ICIC Express Letters
Volume **15**, Number **6**, June **2021**

ICIC International ©2021 ISSN 1881-803X
pp. 553–565

ENSEMBLE METHODS WITH DEEP CONVOLUTIONAL NEURAL NETWORKS FOR PLANT LEAF RECOGNITION

Thipwimon Chompookham[1] and Olarik Surinta[2,*]

[1]Department of Information Technology
Faculty of Informatics
Mahasarakham University
Maha Sarakham 44150, Thailand
61011261002@msu.ac.th

[2]Multi-Agent Intelligent Simulation Laboratory (MISL)
Department of Information Technology
Faculty of Informatics
Mahasarakham University
Maha Sarakham 44150, Thailand
*Corresponding author: olarik.s@msu.ac.th

Received October 2020; accepted January 2021

Abstract. *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) obtains 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 outperforms 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.*
Keywords: Plant leaf recognition, Convolutional neural network, Ensemble method, Ensemble learning method, Ensemble convolutional neural network

1. **Introduction.** Plants are essential to human life and can be used as food and even medicine [1]. 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

DOI: 10.24507/icicel.15.06.553

plant species [2]. 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 [3, 4, 5, 6]. 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 proposed artificial intelligence systems to stop disease spread and recognize the disease types. Many benchmark plant leaf and plant leaf disease datasets were invented, such as PlantCLEF, Leafsnap, PlantVillage, and PlantDoc [7, 8, 9, 10], to create effective learning models. The plant leaf images in the benchmark datasets were typically collected from natural environments. Hence, the collection of leaf images is more diverse 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 lose 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, central, and northeast Thailand.

2. **Related Work.**

Image processing and machine learning techniques have been proposed to address plant leaf recognition problems. Wang et al. [11] 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 [11, 12]. The feature vector was recognized using machine learning techniques, such as K-nearest neighbour (KNN) and support vector machine (SVM). Chompookham et al. [13] 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 subregion. Principal component analysis (PCA) was used to reduce the dimension of the feature vector. The most correlated variables are given to the SVM classifier.

Convolutional neural network (CNN) methods are currently employed to recognize plant leaf and disease. Atabay [14] invented a new CNN architecture with five layers for plant leaf datasets. This CNN architecture comprises four sets of convolutional layers and one softmax 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 [15] 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. [6] proposed to use AlexNet and GoogLeNet for three plant leaf datasets: AgrilPlant, LeafSnap, and Folio. Two strategies of training, including 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 bags of visual word (BOW) and local descriptor combined with the machine learning techniques (K-nearest neighbor, support vector machine, and multilayer perceptron).

Pawara et al. [16] 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 obtained 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 [17] 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, the two-level architecture, called stacked CNN [4] 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 the 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 [18], unweighted average, and weighted average ensemble methods [19].

The CNN architectures were proposed to address many recognition applications [20, 21]. Also, the recognition performance was enhanced when the ensemble method was combined. In this study, we proposed the framework of the ensemble CNN as follows.

3. **Ensemble Convolutional Neural Networks Framework.** 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 [22]. In this section, we introduce the ensemble convolutional neural networks (CNNs) framework to address the plant leaf recognition problems, as shown in Figure 1. 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 3.1. In the second part, the output probabilities of the CNN models are given to be recognized by the ensemble

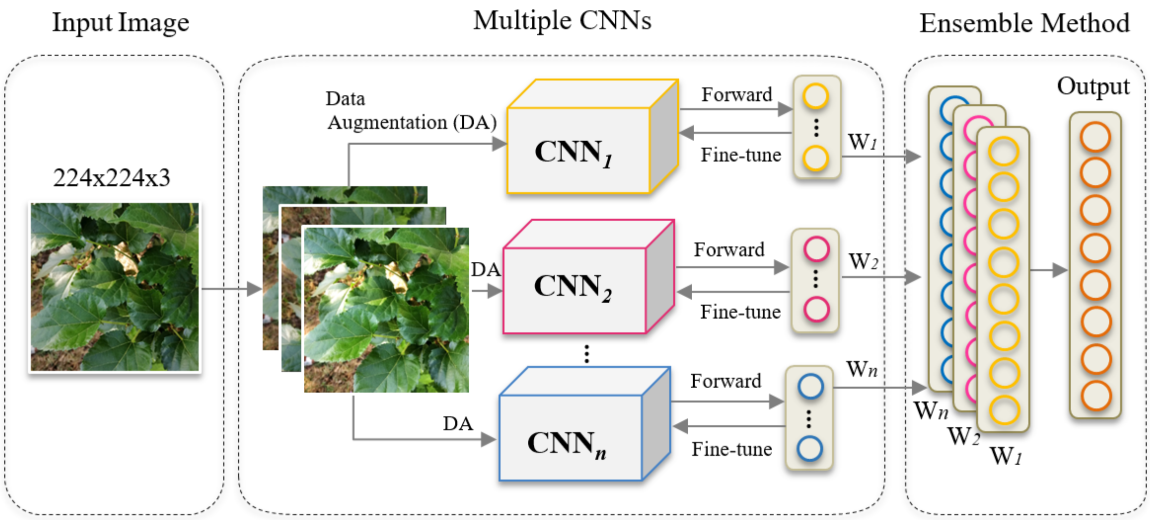

FIGURE 1. The framework of the proposed ensemble CNNs

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 3.2.

3.1. **Multiple convolutional neural networks.** This section briefly describes the CNN that is combined in multiple CNNs (including 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.

3.1.1. *Convolutional neural network architectures.*

MobileNetV1. MobileNetV1 was proposed by Howard et al. [23]. 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 $3 \times 3 \times 3$ to $1 \times 1 \times 3$ convolutions. After that, the depthwise convolutions reduced to 1×1 convolutions, called pointwise convolutions.

MobileNetV2. MobileNetV2 was proposed by Sandler et al. [24], which improved on MobileNetV1 [23]. The MobileNetV2 architecture comprises 11 layers: one fully 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 [25] designed the Xception architecture. This architecture, the extreme version of the inception module, 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 $229 \times 229 \times 3$ 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 2017, Huang et al. [26] invented DenseNet architecture. In this architecture, the knowledge is collected according to the connections from the current layer and in the following layers are combined, 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 affected to 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. [27] 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.

3.1.2. *Optimization algorithms for CNN architectures.* The optimization algorithms were invented to deal with minimizing the objective function [28]. 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 [29]. The SGD optimizer can be computed as

$$\theta = \theta - \eta \cdot \nabla_\theta J\left(\theta; x^i; j^i\right) \tag{1}$$

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

RMSProp. Hinton et al. [30] 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 [29].

$$\begin{gathered} \theta_{t+1} = \theta_t - \frac{n}{\sqrt{E[g^2]_t + \epsilon}} g_t \\ E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma)\, g_t^2 \end{gathered} \tag{2}$$

where $E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma)\, g_t^2$ is squared gradients for each weight, g_t is the gradient of the cost function, and γ is a decay constant. Note that the best values of decay constant and learning rate are 0.9 and 0.001 [28, 29].

3.2. **Ensemble methods.** In this section, the idea of the ensemble method combines with several weights (see Figure 1) that are learned from the CNN models to generate the optimal predictive model. In this section, we mainly emphasize three ensemble methods as follows.

3.2.1. *Unweighted average.* The most common of the ensemble methods is the unweighted average method. In this method, first, the probability values $(w_1, w_2, \ldots, w_n)$, which is the output of the last layer of the CNN models, are calculated using the softmax activation function [31]. Second, we average all the probability values of the CNN models and select 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.

3.2.2. *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 [32]. The unweighted majority vote method is calculated as $p' = \frac{1}{n}\sum_{i=1}^{n} \arg\max(\vec{y})$, where $\arg\max(\vec{y})$ is the highest probability value of weight vector $\vec{y}$ and n is the number of ensemble CNN models.

3.2.3. *Weighted average.* The weighted average method is the extended version of the unweighted average by multiplying the different weight values to the CNN outputs [32]. 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.

4. **Plant Leaf Datasets.** In the section, we introduce the benchmark mulberry leaf dataset. We collected mulberry leaves that grow in Thailand. In this dataset, the mulberry leaf images are diverse in brightness, shadow, and even camera angles because the mulberry leaf images were captured from the natural environment. We provide the mulberry leaf dataset with the aim of the 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.

4.1. **Mulberry leaf dataset.** The mulberry leaf dataset is a collection of 10 cultivars that are taken in the 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).

In this research, the mulberry leaf images were captured from the natural environments, as shown in Figure 2. We recorded the images from different perspectives. There is a shadow that appears in the photo when holding the camera at a low position. However, when shooting from an eye-level position, the resulting image is sharp and the backlit image does not appear. All leaf images are recorded in the JPEG format. The mulberry leaf images were resized into 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),

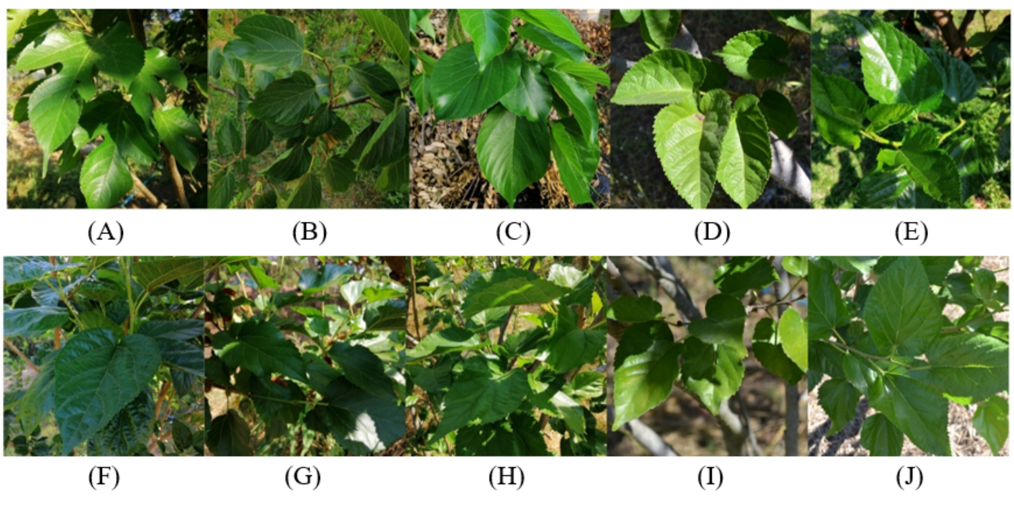

FIGURE 2. Illustration of the ten mulberry leaf cultivars including (A) King Red, (B) King White, (C) Taiwan Maechor, (D) Taiwan Strawberry, (E) Black Austurkey, (F) Black Australia, (G) Chiang Mai 60, (H) Buriram 60, (I) Kamphaeng Saen 42, and (J) Mixed Chiang Mai 60+Buriram 60

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 Black Australia (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.

4.2. **PlantVillage dataset.** The PlantVillage dataset is a collection of plant images proposed by Penn State University [9] 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 detail of these datasets is as follows.

4.2.1. *Tomato leaf disease dataset.* This dataset consists of 10 categories: nine diseased tomato leaves and one healthy leaf [17, 33]. 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 3(A).

4.2.2. *Corn leaf disease dataset.* This dataset contains four classes and has 3,852 images [3, 34]. 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 3(B).

(A) Tomato leaf disease dataset

(B) Corn leaf disease dataset

Figure 3. Examples of leaf disease datasets: (A) Tomato leaf images, including Bacterial Spot, Early Blight, Late Blight, Leaf Mold, Septoria Leaf Spot, Spider Mites Two-spotted Spider Mite, Target Spot, Tomato Mosaic Virus, Tomato Yellow Leaf Curl Virus, and Healthy, respectively; (B) corn leaf images, including Cercospora Leaf Spot Gray Leaf Spot, Common Rust, Northern Leaf Blight, and Healthy (from left to right)

5. Experimental Setup and Results.

5.1. **Mulberry leaf dataset.** In this experiment, the pre-trained models of five convolutional neural networks (CNNs) consisting of MobileNetV1, MobileNetV2, NASNetMobile, DenseNet121, and Xception, are 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 contains 3,719 images and the independent test set includes 1,543 images. The experimental settings used to train the CNN models are as follows: image resolution is 224×224 pixels, the three optimization algorithms are SGD, Adam, and RMSProp, the learning rate is 0.1, 0.01, 0.001, and 0.0001, the batch size is 8, 16, 32, and 64, and the number of iterations is 500 epochs.

Table 1. 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	$\mathbf{98.61 \pm 0.002}$	90.80
Xception	RMSProp	0.0001	8	97.94 ± 0.006	91.0

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 1.

From the results in Table 1, it can be seen that DenseNet121 outperforms other CNN methods with a cross-validation accuracy of 98.61%. Moreover, MobileNetV2 is the best CNN model when applied to the test set. The recognition performance of MobileNetV2 is 91.19%, while the worst recognition performance is 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 obtained 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 can guarantee that these CNN models are not overfitted on the tomato and corn leaf disease datasets when using the same CNN parameters.

In Table 2, we showed 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 is the Xception model when training the model with three data augmentation 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 are MobileNetV2, and NASNetMobile although mobileNetV1 and DenseNet121 models achieved results higher than 90% when mixed with 10 data augmentation techniques (DA6).

Table 2. Performance evaluation of the CNNs and data augmentation techniques on the mulberry leaf dataset

Data augmentation	Test accuracy (%)				
	MobileNetV1	MobileNetV2	NASNetMobile	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

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 2 are applied to training the CNNs as well. 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 Tables 3 and 4.

Table 3. Performance evaluation of the CNNs on the tomato leaf disease dataset

Methods	Model size	Accuracy of data augmentation (%)				
		No DA	DA3	DA4	DA5	DA6
VGG16 [17]	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 4. Performance evaluation of the CNNs on the corn leaf disease dataset

Methods	Model size	Accuracy of data augmentation (%)				
		No DA	DA3	DA4	DA5	DA6
RGB+LinearSVM [34]	N/A	88.00	–	–	–	–
BoF+LinearSVM [3]	N/A	83.70	–	–	–	–
MobileNetV1	25.0 MB	97.92	98.44	99.20	98.44	99.20
MobileNetV2	17.9 MB	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**

From the results in Table 3, it can be seen that Kumar and Vani [17] proposed VGG16 for recognition in the tomato leaf disease dataset and received the 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 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 4, the 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 shows a better result than the previous studies with approximately 10% accuracy.

5.3. **Experiments of the ensemble CNN models on the plant leaf datasets.** As can be seen from Table 1, we decided to use 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 are then used in the decision layer of the ensemble method. We use three ensemble methods to recognize the plant leaf datasets, including the unweighted majority vote, unweighted average, and weighted average methods.

Table 5 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 perform 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 augmentation technique show the best accuracy on mulberry leaf and corn leaf disease datasets. On the other hand, the recognition performance with applying the data augmentation technique is 99.93% on the tomato leaf disease dataset.

Table 5. Performance of the ensemble CNN methods applied on plant leaf datasets

Datasets/DA	Test accuracy (%)					
	Unweighted majority vote		Unweighted average		Weighted average	
	3-EnsCNNs	5-EnsCNNs	3-EnsCNNs	5-EnsCNNs	3-EnsCNNs	5-EnsCNNs
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 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 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

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

6. **Conclusion.** In this paper, we have proposed ensemble convolutional neural network (CNN) architectures to improve recognition performance on the plant leaf datasets. In order to obtain the CNN based models, we first compare five state-of-the-art CNNs: MobileNetV1, MobileNetV2, NASNetMobile, DenseNet121, and Xception. The CNN models are trained with a transfer learning technique and the training sample enlarged using data augmentation techniques. We evaluate five CNN models on the mulberry leaf dataset and two plant leaf disease datasets: tomato and corn. Second, we select 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, are applied as a 5-EnsCNNs. Finally, three ensemble methods: the unweighted majority vote, unweighted average, and weighted average methods, are proposed to classify the output of CNN models. Due to the best experiment, the weighted average method is selected.

With the individual CNN model, the DenseNet121 obtained 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, can 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 obtained high accuracy when compared to the 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 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 obtained 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 contain only one leaf in the image (see Figure 3) while the mulberry leaf images were taken from the natural environment with different perspectives, sunlight conditions, and several leaves appear in the image (Figure 2).

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) [35], AutoAugment [36], and sample paring [37] methods. Another direction for future work is designing new ensemble CNNs. And 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 [38, 39].

REFERENCES

[1] J. X. Du, X. F. Wang and G. J. Zhang, Leaf shape based plant species recognition, *Applied Mathematics and Computation*, vol.185, no.2, pp.883-893, 2007.

[2] A. H. Vo, H. T. Dang, B. T. Nguyen and V.-H. Pham, Vietnamese herbal plant recognition using deep convolutional features, *International Journal of Machine Learning and Computing*, vol.9, no.3, pp.363-367, 2019.

[3] K. R. Aravind, P. Raja, K. V. Mukesh, R. Aniirudh, R. Ashiwin and C. Szczepanski, Disease classification in maize crop using bag of features and multiclass support vector machine, *International Conference on Inventive Systems and Control (ICISC)*, pp.1191-1196, 2018.

[4] V. Mokeev, An ensemble of learning machine models for plant recognition, in *Analysis of Images, Social Networks and Texts. AIST 2019. Communications in Computer and Information Science*, van der Aalst W. et al. (eds.), Kazan, Russia, Springer, 2019.

[5] T. Munisami, M. Ramsurn, S. Kishnah and S. Pudaruth, Plant leaf recognition using shape features and colour histogram with K-nearest neighbour classifiers, *Procedia Computer Science*, vol.58, pp.740-747, 2015.

[6] P. Pawara, E. Okafor, O. Surinta, L. Schomaker and M. Wiering, Comparing local descriptors and bags of visual words to deep convolutional neural networks for plant recognition, *The 6th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, Porto, Portugal, pp.479-486, 2017.

[7] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez and J. V. B. Soares, Leafsnap: A computer vision system for automatic plant species identification, in *Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato and C. Schmid (eds.), Berlin, Heidelberg, Springer, 2012.

[8] H. Goëau, P. Bonnet and A. Joly, LifeCLEF plant identification task 2015, *Conference and Labs of the Evaluation Forum (CLEF)*, Toulouse, France, pp.1-15, 2015.

[9] D. P. Hughes and M. Salathé, An open access repository of images on plant health to enable the development of mobile disease diagnostics, *arXiv.org*, arXiv: abs/1511.08060, 2015.

[10] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat and N. Batra, PlantDoc: A dataset for visual plant disease detection, *The 7th ACM IKDD CoDS and 25th COMAD*, Hyderabad, India, pp.249-253, 2020.

[11] X. F. Wang, D. S. Huang, J. X. Du, H. Xu and L. Heutte, Classification of plant leaf images with complicated background, *Applied Mathematics and Computation*, vol.205, no.2, pp.916-926, 2008.

[12] B. Patil, A. Pattanshetty and S. Nandyal, Plant classification using SVM classifier, *The 3rd International Conference on Computational Intelligence and Information Technology (CIIT)*, Mumbai, India, pp.519-523, 2013.

[13] T. Chompookham, S. Gonwirat, S. Lata, S. Phiphiphatphaisit and O. Surinta, Plant leaf image recognition using multiple-grid based local descriptor and dimensionality reduction approach, *The 3rd International Conference on Information Science and System (ICISS)*, Cambridge, UK, pp.72-77, 2020.

[14] H. A. Atabay, A convolutional neural network with a new architecture applied on leaf classification, *Institute of Integrative Omics and Applied Biotechnology (IIOAB)*, vol.7, pp.326-331, 2016.

[15] W.-S. Jeon and S.-Y. Rhee, Plant leaf recognition using a convolution neural network, *The International Journal of Fuzzy Logic and Intelligent Systems*, vol.17, no.1, pp.26-34, 2017.

[16] P. Pawara, E. Okafor and L. Schomaker, Data augmentation for plant classification, *Advanced Concepts for Intelligent Vision Systems (ACIVS)*, Antwerp, Belgium, pp.615-626, 2017.

[17] A. Kumar and M. Vani, Image based tomato leaf disease detection, *International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kanpur, India, pp.1-6, 2019.

[18] O. Surinta, L. Schomaker and M. Wiering, A comparison of feature and pixel-based methods for recognizing handwritten Bangla digits, *The 12th International Conference on Document Analysis and Recognition (ICDAR)*, Washington, D.C., USA, pp.165-169, 2013.

[19] L. Guo, S. Du, Y. Chi, W. Cui, P. Song, J. Zhu, S. Geng and M. Xu, 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)*, Budapest, Hungary, pp.1-6, 2019.

[20] D. Kreuter, H. Takahashi, Y. Omae, T. Akiduki and Z. Zhang, Classification of human gait acceleration data using convolutional neural networks, *International Journal of Innovative Computing, Information and Control*, vol.16, no.2, pp.609-619, 2020.

[21] S. Park, Y. Suh and J. Lee, Clothing classification using CNN and shopping mall search system, *ICIC Express Letters, Part B: Applications*, vol.11, no.8, pp.773-780, 2020.

[22] C. Zhang, Z. Yan, C. Ma and X. Xu, Prediction of direct normal irradiance based on ensemble deep learning models, *IEEE 3rd International Conference on Electronics Technology (ICET)*, Chengdu, China, pp.425-432, 2020.

[23] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, MobileNets: Efficient convolutional neural networks for mobile vision applications, *arXiv.org*, arXiv: abs/1704.04861, 2017.

[24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, *arXiv.org*, arXiv: abs/1801.04381, 2018.

[25] F. Chollet, Xception: Deep learning with depthwise separable convolutions, *arXiv.org*, arXiv: abs/1610.02357, 2016.

[26] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, Densely connected convolutional networks, *arXiv.org*, arXiv: abs/1608.06993, 2018.

[27] B. Zoph, V. Vasudevan, J. Shlens and Q. V. Le, Learning transferable architectures for scalable image recognition, *arXiv.org*, arXiv: abs/1707.07012, 2018.

[28] P. Li, *Optimization Algorithms for Deep Learning*, http://lipiji.com/docs/li2017optdl.pdf (Accessed: 1 October 2020), pp.1-9, 2017.

[29] S. Ruder, An overview of gradient descent optimization algorithms, *arXiv.org*, arXiv: abs/1609.04747, 2017.

[30] G. Hinton, N. Srivastava and K. Swersky, Lecture 6.5 – RMSProp: Divide the gradient by a running average of its recent magnitude, *COURSERA: Neural Networks for Machine Learning*, 2012.

[31] C. Ju, A. Bibaut and M. van der Laan, The relative performance of ensemble methods with deep convolutional neural networks for image classification, *Journal of Applied Statistics*, vol.45, pp.2800-2818, 2018.

[32] B. Harangi, Skin lesion classification with ensembles of deep convolutional neural networks, *Journal of Biomedical Informatics*, vol.86, pp.25-32, 2018.

[33] H. Durmuş, E. O. Güneş and M. Kırcı, Disease detection on the leaves of the tomato plants by using deep learning, *International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, Fairfax, USA, pp.1-5, 2017.

[34] B. S. Kusumo, A. Heryana, O. Mahendra and H. F. Pardede, Machine learning-based for automatic detection of corn-plant diseases using image processing, *International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*, Tangerang, Indonesia, pp.93-97, 2018.
[35] C. Shorten and T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, *Journal of Big Data*, vol.6, no.1, pp.1-48, 2019.
[36] E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan and Q. V. Le, AutoAugment: Learning augmentation strategies from data, *Conference on Computer Vision and Pattern Recognition (CVPR)*, CA, USA, pp.113-123, 2019.
[37] H. Inoue, Data augmentation by pairing samples for images classification, *arXiv.org*, arXiv: abs/ 1801.02929, 2018.
[38] A. Darwish, Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications, *Future Computing and Informatics Journal*, vol.3, no.2, pp.231-246, 2018.
[39] G. N. Joel and S. M. Priya, 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*, vol.7, pp.56-61, 2018.