of the model. Learning is stopped if model efficiency is not increased [62]. Chi et al.[63] used a feed-forward neural network (FNN) to learn about acoustic sources in an ocean waveguide. The method known as fitting-based early stopping (FEAST) was used to evaluate error when the error value was the lowest and there was no sign of further decline. The learning halt is performed by experimenting with this method to optimize the accuracy of the FNN in the test data. In addition, Zhang et al.[64] emphasized that the number of learning epochs is essential for the model's validity and designed two different early stopping criteria to help select the most suitable range to stop learning. Mahsereci et al.[65] presented a new way to stop learning. The traditional method one breaks down the training and validation sets and uses the monitoring kit to evaluate the model's performance to stop learning. As a result, model performance decreases. This new method will use fast-to-compute local statistics of the computed gradients instead of using the validation set to stop learning. This increases the training data and improves model performance.

4.2.4 Fusion

Zhao et al. [66] designed a fusion feature learning network to identify people from pedestrian images. Instead of using fusion from a single pooling operation, this design used fusion from max pooling and average pooling, which, according to the experiments, obtained models with 81.80% better identification performance. In addition, Li et al. [67] presented Deep Learning models to extract features from images and feature fusion to classify hyperspectral images (HSI) using 2D-CNN to extract features This was unlike conventional feature extraction using 1D-CNN or other 1D methods, where experiments showed that deep feature fusion was more effective compared to the spectral and local spatial feature extraction, feature fusion based, optimized for small-scale training data, and global spatial feature extraction. Chaib et al.[68] described the optical geometry group network (VGG-Net) model used for deep feature extraction to extract the characteristics of a very high-resolution (VHR) image. It selects from fully connected layers and then combines the attributes (feature fusion). Experimentally, the proposed approach performed better than the state-of-the-art approaches.

4.2.5 Ensemble

A single model to predict results may not be enough to deliver good results. Ensemble learning is combining multiple models for better results. In addition, ensemble learning reduces the variance error of predictions [69] Ju et al. [41] presented four collective learning methods; unweighted average, majority vote, optimal bayes, and super learner, combined with deep convolutional neural networks. The neural networks consisted of Neural Networks, GoogLeNet, VGG, and ResNet; the test was performed with the CIFAR10 dataset composed of 45,000 training sets, 5,000 validation sets, and 10,000 test sets. Results showed that collective learning with a super leaner method delivered the best performance with up to 95.02% accuracy.

4.3 Research Method

4.3.1 Convolutional neural networks architecture

4.3.1.1 MobileNetV2 Sandler et al. [47] presented MobileNetV2 based on MobileNetV1 and works well on mobile devices. The MobileNetV2 architecture consists of the initial full convolution layer with 32 filters, followed by 19 residual bottleneck layers. Within the residual block, there are three layers; expansion convolution layer, depthwise convolution layer, and projection convolution layer, resulting in a reduced size that is smaller than MobileNetV1 but more efficient.

4.3.1.2 NASNetMobile is another architecture that works well on mobile devices. Zoph et al. [32] have presented a CNN architecture-based search network architecture (NASNet: Neural Architecture Search Network). Using Recurrent Neural Network (RNN) and reinforcement learning, NASNet searches for a building block consisting of normal cells and appropriate reduction cells. It first searches for the best cells on a small dataset like CIFAR-10 and then transfers it to a larger dataset like ImageNet. NASNetMobile uses the NASNet architecture to scale down the model by reducing the number of a normal cells, resulting in a smaller model suitable for use with mobile devices or embedded devices.

4.3.1.3 EfficientnetB1 was presented by Tan & Le [48] and is one of the models from the EfficientNet architecture, a convolutional neural network that uses a compound coefficient method of compound scaling all three dimensions of depth, width, and resolution to enlarge. It makes available models of various sizes, including EfficientNetB0 – EfficientNetB7 sorted from the smallest to the largest size. EfficientNets can transfer learning well. It also had validity values on CIFAR-100 (91.7%) and the Flowers dataset (98.8%).

4.3.2 Baseline learning rate schedule

Learning rate is one hyperparameter used in network learning that optimizes the model. Learning rate schedules can reduce the learning rate during network learning. Some of the popular learning rate schedules are:

4.3.2.1 Time-based decay learning rate

The time-based decay learning rate schedule is the simplest form of the learning rate, and it takes the form of equation [70] as follows (1):

$$lr_n = \frac{lr_0}{1+dn} \tag{1}$$

where n is the iteration step (epoch), lr_0 is the initial learning rate, and d is the decay rate.

Based on the equation, the learning rate is reduced gradually with every cycle. The learning rate can be initialized in the variable lr_0, where n is the number of epochs starting from 1, and the rate of decline for each cycle can be set at variable d.

In this experiment, the lr_0variables were substituted as follows [0.1, 0.01, 0.001, 0.0001] and the variable d [0.1, 0.01, 0.001, 0.0001] to find the values that yield the most efficient model, using MobileNetV2 to training and testing with the Plant Pathology Dataset. The experiments revealed the importance of the variables that made the model the most efficient: lr_0= 0.01 and d = 0.0001. In each cycle, the learning rate was adjusted in increments of 0.0001, as shown in Figure 16 (a). to be used for further testing on other CNNs.

4.3.2.2 Step decay learning rate

Step decay learning rate is a popular and widely used learning rate schedule [71]. The step decay learning rate uses a fixed learning rate in a specified number of cycles and then gradually reduces the learning rate [60]. The form of equation [70] is as follows (2):

$$lr_n = lr_0 d^{floor\left(\frac{n-1}{r}\right)} \tag{2}$$

where n is the iteration step (epoch), lr_0 is the initial learning rate, d is the decay rate (which is set to a real number between 0 and 1), r is the drop rate, and $floor$ is a function that is used to return the largest integer value that is less than or equal to a number.

In this experiment, I defined the variables d = 0.5 and r = 10. The result was that the learning rate was halved with every ten epochs of training the model, shown in Figure 16 (b).

4.3.3 New learning rate schedule

4.3.3.1 Equal learning rate range (ELRR)

ELRR starts with the highest learning rate and then decreases the learning rate, decreasing steadily by a predetermined number of epochs to the lowest learning rate. The form of the equation (4) is:

$$lr_n = lr_{max} - (n-1)\left(\frac{lr_{max} - lr_{min}}{n_{max} - 1}\right) \quad (4)$$

where n is the iteration step (epoch), n_{max} is the maximum epoch, lr_{max} is the maximum learning rate, and lr_{min} is the minimum learning rate.

The equation calculates the learning rate decline interval and rate constant from the maximum to the lowest value. In this experiment, we tried to find the best range for this equation. and the next equation by defining the range of maximum values and the minimum values are (0.1,0.01), (0.1,0.001), (0.1,0.0001), (0.01,0.001), (0.01,0.0001), and (0.001,0.0001). The best performance range was (0.01,0.0001), as shown in Figure 16 (a). This range was chosen in the experiment of this equation and the next equation.

4.3.3.2 Step decay equal learning rate range (SD-ELRR)

The SD-ELRR equation applies the step decay learning rate equation by incorporating a method of determining the learning rate reduction range from the ELRR equation and adding it to the step decay equation. The SD-ELRR equation provides a constant learning rate in a given number of epochs before being downgraded, with the learning rate decreasing steadily at a specified interval, with the following patterns (5):

$$lr_n = lr_{max} - floor\left(\frac{n-1}{r}\right)\left(\frac{lr_{max} - lr_{min}}{ceil\left(\frac{n_{max}}{r}\right) - 1}\right) \quad (5)$$

where n is the iteration step (epoch), n_{max} is the maximum epoch, lr_{max} is the maximum learning rate, lr_{min} is the minimum learning rate, r is the drop rate , $floor$ is a function that is used to return the largest integer value that is less than or equal to a number, and $ceil$ is a function that is used to return the smallest integer value that is bigger than or equal to a number.

In this experiment, I defined variable $r = 10$, meaning that the learning rate dropped steadily every ten cycles, as shown in Figure 16 (b).

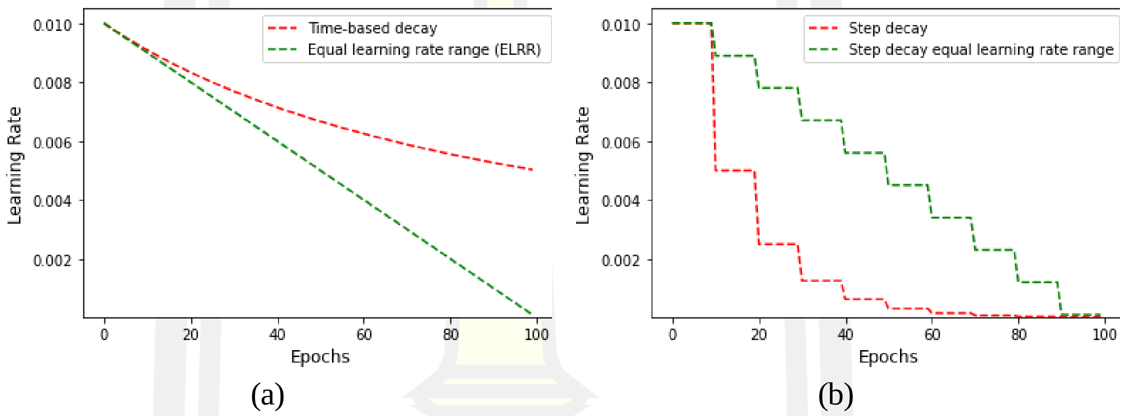


Figure 16 Learning rate schedule: (a) time-based decay and equal learning rate range (ELRR) and (b) step decay and step decay equal learning rate range (SD-ELRR)

4.3.4 Early stopping

Early stopping is an effective meta-algorithm that works with the training process, and which monitors the performance of models in every epoch of training and uses the performance obtained by the validation set as a condition to stop training [63]. Learning stops after performance or loss does not increase within a specified number of cycles (patience), as shown in Figure 17. This significantly reduced the learning curve [72].

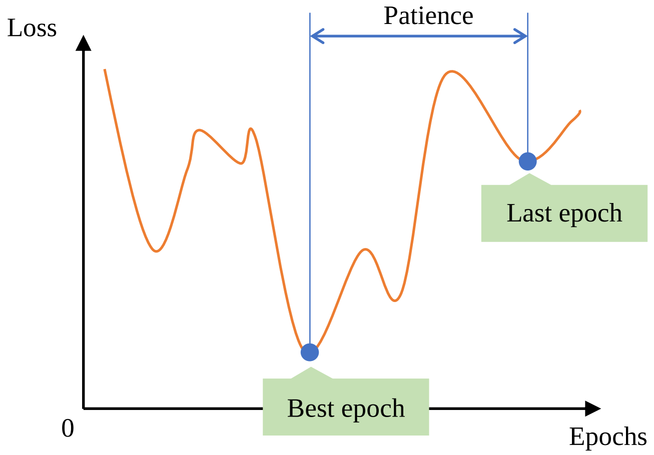


Figure 17 Early stopping based on the metric loss

4.3.5 Fusion

Deep fusion experiments were performed by taking the best model of the original equation and the best model of the new equation, two of each model, to feature fusion by eliminating the fully connected layer of the model, then feature fusion is imported to the softmax layer, as shown in Figure 18. This experiment is a feature fusion obtained from EfficientnetB1 and MobileNetV2 with the final layer of both models are global average pooling layer.

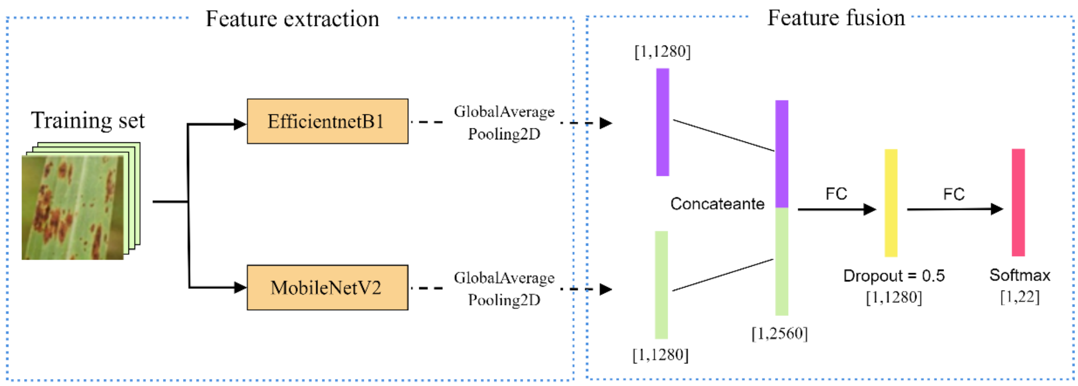


Figure 18 proposed deep feature fusion from the combination of features provided by EfficientnetB1 and MobileNetV2 with the Plant Pathology Dataset.

4.3.6 Ensemble

4.3.6.1 Vote ensemble [45] is based on the predictions of many models. Models are created in the same or different ways with the same data set, selecting the highest probability as a vote, then combining the vote, and the final prediction chooses to predict the class with a total vote. The vote ensemble method is calculated by equation (6):

$$\hat{y} = \frac{1}{n}\sum_{i=1}^{n} arg\,max\, y_i \tag{6}$$

where $arg\,max$ is the highest probability value of weight vector y_i, and n is the number of ensemble CNN models.

4.3.6.2 Average Ensemble [73] takes all the likelihood values derived from CNN models' predictions and calculates each class by selecting the most average class answer. The average ensemble method provides better accuracy because all classes of each model are considered. The average ensemble method is given by equation (7):

$$\hat{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{7}$$

where y_i is weight vector, and n is the number of ensemble CNN models.

4.3.6.3 Weighted Averaging Ensemble differs from the first two ways. Each model has equal rights, but is weighted. Averaging Ensemble gives priority weight to one or more models. The predicted result of each class is multiplied by the weighted value and then calculated for the mean and selects the class with the highest average [40]

4.4 Dataset and preprocessing

The research used plant leaf images from the Cropped-PlantDoc and Plant Pathology datasets.

4.4.1 Cropped-PlantDoc dataset

PlantDoc [49] is a dataset that collects plant leaf images from the Internet. The landing of approximately 20,900 images of plant leaves is obtained, then sorts them out, selects images that are clear images of the disease and eliminates duplicate images which are then grouped by category and examined by experts on each plant disease. Finally, there was a dataset of 2,567 plant leaf images from 13 plant species, including 27 types of diseased and healthy plant leaves. Cropped-PlantDoc [49] is a series of images that take plant leaf images from the PlantDoc dataset and cut only the part of the leaves. This resulted in new data with 8,883 images of plant leaves, shown as shown in Figure 19 and an example image shown in Figure 20.

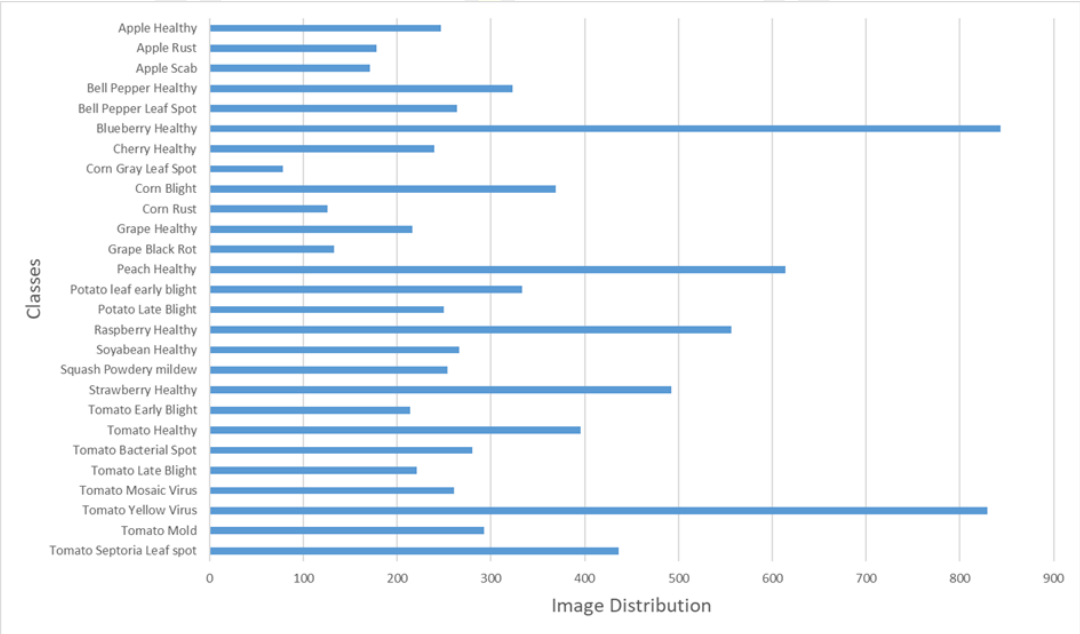


Figure 19 Statistics of Cropped-PlantDoc Dataset

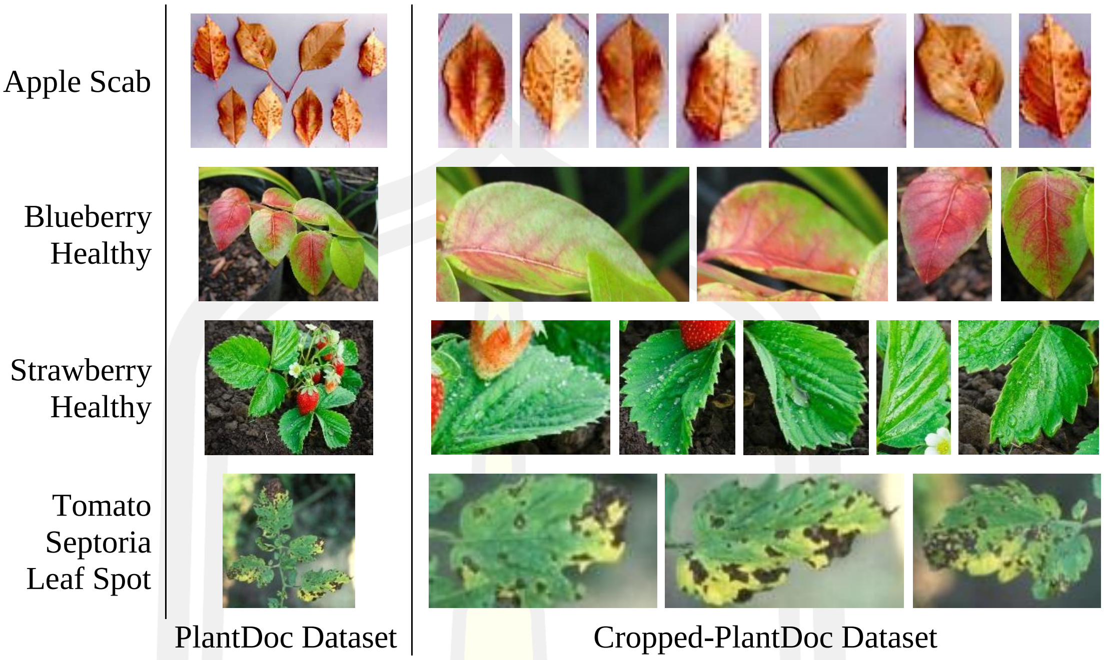


Figure 20 example images from the PlantDoc and Cropped-PlantDoc dataset. [49]

4.4.2 Plant Pathology dataset

All leaf images in the Plant Pathology dataset were collected from Shri Mata Vaishno Devi University, Katra, and were photographed in a closed environment. Then the images of the entire dataset were divided into 22 classes; 4503 images (diseased and healthy leaves) are shown in Figure 21, with an example picture shown in Figure 22 [74].

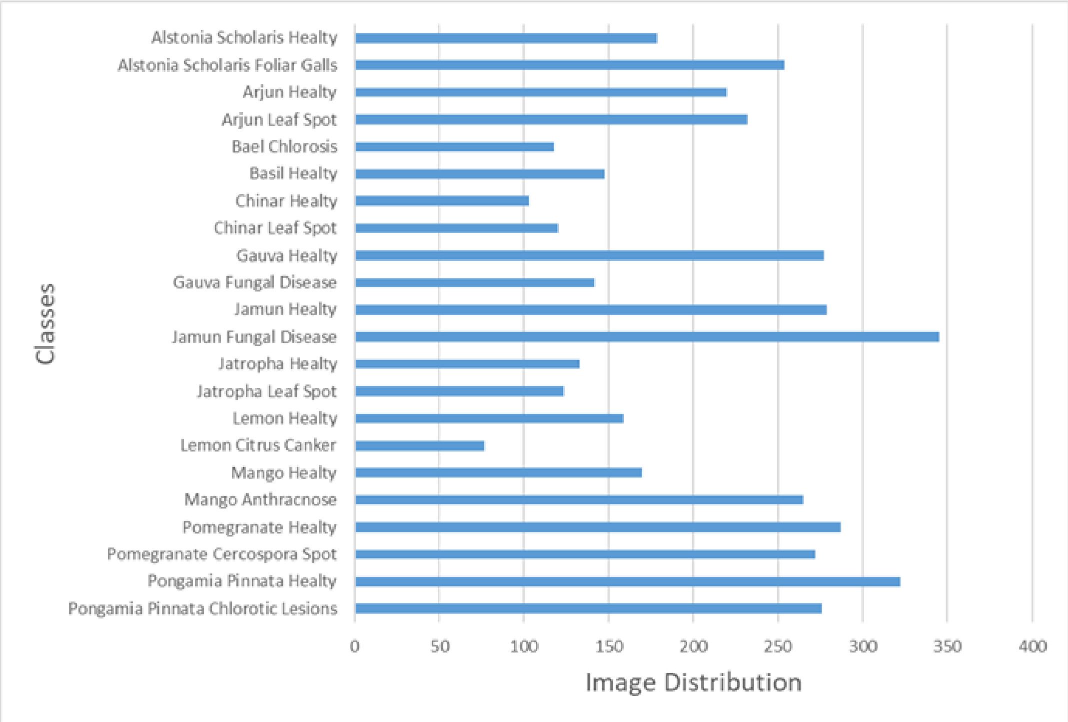


Figure 21 Statistics of Plant Pathology Dataset

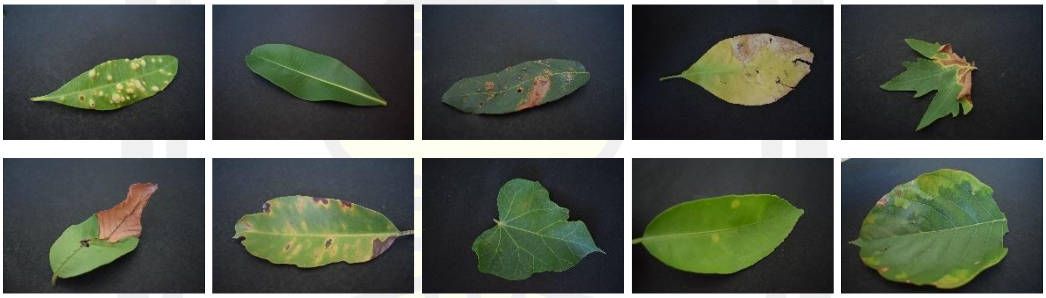


Figure 22 example images from the Plant Pathology Dataset

4.5 Experimental Results and Discussion

This study looked at three Architecture CNNs consisting of EfficientnetB1, MobileNetV2, and NASNetMobile which were tested on two datasets, the Cropped-PlantDoc and the Plant Pathology Dataset. The experimental data were randomly divided into 20% test sets for performance testing. Then, the remaining 80% of the images were used as a training set using the K-fold cross-validation method, setting K=5, and resized image to a size of 224x224 pixels before being learned by EfficientnetB1, MobileNetV2, and NASNetMobile, and testing the performance of CNN by requiring fine-tuning data to be trained. The parameters used to train were Epoch = 100, Batch Size = 16, and the

Optimizer = Stochastic Gradient Descent (SGD) algorithm. The TensorFlow Platform was used as an experiment running (Run) on a Linux Operating System using an Intel(R) Core-i5 computer, 2320 CPU @ 3.00GHz, 16GB RAM, GeForce GTX 1060Ti GPU.

4.5.1 Learning Rate Schedule

This experiment tested the effectiveness of CNN in three architectures; EfficientnetB1, MobileNetV2, and NASNetMobile, using four different learning rate schedule methods; time-based decay, Step decay, ELRR, and SD-ELRR. The results are shown in Table 10.

Table 10 Performances of the learning rate schedule methods

Datasets	CNN Architectures	Training time, Testing Time/image	Time-based decay		Step decay		ELRR		SD-ELRR	
			5-CV	Test (%)	5-CV	Test (%)	5-CV	Test (%)	5-CV	Test (%)
Plant Pathology	MobileNetV2	1h 5min, 0.13s	92.61 ± 1.51	97.34	93.06 ± 2.53	98.11	97.89 ± 0.60	98.34	97.58 ± 0.29	**98.78**
	NASNetMobile	2h 35min, 0.18s	95.25 ± 2.26	97.78	87.86 ± 6.12	98.34	97.97 ± 0.32	97.78	97.78 ± 0.38	97.45
	EfficientnetB1	2h 10min, 0.151s	98.14 ± 0.43	98.11	97.69 ± 0.35	98.37	95.25 ± 2.26	98.00	98.53 ± 0.31	97.89
Cropped-PlantDoc	MobileNetV2	1h 55min, 0.130s	71.57 ± 0.73	73.72	67.72 ± 1.28	73.49	72.65 ± 0.23	73.04	72.35 ± 1.06	72.65
	NASNetMobile	4h 30min, 0.194s	64.59 ± 2.34	79.85	66.25 ± 0.52	80.47	78.26 ± 1.20	77.38	76.83 ± 1.37	80.30
	EfficientnetB1	4h 10min, 0.163s	67.21 ± 1.54	80.02	71.20 ± 1.46	79.68	80.28 ± 1.17	80.59	79.97 ± 1.27	**81.82**

Table 10 shows that the results of the experiment with the learning rate schedule showed that MobileNetV2, which uses the SD-ELRR equation with plant pathology datasets, is the most effective compared to other CNNs and other equations. MobileNetV2 also takes less time to learn than other CNNs. As for the Cropped-PlantDoc dataset, the SD-ELRR equation still enables the highest performance values when used with EfficientnetB1. The improved accuracy of the new maybe because it can define the range of the learning rate schedule.

I have selected the best CNN of each learning rate schedule and compared it with receiver operating characteristics (ROC), as shown in Figure 23, which shows that learning efficient EfficientnetB1 models using time-based decay equations is less effective than other models.

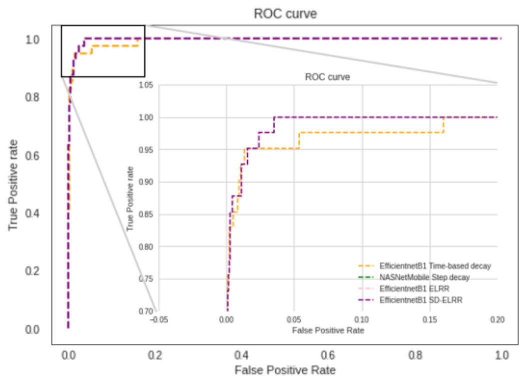


Figure 23 Illustration of the ROC curve for CNN models. The highlighted area is zoomed in at the upper left area of the curve.

4.5.2 One-Way ANOVA

Based on the results of the above experiments, the researchers wanted to determine if the five equations of each data set were significantly different statistically. The researchers used the Oneway ANOVA (analysis of variance) method, which compares the parameters of two or more samples and has homogeneity of variance, in which the model accuracy in each of the two data sets is used in the calculations.

The researcher used the validity of the test kit. The results obtained from 5-fold cross-validation in each equation were averaged ($\bar{X}$) and analyzed using one-way ANOVA. The results of data analysis comparing the difference in accuracy found that the five groups of equations were significantly different at the 0.05 level, so the differences were tested individually (Multiple Comparisons) and comparison results are shown in table 11.

Table 11 shows statistical values comparing differences in accuracy values classified by a group of equations on a pair of Plant Pathology and Cropped-PlantDoc datasets.

Dataset	Equation		Time-based decay	Step decay	ELRR	SD-ELRR
		$\bar{X}$	**95.33**	**92.87**	**98.11**	**97.96**
Plant Pathology	Time-based decay	95.33		2.46	-2.78*	-2.63*
	Step decay	92.87	-2.46		-5.24*	-5.09*
	ELRR	98.11	2.78*	5.24*		0.15
	SD-ELRR	97.96	2.63*	5.09*	-0.15	
		$\bar{X}$	**68.38**	**68.48**	**77.28**	**76.76**
Cropped-PlantDoc	Time-based decay	68.38		-0.1	-8.9*	-8.38*
	Step decay	68.48	0.1		-8.8*	-8.28*
	ELRR	77.28	8.9*	8.8*		0.52
	SD-ELRR	76.76	8.38*	8.28*	-0.52	

* The mean difference is significant at the 0.05 level.

Table 11 compares the difference in the mean accuracy of the equations by pairs on the Plant Pathology and Cropped-PlantDoc datasets. It was found that the results obtained from the pairwise comparison for the baseline equations were not statistically significantly different from each other. However, the new equations that the researcher designed and the base equations were significantly different at the 0.05 level.

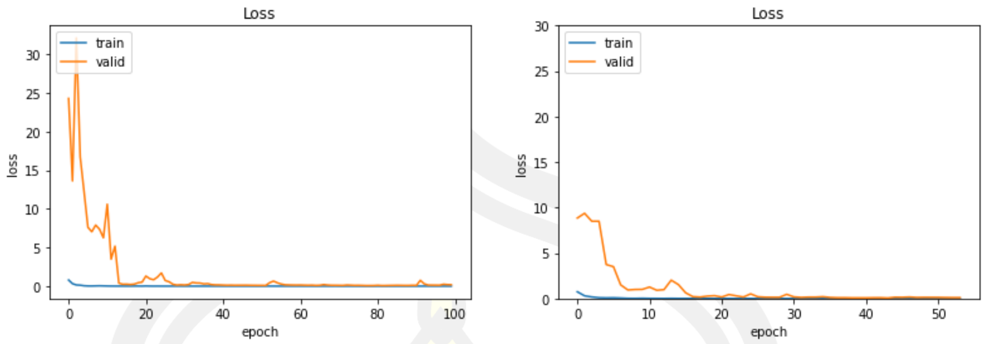
4.5.3 Learning Rate Schedule with Early Stopping

From the experiment in Section 5.2, the researcher added early stopping of learning by determining the loss in validation sets. If the loss does not decrease within the last ten learning epochs (patience = 10) learning will stop. The experimental results are shown in Table 12

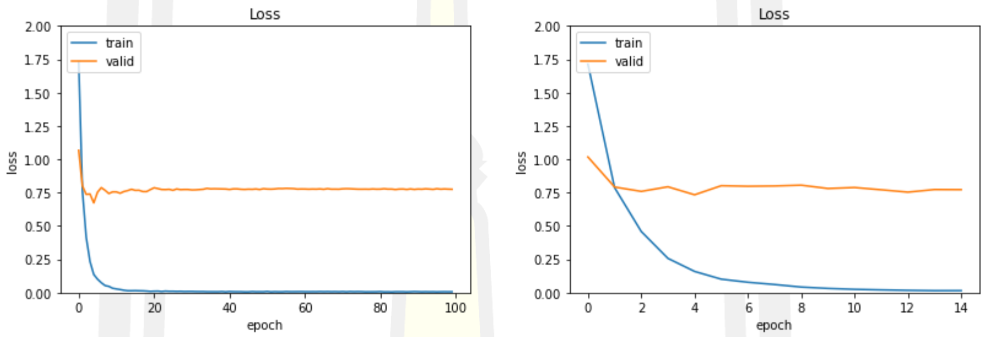
Table 12 Performance of the learning rate schedule methods with early stopping on Plant pathology dataset and Cropped-PlantDoc

Dataset	Learning rate schedule methods	Evaluation Metric Accuracy (%) / Epoch stopping / Training time		
		MobileNetV2	NASNetMobile	EfficientnetB1
Plant Pathology	Time-based decay	97.00 / 35 / 24 min	97.45 / 55 / 1h 30min	98.22 / 16 / 23min
	Step decay	97.89 / 52 / 33min	97.11 / 44 / 1h 11min	96.45 / 14 / 13min
	ELRR	97.67 / 54 / 30min	98.00 / 70 / 1h 9min	97.34 / 21 / 29min
	SD-ELRR	**98.45 / 54 / 28min**	96.89 / 41 / 55min	96.89 / 22 / 27min
Cropped-PlantDoc	Time-based decay	71.41 / 50 / 58min	73.21 / 32 / 1h 27min	77.27 / 15 / 37min
	Step decay	70.40 / 27 / 32min	79.18 / 32 / 1h 27min	**79.91 / 14 / 26min**
	ELRR	71.75 / 52 / 51min	79.01 / 58 / 1h 49min	77.60 / 13 / 32min
	SD-ELRR	72.26 / 40 / 46min	78.00 / 58 / 2h 39min	78.90 / 19 / 49min

From Table 12, the adoption of early stopping techniques significantly reduces the time it takes to learn. But it does not improve the accuracy of most models. The decrease in accuracy was not much lower than the previous one at 0.2 – 1.2 %, and in the Plant pathology dataset, MobileNetV2, which uses the SD-ELRR equation, remains the most effective compared to other CNNs, stopping learning in epoch 54 and taking only 28 min, shown in Figure 7(a). The Cropped-PlantDoc dataset sees efficientnetB1 using the step decay equation as efficiently as possible and also achieves more accuracy requiring 79.68% of 100 cycles to learn, rising to 79.91%, stopping learning in round 14 and taking only 26 min to train, which is six times less, as shown in Figure 24



(a)



(b)

Figure 24 Compared training and valid loss without early stopping and with early stopping of the learning rate schedule: (a) SD-ELRR methods use MobileNetV2 on Plant pathology dataset and (b) Step decay methods use EfficientnetB1 Cropped-PlantDoc dataset

Figure 24 shows that the loss of the training set and validation set will gradually decrease. After an increase in the number of epochs, the validation loss begins to stabilize and begins to tend to increase, which is the beginning of the overfit occurrence of learning, in which early stopping immediately stops learning when the validation loss is lowered to the lowest level and starts to stabilize to 10 epochs. In addition to learning with the early stopping technique, the problem of local minima can arise because the model can perform better after the specified number of cycles has passed.

4.5.4 Fusion

Following on from the experiment in Section 5.2, the investigator used the two best models of the original equation and the new equation of the two datasets. In the first dataset, the Plant Pathology Dataset chose EfficientnetB1 and NASNetMobile using Step decay and MobileNetV2 equations using the SD-ELRR equation, and

EfficientnetB1 using the SD-ELRR equation. The Crop-PlantDoc Dataset selected EfficientnetB1 using the Time-based decay equation with NASNetMobile using the step decay equation and EfficientnetB1 and NASNetMobile using the SD-ELRR equation. Finally, feature fusion, in which experimental results were obtained from each pair of models with both datasets was as shown in Table 13.

Table 13 Performances of the deep fusion on Plant Pathology Dataset

Dataset	Fusion	Training time	Test (%)
Plant Pathology	EfficientnetB1 (Step decay) + NASNetMobile (Step decay)	5h 27min	98.22
	EfficientnetB1 (ELRR) + MobileNetV2 (SD-ELRR)	4h 10min	**98.56**
Cropped-PlantDoc	EfficientnetB1 (Time-based decay) + NASNetMobile (step decay)	11h 41min	80.47
	EfficientnetB1 (SD-ELRR) + NASNetMobile (SD-ELRR)	11h 53min	**82.44**

4.5.5 Ensemble

Like the Fusion experiment, this exeriment used the two best models of the original equation and the new equation of the two datasets. But this experiment used the model to predict the test set results, and the probabilistic results from the two models were used for ensemble learning by three methods; Unweighted Majority Vote, Unweighted average, and Weighted average. The results of the experiments with both data sets are as shown in Table 14.

Table 14 Performances of the ensemble learning methods on Plant Pathology and Cropped-PlantDoc Dataset

Dataset	CNN Architectures	Training time	Unweighted Majority Vote	Unweighted average	Weighted average
Plant Pathology	EfficientnetB1 (Step decay) + NASNetMobile (Step decay)	4h 45min	98.33	98.45	33
	EfficientnetB1 (ELRR) + MobileNetV2 (SD-ELRR)	3h 15min	98.54	**98.67**	34
Cropped-PlantDoc	EfficientnetB1 (Time-based decay) + NASNetMobile (step decay)	4h 40min	83.46	83.34	54
	EfficientnetB1 (SD-ELRR) + NASNetMobile (SD-ELRR)	4h 41min	83.85	**84**	50

4.5.6 Comparison of experimental

Comparison of experimental results between proposed methods and from other Cropped-PlantDoc papers as shown in Table 15.

Table 15 performance evaluation of the CNNs on Cropped-PlantDoc

CNN Architectures	**Training Time**	**Test ACC**
VGG16 [49]	N/A	60.41
InceptionResNet V2 [49]	N/A	70.53
Inception V3 [75]	N/A	77.08
MobileNetV2 (Time-based decay)	1h 55min	73.72
NASNetMobile (Step decay)	4h 30min	80.47
EfficientnetB1 (SD-ELRR)	4h 10min	81.82
Deep Fusion EfficientnetB1 (SD-ELRR) + NASNetMobile (SD-ELRR)	11h 53min	82.44
Ensemble learning Unweighted average method EfficientnetB1 (SD-ELRR) + NASNetMobile (SD-ELRR)	4h 41min	**84**

4.5.7 Gradient-weighted Class Activation Mapping (Grad-CAM)

To understand the prediction of the effects of artificial intelligence and to gain confidence in the model used for predictions, the researcher then used the Grad-CAM method to visualize with a heatmap what the model actually saw, such as diseased and healthy leaves,. The model correctly considered the leaf part, as shown in Figure 25, making it possible to trust the predictive results of the model generated.

In addition, the researchers showed a Confusion Matrix of EfficientnetB1 using the SD-ELRR equation shown in Figure 26 (a) compared to the confusion matrix obtained by ensemble learning in an unweighted average ensemble learning method using the EfficientnetB1 (SD-ELRR) model in combination with NASNetMobile (SD-ELRR) in experiments with the Cropped-PlantDoc Dataset as shown in Figure 26 (b), where it is evident that unweighted average ensemble learning had the tremendous increase in accuracy.

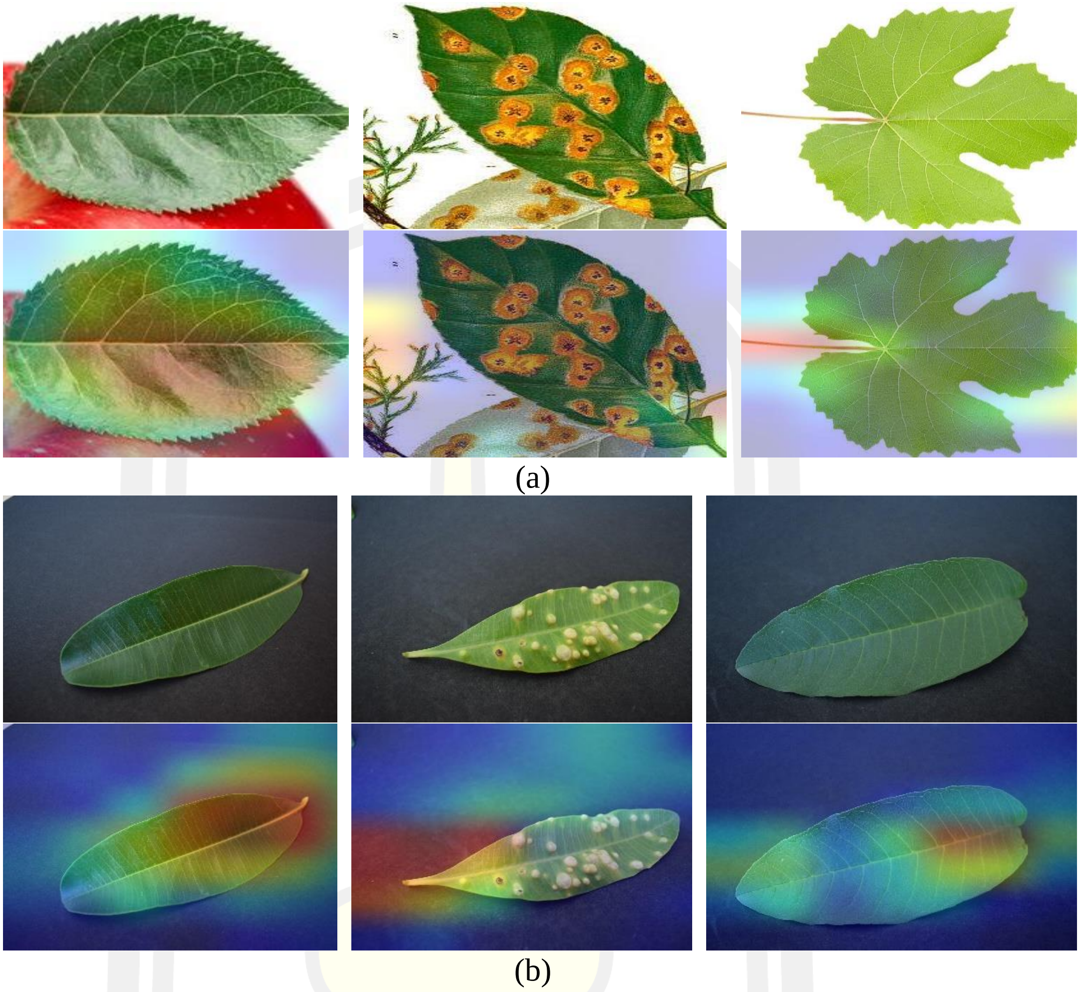
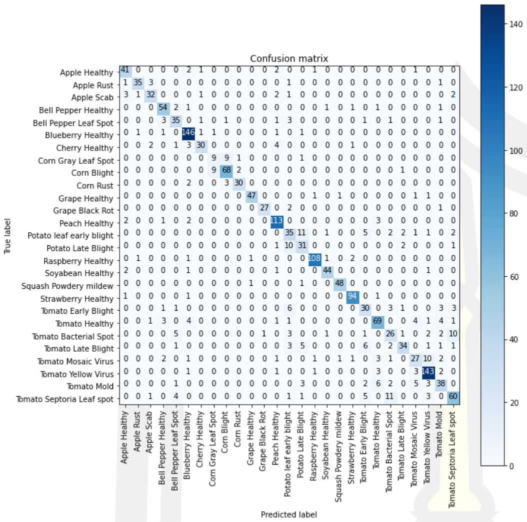
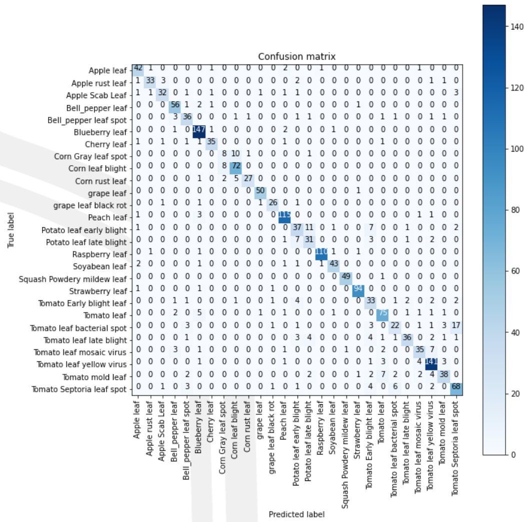


Figure 25 Grad-CAM generated with (a) Plant Pathology dataset using MobileNetV2 and (b) Cropped-PlantDoc dataset using EfficientnetB1.



(a)



(b)

Figure 26 Confusion matrix of (a) EfficientnetB1 using the SD-ELRR equation compared to (b) ensemble learning using an unweighted average method using a model of EfficientnetB1 (SD-ELRR) in combination with NASNetMobile (SD-ELRR) on Cropped-PlantDoc dataset

4.6 Conclusion

In this study, we compared how plant leaf disease is recognized, testing with two sets of plant pathology data was tested, including Plant Pathology and Cropped-PlantDoc Datasets, to demonstrate the recognition efficiency of CNN. The experiment was divided into three parts: - 1) Deep CNN testing on three architectures, EfficientnetB1, MobileNetV2, and NASNetMobile, which uses the method of adjusting learning rate schedules during learning with two basic equations; constant learning rate, time-based decay, and step decay, and two self-enhanced equations, ELRR and SD-ELRR, to compare the results of each equation. The experiments, showed that the ELRR and SD-ELRR equations improved the efficiency of plant leaf disease recognition significantly better than the basic equations in the entire plant disease dataset. 2) A repeat of the experiment in step 1, only this time using the early stopping method to stop training. It looked at the validation loss. If the validation loss did not decrease within ten epochs, it stopped training. This experiment showed that implementing early stopping even reduces learning time. 3) The performance of two best CNN models for deep fusion and ensemble learning was compared. The experiment results concluded that ensemble learning was more efficient than deep fusion.

Chapter 5
Discussion

In this thesis, I have demonstrated that the proposed algorithms are very efficient for improving plant leaf image recognition. I contributed three main types of research; data augmentation, ensemble learning, and learning rate schedule, using deep learning techniques. I will now briefly discuss the challenges of plant leaf image recognition.

For the data augmentation task, I have experimented with training techniques; online, offline, and mixed training techniques. I selected the basic image manipulations, which consist of seven techniques: rotation, brightness, width shift, height shift, zoom, cutout, and mixup., I found that the combination of rotation, shift, and zoom techniques significantly increased the performance of the CNN architectures.

For the ensemble learning task, I used four lightweight CNN architectures; InceptionResNetV2, NASNetMobile, MobileNetV2, and EfficientNetB1 to train and create robust CNN models from images of plants leaf diseases. I examined the performance of ensemble CNNs from two to ten models. For the ensemble learning method, I classified the output using an unweighted majority vote and unweighted average methods. For the stacking ensemble learning method, the stacked output probabilities of CNN models were trained using the machine learning methods; logistic regression (LR), support vector machine (SVM), K-nearest neighbors (KNN), random forest (RF), and long short-term memory (LSTM) network. I examined the performance of each classifier with a combination of two to ten CNN models.

For the optimized the model with learning rate schedules, I designed and used new equations. It can help significantly increase the efficiency of predictions, which is a part of the learning solution of the model. Using early stopping techniques to stop learning before the specified number of epochs has been reached can significantly reduce the time it takes to learn the model. The accuracy is not much different from the original. Finally, combining the features obtained by CNN 2 CNN to train the resulting attributes (deep fusion) can help increase learning efficiency. However, compared to the ensemble learning unweighted average method, it produces better results without the effort of improving or learning more like the deep fusion method

5.1 Answers to the Research Questions

Objective 1. I aimed to investigate deep learning that can be used in plant leaf disease recognition. I also enhanced the performance of the deep learning method when combining data augmentation techniques and training techniques

Research Question 1. Generally, convolutional neural networks take a large amount of data to learn to build an effective model. To avoid overfitting, most of the time, data collection problems are encountered. Therefore, there is an idea to create new data based on the existing data, called Data Augmentation. There are several methods of doing his such as Rotation, Brightness, Shift, Zoom, Cutout, and Mixup. Data Augmentation techniques can be divided into three processes: Online, Offline, and Mixed. In addition, can using convolutional neural networks in combination with learning techniques and data augmentation help increase the efficiency of plant leaf image recognition performance?

Answering RQ1. This thesis used 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.

The performance of the deep learning method was improved when combining data augmentation techniques and training techniques. In this thesis, the image manipulation techniques consisting of width and height shift, rotation, zoom, brightness, cutout , and mixup are used. I also tested three training techniques using offline, online, and mixed methods.

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

Objective 2. I aim to improve the accuracy performance of the deep learning method for plant leaf disease recognition using ensemble learning methods.

Research Question 2. Ensemble learning is a combination of different models, and independent of several models together to increase the efficiency of the model. They are divided into unweighted majority vote, unweighted average, and stacking ensemble. Therefore, can I use convolutional neural networks combined with ensemble learning to increase the efficiency of plant leaf image recognition? Is Stacking Ensemble suitable for improving plant leaf image recognition? Because of stacking the output probabilities of each CNN model and providing as output to train to create the second model using the machine learning classifier, do the number of models used in collective learning and the classification method make the Stacking Ensemble method more efficient?

Aswering RQ2. This thesis used a stacking ensemble of deep CNNs to evaluate plant leaf disease datasets; PlantDoc, Crop-PlantDoc, and iCassava2019. I used four CNN architectures, InceptionResNetV2, NASNetMobile, MobileNetV2, and EfficientNetB1, to train on the plant leaf disease images accordingly to obtain the fittest CNN model that applies in the meta-learner process. In the meta-learner process, I applied the output probabilities obtained from the fittest CNN models as inputs of a classifier. I employed five classifiers consisting of logistic regression (LR), support vector machine (SVM), K-nearest neighbors (KNN), random forest (RF), and long short-term memory (LSTM) network. Stacking ensemble was integrated with the best CNN model from the first process and the classifier from the second process to recognize and evaluate the plant leaf disease images.

Objective 3. I proposed to use the new learning rate schedule to improve the performance of the plant leaf disease classification.

Research Question 3. The thing related to neural network learning is hyperparameters, For example, the learning rate, momentum, and activation function. The most critical variable is the learning rate. Therefore, can I use the learning rate schedules when training deep learning neural networks to improve plant leaf image recognition? I created a new equation to compare it with the original equation to see which gives better performance.

Answering RQ3. I proposed the new learning rate schedules, called equal learning rate range (ELRR) and step decay equal learning rate range (SD-ELRR). These were presented and compared with two baseline learning rate schedules; time-based decay and step decay, then using three CNNs architectures: EfficientnetB1, MobileNetV2, and NASNetMobile. The CNNs were tested on two plant leaf disease datasets: Plant Pathology and Cropped-PlantDoc Datasets. The results showed that the ELRR and SD-ELRR equations improved the efficiency of plant leaf disease recognition significantly better than the basic equations in the entire plant disease dataset.

5.2 Future work

Several future direction are suggested for researchers interested in inventing better plant leaf recognition systems in plant leaf images using deep learning techniques. I divide the future work toward three tasks; data augmentation, ensemble learning, and learning rate schedules.

For data augmentation techniques, I will concentrate on improving the performance of plant leaf disease recognition. I will study and apply other data augmentation techniques such as AutoAugment and neural style transfer.

In recent years, for ensemble learning techniques, various deep learning architectures for plant leaf images have been popular and have been applied to improve plant leaf disease recognition performance. I will focus on experiments with the other CNN frameworks, such as snapshot ensemble CNN and 1D-CNN.

The learning rate is a crucial hyperparameter that affects the learning of the model. I will experiment with other hyperparameters to tune CNN, such as Dropout, Momentum or Activation function.

REFERENCES

[1] Y. Zhong and M. Zhao, “Research on deep learning in apple leaf disease recognition,” *Comput. Electron. Agric.*, vol. 168, p. 105146, Jan. 2020, doi: 10.1016/j.compag.2019.105146.

[2] A. Darwish, D. Ezzat, and A. E. Hassanien, “An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis,” *Swarm Evol. Comput.*, vol. 52, p. 100616, Feb. 2020, doi: 10.1016/j.swevo.2019.100616.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] S. V. Militante, B. D. Gerardo, and N. V. DIonisio, “Plant leaf detection and disease recognition using deep learning,” in *2019 IEEE Eurasia Conference on IOT, Communication and Engineering, ECICE 2019*, Oct. 2019, pp. 579–582, doi: 10.1109/ECICE47484.2019.8942686.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] P. Enkvetchakul and O. Surinta, “Effective data augmentation and training techniques for improving deep learning in plant leaf disease recognition,” *Appl. Sci. Eng. Prog.*, vol. 15, no. 3, pp. 1–12, Jul. 2022, doi: 10.14416/j.asep.2021.01.003.

[16] P. Enkvetchakul and O. Surinta, “Stacking ensemble of lightweight convolutional neural networks for plant leaf disease recognition,” *ICIC Express Lett.*, vol. 16, no. 5, pp. 521–528, 2022, doi: 10.24507/icicel.16.05.521.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] S. Mahmoud and S. Yaroshchak, "Bagging of convolutional neural networks for diagnostic of eye diseases," in *the 4th International Conference on Computational Linguistics and Intelligent Systems (COLINS)*, 2020, pp. 715–729.

[41] C. Ju, A. Bibaut, and M. van der Laan, "The relative performance of ensemble methods with deep convolutional neural networks for image classification," *J. Appl. Stat.*, vol. 45, no. 15, pp. 2800–2818, Apr. 2018, doi: 10.1080/02664763.2018.1441383.

[42] M. Khanramaki, E. Askari Asli-Ardeh, and E. Kozegar, "Citrus pests

classification using an ensemble of deep learning models," *Comput. Electron. Agric.*, vol. 186, p. 106192, Jul. 2021, doi: 10.1016/j.compag.2021.106192.

[43] M. Gour and S. Jain, "Stacked convolutional neural network for diagnosis of COVID-19 disease from x-ray images," Jun. 2020, Accessed: Sep. 23, 2020. [Online]. Available: http://arxiv.org/abs/2006.13817.

[44] M. Kim, M. Lee, M. An, and H. Lee, "Effective automatic defect classification process based on CNN with stacking ensemble model for TFT-LCD panel," *J. Intell. Manuf.*, vol. 31, no. 5, pp. 1165–1174, Jun. 2020, doi: 10.1007/s10845-019-01502-y.

[45] T. Chompookham and O. Surinta, "Ensemble methods with deep convolutional neural networks for plant leaf recognition," *ICIC Express Lett.*, vol. 15, no. 6, pp. 553–565, 2021, doi: 10.24507/icicel.15.06.553.

[46] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, Feb. 2017, pp. 4278–4284, Accessed: Mar. 08, 2021. [Online]. Available: https://arxiv.org/abs/1602.07261v2.

[47] 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.

[48] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *36th International Conference on Machine Learning, (ICML)*, May 2019, vol. 2019-June, pp. 10691–10700, Accessed: Mar. 08, 2021. [Online]. Available: http://arxiv.org/abs/1905.11946.

[49] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: A dataset for visual plant disease detection," in *ACM International Conference Proceeding Series*, Nov. 2020, pp. 249–253, doi: 10.1145/3371158.3371196.

[50] M. Nagano and T. Fukami, "Development of a skin texture evaluation system using a convolutional neural network," *Int. J. Innov. Comput. Inf. Control*, vol. 16, no. 5, pp. 1821–1827, 2020, doi: 10.24507/ijicic.16.05.1821.

[51] J. G. A. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," *Biosyst. Eng.*, vol. 172, pp. 84–91, Aug. 2018, doi: 10.1016/j.biosystemseng.2018.05.013.

[52] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-

tuning deep learning models for plant disease identification," *Comput. Electron. Agric.*, vol. 161, pp. 272–279, Jun. 2019, doi: 10.1016/j.compag.2018.03.032.

[53] R. G. De Luna, E. P. Dadios, and A. A. Bandala, "Automated image capturing system for deep learning-based tomato plant leaf disease detection and recognition," in *IEEE Region 10 Annual International Conference, Proceedings (TENCON)*, Feb. 2019, vol. 2018-Octob, pp. 1414–1419, doi: 10.1109/TENCON.2018.8650088.

[54] X. Wu, R. Ward, and L. Bottou, "WNGrad: Learn the learning rate in gradient descent," *arXiv*, Mar. 2018, Accessed: Apr. 03, 2021. [Online]. Available: http://arxiv.org/abs/1803.02865.

[55] Z. Xu, A. M. Dai, J. Kemp, and L. Metz, "Learning an adaptive learning rate schedule," *arXiv*, Sep. 2019, Accessed: Apr. 03, 2021. [Online]. Available: http://arxiv.org/abs/1909.09712.

[56] H. Zhang, L. Zhang, and Y. Jiang, "Overfitting and underfitting analysis for deep learning based end-to-end communication systems," Oct. 2019, doi: 10.1109/WCSP.2019.8927876.

[57] L. Rice, E. Wong, and J. Z. Kolter, "Overfitting in adversarially robust deep learning," *arXiv*, Feb. 2020, Accessed: Mar. 21, 2021. [Online]. Available: http://arxiv.org/abs/2002.11569.

[58] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016, doi: 10.1109/TGRS.2016.2584107.

[59] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using EfficientNet deep learning model," *Ecol. Inform.*, vol. 61, p. 101182, Mar. 2021, doi: 10.1016/j.ecoinf.2020.101182.

[60] Z. Li and S. Arora, "An exponential learning rate schedule for deep learning," *arXiv*, Oct. 2019, Accessed: Apr. 04, 2021. [Online]. Available: http://arxiv.org/abs/1910.07454.

[61] W. An, H. Wang, Y. Zhang, and Q. Dai, "Exponential decay sine wave learning rate for fast deep neural network training," in *IEEE Visual Communications and Image Processing (VCIP)*, Feb. 2018, vol. 2018-Janua, pp. 1–4, doi: 10.1109/VCIP.2017.8305126.

[62] G. Raskutti, M. J. Wainwright, and B. Yu, "Early stopping and non-parametric

regression: An optimal data-dependent stopping rule," *J. Mach. Learn. Res.*, vol. 15, pp. 335–366, Jun. 2013, Accessed: Jun. 04, 2021. [Online]. Available: http://arxiv.org/abs/1306.3574.

[63] J. Chi, X. Li, H. Wang, D. Gao, and P. Gerstoft, "Sound source ranging using a feed-forward neural network trained with fitting-based early stopping," *J. Acoust. Soc. Am.*, vol. 146, no. 3, pp. EL258–EL264, Sep. 2019, doi: 10.1121/1.5126115.

[64] T. Zhang, T. Zhu, K. Gao, W. Zhou, and P. S. Yu, "Balancing learning model privacy, fairness, and accuracy with early stopping criteria," *IEEE Trans. Neural Networks Learn. Syst.*, pp. 1–13, 2021, doi: 10.1109/tnnls.2021.3129592.

[65] M. Mahsereci, L. Balles, C. Lassner, and P. Hennig, "Early stopping without a validation set," Mar. 2017, Accessed: Jan. 18, 2022. [Online]. Available: https://arxiv.org/abs/1703.09580v3.

[66] C. Zhao, X. Lv, Z. Zhang, W. Zuo, J. Wu, and D. Miao, "Deep fusion feature representation learning with hard mining center-triplet loss for person re-identification," *IEEE Trans. Multimed.*, vol. 22, no. 12, pp. 3180–3195, Dec. 2020, doi: 10.1109/TMM.2020.2972125.

[67] X. Li, M. Ding, and A. Pižurica, "Deep feature fusion via two-Stream convolutional neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2615–2629, Apr. 2020, doi: 10.1109/TGRS.2019.2952758.

[68] S. Chaib, H. Liu, Y. Gu, and H. Yao, "Deep feature fusion for VHR remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4775–4784, Aug. 2017, doi: 10.1109/TGRS.2017.2700322.

[69] K. El Asnaoui, "Design ensemble deep learning model for pneumonia disease classification," *Int. J. Multimed. Inf. Retr.*, pp. 1–14, Feb. 2021, doi: 10.1007/s13735-021-00204-7.

[70] J. Park, D. Yi, and S. Ji, "A novel learning rate schedule in optimization for neural networks and it's convergence," *Symmetry (Basel).*, vol. 12, no. 4, p. 660, Apr. 2020, doi: 10.3390/SYM12040660.

[71] R. Ge, S. M. Kakade, R. Kidambi, and P. Netrapalli, "The step decay schedule: A near optimal, geometrically decaying learning rate procedure for least squares," *Adv. Neural Inf. Process. Syst.*, vol. 32, Apr. 2019, Accessed: Apr. 04, 2021. [Online]. Available: http://arxiv.org/abs/1904.12838.

[72] P. Nevavuori, N. Narra, and T. Lipping, “Crop yield prediction with deep convolutional neural networks,” *Comput. Electron. Agric.*, vol. 163, p. 104859, Aug. 2019, doi: 10.1016/J.COMPAG.2019.104859.

[73] S. Noppitak and O. Surinta, “Ensemble convolutional neural network architectures for land use classification in economic crops aerial images,” *ICIC Express Lett.*, vol. 15, no. 6, pp. 531–543, 2021, doi: 10.24507/icicel.15.06.531.

[74] S. S. Chouhan, U. P. Singh, A. Kaul, and S. Jain, “A data repository of leaf images: Practice towards plant conservation with plant pathology,” in *4th International Conference on Information Systems and Computer Networks (ISCON)*, Nov. 2019, pp. 700–707, doi: 10.1109/ISCON47742.2019.9036158.

[75] J. P. Schwarz Schuler, S. Romani, M. Abdel-Nasser, H. Rashwan, and D. Puig, “Color-Aware Two-Branch DCNN for efficient plant disease classification,” *MENDEL*, vol. 28, no. 1, pp. 55–62, Jun. 2022, doi: 10.13164/MENDEL.2022.1.055.

BIOGRAPHY

NAME	Prem Enkvetchakul
DATE OF BIRTH	15 june 1983
PLACE OF BIRTH	Buriram
ADDRESS	6-8 Tambon Naimuang, Amphur Muang, Buriram 31000
POSITION	Lecturer
PLACE OF WORK	Buriram Rajabat University
EDUCATION	2005 Bachelor of Science (B.Sc.) Computer Science, Khon Kaen University 2007 Master of Science (M.Sc.) Information Technology, Khon Kaen University 2022 Doctor of Philosophy (Ph.D) Information Thechnology, Mahasarakham University.
Research output	Enkvetchakul & Surinta, Effective data augmentation and training techniques for improving deep learning in plant leaf disease recognition, Appl. Sci. Eng. Prog., vol. 15, no. 3, pp. 1–12, Jul. 2022. Enkvetchakul and Surinta, “Stacking ensemble of lightweight convolutional neural networks for plant leaf disease recognition,” ICIC Express Lett., vol. 16, no. 5, pp. 521–528, 2022.