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

ICIC International ©2021 ISSN 1881-803X
pp. **669–680**

ENHANCEMENT OF PLANT LEAF DISEASE CLASSIFICATION BASED ON SNAPSHOT ENSEMBLE CONVOLUTIONAL NEURAL NETWORK

Thararat Puangsuwan[1] and Olarik Surinta[2,*]

[1]Faculty of Science and Arts
Burapha University Chanthaburi Campus
Chanthaburi 22170, Thailand
thararat@buu.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. *Plant diseases are one of the most serious issues that can decrease the value and volume of plant goods. It is time-consuming for farmers to discover and identify the disease by observing the leaves of plants, even with specialists scientists and laboratory processes. This study proposed the deep learning approach to address the real-world problems that are contained in the PlantDoc dataset. The deep learning method aims to classify plant leaf disease images from the PlantDoc dataset. First, four state-of-the-art convolutional neural networks (CNNs): VGG16, MobileNetV2, InceptionResNetV2, and DenseNet201, were proposed to enhance the plant leaf disease classification performance. As a result, for the baseline CNN model, DenseNet201 showed better performance with an accuracy of 67.18%, while the second-best CNN model was the InceptionResNetV2 with an accuracy of 61.75%. In addition, the data augmentation techniques (rotation, zoom, brightness, cutout, and mixup) were combined in the training process. The InceptionResNetV2 when combined with the rotation technique obtained an accuracy of 66.02% and outperformed all other CNNs. Importantly, based on our experimental results, the data augmentation techniques with brightness, cutout, and mixup were less satisfactory on the PlantDoc dataset. Second, we proposed the snapshot ensemble to improve the performance of the CNN models. We evaluated the classification performance by applying the snapshot ensemble with 4 and 5 cosine annealing cycles and optimized the learning rate using a stochastic gradient descent algorithm. We also examined the snapshot ensemble with the weighted and unweighted ensemble methods. The experimental results showed that the DenseNet201 when training with the snapshot ensemble method (4-cycle) obtained the accuracy of 69.51%.*

Keywords: Snapshot ensemble method, Cyclical cosine annealing, Convolutional neural network, Deep learning, Data augmentation technique, Plant disease classification

1. **Introduction.** Many countries in the world have placed importance on preserving sufficient numbers of plants and are continually applying greater efforts to increasing the quantity and quality of plants. Not only are plants the only source of oxygen for human, but they are also considered to be a crucial food and products resource. However, the problems of plant diseases nowadays arising for many reasons, without any immediate preventive actions will have an effect on the reduction of both quality and quantity of plant products. The symptoms caused by the plant diseases can appear on their leaves and

DOI: 10.24507/icicel.15.06.669

can be observed by the naked eyes. As the classification of plant disease requires experts and laboratory methods, which are time-consuming and costly, the automated plant leaf disease classification technique [1] is necessary for this challenging task to decrease time spent and labor cost.

The trend in the introduction of deep learning methods is gaining attention and the methods have been applied to various problems. Deep learning is an approach that many researchers are interested in selecting as a solution that can be applied to detection, segmentation, and diagnosis [2, 3]. Moreover, the approach is considered as an effective way to achieve classification tasks [4, 5, 6, 7, 8]. Therefore, there are many studies in different fields using deep learning methods, such as human motion detection, medical treatment, science, and agriculture [9, 10, 11]. This research is bringing about several benefits to these fields.

The problem with plant diseases is one which researchers are trying to use deep learning based models to solve. Some studies have applied deep learning methods to detecting and diagnosing plant disease using a dataset of healthy and diseased plant leaf images [12]. In addition, the deep learning approach is used to automatically estimate the severity of the disease based on images of the affected leaves. Convolutional neural network (CNN) is a deep learning method and is a widely used technique to classify images. The CNN method is increasingly being adapted to plant leaf disease classification [13].

The objective of the study is to enhance the classification of plant leaf disease images from the PlantDoc dataset [14] based on snapshot ensemble CNN techniques. We applied state-of-the-art convolutional neural networks and data augmentation techniques to evaluating the plant disease classification performance on the PlantDoc dataset and achieved the highest record of 69.51% accuracy. The CNN architectures employed in the experiments comprised VGG16, MobileNetV2, InceptionResNetV2, and DenseNet201. The pre-trained models of four CNNs were used to decrease training time and also increase CNN performance. The data augmentation techniques were proposed to reconstruct the new training image while training the CNN model, containing rotation, zoom, brightness, cutout [15], and mixup [16]. Furthermore, we performed experiments with ensemble methods. With the ensemble methods, the output probabilities of the CNNs were provided to calculate the output with the snapshot, weighted, and unweighted ensemble methods. Importantly, the snapshot ensemble method used the stochastic gradient descent to optimize the cost function to the minimum cost by using the cosine annealing cycles to change to the learning rate over training epochs.

This paper is organized in the following way. A literature review in Section 2 describes the literature related to the proposed method. Section 3 presents the theoretical background of the snapshot ensemble convolutional neural networks. Section 4 describes the PlantDoc dataset used in the experiments and explains the experimental results. The conclusion and future directions are presented in Section 5.

2. **Literature Review.** Hidayat et al. [17] implemented CNN architecture to classify diseases in crop plants and evaluated it on a dataset of corn disease images from PlantVillage with 3,854 datasets. Moreover, it was developed to be usable through a mobile application. In many studies, classification of disease often focuses on leaves [13], as they are more visible than other parts of the plant. Militante et al. [18] proposed to use CNN to design a system for the recognition of plant leaf diseases in a wide variety of plants and to use it by combining it with data augmentation techniques to enhance the image dataset. It achieved an accuracy rate of up to 96.5%. Syarief and Setiawan [19] proposed the CNN models to extract the deep feature from the maize leaf disease images. The deep features were then transferred to the machine learning techniques (including support vector machine (SVM), k-nearest neighbors, and decision tree) for classification. The experimental

result showed that AlexNet combined with SVM was the best way to classify maize leaf diseases.

There have been many efforts to enhance CNN for plant leaf disease classification and detection. Alruwaili et al. [7] proposed an effective solution for improving the classifier model based on the deep learning approach. The AlexNet architecture was proposed to generate the model of the plant diseases. When comparing with the other existing methods it achieved 99.11% accuracy. Begum et al. [20] used three CNN architectures: modified MobileNet, Xception, and InceptionResNetV2 to diagnose the leaf diseases in tomato, potato, and bell pepper. The experimental results showed that the MobileNet was the best model with an average accuracy of up to 97.0%. Hu et al. [21] proposed a CNN algorithm for corn leaf disease recognition by using data augmentation for enhancing the training set and transfering learning technique to improve the accuracy of the CNN model. The optimized CNN showed an average accuracy of 97.6% on a subset of the PlantVillage dataset that contained four categories (corn gray leaf spot, corn common rust, corn northern leaf blight, and healthy leaves).

For the snapshot ensemble of the CNN, Nguyen and Pernkopf [22] proposed the new CNN model that was trained on a cosine cycle learning rate schedule to classify the lung sound. The CNN model contained 7 convolution layers with a different number of filters and stride. The input and output shapes of the CNN model were $50 \times N \times 1$ and 4 units, where N is the number of temporal feature frames. For the snapshot ensemble, the multiple output probabilities of the CNN models for each snapshot were then computed using the average weight method. The result showed an accuracy of 78.4% on the ICBHI 2017 dataset. The result of the proposed CNN model combined with the snapshot ensemble method also outperformed the other existing methods. Additionally, Dede et al. [2] proposed to use the snapshot ensemble method to classify aerial scene images. The pretrained CNN architectures, including Inception and DenseNet, were selected to create the CNN models and evaluated the performance on the AID and RESISC45 datasets. They performed experiments by training Inception and DenseNet with the snapshot ensemble method. The results showed that the snapshot ensemble of the Inception outperformed other methods on the RESISC45 dataset. Consequently, the snapshot ensemble of the DenseNet obtained 96.59% accuracy on the AID dataset, which was higher than with the Inception model.

It can be seen that the CNN architectures can be proposed to address the classification problem in several applications, such as human gait, medical treatment, and agriculture. Additionally, the classification performance was improved when we combined the CNN architecture with the ensemble method. In this paper, we then proposed the snapshot ensemble CNN to solve the plant leaf disease classification. The framework of the snapshot ensemble CNN is then described as follows.

3. **Snapshot Ensemble Convolutional Neural Networks.** This section applied snapshot ensemble convolutional neural networks (CNNs) to solving problems in the classification of plant diseases, especially plant leaf diseases. When training the single CNN using the stochastic gradient descent algorithm with a standard learning rate schedule, the network discovers only the global minimum cost. The snapshot ensemble [23] was proposed to address finding different local minimum costs by using a cyclic learning rate schedule. Hence, the best weight model in each cycle is then selected as the snapshot. The optimization algorithm and cyclical cosine annealing proposed in the snapshot ensemble method are described as follows.

3.1. **Stochastic gradient descent optimization algorithm.** The stochastic gradient descent is the optimization algorithm that is usually used in convolutional neural work, called SGD optimizer. The SGD optimizer considers the loss function $l(\hat{y}, y)$ that measures the prediction cost ($\hat{y}$) when the actual class is y. The SGD optimizer allows the loss function to update the weight parameter for each training example x until it progresses to the minimum cost [24, 25, 26], as shown in Figure 1. The equation of the SGD optimizer is defined as

$$\theta_{t+1} = \theta_t - \alpha_t \bigtriangledown_\theta Q(x_t, \theta_t) \tag{1}$$

where θ is a weight parameter, θ_t, $t = 1, 2, \ldots$ is a stochastic process, α is the learning rate, and $\theta_t - \alpha_t \bigtriangledown_\theta Q(x_t, \theta_t)$ is the weight updates of each weight coefficient θ.

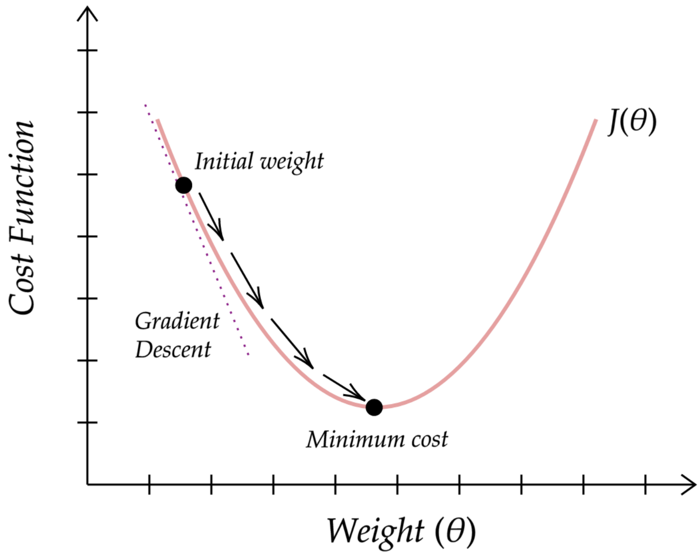


Figure 1. Illustration of the optimal learning rate

3.2. **Cyclical cosine annealing.** Smith [27] invented the novel learning rate, which is the hyper-parameter for training a deep neural network, called cyclical learning rates. The main idea is to train the network with cyclical learning rates instead of fixed learning rates. Consequently, a cosine annealing was proposed by Loshchilov and Hutter [28] to decay the learning rate with a cosine function. In these methods, the learning rate starts with a high value and then decreases from decreases stepwise at random times to discover multiple local minimum values. The cosine annealing learning rate helps the model gather the first local minimum after a few epochs [23]. The cyclical cosine annealing learning rate is defined as

$$\alpha(t) = \frac{\alpha_0}{2}\left(\cos\left(\frac{\pi \bmod (t-1, \lceil T/M \rceil)}{\lceil T/M \rceil}\right)\right) \tag{2}$$

where α is the learning rate. The initial learning rate α_0 starts with a large value and then anneals to a lower value. t is the epoch number, T is the total number of training epochs, and M is the number of cycles that are divided from the training process. The training process using cyclical cosine annealing learning ($M = 4$) is shown in Figure 2.

3.3. **Ensemble methods.** A brief definition of the ensemble methods proposed in the experiment is explained in this section.

Snapshot Ensemble. The learning rate is computed by using cosine annealing (see Equation (2)). At the start of each training cycle, a high learning rate is defined. Hence, the learning rate is decreased in each iteration until it obtains the local minimum according to the training loss. In this process, the best weighted model is taken, called a snapshot [23]. The number of weighted models that are used in the ensemble method depends on training cycles (M). For the ensemble prediction, we average the weighted models of the

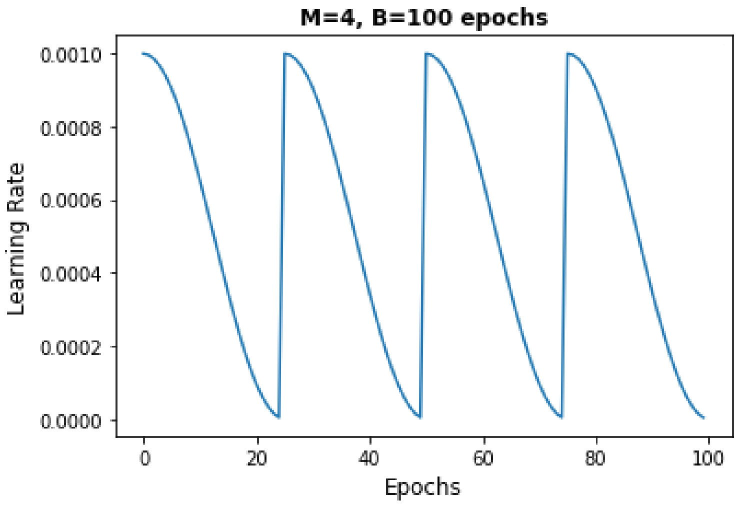


Figure 2. Illustration of the training process using 4 cosine annealing cycles ($M = 4$) and training with 100 epochs ($B = 100$)

last n models, which is calculated using the softmax function. The snapshot ensemble is computed as

$$y_{ensemble} = \frac{1}{n}\sum_{i=1}^{n} softmax(\vec{w}) \tag{3}$$

where $\vec{w}$ is weight vector of the CNN models, n is the number of snapshot models and $n \leq M$.

Unweighted Ensemble. The weighted models $(w_1, w_2, \ldots, w_n)$, which are the output probabilities of the CNNs computed by the softmax function, are averaged [29]. The unweighted ensemble is given by

$$y = \frac{1}{n}\sum_{i=1}^{n} w_i \tag{4}$$

where w is weighted model of the CNNs and n is the number of weighted models.

Weighted Ensemble. The weight parameters α are multiplied by the CNN weighted models (w_i). In the weighted ensemble method, the higher weight parameter is assigned to the CNN weighted model given a higher accuracy. The weighted ensemble is given by

$$y = \frac{1}{n}\sum_{i=1}^{n} \alpha_i w_i \tag{5}$$

where w is weighted model of the CNNs, α is the weight parameters, and n is the number of weighted models.

4. **Experimental Results.** In this section, we described the PlantDoc dataset that was used in the experiments. For the experimental result, we used TensorFlow deep learning framework that was running on Ubuntu version 18. All experiments were evaluated with Intel(R) Core-i9-9900K CPU@3.60Gz ×16, 32GB RAM, and GPU NVIDIA GeForce GTX1080Ti to guarantee consistent performance.

4.1. **PlantDoc dataset.** The PlantDoc dataset was collected taking account of the real-world problems. The dataset provided 2,598 images and 28 classes (18 disease and 10 healthy classes) for experimenting on the classification problem. Then, the raw images called uncropped images were used for evaluating the performance of the deep learning methods [14]. This dataset contains various samples of leaf images, i.e., leaf image records

(A) (B) (C) (D)

FIGURE 3. Sample images of the PlantDoc dataset: (A) Apple leaf, (B) corn leaf blight, (C) soybean leaf, and (D) tomato septoria leaf spot

from the natural environment, leaf images with the white background, other objects in the image, and composites of leaf images in a presentation slide which contained much text, as shown in Figure 3.

4.2. **Experiments of the CNN architectures.** For the CNN based experiments, we compared four state-of-the-art CNN architectures: VGG16, InceptionResNetV2, MobileNetV2, and DenseNet201 on the PlantDoc dataset. The PlantDoc dataset was split into 80% for the training set (2,078 images) and 20% for the test set (520 images). We followed a default input layer for all CNN architectures (see Table 1), except only the output layer. We selected the output layer as 28 units, according to the category of the PlantDoc dataset.

TABLE 1. Classification performances (training time, accuracy, and F1-score) for the state-of-the-art CNN architectures on the PlantDoc dataset

CNN architectures	Training time	Batch size	Accuracy (%)	F1-score
VGG16 [14] Image size = 100×100 Pre-trained weights = ImageNet	N/A	N/A	13.74	0.12
VGG16 [14] Image size = 100×100 Pre-trained weights = ImageNet+PVD	N/A	N/A	29.73	0.28
VGG16 Image size = 224×224 Pre-trained weights = ImageNet	21 min	16	51.07	0.56
MobileNetV2 Image size = 224×224 Pre-trained weights = ImageNet	18 min	32	59.42	0.61
InceptionResNetV2 Image size = 299×299 Pre-trained weights = ImageNet	1 h 15 min	32	61.75	0.61
DenseNet201 Image size = 224×224 Pre-trained weights = ImageNet	1 h 4 min	8	**67.18**	0.67

Furthermore, to compare the experimental result, we first selected the VGG16 model with an image size of 224×224 pixels and compared it with VGG16 with an image size of 100×100 pixels [14]. Second, MobileNetV2 architecture, which improved the performance of MobileNet architecture and provided a small model size, was examined. Third, we experimented on the very deep CNN models of DenseNet201 (201 layers) and InceptionResNetV2 (572 layers). We experimented with stochastic gradient descent (SGD) optimization for training the CNN networks. We set the parameters as follows: momentum = 0.9, learning rate = 0.001, training epoch = 100, and batch sizes were 8, 16, 32, 64, and 128 images per batch. The accuracy and F1-score were reported.

As shown in Table 1, Singh et al. [14] presented the experimental results of the VGG16 on the PlantDoc dataset by using an image size of 100×100 pixels. In their experiments, the pre-trained model trained on the ImageNet and PlantVillage (PVD) datasets was used. The best accuracy was only 29.73% when using the ImageNet+PVD as a pre-trained weight. However, we proposed to use VGG16, InceptionResNetV2, MobileNetV2, and DenseNet201 and used the same setting as Singh et al. [14], except for the PVD pre-trained weight, size of the images, and batch size. The experiments of the VGG16 using parameter settings for image size of 224×224 pixels, batch size of 16, obtained an accuracy of 51.07%. The accuracy was higher (around 20%) than Singh et al. [14]. Additionally, DenseNet201 outperformed other CNN architectures on the PlantDoc dataset with an accuracy of 67.18% and the F1-score with 0.67.

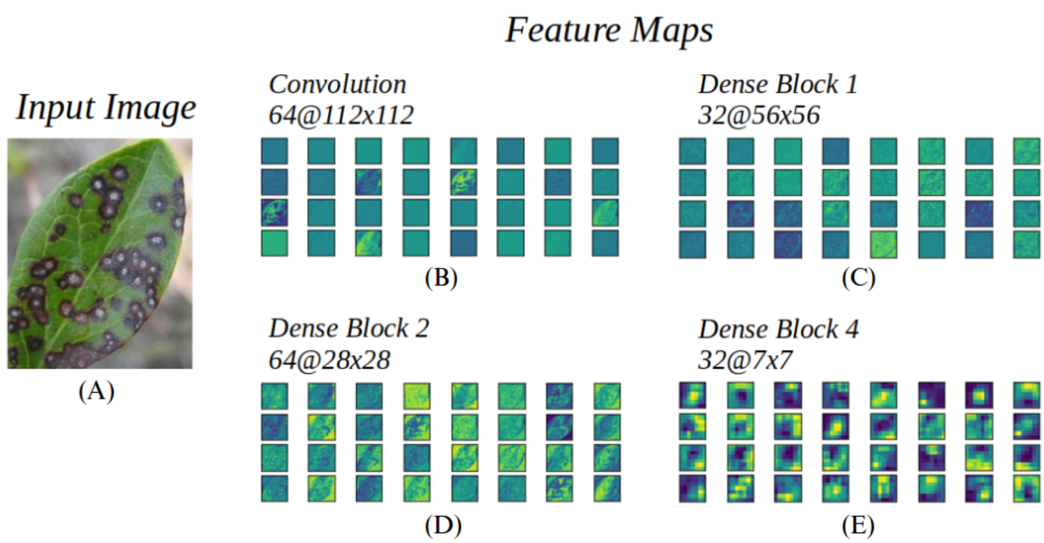


Figure 4. The feature maps visualization of the DenseNet201 architecture: (A) Input image, (B) first convolution layer, the output from (C) dense block layer 1, (D) dense block layer 2, and (E) dense block layer 4

Figure 4 visualizes the feature map of the DenseNet201 architecture, which is the output of the convolutional layers corresponding to the leaf image. The Grad-CAM images that localize class-discriminative regions [30] are shown in Figure 5. Note that the red regions indicate class-discriminative regions for the CNN network.

Table 2 shows the performances obtained from the CNN architectures combined with different data augmentation techniques: rotation, zoom, brightness, cutout, and mixup. The experimental results showed that the InceptionResNetV2 combined with the rotation method outperformed all CNN models. For the data augmentation techniques, the CNN model combined with the rotation and zoom techniques outperformed the CNN model combined brightness, cutout, and mixup techniques on the PlantDoc dataset. The results showed that the InceptionResNetV2 combined with the rotation method achieved the highest accuracy of 66.02%. On the other hand, the brightness, cutout, and mixup achieved extremely low performance between 3-10%. Consequently, based on our experiments on the PlantDoc dataset, the rotation techniques were used in the following experiments.

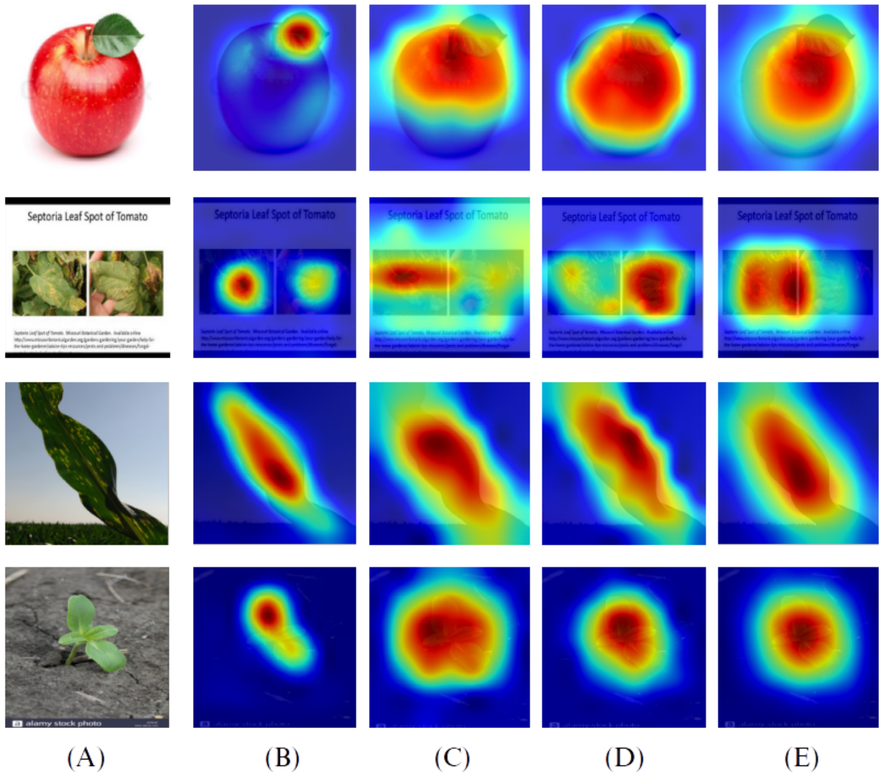


FIGURE 5. (color online) Illustration of (A) PlantDoc images and the Grad-CAM visualization of the (B) VGG16, (C) MobileNetV2, (D) Inception-ResNetV2, and (E) DenseNet201 models that localize class-discriminative regions for PlantDoc images

TABLE 2. Classification performances for the CNN architectures combined with different data augmentation techniques

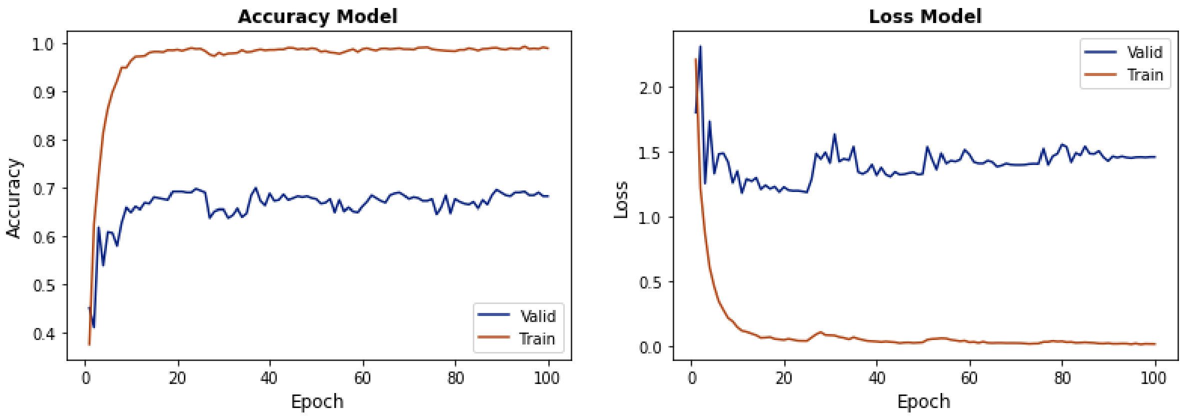
Data augmentation		Accuracy of CNN architectures (%)			
Techniques	Parameters	InceptionResNetV2	MobileNetV2	DenseNet201	VGG16
Rotation	$[-30, 30]$	**66.02**	61.55	58.83	55.92
Zoom	$[0.5, 1.0]$	65.05	59.61	62.14	55.87
Brightness	$[0.2, 1.0]$	5.63	2.52	6.99	10.12
Cutout	0.2	4.27	10.10	4.47	10.10
Mixup	0.4	4.66	4.27	3.50	10.21

4.3. **Results of the snapshot ensemble CNN.** Based on the experiment results as shown in Table 2, we then selected the rotation technique for the data augmentation technique. For the CNN models, we proposed to use four CNN architectures, including VGG16, MobileNetV2, InceptionResNetV2, and DenseNet201. In these experiments, we compared the snapshot ensemble using different cosine annealing cycles. The experimental results were shown in Table 3.

Table 3 presents the classification results using the snapshot ensemble with four and five cosine annealing cycles. For four cosine annealing cycles, first, we trained the CNN models without using the data augmentation technique. The result showed that the DenseNet201 outperformed other CNN models with an accuracy of 69.32%. In comparison, the VGG16 achieved an accuracy of only 59.91%. Second, we evaluated the data augmentation technique using rotation. The DenseNet201 achieved an accuracy of 69.51% and outperformed all CNN models. The InceptionResNetV2 model required the most computation time with 1 hour and 43 minutes. However, the DenseNet201 model

Table 3. Evaluation of the classification results using the snapshot ensemble with 4 and 5 cosine annealing cycles

CNN architectures	Training time	4 cycles		5 cycles	
		No. of models	Accuracy (%)	No. of models	Accuracy (%)
Without data augmentation					
VGG16	27 min	3	59.91	3	60.00
MobileNetV2	17 min	1	62.72	3	60.39
InceptionResNetV2	1 h 42 min	3	66.99	4	67.38
DenseNet201	1 h 4 min	3	**69.32**	5	**67.38**
With rotation technique					
VGG16	28 min	3	60.78	2	59.81
MobileNetV2	18 min	4	62.72	4	62.85
InceptionResNetV2	1 h 43 min	1	66.99	3	66.99
DenseNet201	1 h 4 min	4	**69.51**	5	**68.54**



(A) Accuracy model (B) Loss model

Figure 6. Illustration of (A) accuracy and (B) loss models for the snapshot ensemble with four cosine annealing cycles. Note that we trained the DenseNet201 model combined with the rotation data augmentation using a momentum of 0.9 and a learning rate between 0-0.001.

also required more computation time. It spent 1 hour and 4 minutes of training, while VGG16 and MobileNetV2 required the computation time of only 28 and 18 minutes, respectively. Furthermore, based on our experiments, the data augmentation technique, especially the rotation technique, enhances the performance of the snapshot ensemble method CNN.

For the evaluation of the snapshot ensemble with five cosine annealing cycles, undoubtedly, the DenseNet201 achieved the best accuracy of 68.54% when training with data augmentation and 67.38% when training without data augmentation techniques. We concluded that the snapshot ensemble trained with the DenseNet201 model showed the best accuracy on the PlantDoc dataset.

Figure 6 shows the accuracy and the loss model for the snapshot ensemble with four cosine annealing cycles. We trained the DenseNet201 model combined with the rotation method. Figure 6(B) shows the loss values of the snapshot ensemble using four cosine annealing cycles. We trained the CNN model with 100 epochs to avoid overfitting, which means one cycle for every 25 epochs. In each cycle, the lowest loss model was selected, called a snapshot. In this experiment, we obtained four DenseNet201 models. We then used these CNN models in the snapshot ensemble.

4.4. **Comparison with other ensemble methods.** In this experiment, we compared three ensemble methods: snapshot ensemble, weighted ensemble, and unweighted ensemble. We combined three CNN models for weighted and unweighted ensemble methods, including InceptionResNetV2, MobileNetV2, and DenseNet201. The output probabilities of the CNN models were then transferred to the ensemble methods. For the weighted ensemble method, the output probabilities of the CNN models were then computed with the weight of InceptionResNetV2 = 0.5, MobileNetV2 = 0.2, and DenseNet201 = 0.3. For the snapshot, we selected the snapshot model of 4 cosine annealing cycles.

Table 4. Classification accuracies of three ensemble methods on the PlantDoc dataset

Ensemble methods	Accuracy (%)
Unweighted ensemble	68.54
Weighted ensemble	69.32
Snapshot ensemble ($M = 4$)	**69.51**

Table 4 shows that the snapshot ensemble method with the DenseNet201 model had the best performance with an accuracy of 69.51%. The snapshot ensemble method was slightly better than the weighted ensemble method. In comparison, the unweighted ensemble method achieved a rather worse performance.

5. **Conclusions and Future Directions.** In this paper, we evaluated the deep learning method on the PlantDoc dataset. We first proposed to use state-of-the-art CNN models: VGG16, MobileNetV2, InceptionResNetV2, and DenseNet201 to classify the plant leaf and plant leaf disease images. The pre-trained CNN models were employed due to reduced computation time and because they increased the accuracy performance. In the experiments, we combined the data augmentation techniques, including rotation, zoom, brightness, cutout, and mixup, while training the CNN models. The main idea was to discover baseline CNN models and the best setting (momentum and batch size) of each CNN model. Second, we trained four CNN models based on the snapshot ensemble method with 4 and 5 cosine annealing cycles. Finally, three ensemble methods: snapshot ensemble, weighted ensemble, and unweighted ensemble, were compared.

The results showed that, DenseNet201 model is the best model when training without applying the data augmentation technique. However, when applying the data augmentation techniques, the InceptionResNetV2 combined with the rotation technique outperformed MobileNetV2 and DenseNet201 models. However, the VGG16 model showed the lowest performance, with an accuracy of 55.92%. Surprisingly, the data augmentation techniques that included brightness, cutout, and mixup, when combined with the CNN models performed extremely poorly, with an accuracy of approximately 5-10%. For the snapshot ensemble, we trained the CNN models with only 100 epochs. We experimented with 4 and 5 cosine annealing cycles and used stochastic gradient descent (SGD) as the optimization algorithm. The results showed that the snapshot ensemble (4-cycle) trained with the DenseNet201 model outperformed all CNN models. The accuracy of the snapshot ensemble CNN model was 69.51%. Additionally, we then compared the snapshot ensemble method with weighted and unweighted ensemble methods. The snapshot ensemble method showed the best performance on the PlantDoc dataset.

As a future direction, it is important to realize that, the performance of the snapshot ensemble method is not high enough (∼69%). We will discover the learning rate that can enhance the performance of the snapshot ensemble method. Then, we still need to improve the performance of the plant leaf recognition system by offering the meta-learner method. Also, we need detection or segmentation algorithms to detect and segment the leaves in the image before the classification process, such as YOLO and Mark R-CNN [31, 32]. Moreover, ensemble learning and meta-learning are other directions that need investigation [33, 34].

Acknowledgment. This research project is financially supported by Mahasarakham University.

REFERENCES

[1] L. Malathi, P. Yogashree and A. Thamaraiselvi, Automatic detection and classification of plant leaf disease using deep convolutional neural network, *Journal of Xi'an University of Architecture & Technology*, vol.12, no.5, pp.674-689, 2020.

[2] M. A. Dede, E. Aptoula and Y. Genc, Deep network ensembles for aerial scene classification, *IEEE Geoscience and Remote Sensing Letters*, vol.16, no.5, pp.732-735, 2019.

[3] S. A. Oliveira, B. Seguin and F. Kaplan, dhSegment: A generic deep-learning approach for document segmentation, *The 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Niagara Falls, NY, USA, pp.7-12, 2018.

[4] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk and D. Stefanovic, Deep neural networks based recognition of plant diseases by leaf image classification, *Computational Intelligence and Neuroscience*, 2016.

[5] W. Liu, M. Zhang, Z. Luo and Y. Cai, An ensemble deep learning method for vehicle type classification on visual traffic surveillance sensors, *IEEE Access*, vol.5, pp.24417-24425, 2017.

[6] S. Wang and H. Chen, A novel deep learning method for the classification of power quality disturbances using deep convolutional neural network, *Applied Energy*, vol.235, pp.1126-1140, 2019.

[7] M. Alruwaili, S. A. El-Ghany and A. Shehab, An enhanced plant disease classifier model based on deep learning techniques, *International Journal of Engineering and Advanced Technology*, vol.9, no.1, pp.7159-7164, 2019.

[8] 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.

[9] M. Bakator and D. Radosav, Deep learning and medical diagnosis: A review of literature, *Multimodal Technologies and Interaction*, vol.2, no.3, 2018.

[10] A. Kamilaris and F. X. Prenafeta-Boldú, Deep learning in agriculture: A survey, *Computers and Electronics in Agriculture*, vol.147, pp.70-90, 2018.

[11] Y. Omae, M. Mori, T. Akiduki and H. Takahashi, A novel deep learning optimization algorithm for human motions anomaly detection, *International Journal of Innovative Computing, Information and Control*, vol.15, no.1, pp.199-208, 2019.

[12] K. P. Ferentinos, Deep learning models for plant disease detection and diagnosis, *Computers and Electronics in Agriculture*, vol.145, pp.311-318, 2018.

[13] M. Nagaraju and P. Chawla, Systematic review of deep learning techniques in plant disease detection, *International Journal of System Assurance Engineering and Management*, vol.11, pp.547-560, 2020.

[14] D. Singh, N. Jain and P. Kayal, PlantDoc: A dataset for visual plant disease detection, *The ACM India Joint International Conference on Data Science and Management of Data (CoDS-COMAD)*, Hyderabad, India, 2019.

[15] T. DeVries and G. W. Taylor, Improved regularization of convolutional neural networks with cutout, *arXiv.org*, arXiv: 1708.04552, 2017.

[16] H. Zhang, M. Cisse, Y. N. Dauphin and D. Lopez-Paz, mixup: Beyond empirical risk minimization, *The 6th International Conference on Learning Representations (ICLR)*, Vancouver, Canada, 2018.

[17] A. Hidayat, U. Darusalam and Irmawati, Detection of disease on corn plants using convolutional neural network methods, *Journal of Computer Sciences and Information*, vol.12, no.1, pp.51-56, 2019.

[18] S. V. Militante, B. D. Gerardo and N. V. Dionisio, Plant leaf detection and disease recognition using deep learning, *IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, Yunlin, Taiwan, 2019.

[19] M. Syarief and W. Setiawan, Convolutional neural network for maize leaf disease image classification, *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol.8, no.3, pp.1376-1381, 2020.

[20] A. S. Begum, S. Savitha, S. Shahila and S. Sharmila, Diagnosis of leaf disease using enhanced convolutional neural network, *International Journal of Innovative Research in Applied Sciences and Engineering*, vol.3, no.12, pp.579-586, 2020.
[21] R. Hu, S. Zhang, P. Wang, G. Xu, D. Wang and Y. Qian, The identification of corn leaf diseases based on transfer learning and data augmentation, *The 3rd International Conference on Computer Science and Software Engineering (CSSE)*, Beijing, China, 2020.
[22] T. Nguyen and F. Pernkopf, Lung sound classification using snapshot ensemble of convolutional neural networks, *The 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Montreal, QC, Canada, pp.760-763, 2020.
[23] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft and K. Q. Weinberger, Snapshot ensembles: Train 1, get M for free, *The 5th International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.
[24] L. Bottou, Large-scale machine learning with stochastic gradient descent, *The 19th International Conference on Computational Statistics (COMPSTAT)*, Paris, France, pp.177-186, 2010.
[25] L. Bottou, Stochastic gradient descent tricks, in *Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science*, G. Montavon, G. B. Orr and K. R. Müller (eds.), Berlin, Heidelberg, Springer, 2012.
[26] S. Ruder, An overview of gradient descent optimization algorithms, *arXiv.org*, arXiv: 1609.04747, 2017.
[27] L. N. Smith, Cyclical learning rates for training neural networks, *arXiv.org*, arXiv: 1506.01186, 2017.
[28] I. Loshchilov and F. Hutter, SGDR: Stochastic gradient descent with warm restarts, *The 5th International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.
[29] 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, no.15, pp.2800-2818, 2018.
[30] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, *arXiv.org*, arXiv: 1610.02391, 2019.
[31] J. Redmon and A. Farhai, YOLOv3: An incremental improvement, *arXiv.org*, arXiv: 1804.02767, 2018.
[32] K. He, G. Gkioxari, P. Dollár and R. Girshick, Mask R-CNN, *arXiv.org*, arXiv: 1703.06870, 2018.
[33] X. Liu, X. Wang and S. Matwin, Interpretable deep convolutional neural networks via meta-learning, *International Joint Conference on Neural Networks*, Rio de Janeiro, Brazil, 2018.
[34] M. Khosla, K. Jamison, A. Kuceyeski and M. R. Sabuncu, Ensemble learning with 3D convolutional neural networks for functional connectome-based prediction, *NeuroImage*, vol.199, pp.651-662, 2019.