

Grape Leaves Diseases Classification using Ensemble Learning and Transfer Learning

Andrew Nader, Mohamed H.Khafagy, Shereen A. Hussien

Department of Computer Science
Faculty of Computers and Information
Fayoum University
Egypt

Abstract—Agriculture remains an important sector of the economy. Plant diseases and pests have a big impact on plant yield and quality. So, prevention and early detection of crop disease are some of the measures that must be implemented in farming to save the plants at an early stage and thereby reduce the overall food loss. Grapes are the most profitable fruit, but they are also vulnerable to a variety of diseases. Black Measles, Black Rot, and Leaf Blight are diseases that affect grape plants. Manual disease diagnosis can result in improper identification and use of pesticides, and it takes a long time. A variety of deep learning approaches have been used to address this issue of the identification and classification of grape leaf diseases. However, there are also limits to such approaches. Therefore, this paper uses deep learning with the concept of ensemble learning based on three famous Convolutional Neural Network (CNN) architectures (Visual Geometry Group (VGG16), VGG19, and Extreme Inception (Xception)). These three models are pre-trained with ImageNet. The performance of the proposed approach is analyzed using the Plant Village (PV) dataset of common grape leaf diseases. The Proposed model gives higher performance than the results achieved by using each Deep Learning architecture separately and compared with the recent approaches in this study. The proposed system outperformed the others with 99.82% accuracy.

Keywords—Ensemble learning; grape leaf diseases; convolutional neural network (CNN); transfer learning

I. INTRODUCTION

Agriculture is one of the most important sectors in the world, providing humans with food, raw materials, and other essentials. According to the United Nations (FAO) Food and Agriculture Organization, the world's population will reach 9.1 billion by 2050. As a result, to meet the nutrient needs of such a large population, the food growth rate needs to be boosted to 70% by 2050 [1]. Plant disease is one of the leading causes of crop loss. A plant disease is an abnormal condition that modifies the appearance or function of a plant. The visible effects of the disease on plants are described as symptoms. A symptom is any detectable change in the plant's color, form, and/or functions in response to a pathogen or disease-causing agent. The leaf is the most essential portion of the plant to inspect for illness. Plant leaf diseases are divided into three categories: fungal, viral, and bacterial [12, 13].

A. Fungal Diseases

Parasitic organisms have a system of branching threads that make up their bodies. Fungi can enter the host via stomata, but

many can pierce a solid surface. Once within, they can either expand through the live cells or stay primarily in the gaps between them. Black Rot, Black Measles, Leaf Blight, and Leaf Rust are signs of these diseases.

B. Viral Diseases

Viruses are not dispersed by water or wind, unlike bacteria and fungi. But insects and worms are the primary vectors of viral pathogens to plants.

C. Bacterial Diseases

Bacteria are small. There are two hundred kinds of bacteria that cause diseases in plants. Their form determines their classification. Spherical, rod-shaped microorganisms and twisted rods are the three primary kinds that may be known. There are many symptoms of a bacterial infection. Leaf Spot is considered the most prevalent of them.

Plant diseases, and how to quickly diagnose and address them to improve the health of crops, are among the most important problems that face agriculture and have an impact on its trade. Identifying the disease before it spreads over the farm to other plants and treating it is a massive challenge in and of itself. It takes a lot of time and effort to determine what kind of disease it has. Furthermore, not all disease types can be accurately identified by the farmer's naked eye.

Initially, plant infections can be detected with the help of an agricultural professional who is familiar with plant diseases. However, manual plant disease identification and determination is a strenuous task and takes a long time. A person's knowledge and experience determine the accuracy of a manual prediction [2].

To overcome the above problems, many studies on machine learning (ML)-based technologies, such as support vector machines (SVM) [3], K-nearest neighbors (KNN) [4], have recently been applied to improve decision-making to classify plant diseases [5]. However, all these proposed approaches face several challenges, including identifying regions of concern for processing and analysis (feature representation). Now, Deep Learning (DL) is considered the next evolution of machine learning, such as convolutional neural networks (CNN) [6]. This work proposes an approach for classifying grape plant diseases based on Ensemble Learning that aggregates three customized CNN architectures with trained weights (VGG16, VGG19, and Xception). The following are the study's major contributions:

- Enhance classification performance.
- Reduce overfitting.

The rest of this paper is structured as follows: In Section II, related works present the most common plant disease terminologies and techniques. The proposed model and materials are illustrated in Section III. Section IV discusses the results of the proposed model and compares them with other related models. Section V, the conclusion. Section VI, the future work.

II. RELATED WORK

Many methods for disease classification from the planet leaf have been presented, especially on the Plant village dataset. The Plant village Dataset [7] is a well-known, publicly available crop leaf collection that contains thousands of images. It has been used in various research and achieved good results. Some of these are as follows:

Akshai KP et al. [8] proposed a method to classify images of grape plant diseases from the Plantvillage dataset using the trained model. The CNN, VGG19, ResNet-152v2, and DenseNet models are all trained. The DenseNet model was the most accurate, with a score of 98.27%. For the convolution layer, a rectified linear activation function, or ReLU activation function, is utilized, and for the output layer, a Softmax activation function is used. The images were reduced to 224x224 pixels using Kera's image data generator, then augmentations like rotation, zoom, and shift were added. At a ratio of 80:20, the dataset is split into training and validation sets. This work had the problem of splitting the dataset into training and validation only and testing the model with the same data, which led to overfitting.

Y. Nagaraju et al. [9] proposed a fine-tuned VGG-16 network to categorize eight different apple and grape leaf types together. The Kera's library is used to load the pre-trained VGG-16 network. In a typical model, the SoftMax (classifier) layer is removed, and a new output layer (classifier) with a SoftMax activation function is added. The disease dataset for apples and grape leaves is split 80:20 with an accuracy of 97.87%. This work had the problem that although using 30 epochs, the accuracy was 97.87%.

E. Hirani et al. [10] proposed deep learning methods for identifying plant diseases. A Plantvillage dataset is used at a ratio of 80:20, with 70295 images for training and 17572 images for validation. This work used three methods: a customized convolutional neural network, INCEPTIONv3, Small Transformer Network (STN), and Large Transformer Network (LTN), with an accuracy of 95.566%, 97.14%, 97.66%, and 97.98%. This work had a problem with the resolution of images at 256*256 that needed more computation and time in the training and testing phases.

K. Z. Thet et al. [11] proposed a fine-tuning VGG16 with the GAP layer instead of VGG16's two fully connected layers before the SoftMax layer to classify the diseases on grape leaves. This work has achieved 98.4% more accuracy than others. It mostly focused on five diseases that are prevalent in Myanmar Grapevine Yard. This work had a problem with the

overall performance where the number of epochs and batch size were not determined.

III. PROPOSED METHODOLOGY

A. Dataset Description

The Plant Village dataset [7] is used to examine the proposed model performance in the classification of grape leaves diseases. It is a large and freely accessible database. It contains over 55,000 RGB images divided into 38 classes representing 14 different plant species. There are 12 healthy leaf classes as in Fig. 3 and 26 unhealthy leaf classes as in Fig. 1, 2, 4 out of 38 total. The grape leaves images are used in this study, which includes 4,062 images divided into four classes shown in Table I. The following is a summary of each selected disease [13, 14]:

1) *Grape: black rot*: Black rot fungus attacks on leaves grown in warm, wet seasons, the black rot fungus attacks the upper surface of the leaves, turning them reddish-brown and causing round to angular dots to emerge. As the spots merge, irregular reddish-brown blotches appear as shown in Fig.1.

2) *Grape: leaf blight*: It is also a fungal disease on leaves grown in high humidity conditions, caused by *Exserohilum turcicum*. At first, small yellow dots emerge along the leaf margins, then grow to become brown patches as shown in Fig. 4.

3) *Grape: esca (black measles)*: The esca fungus can affect leaves at any time during the growing season, but it is most common in July and August. The symptom is an interveinal "striping". In red varieties, the "stripes" are dark red, while in white cultivars, they are yellow. Fig. 2.

B. Proposed System

In this work, a powerful approach for classification grape leaves diseases based on Ensemble Learning has been presented as shown in Fig. 5.



Fig. 1. Grape_Black_Rot.



Fig. 2. Grape_Esca (Black Measles).



Fig. 3. Grape_Healthy.



Fig. 4. Grape_Leaf Blight.

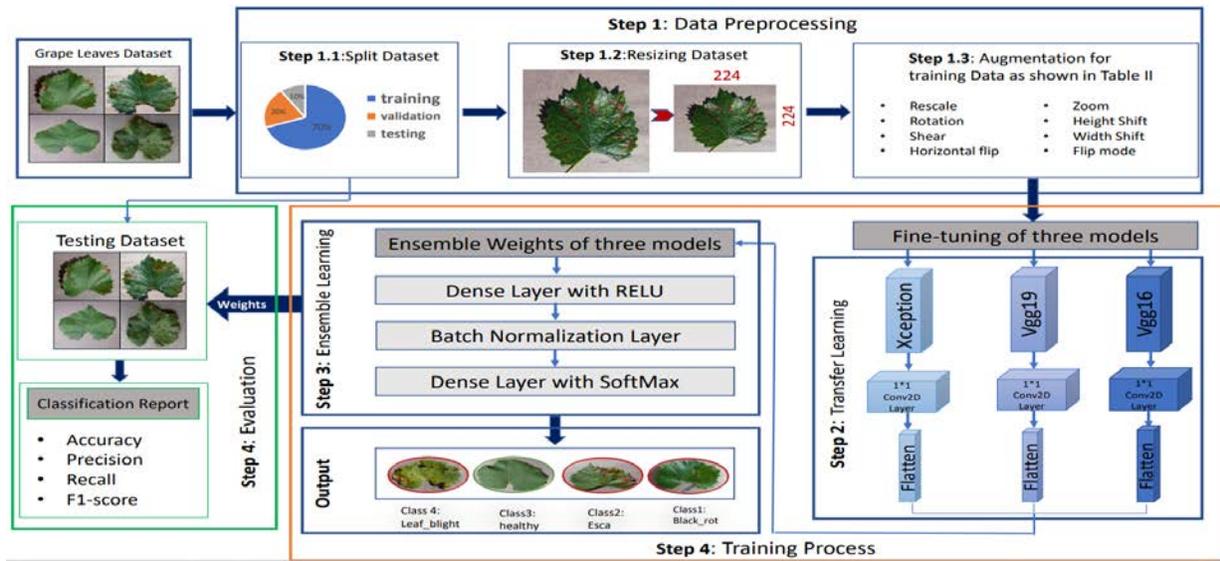


Fig. 5. The Framework of the Proposed Model.

1) *Data preprocessing*: Preprocessing of the dataset is one of the major roles in training a model. First, divided dataset into three categories: training 70%, validation 20%, and 10% for testing, as shown in Table I. Second, grape images are resized to 224*244*3 pixels. Third, applied data augmentation techniques to generate more training data and avoid overfitting. These techniques include rescale, rotation, shear, horizontal flip, zoom, height shift, width shift, and flip mode on the training set. The data augmentation is applied only to the training data shown in Table II.

TABLE I. DESCRIPTION OF PLANT VILLAGE DATASET

Grape Dataset						
Fig	Class Name	Cause of disease	Total samples	Training Samples (70%)	Validation Samples (20%)	Test Samples (10%)
1	Black Rot	Fungus	1180	826	236	118
2	Esca		1383	969	276	138
3	Leaf blight		1076	754	215	107
4	Healthy	-----	423	296	85	42

TABLE II. AUGMENTATION TECHNIQUES SUMMARY

Techniques	Values
Rescale	1./255
Rotation	30
Shear	0.1
Horizontal flip	True
Zoom	0.3
Height shift	0.2
Width shift	0.2
Flip mode	Nearest

2) *Model building*: Transfer learning is a great approach because it allows to use a pre-trained CNN model with multiple datasets to train a specific dataset.

VGG-16 and VGG-19 [15] refer to the "Visual Geometry Group". They're two different versions of the same structure. The differences between them are as follows, respectively: The VGG-16 consists of 16 layers of the deep neural network, whereas the VGG-19 consists of 19 layers. Both networks contain blocks, where each block is composed of 2D convolution and pooling layers. The Conv2D layer [16] extracts a feature of the image using filters or kernels. The filter is passed throughout the width and height of the input and the dot products function between the input and filter is calculated at every position. Convolution Layer Formula (1).

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{s} \right\rfloor + 1 \quad (1)$$

It also contains ReLU (Rectified Linear Activation Function) [17, 18] that returns all negative values set to zero (2). The function and gradient in ReLU (2) and (3).

$$\text{ReLU}(x) = \max(0, x) \quad (2)$$

$$\frac{d}{dx} \text{ReLU}(x) = 1 \text{ if } x > 0; \text{ otherwise } \quad (3)$$

The Conv2D layer is followed by a pooling layer to reduce the computation and the number of parameters. Max pooling (4) is one of the most used pooling operations. Then the matrix is flattened into a vector. The flattened vector is passed into the FC (Fully Connected) layer. FC is used to connect each node in one layer to each node in another layer. The last layer is SoftMax. It is located at the end of the FC layer, which predicts a multi-class.

$$S(x) = \max_{i=1}^N x_i \quad (4)$$

Xception [19] refers to Extreme Inception. First, the data passes through the entering flow, then eight times through the

middle flow, and finally through the exit flow. Batch normalization is applied to all convolution and separable convolution layers. Separable convolutions are time-saving and more efficient than classical convolutions.

The proposed model depends on a customized CNN with trained weights from the VGG16, VGG19, and Xception models. The learning scenario starts with receiving grape leaves images from the input layer. The input layer is shared with three pre-trained networks. Three models are reshaped by freezing all their layers. But we removed the top layer (output layer) from each model to add the proposed output layers.

Added two layers, a Conv 1x1 layer with 1024 filters with padding-zero and stride-one, to collect the most important features and allow for reduced dimensionality followed by a flattening layer to convert the matrix to the tensor of one dimension (vector). As illustrated in Fig. 6-8.

3) *Ensemble learning*: After applying the flattening layer to each model, added the merged layer to aggregate the flattening layer from each network and use it as an input (new input) for the ensemble learning model. Then, added a dense

layer that allowed every neuron in this layer to connect to the next layer by weight followed by a batch normalization layer, which allows each layer to make learning more independent. Finally, added the output layer with the SoftMax activation function to predict the final output.

4) *Training phase*: The proposed model compiled with the Adam optimizer [20] is a stochastic gradient descent method with a learning rate equal 2e-5. The loss function is a categorical cross-entropy [21] that is used in the multi-class classification task equation (5) to calculate the loss. The proposed model is fitted at epochs where value equals 10, train batch size equals 16, and validate batch size equals 8.

$$\text{Loss} = -\sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i \quad (5)$$

5) *Testing phase*: The best-saved weights were loaded after those testing images were loaded. Finally, images were resized to 224*224 and fed the testing images to the model to classify four classes of the grape leaves' diseases.

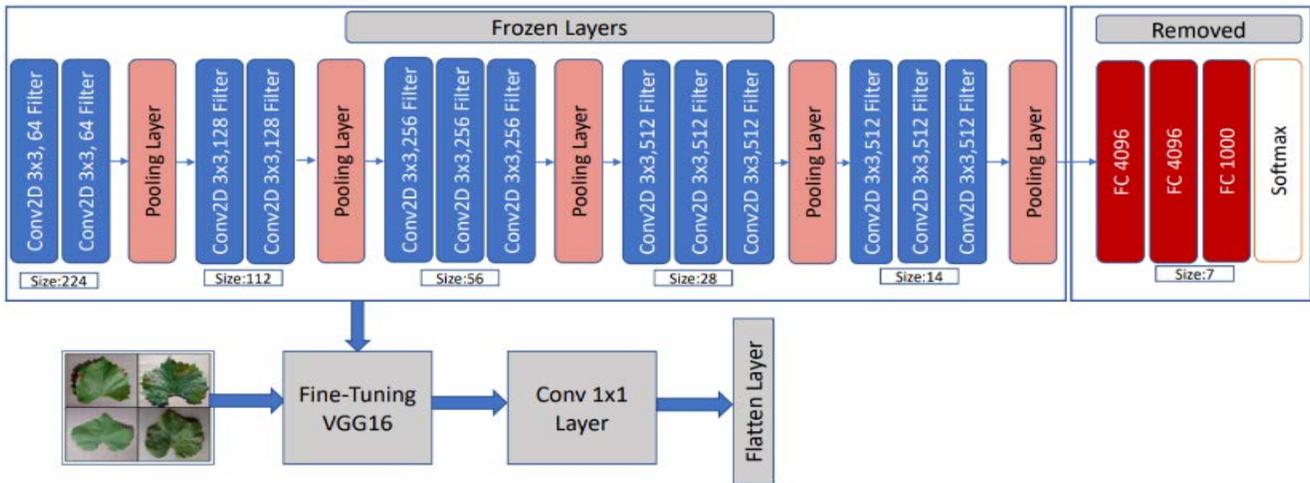


Fig. 6. Fined Tuned for VGG16.

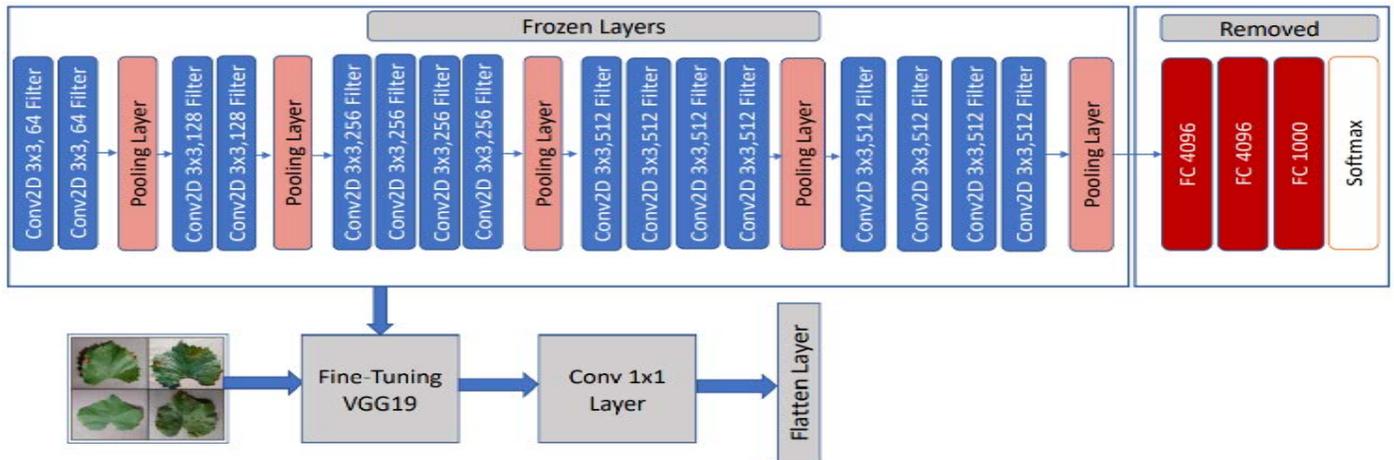


Fig. 7. Fined Tuned for VGG19.

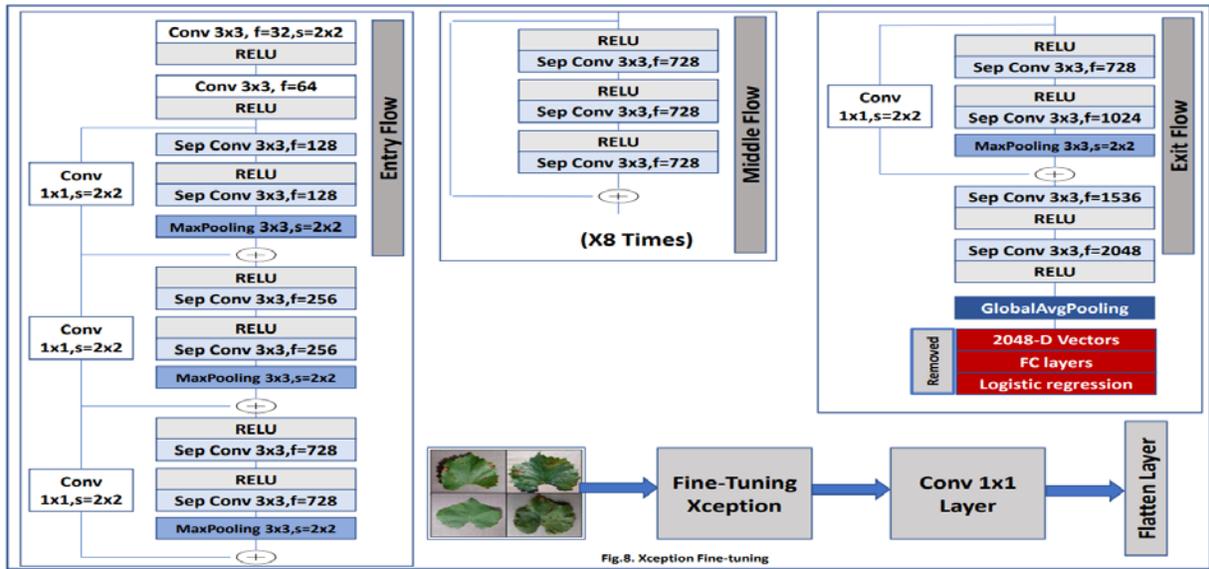


Fig. 8. Fined Tuned for Xception.

IV. EXPERIMENTAL WORK & RESULTS

A. Experimental Settings

This work demonstrates that the highest accuracy is achieved with a relatively small dataset of grape leaves in the plant village dataset using ensemble learning (e.g., VGG16 Model, VGG19, and Xception). The dataset is divided into a ratio of 7:2:1 as shown in Table I. All experiments are implemented using Colab [22] provided by Google, the Keras framework [23] that can run on top of TensorFlow, and the Python programming language. All experiments were conducted on a 12 GB NVIDIA Tesla K80 GPU (Graphical Processing Unit) and 12 GB of RAM.

B. Experimental Evaluation

The performance of the proposed model is evaluated using accuracy (9), precision (6), recall (7), F1-score (8), and confusion matrix in the testing phase equations shown in Table III. Moreover, loss, accuracy, validation loss, and validation accuracy are calculated during different epochs in the training phase, as shown in Fig. 10-17. Also, we compared our proposed model performance as shown in Table V with other models (e.g. [8]) that work on the same dataset as shown in Table IV and with each architecture used in the proposed model separately. Summary of the training model in the Fig. 9 to understand the underlying parameters.

TABLE III. PERFORMANCE EQUATIONS SUMMARY

Assessments	Equation	Equ.No
Precision (P)	$\frac{TP}{TP + FP}$	(6)
Recall (R)	$\frac{TP}{TP + FN}$	(7)
F1-Score (F)	$2 * \frac{P \cdot R}{P + R}$	(8)
Accuracy (Acc)	$\frac{TP + TN}{TP + TN + FP + FN}$	(9)

Where:

True Positive (TP) the model predicts the positive class correctly. True Negative (TN) model correctly classifies the negative class. In a false positive (FP), the model predicts the positive class incorrectly. In false negative (FN), the model predicts the negative class incorrectly.

The total training time for the proposed model is about 24 min 28 sec \pm 2 min with 10 epochs, and the test time is about 5 seconds. Other models in [8] use 20 epochs in training. However, our proposed method achieved the highest accuracy, showing the feasibility of our proposed method. Compared with existing methodologies, we can see that our proposed model has some degree of competition in terms of accuracy and precision, as shown in Tables IV and V. The robustness of our suggested approach is confirmed by these results.

TABLE IV. RESULT OF EXISTING MODELS

Model	Precision	Recall	F1-score	Accuracy	epochs
CNN [8]	94.60	94.58	94.56	94.58	20
VGG [8]	95.54	95.32	95.32	95.32	20
RESNET [8]	97.11	97.04	97.05	97.04	20
DENSENET [8]	98.31	28.27	98.28	98.27	20

TABLE V. CLASSIFICATION REPORT FOR PROPOSED MODELS

Model	Classes	Precision	Recall	F1-score	Support	Accuracy
Fined Tuned VGG16	Black Rot	1.00	0.92	0.96	118	0.96
	Esca	1.00	0.94	0.97	138	
	Healthy	0.98	1.00	0.99	107	
	Leaf blight	0.74	1.00	0.85	42	
Fined Tuned VGG19	Black Rot	1.00	0.97	0.98	118	0.99
	Esca	0.97	1.00	0.99	138	
	Healthy	1.00	1.00	1.00	107	
	Leaf blight	1.00	1.00	1.00	42	
Fined Tuned Xception	Black Rot	0.98	0.99	0.99	118	0.99
	Esca	0.99	0.99	0.99	138	
	Healthy	1.00	0.99	1.00	107	
	Leaf blight	0.98	1.00	0.99	42	
Ensemble Model (Proposed)	Black Rot	1.00	1.00	1.00	118	1.00
	Esca	1.00	1.00	1.00	138	
	Healthy	1.00	1.00	1.00	107	
	Leaf blight	1.00	1.00	1.00	42	

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 224, 224, 3)]	0	[]
vgg16 (Functional)	(None, None, None, 512)	14714688	['input_1[0][0]']
vgg19 (Functional)	(None, None, None, 512)	20024384	['input_1[0][0]']
xception (Functional)	(None, None, None, 2048)	20861480	['input_1[0][0]']
conv2d (Conv2D)	(None, 7, 7, 1024)	525312	['vgg16[0][0]']
conv2d_1 (Conv2D)	(None, 7, 7, 1024)	525312	['vgg19[0][0]']
conv2d_6 (Conv2D)	(None, 7, 7, 1024)	2098176	['xception[0][0]']
flatten (Flatten)	(None, 50176)	0	['conv2d[0][0]']
flatten_1 (Flatten)	(None, 50176)	0	['conv2d_1[0][0]']
flatten_2 (Flatten)	(None, 50176)	0	['conv2d_6[0][0]']
concatenate (Concatenate)	(None, 150528)	0	['flatten[0][0]', 'flatten_1[0][0]', 'flatten_2[0][0]']
dense (Dense)	(None, 100)	15052900	['concatenate[0][0]']
batch_normalization_4 (Batch Normalization)	(None, 100)	400	['dense[0][0]']
dense_1 (Dense)	(None, 4)	404	['batch_normalization_4[0][0]']

 Total params: 73,803,056
 Trainable params: 73,748,328
 Non-trainable params: 54,728

Fig. 9. Summary for the Proposed Model.



Fig. 10. Training and Validation Accuracy for VGG16.

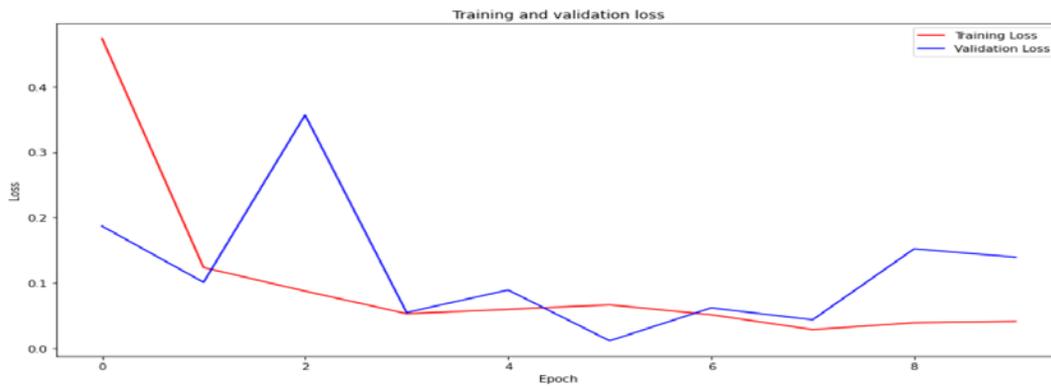


Fig. 11. Training and Validation Loss for VVG16.

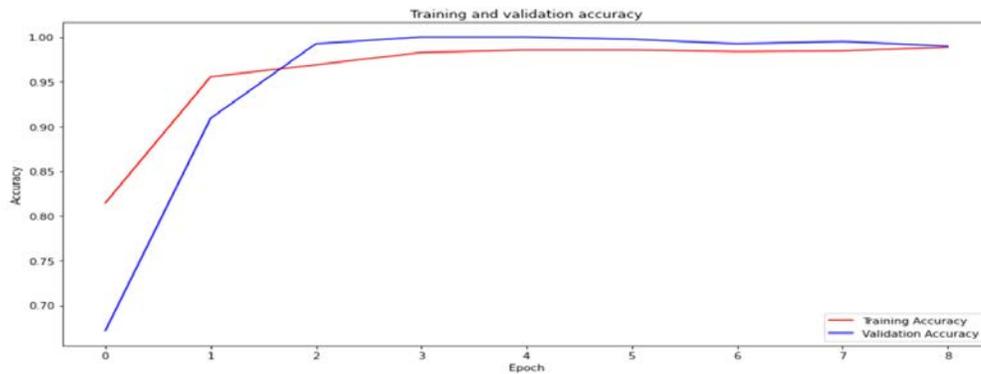


Fig. 12. Training and Validation Accuracy for VVG19.

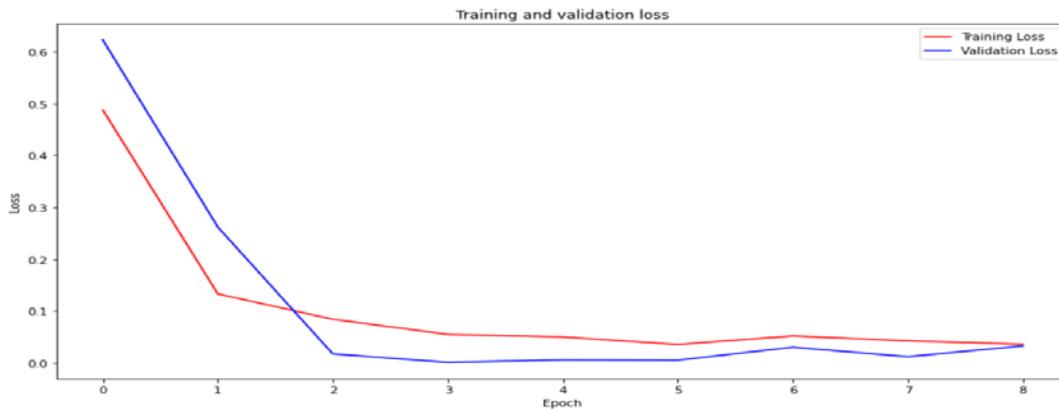


Fig. 13. Training and Validation Loss for VVG19.

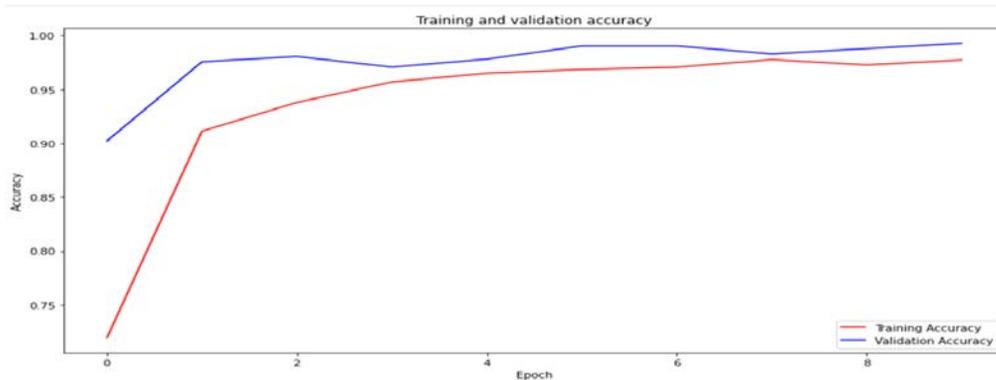


Fig. 14. Training and Validation Accuracy for Xception.

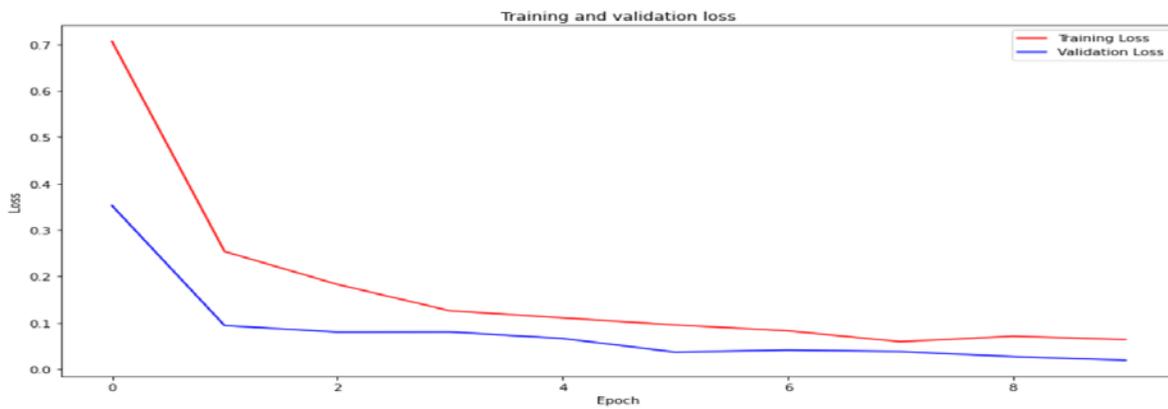


Fig. 15. Training and Validation Loss for Xception.

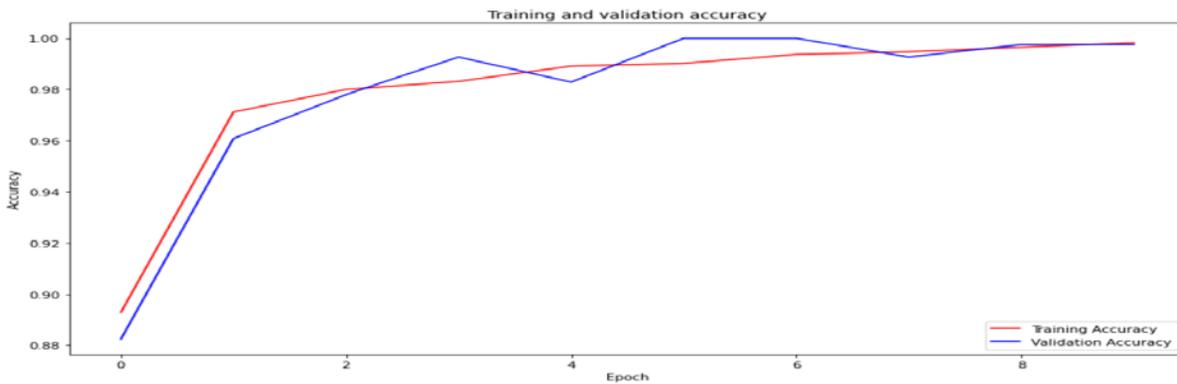


Fig. 16. Training and Validation Accuracy for Proposed Model.



Fig. 17. Training and Validation Loss for Proposed Model.

V. CONCLUSION

Visual observation for classifying grape leaf diseases can be misleading because of a lack of prior knowledge and similarities among diseases. This paper introduces an automated mechanism for classifying grape leaves as healthy or diseased (e.g., black measles, black rot, and leaf blight) using transfer learning (e.g., VGG16, VGG19, and Xception). This classification has been done by extracting the features from the grape images using different pre-trained networks and then using the ensemble learning method for these networks to enhance the diagnosis accuracy. The novelty of the work lies in the fact that, instead of training each network alone and then concatenating the final results to get better results, all the networks are trained together, allowing the fully connected

layer to discover the best ensemble method to concatenate the results of previous networks. The proposed model achieved an accuracy of 99.82% compared to recent models that used grape leaf disease in the training process across the existing online Plant Village dataset. This work demonstrated the value and benefits of using ensemble learning and transfer learning. This study has achieved its objectives to improve the performance of classifying plant diseases and reducing overfitting.

VI. FUTURE WORK

Looking ahead, we hope to test the power of our model on more complex datasets and increase the number of categories of agricultural diseases.

REFERENCES

- [1] "Expert meeting on how to feed the world in 2050," Fao.org. [Online]. Available: <https://www.fao.org/3/ak971e/ak971e.pdf>.
- [2] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using EfficientNet deep learning model," *Ecol. Inform.*, vol. 61, no. 101182, p. 101182, 2021.
- [3] S. Ghosh, A. Dasgupta, and A. Swetapadma, "A study on support vector machine based linear and non-linear pattern classification," in 2019 International Conference on Intelligent Sustainable Systems (ICISS), 2019.
- [4] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," in 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019.
- [5] R. Gebbers and V. I. Adamchuk, "Precision agriculture and food security," *Science*, vol. 327, no. 5967, pp. 828–831, 2010.
- [6] Z. J. Wang et al., "CNN explainer: Learning convolutional neural networks with interactive visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 1396–1406, 2021.
- [7] Ali, A. (2019). *PlantVillage Dataset* [Data set] [Online]. Available: <https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset>.
- [8] A. Kp and J. Anitha, "Plant disease classification using deep learning," in 2021 3rd International Conference on Signal Processing and Communication (ICPSC), 2021.
- [9] Y. Nagaraju, Venkatesh, S. Swetha, and S. Stalin, "Apple and grape leaf diseases classification using transfer learning via fine-tuned classifier," in 2020 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT), 2020.
- [10] E. Hirani, V. Magotra, J. Jain, and P. Bide, "Plant disease detection using deep learning," in 2021 6th International Conference for Convergence in Technology (I2CT), 2021.
- [11] K. Z. Thet, K. K. Htwe, and M. M. Thein, "Grape leaf diseases classification using convolutional neural network," in 2020 International Conference on Advanced Information Technologies (ICAIT), 2020.
- [12] Uky.edu.[Online]. Available: https://uknowledge.uky.edu/cgi/viewcontent.cgi?article=1182&context=anr_reports. [Accessed: 06-Apr-2022].
- [13] Jim Isleib, Michigan State University Extension, "Signs and symptoms of plant disease: Is it fungal, viral or bacterial?," *Field Crops*. [Online]. Available: https://www.canr.msu.edu/news/signs_and_symptoms_of_plant_disease_is_it_fungal_viral_or_bacterial. [Accessed: 06-Apr-2022].
- [14] "Grapes: Diseases and Symptoms," *Vikaspedia.in*. [Online]. Available: <https://vikaspedia.in/agriculture/crop-production/integrated-pest-managment/ipm-for-fruit-crops/ipm-strategies-for-grapes/grapes-diseases-and-symptoms>. [Accessed: 06-Apr-2022].
- [15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv [cs.CV]*, 2014.
- [16] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in 2017 International Conference on Engineering and Technology (ICET), 2017.
- [17] J. Brownlee, "A gentle introduction to the rectified linear unit (ReLU)," *Machine Learning Mastery*, 08-Jan-2019. [Online]. Available: <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>. [Accessed: 06-Apr-2022].
- [18] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," *arXiv [cs.NE]*, 2018.
- [19] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
- [20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv [cs.LG]*, 2014.
- [21] Y. Zhou, X. Wang, M. Zhang, J. Zhu, R. Zheng, and Q. Wu, "MPCE: A maximum probability based cross entropy loss function for neural network classification," *IEEE Access*, vol. 7, pp. 146331–146341, 2019.
- [22] E. Bisong, "Google Colaboratory," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, Berkeley, CA: Apress, 2019, pp. 59–64.
- [23] Keras Team, "Getting started," *Keras.io*. [Online]. Available: https://keras.io/getting_started/. [Accessed: 06-Apr-2022].