The Automated Plant Leaf Image Classification System using Deep Learning

Thipwimon Chompookham

A Thesis Submitted in Partial Fulfillment of Requirements for
degree of Doctor of Philosophy in Information Technology
October 2022

ระบบจำแนกรูปภาพใบพืชอัตโนมัติโดยใช้การเรียนรู้เชิงลึก

วิทยานิพนธ์

ของ

ทิพวิมล ชมภูคำ

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ตุลาคม 2565

The Automated Plant Leaf Image Classification System using Deep Learning

Thipwimon Chompookham

A Thesis Submitted in Partial Fulfillment of Requirements

for Doctor of Philosophy (Information Technology)

October 2022

The examining committee has unanimously approved this Thesis, submitted by Mrs. Thipwimon Chompookham , 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. Chatklaw Jareanpon , Ph.D.)

..... Committee
(Asst. Prof. Phatthanaphong Chompoowises , 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

....
(Asst. Prof. Sasitorn Kaewman , M.Sc.) (Assoc. Prof. Krit Chaimoon , Ph.D.)
Dean of The Faculty of Informatics Dean of Graduate School

TITLE The Automated Plant Leaf Image Classification System using Deep Learning

AUTHOR Thipwimon Chompookham

ADVISORS Assistant Professor Olarik Surinta , Ph.D.

DEGREE Doctor of Philosophy **MAJOR** Information Technology

UNIVERSITY Mahasarakham University **YEAR** 2022

ABSTRACT

Knowledge of botany is necessary in order to classify plants accurately. Sometimes, even experts can misclassify a plant. To reduce errors that are made by a human, in this thesis, we aimed to invent an automated plant leaf classification system that could classify plants from leaves using deep learning techniques. The proposed method could also classify diverse plants and identify plant diseases from leaves. In this thesis, we presented three approaches to addressing the challenges of plant leaf classification.

In the first approach, we invented a method to classify various healthy plant leaf images taken in the laboratory, called the multiple-grid method. This method could extract robust features from the local area using different feature extraction methods: histogram of oriented gradients (HOG), local binary patterns (LBP), and color histogram. Hence, the principal component analysis (PCA) technique was proposed to reduce the size of the feature and finally fed to the machine learning techniques: support vector machine (SVM) and multi-layer perceptrons (MLP). The proposed method achieved high accuracy in plant leaf image recognition.

In the second approach, the leaf images (healthy and diseased) taken in the natural environments were classified using the ensemble convolutional neural networks (CNNs) method. For the CNN models, we created various CNN models based on five architectures: MobileNetV1, MobileNetV2, Xception, DenseNet121, and NASNetMobile. The CNN models were fine-tuned with different parameters, including optimizers, batch sizes, and data augmentations. For the ensemble learning method, we classified the output probabilities of 3 CNN models (called 3-EnsCNNs) and 5 CNN models (called 5-EnsCNNs) with three different ensemble learning methods: unweighted majority vote, unweighted average, and weighted average. As a result, the ensemble CNN with the weighted average method outperformed other ensemble learning methods on three different plant leaf datasets.

In the third approach, we automatically selected the best-CNN models using the ant colony optimization (ACO) algorithm used in the ensemble CNN method. According to the ACO algorithm, we first proposed the new fitness function computed by the loss and error while training the CNN models. Second, the learning rate schedule was included in the ACO algorithm to decrease the fitness value between each

CNN model while training the ACO algorithm. We compared the performance of two learning rate schedules: the time-based and cyclical learning rate, and found that two learning rate schedules contributed to improving the ACO algorithm. Consequently, the proposed ACO algorithm outperformed the existing methods on mulberry leaf and turkey plant datasets.

We also found that many deep learning techniques could be proposed for automated plant leaf image classification. However, when we focus on the ensemble CNNs method, we should have an automated method to select the best-CNN models. Further, the proposed ACO algorithm is one of the best solutions for creating an automated plant leaf classification system. Adding the new robust CNN models to the system enables the proposed method to train the ACO algorithm and automatically choose the best-CNN models.

Keyword : Plant Leaf Recognition, Multiple Grids Approach, Local Descriptor, Dimensionality Reduction, Support Vector Machine, Multi-Layer Perceptron, Convolutional Neural Network, Ensemble Method, Ensemble Learning Method, Ensemble Convolutional Neural Network, Ant Colony Optimization, Automatic Model Selection, Metaheuristic, Learning Rate Schedules, Time-based Learning Rate Schedule, Cyclical Learning Rate

ACKNOWLEDGEMENTS

This thesis was accomplished with excellent assistance and advice from Assistant Professor Dr. Olarik Surinta, thesis advisor, who transfers research knowledge and skills, sacrifices time to give advice, and allows me to develop the necessary skills while studying the Ph.D. I greatly appreciate and would like to express my deep gratitude for this opportunity. Thank you to the National Science and Technology Development Agency for providing government scholarships in science and technology to study the Information Technology program at Mahasarakham University. In addition, I would like to thank my classmates, colleagues, and research teams who have spent time together in the research room. Also, my family has always supported and assisted me in various fields until graduation.

Thipwimon Chompookham

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Introduction

Plants are critical to human life and have many side-benefits, such as being used as food, medicine, and industry (Du, Wang, & Zhang, 2007). Medicinal plants have especially been used in folk medicine since ancient times. Herbs are often identified by experience through touch or olfaction (Hoffman, Cruickshanks, & Davis, 2009). However, some plants have similar botanical characteristics but have different benefits and toxins.

Nowadays, researchers use artificial intelligence (AI) to classify plant species and enable people without botanical knowledge to use AI to classify plant species accurately (Vo, Dang, Nguyen, & Pham, 2019). Generally, many different plant organs are useful for classification, including leaves, bark, flowers, seeds and stems. However, leaf analysis is most important as leaves has some features and characteristics, including texture, shape, color, and other geometric features (Z. Q. Zhao et al., 2015). The performance of many plant recognition systems depends on the features extracted from leaves (Suwais, Alheeti, & Dosary, 2022)**.**

We found that most of the research in plant disease classification uses leaves to determine plant diseases (Aravind et al., 2018; A. Kumar & Vani, 2019; Puangsuwan & Surinta, 2021; Turkoglu, Yanikoğlu, & Hanbay, 2021; Enkvetchakul & Surinta, 2022). Hence, early classification could decrease the severity and spread of the disease. It also could effectively help an agriculturalist prevent diseases. Most researchers classified plants based on leaves since only the leaves are sufficient for classification (Khmag, 2017; Wäldchen, Rzanny, Seeland, & Mäder, 2018). However, the most severe difficulty in recognizing leaf images is that the leaf images were taken in natural environments. So, images taken from different perspectives have a chance to increase noise, backlighting and shadow in the image, which are barriers to accurate classification.

Obviously, a person with limited knowledge about plants and plant diseases will benefit from this research area. In this thesis, we proposed an automated classification system to address the challenge problems of plant leaf images taken in real-world environments. Accurate performance was achieved.

1.2 Research Aim

This thesis aims to develop an automatic plant leaf classification system using a deep learning technique.

1.3 Research Questions

Classifying a plant species is not easy for ordinary people who would like to be an expert in the task which requires time spent in study. We can now use artificial intelligence to classify some plant species instead of botanists. However, creating the automated plant classification system is not completed research. The is still a gap to improve the performance of the system. In this thesis, we have three research questions

that could enhance the performance of the automated plant classification system, as follows.

RQ1. Plant species generally can be classified from plant organs, such as leaves, bark, flowers, seeds, and stems. However, the leaf is the most distinctive plant part that could be easier classified than other parts. Is it possible to classify plant leaf images using image processing and machine learning methods? Extracting the features from plant leaves is an important method. Furthermore, many local descriptor methods are proposed to extract the robust features (called handcraft features) from objects that appear in the image. Could machine learning techniques accurately classify the plant leaf images that extract the handcraft features using local descriptor methods and color features?

RQ2. In the previous RQ, image processing and machine learning methods were proposed to classify plants from plant leaf images. However, most plant diseases manifest symptoms on the leaves, such as downy mildew, leaf spot, leaf blotch/leaf blight, and rust. Could we classify the disease if the disease appears on the plant leaf? Could we classify the plant leaf diseases using the deep learning method, such as convolutional neural network (CNN)? Additionally, is there any method to enhance the performance of the deep learning method?

RQ3. If the ensemble learning with the weighted average method achieves better classification performance than using only one CNN model. How could we select the best-CNN models to create the ensemble CNNs method? Could we use ant colony optimization (ACO) to compute the optimal route that combines the best-CNN models and use the CNN models in the ensemble learning method?

We will present a concrete answer to all research questions (RQ1 to RQ3) in Chapter 5.

1.4 Contributions

The significant contribution of the thesis is a novel image classification system for an automatic selection of best-CNN models using the proposed ACO algorithm for classified plant leaf images. In this thesis, we performed experiments on four plant leaf datasets: Folio (Munisami, Ramsurn, Kishnah, & Pudaruth, 2015), Mulberry leaf (Chompookham & Surinta, 2021), PlantVillage (especially, Tomato and corn leaf diseases) (Hughes & Salathé, 2015), and Turkey-plant disease (Turkoglu et al., 2021).

The first part of the thesis concentrated on the traditional method of image processing and machine learning techniques. Three feature extraction methods were employed to extract the robust features from the subarea of the divided leaf images using a grid-based method. Hence, the dimension reduction method was proposed due to the enormous dimension of the feature vector before transferring features to the machine learning technique to create the model and classify, as presented in Chapter 2.

In the second part, the CNN architectures, which are one of the most successful deep learning methods, were used to classify plant leaves and plant leaf diseases.

Moreover, the output probabilities of each state-of-the-art CNN model were then combined and operated by the ensemble learning method to classify the final output, called the ensemble CNNs method, as shown in Chapter 3.

Finally, a novel automatic model selection method based on the ACO algorithm is presented in Chapter 4. The principle of the ACO algorithm is using an agent, called ant, to find the shortest route from one location to another. It will spread the pheromones along that particularly suitable path. Then, other ants could follow that specific route. However, other ants could find other routes if that path is the shortest. In this thesis, the ACO algorithm is proposed to choose the best-CNN models as the best route to obtain the highest classification accuracy. We create the ensemble CNN model using the best-CNN models that are strongly suggested by the proposed ACO algorithm. Furthermore, two main functions were added to the ACO algorithm; these were 1) the new fitness function that is computed from the loss and error values of the interested CNN models, and 2) the learning rate schedules, which is used in the training process, is proposed to learn on the new fitness function and attempt to lower the fitness value between each CNN model. The proposed method also directly affects the distribution of the pheromones, increasing the chance of discovering the new best route.

This thesis is based on the following publications.

- **Chompookham, T.**, Gonwirat, S., Lata, S., Phiphiphatphaisit, S., & Surinta, O. (2020). Plant Leaf Image Recognition Using Multiple-Grid Based Local Descriptor and Dimensionality Reduction Approach. *The 3rd International Conference on Information Science and System (ICISS)*, 72–77. https://doi.org/10.1145/3388176.3388180
- **Chompookham, T.**, & Surinta, O. (2021). Ensemble methods with deep convolutional neural networks for plant leaf recognition. *Letters, ICIC Express*, *15*(6), 553–565. https://doi.org/10.24507/icicel.15.06.553

1.5 The Automated Plant Leaf Image Recognition System

The main objective of the research presented in this thesis is to study the traditional methods (image processing and machine learning techniques), deep learning, and optimization model selection algorithm to automatically select the best models for the ensemble learning method. In this chapter, we presented the basic knowledge that assists the reader in understanding the broad idea of the automated plant leaf image recognition system, as described below.

1.5.1 Plant

Plant anatomy or plant structure describes the physical and external forms of the structure and role of plants. The body of plants generally consists of parts (Evert, 2006), as shown in Figure 1, which are further divided into two parts.

1) The root system is usually the underground system. It attaches the plant to the soil. The root system absorbs water and minerals into the stem as food storage.

2) The shoot system appears on the ground, including leaf, node, stem, flower, and fruit. Actually, we classify plants from the shoots.

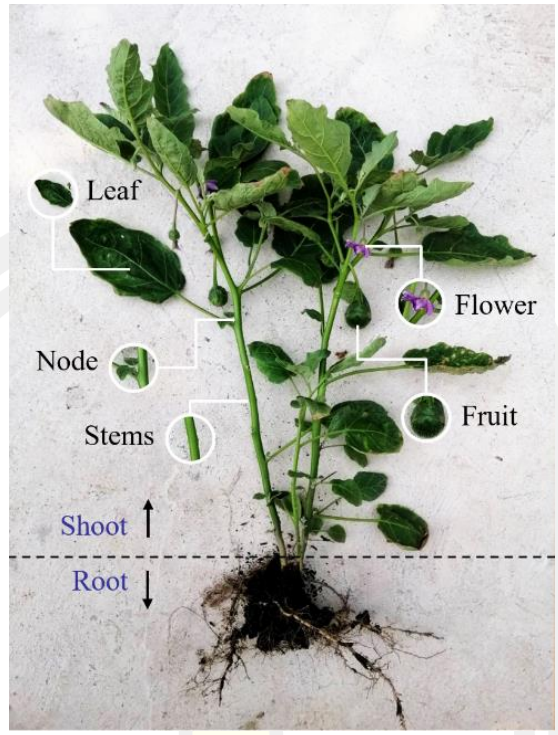


Figure 1 *Illustrates the basic plant structure.*
("Characteristics and Structures of Plants," 2020)

To classify the plant leaves, botanists mainly focus on the external features of the leaves. The leaves are thin, spread above the ground, and can photosynthesize. Botanists usually analyze the pattern of the leaves, such as size, shape, color, lobes, and veins (Mauseth, 2016). The different parts of the plant leaf are shown in Figure 2, while various plant leaves are shown in Figure 3.

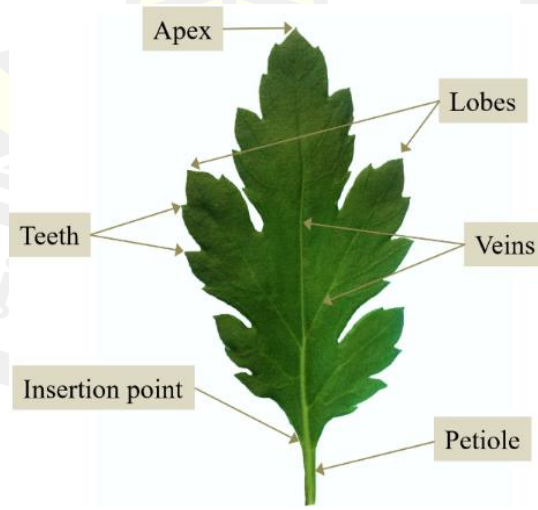


Figure 2 *Illustrates the leaf features.*
(Cope, Corney, Clark, Remagnino, & Wilkin, 2012)

Figure 2 presents the features of the plant leaf (Cope et al., 2012; Valliammal, 2013) as follows.

- Apex is the top of the leaf and has different characteristics.
- Lobes are leaf margins that have a groove concave towards the midline of the leaf.
- Teeth are the edge of the plant leaf.
- Veins are the center of the leaves connected to the petiole and transfer water and nutrients to the rest of the plant.
- Insertion point is the point between the petiole and the leaf.
- Petiole is the substantial part that that connects the leaf to the stem.

Figure 3 *Examples of different shapes of plant leaf*
(Munisami et al., 2015)

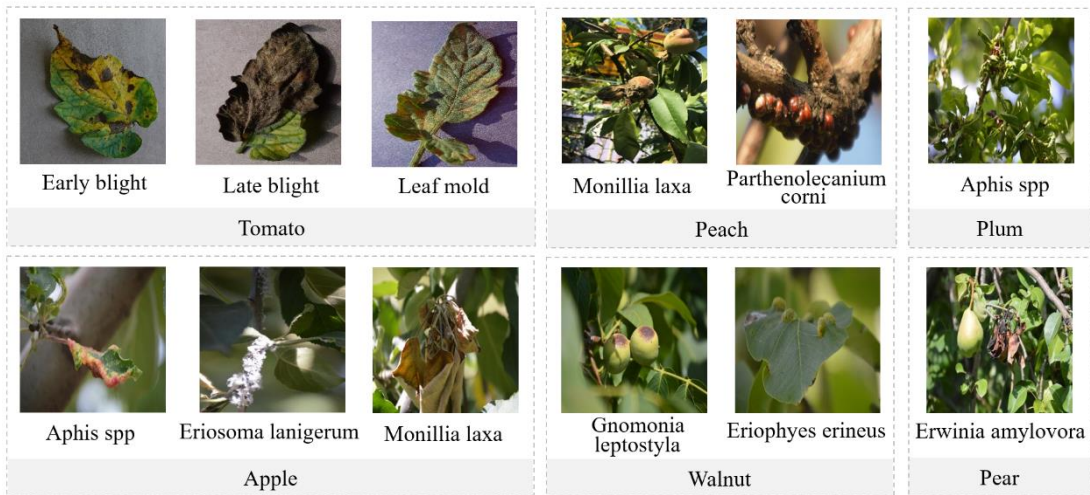


Figure 4 *Examples of plant leaf diseases.*
(Turkoglu et al., 2021)

1.5.2 Plant Diseases

Plant disease is a serious concern as it destroys plants and agricultural products, reduces product quality, and increases the cost of production. Plant disease can affect every part of the plant, especially the leaves. The causes of plant diseases are bacteria, insects, fungi, and nutrition deficiencies (Sinha & Shekhawat, 2020), as shown in Figure 4. The early detection of the disease is a good solution to prevent the spread of the disease.

In artificial intelligence, image processing and machine learning techniques are proposed to classify plant leaf diseases (Sladojevic, Arsenovic, Anderla, Culibrk, & Stefanovic, 2016; Kusumo, Heryana, Mahendra, & Pardede, 2018; Approach & Leonowicz, 2021), as shown in the following section.

1.5.3 Image Processing and Machine Learning Techniques

Traditionally, image processing and machine learning techniques are proposed to solve many problems in image classification, such as plant leaf disease classification, and aim to precisely classify images into the appropriate category (Ponnusamy, Sathiamoorthy, & Manikandan, 2017). The traditional framework of the image processing and machine learning techniques is shown in Figure 5. For plant leaf image classification, first, the leaf images could be taken from the laboratory with a white background. It is uncomplicated to apply image processing techniques to extract leaves and background. Second, all leaves are then sent to transform (such as rotation, resize, or translation) and improve the image quality (Sonka, Hlavac, & Boyle, 1993). These methods are called the image pre-processing process. Third, a feature extraction method is required to extract the robust features from the leaf images. Many well-known feature extraction methods, such as local binary patterns (LBP) (Ojala, Pietikainen, & Maenpaa, 2002), scale-invariant feature transform (SIFT) (Lowe, 2004), histogram of oriented gradients (HOG) (Dalal & Triggs, 2005), and speeded up robust feature (SURF) (Bay, Tuytelaars, & Van Gool, 2006), have been proposed to extract the features from the keypoint or patterns of the images. Several methods also extract features depending on geometric, statistical, and color features (Mutlag, Ali, Aydam, & Taher, 2020). Fourth, the machine learning model is created according to the robust features extracted from the previous process. This process is called classification. The well-known and successful machine learning techniques that could be proposed to address the image classification problems, such as support vector machines (SVM) (Vapnik, 1998), artificial neural networks (ANN) (Jain, Mao, & Mohiuddin, 1996), and k-nearest neighbor (K-NN) (Altman, 1992). Finally, the output of the classification process is the accurate class labels.

Figure 5 *Illustrated processes of the image classification using image processing and machine learning techniques.*
(Ponnusamy et al., 2017)

1.5.4 Deep Learning Technique

The deep learning technique (G. E. Hinton, Osindero, & Teh, 2006) is very effective and widely applied to many problems, such as detection, classification, and clustering (Abas, Ismail, Yassin, & Taib, 2018; Durmuş, Güneş, & Kırcı, 2017; Harangi, 2018; S. Park, Suh, & Lee, 2020). However, CNN architecture is the most well-known deep learning technique (Ganaie, Hu, Malik, Tanveer, & Suganthan, 2021). Yan LeCun proposed the first CNN architecture that contained only five layers, called LeNet5 (LeCun, Bottou, Bengio, & Haffner, 1998). LeNet5 was proposed to classify the handwritten digits. Furthermore, the CNN architecture became popular when Alex Krizhevsky proposed the novel CNN architecture, called AlexNet, which contained eight layers and won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Krizhevsky, Sutskever, & Hinton, 2012). The AlexNet architecture was run for around six days at that particular time on two GTX 580 3G GPU. After that, many CNN architectures were proposed and were successful in the ILSVRC competition, such as VGG (Simonyan & Zisserman, 2014), Inception (Szegedy et al., 2014), ResNet (He, Zhang, Ren, & Sun, 2015), MobileNet (Howard et al., 2017a; M Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018), DenseNet (Huang, Liu, Van Der Maaten, & Weinberger, 2018) and NASNet (Zoph, Vasudevan, Shlens, & Le, 2018). The CNN architectures have also been applied in the agriculture domain (Amara, Bouaziz, & Algergawy, 2017; DeChant et al., 2017; Mohanty, Hughes, & Salathé, 2016; Sladojevic et al., 2016).

The basic concept of CNN architecture is shown in Figure 6. The CNN architecture mainly contains convolutional and pooling layers that are designed to extract robust features from the images using convolutional operation (LeCun et al., 1998), which is the mathematical calculation. Hence, the features are fed to the neural networks with fully connected calculations. It is called a fully connected layer. Further, the output of the fully connected layer is the prediction class. The calculation of the convolutional and pooling layers is illustrated in Figures 7 and 8.

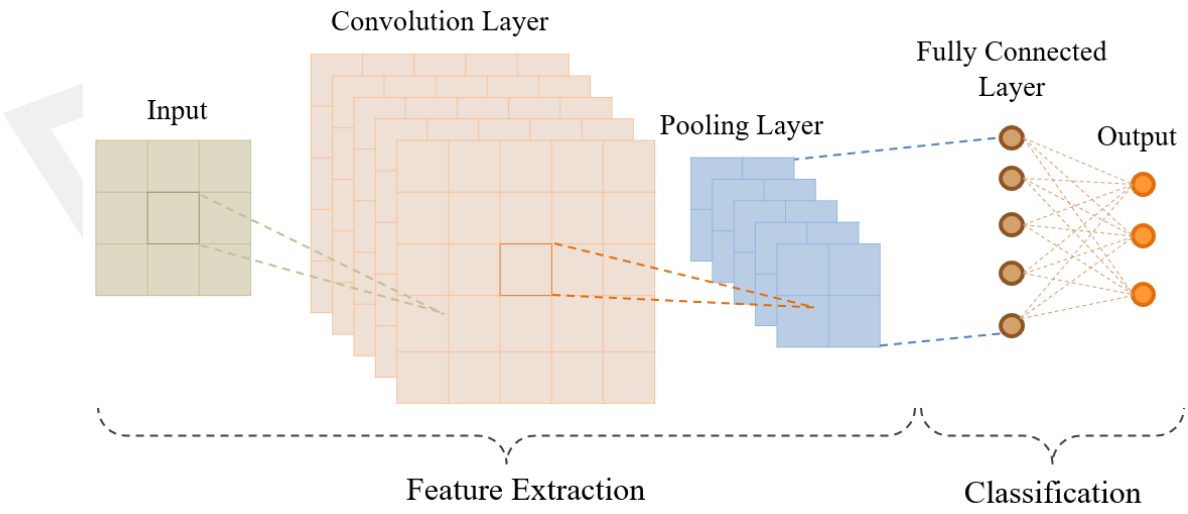


Figure 6 *Illustration of the convolutional neural network architecture*

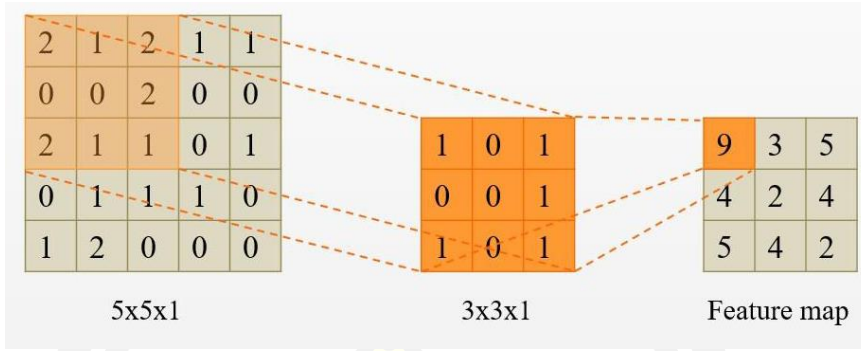


Figure 7 *Illustration of the convolutional operation*

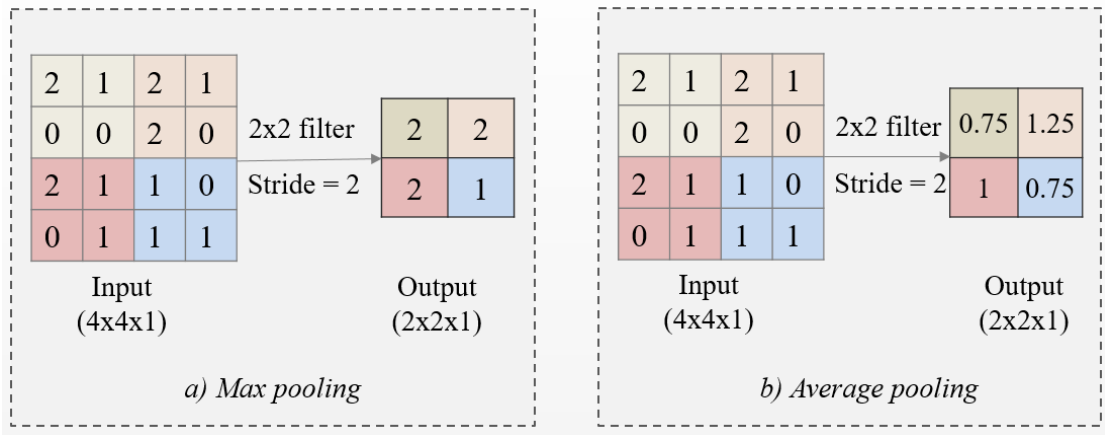


Figure 8 *Illustration of the pooling operations: (a) max pooling and (b) average pooling.*

1.5.5 Ensemble Learning

The ensemble learning method was proposed to improve the efficiency of a single classifier model. The main idea is to propose multiple classifiers to generate robust models. Then, the outputs of each robust model are combined to produce one final output (Hansen & Salamon, 1990), further reducing classification errors.

In this thesis, we combined output from various CNN models and then aggregate the CNN output to the ensemble learning method to classify the final result. We briefly introduce three ensemble learning methods that were used in our experiments, as follows.

1) Unweighted Majority Vote Method

The unweighted majority vote is the simplest calculation method in which the outputs of each classifier are counted (Dogan & Birant, 2019). So, the class with the highest majority is selected as the final classification.

2) Unweighted Average Method

In this method, the output probabilities derived from CNN models are averaged. Also, the highest probability value is selected as the final classification (Ju, Bibaut, & van der Laan, 2018).

3) Weighted Average Method

The weighted average method is an extension of the unweighted average method by assigning different weighted parameters to the output probabilities. The best classifier result a high weighted value and assign the lowest weight to the worst classifier (Harangi, 2018). Therefore, the summation of all weighted values is equal one.

1.5.6 Ant Colony Optimization

Ant Colony Optimization (ACO) (M. Dorigo, 1992) is a metaheuristics method (Blum & Roli, 2003; Glover & Kochenberger, 2003) inspired by nature in solving complicated combinatorial optimization (CO) (Papadimitriou & Steiglitz, 1982). In this thesis, the ACO is proposed as the automated model selection method to discover the robust CNN models. Actually, the researcher should manually select numbers of robust CNN models. With the manual model selection, it takes too much time to discover the best combination of CNN models to achieve the highest classification result. When the proposed ACO method is examined, it could be automated to select the optimal number of CNN models within the shortest time. Furthermore, the output probabilities of the CNN models selected by the proposed ACO method are classified using the ensemble learning method.

Chapter 2

Plant Leaf Image Recognition using Feature Extraction and Machine Learning

The process of plant species classification is a significant and challenging problem. Focus on plant leaf image classification is the main objective of many researchers because plant leaves are found almost all year round. The achieved method of plant leaf image recognition is based on extracting robust features from the plant leaf and uses the well-known machine learning technique as a classification method. As a result, recognition accuracy is often not very high. In order to improve recognition accuracy, first, we proposed a multiple-grid technique to divide the leaf image into small grids. Second, compute the feature from each grid using well-known local descriptors. Third, dimensionality reduction is proposed to transform and decrease the correlated variables of the feature vector. Finally, the feature vector with a relatively low-dimensional is transferred to the machine learning techniques, which are the support vector machine and multi-layer perceptron algorithms. We have evaluated and compared the proposed algorithm with the bag of visual words method and the deep convolutional neural network, including AlexNet and GoogLeNet architectures, on the Folio leaf image dataset. The experiments showed that the proposed algorithm has improved and obtained very high accuracy.

2.1 Introduction

Plants are living things that relate directly to humans and are used as food and medicine. Botanists have collected and studied various plant species, which can be of some benefit to humans. However, while the physical characteristics of some plants are similar, they have different benefits and toxins. As such, the ability to distinguish the types of plants requires advanced knowledge of botany. A typical plant classification problem is the diversity of plants and their botanical characteristics. Researchers find that classification of plant species is a challenging problem. Nowadays, computer vision and machine learning techniques are proposed as tools for recognizing plants.

This research aims to use image processing and machine learning for plant classification by classifying plant leaf photos taken from the laboratory. Wäldchen and Mäde (2018) said that over the past 10 years, researchers have tried to recognize plants from various parts, including leaves, plant blossoms, and fruits (Caballero & Aranda, 2010; Cerutti, Tougne, Mille, Vacavant, & Coquin, 2013; Cho, 2012). Most researchers are interested in the leaves because the plant leaves have a specific shape, surface shape, color, and leaf structure (Caglayan, Guclu, & Can, 2013; Hossain & Amin, 2010). The plant leaf images used in this research are divided into two conditions 1) Plant leaf taken in an outside environment (Wang, Huang, Du, Xu, & Heutte, 2008) and 2) Plant leaf taken in a laboratory on a white background (Munisami et al., 2015; Pawara, Okafor, Schomaker, & Wiering, 2017; Pawara, Okafor, Surinta, Schomaker, & Wiering, 2017).

Cerutti, Tougne, Coquin, & Vacavant (2013) used curvature-scale space for recognizing margin shape and Leaf classification from plant leaf characteristics by semi-supervised fuzzy C-means (FCM) training the margin shape with 12 terms. Then, it learns on the Pl@ntLeaves database, which is divided into three subsets, which are scan, pseudoscan, and photograph, using the Top-K method evaluated on the test set. The experimental results showed that the accuracy rates of the Top-K method using K=10 on the scan, pseudoscan, and photograph datasets were obtained as 95%, 92%, and 80%, respectively.

Munisami et al. (2015) collected images of plant leaves taken in the laboratory, called the Folio dataset. The Folio dataset contains 637 images of 32 plant species. In their research, feature extraction methods, including plant shape and color histogram, were proposed to extract features from the plant images, which were the used to create the model using K-nearest neighbor. Their method achieved an accuracy rate of 87.3%. Pawara et al. (2017) used deep convolutional neural networks (CNNs), including AlexNet and GoogLeNet architectures, to classify plant leaf image datasets. Moreover, their experiments used local descriptors: histogram of oriented gradients (HOG) and bags of visual words (BOW) to extract the features. Then, the support vector machine (SVM), multi-layer perceptron (MLP), and K-nearest neighbor (KNN) were employed for the classification of plant leaf images. Their experiments divided datasets into two sets: 80% for the training set and 20% for the test set. The results showed that the AlexNet architecture trained with the fine-tuned model was the most accurate, with a 97.67% accuracy. Moreover, Pawara, Okafor, Schomaker, et al. (2017) used 6 data augmentation methods that were rotation, blur, contrast, scaling, illumination, and projective transformation. The data augmentation methods can add images from the training set up to 25 times. The training set increased to 11,125 images and was trained using the AlexNet architecture. The experimental results showed that increasing the training set using the contrast data augmentation method increased the accuracy to 99.04%. Moreover, when evaluated using the GoogLeNet architecture, it was found that the illumination method achieved the highest accuracy rate at 99.42%.

Another set of plant leaf images taken in the laboratory was the Flavia dataset presented by Salman, Semwal, Bhatt, & Thakkar (2017). This includes 32 plant species and contains 1,907 leaf images. The shape feature of the plant leaves was extracted before being classified by the SVM method. The accuracy obtained from their method was 85%. At the same time, Arafat et al. (2016) developed an automatic leaf classification system using colored SIFT as a feature extraction method and SVM as a classification. Khmag (2017) extracted robust features by geometrical and shape features. Then, the features were sent to classify using the SVM classifier and obtained an accuracy above 97%. Chaki, Parekh, & Bhattacharya (2015) proposed texture-based constraints to extract and classify features using the MLP method. For the MLP, the MLP network contained an input layer with 44 nodes, a hidden layer with 30 nodes, and 31 nodes as an output layer. Their method achieved an accuracy rate of 87.1% on the test set.

Contributions: The research focuses on the importance of plant leaf recognition by experimenting with the Folio dataset, which contains 32 different plant species. This research used feature extraction methods and a dimensionality reduction approach to extract the relative component from leaf images that are divided into multiple grids.

The proposed method is simple but effective. The multiple grids divide plant leaves into sub-regions, then the method brings the sub-region to calculate the special features using various feature extraction techniques that extract the distinctive characteristics of the plant leaves. The feature extraction methods are a histogram of oriented gradients (HOG), local binary pattern (LBP), and color histogram. Finally, the features are fed to the dimensionality reduction method using principal component analysis (PCA) to reduce the feature vector size. The size reductions have a direct effect on training time and increase the recognition efficiency. This paper used the feature vector in the training process and recognition by a support vector machine (SVM) and Multi-layer perceptron (MLP). This proposed method obtained a very high recognition rate compared to the deep learning method.

Paper Outline: This paper has been organized as follows. In Section 2.2, the method for plant leaf recognition is explained. Section 2.3, the dataset and pre-processing with plant leaf images, which are used in our experiments are described. In Section 2.4, experimental results are presented. The last section discusses the significant findings from this study and suggests future work.

2.2 Proposed Plant Leaf Recognition Method

This study uses multiple grids and dimensionality reduction based on three feature extraction techniques. The process of this research is shown in Figure 9. The input images were forwarded to the multigrid-based process to divide the images into sub-region. A sub-region was calculated by three feature extraction techniques and followed by PCA to decrease the number of feature vectors. Finally, all features were concatenated and used as a feature vector $(f_1, f_2, \ldots, f_n)$. Then, the feature vector was transferred to the classification process.

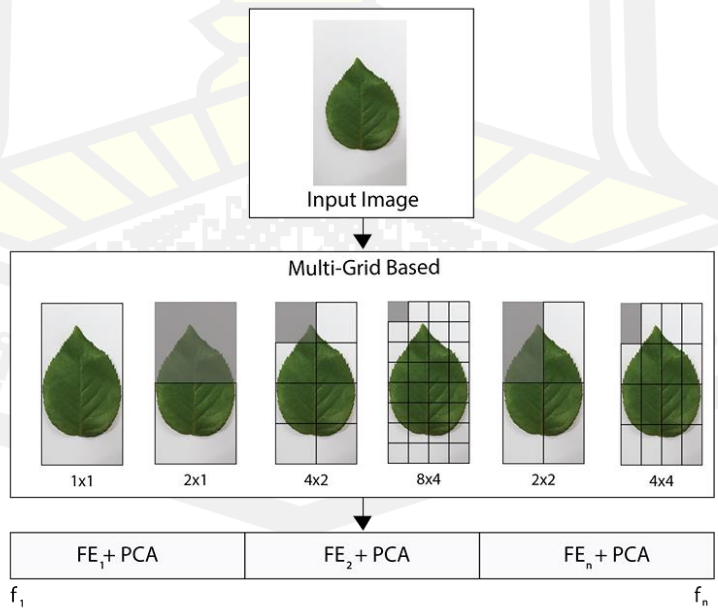


Figure 9 *Proposed plant leaf recognition method.*

2.2.1 Multiple Grid-based Technique

The working process of the multiple-grid technique is to divide the leaf images into sub-regions using a grid method to determine the sub-region. In these experiments, the grid method was determined at 6 different sizes, 1×1, 2×1, 4×2, 8×4, 2×2, and 4×4. After that, each sub-region was calculated to find the feature vector by HOG, LBP, and color histogram.

2.2.2 Feature Extraction Techniques

2.2.2.1 Histogram of Oriented Gradients (HOG)

HOG was introduced by Dalal and Triggs (2005). It is the feature extraction method that extracts the characteristics of the image by calculating the oriented gradients from gradients image by finding gradients in horizontal (G_x) and vertical directions (G_y) which are calculated from pixel intensities ($I(x,y)$) at coordinate (x,y) as the following equation:

$$G_x = I(x+1,y) - I(x-1,y) \quad (1)$$

$$G_y = I(x,y+1) - I(x,y-1) \quad (2)$$

The magnitude (M) and gradient orientation (θ) are calculated as the following equation:

$$M(x,y) = \sqrt{G_x^2 + G_y^2} \quad (3)$$

$$\theta_{x,y} = tan^{-1}\frac{G_y}{G_x} \quad (4)$$

where $M(x,y)$ is magnitude of gradients, $\theta_{x,y}$ is gradient orientation at coordinate x,y. Then, gradient orientation values will be taken to the weighted vote process and will be kept in the orientation bins (β) (Karaaba, Surinta, Schomaker, & Wiering, 2015).

Finally, gradient orientation values, which are kept in each orientation bin, will be taken to do the normalization by the L2-norm method.

2.2.2.2 Local Binary Patterns (LBP)

LBP was proposed by Ojala et al. (2002) for invariant texture classification. LBP is first designed for extracting characteristics of pixel points from neighborhood pixels which are calculated from gray values as the following equation:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \quad (5)$$

where

g_c is the gray value of the central pixel.
g_p is the gray value of its neighbor pixels.
P is the total number of involved neighbors.
R is the radius of the neighborhood.

The central pixel will be used as the threshold value (T) to compare with the values of neighborhood pixels, $s(x) = \{1, x \geq T\ 0, x < T$. The following process brings the values 1 and 0 from neighborhood pixels together as concatenates. Then, they were converted from binary format to decimal format. Consequently, the decimal values were put into the specified bins and used as the feature vector.

2.2.2.3 Color Histogram

This research used two color models: RGB and HSV color models. We used only hue (H) values because hue values show the true color. Therefore, the color values used for histogram creation consist of red (R), green (G), blue (B), and hue. Thus, the histogram of the RGB and H values consists of 256 and 360 color values.

2.2.3 Dimensionality Reduction

From the Multiple-grid based method, a lot of sub-regions will be created, which are used for calculation of unique features. This causes high dimensionality of the feature vector and results in computational complexity. Therefore, dimensionality reduction is one of the best ways to minimize the feature vector. This research uses PCA (Cootes, Taylor, Cooper, & Graham, 1995) in feature vector reduction. The feature vector from each technique has been reduced to only 80 Features. These techniques also improved the accuracy rate.

2.2.4 Classification Algorithms

This research used two algorithms, SVM (Vapnik, 1998) and MLP (Haykin, 2008), as classification models. The SVM with the RBF kernel and MLP by determining the two hidden layers were employed. The dropout method was selected for the prevention of an overfitting problem.

2.3 Plant Leaf Dataset

The plant leaves images used in the experiment were taken in the laboratory. Thus, most images have a white background. The background makes the leaves prominent and clearly separates them from the background.

2.3.1 Folio Dataset

The leaf images used in the experiment were the Folio dataset, presented in 2015 (Munisami et al., 2015). The images represent 32 plant species (see Figure 10) and contain 637 images in the dataset. All images were taken in the laboratory with a white background and were stored in JPEG format. The size of the images is 2322×4128 and 2448×3264 pixel resolution. The plants were cultivated on the University of Mauritius

farm. In the Folio dataset, twenty images of each plant species were collected, except for mulberry with 19 images and eggplant with 18 images.

Some Image differentiation of each species is shown in Figure 11. Some plant leaves still have similar shapes, e.g., star apple and pomme jacquot (See Figure 12). The factors mentioned above have directly affected the accuracy of recognition.

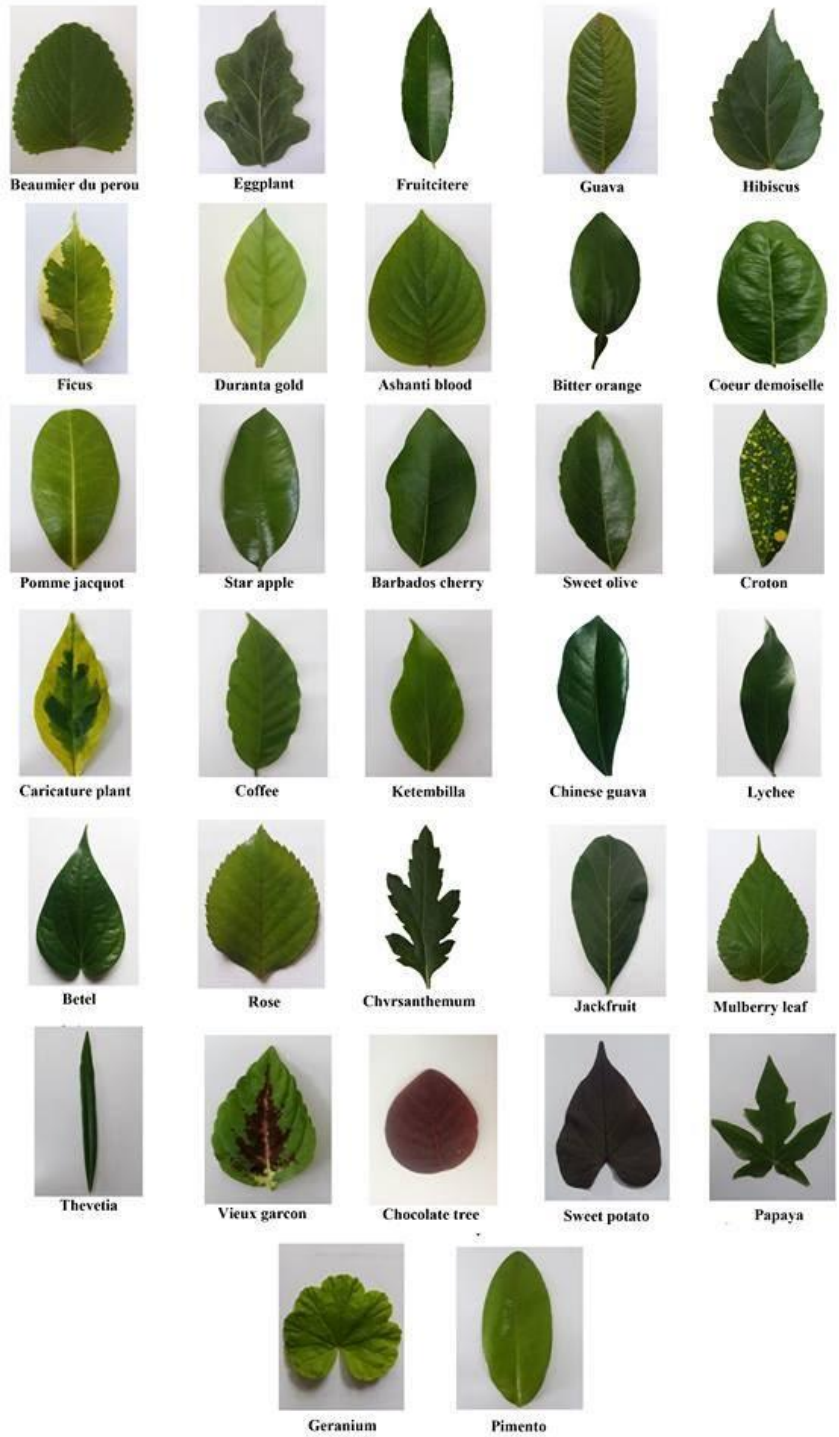
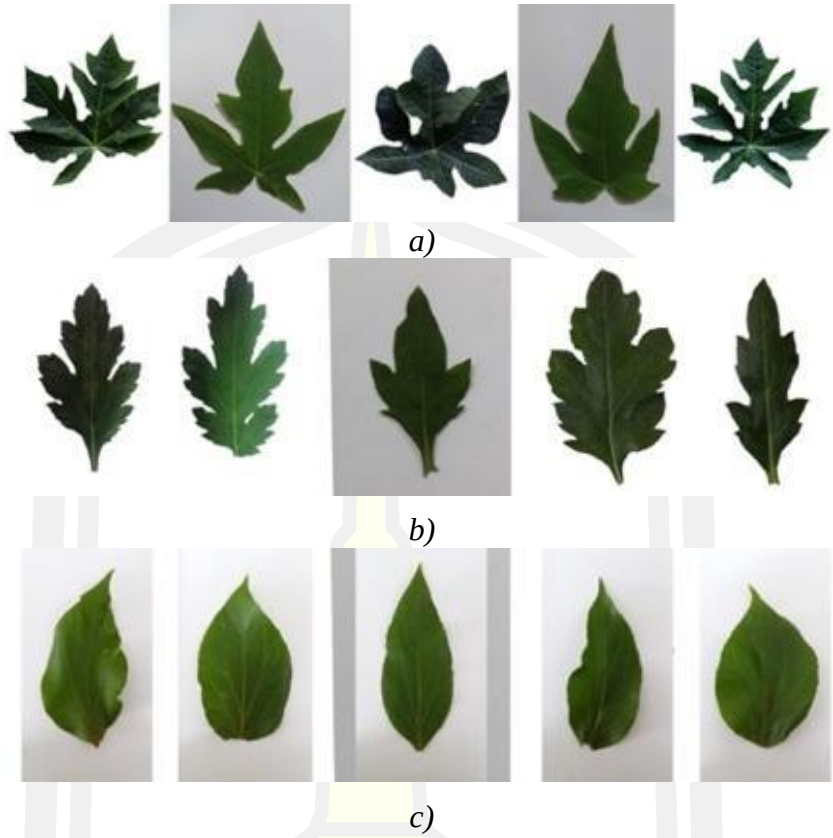


Figure 10 *Examples of 32 plant leaves of the Folio dataset.*



a)

b)

c)

Figure 11 *Examples of plant leaf images of the Folio dataset: a) papaya, b) Chrysanthemum, and c) Ketembilla.*

a) *b)*

Figure 12 *Illustrated the similar shape between different plant leaves. The leaf images of a) star apple and b) pomme jacquot leaves.*

2.3.2 Dataset Pre-processing

The pre-processing of the plant leaf images from the Folio dataset is very simple. The process starts by converting all the images to black and white to find the plant leaf area, called the region of interest (ROI). Then, crop the plant image according to the ROI. The following process is to check the leaf images which are in the horizontal

or vertical positions. If a particular image is in the horizontal position, then the rotation method is used to rotate the image into the vertical position, as shown in Figure 10, followed by resizing all images to the vertical size of 400 pixels. So, the width of each image will have different sizes depending on the actual size to avoid image distortion. Consequently, the feature extraction methods are computed from this process.

2.4 Experimental Results

We compared the feature extraction techniques, color histogram, LBP, HOG, PCA, and HOG-BOW, to two deep learning techniques: AlexNet and GoogLeNet.

In this experiment, 10-fold cross-validation was examined to evaluate the results of the plant leaf recognition methods. We used a training set of 80% of 637 images in total. For the evaluation metric, the recognition rate (accuracy) and standard deviation were used to measure the performance of each feature extraction technique. Moreover, we used SVM algorithm and then the grid-search technique was used to search for the best parameters. We found that the best C and gamma parameters of the SVM with the RBF kernel were 100 and 0.1, respectively. For MLP, two hidden layers were used where the size of each hidden layer is 512 and 512 hidden units, respectively. The dropout regularization was used to prevent neural networks from overfitting. The dropout rates of 0.5 for all hidden units were selected. Further, as for the output layer, the softmax function was used. Tables 1 and 2 show the experimental results (average test accuracy and standard deviation).

Table 1 *Plant leaf recognition results of the 15 different techniques on the Folio dataset*

Multiple Grid Methods	Training Time (Sec)		Test Accuracy (%)	
	SVM	MLP	SVM	MLP
Color-Histogram	221.86	232.42	96.25±1.87	95.94±1.94
LBP	278.80	284.80	94.45±1.06	91.87±2.22
HOG	201.27	206.83	94.14±2.45	94.14±2.34
Color-Histogram-PCA	182.88	189.49	97.73±1.30	97.11±1.28
LBP-PCA	278.15	285.29	94.14±1.06	94.14±1.74
HOG-PCA	202.12	209.53	93.83±2.62	93.91±1.83
Color-Histogram-LBP	496.61	511.65	97.81±1.15	96.09±1.65
Color-Histogram-HOG	419.10	435.47	98.13±1.39	96.64±1.38
LBP-HOG	481.14	489.10	97.50±1.46	96.87±1.98
Color-Histogram-LBP-HOG	697.46	716.77	98.67±0.91	97.42±1.48
Color-Histogram-LBP-PCA	460.96	469.78	98.67±1.11	98.28±1.51
Color-Histogram-HOG-PCA	384.91	393.20	98.59±1.46	98.28±1.32
LBP-HOG-PCA	480.19	488.94	97.50±1.46	97.58±1.01
Color-Histogram-LBP-HOG-PCA	663.01	672.19	**99.06±0.89**	**98.75±0.92**
HOG-BOW (Pawara, Okafor, Surinta, et al., 2017)	-	-	92.78±2.17	92.37±1.78

Table 2 *Comparing results between proposed methods and fine-tuned deep learning methods on the Folio dataset*

Method	Test Accuracy (%)
AlexNet (Pawara, Okafor, Surinta, et al., 2017)	97.67±1.60
GoogleNet (Pawara, Okafor, Surinta, et al., 2017)	97.63±1.84
Proposed Method (Color-Histogram-LBP-HOG-PCA)	99.06±0.89

The results in Table 1 showed the recognition performances obtained from the combination of multiple grid approaches with various feature extraction techniques, the result of the HOG-BOW method, and the training time on the Folio dataset. Then, 15 different results are shown in Table 1. Here, the HOG-BOW method achieved an inferior performance compared to other feature extraction techniques. On the other hand, the Color-Histogram-LBP-HOG-PCA, when combined with the SVM algorithm with the RBF kernel, significantly outperformed other techniques and provided a high accuracy of 99.06%. Subsequently, the plant leaf recognition obtained a high accuracy of 98.75% when combined with the Color-Histogram-LBP-HOG-PCA and MLP algorithm.

We also compared our proposed method with the find-tuned deep CNNs, which are AlexNet and GoogleNet architectures (Pawara, Okafor, Surinta, et al., 2017). The accuracy of results compared between our proposed method and the fine-tuned deep CNNs are shown in Table 2.

2.5 Conclusion

In this paper, we investigated many different plant leaf recognition techniques on the Folio dataset. From the experimental results, we concluded that the performance of multiple grids and dimensionality reduction-based descriptors, which is our proposed method, was much better than the histogram of oriented gradients combined with the bag-of-words technique and fine-tuned deep CNN architectures, which are AlexNet and GoogleNet architectures as well. We also showed that PCA, which is a dimensionality reduction technique, increased the accuracy performance and decreased the number of feature vectors of the plant leaf recognition system. Because of the high accuracy of the deep CNNs, in future work, we would like to study the effect of parallel CNN architecture and use this architecture to train the plant leaf images. This technique may be necessary to improve training times and accuracy performance.

Chapter 3

Ensemble Learning Methods with Deep Convolutional Neural Networks

Recognition of plant leaves and diseases from images is a challenging task in computer vision and machine learning. This is because various problems directly affect the performance of the system, such as the leaf structure, differences of the intra-class, similarity of shape between inter-class, perspective of the image, and even recording time. In this paper, we propose the ensemble convolutional neural network (CNN) method to tackle these issues and improve plant leaf recognition performance. We trained five CNN models; MobileNetV1, MobileNetV2, NASNetMobile, DenseNet121, and Xception, accordingly to discover the best CNN based model. Ensemble methods; unweighted average, weighted average, and unweighted majority vote methods, were then applied to the CNN output probabilities of each model. We have evaluated these ensemble CNN methods on a mulberry leaf dataset and two leaf disease datasets; tomato and corn leaf disease. As a result, the individual CNN model shows that MobileNetV2 outperforms every CNN model with an accuracy of 91.19% on the mulberry leaf dataset. The Xception combined with data augmentation techniques (Height Shift+Vertical Flip+Fill Mode) achieved an accuracy of 91.77 %. We achieved very high accuracy above 99% from the DenseNet121 and Xception models on the leaf disease datasets. For the ensemble CNNs method, we selected the based models according to the best CNN models and predicted the output of each CNN with the weighted average ensemble method. The results showed that 3-Ensemble CNNs (3-EnsCNNs) performed better on plant leaf disease datasets, while 5-EnsCNNs outperformed on the mulberry leaf dataset. Surprisingly, the data augmentation technique did not affect the ensemble CNNs on the mulberry leaf and corn leaf disease datasets. On the other hand, application of data augmentation was slightly better than without only on the tomato leaf disease dataset.

3.1 Introduction

Plants are essential to human life and can be used as food and even medicine (Du et al., 2007). There is a wide diversity of plants in nature. Importantly, some plant leaves look very similar, such as the shape of the Japanese maple and coral plants or cannabis. It is quite difficult for people who are not familiar with the plants to identify them. Thus, the identification of plants requires expertise, such as that of taxonomic botanists, and plant scientists. Therefore, researchers have implemented plant identification systems so that people without botanic knowledge can use them as an identification tool to recognize the plant species (Vo et al., 2019). Usually, plants can be classified from various components, called plant organs, such as leaves, flowers, bark, seeds, and stems. However, if we want to consider plant diseases, most diseases are determined by the leaves. Therefore, researchers have used plant leaves to classify plant categories and diseases (Aravind et al., 2018; Mokeev, 2019; Munisami et al., 2015; Pawara, Okafor, Surinta, et al., 2017). Importantly, the spread of disease is a big problem for agriculturists because it affects agricultural products and profit on trading. It is

necessary for farmers to inspect agricultural products early to prevent and treat the disease in time.

Due to the fast spread of disease, many researchers have proposed artificial intelligence systems to stop disease spread and recognize the disease types. Many benchmark plant leaf and plant leaf disease datasets were compiled, such as PlantCLEF, Leafsnap, PlantVillage,and PlantDoc (Goëau, Bonnet, & Joly, 2015; Hughes & Salathé, 2015; N. Kumar et al., 2012; Singh et al., 2020), to create effective learning models. The plant leaf images in the benchmark datasets were typically collected from natural environments. Hence, these collections of leaf images are more complex than images collected under standardized conditions in a laboratory, such as camera angles when capturing the leaves, different objects appear in the image, brightness and contrast while taking the picture, zoom in and out into the leaf, and even loss of focus. These issues affect the accuracy of the plant leaf recognition systems.

The objective of this research is to improve plant leaf recognition based on the ensemble CNN method.

The following are contributions of this research;

1. In this paper, we propose the ensemble convolutional neural network (CNN) method to overcome challenges in plant leaf and plant leaf disease recognition. To discover the best CNN model, we first trained five CNN models consisting of MobileNetV1, MobileNetV2, NASNetMobile, DenseNet121, and Xception. Second, we chose the best three and five CNN models, called 3-EnsCNNs and 5-EnsCNNs. Finally, the CNN output probabilities of each CNN model were then given to the ensemble method to do the actual classification.

2. We compared three ensemble methods, namely the unweighted majority vote, unweighted average, and weighted average, for the plant leaf and plant leaf disease recognition. The experimental results showed that the weighted average method outperformed the other ensemble methods and was also significantly better than the individual CNN model.

3. This paper also provides a new standard mulberry leaf dataset for comparison of image recognition methods. The mulberry leaf dataset contains 5,262 leaf images and includes 10 species that grow in Northern and Northeast Thailand.

3.2 Related Work

Image processing and machine learning techniques have been proposed to address plant leaf recognition problems. Wang et al. (2008) proposed a framework for recognizing the plant leaf with a complicated background. The feature vector was extracted from the shape of the leaf using Hu geometric and Zernike orthogonal moments. The moving center hypersphere method was used as a classifier. The feature vector was also extracted from the shape, color, edge, and direction of the plant leaf (Patil, Pattanshetty, & Nandyal, 2013; Wang et al., 2008). The feature vector was recognized using machine learning techniques, such as K-nearest neighbour (KNN) and support vector machine (SVM). Chompookham et al. (2020) presented a multiple grid method that divided the plant leaf images into sub-regions. The feature extraction techniques, including a histogram of oriented gradients (HOG), local binary pattern (LBP), and color histogram, were proposed to extract features from each sub-region.

Principal component analysis (PCA) was used to reduce the dimension of the feature vector. The most correlated variables were given to the SVM classifier.

CNN methods are currently employed to recognize plant leaves and diseases. Atabay (2016) invented a new CNN architecture with five layers for plant leaf datasets. This CNN architecture comprises four sets of convolutional layers and one soft max layer. In this architecture, the exponential linear unit (ELU) is employed instead of the rectified linear unit (ReLU) after the max-pooling layer. The proposed CNN architecture provided 97.27% accuracy on the Flavia dataset and 99.11% on the Swedish dataset. Jeon and Rhee (2017) modified the GoogLeNet architecture for plant leaf recognition by changing the first Inception layer from 3 to 5 layers. The modified GoogLeNet architecture performed better with the leaf image dataset and the damaged leaf image. Furthermore, Pawara et al. (2017) proposed to use AlexNet and GoogLeNet for three plant leaf datasets: AgrilPlant, LeafSnap, and Folio. Two strategies of training, training from scratch and fine-tuned, were presented. The performance data showed that GoogLeNet with fine-tuning outperformed AlexNet with fine-tuning and training from scratch on AgrilPlant and LeafSnap datasets, whereas AlexNet with fine-tuning, showed the best performance on the Folio dataset. Additionally, the CNN architectures performed around 20% better than BOW and local descriptor combined with the machine learning techniques (KNN, SVM, and MLP).

Pawara et al. (2017) used both scratch and pre-trained weights to train the AlexNet and GoogLeNet models. The data augmentation techniques, including rotation, blur, contrast, scaling, illumination, and projective transformation, were used to generate new images. With these data augmentation techniques, the size of the training set was increased by 25 times. As a result, the CNN model that trained from scratch obtains more benefits from data augmentation techniques. For the Swedish dataset, the fine-tuned AlexNet and GoogLeNet achieved 99.76% and 99.92% accuracy. For the Folio dataset, approximately 99% accuracy was achieved from the fine-tuned AlexNet and GoogLeNet. Consequently, the results of the AgrilPlant were 97.27% with fine-tuned AlexNet and 98.60% with fine-tuned GoogLeNet. Moreover, Kumar and Vani (2019) compared four CNN architectures of LeNet, VGGNet, Xception, and ResNet50, and trained from scratch for tomato leaf disease recognition. The result illustrated that the VGGNet outperformed other CNN models.

For the ensemble CNN method, a two-level architecture, called stacked CNN (Mokeev, 2019) was proposed. In the first level, two CNN models are created by learning the data from the plant dataset. In the second level, the predictive values of the CNN models are then learned again using machine learning techniques, such as random forest, gradient boosting, and extreme gradient boosting. As a result, the stacked CNN combined with the gradient boosting classifier was the best method and obtained an F1-score of 0.953. Moreover, the ensemble CNN method can also compute the probability output obtained from CNN models to find the final result. Three ensemble methods comprised an unweighted majority vote (Surinta, Schomaker, & Wiering, 2013), unweighted average, and weighted average ensemble methods (Guo et al., 2019).

The CNN architectures were proposed to address many recognition applications (Kreuter, Takahashi, Omae, Akiduki, & Zhang, 2020; S. Park et al., 2020). Also, the recognition performance was enhanced when the ensemble method was combined. In this study, we proposed the framework of the ensemble CNN.

3.3 Ensemble Convolutional Neural Networks Framework

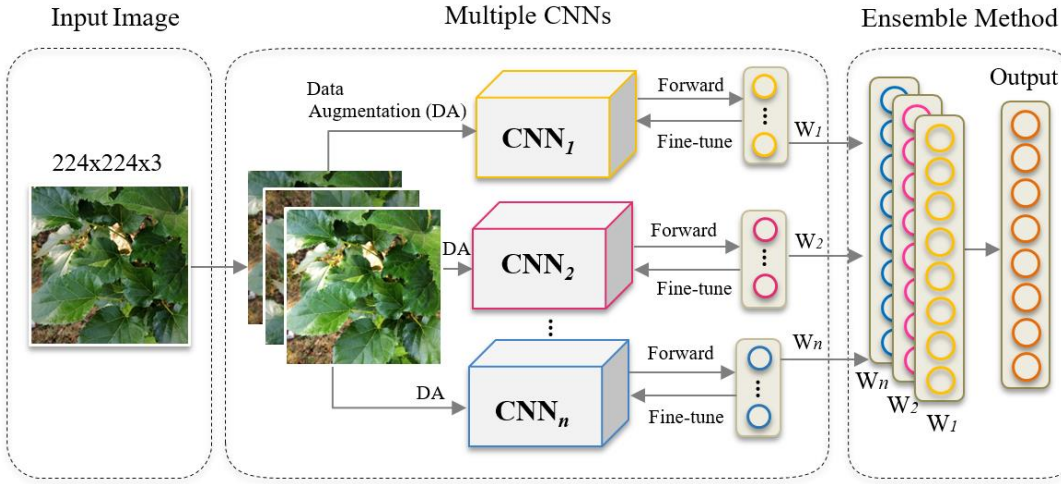


Figure 13 *The framework of the proposed ensemble CNNs*

The effect of multiple CNN models on the ensemble learning framework is regularly better than a single CNN model because ensemble learning can well integrate the advantages of multiple CNN models (Zhang, Yan, Ma, & Xu, 2020). In this section, we introduce the ensemble CNNs framework to address the plant leaf recognition problems, as shown in Figure 13. The first part of the framework combines with state-of-the-art CNN architectures, called multiple CNNs. In order to find the baseline CNN models, five pre-trained CNN models: MobileNetV1, MobileNetV2, NASNetMobile, DenseNet121, and Xception are proposed. Subsequently, the transfer learning and data augmentation techniques are applied in this step. The details of the ensemble CNNs are described in Section 3.3.1 in the second part, the output probabilities of the CNN models are given to be recognized by the ensemble methods. We propose to use three ensemble methods, namely the unweighted majority vote, unweighted average, and weighted average to do the actual classification. The ensemble methods are explained in Section 3.3.2

3.3.1 Multiple Convolutional Neural Networks

This section briefly describes the CNN that is combined in multiple CNNs: MobileNetV1, MobileNetV2, Xception, DenseNet121, and NASNetMobile. We also present the optimization algorithms (stochastic gradient descent and RMSProp) that are applied to optimizing the CNN model, as follows.

3.3.1.1 Convolutional Neural Network Architectures

MobileNetV1. MobileNetV1 was proposed by Howard et al. (2017). It was designed to address a huge number of parameters by using factorized convolutions, which included depthwise and pointwise convolutions, called depthwise separable convolution. Due to the depthwise convolutions, each input channel is computed with

the kernel size of 3×3. The output of the depthwise convolutions decreased from 3×3×3 to 1×1×3 convolutions. After that, the depthwise convolutions reduced to 1×1 convolutions, called pointwise convolutions.

***MobileNetV2*.** MobileNetV2 was proposed by Sandler et al. (2018), which improved on MobileNetV1 (Howard et al., 2017b). The MobileNetV2 architecture comprises 11 layers: one convolution layer, seven inverted residual blocks, one convolution layer, one average pooling layer, and one convolution layer. The inverted residual block contains three layers: 1×1 convolution with ReLU6 activation function, depthwise separable convolution with ReLU6, and 1×1 convolution with the linear transformation.

***Xception*.** Chollet (2016) designed the Xception architecture. This architecture is the extreme version of the inception module and was implemented to address the problems of deeper networks, computation time, and overfitting. The depthwise separable convolutions are applied in the extreme inception module. The Xception architecture is divided into three main flows: entry, middle, and exit. In the entry flow, the first layer is the input image with 229×229×3 pixels, followed by 32 convolution layers with ReLU, 64 convolution layers with ReLU, and three residual connections. In the middle flow, eight stacked residual connections are attached. The exit flow is a stack of one residual connection, followed by two depthwise separable convolutions and global average pooling.

***DenseNet121*.** In 2018, Huang, Liu, Van Der Maaten, & Weinberger (2018) invented DenseNet architecture. In this architecture, the knowledge is collected according to the connections from the current layer and are combined in the following layers, called DenseNet. The DenseNet architecture contains a convolution layer, pooling layer, three dense blocks and transition layers, one dense block, and a classification layer. According to the size of the bottleneck, the layers of the DenseNet can increase from 121 to 264 depth. The concept of the growth rate of the convolution layers was implemented, and then, the next convolution layer was double increased. The bottleneck structure is implemented and directly impacts a decrease in the number of the parameters. Also, the number of the parameters of the DenseNet architecture is smaller than that of the ResNet architecture.

***NASNetMobile*.** Zoph et al. (2018) proposed a neural architecture search, called NASNet. The NASNet architecture can also be scalable by increasing normal and reduction cells using a recurrent neural network (RNN). Then, reinforcement learning was proposed to search for the best architecture. Also, the NASNet architectures consist of NASNetLarge and NASNetMobile.

3.3.1.2 Optimization Algorithms for CNN Architectures

The optimization algorithms were invented to deal with minimizing the objective function (P. Li, 2017). Consequently, the best optimizer can guarantee the optimal value with fast learning and obtain more reliable performance. We briefly explain two optimization algorithms used in our experiments as follows.

Stochastic Gradient Descent (SGD). One of the most popular optimization algorithms is the SGD algorithm. In the SGD optimizer, the algorithm allows updating the parameter until it converges to the minimum and enables moving to the better local minima (Ruder, 2017). The SGD optimizer can be computed as:

$$\theta = \theta - \eta \cdot \nabla_\theta J\left(\theta; x^i; j^i\right) \tag{6}$$

Where θ is objective function, η is learning rate, $\theta = \theta - \eta \cdot \nabla_\theta J\left(\theta; x^i; j^i\right)$ update the parameters of the objective function, and $x^i, j^i$are training examples and labels.

RMSProp. Hinton et al. (2012) invented a mini-batch version of the RProp algorithm, namely the RMSprop algorithm. The RMSprop algorithm is the adaptive learning rate method. It uses the sign gradient to calculate and update the value of the learning rate (Ruder, 2017).

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$$
$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma) g_t^2 \tag{7}$$

where $E[g^2]_t = \eta E[g^2]_{t-1} + (1-\eta) g_t^2$ is squared gradients for each weight, $g_t$is the gradient of the cost function, γ is a decay constant. Note that the best values of decay constant and learning rate are 0.9 and 0.001 (P. Li, 2017; Ruder, 2017).

3.3.2 Ensemble Methods.

In this section, the idea of the ensemble method combines with several weights (see Figure 13) that are learned from the CNN models to generate the optimal predictive model. In this section, we mainly emphasize three ensemble methods as follows.

Unweighted Average. The most common of the ensemble methods is the unweighted average method. In this method, first, the probability values ($w_{1,} w_{2,} \dots, w_n$), which is the output of the last layer of the CNN models, are calculated using the softmax activation function (Ju et al., 2018). Second, we average all the probability values of the CNN models and selected the highest probability as a result. The unweighted average method is computed as $p' = \frac{1}{n}\sum_{i=1}^{n} \vec{y}$, where $\vec{y}$ is the weight vector and n is the number of ensemble CNN models.

Unweighted Majority Vote. In this method, instead of averaging all the probability values of the CNN models, the highest probabilities are selected as the output. Then, it votes by counting the majority from all the predicted labels and makes a final decision (Harangi, 2018). The unweighted majority vote method is calculated as $p' = \frac{1}{n}\sum_{i=1}^{n} argmax\,(\vec{y})$, where arg max ($\vec{y}$) is the highest probability value of weight vector $\vec{y}$ and n is the number of ensemble CNN models.

Weighted Average. The weighted average method is the extended version of the unweighted average by multiplying the different weight values to the CNN outputs (Harangi, 2018). Additionally, the sum of weight values is equal to one. The equation of the weighted average method is given by $p' = \frac{1}{n}\sum_{i=1}^{n} \alpha_i \vec{y}$, where α is the weight values that multiply with the weight vector $\vec{y}$ and n is the number of ensemble CNN models.

3.4 Plant Leaf Datasets

In the section, we introduce the benchmark mulberry leaf dataset. We collected mulberry leaves that were growing in Thailand. In this dataset, the mulberry leaf images are diverse in brightness, shadow, and even camera angles because the images were captured from the natural environment. We provide the mulberry leaf dataset with the aim of plant leaf recognition. We have also evaluated the deep learning algorithms for the tomato and corn leaf disease datasets classification, which is the subset of the PlantVillage dataset.

3.4.1 Mulberry Leaf Dataset

The mulberry leaf dataset is a collection of images of 10 cultivars that were taken in natural environments using DSLR cameras and smartphones. We collected the data from three regions of Thailand: northern (Chiang Mai), central (Phitsanulok), and northeast (Nakhon Ratchasima, Buriram, and Maha Sarakham) The mulberry field areas are shown in Figure 14. In this research, the mulberry leaf images were captured from the natural environments, as shown in Figure 15. We recorded the images from different perspectives. There is a shadow that appears in the photograph when holding the camera at a low position. However, when shooting from an eye-level position, the resulting image is sharp and the image is not then backlit. All leaf images were recorded in the JPEG format.

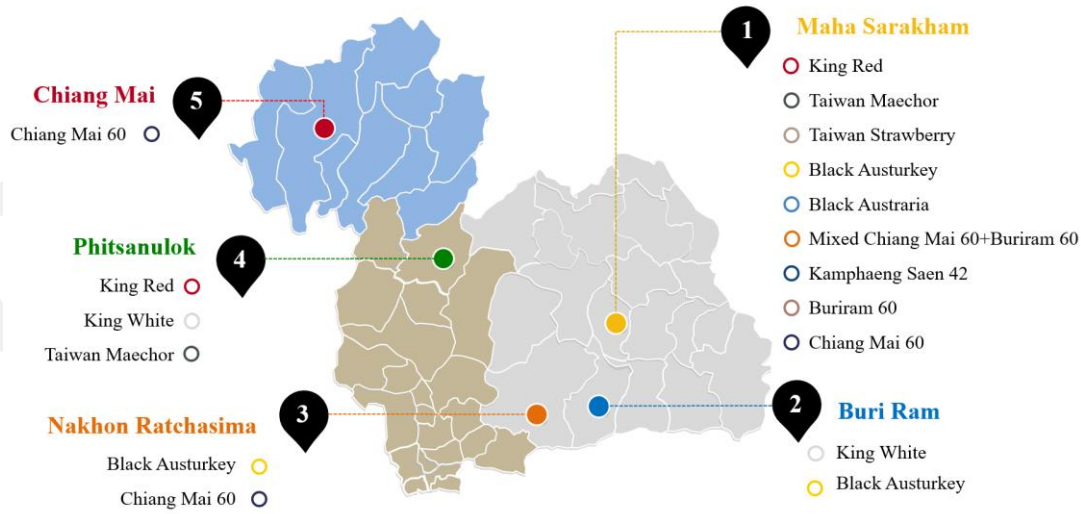


Figure 14 *Illustration of the mulberry field area in Thailand has been collected as a dataset in this study consisting of Maha Sarakham, Buriram, Nakhon Ratchasima, and Phitsanulok, and Chiang Mai.*

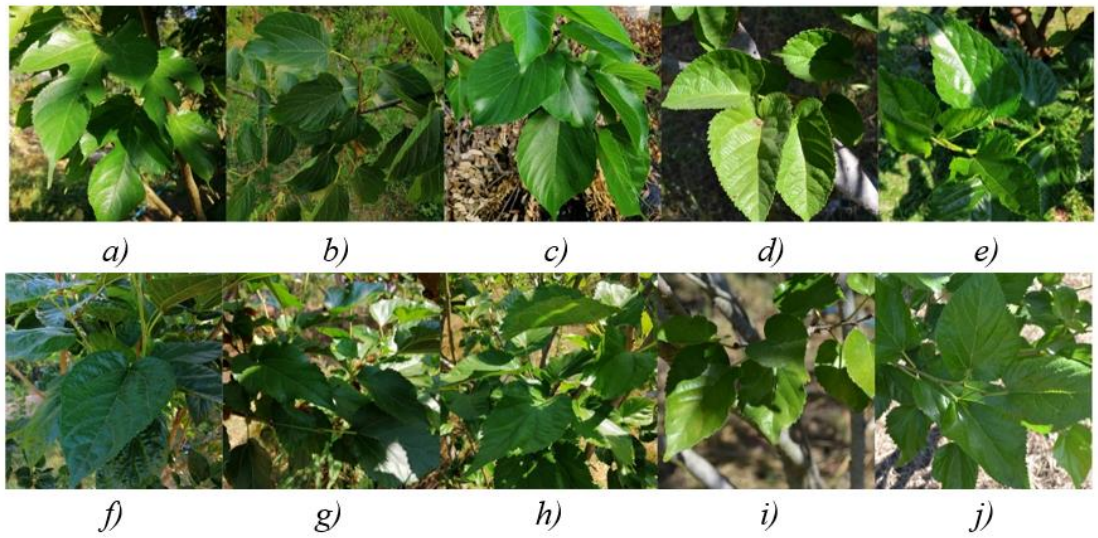


Figure 15 *Illustration of the ten mulberry leaf cultivars including a) KingRed, b) King White, c) Taiwan Maechor, d) Taiwan Strawberry, e)Black Austurkey, f) Black Australia, g) Chiang Mai 60, h) Buriram 60,i) Kamphaeng Saen 42, and j) Mixed Chiang Mai 60+Buriram 60*

The mulberry leaf images were resized to 224×224 pixel resolution. The mulberry leaf dataset includes ten cultivars, which are four cultivars from Thailand: Chiang Mai 60 (386 images), Buriram 60 (500 images), Kamphaeng Saen 42 (640 images), and 761 images of mixed-breed mulberry (Chiang Mai 60 + Buriram 60). Three cultivars of Australia consist of King Red (500 images), King White (350 images), and BlackAustralia (637 images). Two cultivars of Taiwan consist of Taiwan Maechor (500 images) and Taiwan Strawberry (500 images). Also, 488 images of the Black Austurkey are from Turkey. This dataset contains 5,262 images in total. Note that mulberry experts advised examination of each mulberry species to label the data and avoid the errors due to the similarity pattern and shape of the leaves.

3.4.2 PlantVillage Dataset

The PlantVillage dataset is a collection of plant images proposed by Penn State University (Hughes & Salathé, 2015) that collects various plant leaves and plant leaf diseases. The PlantVillage dataset has 54,309 images. In our study, we selected only tomato and corn leaf disease datasets. The details of these datasets are as follows.

3.4.2.1 Tomato Leaf Disease Dataset. This dataset consists of 10 categories: nine diseased tomato leaves and one healthy leaf (Durmuş et al., 2017; A. Kumar & Vani, 2019). It contains 18,162 tomato leaf images, including 2,127 bacterial spots, 1,000 early blights, 1,910 late blights, 952 leaf mold, 1,771 septoria leaf spot, 1,676 spider mites, two-spotted spider mite, 1,404 target spots, 373 tomato mosaic virus, 5,357 tomato yellow leaf curl virus, and 1,592 healthy tomato leaves. The tomato leaf diseases dataset is shown in Figure 16.

3.4.2.2 Corn Leaf Disease Dataset. This dataset contains four classes and has 3,852 images (Aravind et al., 2018; Kusumo et al., 2018). One healthy category has 1,162 images and three corn leaf diseases: 513 images of cercospora leaf spot, gray leaf spot, 1,192 images of common rust, and 985 images of northern leaf blight. The corn leaf disease dataset is illustrated in Figure 17.

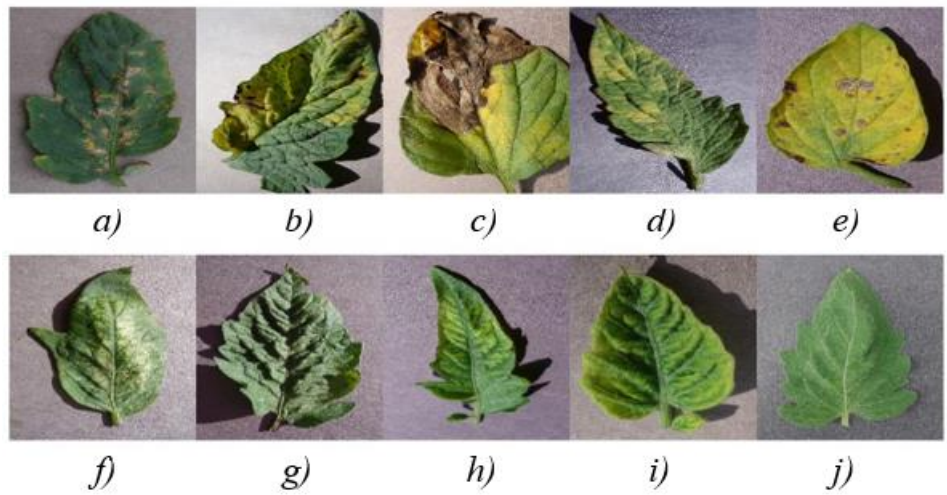


Figure 16 *Examples of leaf disease datasets: tomato leaf disease images, including a) bacterial spot, b) early blight, c) late blight, d) leaf mold, e) septoria leaf spot, f) spider mites two-spotted spider mite, g) target spot, h) tomato mosaic virus, i) tomato yellow leaf curl virus, and j) healthy, respectively*

Figure 17 *Examples of leaf disease datasets: corn leaf disease images, including a) cercospora leaf spot gray leaf spot, b) common rust, c) northern leaf blight, and d) healthy (from left to right)*

Table 3 *The best training hyperparameters and the accuracy (%) of each single model obtained with 5-fold cross-validation and test set on the mulberry leaf dataset.*

Models	Optimizer	Learning Rate	Batch Size	Validation	Test Accuracy
MobileNetV1	RMSprop	0.0001	8	97.35 ± 0:005	89.83
MobileNetV2	RMSprop	0.0001	16	97.08 ± 0:006	**91.19**
NASNetMobile	RMSprop	0.0001	8	97.38 ± 0:004	86.65
DenseNet121	SGD	0.01	8	**98.61 ± 0:002**	90.80
Xception	RMSprop	0.0001	8	97.94 ± 0:006	91.00

3.5 Experimental Setup and Results

3.5.1 Mulberry Leaf Dataset

Mulberry Leaf Dataset. In this experiment, the pre-trained models of five CNNs consisting of MobileNetV1, MobileNetV2, NASNetMobile, DenseNet121, and Xception, were used as the initial weight, and then trained on the Mulberry leaf dataset. The results in Table 1 are based on 5-fold cross-validation to avoid overfitting and on an independent test set. The training set contained 3,719 images and the independent test set included 1,543 images. The experimental settings used to train the CNN models were as follows; image resolution is 224×224 pixels, the three optimization algorithms were SGD, Adam, and RMSprop, the learning rate was 0.1, 0.01, 0.001, and 0.0001, the batch size was 8, 16, 32, and 64, and the number of iterations was 500 epochs.

For the experimental results, we discovered that the RMSprop optimizer achieved higher accuracy when training with MobileNetV1, MobileNetV2, NASNetMobile, and Xception models. The SGD optimizer gave better results when training with the DenseNet121. In contrast, Adam Optimizer showed worse performance on all CNN models. The best parameters for each CNN model are shown in Table 3.

From the results in Table 3, it can be seen that DenseNet121 outperforms other CNN methods with a cross-validation accuracy of 98.61%. Moreover, MobileNetV2 was the best CNN model when applied to the test set. The recognition performance of MobileNetV2 was 91.19%, while the worst recognition performance was NASNetMobile. We evaluated the single CNN model using 5-fold cross-validation on the mulberry dataset to avoid overfitting. The result showed that all CNN models achieved high accuracy and low standard deviation values. The accuracy of the CNN models was slightly decreased by approximately 7% on the independent test set. Consequently, it is guaranteed that these CNN models are not overfitted on the tomato and corn leaf disease datasets when using the same CNN parameters.

In Table 4, we show the experimental results with the data augmentation techniques and CNN models on the mulberry dataset. We compared six data augmentation techniques (DA) consisting of DA1-Height Shift, DA2-Vertical Flip, DA3-Fill Mode, DA4-Height Shift+Fill Mode, DA5-Height Shift+Vertical Flip+Fill Mode, and DA6-Mixed DA. We defined the data augmentation as; Height shift = 0.25, Fill Mode = Reflect, and Flip Vertical = True. The experiments showed that the best performance was with the Xception model when training the model with three data augmentation

Table 4 *Performances evaluation of the CNNs and data augmentation techniques on the mulberry leaf dataset.*

Data Augmentation	Test Accuracy				
	MobileNetV1	MobileNetV2	NASNet Mobile	DenseNet121	Xception
DA1	86.58	91.12	88.42	90.03	90.45
DA2	89.31	90.02	88.69	90.01	90.25
DA3	87.49	90.02	88.69	90.01	90.25
DA4	88.20	90.80	88.66	90.47	88.53
DA5	88.85	**91.32**	**90.15**	90.67	**91.77**
DA6	**90.34**	84.06	89.63	**91.06**	84.12

Techniques (DA5) (height shift, vertical flip, and fill mode) with the accuracy of 91.77%. Xception outperforms every CNN model. Subsequently, the CNN models that obtained high accuracy when combined with three data augmentation techniques were MobileNetV2, and NASNetMobile although mobileNetV1 and DenseNet121 models achieved results higher than 90% when mixed with 10 data augmentation techniques (DA6).

3.5.2 Experiments on the Tomato and Corn Leaf Datasets

In this section, we compared CNN architectures composed of VGG16, MobileNetV1, MobileNetV2, NASNetMobile, DenseNet121, and Xception to obtain the best performance on the tomato and corn leaf disease datasets. The best data augmentation techniques that we found from Table 4 were also applied to training the CNNs. For the leaf (tomato and corn) disease datasets, we divided 90% of data as a training set and 10% as a test set. The test accuracy is shown in Table 5 and Table 6.

From the results in Table 5, it can be seen that Kumar and Vani (2019) proposed VGG16 for recognition in the tomato leaf disease dataset and achieved an accuracy of 99.25% without applying the data augmentation technique. In these experiments, we considered training the CNN models applying the data augmentation and without

Table 5 *Performance evaluation of the CNNs on the tomato leaf disease dataset.*

Methods	Model Size	Data Augmentation				
		No DA	**DA3**	**DA4**	**DA5**	**DA6**
VGG16 (A. Kumar & Vani, 2019)	N/A	99.25	-	-	-	-
MobileNetV1	25.0 MB	99.26	99.60	99.26	99.46	99.33
MobileNetV2	17.9 MB	99.26	99.26	99.86	99.13	99.20
NASNetMobile	37.3 MB	99.33	99.46	99.73	99.26	99.53
DenseNet121	27.9 MB	99.46	99.53	**99.87**	99.53	99.66
Xception	159 MB	99.66	99.73	99.20	**99.87**	99.80

Table 6 *Performance evaluation of the CNNs on the corn leaf disease dataset.*

Methods	Model Size	Data Augmentation (DA)				
		No-Aug	**DA3**	**DA4**	**DA5**	**DA6**
RGB+LinearSVM (Kusumo et al., 2018)	N/A	88.00	-	-	-	-
BoF+LinearSVM (Aravind et al., 2018)	N/A	83.70	-	-	-	-
MobileNetV1	24.9 MB	97.92	98.44	99.20	98.44	99.20
MobileNetV2	17.9MB	98.18	97.66	99.21	97.40	98.18
NASNetMobile	37.3 MB	98.44	95.83	99.21	98.96	99.21
DenseNet121	27.9 MB	98.44	98.70	98.70	98.70	98.96
Xception	159 MB	98.70	98.70	98.44	98.70	**99.22**

applying the data augmentation techniques. The results showed that the DenseNet121 and Xception using data augmentation surpassed all CNN models with an accuracy of 99.87%.

As seen in Table 6, accurate results appeared when applying the data augmentation techniques. It shows that Xception combined with mixed data augmentation techniques (DA6) provided an accuracy of 99.22%. Moreover, the MobileNetV2 and NASNetMobile combined with two data augmentation techniques (DA4 – Height Shift+Fill Mode), and NASNetMobile combined with mixed data augmentation techniques (DA6), provided an equal accuracy of 99.21%. Furthermore, without applying the data augmentation technique, CNN architecture still showed a better result than the previous studies with an improvement of approximately 10% in accuracy.

3.5.3 Experiments of the Ensemble CNN models on the Plant Leaf Datasets.

As can be seen from Table 3, we decided to use the three best CNNs to construct the ensemble CNNs, including Xception, MobileNetV2, and DenseNet121, called 3 Ensemble CNNs (3-EnsCNNs). We also selected five CNNS (MobileNetV1, MobileNetV2, NASNetMobile, DenseNet121, and Xception) combined with data augmentation techniques, called 5 Ensemble CNNs (5-EnsCNNs). In these experiments, the outputs after applying the softmax function of every single CNN model were then used in the decision layer of the ensemble method. We used three ensemble methods to recognize the plant leaf datasets, including the unweighted majority vote, unweighted average, and weighted average methods.

Table 7 shows the results obtained with the ensemble CNNs. For the ensemble methods, the results emphasize that the weighted average outperforms the unweighted majority vote and average methods on three plant leaf datasets. Subsequently, the 3-EnsCNNs performed better than 5-EnsCNNs on tomato and corn leaf disease datasets, except for the mulberry leaf dataset that obtained the best result when applying 5-EnsCNNs. The data augmentation techniques, surprisingly, without the data

Table 7 *Performance of the ensemble CNN methods applied on plant leaf datasets.*

Datasets/DA	Test accuracy (%)					
	Unweighted Majority Vote		**Unweighted Average**		**Weighted Average**	
	3-Ens CNNs	**5-Ens CNNs**	**3-Ens CNNs**	**5-Ens CNNs**	**3-Ens CNNs**	**5-Ens CNNs**
Mulberry leaf dataset						
No DA	92.81	93.65	94.55	94.68	94.49	**94.75**
DA	92.61	92.81	94.03	94.23	94.41	94.55
Tomato leaf disease dataset						
No DA	99.20	99.20	99.79	99.86	99.86	99.79
DA	99.26	99.20	99.86	99.79	**99.93**	99.86
Corn leaf disease dataset						
No DA	98.44	98.70	99.45	99.21	**99.47**	99.24
DA	98.44	98.70	99.21	99.21	99.31	99.30

augmentation technique show the best accuracy on mulberry leaf and corn leaf disease datasets. On the other hand, the recognition performance with application of the data augmentation technique was 99.93% on the tomato leaf disease dataset.

As a result, the weighted average ensemble approach also achieved accuracies of 99.93% and 99.47% on the tomato and corn leaf disease dataset, respectively. The results lead us to conclude that the ensemble methods can increase the performance of the CNN architectures.

3.6 Conclusion

In this paper, we have proposed ensemble CNN architectures to improve recognition performance on the plant leaf datasets. In order to obtain the CNN based models, we first compared five state-of-the-art CNNs: MobileNetV1, MobileNetV2, NASNetMobile, DenseNet121, and Xception. The CNN models were trained with a transfer learning technique and the training sample enlarged using data augmentation techniques. We evaluated five CNN models on the mulberry leaf dataset and two plant leaf disease datasets: tomato and corn. Second, we selected the three best CNN models to establish the ensemble CNNs: Xception, MobileNetV2, and DenseNet121, called 3-EnsCNNs. Additionally, five CNN models: MobileNetV1, MobileNetV2, NASNetMobile, DenseNet121, and Xception, were applied as 5-EnsCNNs. Finally, three ensemble methods: the unweighted majority vote, unweighted average, and weighted average methods, were proposed to classify the output of CNN models. The weighted average method was selected from the best experimental result.

With the individual CNN model, the DenseNet121 achieved 98.61% accuracy with cross-validation and outperformed all models. Additionally, MobileNetV2 showed the highest performance on the test set of the mulberry leaf dataset with an accuracy of

91.19%. In the best of our experiments, the data augmentation techniques: Height Shift, Vertical Flip, and Fill Mode, could slightly improve the performance of the CNN models, especially by significantly increasing the efficiency of the Xception model. The Xception combined with data augmentation techniques obtained an accuracy of 91.77%. For tomato and corn leaf disease datasets, the DenseNet121 and Xception achieved very high accuracy above 99%. Our experimental results also achieved high accuracy when compared to previous work.

To create a mobile application to address the issue of plant leaf recognition, we recommend using the CNN models of MobileNetV2, MobileNetV1, DenseNet121, and NASNetMobile, respectively. These CNN models provided accuracy above 99% on the tomato and corn leaf disease datasets. The size of these CNN models is approximately 25-40 MB, which is relatively small. In comparison, we recommend using MobileNetV2 for plant leaf recognition.

As for the ensemble CNN, the experimental results showed that the 3-EnsCNNs achieved the highest accuracy performance on the tomato and corn leaf disease datasets. Moreover, 5-EnsCNNs outperformed 3-EnsCNNs only on the mulberry leaf dataset. Surprisingly, ensemble CNN without data augmentation techniques achieved the highest accuracy on two plant leaf datasets, mulberry and corn. However, more than 99% accuracy was obtained from the tomato and corn leaf disease datasets. The highest accuracy of 94.75% was obtained with the mulberry leaf dataset because the tomato and corn leaf disease images contained only one leaf in the image (see Figure 16 and Figure 17) while the mulberry leaf images were taken from the natural environment with different perspectives, sunlight conditions, and several leaves appear in the image (Figure 14 Illustration of the mulberry field area in Thailand has been collected as a dataset in this study consisting of Maha Sarakham, Buriram, Nakhon Ratchasima, and Phitsanulok, and Chiang Mai.).

There is still a deficiency in improving the accuracy of the mulberry leaf dataset because the ensemble CNN method achieved only 94.75% accuracy. In future work, we plan to work on other data augmentation techniques such as generative adversarial networks (GAN) (Shorten & Khoshgoftaar, 2019), AutoAugment (Cubuk, Zoph, Mané, Vasudevan, & Le, 2019), and sample paring (Inoue, 2018) methods. Another direction for future work would be designing new ensemble CNNs. Bio-inspired algorithms, such as an artificial bee colony, bat algorithm, particle swarm optimization, and ant colony optimization will be employed to optimize the weight of the ensemble method (Darwish, 2018; Joel & Priya, 2018)

Chapter 4
Automated Model Selection using Evolutionary Ant Colony Optimization with Learning Rate Schedule to Recognize Plant Leaf Images

The model selection method is a necessary process proposed to discover robust models that enhance the performance of the recognition systems. In this research, a new ant colony optimization (ACO) is proposed to select the robust models of a convolutional neural network (CNN). Further, the robust models are performed in the ensemble learning method, called ensemble CNNs. The advantage of the evolutionary ACO algorithm is that it guarantees to select the set of robust CNN models every running time because the new fitness function and the learning rate schedule embedded in the ACO algorithm increases the distribution of the pheromones. When the new CNN models were added to the systems, the proposed ACO algorithm allowed an agent to find the new CNN model, while the original ACO algorithm always selected the same CNN model. For the evaluation, we assessed the proposed ACO algorithm on two plant leaf datasets: mulberry and Turkey-plant, and also compared the results with existing methods. In our experiments, we trained 15 CNN models with different tuning parameters. These CNN models were used in the automatic model selection based on the ACO algorithm. We first compared two ACO algorithms, including the ant colony system (ACS) and the max-min ant system (MMAS). The result showed that the MMAS algorithm outperformed the ACS algorithm. Hence, three ensemble learning methods (unweighted average, weighted average, and cost-sensitive learning) were evaluated and it was found that the weighted average method is the best ensemble method. Additionally, the weighted parameters were discovered by the grid-search method was executed when finding the weighted parameters. The proposed ACO algorithm achieved an accuracy above 99.33% and 95.34% on the Turkey-plant and mulberry leaf datasets, respectively.

4.1 Introduction

Recognition of plant species and diseases by humans requires experience, so only a small mistake could cause many problems. In this case, many expert people are required. On the other hand, due to computer technology advancements, many researchers have proposed plant recognition systems to detect and recognize plant diseases and classify plant species (DeChant et al., 2017; Hassan et al., 2021; X. Li & Chen, 2010). A plant recognition system could be invented to prevent the risk of using the wrong plant species in medicine and to stop the spread of diseases on the farms in the early phase (Dhaware & Wanjale, 2017; Sinha & Shekhawat, 2020). Consequently, taking advantage of the precision and speed of computer technology is very useful in creating highly efficient plant recognition systems. So, it could be performed automatically with fewer errors and reduced working time (Fathi Kazerouni, Mohammed Saeed, & Kuhnert, 2019; Hughes & Salathé, 2015).

Agriculturalists could recognize plant species and diseases by plant leaves. Hence, when creating the plant recognition systems, many researchers collected plant leaves and took images in laboratories with white backgrounds (Arafat, Saghir, Ishtiaq, & Bashir, 2016; Munisami et al., 2015; Pawara, Okafor, Surinta, et al., 2017).

Nevertheless, this may not be suitable for actual use. When leaf images are taken in the laboratory, image processing techniques could propose extracting shape, color, and texture from images, called a feature, and then recognize features using machine learning techniques (Patil et al., 2013; Wang et al., 2008). In comparison, many researchers took plant leaves in real-world environments with many conditions, such as complex backgrounds, light, shadow, and perspective, while taking the images (Chompookham & Surinta, 2021; Kusumo et al., 2018; Turkoglu et al., 2021; Vo et al., 2019). Under these circumstances it might be more challenging to increase the performance of the plant leaf recognition systems.

Deep learning techniques, especially convolutional neural networks (CNNs), are used when leaf images are taken in real-world environments (Atabay, 2016; A. Kumar & Vani, 2019; Turkoglu et al., 2021). The biggest advantage of the CNN method is that they combine the feature extraction method and recognition into one architecture. Then, extracting the shape and region of interest are not required. However, using only one CNN model does not guarantee the highest performance on plant leaf recognition. Importantly, to deal with the problems of accuracy performance, the ensemble CNNs method is proposed due to the power of multiple CNN models that could make a better recognition than using only one CNN model (Chompookham & Surinta, 2021; Enkvetchakul & Surinta, 2022; Mokeev, 2019; Puangsuwan & Surinta, 2021). The problem of the ensemble CNNs method is discovering the best combination between various CNN models and the best number of CNN models used in ensemble learning.

Contribution. In this research, we proposed the ant colony optimization (ACO) algorithm as the model selection method to automatically discover the best combination of CNN models. We aim to present a new model selection based on the ACO algorithm by adding two functions to the ACO algorithm, including two new fitness functions and two learning rate schedules (time-based and cyclical learning). These two functions allow the ACO algorithm to find the robust CNN models. Consequently, the robust CNN models are used in the ensemble learning method.

The original ACO algorithm computed the pheromones table based on the fitness function, which is highly possible to select the same CNN models, even if the system has new robust CNN models because the values of pheromones are not distributed. Therefore, the proposed ACO algorithm could distribute the values of the pheromones table and have a high chance of selecting new robust CNN models. To demonstrate the significant improvement of the new ACO algorithm, we evaluated the proposed algorithm on two plant leaf datasets: mulberry leaf and Turkey-plant and achieved high accuracy.

This paper has been organized as follows. Section 4.2 summarizes the overview of related work. Section 4.3 describes the proposed ant colony optimization algorithm for plant leaf image recognition. Two plant leaf datasets are described in Section 4.4. Evaluation metrics and experimental results are presented in Sections 4.5, 4.6., and 4.7 the discussion and conclusion are presented in Sections 4.8 and 4.9.

4.2 Related Work

Much recent research has been conducted to address problems of plant leaf recognition. This section describes the related work on recognizing plants, including the ACO, CNN, and ensemble learning methods.

4.2.1 Ant Colony Optimization (ACO) Algorithm

The ACO algorithm was proposed in many applications (Fahmi, Zarlis, Nababan, & Sihombing, 2020; S. Li, Wei, Liu, Zhu, & Yu, 2022; Pitakaso, Almeder, Doerner, & Hartl, 2007). However, only a few studies proposed the ACO algorithm in the plant recognition domain. Li and Chen (2010) and Ghasab et al. (2015) used ACO as the feature selection method. In 2010, shape features of the weed leaf were extracted (X. Li & Chen, 2010). The ACO algorithm was used to search for the best feature subset. Hence, the SVM algorithm was employed to create the model from the best feature subset selected using the ACO algorithm. For the feature selection methods, they compared the ACO algorithm with the genetic algorithm (GA) in terms of accuracy and number of features. The result showed that the ACO algorithm outperformed the GA algorithm in accuracy and number of features.

Further, Ghasab et al. (2015) presented an expert system for automatically recognizing different plant species from leaf images. In their study, firstly, the shape, morphology, color, and texture of the plant leaves were extracted as possible features. Secondly, the ACO algorithm was applied as the feature decision-making method to select the best features. Lastly, the selected features were then transferred to a support vector machine (SVM) to create a robust model and classify plant species. When evaluating their proposed method, around 2,050 images selected from FCA and Flavia datasets were tested and the results achieved an accuracy of 95.53%.

4.2.2 Convolutional Neural Networks (CNNs)

Due to the success of the deep learning technique, numerous researchers mainly use CNNs to address their problems. Many new architectures, including ResNet, NASNet, DenseNet, ResNext, EfficientNet, etc. (Huang et al., 2018; Tan & Le, 2019; Xie, Girshick, Dollár, Tu, & He, 2016; Zoph et al., 2018), have been proposed. Further, CNNs are proposed in agriculture (Adhitya, Prakosa, Köppen, & Leu, 2019; Neforawati, Herman, & Mohd, 2019). Pawara et al. (2017) compared the performance of the well-known CNN architectures (AlexNet and GoogLeNet) with two feature extraction methods; a histogram of oriented gradients (HOG) and a bag of visual words with HOG (BOW-HOG) on three plant datasets, including AgrilPlant, LeafSnap, and Folio. Further, two feature extraction methods were trained by machine learning techniques; K-nearest neighbor (KNN), support vector machine (SVM), and multilayer perceptron (MLP). The experimental results showed that both CNN architectures outperformed the hand-crafted features methods and achieved an accuracy above 97% on three datasets. Jeon and Rhee (2017) used GoogLeNet to improve the performance of plant leaf recognition. GoogLeNet was trained on the distortion and discoloration of leaf images, called damaged leaf images. The experimental results showed that above 94% accuracy was obtained when trained on the damaged leaf images with damage of around 30%. Bisen (2021) proposed a new CNN architecture that contains 11 layers to

recognize 15 plant species via leaf images in the Swedish dataset. Subsequently, data augmentation techniques were used while training the proposed CNN, including rotation, width shift, height shirt, zoom, rescale, and resize. Their proposed CNN achieved an accuracy of 97% and outperformed the existing method (Turkoglu et al., 2021).

Hassan et al. (2021) used many deep CNN models (InceptionV3, InceptionResNetV2, MobileNetV2, and EfficientNetB0) to identify and diagnose plant leaf diseases on the PlantVillage dataset. This dataset has 14 plant species and contains 54,305 images of 38 classes (healthy and diseased). While training the CNN models, the dataset was divided into training and test sets in their experiments with different ratios (80:20, 70:30, and 60:40). The results showed that the CNN models had an accuracy rate above 90% when trained with few epochs (30-50). Subsequently, The EfficientNetB0 slightly outperformed three other CNN models: InceptionV3, InceptionResNetV2, and MobileNetV2.

Moreover, Quach et al. (2022) used the CNN model as the feature extraction method. Their method extracted the robust features from the leaf images by many feature extraction methods; shape, texture, color, Fourier descriptor, vertical and horizontal projection, and vein. Hence, the CNN model computed these robust features. The output was designated with 100 features and fed to the SVM classifier. Their method achieved an accuracy of 99.58% on the test set of the Flavia leaf dataset.

4.2.3 Ensemble Learning Methods

Recognizing plant leaf images using a single CNN model does not always guarantee good results. The ensemble learning method is proposed to improve the performance of the single CNN model by combining output of various CNN models and classifying using ensemble learning (Chompookham & Surinta, 2021; Enkvetchakul & Surinta, 2022; Mokeev, 2019) .

Peker (2021) proposed a multi-channel capsule network ensemble (MCCNE). The multi-channel included five channels: R-channel, G-channel, B-channel, Gabor filter, and principal component analysis (PCA). As a result, the input images were computed by five different techniques. Hence, the features computed from each feature extraction technique were sent to the capsule network to create the model. Further, the output of each capsule network was classified using the majority voting method. Their proposed method was evaluated on a tomato dataset containing nine disease classes and one healthy class. Consequently, when evaluating the single model, which is the Gabor filter combined with a capsule network, it achieved an accuracy rate of 96.15%. However, the result showed that the MCCNE method, which is the ensemble learning method, achieved an accuracy of 98.15% on the tomato leaf disease dataset and outperformed the existing methods.

Turkoglu et al. (2021) trained the deep learning models of six state-of-the-art CNN models (AlexNet, GoogLeNet, ResNet18, ResNet50, ResNet101, and DenseNet201) on the Turkey-plant dataset, which contain 15 diseases of 4,447 images. For the experiments, first, the experimental results of the CNN models showed that the AlexNet and GoogLeNet achieved accuracy less than 90%, while other CNN models achieved an accuracy above 90%. Therefore, DenseNet201 was the best single CNN model evaluated on the Turkey-plant dataset. Second, they experimented with using CNN models to extract spatial features and concatenate them, followed by the SVM

algorithm. This method was called PlantDiseaseNet-EF. Third, the method of extracting features using the CNN model and classifying using SVM was proposed. Hence, the prediction outputs were then classified using a majority vote. This method is called PlantDiseaseNet-MV. Consequently, the experimental results of PlantDiseaseNet-EF and PlantDiseaseNet-MV performed much better than the single CNN model. Moreover, the PlantDiseaseNet-MV method achieved an accuracy of 97.56% and slightly outperformed the PlantDiseaseNet-EF method.

To enhance the performance of plant leaf recognition, Chompookham and Surinta (2021) proposed ensemble CNNs to recognize plant leaf images on three benchmark datasets: mulberry leaf, tomato, and corn leaf diseases. First, five CNN models; MobileNetV1, MobileNetV2, NASNetMobile, DenseNet121, and Xception, and various data augmentation techniques were used as the single CNN model to recognize leaf image datasets. However, the single CNN model achieved only approximately 90% accuracy on the mulberry leaf dataset. On the other hand, the CNN model achieved above 99% accuracy on the tomato and corn leaf disease datasets. Furthermore, the ensemble CNNs with combined output probabilities of 3 (called 3-EnsCNNs) and 5 (called 5-EnsCNNs) robust CNN models and classified using three ensemble learning methods: unweighted majority vote, unweighted average, and weighted average. As a result, the 5-EnsCNNs with the weighted average ensemble method achieved an accuracy of 94.75% on the mulberry leaf dataset, which is improved by around 4.75%.

Additionally, Prem and Surinta (2022) trained four lightweight CNN architectures (EfficientNetB1, InceptionResNetV2, MobileNetV2, and NASNetMobile) with data augmentation techniques, then sent the output probability of each CNN model to classify using the ensemble learning methods: unweighted majority vote and unweighted average. The plant leaf images were randomly selected and trained by the CNN model. Their method was different from the other research in that it combined the output from the same CNN model, while other research combined the output from various CNN models. As a result, the ensemble CNNs outperformed the single CNN model on all plant leaf datasets. Consequently, the EfficientNetB1 is the best CNN architecture when combined with the ensemble CNNs.

4.3 The Proposed Ant Colony Optimization for Automated Model Selection

4.3.1 Overview of the Ensemble CNNs Framework

In this section, we present the ensemble CNNs framework for plant leaf recognition, as shown in Figure 18. The details of the ensemble CNNs are described as follows.

A. The Training Scheme. First, many pre-trained models of the state-of-the-art CNN architectures were trained and fine-tuned to create the robust CNN model on the plant leaf dataset. Second, the ACO algorithm, which is the metaheuristic optimization algorithm, was proposed as the model selection method to select the robust CNN models from all the CNN models. For example, as shown in Figure 18, CNN models 3, 1, and 5 were automatically selected by the ACO algorithm. Third, we combined the particular CNN models. Also, the output probabilities of each model were used to find the weighted parameters by the grid-search method. Finally, the selected CNN models and the weighted parameters were transferred to the test scheme.

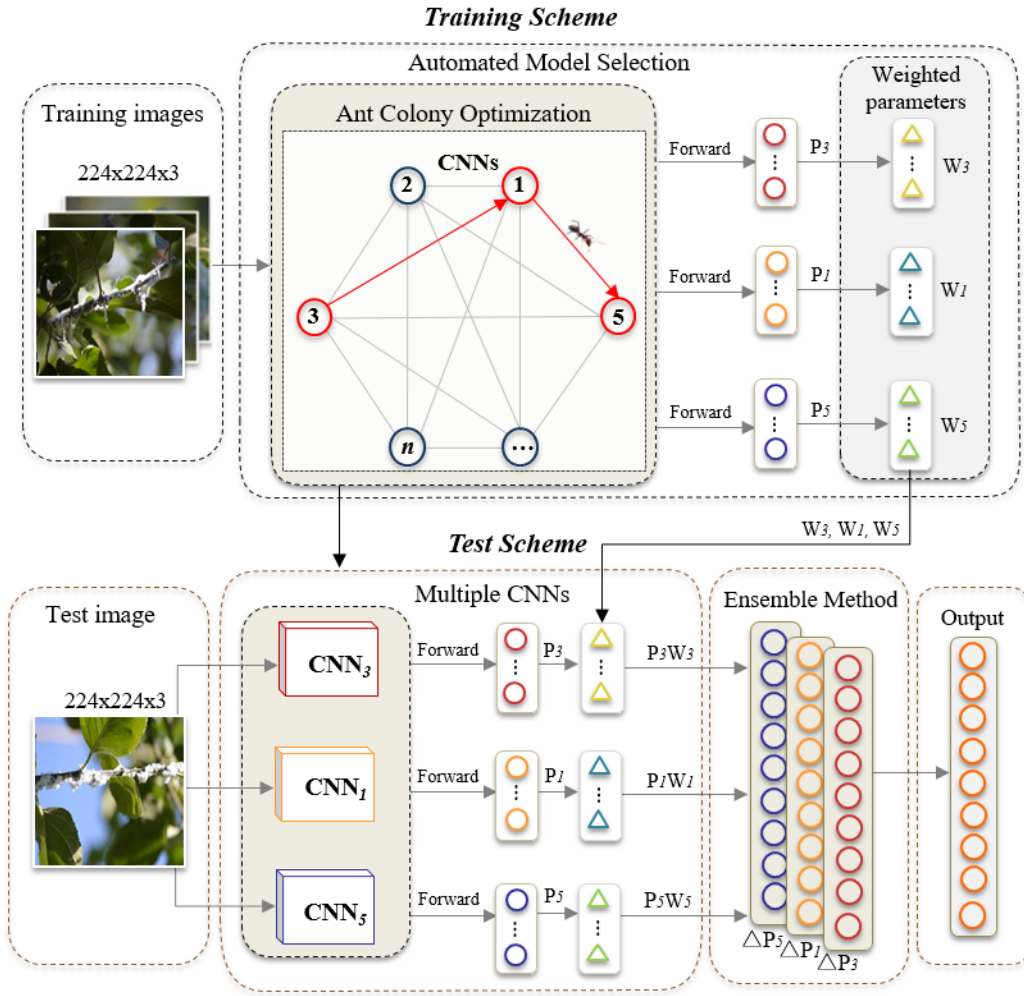


Figure 18 *Illustration of the ensemble CNNs based on the automatic model selection by the proposed ACO algorithm.*

B. The Test Scheme. The test image was directly sent to CNN models, which were selected in the training scheme, to compute the output probabilities. Hence, the output probabilities of each CNN model were computed with the weighted parameters to obtain the final recognition, called the ensemble method.

4.3.2 Ant Colony Optimization (ACO)

The ACO algorithm is the metaheuristic method employed to solve complex optimization problems with the best solutions (S. Li et al., 2022; Pettersson & Lundell Johansson, 2018). The design of the ACO algorithm was inspired by ant behavior during a foraging cycle for food (M. Dorigo, 1992). In the foraging cycle, ants spread pheromones that trail to a food source. Hence, other ants could follow the pheromones until they found the food and carry it back to the nest. Thereby, the shortest route has more pheromones, so ants could easily track the shortest route without returning to their nest. The most successful application that applied the ACO algorithm was in the traveling salesman problem, which used the ACO algorithm to find the shortest route such that a salesman visits each city only once until returning to the origin city

(Dewantoro, Sihombing, & Sutarman, 2019). The ACO algorithm has been applied in many domains, such as path planning for mobile robots (Chen et al., 2021), optimization routes in wireless sensor networks (Sharmin, Anwar, & Motakabber, 2018), and searching the penalty parameter in the kernel function of the support vector machine (SVM) algorithm (Yan, Enhua, & Shuangting, 2020).

Algorithm 1: The proposed ACO Algorithm for model selection

Input:
number of iterations: N
number of ants: M
number of CNN model: R
pheromone: τ
fitness function: f
weight parameters: alpha (α), beta (β), evaporation rate (p), learning rate value (η)

Initialization:
$f_{i,j}$ define by fitness function
if TL model using Equation (8)
if TLEW model using Equation (9)

Process:
For i =1 to N **do**
For j =1 to M **do**
Route construction by ant system
if ACS algorithm using Equation (11)
if MMAS algorithm using Equation (13)
Create ensemble CNNs
For k =1 to R **do**
if unweighted average method using Equation (17)
if weighted average method using Equation (18)
if cost-sensitive learning method using Equation (20)
End for
End for
Calculate η by learning rate schedule
if time-based scheduler using Equation (15)
if cyclical learning rate scheduler using Equation (16)
Update the pheromone by ant system
if ACS algorithm using Equations (12)
if MMAS algorithm using Equations (12) and (14)
Save best ACO solution ($T_{(i,j)}^{best}$)
End for

Output:
The best ACO solution ($T_{(i,j)}^{best}$)

In this research, the ACO algorithm is proposed as the model selection method to select robust CNN models by considering the loss and error values from various CNN models. First, instead of the fitness value between pairs of the cities, the validation loss and error rate of each CNN model was used to create the fitness function. Second, the learning rate schedule was embedded in the ACO algorithm. These two proposed methods enabled the ACO algorithm to compute and adjust the pheromones. The pheromones will distribute and increase the chance of selecting the robust CNN models. So, the set of the robust CNN models has been used in the ensemble CNNs method. The process of the proposed ACO algorithm is shown in Algorithm 1.

4.3.2.1 Fitness Functions

The fitness function represents the fitness value between the first station (i) and the second station (j). In this study, the training values of the CNN models between two robust CNN models are employed to compute the fitness model ($f(i,j)$), including training loss values, training error, and constant weight parameters. The minimum fitness value of all selected CNN models reflects the most attractive path. We proposed two fitness functions that use while constructing the robust route of CNN models. Two fitness functions are presented as follows.

1) Training Loss Model (TL)

The most uncomplicated fitness function is computed using the training loss value of each CNN model. The training loss model is computed by Equation 8.

$$f(i,j) = \left|l_{\mathrm{i}} - l_{\mathrm{j}}\right|, \tag{8}$$

where l_{i} is training loss value of CNN model i and l_{j} is training loss value of CNN model j, $i = 1, 2, \dots, m,\ j = 1, 2, \dots, n,$ and m, n are the numbers of CNN models.

2) Training Loss and Error with Weight Parameter Model (TLEW)

We computed the fitness function with training loss and training error values of the CNN model. In addition, the optimal weight parameter is added to the fitness function to control the contributing CNN models. The TLEW model is computed by Equation 9.

$$f(i,j) = \sqrt{((l_i - e_i) + w_i)^2 + ((l_{\mathrm{j}} - e_j) + w_j)^2}, \tag{9}$$

where l_i and e_i is training loss and training error values of CNN model i and w_i is the optimal weight parameter, and w_i is defined as follows.

$$w_i = \begin{cases} 1 & if \ \ x \geq 93 \\ 2 & if \ 92 \leq x < 93 \\ 3 & if \ 91 \leq x < 92 \\ 4 & if \ 90 \leq x < 91 \\ n & otherwise \ldots < x < \ 90, \end{cases}$$

where x is the accuracy of each CNN model and n is the maximum weight parameter.

Consequently, the fitness table of the whole station is constructed as shown in Equation 10.

$$\begin{matrix} \\ j \end{matrix} \begin{matrix} \begin{matrix} \qquad\qquad i \end{matrix} \\ \begin{bmatrix} 0 & d(i_1, j_2) & d(i_1, j_{n-1}) & d(i_1, j_n) \\ d(i_2, j_1) & 0 & d(i_2, j_{n-1}) & d(i_2, j_n) \\ d(i_{m-1}, j_1) & d(i_{m-1}, j_2) & 0 & d(i_{m-1}, j_n) \\ d(i_m, j_1) & d(i_m, j_2) & d(i_m, j_{n-1}) & 0 \end{bmatrix} \end{matrix} \tag{10}$$

where $f(i_1, j_2)$ is the fitness value between CNN model 1 and CNN model n. The fitness values between the same station $f(i_1, j_1)$ is zero.

4.3.2.2 Route Construction and Pheromone Updating Rule

In this section, the ants must decide which paths to walk along to complete the solution, which is the shortest route, called route construction. Further, while constructing the shortest route, the ant evaluates the solution, modifies the trial values, and updates the best solution in the pheromone table, which other ants will use to find another route in the future. In this research, we used two systems to search for the best solution; the ant colony system (ACS) and the max-min ant system (MMAS), which are presented as follows.

A. Ant colony system (ACS)

The ACS algorithm was proposed by Dorigo, Di Caro, and Gambardella (1997) as a meta-heuristic algorithm for optimization problems such as the traveling salesman problem. The ants search for food and walk from one station to another until they get food. The pheromones ($\tau_{(i,j)}$) are spread to the route they walk, called the route construction process (p_{ij}^k). Then, other ants can follow that route to bring food. Further, other ants can decide to walk on a different route to find a better solution. The ant will spread pheromones when the new route is the better solution, called the pheromone updating rule. The ACS algorithm uses the pseudorandom proportional rule (Marco Dorigo & Gambardella, 1997) to construct the best route by Equation 11 and update the pheromone in each iteration when the ant walks through that route, as computed by Equation 12.

$$p_{(i,j)}^{k} = \begin{cases} argmax_{l \in Y_i^k} \left(\tau_{(i.j)}^{\alpha} \times F_{(i,j)}^{\beta}\right) & if\ (q \leq q_0), \\ \frac{\left(\tau_{(i.j)}^{\alpha} \times F_{(i,j)}^{\beta}\right)}{\sum_{l \in Y_i^k}\left(\tau_{(i.j)}^{\alpha} \times F_{(i,j)}^{\beta}\right)}, when\ j \in Y_i^k & otherwise(q > q_0), \end{cases} \quad (11)$$

where $F_{(i,j)} = \frac{1}{f_{(i,j)}}$, $f_{(i,j)}$ is a fitness function, q is a random value between zero to one, q_0 is the probability value that the ant constructs the best possible move, α and β defined as the relative influence between the heuristic information and the pheromone levels, α and β are equal or greater than one, and Y_i^k is the stations that have not yet traveled when moving from station i to station j.

In the ACS algorithm, however, only one ant allows updated the pheromones, as a global update, in each iteration, as computed by Equation 12.

$$\tau_{(i,j)}^{n} = \begin{cases} (1-p)\left(\tau_{(i,j)}^{n-1}\right) & if\ (i,j) \text{ is not travel path}, \\ (1-p)\left(\tau_{(i,j)}^{n-1}\right) + \eta \bigtriangledown T_{(i,j)}^{best} & if\ (i,j) \text{ is travel path}, \end{cases} \quad (12)$$

where p is the evaporation rate of the pheromone and $p \in (0,1)$, n is number of iterations, $\bigtriangledown T_{(i,j)}^{best}$ is $\frac{1}{L_{best}}$, L_{best} is the total fitness that provides the best solution for each iteration, and η is the learning rate.

B. Max-Min Ant System (MMAS)

The MMAS algorithm was proposed by Stutzle and Hoos (2000). In the MMAS algorithm, the random proportional rule was suggested to construct the best route by randomizing the probability value when ant (k) at the station i walk to station j. In this process, the ant must check whether the route is traveled or not yet in each iteration. However, when an ant walks through that particular station, that station will be removed from the memory. As a result, that ant can walk through that station only once. Further, only the best ant can improve the pheromones. The MMAS algorithm computed the route construction by Equation 13.

$$p_{(i,j)}^{k} = \frac{\left(\tau_{(i.j)}^{\alpha} \times F_{(i,j)}^{\beta}\right)}{\sum_{l \in Y_i^k}\left(\tau_{(i.j)}^{\alpha} \times F_{(i,j)}^{\beta}\right)} \text{ when } j \in Y_i^k, \quad (13)$$

The MMAS algorithm updates the pheromones according to the maximum and minimum values. This research sets the maximum and minimum values as τmin=0.1 and τmax=0.95. Hence, $\tau_{(i,j)}^{n}$ is the pheromone values between τmax and τmin ($\tau min \leq \tau_{(i,j)}^{n} \leq \tau max$). The MMAS algorithm increases the chance of selecting a route that was never selected before, which is calculated by Equation 14.

$$\tau_{(i,j)}^{n} = \begin{cases} \tau min & \text{if } \tau_{(i,j)}^{n} < \tau min, \\ \tau max & \text{if } \tau_{(i,j)}^{n} > \tau max, \end{cases} \quad (14)$$

In this research, the learning rate (η) used in Equations 12 and 14 is further computed by Equations 15 and 16 when selecting time-based and cyclical Learning rate schedules, respectively.

4.2.3.3 Learning Rate Schedule

Deep learning techniques have the intent of discovering the optimal parameters when training the deep learning models iteratively to minimize a given function to the local minimum (P. Li, 2017; Sun, Cao, Zhu, & Zhao, 2019). Many techniques are proposed to find the local minimum, such as using different optimization algorithms (i.e., SGD, Adam, Adagrad, and adaDelta), tuning the hyperparameters (momentum, decay, and learning rate), and applying various learning rate schedules, which can increase the performance and decrease the training time of the deep learning techniques. However, to optimize the parameters, the gradient is computed with a learning rate value that could change at every training iteration using the learning rate schedule method (J. Park, Yi, & Ji, 2020).

In this section, we developed the ACO algorithm by adding the learning rate schedule to the algorithm to change the learning rate value while training the ACO algorithm, with the objective function of decreasing the fitness values between each CNN model and increasing the chance of distributing the pheromones. We briefly describe two learning rate schedules: the time-based learning rate and cyclical learning rate (CLR), that are used in the experiments, as follows.

A. Time-based Learning Rate Schedule

The uncomplicated learning rate schedule is the time-based learning rate schedule (J. Park et al., 2020). The time-based scheduler yields the learning rate value to drop quickly at the start of the training scheme. The demonstration of the learning rate values of the time-based scheduler is shown in Figure 19 a). The time-based scheduler is computed by Equation 15.

$$\eta_{n+1} = \frac{\eta_n}{1+(d*n)} \quad (15)$$

where η_n is the learning rate at iteration n , n is the number of iterations, d is the decay value, and avoid 0 in the denominator by adding 1.

B. Cyclical Learning Rate (CLR)

The CLR schedule was proposed by Smith (2017) to adjust the learning rate value linearly for a few iterations. In the CLR scheduler, the maximum and minimum learning rates are defined, then the learning rate is linearly increased to the maximum and linearly decreased to the minimum values for a few iterations. The change in the learning rate looks like a triangle shape, so it is called the triangular

learning rate policy. The demonstration of the learning rate values of the CLR scheduler is shown in Figure 19 b). The algorithm of the CLR scheduler is computed by Equation 16.

$$LocalCycle = math.floor(1 + epoch/(2 * stepsize)) \quad (16)$$
$$LocalX = math.abs(1 + epoch\ /\ stepsize - 2 * cycle + 1)$$
$$LocalLR = minLR + (maxLR - minLR) * math.max(0, (1 - x))$$

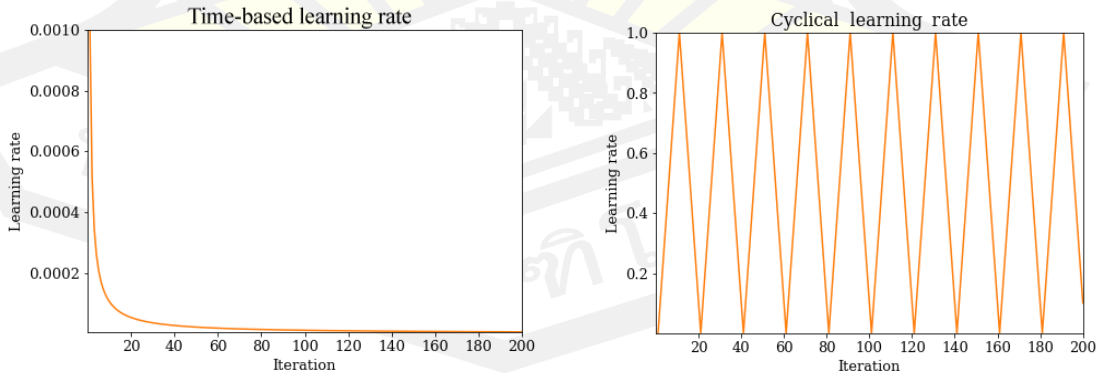
where $epoch$ is the number of iterations when training, $stepsize$ is the number of iterations in half a cycle, $minLR$ is the minimum learning rate, $maxLR$ is the maximum learning rate, and x is training data.

4.3.5 Ensemble CNNs

The ensemble method was proposed to enhance the recognition performance by combining the output of various machine learning models to assemble the final optimal recognition model (Alabbas, Khudeyer, & Jaf, 2016; Chompookham & Surinta, 2021). Consequently, the ensemble method guarantees better performance (Dietterich, 2000; Ganaie et al., 2021).

In this research, the machine learning models employed in the ensemble method were pre-trained state-of-the-art CNN models, comprising MobileNetV1, MobileNetV2, DenseNet121, NASNetMobile, and Xception (Chollet, 2016; Howard et al., 2017a; Huang et al., 2018; Mark Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018; Zoph et al., 2018). We then trained CNN models with the following fine-tuned hyperparameters; optimization algorithms, data augmentation techniques, and learning rate (H. Li et al., 2020; Poojary, Raina, & Mondal, 2020). In this step, we collected diverse CNN models already used in the model selection by the ACO algorithm. Further, the output probabilities of the CNN models selected by the ACO algorithm were performed in the ensemble method.

This section briefly presents the ensemble methods; unweighted average, weighted average, and cost-sensitive probability, that were performed in the experiments.



a) *b)*

Figure 19 *Illustration of the learning rate values when using different learning rate schedules: a) time-based learning rate and b) cyclical learning rate.*

A. Unweighted Average Method

In this research, the output probabilities of CNN models were used in the ensemble methods. Further, the output probabilities were computed by the softmax function. For the unweighted average method, all the models have the same priority (Ju et al., 2018). Assigning a high weight value to any model is not necessarily. Furthermore, the outputs are averaged and the highest probability ($arg\,max(p')$) is selected as a final recognition (J. Li et al., 2021). The unweighted average method is calculated by Equation 17.

$$arg\,max(p') = \frac{1}{n}\sum_{i=1}^{n}(\vec{y}), \tag{17}$$

where $\vec{y}$ is the weight vector and n is the number of CNN models.

B. Weighted Average Method

For the weighted average method, assigning weight values to the models is required. Weights were proposed to compute with the output probabilities of the CNN models (Florea & Andonie, 2019). The highest weight was given to the most achieved CNN model, although the lower weight was assigned to the other CNN models (Harangi, 2018). For the final recognition, the outputs were averaged and then the argmax function was proposed to select the final output. The weighted average method is calculated by Equation 18.

$$arg\,max(p') = \frac{1}{n}\sum_{i=1}^{n}(\vec{y}\alpha_i), \tag{18}$$

where n is the number of CNN models and α_i is the weight values that compute with the weight vector ($\vec{y}$).

In this research, a grid-search method was used to discover the optimal weight parameters for each CNN model. The set of weight parameters was passed to the softmax activation function to scale the weight before computing the weights with the output probabilities. Consequently, the summation of all weights is equal to one (Nwankpa, Ijomah, Gachagan, & Marshall, 2018). The softmax activation function was computed by Equation 19.

$$\sigma(\vec{Z}) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_i}}, \tag{19}$$

where $\vec{z}$ is an input weight vector, z_i is the elements of $\vec{z}$, e^{z_i} is the exponential function for z_i, e^{z_j} is the exponential function for z_j, where $j = 1, .., K$, and K is the number of weight vectors.

C. Cost-sensitive Learning Method

Rojarath and Songpan (2021) proposed a cost-sensitive learning method that is designed for weighted voting in the ensemble learning method. In the cost-sensitive

learning method, the true positive (TP_i) rate (is called sensitivity) and output probabilities ($Prob_i$) of class i were multiplied and used as the weight parameter of class i. In our research, the output probabilities were provided by the CNN models. Hence, the weight parameters were computed depending on the number of models (n). The weight parameters of class i were then averaged and proposed as a new weight of class i. The new weight of each class is computed by Equation 20.

$$NewWeight_i = \frac{1}{n}\sum_{j=1}^{n}(TP_{i,j} * Prob_{i,j}), \quad (20)$$

where $TP_{i,j}$ is true positive of class i and model j, $Prob_{i,j}$ is the output probabilities of class i and model j, when N is the number of models, and TP_i is computed by Equation 21.

$$TP_i = \frac{Predicted\ class_i}{Actual\ class_i} \quad (21)$$

4.4 Plant Leaf Datasets

Two plant leaf image datasets used in the experiment were taken from natural environments.

4.4.1 The Mulberry Leaf Dataset

Chompookham and Surinta (2021) collected the mulberry leaf dataset using smartphones and DSLR cameras. This dataset has ten cultivars and contains 5,262 mulberry leaf images taken from natural environments in 5 provinces of Thailand; Maha Sarakha, Phitsanulok, Chiang Mai, Buriram, and Nakhon Ratchasima. Further, ten mulberry cultivars a) Black Austurkey, b) Black Australia, c) Taiwan Maechor, d) Taiwan Strawberry, e) King Red, f) King White, g) Kamphaeng Saen 42, h) Chiang Mai 60, i) Buriram 60, and j) mixed breed mulberry Chiang Mai 60 + Buriram 60, as shown in Figure 20.

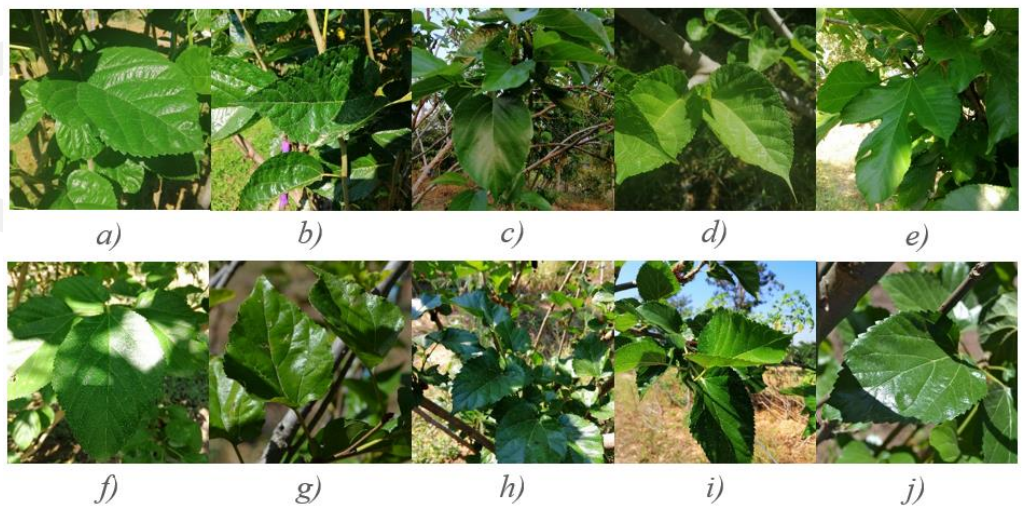


Figure 20 *Examples of the mulberry leaf dataset.*

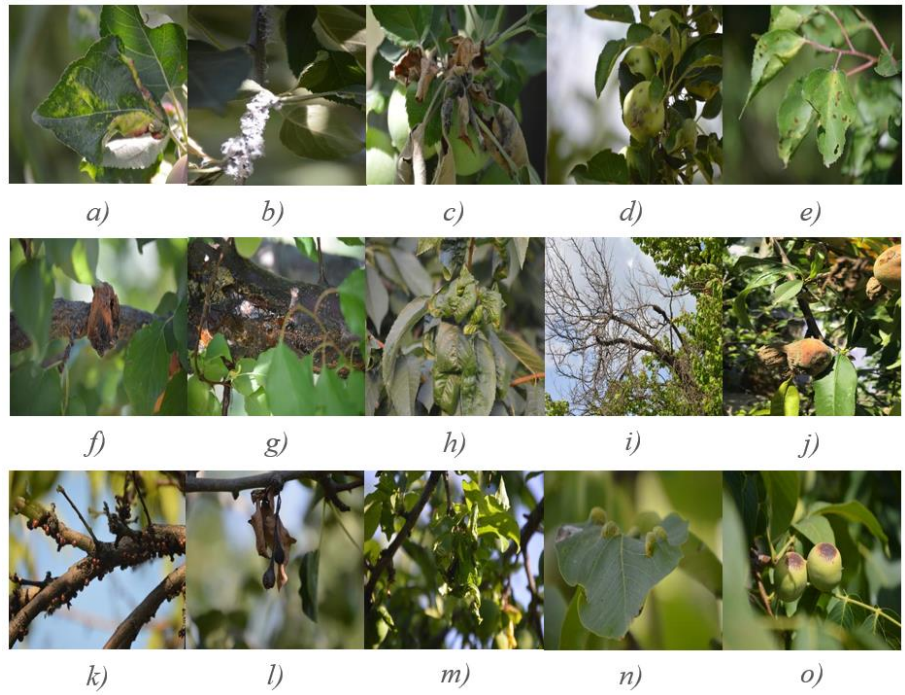


Figure 21 *Examples of the turkey-plant disease image dataset.*

4.4.2 The Turkey-plant Dataset

Turkoglu et al. (2021) collected the Turkey-plant disease image dataset in 2021, which is of common diseases and pests found in Turkey. The challenge of this dataset is that the dataset contains unconstrained images, including different perspectives and different parts of plants. The Turkey-plant disease images were taken from natural environments using a Nikon 7200D camera with resolution of 4000x6000 pixels and stored in the RGB channel. This dataset consists of 15 categories and contains 4,447 plant disease images, including a) Apple Aphis Spp, b) Apple Eriosoma Lanigerum, c) Apple Monillia Laxa, d) Apple Venturia Inaequalis, e) Apricot Coryneum Beijerinckii, f) Apricot Monillia Laxa, g) Fruit Trees Cancer Symptom, h) Cherry Aphis Spp, i) Fruit Trees Drying Symptom, j) Peach Monillia Laxa, k) Peach Parthenolecanium Corni, l) Pear Erwinia Amylovora, m) Plum Aphis Spp, n) Walnut Eriophyes Erineus, and o) Walnut Gnomonia Leptostyla, as shown in Figure 21.

4.5 Performance Evaluation

We used six evaluation metrics to measure the performance of the proposed method (Hicks et al., 2022; Hossin & M.N, 2015), as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{22}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{23}$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} \tag{24}$$

$$\text{Specificity} = \frac{TN}{TN+FP} \tag{25}$$

$$\text{F1} - \text{score} = \frac{2TP}{2TP+FP+FN} \tag{26}$$

where TP, TN, FP, FN are true positive, true negative, false positive, and false negative, respectively.

We also present the graph of the receiver operating characteristic (ROC) curve (Hajian-Tilaki, 2013; R. Kumar & Indrayan, 2011) that is used to comprehensively measure each performance index. In the ROC, the vertical axis is the true positive rate (TPR), which is equivalent to sensitivity, and the horizontal axis is the false positive rate (FPR), defined in the following:

$$\text{FPR} = 1 - \text{Specificity} \tag{27}$$

We also present the area under the ROC curve (AUC) value to the separability measurement that the proposed model can determine between classes. Thus, the AUC value close to one perfectly classifies all the positive and negative classes.

4.6 Experimental Results

4.6.1 Implementation Detail

For the implementation detail, we employed ant colony optimization (ACO) to automate select CNN models. Then, the selected CNN models were sent to classify using ensemble CNN models. The TensorFlow deep learning framework running on Google Colaboratory was used for all experiments. In these experiments, we evaluated the proposed method on two datasets, mulberry leaf and Turkey-plant. The mulberry leaf dataset was divided into training, validation, and test sets with a ratio of 70:10:20. We divided the Turkey-plant dataset with a ratio of 80:10:10 for training, validation, and test sets, respectively. The setups of each algorithm are described as follows.

CNN. According to the ACO algorithms that we employed to choose the most appropriate CNN models, we planned to create many different CNN models based on pre-trained MobileNetV1 (Howard et al., 2017a), MobileNetV2 (M Sandler et al., 2018), DenseNet121 (Huang et al., 2018), NASNetMobile (Zoph et al., 2018), and Xception (Chollet, 2016) models. Furthermore, we trained CNN models with the following hyperparameters: optimizers (SGD and RMSProp) (G. Hinton, Srivastava, & Swersky, 2012; Ruder, 2017), data augmentation techniques (height shift, vertical flip, and fill mode) (Perez & Wang, 2017; Shorten & Khoshgoftaar, 2019), learning rate (0.1, 0.01, and 0.001). As a result, 15 CNN models were obtained and used in the ACO process.

ACO. In the first stage, two ACO algorithms were proposed to evaluate the performance of the ACO framework, including the ACS and MMAS. These two ACO

algorithms were used to optimize the pheromones, which is the main objective of ACO algorithms. In addition, we also added two learning rate schedules to the ACO algorithms, including time-based and cyclical learning. Indeed, we evaluated the learning rate schedules with various learning rates values between 0.1 and 0.0001. In the second stage, we proposed two new fitness functions as follows. The fitness function computed from 1) the training loss is called FF_{tl} and 2) the loss, error, and weighted accuracy is called FF_{tlew}. Hence, we employed a grid-search technique (Zöller & Huber, 2019) to optimize the basic ACO parameters, including alpha (in the range between 1 and 5), beta (in the range between 1 and 9), evaporation rate (between 0 and 1), number of ants (in the range from 20 to 100), and number of iterations (200 iterations).

Ensemble Learning. For ensemble learning, three ensemble learning methods were compared, including unweighted average, weighted average, and cost sensitive. In these processes, we also employed the grid-search algorithm to find the optimal weighted parameters for ensemble learning.

4.6.2 Recognition Performance on the Mulberry Leaf Dataset

4.6.2.1 Assessment of Ant Adaptation in ACO Algorithm

In this section, we experimented with the number of ants that correlated with accuracy performance and response time obtained from finding the best route with ACS and MMAS algorithms. We determined the number of ants per route as $10, 20, 30, ...,$ and 70. The maximum iteration was assigned to 200 iterations. In this experiment, the ACO parameters with the following values; $\alpha = 1$, $\beta = 1$, and $p = 0.95$, were defined. Consequently, the ACO algorithm was allowed to find the best routs, which are the appropriate CNN models. Afterward, we sent the output of the CNNs to classify using the unweighted ensemble learning method. The results are presented in Table 8.

As shown in Table 8, it was found that increasing the number of ants also increases response time while finding the best route. However, many ants are not guaranteed to find the optimal route and high accuracy. The ACS algorithm obtained the highest accuracy of 95.06% when using 50 ants, while the MMAS algorithm achieved the highest accuracy of 95.09% when using only 20 ants. Moreover, we compared the results of two algorithms (MMAS and ACS) using the paired t-test method and found that the MMAS algorithm is significant compared to the ACS algorithm ($p < .05$).

4.6.2.2 Effect of Adding Learning Rate Schedule to ACO

This section first focused on adjusting ACO parameters, including the fitness function, ACO algorithm, beta, and decay. Second, we experimented with adding the learning rate schedules (Time-based and cyclical) into the ACO algorithms. For the fitness function, we proposed two models consisting of FF_{tl} and FF_{tlew} to present the pheromone distribution after ants had walked along the possible paths. The best number of ants in each ACO algorithm was selected from the previous experiment, as shown in Section 4.6.2.1. Hence, we used the grid-search method to search the optimal beta and decay parameters. For the learning rate schedule, the learning rate value

between 0.1-0.0001 was selected. Finally, after the ants chose the most acceptable CNN models, we used ensemble learning with the unweighted average method to evaluate the proposed method. The experimental results are presented in Table 9.

The experimental results from Table 9 show that optimizing the ACO parameters slightly affects the accuracy performance between 0.05-0.07%. We also evaluated the performance of the ACO algorithms (MMAS and ACS) when applying two different fitness functions (FF_{tl} and FF_{tlew}). First, we discovered that the FF_{tlew} model slightly obtained better accuracy when combined with MMAS and ACS algorithms than the FF_{tl} model with statistically significance at $p < .01$ (t-value = -7.25379 and p-value = 0.000014). Second, we observed two learning rate schedules: time-based and cyclical, that applied to the ACO algorithms. In comparison, the ACS algorithm when combined with cyclical learning rate schedules achieved an accuracy of 95.17%. The results showed that the cyclical learning rate schedules always achieved better accuracy than the Time-based learning rate schedules.

Table 8 *Evaluation performances (average accuracy and standard deviation) of the ACO algorithms.*

ACO	No. of Ants	Accuracy	Response Time (s.)
ACS	10	94.88±0.092	9.32
	20	94.91±0.098	14.5
	30	94.91±0.074	15.6
	40	95.02±0.096	16.4
	50	**95.06±0.071**	20.6
	60	94.91±0.232	22.7
	70	94.88±0.137	27.6
MMAS	10	95.02±0.071	6.71
	20	**95.09±0.085**	13.1
	30	95.01±0.000	14.9
	40	95.01±0.000	20.4
	50	95.04±0.058	28.3
	60	95.00±0.029	32.6
	70	95.04±0.035	36.4

Table 9 *Accuracy performance (%) of the ACO algorithms on the mulberry leaf dataset when applying learning rate schedule and training on different fitness functions.*

Fitness Function	ACO	Beta	Decay	Learning Rate Schedule	Learning Rate	Accuracy (%)
FF_{tl}	ACS	1	0.95	None	-	95.06±0.071
				Time-based	0.1	95.04±0.058
				Cyclical	0.001	95.02±0.029
	MMAS	1	0.95	None	-	95.09±0.085
				Time-based	0.001	95.04±0.035
				Cyclical	0.1	95.04±0.058
$\boldsymbol{FF_{tlew}}$	**ACS**	**9**	**0.95**	None	-	95.13±0.106
				Time-based	0.001	95.15±0.085
				Cyclical	**0.001**	**95.17±0.035**
	MMAS	**8**	**0.95**	None	-	95.14±0.092
				Time-based	0.001	95.11±0.074
				Cyclical	**0.01**	**95.15±0.054**

As shown in Figure 22, we illustrated the distribution of the pheromone in epoch 1 (first column), epoch 75 (second column), and epoch 150 (third column) when using the ACS algorithm with different learning rate schedules. We found that the ACS algorithm makes the distribution of pheromones poor (see Figure 22 a)). As a result, the ACS algorithm chooses the same CNN models without considering the new models. In comparison, when using the ACS algorithm with a cyclical learning rate schedule, the pheromone distribution gets better after epoch 75, as shown in Figure 22 c) in the second and third columns. Significantly, addition of a learning rate schedule increases the chance that the ACS algorithm selects the new best CNN model.

We demonstrate two graphs to confirm that adding the learning rate schedule improves the performance of the ACS algorithm. Figure 23 a) represents the fitness value of ACS+Cyclical that dropped after epoch 75, while the fitness value of the ACS algorithm did not reduce after epoch 75. Thus, the chance of discovering the a CNN model is increased. Consequently, Figure 23 b) illustrates the accuracy of the ACS+Cyclical and ACS algorithms. The graph showed that the ACS+Cyclical algorithm outperformed using only the ACS algorithm in every epoch, except only epoch 125.

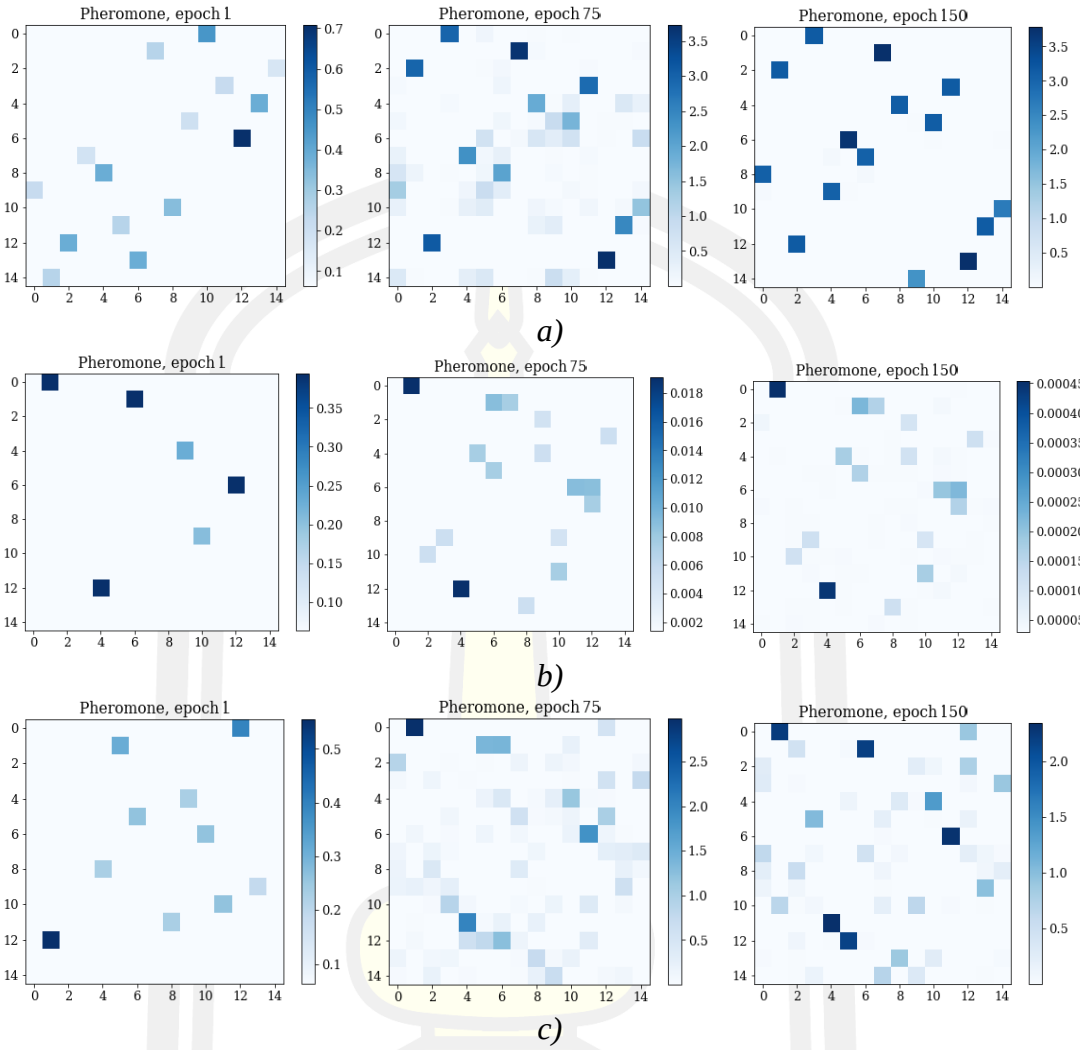


Figure 22 *Illustrated the adaptation of the pheromone when using (a) ACS algorithm, (b) ACS algorithm with time-based learning rate schedule, and (c) ACS algorithm with cyclical learning rate schedule.*

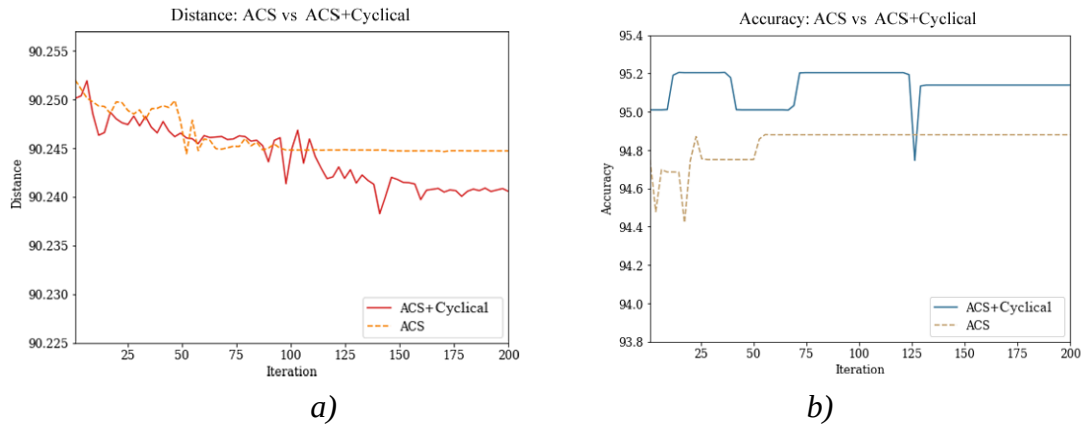


Figure 23 *Illustration of a) the fitness values and b) accurate performance of the ACS and ACS+Cyclical algorithms.*

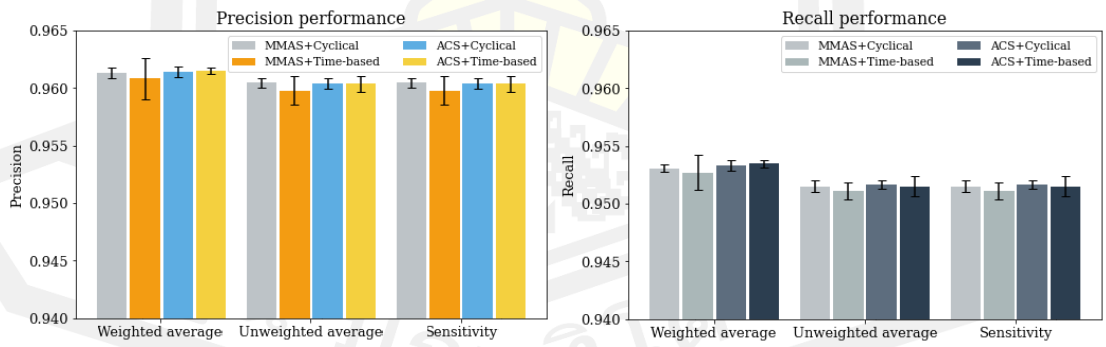
Table 10 *Performance of the ACO algorithms when classified the results with ensemble learning methods on the mulberry leaf dataset.*

ACO	Learning Rate Schedule	Learning rate	Ensemble Learning Method					
			Unweighted Average		Weighted Average		Cost-sensitive	
			Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
ACS	Time-based	0.001	95.15±0.085	0.9500±0.0009	**95.34±0.031**	0.9520±0.0003	95.15±0.085	0.9500±0.0009
	Cyclical	0.001	**95.17±0.035**	0.9501±0.0004	95.33±0.046	0.9519±0.0005	**95.17±0.035**	0.9501±0.0004
MMAS	Time-based	0.001	95.11±0.074	0.9497±0.0007	95.26±0.149	0.9513±0.0018	95.11±0.074	0.9497±0.0007
	Cyclical	0.01	95.15±0.054	0.9501±0.0005	95.30±0.036	0.9517±0.0003	95.15±0.054	0.9501±0.0005

4.6.2.3 Performance of Ensemble Methods

From the experiment in Section 4.6.2.2, we used the unweighted ensemble learning method to evaluate the performance of the ACO algorithms. As a result, we found that the ACO algorithms achieved the highest performance when training using FF_{tlew} fitness function. Hence, in this section, we experimented with three ensemble learning methods; an unweighted average, weighted average, and cost-sensitive, to present the ensemble learning methods affecting the performance of the image classification system.

As shown in Table 10, the cost-sensitive and unweighted average methods presented the same performance with 95.17% accuracy when using the ACS algorithm with a cyclical learning rate (learning rate = 0.001). On the other hand, the weighted average methods achieved the highest performance with an accuracy of 95.34% when using the ACS algorithm with the time-based learning rate schedule (learning rate =0.001). The precision, recall, and the ROC curve are shown in Figure 24 a), 24 b), and 25, respectively. So, we obtained the AUC value of 0.997 (see Figure 25).



a) *b)*

Figure 24 *Illustration of the precision a), and recall b) of the ensemble methods.*

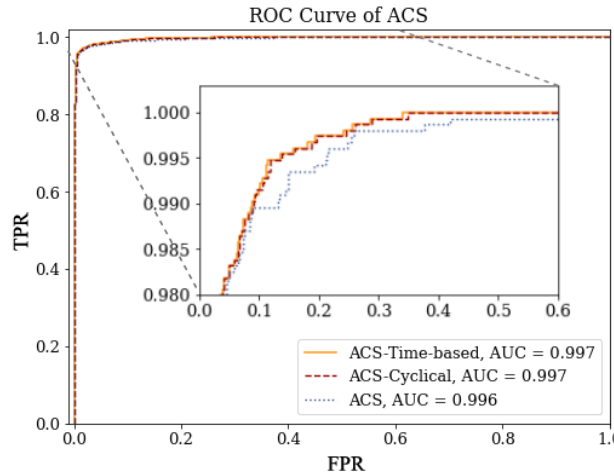
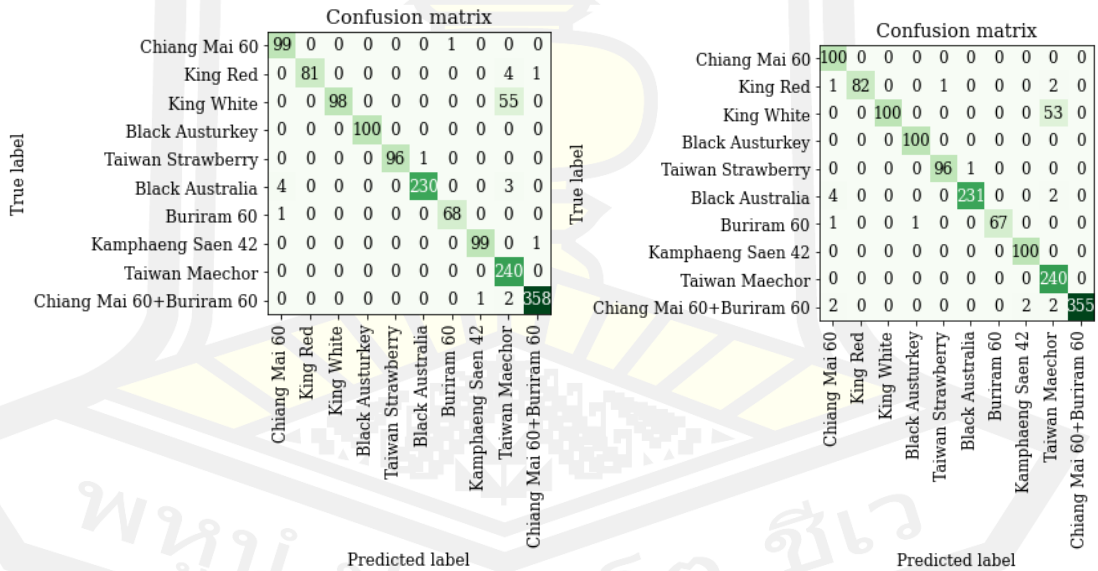


Figure 25 *Illustration of the receiver-operating characteristic curve of the ACS algorithm with two learning rate schedule: time-based and cyclical.*

The confusion matrices of the unweighted average and weighted average learning methods are shown in Figure 26. We found that the most misclassified class was the class of king white. It was classified as the Taiwan Maechor. We also visualized the misclassified images. We found that misclassified images were taken with a lower perspective, were backlit, and in low light, as shown in Figure 27.



a) *b)*

Figure 26 *The confusion matrices of the unweighted average a) and weighted average b) ensemble method on the mulberry leaf dataset.*

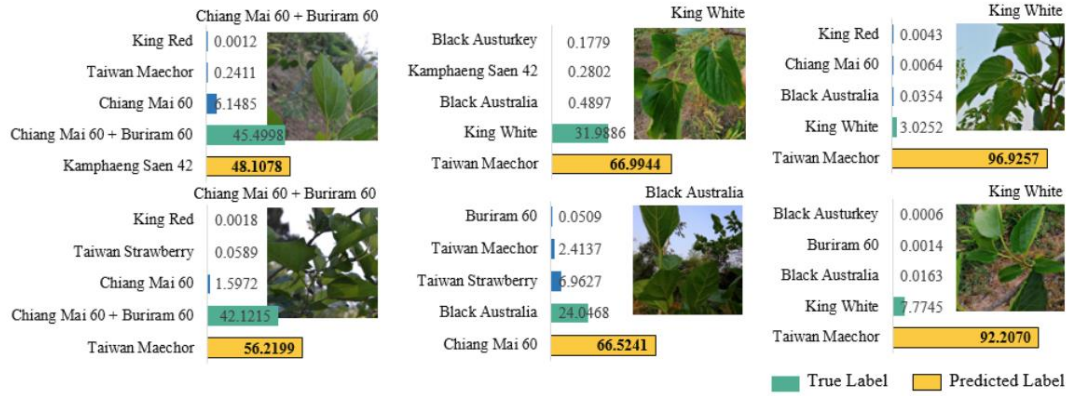


Figure 27 *Illustration of the misclassified images on the mulberry leaf dataset.*

Figure 28 shows the model selection or shortest path using the ACO algorithm. Figure 28 a) and 28 b) show different shortest paths of the 5 CNN models that achieved an accuracy of 95.20%. Figures 28 c) and 28 d) show that using more CNN models sometimes did not achieve the highest accuracy. The ACO algorithm selected 8 (see Figure 28 c)) and 7 (see Figure 28 d)) CNN models and achieved slightly less performance with an accuracy of 95.14%.

4.6.3 Recognition Performance on the Turkey-plant Dataset

The previous section reported on the evaluation of the ACO algorithm on the mulberry leaf dataset and found that the proposed method assigned the best CNN models for use in the ensemble learning method. In this section, we then experimented with the ACO algorithm on the Turkey-plant dataset to ensure that the proposed algorithm always selects the best model. The performance of the proposed method is shown as follows.

4.6.3.1 Performances Evaluation of CNNs

This section reports on using the pre-trained CNN models that were trained on the ImageNet and mulberry leaf datasets consisting of MobileNetV1, MobileNetV2, NASNetMobile, DenseNet121, and Xception. For the fine-tuning scheme, we divided 90% of the Turkey-plant dataset as a training set, 10% as a validation set, and 10% as a test set. Then, we fine-tuned the CNN models with the following parameters; two optimization algorithms (SGD and RMSProp), the learning rate value of 0.1 and 0.0001, and the batch size of 8 and 16.

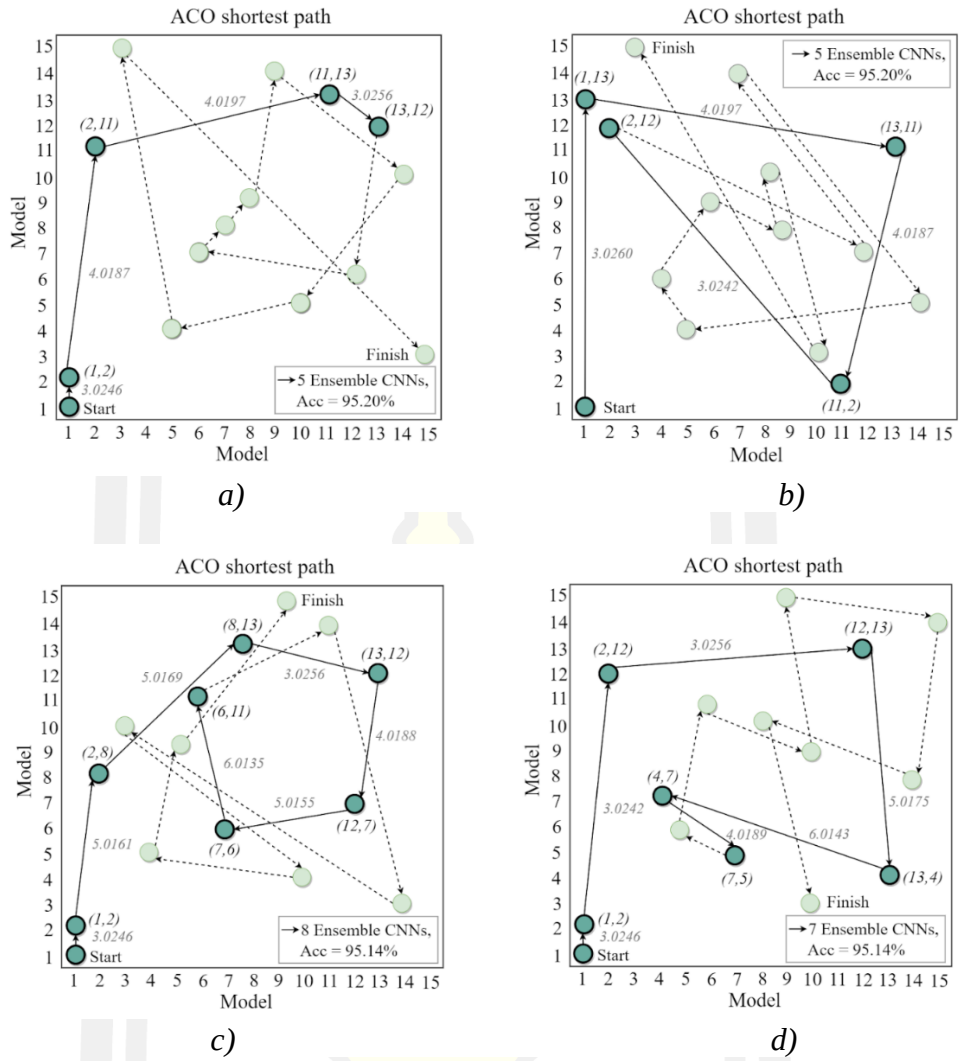


Figure 28 *Illustrated four different paths that selected using the ACO algorithm, including a) 1->2->11->13->12, b) 1->13->11->2->12, c) 1->2->8->13->12->7->6->11, and d) 1->2->12->13->4->7->5. Note that the arrow sign (->) means the sequence of the CNN models when experimenting in the ensemble learning method.*

Table 11 *Performance evaluation of the CNN models on the Turkey-plant dataset.*

Model	Optimizer	Learning rate	Batch Size	Accuracy (%)	
				Pre-trained Model	Pre-trained Model of Mulberry Dataset
MobileNet	RMSprop	0.0001	8	96.67	96.90
MobileNetV2	RMSprop	0.0001	16	95.34	95.79
NASNetMobile	RMSprop	0.0001	8	87.80	91.57
DenseNet121	SGD	0.01	8	98.00	**98.89**
Xception	RMSprop	0.0001	8	97.34	96.67

As shown in Table 11, we found that the DenseNet121 architecture achieved the highest accuracy of 98.89% when using SGD optimization and fine-tuned on the mulberry leaf dataset. In contrast, NASNetMobile obtained an accuracy of only 87.80% when using the pre-trained model of the ImageNet, which is the worst performance. In the following experiments, however, we used the ACO algorithm to select the best CNN models from all the CNN models trained in this section.

4.6.3.2 Comparison of Learning Rate Schedule in the ACO Algorithm

To compare the accuracy performance of the learning rate schedule (time-based and cyclical), we then adjust the following parameters; learning rate, beta, decay, and fitness function.

As shown in Table 12, the experimental results indicated that training the ACO algorithm using the learning rate schedule consistently achieved better performance than without using the learning rate schedule. Consequently, using the FF_{tlew} fitness function achieved higher performance than the FF_{tl} fitness function. The result showed that the MMAS method when tuning with beta = 8 and decay = 0.95 and using a time-based learning rate schedule with a learning rate of 0.001 achieved an accuracy of 99.11% on the Turkey-plant dataset.

Table 12 *Accuracy performance (%) of the ACO algorithms on the Turkey-plant dataset when applying learning rate schedule and training on different fitness functions.*

Fitness Function	ACO	Beta	Decay	Learning Rate Schedule	Learning Rate	Accuracy (%)
FF_{tl}	ACS	1	0.95	None	-	98.94±0.186
				Time-based	0.1	99.02±0.121
				Cyclical	0.001	99.02±0.121
	MMAS	1	0.95	None	-	98.94±0.099
				Time-based	0.001	99.07±0.099
				Cyclical	0.1	98.94±0.186
$\boldsymbol{FF_{tlew}}$	ACS	9	0.95	None	-	98.98±0.121
				Time-based	0.001	98.98±0.336
				Cyclical	0.001	99.02±0.121
	MMAS	**8**	**0.95**	None	-	99.02±0.121
				Time-based	**0.001**	**99.11±0.157**
				Cyclical	0.01	99.02±0.121

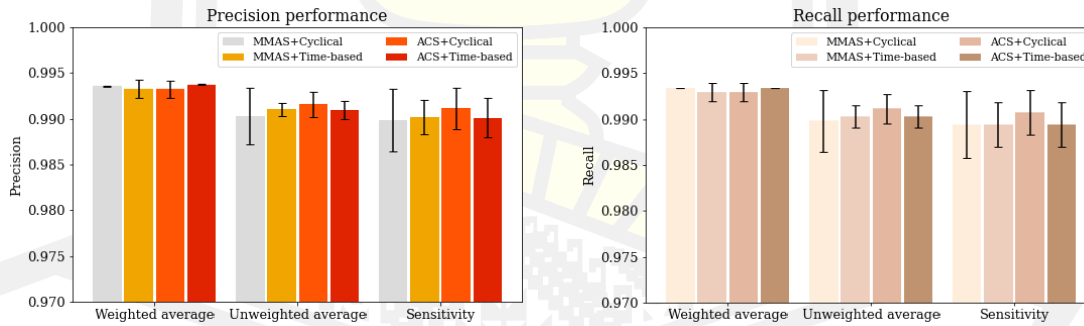
Table 13 *Performance of the ACO algorithms with ensemble learning methods on the Turkey-plant dataset.*

ACO	Learning rate schedule	Learning rate	Ensemble Learning Methods					
			Unweighted Average		Weighted average		Cost-sensitive	
			Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
ACS	Time-based	0.001	98.98±0.336	0.9903±0.0031	**99.33±0.000**	0.9934±0.0000	98.94±0.364	0.9893±0.0037
	Cyclical	0.001	99.02±0.121	0.9910±0.0007	99.29±0.098	0.9929±0.0010	98.94±0.243	0.9894±0.0024
MMAS	Time-based	0.001	99.11±0.157	0.9916±0.0014	99.29±0.098	0.9929±0.0010	99.07±0.243	0.9907±0.0024
	Cyclical	0.01	99.02±0.121	0.9910±0.0010	**99.33±0.000**	0.9934±0.0000	98.94±0.243	0.9894±0.0024

4.6.3.3 Performance Evaluation of Ensemble Learning Methods

In this experiment, we compared the accuracy and F1-score results of three ensemble learning methods; unweighted average, weighted average, and cost-sensitive. We show the obtained results with the ensemble learning methods on the Turkey-plant dataset in Table 13.

To summarize the experimental results, the weighted average ensemble learning method outperformed both the unweighted average and cost-sensitive in terms of accuracy. The weighted average method achieved an accuracy of 99.33% and an F1-score of 99.34% on the Turkey-plant dataset. On the other hand, the cost-sensitive method achieved the worst performance. Furthermore, the precision, recall, and ROC curve are shown in Figure 29 a), 29 b), and 30, respectively. The AUC value (see Figure 30) showed that the ACS method using the cyclical learning rate achieved a very high value of 0.99995.



a) *b)*

Figure 29 *Precision a), and recall b) of the ensemble CNN methods on the turkey-plant dataset.*

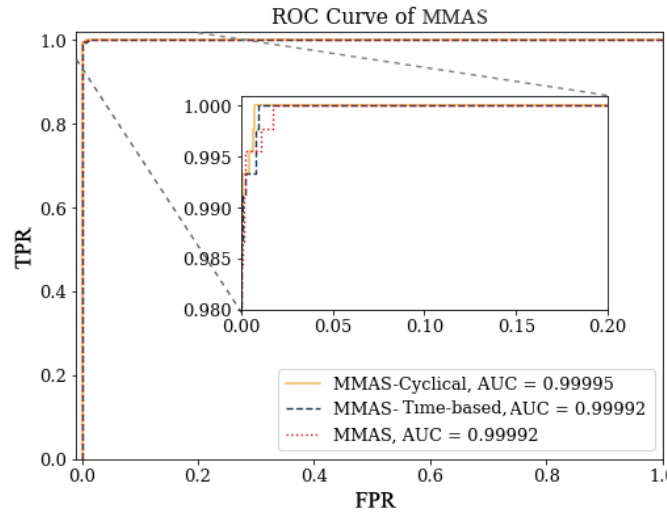


Figure 30 *Comparison of ROC curve of MMAS with different learning rate schedule on the turkey-plant dataset.*

We present the confusion matrix of the weighted average method in Figure 31. It shows only three images that were misclassified. The misclassified images are visualized in Figure 32.

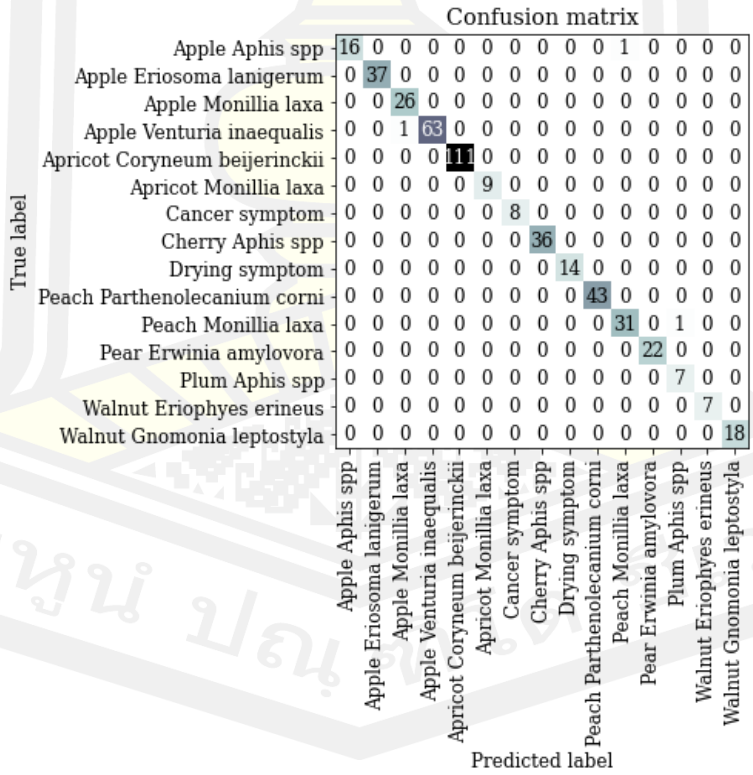


Figure 31 *The confusion matrix of the weighted average method on the Turkey-plant dataset.*

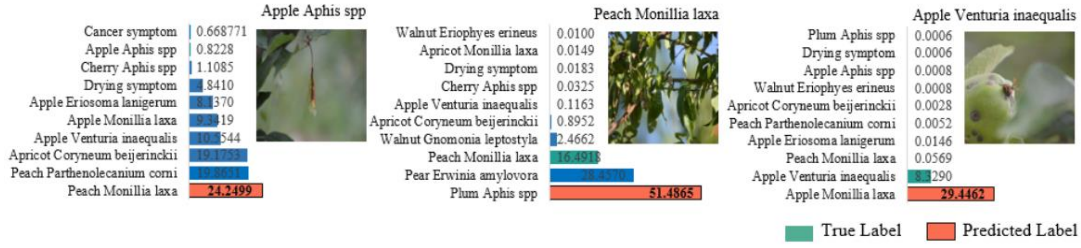
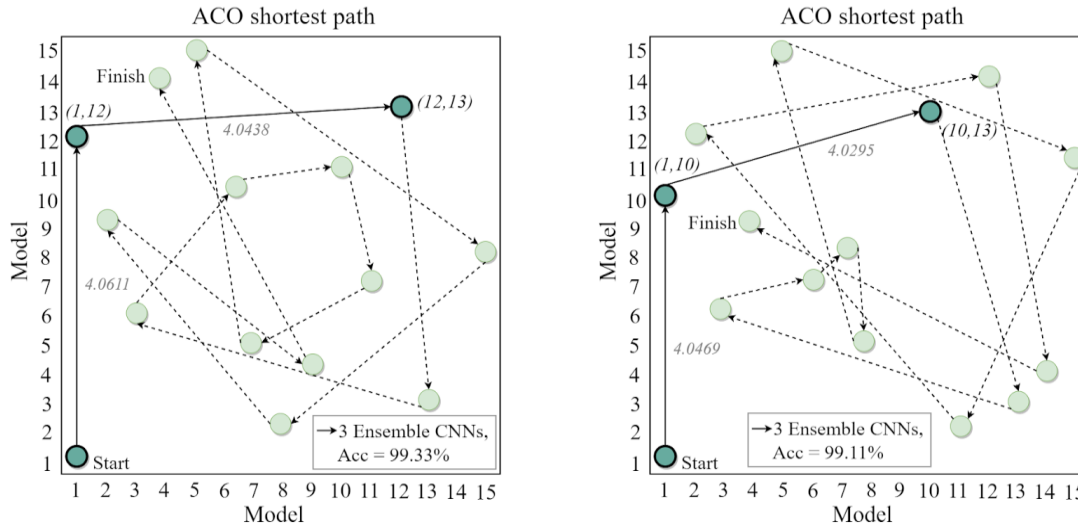


Figure 32 *Illustration of the three misclassified images on the Turkey-plant dataset.*



a) *b)*

Figure 33 *Illustrated two paths that were selected using the ACO algorithm, including a) 1->12->13 and b) 1->10->13.*

Figure 33 illustrates the shortest path using the ACO algorithm. Figure 33 a) and 33 b) present different shortest paths of the 3 CNN models. As a result, Figure 33 a) achieved an accuracy of 99.33% and Figure 33 b) attained an accuracy of 99.11%.

4.7 Big-O Analysis Results

We used the Big-O analysis to measure the time computation performance of the proposed algorithm, as follows.

4.7.1 Big-O analysis of ACO algorithm

As shown in Algorithm 2, Lines 2-6 are used as the route construction by ant systems: ACS and MMAS. An ant will find the best route in each iteration using ACS or MMAS algorithms. In this case, Big-O = O(|M|), where |M| is the number of ants. Line 7 is the pheromone updating that uses Big-O = O(1), but it computes the route in each iteration. Hence, Big-O = (1.|N|). In Line 10, the pseudo-code allows the program to save the best route, using Big-O = O(1). However, it computes in every iteration. In

this algorithm, the Big-O analysis is O(|M|+1.|N|+1.|N|). In conclusion, the Big-O analysis of this algorithm is Big-O = O(|M|.|N|).

Algorithm 2: ACO algorithm

```
1.    For i =1 to N do
2.        For j =1 to M do
3.            Route construction by ant system
4.                if ACS algorithm
5.                if MMAS algorithm
6.        End for
7.        Update the pheromone by ant system
8.            if ACS algorithm
9.            if MMAS algorithm
10.       Save best ACO solution (T_(i,j)^best)
11.   End for
```

4.7.2 The Big-O analysis of the ACO algorithm combined with an ensemble method

Algorithm 3: ACO algorithm combined with ensemble method

```
1.  For i =1 to N do
2.        For j =1 to M do
3.            Route construction by ant system
4.                if ACS algorithm
5.                if MMAS algorithm
6.            Create ensemble CNNs
7.            For k =1 to R do
8.                if unweighted average method
9.                if weighted average method
10.               if cost-sensitive learning method
11.           End for
12.       End for
13.       Update the pheromone by ant system
14.           if ACS algorithm
15.           if MMAS algorithm
16.       Save best ACO solution (T_(i,j)^best)
17.       End for
```

As shown in Algorithm 3, Lines 2-12 are used as the route construction by ant systems. In this study, Big-O = O(|M|.|R|), where |M| is the number of ants and |R| is the number of CNN models. Line 13 is the pheromone updating that uses Big-O = O(1), but it computes the route in every iteration , then Big-O = (1.|N|). In Line 16, the pseudo-code allows the program to save the best route, using Big-O = O(1). In this algorithm,

the Big-O analysis is O(|M|.|R|+1.|N|+1.|N|). In conclusion, the Big-O analysis of this algorithm is Big-O = O(|M|.|R|.|N|).

4.7.3 The Big-O Analysis of the ACO algorithm combined learning rate schedule and ensemble method

Algorithm 4: ACO algorithm combined learning rate schedule and ensemble method

1. **For** i =1 to N **do**
2. **For** j =1 to M **do**
3. Route construction by ant system
4. **if** ACS algorithm
5. **if** MMAS algorithm
6. Create ensemble CNNs
7. **For** k =1 to R **do**
8. **if** unweighted average method
9. **if** weighted average method
10. **if** cost-sensitive learning method
11. **End for**
12. **End for**
13. Calculate η by learning rate schedule
14. **if** time-based scheduler
15. **if** cyclical learning rate scheduler
16. Update the pheromone by ant system
17. **if** ACS algorithm
18. **if** MMAS algorithm
19. Save best ACO solution ($T_{(i,j)}^{best}$)
20. **End for**

As shown in Algorithm 4, Lines 2-12 are used as the route construction by ant systems. In this study, Big-O = O(|M|.|R|), where |M| is the number of ants and |R| is the number of CNN models. Line 13 computes the learning rate using Big-O = O(1), but it computes in every iteration, then Big-O = (1.|N|). Line 16 is the pheromone updating that uses Big-O = O(1), but it computes the route in every iteration, then Big-O = (1.|N|). In Line 19, the pseudo-code allows the program to save the best route, using Big-O = O(1). In this algorithm, the Big-O analysis is O(|M|.|R|+1.|N|+1.|N|). In conclusion, the Big-O analysis of this algorithm is Big-O = O(|M|.|R|.|N|).

To conclude, the ACO algorithm has a Big-O analysis with higher performance than the ACO algorithm combined with the ensemble method. Additionally, Big-O analysis of the ACO algorithm combined with the ensemble method shows a similar analysis with the ACO algorithm combined learning rate schedule and ensemble method.

4.8 Comparison of the Proposed ACO Algorithm and Other Existing Methods

We compared our proposed ACO algorithm and existing methods on two datasets: mulberry leaf and Turkey-plant. The experimental results showed that the proposed ACO algorithm, which changes the fitness function and adds a learning rate schedule to find the shortest path of the CNN models, strongly outperformed the previous method on both mulberry leaf and Turkey-plant datasets with an accuracy of 95.34% and 99.33%, respectively. The comparative results of the two plant leaf datasets are shown in Table 14.

Table 14 *Recognition performance on the Turkey-plant and the mulberry leaf datasets with the existing methods.*

Dataset	Reference	Method	Accuracy (%)
The mulberry leaf dataset	Chompookham and Surinta (Chompookham & Surinta, 2021)	5-EnsCNNs	94.75
	Our proposed	**Automatic model Selection** (The proposed ACO algorithm and Ensemble CNNs with weighted average method)	**95.34**
The Turkey-plant dataset	Turkoglu et al. (Turkoglu et al., 2021)	PlantDiseaseNet: Ensemble CNNs with 5 CNN models	97.56
	Our proposed	**Automatic model Selection** (The proposed ACO algorithm and Ensemble CNNs with weighted average method)	**99.33**

4.9 Discussion

In this paper, we proposed an automatic model selection based on the ant colony optimization (ACO) algorithm that aims to select the robust convolutional neural network (CNN) model. In the original ACO algorithm, however, we found that the ACO algorithm selects the shortest path based on the attractiveness values calculated from the pheromones table. It is highly possible that the ants may walk on the same path. So, the chance of finding new routes is low. We then proposed the new fitness function, called FF_{tlew} and added a learning rate schedule to the ACO algorithm that distributes the value of pheromones to the appropriate pheromone values, as shown in

Figures 22 a) and 22 c). Consequently, all the selected CNN models, which were automatically selected using the ACO algorithm, achieved better performance when classified using the ensemble CNNs method.

4.10 Conclusion

This research proposed a new ant colony optimization (ACO) algorithm that aims to select the robust convolutional neural network (CNN) models in ensemble CNNs. For the proposed ACO algorithm, we first designed a new fitness function, and second, the learning rate schedule was added to the ACO algorithm to learn the fitness function and decrease the fitness function in each iteration. The advantage of the proposed methods is the distribution of the pheromone values, so it could take the chance to select the new robust CNN models, not select only the old CNN models. Subsequently, the robust CNN models selected by the proposed ACO algorithm were used in the ensemble learning method. Furthermore, when the proposed ACO algorithm selects two sets of CNN models, these two sets of CNN models always attained high performance.

We evaluated the proposed ACO algorithm on two plant leaf datasets: mulberry and Turkey-plant. First, we trained state-of-the-art CNN models (MobileNetV1, MobileNetV2, DenseNet121, NASNetMobile, and Xception) with fine-tuned various hyperparameters, including data augmentation techniques (height shift, vertical flip, and fill mode), optimization algorithms (SGD and RMSProp), and learning rate (0.1, 0.01, and 0.001), which is 15 CNN models in total. Hence, the proposed ACO algorithm enables the automatic selection of robust CNN models. Second, for the ACO algorithm, we also compared two ACO frameworks: the ant colony system (ACS) and the max-min ant system (MMAS), to present the best ACO framework. We found that the MMAS framework outperformed the ACS framework. Third, three ensemble learning methods; unweighted average, weighted average, and cost-sensitive, were compared. In our experiments, the weighted average method performed the best. Further, the grid-search method was proposed to discover the weighted parameters. As a result, the proposed ACO algorithm achieved high accuracy on the Turkey-plant dataset with above 99.33% and achieved 95.34% on the mulberry leaf dataset. We compared our proposed ACO algorithm with other methods and found that the proposed ACO algorithm outperformed the existing methods on mulberry and Turkey-plant leaf datasets.

There is still space for improving the accuracy of the mulberry leaf dataset because the proposed ACO algorithm achieved an accuracy of only 95.34%. To enhance the performance, we plan to work on the ACO algorithm using local search in future work. Other meta-heuristic methods include particle swarm optimization (PSO) (Gad, 2022; Jana, Mitra, Pan, Sural, & Chattaraj, 2019) and artificial bee colony optimization (M. Zhao, Song, & Xing, 2022), also the aim of our considerations. The combination and hybrid approaches, such as PSO & line spectral frequencies (LSF) (Neekabadi & Kabudian, 2018), genetic algorithm & voltage source inverter (VSI) (Lopez, Cruz, & Gutierrez, 2021), harmony search & evolution strategy, will be considered in future work.

Chapter 5

Discussion

The research objective described in the dissertation is to design and develop an automatic image classification system using deep learning to classify plant disease problems. Firstly, we focused on classifying plant leaf images taken in the lab. We introduced multiple grids to divide plant leaf images. The divided images were then extracted for robust features using traditional feature extraction methods. Then, the dimensionality reduction method was used to reduce the feature vector size and bring the feature vector learned and classified using machine learning methods. Secondly, we proposed the ensemble convolutional neural network (CNN) method to create robust CNN models and then combined the output of each CNN model to generate the optimal predictive model, called the ensemble method. In this process, we classified plant leaf disease images that were captured from natural environments. Finally, we focused on the optimization algorithm called ant colony optimization (ACO) to automatically select the robust CNN models. We improved the performance of the ACO algorithm by proposing a new fitness function and adding the objective function to the ACO algorithm to distribute the pheromone values. We proved that the ACO algorithm when adding the proposed method could automatically select the robust CNN models and improve the efficiency of the image classification system.

We briefly discussed the challenges of an automatic image classification system using the traditional method and deep learning method.

In Chapter 2, due to the problem of classifying leaves of different species with similar leaf shapes and leaves of the same plant but with different leaf shapes, we presented multiple grids and a dimensionality reduction-based descriptor approach to solving this problem. First, the multiple grid method was used to divide the leaf images into subareas and then calculate the subarea using robust feature extraction methods, including histogram of oriented gradients (HOG), local binary patterns (LBP), and color histogram. The distinctive features of the plant leaf were extracted in this process. Second, the distinctive features were fed to principal component analysis (PCA), which is a dimensionality reduction method, to reduce the feature vector size. Finally, the machine learning techniques, including support vector machine (SVM) and multi-layer perceptrons (MLP), were used to create the model from the reduction features. The experimental result showed that the proposed method achieved high accuracy with more than 99% on the Folio dataset.

Chapter 3 presented the ensemble CNN method to classify plant leaf images taken in natural environments. First, the robust CNN models were created based on state-of-the-art CNN architectures, including MobileNetV1, MobileNetV2, Xception, DenseNet121, and NASNetMobile. We fine-tuned the CNN models using three optimization algorithms (stochastic gradient descent (SGD), Adam, and RMSProp), different learning rates (0.1, 0.01, 0.001, and 0.0001), batch sizes (8, 16, 32, and 64), and different data augmentation methods. Second, three and five robust CNN models were discovered and used in the ensemble learning method, called 3-EnsCNNs and 5-EnsCNNs. Finally, the output probabilities of CNN models were then transferred to the ensemble learning method to classify the plant leaf images. Three ensemble methods were compared for the ensemble learning method: unweighted majority vote,

unweighted average, and weighted average. The proposed methods were evaluated on three plant leaf image datasets: mulberry, tomato, and corn. The experimental results showed that the weighted average ensemble method outperformed other ensemble methods for the ensemble learning method. The ensemble CNN method achieved more than 99% accuracy on the tomato and corn leaf diseased datasets and above 94% on the mulberry leaf dataset.

Chapter 4 aimed to select the best-CNN models for use in the ensemble CNN method. We then proposed an automatic model selection based on the ACO algorithm. The most significant of the proposed ACO algorithms is the chance of finding new routes and still obtaining a high classification performance. To improve the performance of the ACO algorithm, we presented the new fitness function and learning rate schedules used for calculating the pheromones. The proposed method could distribute the pheromone to the appropriate values. The proposed ACO algorithm increases the chance of automatically selecting the best-CNN models. Further, the best-CNN models were used in the ensemble learning method, called the ensemble CNN method. The ensemble CNN method is highly accurate when classified using the best-CNN models.

5.1 Answers to the Research Questions

This section answered three research questions (RQ) related to improving the plant leaf classification system in detail, according to the research question in Section 1.

RQ1: Plant species generally can be classified from plant organs, such as leaves, bark, flowers, seeds, and stems. However, the leaf is the most distinctive plant part that could be easier classified than other parts. Is it possible to classify plant leaf images using image processing and machine learning methods? Extracting the features from plant leaves is an important method. Furthermore, many local descriptor methods are proposed to extract the robust features (called handcraft features) from objects that appear in the image. Could machine learning techniques accurately classify the plant leaf images that extract the handcraft features using local descriptor methods and color features?

To answer RQ1, we focused on improving the image processing and machine learning methods to classify plant leaf images taken in a lab with a white background, called the Folio dataset, which contained 32 different plant leaf species. First, we proposed a grid-based technique to divide plant leaf images into subareas and then extracted the features using different feature extraction methods: HOG, LBP, and color histogram. We also used PCA to decrease the feature size calculated from the feature extraction methods. Hence, relatively low-dimensional features were transferred to the machine learning techniques to create a model and classify. To select the best machine learning technique, we compared two techniques: SVM and MLP, in terms of accuracy and computation time. The results showed that the SVM technique slightly outperformed the MLP technique. When comparing the feature extraction techniques, we found that the color histogram technique was better than the HOG and LBP methods. Consequently, the combination of these three techniques showed outstanding

results. As a result, the proposed method achieved an accuracy above 98.7% on the Folio dataset.

RQ2: In the previous RQ, image processing and machine learning methods were proposed to classify plants from plant leaf images. However, most plant diseases showed on the leaves, such as downy mildew, leaf spot, leaf blotch/leaf blight, and rust. Could we classify the disease if the disease appears on the plant leaf? Could we classify the plant leaf diseases using the deep learning method, such as CNN? Additionally, is there any method to enhance the performance of the deep learning method?

From work to answer RQ1, we discovered that image processing and machine learning techniques could be proposed to recognize the plant leaf images. In this research, we aimed to use a deep learning technique to classify plant leaf images: healthy and disease, taken from natural environments. To include differnet plant leaf diseases, we then selected the tomato and corn leaf disease datasets, which is the subset of the PlantVillage dataset that is taken from natural environments. For healthy plants, we collected 5,262 mulberry leaf images of ten cultivars from five provinces of Thailand: Chiang Mai, Phitsanulok, Nakhon Ratchasima, Buriram, and Maha Sarakham, called mulberry leaf dataset. This dataset was taken from natural environments and images were recorded from different perspectives.

For the deep learning technique, we first used CNNs to create a robust model. Five CNN models: MobileNetV1, MobileNetV2, NASNetMobile, DenseNet121, and Xception, were trained. The data augmentation techniques, including height shift, vertical flip, fill mode, and mixed-method, were also employed when training the CNN models. In this process, we create various robust CNN models. Second, we use the ensemble CNNs method to classify the plant leaf images. We combined three and five CNN models, called 3-EnsCNNs and 5-EnsCNNs. The multiple CNN outputs could benefit more from classification than only single CNN output. The experimental results showed that the ensemble CNNs method consistently outperformed the single CNN on the mulberry leaf dataset and two leaf disease datasets: tomato and corn leaf disease. As a result, we achieved an accuracy of 94.75% on the mulberry leaf dataset and above 99.4% on two leaf disease datasets.

We could guarantee from our experimental results that the ensemble CNNs method could enhance the performance of the CNN model on the plant leaf classification.

RQ3: If the ensemble learning with the weighted average method performs better classification performance than using only one CNN model. How could we select the best-CNN models to create the ensemble CNNs method? However, could we use ACO can automatically select robust CNN models and ensemble the CNN models in the ensemble learning method?

We found from RQ2 that the ensemble CNNs method performs better than one CNN model. However, selecting the best-CNN models is not manageable when we have too many of them. We proposed the automated method to select the best-CNN models based on ACO. The proposed ACO method computed the pheromone with the

new fitness function and learning rate schedules, such as the time-based and cyclical. This technique could create more distributed pheromone values. Then, ants could create the new best path. As a result, the proposed method could automatically select many sets of the best-CNN models. We then created ensemble CNNs that were automatically selected using the proposed ACO method and evaluated on the mulberry leaf and Turkey-plant disease image datasets. The result showed that each set of the best-CNN models achieved high accuracy.

5.2 Future Work

This thesis presented a novel automated plant leaf classification system using a deep learning method. However, the research still has a gap in enhancing the performance of plant leaf image classification. Firstly, if the plant leaf images are inadequate, the data augmentation techniques, such as generative adversarial networks (GAN) (Shorten & Khoshgoftaar, 2019), AutoAugment (Cubuk, Zoph, Mané, Vasudevan, & Le, 2019), and the sample paring method (Inoue, 2018), are the most acceptable solution to generate new plant leaf images. Secondly, the local search and incremental local search are suggested for the ACO algorithm. Other complex and robust meta-heuristic methods, such as particle swarm optimization (PSO) (Gad, 2022; Jana et al., 2019), artificial bee colony optimization (M. Zhao et al., 2022), and also the hybrid approaches, such as PSO & line spectral frequencies (LSF) (Neekabadi & Kabudian, 2018), genetic algorithm & voltage source inverter (VSI) (Lopez et al., 2021), and harmony search & evolution strategy (Weyland, 2015) are solutions to selecting robust CNN models.

REFERENCES

Abas, M. A. H., Ismail, N., Yassin, A. I. M., & Taib, M. N. (2018). VGG16 for Plant Image Classification with Transfer Learning and Data Augmentation. *International Journal of Engineering & Technology*, *7*, 90–94. https://doi.org/10.14419/ijet.v7i4.11.20781

Adhitya, Y., Prakosa, S., Köppen, M., & Leu, J.-S. (2019). Convolutional Neural Network Application in Smart Farming. In *Soft Computing in Data Science* (pp. 287–297). https://doi.org/10.1007/978-981-15-0399-3_23

Alabbas, M., Khudeyer, R., & Jaf, S. (2016). Improved Arabic Characters Recognition by Combining Multiple Machine Learning Classifiers. *2016 International Conference on Asian Language Processing (IALP)*, 262–265. https://doi.org/10.1109/IALP.2016.7875982

Altman, N. S. (1992). An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician*, *46*(3), 175–185. https://doi.org/10.2307/2685209

Amara, J., Bouaziz, B., & Algergawy, A. (2017). A Deep Learning-based Approach for Banana Leaf Diseases Classification. *Datenbanksysteme Für Business, Technologie Und Web (BTW 2017)*, 79–88.

Arafat, S. V., Saghir, M. I., Ishtiaq, M., & Bashir, U. (2016). Comparison of Techniques for Leaf Classification. *2016 Sixth International Conference on Digital Information and Communication Technology and Its Applications (DICTAP)*, 136–141. https://doi.org/10.1109/DICTAP.2016.7544015

Aravind, K. R., Raja, P., Mukesh, K. V, Aniirudh, R., Ashiwin, R., & Szczepanski, C. (2018). Disease Classification in Maize Crop using Bag of Features and Multiclass Support Vector Machine. *International Conference on Inventive Systems and Control (ICISC)*, 1191–1196. https://doi.org/10.1109/ICISC.2018.8398993

Atabay, H. A. (2016). A Convolutional Neural Network with a New Architecture Applied on Leaf Classification. *Institute of Integrative Omics and Applied Biotechnology (IIOAB)*, *7*, 326–331.

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded Up Robust Features. In A. Leonardis, H. Bischof, & A. Pinz (Eds.), *Computer Vision-European Conference on Computer Vision* (pp. 404–417). Berlin, Heidelberg: Springer Berlin Heidelberg.

Bisen, D. (2021). *Deep Convolutional Neural Network Based Plant Species Recognition Through Features of Leaf*. 6443–6456.

Blum, C., & Roli, A. (2003). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*, *35*(3), 268–308. https://doi.org/10.1145/937503.937505

Caballero, C., & Aranda, C. (2010). Plant Species Identification using Leaf Image Retrieval. *CIVR 2010 - 2010 ACM International Conference on Image and Video Retrieval*, 327–334. https://doi.org/10.1145/1816041.1816089

Caglayan, A., Guclu, O., & Can, A. B. (2013). A Plant Recognition Approach using Shape and Color Features in Leaf Images. *International Conference on Image Analysis and Processing*, *8157*, 161–170. https://doi.org/10.1007/978-3-642-41184-7_17

Cerutti, G., Tougne, L., Coquin, D., & Vacavant, A. (2013). Curvature-Scale-Based Contour Understanding for Leaf Margin Shape Recognition and Species Identification. *8th International Conference on Computer Vision Theory and Applications, VISAPP 2013*, *1*, 277–284. https://doi.org/10.5220/0004225402770284

Cerutti, G., Tougne, L., Mille, J., Vacavant, A., & Coquin, D. (2013). Understanding Leaves in Natural Images – A Model-Based Approach for Tree Species Identification. *Computer Vision and Image Understanding*, *117*(10), 1482–1501. https://doi.org/10.1016/j.cviu.2013.07.003

Chaki, J., Parekh, R., & Bhattacharya, S. (2015). Plant Leaf Recognition using Texture and Shape Features with Neural Classifiers. *Pattern Recognition Letters*, *58*, 61–68. https://doi.org/10.1016/j.patrec.2015.02.010

Characteristics and Structures of Plants. (2020). Retrieved January 20, 2020, from https://universe-review.ca/R10-34-anatomy2.htm

Chen, L., Su, Y., Zhang, D., Leng, Z., Qi, Y., & Jiang, K. (2021). Research on path planning for mobile robots based on improved ACO. *2021 36th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 379–383. https://doi.org/10.1109/YAC53711.2021.9486664

Cho, S. Y. (2012). Content-Based Structural Recognition for Flower Image Classification. *Proceedings of the 2012 7th IEEE Conference on Industrial Electronics and Applications, ICIEA 2012*, 541–546. https://doi.org/10.1109/ICIEA.2012.6360787

Chollet, F. (2016). Xception: Deep Learning with Depthwise Separable Convolutions. *ArXiv*, *1610.02357*, 1–8. https://doi.org/10.48550/arXiv.1610.02357

Chompookham, T., Gonwirat, S., Lata, S., Phiphiphatphaisit, S., & Surinta, O. (2020).

Plant Leaf Image Recognition Using Multiple-Grid Based Local Descriptor and Dimensionality Reduction Approach. *The 3rd International Conference on Information Science and System (ICISS)*, 72–77. https://doi.org/10.1145/3388176.3388180

Chompookham, T., & Surinta, O. (2021). Ensemble Methods with Deep Convolutional Neural Networks for Plant Leaf Recognition. *Letters, Icic Express*, *15*(6), 553–565. https://doi.org/10.24507/icicel.15.06.553

Cootes, T. F., Taylor, C. J., Cooper, D. H., & Graham, J. (1995). Active Shape Models – Their Training and Application. *Computer Vision and Image Understanding*, *61*(1), 38–59. https://doi.org/10.1006/cviu.1995.1004

Cope, J. S., Corney, D., Clark, J. Y., Remagnino, P., & Wilkin, P. (2012). Plant Species Identification using Digital Morphometrics: A Review. *Expert Systems with Applications*, *39*(8), 7562–7573. https://doi.org/10.1016/j.eswa.2012.01.073

Cubuk, E. D., Zoph, B., Mané, D., Vasudevan, V., & Le, Q. V. (2019). AutoAugment: Learning Augmentation Strategies from Data. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 113–123. https://doi.org/10.1109/CVPR.2019.00020

Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 886–893. https://doi.org/10.1109/cvpr.2005.177

Darwish, A. (2018). Bio-inspired Computing: Algorithms Review, Deep Analysis, and the Scope of Applications. *Future Computing and Informatics Journal*, *3*(2), 231–246. https://doi.org/10.1016/j.fcij.2018.06.001

DeChant, C., Wiesner-Hanks, T., Chen, S., Stewart, E. L., Yosinski, J., Gore, M. A., … Lipson, H. (2017). Automated Identification of Northern Leaf Blight-Infected Maize Plants from Field Imagery Using Deep Learning. *Phytopathology*, *107*(11), 1426–1432. https://doi.org/10.1094/PHYTO-11-16-0417-R

Dewantoro, R. W., Sihombing, P., & Sutarman. (2019). The Combination of Ant Colony Optimization (ACO) and Tabu Search (TS) Algorithm to Solve the Traveling Salesman Problem (TSP). *2019 3rd International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM)*, 160–164. https://doi.org/10.1109/ELTICOM47379.2019.8943832

Dhaware, C. G., & Wanjale, K. H. (2017). A Modern Approach for Plant Leaf Disease Classification Which Depends on Leaf Image Processing. *International Conference on Computer Communication and Informatics (ICCCI)*, 1–4. https://doi.org/10.1109/ICCCI.2017.8117733

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. *Multiple Classifier Systems*, 1–15. Berlin, Heidelberg: Springer Berlin Heidelberg.

Dogan, A., & Birant, D. (2019). A Weighted Majority Voting Ensemble Approach for Classification. *International Conference on Computer Science and Engineering*, 366–371. https://doi.org/10.1109/UBMK.2019.8907028

Dorigo, M. (1992). *Optimization, Learning and Natural Algorithms*. Politecnico di Milano, Italian.

Dorigo, Marco, & Gambardella, L. M. (1997). Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, *1*(1), 53–66. https://doi.org/10.1109/4235.585892

Du, J. X., Wang, X. F., & Zhang, G. J. (2007). Leaf Shape Based Plant Species Recognition. *Applied Mathematics and Computation*, *185*(2), 883–893. https://doi.org/10.1016/j.amc.2006.07.072

Durmuş, H., Güneş, E. O., & Kırcı, M. (2017). Disease Detection on the Leaves of the Tomato Plants by Using Deep Learning. *International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, 1–5. https://doi.org/10.1109/Agro-Geoinformatics.2017.8047016

Enkvetchakul, P., & Surinta, O. (2022). Stacking Ensemble of Lightweight Convolutional Neural Networks for Plant Leaf Disease Recognition. *ICIC Express Letters*, *16*(5), 521–528. https://doi.org/10.24507/icicel.16.05.521

Evert, R. F. (2006). *Esau's Plant Anatomy, Meristems, Cells, and Tissues of the Plant Body: their Structure, Function, and Development* (3rd ed.). New Jersey: John Wiley & Sons, Inc.

Fahmi, H., Zarlis, M., Nababan, E., & Sihombing, P. (2020). Ant Colony Optimization (ACO) Algorithm for Determining The Nearest Route Search in Distribution of Light Food Production. *Journal of Physics: Conference Series*, *1566*, 1–7. https://doi.org/10.1088/1742-6596/1566/1/012045

Fathi Kazerouni, M., Mohammed Saeed, N. T., & Kuhnert, K.-D. (2019). Fully-automatic Natural Plant Recognition System using Deep Neural Network for Dynamic Outdoor Environments. *SN Applied Sciences*, *1*(756), 1–18. https://doi.org/10.1007/s42452-019-0785-9

Florea, A.-C., & Andonie, R. (2019). Weighted Random Search for Hyperparameter Optimization. *International Journal of Computers Communications & Control*, *14*(2), 154–169. https://doi.org/10.15837/ijccc.2019.2.3514

Gad, A. G. (2022). Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. *Archives of Computational Methods in Engineering*, 1–23. https://doi.org/10.1007/s11831-021-09694-4

Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M., & Suganthan, P. N. (2021). Ensemble Deep Learning: A Review. *ArXiv*, *2104.02395*, 1–49. https://doi.org/10.48550/ARXIV.2104.02395

Ghasab, M. A. J., Khamis, S., Mohammad, F., & Fariman, H. J. (2015). Feature Decision-Making Ant Colony Optimization System for an Automated Recognition of Plant Species. *Expert Systems with Applications*, *42*(5), 2361–2370. https://doi.org/10.1016/j.eswa.2014.11.011

Glover, F. W., & Kochenberger, G. A. (2003). *Handbook of Meta-Heuristics* (1st ed.). https://doi.org/10.1007/b101874

Guo, L., Du, S., Chi, Y., Cui, W., Song, P., Zhu, J., … Xu, M. (2019). A Multi-model Ensemble Method using CNN and Maximum Correntropy Criterion for Basal Cell Carcinoma and Seborrheic Keratoses Classification. *International Joint Conference on Neural Networks (IJCNN)*, 1–6. https://doi.org/10.1109/IJCNN.2019.8852434

Hajian-Tilaki, K. (2013). Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation. *Caspian Journal of Internal Medicine*, *4*(2), 627–635.

Hansen, L. K., & Salamon, P. (1990). Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(10), 993–1001. https://doi.org/10.1109/34.58871

Harangi, B. (2018). Skin Lesion Classification with Ensembles of Deep Convolutional Neural Networks. *Journal of Biomedical Informatics*, *86*, 25–32. https://doi.org/10.1016/j.jbi.2018.08.006

Hassan, S. M., Maji, A. K., Jasi´nski, M., Leonowicz, Z., & Jasi´nska, E. (2021). Identification of Plant-Leaf Diseases using CNN and Transfer-Learning Approach. *Electronics*, *10*(1388), 2–19. https://doi.org/10.3390/electronics10121388

Haykin, S. (2008). *Neural Networks and Learning Machines* (3rd ed.). Pearson Prentice Hall.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *ArXiv*, *1512.03385*, 1–12. https://doi.org/10.48550/ARXIV.1512.03385

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, *18*(7), 1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527

Hinton, G., Srivastava, N., & Swersky, K. (2012). Lecture 6.5—RMSProp: Divide the Gradient by a Running Average of Its Recent Magnitude. *COURSERA: Neural Networks for Machine Learning*, pp. 26–31.

Hoffman, H. J., Cruickshanks, K. J., & Davis, B. (2009). Perspectives on Population-Based Epidemiological Studies of Olfactory and Taste Impairment. *Annals of the New York Academy of Sciences*, *1170*, 514–530. https://doi.org/10.1111/j.1749-6632.2009.04597.x

Hossain, J., & Amin, M. A. (2010). Leaf Shape Identification Based Plant Biometrics. *Proceedings of 2010 13th International Conference on Computer and Information Technology, ICCIT 2010*, 458–463. https://doi.org/10.1109/ICCITECHN.2010.5723901

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., … Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *ArXiv*, *1704.04861*, 1–9. https://doi.org/10.48550/ARXIV.1704.04861

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2018). Densely Connected Convolutional Networks. *ArXiv*, *1608.06993*, 1–9. https://doi.org/10.48550/ARXIV.1608.06993

Hughes, D. P., & Salathé, M. (2015). An Open Access Repository of Images on Plant Health to Enable the Development of Mobile Disease Diagnostics. *ArXiv*, *1511.08060*, 1–13. https://doi.org/10.48550/ARXIV.1511.08060

Inoue, H. (2018). Data Augmentation by Pairing Samples for Images Classification. *ArXiv*, *1801.02929*, 1–8. https://doi.org/10.48550/ARXIV.1801.02929

Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). Artificial Neural Networks: A Tutorial. *Computer*, *29*(3), 31–44. https://doi.org/10.1109/2.485891

Jana, G., Mitra, A., Pan, S., Sural, S., & Chattaraj, P. K. (2019). Modified Particle Swarm Optimization Algorithms for the Generation of Stable Structures of Carbon Clusters, Cn (n = 3–6, 10). *Frontiers in Chemistry*, *7*, 1–13. https://doi.org/10.3389/fchem.2019.00485

Jeon, W.-S., & Rhee, S.-Y. (2017). Plant Leaf Recognition using a Convolution Neural Network. *The International Journal of Fuzzy Logic and Intelligent Systems*, *17*(1), 26–34. https://doi.org/10.5391/IJFIS.2017.17.1.26

Jeon, W., & Rhee, S. (2017). Plant Leaf Recognition Using a Convolution Neural Network Plant Leaf Recognition Using a Convolution Neural Network. *International Journal of Fuzzy Logic and Intelligent Systems*, *17*(1), 26–34. https://doi.org/10.5391/IJFIS.2017.17.1.26

Joel, G. N., & Priya, S. M. (2018). Improved Ant Colony on Feature Selection and Weighted Ensemble to Neural Network Based Multimodal Disease Risk Prediction (WENN-MDRP) Classifier for Disease Prediction Over Big Data. *International Journal of Engineering & Technology*, *7*, 56–61. https://doi.org/10.14419/ijet.v7i3.27.17654

Ju, C., Bibaut, A., & van der Laan, M. (2018). The Relative Performance of Ensemble Methods with Deep Convolutional Neural Networks for Image Classification. *Journal of Applied Statistics*, *45*(15), 2800–2818. https://doi.org/10.1080/02664763.2018.1441383

Karaaba, M., Surinta, O., Schomaker, L., & Wiering, M. A. (2015). Robust Face Recognition by Computing Distances from Multiple Histograms of Oriented Gradients. *Proceedings - 2015 IEEE Symposium Series on Computational Intelligence, SSCI 2015*, 203–209. https://doi.org/10.1109/SSCI.2015.39

Khmag, A. (2017). Recognition System for Leaf Images Based on Its Leaf Contour And Centroid. *2017 IEEE 15th Student Conference on Research and Development (SCOReD)*, 467–472. https://doi.org/10.1109/SCORED.2017.8305438

Kreuter, D., Takahashi, H., Omae, Y., Akiduki, T., & Zhang, Z. (2020). Classification of Human Gait Acceleration Data using Convolutional Neural Networks. *International Journal of Innovative Computing, Information and Control*, *16*(2), 609–619. https://doi.org/10.24507/ijicic.16.02.609

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *The Proceedings of the 25th Nternational Conference on Neural Information Processing Systems*, *1*, 1097–1105. https://doi.org/10.1145/3065386

Kumar, A., & Vani, M. (2019). Image Based Tomato Leaf Disease Detection. *International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1–6. https://doi.org/10.1109/ICCCNT45670.2019.8944692

Kumar, N., Belhumeur, P. N., Biswas, A., Jacobs, D. W., Kress, W. J., Lopez, I. C., & Soares, V. B. (2012). Leafsnap: A Computer Vision System for Automatic Plant Species Identification. *Computer Vision–ECCV*, *7573*, 502–516.

https://doi.org/10.1007/978-3-642-33709-3_36

Kumar, R., & Indrayan, A. (2011). Receiver Operating Characteristic (ROC) Curve for Medical Researchers. *Indian Pediatrics*, *48*(4), 277–287. https://doi.org/10.1007/s13312-011-0055-4

Kusumo, B. S., Heryana, A., Mahendra, O., & Pardede, H. F. (2018). Machine Learning-based for Automatic Detection of Corn-Plant Diseases using Image Processing. *International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*, 93–97. https://doi.org/10.1109/IC3INA.2018.8629507

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *In Proceedings of the IEEE*, 2278–2324. https://doi.org/10.1007/BF00774006

Li, H., Chaudhari, P., Yang, H., Lam, M., Ravichandran, A., Bhotika, R., & Soatto, S. (2020). Rethinking the Hyperparameters for Fine-tuning. *ArXiv*, *abs/2002.1*, 1–20. https://doi.org/10.48550/ARXIV.2002.11770

Li, J., Chen, T., Shi, R., Lou, Y., Li, Y.-L., & Lu, C. (2021). Localization with Sampling-Argmax. *ArXiv*, *abs/2110.0*, 1–18. https://doi.org/10.48550/ARXIV.2110.08825

Li, P. (2017). *Optimization Algorithms for Deep Learning*. Retrieved from http://lipiji.com/docs/li2017optdl.pdf

Li, S., Wei, Y., Liu, X., Zhu, H., & Yu, Z. (2022). A New Fast Ant Colony Optimization Algorithm: The Saltatory Evolution Ant Colony Optimization Algorithm. *Mathematics*, *10*(925), 1–22. https://doi.org/10.3390/math10060925

Li, X., & Chen, Z. (2010). Weed Identification Based on Shape Features and Ant Colony Optimization Algorithm. *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, *1*, 384–387. https://doi.org/10.1109/ICCASM.2010.5620445

Lopez, J. R., Cruz, P. P., & Gutierrez, A. M. (2021). VSI LC Filter Optimized by a Genetic Algorithm from Connected to Island Microgrid Operation. *Energy Systems*, 1–19. https://doi.org/10.1007/s12667-021-00432-0

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, *60*(2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Mauseth, J. D. (2016). *Botany : An Introduction to Plant Biology* (Sixth Edit). Jones & Bartlett, Sudbury, MA.

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for

Image-Based Plant Disease Detection. *Frontiers in Plant Science*, *7*, 1–10. https://doi.org/10.3389/fpls.2016.01419

Mokeev, V. (2019). An Ensemble of Learning Machine Models for Plant Recognition. *International Conference on Analysis of Images, Social Networks and Texts (AIST) : Analysis of Images, Social Networks and Texts*, 256–262. https://doi.org/10.1007/978-3-030-39575-9_26

Munisami, T., Ramsurn, M., Kishnah, S., & Pudaruth, S. (2015). Plant Leaf Recognition using Shape Features and Colour Histogram with K-Nearest Neighbour Classifiers. *Procedia Computer Science*, *58*, 740–747. https://doi.org/10.1016/j.procs.2015.08.095

Mutlag, W. K., Ali, S. K., Aydam, Z. M., & Taher, B. H. (2020). Feature Extraction Methods : A Review. *Journal of Physics: Conference Series*, *1591*, 1–10. https://doi.org/10.1088/1742-6596/1591/1/012028

Neekabadi, A., & Kabudian, S. J. (2018). A New Quantum-PSO Metaheuristic and Its Application to ARMA Modeling of Speech Spectrum. *2018 4th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, 55–60. https://doi.org/10.1109/ICSPIS.2018.8700530

Neforawati, I., Herman, N. S., & Mohd, O. (2019). Precision Agriculture Classification using Convolutional Neural Networks for Paddy Growth Level. *Journal of Physics: Conference Series*, *1193*, 1–5. https://doi.org/10.1088/1742-6596/1193/1/012026

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. *Arxiv*, *1811.03378*, 1–20. https://doi.org/10.48550/arXiv.1811.03378

Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 971–987. https://doi.org/10.1109/TPAMI.2002.1017623.

Papadimitriou, C., & Steiglitz, K. (1982). Combinatorial Optimization: Algorithms and Complexity. In *IEEE Transactions on Acoustics, Speech, and Signal Processing* (Vol. 32). https://doi.org/10.1109/TASSP.1984.1164450

Park, J., Yi, D., & Ji, S. (2020). A Novel Learning Rate Schedule in Optimization for Neural Networks and It's Convergence. *Symmetry*, *12*(660), 1–16. https://doi.org/10.3390/sym12040660

Park, S., Suh, Y., & Lee, J. (2020). Clothing Classification using CNN and Shopping

Mall Search System. *ICIC Express Letters*, *11*(8), 773–780. https://doi.org/10.24507/icicelb.11.08.773

Patil, B., Pattanshetty, A., & Nandyal, S. (2013). Plant Classification using SVM Classifier. *Third International Conference on Computational Intelligence and Information Technology (CIIT)*, 519–523. https://doi.org/10.1049/cp.2013.2639

Pawara, P., Okafor, E., Schomaker, L., & Wiering, M. (2017). Data Augmentation for Plant Classification. *Advanced Concepts for Intelligent Vision Systems. (ACIVS)*, 615–626. https://doi.org/10.1007/978-3-319-70353-4_52

Pawara, P., Okafor, E., Surinta, O., Schomaker, L., & Wiering, M. (2017). Comparing Local Descriptors and Bags of Visual Words to Deep Convolutional Neural Networks for Plant Recognition. *6th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, 479–486. https://doi.org/10.5220/0006196204790486

Peker, M. (2021). Multi-channel Capsule Network Ensemble for Plant Disease Detection. *SN Applied Sciences*, *3*(707), 1–10. https://doi.org/10.1007/s42452-021-04694-2

Perez, L., & Wang, J. (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *ArXiv*, *abs/1712.0*, 1–8. https://doi.org/10.48550/ARXIV.1712.04621

Pettersson, L., & Lundell Johansson, C. (2018). *Ant Colony Optimization - Optimal Number of Ants*.

Pitakaso, R., Almeder, C., Doerner, K. F., & Hartl, R. F. (2007). A MAX-MIN Ant System for Unconstrained Multi-level Lot-sizing Problems. *Computers & Operations Research*, *34*(9), 2533–2552. https://doi.org/https://doi.org/10.1016/j.cor.2005.09.022

Ponnusamy, R., Sathiamoorthy, S., & Manikandan, K. (2017). A Review of Image Classification Approaches and Techniques. *International Journal of Recent Trends in Engineering & Research (IJRTER)*, *03*(03), 1–5. https://doi.org/10.23883/IJRTER.2017.3033.XTS7Z

Poojary, R., Raina, R., & Mondal, A. K. (2020). Effect of Data-augmentation on Fine-tuned CNN Model Performance. *IAES International Journal of Artificial Intelligence (IJ-AI)*, *10*(1), 84–92. https://doi.org/10.11591/ijai.v10.i1.pp84-92

Puangsuwan, T., & Surinta, O. (2021). Leaf Disease Classification Based on Snapshot Ensemble Convolutional Neural Network. *ICIC Express Letters*, *15*(6), 669–680. https://doi.org/10.24507/icicel.15.06.669

Quach, B. M., Dinh, V. C., Pham, N., Huynh, D., & Nguyen, B. T. (2022). Leaf Recognition using Convolutional Neural Networks Based Features. *Multimedia Tools and Applications*, 1–25. https://doi.org/10.1007/s11042-022-13199-y

Rojarath, A., & Songpan, W. (2021). Cost-Sensitive Probability for Weighted Voting in an Ensemble Model for Multi-Class Classification Problems. *Applied Intelligence, 51*, 4908–4932. https://doi.org/https://doi.org/10.1007/s10489-020-02106-3 Cost-sensitive

Ruder, S. (2017). An Overview of Gradient Descent Optimization Algorithms. *ArXiv, 1609.04747*, 1–14. https://doi.org/10.48550/ARXIV.1609.04747

Salman, A., Semwal, A., Bhatt, U., & Thakkar, V. M. (2017). Leaf Classification and Identification using Canny Edge Detector and SVM Classifier. *2017 International Conference on Inventive Systems and Control (ICISC)*, 1–4. https://doi.org/10.1109/ICISC.2017.8068597

Sandler, M, Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

Sharmin, A., Anwar, F., & Motakabber, S. M. A. (2018). A Noble Approach of ACO Algorithm for WSN. *2018 7th International Conference on Computer and Communication Engineering (ICCCE)*, 152–156. https://doi.org/10.1109/ICCCE.2018.8539295

Shorten, C., & Khoshgoftaar, T. M. (2019). A Survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(60), 1–48. https://doi.org/10.1186/s40537-019-0197-0

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv*, *1409.1556*, 1–14. https://doi.org/10.48550/ARXIV.1409.1556

Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., & Batra, N. (2020). PlantDoc: A Dataset for Visual Plant Disease Detection. *The 7th ACM IKDD CoDS and 25th COMAD*, 249–253. https://doi.org/10.1145/3371158.3371196

Sinha, A., & Shekhawat, R. S. (2020). Review of Image Processing Approaches for Detecting Plant Diseases. *The Institution of Engineering and Technology*, *14*(8), 1427–1439. https://doi.org/10.1049/iet-ipr.2018.6210

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image

Classification. *Computational Intelligence and Neuroscience, 2016*, 1–11. https://doi.org/10.1155/2016/3289801

Smith, L. N. (2017). Cyclical Learning Rates for Training Neural Networks. *ArXiv*, *1506.01186*, 1–10.

Sonka, M., Hlavac, V., & Boyle, R. (1993). Image Pre-Processing. In *Image Processing, Analysis and Machine Vision* (pp. 56–111). https://doi.org/10.1007/978-1-4899-3216-7_4

Stützle, T., & Hoos, H. H. (2000). MAX – MIN Ant System. *Future Generation Computer Systems*, *16*, 889–914. https://doi.org/10.1016/s0167-739x(00)00043-1

Sun, S., Cao, Z., Zhu, H., & Zhao, J. (2019). A Survey of Optimization Methods from a Machine Learning Perspective. *ArXiv*, *abs/1906.0*, 1–30. https://doi.org/10.48550/ARXIV.1906.06821

Surinta, O., Schomaker, L., & Wiering, M. (2013). A Comparison of Feature and Pixel-Based Methods for Recognizing Handwritten Bangla Digits. *12th International Conference on Document Analysis and Recognition (ICDAR)*, 165–169. https://doi.org/10.1109/ICDAR.2013.40

Suwais, K., Alheeti, K., & Dosary, D. A. (2022). A Review on Classification Methods for Plants Leaves Recognition. *International Journal of Advanced Computer Science and Applications International (IJACSA)*, *13*(2), 92–100. https://doi.org/10.14569/IJACSA.2022.0130211

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., … Rabinovich, A. (2014). Going Deeper with Convolutions. *ArXiv*, *1409.4842*, 1–12. https://doi.org/10.48550/ARXIV.1409.4842

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *ArXiv*, *abs/1905.1*, 1–11. https://doi.org/10.48550/ARXIV.1905.11946

Turkoglu, M., Yanikoğlu, B., & Hanbay, D. (2021). PlantDiseaseNet : Convolutional Neural Network Ensemble for Plant Disease and Pest Detection. *Signal, Image and Video Processing*, *16*, 301–309. https://doi.org/10.1007/s11760-021-01909-2

Valliammal, N. (2013). *Computer-Aided Plant Identification Through Leaf Recognition using Enhanced Image Processing and Machine Learning Algorithms* (Avinashilingam Deemed University For Women). Retrieved from http://hdl.handle.net/10603/29171

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience.

Vo, A. H., Dang, H. T., Nguyen, B. T., & Pham, V. (2019). Vietnamese Herbal Plant

Recognition using Deep Convolutional Features. *International Journal of Machine Learning and Computing*, *9*(3), 363–367. https://doi.org/10.18178/ijmlc.2019.9.3.811

Wäldchen, J., & Mäder, P. (2018). Plant Species Identification using Computer Vision Techniques: A Systematic Literature Review. *Archives of Computational Methods in Engineering*, *25*(2), 507–543. https://doi.org/10.1007/s11831-016-9206-z

Wäldchen, J., Rzanny, M., Seeland, M., & Mäder, P. (2018). Automated Plant Species Identification—Trends and Future Directions. *PLoS Computational Biology*, *14*(4), 1–19. https://doi.org/10.1371/journal.pcbi.1005993

Wang, X. F., Huang, D. S., Du, J. X., Xu, H., & Heutte, L. (2008). Classification of Plant Leaf Images with Complicated Background. *Applied Mathematics and Computation*, *205*(2), 916–926. https://doi.org/10.1016/j.amc.2008.05.108

Weyland, D. (2015). A Critical Analysis of the Harmony Search Algorithm—How not to Solve Sudoku. *Operations Research Perspectives*, *2*, 97–105. https://doi.org/10.1016/j.orp.2015.04.001

Xie, S., Girshick, R. B., Dollár, P., Tu, Z., & He, K. (2016). Aggregated Residual Transformations for Deep Neural Networks. *ArXiv*, *abs/1611.0*, 1–10. https://doi.org/10.48550/ARXIV.1611.05431

Yan, X., Enhua, X., & Shuangting, L. (2020). Leakage Diagnosis of Water Supply Network Based on ACO-SVM. *2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, 109–112. https://doi.org/10.1109/ICAICE51518.2020.00027

Zawbaa, H. M., Abbass, M., Basha, S. H., Hazman, M., & Hassenian, A. E. (2014). An automatic flower classification approach using machine learning algorithms. *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 895–901. https://doi.org/10.1109/ICACCI.2014.6968612

Zhang, C., Yan, Z., Ma, C., & Xu, X. (2020). Prediction of Direct Normal Irradiance Based on Ensemble Deep Learning Models. *IEEE 3rd International Conference on Electronics Technology (ICET)*, 425–432. https://doi.org/10.1109/ICET49382.2020.9119549

Zhao, M., Song, X., & Xing, S. (2022). Improved Artificial Bee Colony Algorithm with Adaptive Parameter for Numerical Optimization. *Applied Artificial Intelligence*, *36*(1), 525–541. https://doi.org/10.1080/08839514.2021.2008147

Zhao, Z. Q., Ma, L. H., Cheung, Y. ming, Wu, X., Tang, Y., & Chen, C. L. P. (2015).

ApLeaf: An Efficient Android-based Plant Leaf Identification System. *Neurocomputing*, *151*, 1112–1119. https://doi.org/10.1016/j.neucom.2014.02.077

Zöller, M.-A., & Huber, M. F. (2019). Survey on Automated Machine Learning. *ArXiv*, *abs/1904.1*, 409–472. https://doi.org/10.48550/ARXIV.1904.12054

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning Transferable Architectures for Scalable Image Recognition. *ArXiv*, *1707.07012*, 1–14. https://doi.org/10.48550/ARXIV.1707.07012

BIOGRAPHY

NAME	Thipwimon Chompookham
DATE OF BIRTH	24 May 1981
PLACE OF BIRTH	Saraburi
ADDRESS	102 No.20, Kham Tao Pattana Subdistrict, Kantharawichai District, Mahasarakham Province 44150
POSITION	Lecturer
PLACE OF WORK	Faculty of Information Technology, Maha Sarakham Rajabhat University
EDUCATION	2006 Computer Science (B.Sc.) Phranakhon Si Ayutthaya Rajabhat University 2014 Master of Science (M.Sc.) Information Technology, Mahasarakham University 2022 Doctor of Philosophy (Ph.D.) Information Technology, Mahasarakham University
Research output	Chompookham, T., Gonwirat, S., Lata, S., Phiphiphatphaisit, S., & Surinta, O. (2020). Plant Leaf Image Recognition Using Multiple-Grid Based Local Descriptor and Dimensionality Reduction Approach. The 3rd International Conference on Information Science and System (ICISS), 72–77. https://doi.org/10.1145/3388176.3388180 Chompookham, T., & Surinta, O. (2021). Ensemble methods with deep convolutional neural networks for plant leaf recognition. Letters, Icic Express, 15(6), 553–565. https://doi.org/10.24507/icicel.15.06.553