Plant Disease Recognition Using Transfer Learning and Evolutionary Algorithms

1st Abdullateef .O Agbaje
School of Electronics and Information Engineering
Beihang University
Beijing, China
lateef.agbaje@buaa.edu.cn

2nd Jin Tian
School of Electronics and Information Engineering
Beihang University
Beijing, China
jintian@buaa.edu.cn

***Abstract*—This study used pre-trained convolutional neural network models to perform plant disease recognition. The pre-trained models were fine-tuned and trained on a Plant Village dataset (a publically opened dataset). The dataset contains 54,303 images divided into 38 classes with 14 distinct plant species and majorly divided into healthy and diseased plants. We implemented different parameters to improve the network's performance during training, such as batch size, image size, different numbers of epochs, and class weight. In this study, we used class weight due to the data imbalance (some classes were better represented than others); the reweighting technique tested different image sizes to evaluate the performance. The pre-trained models were used for transfer learning by freezing the network's last layer. The models used in this study are MobileNetV2, EfficientNet-B5, and InceptionV3; they acquired a diseased classification accuracy of 96.53%, 98.73%, and 96.39%, respectively. The transfer learning models' predictions were combined using two types of ensemble techniques; grid search ensemble and differential evolution algorithm with improved accuracy of 99.14% for grid search algorithm and 99.16% for differential evolution algorithm in classifying diseased and healthy plants. The ensemble method's successful classification of diseased and healthy plants is promising. It can positively impact the further improvement of plant disease recognition and serve as an early warning tool for farmers and consultants in real-time conditions.**

Keywords*— *Differential Evolution, Image Recognition, Plant Disease, Transfer Learning, Deep Learning.

I. Introduction

Agriculture is an important sector to sustain the world population. Technology and agriculture play an essential role to eradicate poverty and hunger. Focusing on agriculture will boost food production, create jobs, eradicate hunger, and increase gross domestic product (GDP). For more agricultural productivity, there is a need for innovative technology to increase production and lower wastage at all levels. Humans depend on agriculture for sustenance. Similarly, agricultural methods have evolved and need technology to meet human needs.

Plant disease recognition is an important aspect to improve food productivity and poverty alleviation. The optical recognition of plant diseases has come a long way in agriculture. Experts in plant phytology also misdiagnose or fail to diagnose the disease affecting a plant; an automatic plant disease recognition system comes into play. It will be difficult for a farmer or an expert in phytology to cover a large farm to detect plant diseases. Also, small-scale farmers may not have access to consultants, and if they have access might be expensive to bear. In this case, a system that can automatically recognise plant disease is essential. This system can serve as an early detection system for farmers and consultants alike.

Drones and smartphones are currently some of the latest technologies to boost agricultural throughput. The use of smartphones and drones can benefit small-scale farmers and large-scale farmers and help with crop and livestock management. Incorporating drone usage in farmland saves time and decreases production costs in the long run.

Harnessing the power of GPU (Graphics Processing Unit), mainly used by gaming applications for machine learning applications, has brought much improvement. It has also given individuals access to research and development of artificial intelligence applications, leading to novel methodologies and models. Different classifiers have been developed to classify images such as K-nearest neighbour (KNN), random forest, support vector machine (SVM), naïve Bayes, decision tree, ANN and deep learning.

Deep learning is the best classifier for distinguishing image features through training and enough data. The demerit of deep learning is the classification without enough data; if a dataset does not have enough data, the predictive power of the deep learning model will be wrong. In recent years, transfer learning has improved the predictive power of deep learning by training a new dataset based on previous training. Transfer learning decreases the time it takes to train a network on a new dataset, improving accuracy. Some works of literature have reported good results in using deep learning; Ji et al. [1] recently used a convolutional neural network to estimate the severity and defects of disease in plant leaves. Chen [2] used transfer learning for deep CNNs to enhance the learning ability of tiny lesion symptoms. A novel deep learning architecture called INC-VGGN for identifying plant disease images achieves a validation accuracy of 91.83% on the public dataset. Gutierrez [3] Performed automated pest detection on greenhouse tomatoes and pepper crops based on machine learning/deep learning strategies to compare approaches.

In this study, we used three pre-trained networks, namely; MobileNetV2, EfficientNet-B5 and InceptionV3, to train on our dataset. We pre-processed the dataset and balanced the dataset using the reweighting technique. Also, using ensemble learning techniques, we combined the result of all the trained networks to get better performance.

In the remainder of this paper, we will present the literature based on plant diseases in section 2. Section 3 presents the deep learning methodologies and the dataset. The 4th section presents the results obtained from the experiment and the performance of each model compared to the ensemble technique. The final section discusses the result obtained, the advantage and limitations of the study, and the further research that can be done to bring advancement in the field.

II. Related Works

Many studies have explored disease detection in plant leaf images using automated detection systems from shape and colour texture. Amongst these studies is a study that shows that judging by the edges of leaves and a Back Propagation Algorithm, the species of leaves could be determined. Leaves disease and pest recognition system was implemented by [4] using a Back Propagation Neural Network, requiring more species of plant leaves and a quicker training algorithm for better recognition ability.

A. *Shape and Texture-Based Identification*

Leaf images can be acquired in some ways, ranging from drones to mobile phones, [5] used phones to capture images of an infected soybean field to classify the brown spot (*septoria leaf blight*) and frog eye disease. Using the k-nearest neighbour classifier achieved 70% and 80% image detection accuracy for brown spot and frog eye disease, respectively of 50 image testing. This system extracted the leaves shape features to train the KNN algorithm.

Other studies have combined a variety of soybean leaf diseases, classifying them according to their health status. The knowledge of leaves' health status is fundamental (healthy or diseased) for disease to be identified, [6] classified soybeans leaf disease by identifying the colour combination and texture characteristics of leaves using some set of rules and three SMV classifiers. The system was able to identify whether the leaves were healthy or diseased and achieved a 90% accuracy.

In [7], tomato leaf images were used to identify two diseases. The researcher used geometric and histogram features to extract useful image features in segmented disease portions, a support vector machine (SVM) algorithm with different kernels as a classifier, and an N-fold cross-validation technique for performance evaluation.

An automatic leaf and fungal disease classification and identification system proposed by an article [8] used bacterial foraging optimisation. A based radial basis function neural network (BRBFNN) that uses a region-growing algorithm that increases the system's efficiency by extracting leaves features based on seed points having certain related features. The bacteria foraging optimisation technique speeds up the network and improves the classification and identification of infected plant leaves.

[9] also proposed a system for digital images processing aiding identification and classification of diseased plant leaves present on different agricultural plants using a combination of algorithms.

B. *Deep Learning-Based Identification*

Modern studies have shown that deep learning techniques are highly successful at image processing-related problems, based on their ability to extract efficient features from images.[10]–[13].

Deep learning has shown to be the most prevalent architecture because its model can learn important features from these input images at different convolutional levels, similar to how the human brain works. Deep learning comprises different components (convolutional, pooling, fully connected layers and activation functions). [14] implemented InceptionV3, InceptionResNetV2, MobileNetV2, and EfficientNetB0 on CNN architectures to identify different plant diseases using the leaves of different disease-affected plants because these models' parameter size is optimal compared to other architectures. With EfficientNetB0, achieving the best accuracy (99.56%) compared to InceptionV3, MobileNetV2, and InceptionResNetV2.

[15] identified plant diseases using healthy and infected images of their leaves with dedicated deep learning models built based on five CNN architectures (AlexNet, AlexNetOWTBn, GoogLetNet, OverFeat and VGG), VGG has the highest accuracy of 99.53%. The drawback is that the author took the test data from the same dataset as the training data that needs to be widened for more robustness in the real-life cultivation system.

A spontaneous plant disease investigative system was proposed by [16] to detect certain agricultural plant diseases, which uses deep learning algorithms assisted by transfer learning affordable to farmers with mobile phones, Internet access and PC to diagnose their plant diseases.

The researcher identified five various cassava diseases with the use of the plain convolutional neural network (PCNN) and deep residual neural network (DRNN) in [17]. The approach showed that DRPN outclassed PCNN archiving an accuracy of 96.75%.

In addition to cassava disease classification, [18] used predictive machine learning models for cassava disease detection. The models were assisted by image augmentation techniques to counter high-class imbalance, and it was able to achieve a 93% accuracy.

Earlier studies have presented that transfer learning is an effective method for different applications with a much lower computational requirement than learning from scratch, which is beneficial to mobile applications. [19], [20]. [21] achieved an accuracy of 99.35%, with a public data set of 54,306 images trained to classify 14 plant species and 26 diseases, showing that transfer learning consistently achieves better results.

Transfer learning model was also used to train a deep convolutional neural network to identify particular cassava leaf diseases and pest damage using inception v3; their method shows the reliability [22].

[23] classified four cassava diseases, including healthy ones, in a novel approach to improve classification performance by incorporating various models to categorise diseases phase by phase. The technique uses deep learning algorithms, and classification is done using faster RCNN applying transfer learning to this pre-trained algorithm, achieving the accuracy and F-measure of 0.96.

Instead of collecting physical samples of plant leaves and analysing them in the laboratories, a more suitable deep learning architecture-processing unit presented an accurate and applicable solution to identify, classify and locate diseases in tomato plants [24].

C. *Transfer Learning Approach*

Transfer learning is a technique used in machine learning where a saved model / pre-trained network on a large dataset uses the knowledge gained from one task and uses it as a starting point to solve a similar task without taking so much computation time.

Pre-trained models are usually used with problems with a limited amount of data, which helps by improving the training and computation time and improves the performance of the deep learning model.

In transfer learning, the last few layers of the pre-trained model are customised for a given task. The customised layer can consist of a fully connected layer, a dropout layer and a classification layer with a softmax activation function which contains the number classes, in our case, 38 classes.

III. Methodology

A. Dataset and Training

A plant village open-access dataset was used to train and test the transfer learning model. The dataset contains 54,303 images divided into 38 diseased, and healthy plant leaves classes. The dataset contains plant leaves from 14 different plants; the images were collected in a laboratory condition. The plant village dataset used in this paper has three sets of images such as coloured image, greyscale, and segmented images. TABLE 1 below displays statistical information on the dataset, including class names and the number of images in a class.

In this paper, we will only use coloured images to train our model because it was shown in previous research [21] that the greyscale and segmented images do not necessarily improve the performance of the model in similar applications.

The dataset was divided into training, validation, and test sets (70-20-10). Thus, 37,995 were used for training, 10,849 images for validation and 5,459 to test the model's performance on unseen images.

Before training the model to recognise diseased leaves, an image pre-processing step was done on the images. During the image pre-processing stage, one of the checks is to know if the data in each class is balanced. There were variations in the number of images in the classes; the data distribution was not balanced.

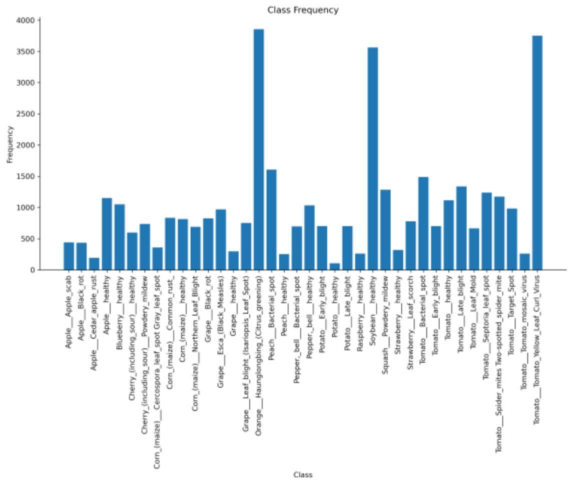


Fig. 1. Class Frequency

Different techniques are available to balance the data; the technique selected is the reweighting technique; the reweighting technique involves weights. We assigned weights to classes based on their frequency distribution. It assigns more weight to classes with less representation to increase the prediction error of under-represented classes when they are incorrectly classified.

TABLE I. Description Of The Plant Village Dataset

Class	Plant Name	Disease Name	Scientific Name	No. of Images
C1	Apple	Apple scab	Venturia inaequalis	441
C2	Apple	Black rot	Botryosphaeria obtusa	434
C3	Apple	Cedar apple rust	Gymnosporangium	192
C4	Apple	Healthy	-	1151
C5	Blueberry	Healthy	-	1051
C6	Cherry (including sour)	Healthy	-	597
C7	Cherry (including sour)	Powdery mildew	Podosphaera clandestina	736
C8	Corn (maize)	Cercospora leaf spot	Cercospora zeae-maydis	359
C9	Corn (maize)	Common rust	Puccinia sorghi	834
C10	Corn (maize)	Healthy	-	813
C11	Corn (maize)	Northern Leaf Blight	Exserohilum turcicum	689
C12	Grape	Black rot	Guignardia bidwellii	826
C13	Grape	Esca (Black Measles)	Phaeomoniella chlamydospora	968
C14	Grape	Healthy	-	296
C15	Grape	Leaf blight	Pseudocercospora vitis	753
C16	Orange	Haunglongbing	-	3854
C17	Peach	Bacterial spot	Xanthomonas campestris	1607
C18	Peach	Healthy	-	251
C19	Pepper-bell	Bacterial spot	Xanthomonas campestris	697
C20	Pepper-bell	Healthy	-	1034
C21	Potato	Early blight	Alternaria solani	700
C22	Potato	Healthy	-	106
C23	Potato	Late blight	Phytophthora infestans	700
C24	Raspberry	Healthy	-	259
C25	Soybean	Healthy	-	3563
C26	Squash	Powdery mildew	Podosphaera xanthii	1284
C27	Strawberry	Healthy	-	319
C28	Strawberry	Leaf scorch	Diplocarpon earliana	776
C29	Tomato	Bacterial spot	Xanthomonas perforans	1488
C30	Tomato	Early blight	Alternaria˙sp.	700
C31	Tomato	Healthy	-	1113
C32	Tomato	Late blight	Phytophthora infestans	1336
C33	Tomato	Leaf mold	Lycopersicon	666
C34	Tomato	Septoria leaf spot	Septoria lycopersici	1239
C35	Tomato	Spider mites Two-spotted spider mite	Tetranychus spp.	1173
C36	Tomato	Target Spot	Corynespora cassiicola	982
C37	Tomato	Tomato mosaic virus	Tomato mosaic	261
C38	Tomato	Tomato Yellow Leaf Curl Virus	Begomovirus	3749

We divided the network training into training with an unbalanced dataset and a balanced dataset using class weight.

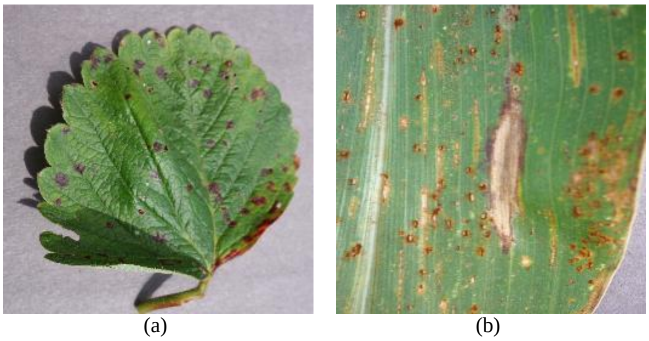


Fig. 2. Sample images from the dataset

Another pre-processing image technique to train the network includes size reduction, rescaling and shuffling of the input images. We applied the exact process to the validation dataset.

B. Ensemble Learning Architecture

This paper uses a transfer learning approach; the selected pre-trained networks include the Mobilenet-V2, EfficientNet-B5 and InceptionV3 models. Also, we introduced another method to get better performance, which is called ensemble learning.

Ensemble learning is an approach that involves the combination of multiple models predictions to improve predictive performance. The steps of creating an ensemble model in this paper include;

- Use the pre-trained models to train the network individually.
- Save and load the individual model and freeze its top layer.
- Combine the output and add dense layers.

Fit and train the newly created ensemble model

C. Pre-trained Model to Individually Train the Network

In this section, individual models are created with the help of transfer learning. The pre-trained model used are MobileNetV2, EfficienrNetB5 and InceptionV3. The feature vector of the pre-trained model was used to carry out the transfer learning; this feature vector has its top layer removed to add a dense layer to fit the number of classes of the dataset.

The image size of the training data was resized for each of the models; the MobileNetV2 model used a (224 × 224) image size as input, the EfficientNet-B5 used (456 × 456) image, and we assigned (299 ×299) pixel size to the InceptionV3 model.

We froze the pre-trained models' initial layers by setting trainable to false. Setting trainable to false will only train the newly added layer in the network, in this case, the dense layer that includes the number of classes in the dataset. After creating the models, the number of trainable parameters for the MobileNetV2 (48,678) was lesser than EfficientNet-B5 and InceptionV3 (77,862 parameters). We then fit them on our training data.

Also, some call-back functions were implemented, such as early stopping after 24 epochs, Model checkpoint to save the best model during training and reducing learning rate on a plateau after 10 epochs.

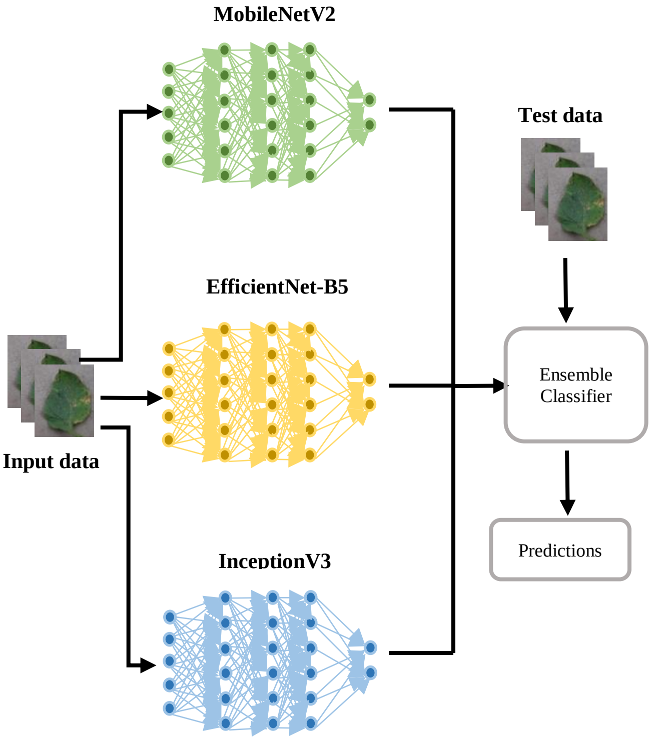


Fig. 3. Ensemble learning architecture

IV. Results

A. Metrics

As stated in the previous section, the pre-trained models used are MobilenetV2, EfficientNet-B5, InceptionV3, and the use of an ensemble learning approach to combine the result of each model for better prediction.

All the models used the dataset to train, using the training parameters shown in **Error! Reference source not found.**. The learning rate was set to reduce automatically after every 10 epochs if there is no improvement in the validation loss. An early stopping call-back was also used to stop training if there is no improvement after 24 epochs.

To get a good view of the model's performance, we used different performance metrics such as;

- **Accuracy:** the successful classification of the plant images against the overall images
- **Training and validation loss:** this shows the average loss of each batch of images over the overall batch.
- **Training epoch:** this is when the model has the best performance.
- **Time per epoch:** this is the average time used to train per epoch.

TABLE II. Parameter to Train the Network

Parameter	Value
Batch size	32
Epochs	30
Reduce LR on a plateau	10
Early stopping	24

- **Precision and recall:** this checks the performance of each model based on the prediction and the actual label.
- **F1 score:** this is the overall performance based on precision and recall.

A reweighting technique was used to avoid overfitting by assigning class weights to each class during the training phase.

B. Parameters and Flowchart

The differential evolution (DE) is a population-based evolution algorithm used in optimisation problems [25]; it is a stochastic population method that improves a candidate solution based on an evolutionary process. The algorithm consists of 3 steps; mutation, crossover and selection. These steps are done iteratively to enable the population to get a better solution in the design space. DE mutates each candidate solution by mixing features from the candidate solution to generate a trial candidate at each iteration.

Several strategies can generate a trial candidate in differential evolution; the best1bin strategy was used in this study. Three members of the population are randomly selected. The weighted difference between two members out of the three selected vectors is calculated and added to the third vector (the best vector), which gives the target candidate.

$$V_{i,G+1} = X_{r1,G} + F\left(X_{r2,G} - X_{r3,G}\right) \quad (1)$$

Where $X_{r1,G}/\ X_{r2,G}\ /\ X_{r3,G}$ = Solution member

$V_{i,G+1}$ = Donor vector

F = Scale factor

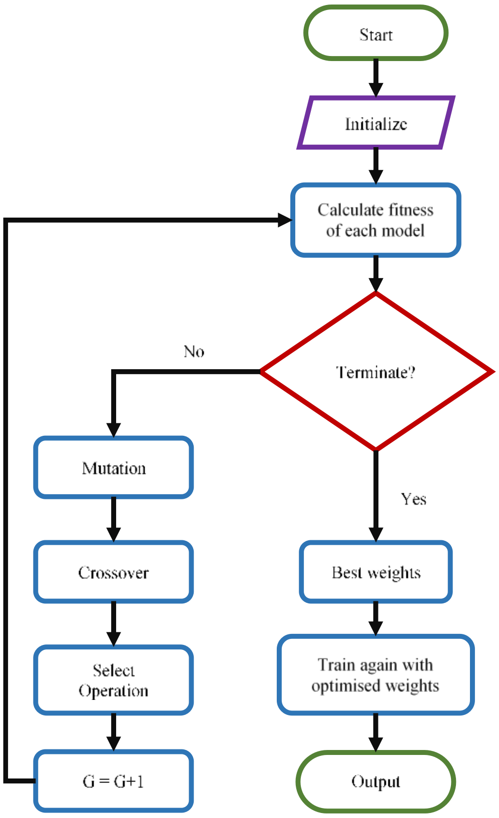


Fig. 4. Differential evolution flowchart

When the population members are well separated, the difference vector will be significant. When the DE evolves over some generations, the population members tend to move towards the same region (they become closer), the magnitude of the difference vector will naturally become smaller. DE naturally alter the search behaviour from exploration to exploitation as the population evolves; this makes DE a self-referential mutation.

The crossover combines solutions from the previous generation the target vector $X_{i,G}$ and the elements of the donor vector $V_{i,G+1}$. The selection step compares the target vector and the trial vector (offspring); if the trial vector is better (minimised) than the target vector, the trial vector is chosen and vice versa.

This study used a loss function as the objective function by initialising weight and test label as input. An upper bound and lower bound value was chosen for each model, the maximum number of generations was also chosen. Listed in TABLE III are the parameters:

C. Performance

TABLE IV shows the performance of the implemented models based on accuracy, loss, training epoch, and time it took the model to train for each epoch. The EfficientNet-B5 has the highest classification accuracy with 98.31%, but the training time is more than that of MobileNetV2, with an average time of 1182 seconds per epoch. MobileNetV2 has the shortest time per epoch, making it lightweight and easy to deploy on smartphones in the field.

We set the epoch of each model to 30; it shows from Fig. 5 that there will be little or no improvement if we continue to train. One of the models did not reach the 30th epoch to reach its best performance. The MobileNetV2 reached its best performance at the 11th epoch, EfficientNet-B5 and InceptionV3 reached their best performance at the 30th epoch.

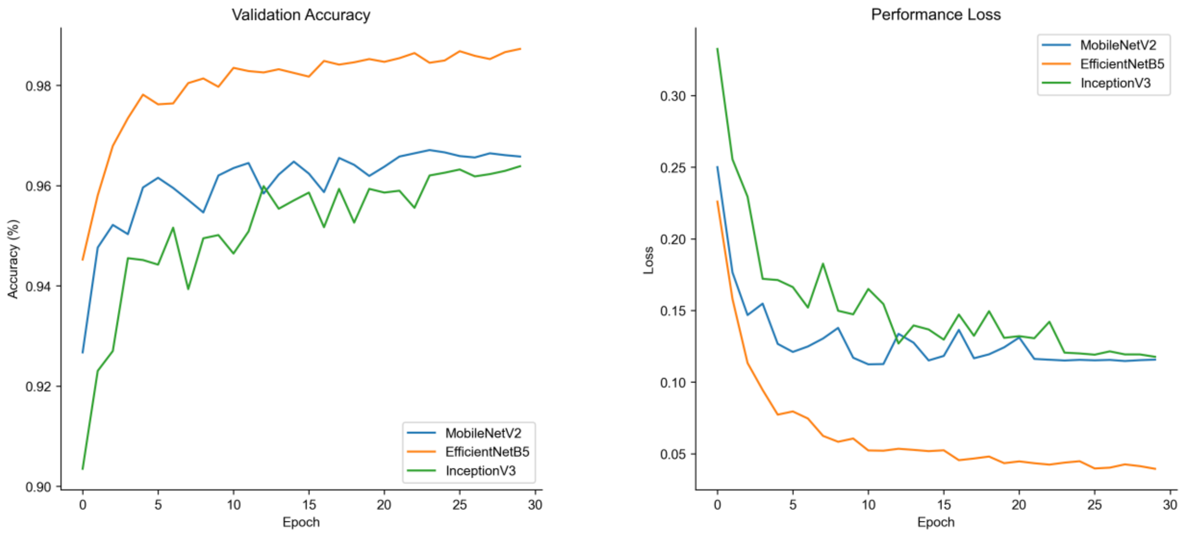
In this paper, a weighted ensemble technique was used; it involves training multiple networks (in this case, pre-trained models) and combining them to get a better prediction, which often gives a better prediction than a single model. The ensemble technique reduces the variance of predictions; without it, individual model training can result in different predictions each time it is trained.

TABLE III. DIFFERENTIAL EVOLUTION PARAMETERS

Parameter	Value
Lower Bound – Upper Bound	0.0 – 1.0
Maximum iteration	1000
tolerance	0.0000001

TABLE IV. TRAINING RESULT

Model	Validation Accuracy	Test Accuracy	Loss	Epoch	Time/Epoch (s)	Training Epoch
MobileNetV2	96.53	96.70	0.1110	30	81	11
EfficientNet-B5	98.73	98.31	0.0394	30	1182	30
InceptionV3	96.39	96.21	0.1176	30	190	30



(a) (b)

Fig. 5. (a) Validation accuracy of the models. (b) Validation loss of the models during training

We experimented with two weighted ensemble methods: the grid search ensemble and the differential evolution algorithm. The grid search ensemble method assigns weight to each ensemble member (model) to find the best weight. The process of assigning weights (coefficients) to each model depends on its' performance or how its prediction is trusted.

One challenge was combining three different models with different image sizes as input. An individual input pipeline was created for each model to get the test data with the appropriate image size for optimal prediction.

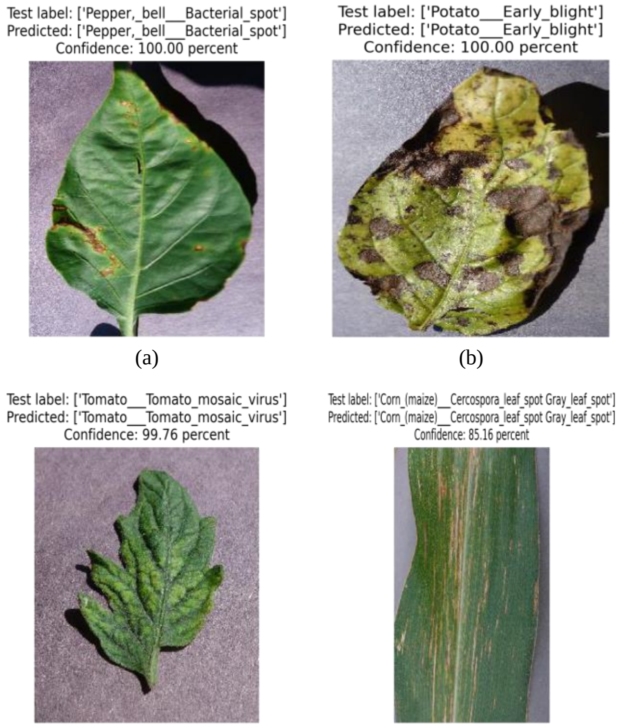
TABLE V indicates that the grid search ensemble technique achieved a better accuracy of 99.14% compared to the individual models. Weights between 0.0 - 1.0 were assigned to the models, assuming the sum of weights of the models will sum up to 1.

Assigning the weights involves the sum of products of the prediction and assigned weights; all possible 3-elements (the models) vectors within 0 and 1 (Cartesian product). The maximum ensemble member after the Cartesian product is selected and its accuracy calculated; this step is done for each generated weight.

Finally, check for the maximum accuracy amongst the different combinations of weights assigned; this is called a grid search weighted ensemble approach. The best performance is achieved when the focus of the weight is on MobileNetV2 and EfficientNet-B5.

TABLE V. MODEL PERFORMANCE

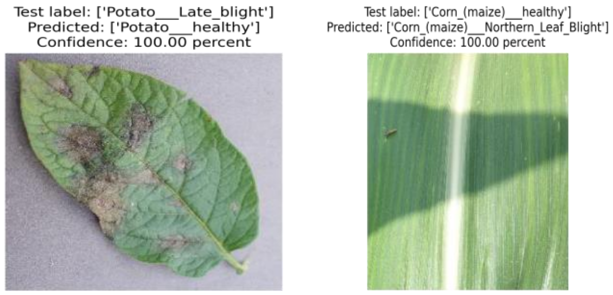
Transfer Learning Model	Precision	Recall	F1-Score	Accuracy
MobileNet V2	0.9679	0.9670	0.9671	96.70
EfficientNet -B5	0.9833	0.9831	0.9832	98.31
InceptionV3	0.9626	0.9621	0.9622	96.21
Grid Search Ensemble	0.9915	0.9914	0.9914	99.14
Differential Evolution	0.9917	0.9916	0.9916	99.16


Test label: ['Pepper,_bell___Bacterial_spot']
Predicted: ['Pepper,_bell___Bacterial_spot']
Confidence: 100.00 percent
Test label: ['Potato___Early_blight']
Predicted: ['Potato___Early_blight']
Confidence: 100.00 percent
Test label: ['Tomato___Tomato_mosaic_virus']
Predicted: ['Tomato___Tomato_mosaic_virus']
Confidence: 99.76 percent
Test label: ['Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot']
Predicted: ['Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot']
Confidence: 85.16 percent

(a) (b)

(c) (d)

Fig. 6. Sample of correct classification of various images of the testing dataset


Test label: ['Potato___Late_blight']
Predicted: ['Potato___healthy']
Confidence: 100.00 percent
Test label: ['Corn_(maize)___healthy']
Predicted: ['Corn_(maize)___Northern_Leaf_Blight']
Confidence: 100.00 percent

(a) (b)

Fig. 7. Samples of incorrect classification

The TABLE V above indicates that DE 99.16% performed better than the grid search algorithm and the focus of the weight was on MobileNetV2 and EfficientNet-B5.

V. Discussion

In this study, three pre-trained models and two ensemble techniques were developed and implemented to identify plant diseases; predictions from the pre-trained neural network models were weighted based on trust in each model to get better predictions.

The implemented models used the publically available plant village dataset during training, with 54,303 images. The images comprised 38 classes of diseased and healthy leaves of 14 species of plants.

The findings from this study prove that the weighted ensemble learning approaches achieved better accuracy than a single model; it also shows the power of transfer learning over the use of untrained models, which might take a long time to train. The implemented models in this paper reached their best performance at different epochs, and the MobileNetV2 had its best result at the 11th epoch; EfficientNet-B5 and InceptionV3 both had 30 epochs, respectively.

The accuracy of each model shows that they can be used on their own and get good predictions. The EfficientNet-B5 has the best performance, and it shows a success rate of 98.31% classification accuracy out of 5,459 unseen test images; it has an error rate of 1.69%, which is quite good for the number of epochs it was trained. All the pre-trained models had trouble classifying some images, such as Maize Cercospora leaf spot, Potato Late blight, and others specific to each model. The weighted ensemble technique combined the models' prediction leaning primarily on the model with a better predictive power to correct the misclassification of other models. Also, the models with lower predictive power help predict that the best model misclassified. The limitation of the grid search ensemble method is the assumption that the assigned vector weights will sum up to 1, but in practice, it does not always sum up to 1. A workaround to the problem was implemented using the L1-norm and dividing the weight by the resulting value. Also, it is an exhaustive searching algorithm which makes it slow and takes more computing power. The DE algorithm is fast and not as computationally expensive as the grid search algorithm. The weight of the DE algorithm was normalised, so the weight of each model can add up to one.

Despite the ensemble method's high predictive power, it is still not a generic solution to plant disease recognition. A database that includes different species of plants and their diseases is needed to get a generic solution. Also, the computing power needed to train a model to classify a robust database filled with plant disease images is much for a researcher with a device with one GPU.

This research proved that combining different models, especially pre-trained models (through transfer learning), improves plant disease identification's predictive power. The study aims to provide a less costly solution to intelligent farming and farmers with limited funds or subsistent farmers. Further study is needed to test the real-time performance of the method used.

References

[1] M. Ji, K. Zhang, Q. Wu, and Z. Deng, "Multi-label learning for crop leaf diseases recognition and severity estimation based on convolutional neural networks," *Soft Comput.*, 2020, doi: 10.1007/s00500-020-04866-z.

[2] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Comput. Electron. Agric.*, vol. 173, no. November 2019, p. 105393, 2020, doi: 10.1016/j.compag.2020.105393.

[3] A. Gutierrez, A. Ansuategi, L. Susperregi, C. Tubío, I. Rankić, and L. Lenža, "A Benchmarking of Learning Strategies for Pest Detection and Identification on Tomato Plants for Autonomous Scouting Robots Using Internal Databases," *J. Sensors*, 2019, doi: 10.1155/2019/5219471.

[4] M. S. P. Babu and B. S. Rao, "Leaves Recognition Using Back Propagation Neural Network-Advance for Pest & Disease Control On Crops. Andhra University, India," *IndiaKisan. Net Expert Advis. Syst.*, 2010.

[5] S. Shrivastava and D. S. Hooda, "Automatic Brown Spot and Frog Eye Detection from the Image Captured in the Field," *Am. J. Intell. Syst.*, pp. 131–134, 2014, doi: 10.5923/j.ajis.20140404.01.

[6] S. Kaur, S. Pandey, and S. Goel, "Semi-automatic leaf disease detection and classification system for soybean culture," *IET Image Process.*, vol. 12, no. 6, pp. 1038–1048, 2018, doi: 10.1049/iet-ipr.2017.0822.

[7] U. Mokhtar, M. A. S. Ali, A. E. Hassanien, and H. Hefny, "Identifying two of tomatoes leaf viruses using support vector machine," 2015, doi: 10.1007/978-81-322-2250-7_77.

[8] S. S. Chouhan, A. Kaul, U. P. Singh, and S. Jain, "Bacterial foraging optimization based radial basis function neural network (BRBFNN) for identification and classification of plant leaf diseases: An automatic approach towards plant pathology," *IEEE Access*, vol. 6, pp. 8852–8863, 2018, doi: 10.1109/ACCESS.2018.2800685.

[9] T. N. Tete and S. Kamlu, "Plant Disease Detection Using Different Algorithms," 2017, doi: 10.15439/2017r24.

[10] M. Woźniak and D. Połap, "Adaptive neuro-heuristic hybrid model for fruit peel defects detection," *Neural Networks*, 2018, doi: 10.1016/j.neunet.2017.10.009.

[11] A. Urbonas, V. Raudonis, R. Maskeliunas, and R. Damaševičius, "Automated identification of wood veneer surface defects using faster region-based convolutional neural network with data augmentation and transfer learning," *Appl. Sci.*, 2019, doi: 10.3390/app9224898.

[12] G. Capizzi, G. Lo Sciuto, C. Napoli, E. Tramontana, and M. Woźniak, "A novel neural networks-based texture image processing algorithm for orange defects classification," *Int. J. Comput. Sci. Appl.*, 2016.

[13] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, 2020, doi: 10.1007/s10462-020-09825-6.

[14] S. M. Hassan, A. K. Maji, M. Jasiński, Z. Leonowicz, and E. Jasińska, "Identification of plant-leaf diseases using cnn and transfer-learning approach," *Electron.*, 2021, doi: 10.3390/electronics10121388.

[15] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, no. January, pp. 311–318, 2018, doi: 10.1016/j.compag.2018.01.009.

[16] K. Rangarajan Aravind and P. Raja, "Automated disease classification in (Selected) agricultural crops using transfer learning," *Automatika*, 2020, doi: 10.1080/00051144.2020.1728911.

[17] D. O. Oyewola, E. G. Dada, S. Misra, and R. Damaševičius, "Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing," *PeerJ Comput. Sci.*, 2021, doi: 10.7717/peerj-cs.352.

[18] G. Sambasivam and G. D. Opiyo, "A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks," *Egypt. Informatics J.*, 2021, doi: 10.1016/j.eij.2020.02.007.

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. F. Li, "Large-scale video classification with convolutional neural networks," 2014, doi: 10.1109/CVPR.2014.223.

[20] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," 2014.

[21] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.*, 2016, doi: 10.3389/fpls.2016.01419.

[22] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, “Deep learning for image-based cassava disease detection,” *Front. Plant Sci.*, 2017, doi: 10.3389/fpls.2017.01852.

[23] I. Sangbamrung, P. Praneetpholkrang, and S. Kanjanawattana, “A novel automatic method for cassava disease classification using deep learning,” *J. Adv. Inf. Technol.*, 2020, doi: 10.12720/jait.11.4.241-248.

[24] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, “A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition,” *Sensors (Switzerland)*, 2017, doi: 10.3390/s17092022.

[25] R. Storn and K. Price, “Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces,” *J. Glob. Optim.*, 1997, doi: 10.1023/A:1008202821328.