ICIC Express Letters ICIC International ©2022 ISSN 1881-803X
Volume **16**, Number **5**, May **2022** pp. **521–528**

STACKING ENSEMBLE OF LIGHTWEIGHT CONVOLUTIONAL NEURAL NETWORKS FOR PLANT LEAF DISEASE RECOGNITION

Prem Enkvetchakul and Olarik Surinta*

Multi-agent Intelligent Simulation Laboratory (MISL)
Department of Information Technology
Faculty of Informatics
Mahasarakham University
Khamriang Sub-District, Kantarawichai District, Mahasarakham 44150, Thailand
61011261006@msu.ac.th; *Corresponding author: olarik.s@msu.ac.th

Received September 2021; accepted November 2021

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

1. **Introduction.**

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

More recently, advances in computer vision techniques have increased efficiency in monitoring and recognition in the agricultural domain [3], such as by detecting plant diseases, recognizing the types of the disease, and counting the number of plants. Further, deep learning methods are used in a large number of agricultural applications. Militante et al. [3] proposed computer vision and deep learning techniques for detecting and diagnosing diseases in plants. The proposed systems can take plant images using a camera and recognize diverse plant disease types. Zhong and Zhao [1] studied the significance of the deep learning method based on convolutional neural network (CNN) architecture to identify the diseases that appeared on apple leaves.

DOI: 10.24507/icicel.16.05.521

The deep learning methods were performed to improve the recognition of plant leaf disease images. However, using only a single deep learning model may not be sufficient to increase the accuracy performance of the plant leaf recognition systems. Furthermore, using ensemble learning with multiple deep learning models can reduce the variance of the recognition errors and improve plant leaf recognition systems [4].

Ensemble learning methods have been proposed in many applications. Khanramaki et al. [5] proposed ensemble CNNs to recognize three common citrus pests: citrus leafminer, sooty mold, and Pulvinaria. For the single deep learning, it achieved an accuracy of 96.05% with the Resnet50 architecture. The ensemble learning models provided an accuracy of 99.04%. Mahmoud and Yaroshchak [4] proposed a bagging ensemble to classify diabetic retinopathy images in a database containing 2,781 pictures. First, the training set was randomly selected for three subsets. Second, the subsets were sent to learning using three different CNN architectures. For the ensemble learning method, finally, the weighted average was used. The result showed that the bagging ensemble with three InceptionV3 models obtained an accuracy of 87.2%.

In [6], the stacked CNN was proposed to diagnose COVID-19 disease from X-ray images. Two CNN models, including the fine-tuning of VGG19 and CovNet30, were proposed to learn from the chest X-ray images. The outputs of the CNN models were stacked and a logistic regression classifier was applied to classifying three classes of COVID-19. It performed with an accuracy of 92.47% on the chest X-ray images of the COVID-19 dataset. Chompookham and Surinta [7] invented ensemble CNN architectures to improve plant leaf classification performance. In their method, five CNN architectures were trained on plant leaf images to create robust CNN models. After that, the three best CNNs models were then combined, the output probabilities of each CNN model were assigned to classify using the ensemble methods: unweighted majority vote, unweighted average, and weighted average. The best ensemble method used in this experiment was the weighted average method.

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

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

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

2. **Proposed Stacking Ensemble of Convolutional Neural Networks.** This section introduces the stacking ensemble of CNNs to recognize the plant leaf disease images, as shown in Figure 1.

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

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

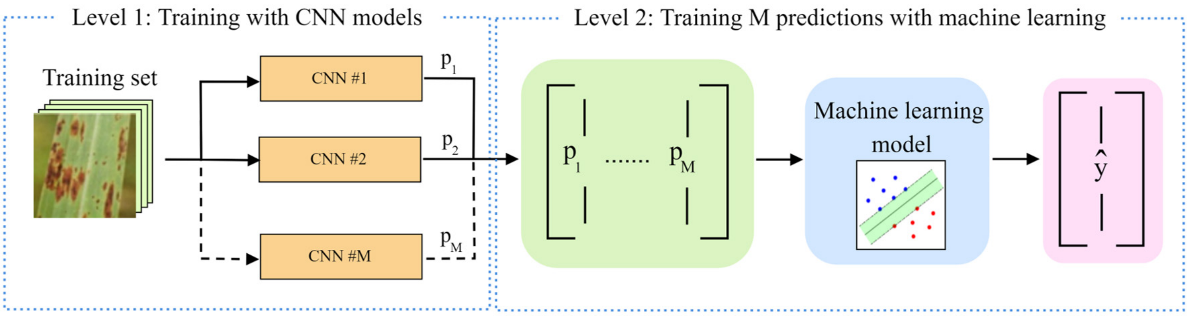

FIGURE 1. Illustration of the proposed stacking ensemble of lightweight CNNs

2.1. Convolutional neural network (CNN) architectures.

InceptionResNetV2. Szegedy et al. [8] proposed a new network architecture that contained the concept of Inception architecture and residual Inception blocks, called InceptionResNetV2. The Inception networks were designed as a tuning network based on InceptionV4, and they were allowed to change the number of filters in the several layers. The Inception block was designed to add the filter-expansion layer for the residual Inception blocks. Hence a 1×1 convolution layer without activation function was used for scaling up the filter dimension.

NASNetMobile. Zoph et al. [9] invented a neural architecture search network (NASNet) to address the expensive computation time while training on the large dataset. First, the NASNet architecture was proposed to search for an optimal architecture building block on a small dataset using reinforcement learning. Second, the building blocks were transferred to learn on a large dataset. The NASNet architecture consisted of two cells: a normal cell and a reduction cell. It was easy to build the NASNet because the normal and reduction cells were stacked and repeated many times. The last layer was the normal cell, followed by the softmax function. In addition, to create the NASNetMobile model, the size of the normal and reduction cells and the number of filters were decreased. The parameters of the NASNetMobile are smaller than the NASNet approximately nine times.

MobileNetV2. MobileNetV2 was designed by Sandler et al. [10] in 2018. It was the extended version of the MobileNetV1. MobileNetV2 contained three main layers: depthwise separable convolutions, linear bottlenecks, and inverted residuals. These layers performed to reduce the number of parameters and computation time when compared with MobileNetV1. In addition, MobileNetV2 was trained using the ReLU6 activation function, allowing it to learn complex patterns in the input data.

EfficientNetB1. EfficientNet was proposed by Tan and Le [11]. It involved scaling the network using four methods: width, depth, resolution, and compound scaling. It was comfortable to scale up a baseline CNN to any purpose resource limitations. Our experiment proposed using EfficientNetB1 to classify the plant leaf datasets because it had parameters with 7.8M. The parameters of EfficientNetB1 were fewer than the DenseNet-169, Xception, Inception-v3, and even ResNet-50.

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

3. **Experimental Setting and Results.** In the experimental setting, we used the TensorFlow library running on Ubuntu operating system version 18. All experiments were evaluated with Intel(R) Core-i5, 2320 CPU @ 3.00GHz, 16GB RAM, and GPU NVIDIA GeForce GT 1060Ti.

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

3.1. **Plant leaf disease datasets.** We experimented on 3 plant leaf disease datasets as follows.

PlantDoc dataset. The PlantDoc dataset contained 2,567 images from 13 plant species collected from the Internet. It included 27 classes of plant leaf disease and of healthy leaf [12]. Examples of the PlantDoc dataset are shown in Figure 2(A).

Crop-PlantDoc dataset. The Crop-PlantDoc dataset is the extended version of the PlantDoc dataset. Singh et al. [12] also provided the ground truth of all images intending to crop all leaves, as shown in Figure 2(B). After cropping all the leaves, the Crop-PlantDoc dataset contained 8,883 images.

iCassava2019 dataset. The iCassava2019 dataset was published on the Kaggle website. It contained 5,656 images of five cassava leaf states, including four types of disease and one healthy type, as shown in Figure 2(C). All cassava leaf disease images were taken from farmers in Uganda and verified by the experts of the National Crops Resources Research Institute (NaCRRI) and AI lab in Makerere University, Kampala [13].

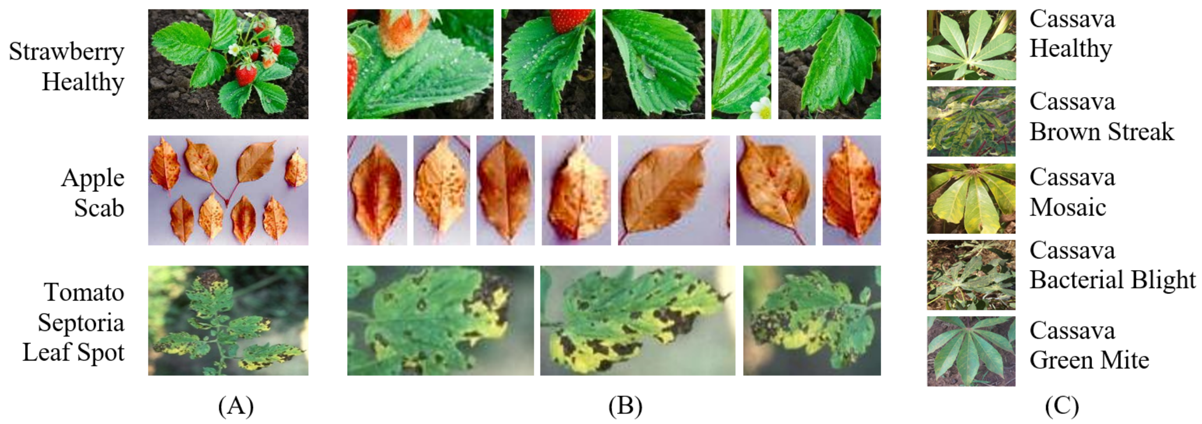

FIGURE 2. Illustration of three plant leaf disease datasets consisting of (A) PlantDoc, (B) Crop-PlantDoc, and (C) iCassava2019

3.2. **Experiments on lightweight convolutional neural networks.** In this experiment, the pre-trained models of four CNNs consisting of InceptionResNetV2, NASNet-Mobile, MobileNetV2, and EfficientNetB1, were trained on the plant leaf disease datasets. The data augmentation techniques [14], including rotation, shift, zoom, and horizontal flip, were combined in this study. In order to determine the average accuracy and standard deviation on the validation set, we randomly selected the training and validation tests and evaluated them ten times.

The experimental data reported in Table 1 indicated that the EfficientNetB1 model had the most significant performance on three plant leaf disease datasets: PlantDoc, Crop-PlantDoc, and iCassava2019. In addition, the InceptionResNetV2 model had the second-best performance on the Crop-PlantDoc and iCassava2019 datasets.

Table 1. Performance evaluation of the lightweight CNNs and data augmentation techniques on plant leaf disease datasets

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

3.3. **Experiments on ensemble learning methods.** We examined the performance of ensemble CNNs from two to ten models to find the optimal numbers of the model.

Table 2 provides accurate results, testing times, and numbers of CNN models for recognition on three plant leaf disease datasets. The experimental results show that the ensemble CNNs with both unweighted majority vote and unweighted average methods performed consistently better than did single CNN. Furthermore, the unweighted average method slightly outperformed the unweighted majority vote. Consequently, the EfficientNetB1

Table 2. Performances of the ensemble learning methods and lightweight CNNs

CNN architectures	**Ensemble methods**	**Evaluation metrics**	**PlantDoc**	**Crop-PlantDoc**	**iCassava2019**
EfficientNetB1	Unweighted majority vote	Accuracy (%)	70.04	86.21	91.34
		Testing time (sec.)	0.12	0.20	0.71
		No. of CNN models	2	8	10
	Unweighted average	Accuracy (%)	**70.82**	**90.55**	**91.61**
		Testing time (sec.)	0.13	0.20	0.71
		No. of CNN models	2	8	10
MobileNetV2	Unweighted majority vote	Accuracy (%)	66.73	85.20	88.34
		Testing time (sec.)	0.29	0.17	0.34
		No. of CNN models	7	9	8
	Unweighted average	Accuracy (%)	67.15	85.20	88.87
		Testing time (sec.)	0.41	0.17	0.38
		No. of CNN models	10	9	9
NASNetMobile	Unweighted majority vote	Accuracy (%)	68.68	88.35	87.81
		Testing time (sec.)	1.29	0.52	1.56
		No. of CNN models	8	9	9
	Unweighted average	Accuracy (%)	68.68	88.69	88.07
		Testing time (sec.)	1.45	0.58	1.56
		No. of CNN models	9	10	9
InceptionResNetV2	Unweighted majority vote	Accuracy (%)	68.87	80.19	90.28
		Testing time (sec.)	1.20	0.46	1.94
		No. of CNN models	7	2	9
	Unweighted average	Accuracy (%)	69.46	80.30	90.64
		Testing time (sec.)	1.20	0.46	1.94
		No. of CNN models	7	2	9

still significantly outperformed other CNNs on all datasets. Surprisingly, the ensemble CNNs combined with two EfficientNetB1 models achieved an accuracy of 70.82% on the PlantDoc dataset. It increased the accuracy of one EfficientNetB1 model by approximately 2%.

3.4. **Experiments on stacking ensemble learning method.** The stacked output probabilities of CNN models were trained using the machine learning methods: LR, SVM, KNN, RF, and LSTM. First, we fine-tuned the hyperparameters of each classifier. The hyperparameters applied to each classifier were as follows. SVM, C = 1, gamma = 0.1, kernel = RBF; KNN, K = 19, distance value = Euclidean, and weight = uniform; RF, estimators = 800, max depth = 30, min samples leaf = 4, min samples split = 10, min features = auto, and bootstrap = true; LSTM, 1 layer with 100 neurons, batch size = 64, optimizer = Adam, epochs = 200. We examined the performance of each classifier with a combination of two to ten CNN models.

Table 3 shows the experimental results of the stacking ensemble learning method. Notably, EfficientNetB1 could be combined with all machine learning techniques and achieved high accuracy. The experiments show that the EfficientNetB1 outperformed other CNN architectures on two plant leaf disease datasets: Crop-PlantDoc and iCassava2019. Consequently, InceptionResNetV2, when combined with the random forest method, achieved the highest accuracy on the PlantDoc dataset. However, the MobileNetV2 was the best CNN architecture in fast prediction if the computation time is considered. Further, the EfficientNetB1 provided the second fastest prediction time.

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

We compared the experimental results with the previous studies. For the PlantDoc and Crop-PlantDoc datasets, Singh et al. [12] achieved an accuracy of 29.73% and 70.53% on the PlantDoc and Crop-PlantDoc datasets. Our stacking ensemble of CNN performed better than Singh et al. [12] with an accuracy of 72.18% and 90.71% on the PlantDoc and Crop-PlantDoc datasets. Furthermore, for the iCassava2019 dataset, our experimental result presents greater accuracy than the accuracy obtained from Enkvetchakul and Surinta [15]. The results reported in Enkvetchakul and Surinta [15] achieved 84.51% accuracy. In comparison, our proposed method achieved an accuracy of 91.87%.

4. Conclusions.

This paper has proposed a stacking ensemble of deep CNNs to recognize plant leaf disease images. First, we chose four lightweight CNNs, that were InceptionResNetV2, NASNetMobile, MobileNetV2, and EfficientNetB1, to compare the accuracy of results. The experiments showed that the EfficeientNetB1 significantly outperformed other CNN models on all plant leaf disease datasets.

We also demonstrated the impact of the ensemble learning method and the stacking ensemble learning method. Second, two types of ensemble learning, unweighted majority vote and unweighted average methods, were proposed to recognize the output probabilities of the CNN models. Ensemble learning with the unweighted average method combined with EfficientNetB1 achieved the best accuracy performance on the three datasets. Third, we proposed to use five machine learning classifiers, consisting of LR, SVM, KNN, RF, and LSTM, to create a model from the output probabilities of the CNN models. We found that EfficientNetB1 still outperformed all CNN models on Crop-PlantDoc and iCassava2019 datasets. It was the only InceptionResNetV2 that achieved better performance on the

Table 3. Performances of the meta-learner methods trained the model using the output probabilities from the lightweight CNNs on (A) PlantDoc, (B) Crop-PlantDoc, and (C) iCassava2019 datasets

(A)

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

(B)

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

(C)

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

PlantDoc dataset. In the best of our experiments, the proposed stacking ensemble of the CNN framework was finally combined with EfficientNetB1, which was the lightweight model and random forest for the classifier. For the meta-learner method, all machine learning methods could further improve plant leaf disease recognition performance.

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

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

REFERENCES

[1] Y. Zhong and M. Zhao, Research on deep learning in apple leaf disease recognition, *Comput. Electron. Agric.*, vol.168, DOI: 10.1016/j.compag.2019.105146, 2020.
[2] A. Darwish, D. Ezzat and A. E. Hassanien, An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis, *Swarm Evol. Comput.*, vol.52, DOI: 10.1016/j.swevo.2019.100616, 2020.
[3] S. V. Militante, B. D. Gerardo and N. V. DIonisio, Plant leaf detection and disease recognition using deep learning, *Proc. of Eurasia Conference on IOT, Communication and Engineering (ECICE)* pp.579-582, 2019.
[4] S. Mahmoud and S. Yaroshchak, Bagging of convolutional neural networks for diagnostic of eye diseases, *Proc. of the 4th International Conference on Computational Linguistics and Intelligent Systems (COLINS)*, pp.715-729, 2020.
[5] M. Khanramaki, E. A. Asli-Ardeh and E. Kozegar, Citrus pests classification using an ensemble of deep learning models, *Comput. Electron. Agric.*, vol.186, DOI: 10.1016/j.compag.2021.106192, 2021.
[6] M. Gour and S. Jain, Stacked convolutional neural network for diagnosis of COVID-19 disease from X-ray images, *arXiv.org*, arXiv: 2006.13817, 2020.
[7] T. Chompookham and O. Surinta, Ensemble methods with deep convolutional neural networks for plant leaf recognition, *ICIC Express Letters*, vol.15, no.6, pp.553-565, 2021.
[8] C. Szegedy, S. Ioffe, V. Vanhoucke and A. A. Alemi, Inception-v4, Inception-ResNet and the impact of residual connections on learning, *Proc. of the 31st AAAI Conference on Artificial Intelligence*, pp.4278-4284, 2017.
[9] B. Zoph, V. Vasudevan, J. Shlens and Q. V. Le, Learning transferable architectures for scalable image recognition, *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)* pp.8697-8710, 2018.
[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.4510-4520, 2018.
[11] M. Tan and Q. V. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, *Proc. of the 36th International Conference on Machine Learning (ICML)*, pp.10691-10700, 2019.
[12] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat and N. Batra, PlantDoc: A dataset for visual plant disease detection, *Proc. of International Conference Proceeding Series*, pp.249-253, 2020.
[13] E. Mwebaze, T. Gebru, A. Frome, S. Nsumba and J. Tusubira, iCassava 2019 fine-grained visual categorization challenge, *arXiv.org*, arXiv: 1908.02900, pp.4321-4326, 2019.
[14] M. Nagano and T. Fukami, Development of a skin texture evaluation system using a convolutional neural network, *International Journal of Innovative Computing, Information and Control*, vol.16, no.5, pp.1821-1827, 2020.
[15] P. Enkvetchakul and O. Surinta, Effective data augmentation and training techniques for improving deep learning in plant leaf disease recognition, *Appl. Sci. Eng. Prog.*, vol.15, no.3, pp.1-12, 2022.