

CLASSIFICATION OF PLANT LEAF DISEASE USING ENSEMBLE DEEP TRANSFER LEARNING

Andrew Nader*

Computer Science Department,
Faculty of Information Systems and Computer
Science, October 6 University,
Giza, Egypt
Andrew.nader.cs@o6u.edu.eg

Mohamed H.Khafagy

Computer Science Department,
Faculty of Computers and
Information, Fayoum University,
Fayoum, Egypt
mhk00@fayoum.edu.eg

Shereen Ahmed

Computer Science Department,
Faculty of Computers and
Information, Fayoum University,
Fayoum, Egypt
sam26@fayoum.edu.eg

Received 2024-03-20; Revised 2024-03-20; Accepted 2024-03-26

Abstract: Crops can be affected by various types of pathogens that cause diseases, leading to significant damage and negative impacts on food production and quality, causing farmers financial losses. Therefore, it is essential to detect and prevent plant diseases at their initial stages promptly. Unfortunately, the process of detecting and controlling plant diseases can be challenging for farmers. However, deep learning techniques can potentially make a significant contribution by accurately classifying plant diseases at their earliest stages. To overcome the limitations of previous research, this work proposes a new method for diagnosing and classifying plant leaf diseases. The proposed approach enhances the classification of popular crops such as tomato, potato, and pepper by utilizing a dataset that comprises nine categories—three of which are healthy and the other six of which are infected—using transfer ensemble learning with MobilenetV3 small and Resnet50. The model has only 37 million training parameters, which saves time and computational power and reduces overfitting while achieving an accuracy of 99.50%. The model's ability to address false negatives and false positives ensures reliable and accurate plant disease classification at an early stage. The proposed method can work with any image dataset and achieve high performance, and it has been validated on the benchmark open dataset Plantvillage, demonstrating excellent performance for the proposed diagnosis approach.

Keywords: Convolutional Neural Network (CNN), Ensemble learning, Transfer learning, Deep Learning, Plant leaf disease classification.

1. Introduction

*Corresponding Author: Andrew Nader

Computer Science Department, Faculty of Information Systems and Computer Science, October 6 University, Giza, Egypt

Email address: Andrew.nader.cs@o6u.edu.eg

Agriculture is the foundation of the economy. Tomatoes, potatoes, and peppers are among the major commercial crops grown worldwide. Plant diseases and the limited utilization of new technologies have a detrimental effect on agricultural production with decreased crop quality and considerable economic losses.

Effective plant disease prevention and treatment begin with early detection. The approach used to monitor diseases is observation with the unaided eye, which is expensive, time-consuming, and requires a high level of knowledge. It is crucial to automatically identify illnesses without the aid of specialists [1].

The automated identification of plant diseases using plant leaves is a major advancement in agriculture. Furthermore, plant disease identification that is prompt and accurate improves crop quality and output [2].

Plant leaves illness usually manifests as visible lesions or scars on the leaves, stems, flowers, or fruits of the plant. Most pest illnesses and diseases have a characteristic visual pattern that can be utilized to pinpoint anomalies. Most disease indicators usually first show up on plant leaves, which are the primary means of recognizing disorders in plants [3].

Symptoms refer to the disease's physical effects on plants. Any noticeable change in a plant's shape, color, or ability as a result of a pathogen or disease-causing agent is referred to as a symptom. The leaves are the most important portion of the plant to look for the disease. Plant diseases can be classified into three categories: bacterial, viral, and fungal. [4].

Pathogenic fungi use a network of branching threads to form their bodies. Parasites can enter their host through stomata. These diseases' symptoms include Leaf blight, leaf rust, black rot, and black measles. Bacteria are quite tiny. There are 200 different types of bacteria that can afflict plants with illness. Their classification is determined by their form. The signs of a bacterial infection are numerous. The most common of these is known as leaf spot. Unlike bacteria and fungi, which are spread by water and wind, viruses are not. However, the main carriers of viral diseases in plants are worms and insects [5].

Plants can contract a variety of diseases and they are frequently difficult to diagnose, even by an agriculturist and pathologist. Still, the main method for diagnosing diseases is visual observation [6]. Farmers in rural places might have to make a costly and time-consuming trip to consult an expert. Many researchers have developed systems for automatic plant disease classification using feature sets for machine learning (ML) [7] to address the issues. A popular supervised machine learning technique used for classification is the support vector machine (SVM) [8]. All these methods, come with a few drawbacks. Accuracy is affected, for instance, by the limitations of feature extraction and the possibility that features extracted are insufficient for precise identity. The next step in machine learning growth is Deep Learning (DL) based features [9], particularly Convolutional Neural Networks, which are the most effective method for automatically learning discriminative and deciding features. Convolutional layers represent learning features from the data. Deep learning has some drawbacks, such as the requirement for a large amount of data to train the model. Performance suffers when a dataset is not large enough [10].

Transfer learning is a technique that uses a previously trained model as the foundation for a model on a new task [11]. Some advantages of using transfer learning are computational cost and decreased training time [12].

This study suggests an ensemble learning-based approach that aggregates two deep learning models trained by transfer learning (MobileNetV3Small and ResNet50) for classifying three different plant species (tomatoes, potatoes, and peppers), leaves of healthy plants and six categories of diseases.

Key challenges and issues are identified by the experts and researchers. As follows,

- Agricultural diseases have a lot of similarities, such as their texture and color, which cause confusion regarding correct recognition.
- Variations in lighting and background are the biggest challenges of precise identification.
- Different types of plants exhibit a variety of diseases, making disease identification challenging.
- Leaf image should be of high quality.
- A powerful computer or a lot of training time is needed if there are many training parameters. This means that the model needs to have fewer training parameters.

1.1 Study Hypotheses:

Hypothesis 1: Using deep learning algorithms with ensemble transfer learning techniques can improve the performance of plant disease classification. Specifically, the model can learn to recognize variations in leaf texture, shape, and color that are indicative of plant diseases by fine-tuning pre-trained convolutional neural networks on a dataset of plant leaf images. This approach produces a more reliable and accurate classification system than conventional machine learning techniques.

Hypothesis 2: By using dropout and early stopping techniques, the complexity and overfitting of plant disease classification models can be reduced. By randomly dropping out nodes during training and stopping training when the validation set's accuracy is stable, leading to improved generalization performance.

Hypothesis 3: By tracking the number of false negatives and false positives in classification models, can increase disease detection's precision and efficiency. By addressing both types of errors, the proposed approach can increase sensitivity of the model, reducing risk of disease spread and crop yield loss.

Considering the difficulties mentioned above, to achieve the best results, adjusted the network using various parameters and performed extensive testing by changing the various parameters. Precision, recall, f1-score, accuracy, test loss, test accuracy, training time, and confusion matrix were all taken into consideration while evaluating performance. Compared performance of proposed model with that of other state-of-the-art machine-learning techniques and deep learning models. The comparison and verification results show that the suggested model can identify healthy and diseased leaves with more performance and robustness than these other data-driven methods, helping to improve production and reduce crop loss.

The rest of the research is arranged as follows. Section 2 discusses the related research to the classify of automatic plant diseases. Section 3 explains proposed approach, materials, and datasets. Section 4 discusses the model's ability to correctly classify nine different classes and provides a discussion of the models. The paper's conclusion is given in Section 5.

2. Related Work

Using appropriate techniques for classifying plant leaf diseases contributes to production gains and loss management over crops. This section includes several current research methods that have produced beneficial results. Here are a few of these:

Hema [13] presented a method to identify 38 different types of plant leaf diseases from Kaggle dataset with help of deep convolutional networks. Transfer learning models such as VGG16 and Resnet34 are all trained. The filter size was 3x3. In VGG 16 and Resnet34, the filter sizes range from 64 to 512. Dataset was divided into ratio of 80%:20%, with an accuracy of 97.77 % and 97.58%, respectively. There was an issue with the overall performance of this research.

Tiwari, D. [14] In this study, only potato images were classified using a transfer learning approach such as VGG19 and logistic regression from the Kaggle plant village dataset, which is divided into the three classes of Healthy Leaf, Early Blight, and Late Blight. Dataset is divided into 70%:30%. Model achieves classification accuracy of 97.8% over test dataset. The issue with this work was that, despite utilizing 143 million characteristics, the accuracy was limited to 97.8%.

Khamparia [15] in this study, Deep Convolutional Encoder Network method for crop disease identification. There are three crops: maize, tomato, and potato, with five diseases and one healthy. Additionally, they noted that system utilized roughly 3.3 million parameters. While their model's testing accuracy was 86.78%, they were able to achieve 100% training accuracy. Training accuracy was much higher than the testing accuracy, there was a chance that the trained model overfit the training set.

Nader, Andrew [16] This work combines the concept of ensemble learning with deep learning using three popular convolutional neural network models (VGG16, VGG19, and Xception). Grape leaf diseases is used to evaluate the suggested method. The dataset was divided into 70% training, 20% validation, and 10% testing. Accuracy was 99.82%, but with many parameters, it is approximately 173 million.

Sanga et al. [17] this work proposed banana disease detection with CNN models (ResNet-18, ResNet-152 VGG-16, ResNet-50, and InceptionV3). Dataset was split into 80% training, 15% validation, and 5% testing. The image size for Resnet is 224 x 244 pixels and 299 x 299 pixels for InceptionV3. Using SGD Optimizer, the ResNet-152 outperformed the others with 99.2% accuracy, but with many parameters, it is approximately 60 million.

M. Bhagat. [18] CNN has been used in this study to identify bacterial patches on pepper leaves. Suggested workflow entails acquiring images, preprocessing them, segmenting them, extracting features, and categorizing plant diseases. A hundred leaves have been tested. The images are taken in an environment with good lighting for classifying two types of leaves: healthy or infected. The proposed model achieves a test accuracy of 96.78%. This work had a problem with test accuracy.

2.1 Related Work Summary:

Previous studies have utilized machine learning, deep learning, and transfer learning techniques to classify and identify plant diseases using Plant Village dataset. However, these studies often encounter accuracy issues and fail to address the reduction of false negative errors.

One common problem in these studies is the absence of a validation set during the data partitioning process. A validation set allows for the model to be evaluated during training, decreasing overfitting, and guaranteeing the generalizability of the model to new data.

Another issue in these studies is the insufficient focus on false negatives, where the model fails to identify a diseased plant, leading to disease spread and crop yield loss. Addressing both false positives and false negatives is critical for accurate and reliable disease detection.

Additionally, a common issue in earlier studies has been the abundance of parameters in the models. A high number of parameters might result in overfitting and poor generalization performance. The number of parameters is directly correlated with the complexity of the model.

In conclusion, reducing the number of parameters in the model, resolving false positives and false negatives, and adding a validation set are the main ways to increase the reliability and accuracy of plant disease classification models.

3. Proposed Methodology

3.1 Dataset Description

The Plantvillage dataset [19] is a massive and freely accessible database. There are 38 classes and nearly 55,000 RGB images, which represent 14 species of plants. Three different crops (tomato, potato, and pepper) are selected from the Plant Village, which contains 17,002 leaf images which are distributed into 9 classes: 3 healthy classes and 6 infected are shown in Table 1. Examples of a healthy and a diseased leaf image of the three crops shown in Figure. 1. In this work, focuses on two types of pathogens: bacteria and fungi. To carry out experiments and evaluate the algorithm.

Table 1 Description of Dataset Used

Host	Disease Name	Pathogen	Causes	No. Image
Tomato	Bacterial_Spot	Bacterial	Xanthomonas perforans	1702
	Early_Blight	Fungal	Alternaria sp.	1920
	Late_Blight	Fungal	Phytophthora infestans	1851
	Healthy	-	-	1926
Potato	Early_Blight	Fungal	Alternaria solani	1939
	Late_Blight	Fungal	Phytophthora infestans	1939
	Healthy	-	-	1824
Pepper	Bacterial_Spot	Bacterial	Xanthomonas campestris pv.	1913
	Healthy	-	-	1988



Figure. 1: Sample of Dataset

3.2 Proposed Method

In this study, a powerful approach focuses on two popular transfer learning architectures and ensemble learning, namely ResNet50 [20], and MobileNetV3 small [21], the overall architecture of the proposed model with three main phases is presented as shown in Figure. 2 and algorithm 1.

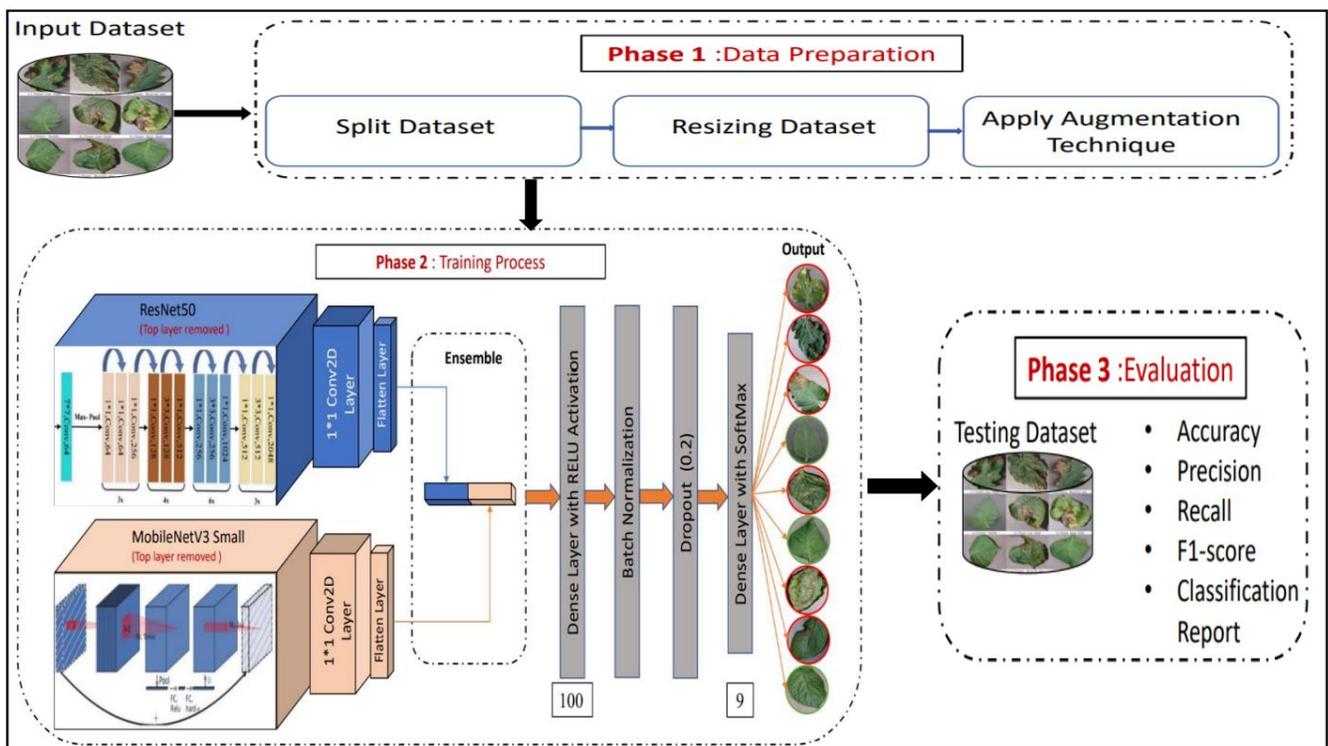


Figure. 2: Architecture of Proposed Model

Algorithm 1: Proposed Model

Pseudocode for the proposed method	
1:	INPUT: <i>Leaf Images of three crops.</i>
2:	OUTPUT: <i>Disease Classification for Three Crops.</i>
3:	Start Procedure
4:	Procedure (<i>Preprocessing</i>)
5:	<i>Input: leaf images of three crops (Tomato, Potato, and Pepper).</i>
6:	<i>Separate dataset into validation, testing, and training.</i>
7:	<i>Resize images to 224x224 pixels.</i>
8:	<i>Apply augmentation techniques including rescaling, rotation, height, and width shifting, zooming, horizontal flip, and brightness range.</i>
9:	End Procedure
10:	Procedure (<i>Building Model</i>)
11:	<i>Load the weights by using the ResNet50 TL model</i>
12:	<i>Freeze the above layers, but don't load the output (fully connected) layers.</i>
13:	<i>Add a 1 x 1 convolution layer with 1024 filters followed by a flattening layer.</i>
14:	<i>Repeat Steps 11, 12, and 13 using the MobileNetV3 Small TL model instead of the ResNet50.</i>
15:	End Procedure
16:	Procedure (<i>Ensemble and FC</i>)
17:	<i>Ensemble the two flattened layers, flat1_ResNet50, and flat2_MobileNetV3.</i>
18:	<i>Add a dense layer that contains 100 neurons with a ReLU activation function, Batch Normalization, Dropout, and a dense layer that contains 9 neurons with a SoftMax activation function.</i>
19:	End Procedure
20:	End Procedure

Phase 1: Data Preparation

Preprocessing datasets is essential when training a model to increase accuracy. The first step Separating the data into three categories: (70%) training, (20%) validation, and (10%) testing. If the training proportion of the data is altered, however, the results may slightly differ. This is done to ensure that model is trained on a sufficiently large dataset while also having enough data to validate its performance and test its accuracy. In the second step, dataset images are resized to 224x224 pixels to be compatible with the pre-trained models used (ResNet50 and MobileNetV3 small). This is a common practice when using pre-trained models, as they often have specific input size requirements. Finally, data augmentation techniques are applied to generate new data from the existing data by creating new variations. Data augmentation is useful in avoiding overfitting and improving performance [22], and it is applied only to the training data to ensure that the model is evaluated on the original, unmodified data during validation and testing, as shown in Table 2.

Table 2 Augmentation Techniques

Techniques	Values
Rescale	1./255
Rotation	30
Height shift	0.4
Width shift	0.4
Brightness range	1.0,2.0
Shear	0.1
Horizontal flip	True
Zoom	0.3
Flip mode	Nearest

Phase 2: Training Process

The proposed model relies on a modified architecture of pre-trained models, namely ResNet50 and MobileNetV3 small. In the learning scenario, input layer receives dataset that is shared with the two pre-trained transfer learning models. Using pre-trained models, the deep learning model can start with a better understanding of the data and learn more quickly, helping it to make more accurate predictions and increase its accuracy. The layers of the two models are frozen, except for the output layer, which is modified to add the proposed final layer. A 1x1 Conv layer with 1024 filters is added to extract the most important features from the input data, followed by a flattened layer to combine the extracted features and use them as new input. An ensemble layer is then added to merge the flattened layer and use it as a new input. Next, fully connected layers with 100 neurons and the ReLU activation function are added, followed by batch normalization and a dropout rate of 0.2. Finally, fully connected layers with 9 neurons and the SoftMax activation function are added to predict the multi-classification final output. A summary of the layers used in the proposed model is shown in Table 3.

3.3 Layers of Proposed Model

A. Convolution layer and ReLU activation function

In convolution layers, features are taken out of an image using a kernel. A kernel is a matrix that is applied to the image and multiplied with the input in order to improve the output in a desired way [23]. The ReLU Rectified Linear Unit is added after the convolution operation used to increase the non-linearity in our images [24] this is done by applying a mathematical equation (eq.1 and eq.2).

$$\text{ReLU}(x) = \max(0, x) \quad (1)$$

$$\frac{d}{dx} \text{ReLU}(x) = 1 \text{ if } x > 0; \text{ otherwise} \quad (2)$$

B. Pooling layers

Pooling layers, such as max pooling, which takes the maximum value in a specific filter zone, or average pooling, which takes the average value in a filter zone, are used to reduce the size of the input data for feature reduction. The Max Pooling and Average Pooling layers are used in the proposed model [25].

C. Fully connected layer and SoftMax

In the fully connected layers (FC), the output from the final pooling or convolutional layer is flattened and then fed into the fully connected layer as input for the FC layer. Last layer performs multi-classification using SoftMax activation function. [26]. The formula is as follows in eq.3:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \tag{3}$$

In Eq. 3, where every z_i value can take any real value and represents a member of the input vector. The term at the bottom of the calculation assures that all the function's output values will sum to 1, where \vec{z} the input vector to the SoftMax and K is the number of classes.

To calculate the output tensors dimensions from the input tensor The formula is as follows in eq.4.

$$W_{out} = \frac{W_{in} - F + 2P}{S} + 1 \tag{4}$$

Where,

- W_{in} : Input tensor
- F : width / height of the kernel
- P : Padding
- S : Stride
- W_{out} : Output width / height

D. Batch Normalization and Dropout

Batch normalization is a layer that allows every layer to learn independently. It is used as a regularization to avoid overfitting. Dropout is a regularization technique that is used to prevent overfitting in the model. Dropouts should be applied after batch normalization and are typically advised to be used after dense layers of the network [27].

Table 3 Summary of Layers

Layers	Action	Parameters
Convolution	Apply filters to extract features.	Number of kernel and size of it. Activation function. Stride and padding. Regularization.
Pooling	Reduce dimensionality	Stride. Size of window.
Fully connected	Combined information from the final feature maps. Final classification.	Number of nodes. Activation function if multiclassification using SoftMax.

5. Experimental Work Results and Discussion

Phase 3: Evaluation

This section evaluates performance of the proposed model using dataset of three different crops (tomato, potato, and pepper) with six diseases and three healthy classes of the plant leaves. The number of images used to train and evaluate the proposed model is 17,002. All experiments were trained using a Collaboratory (Colab) [28] that provides everyone with free GPU resources, with 12 GB of NVIDIA Tesla K80 and 12 GB of RAM using Keras API of TensorFlow [29].

5.1 Performance Measures

The effectiveness of the proposed model has been evaluated using different parameters such as accuracy (5), precision (6), recall (7), F1_score (8), and confusion metrics [30]. Time, loss, accuracy, validation accuracy, validation loss, and validation accuracy are calculated over different training phase epochs.

Accuracy is used to evaluate the overall accuracy of a model as defined in Eq. (5).

$$\text{Accuracy (Acc)} = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

Precision indicates the number of positive samples that belong to the positive class as defined in Eq. (6).

$$\text{Precision (P)} = \frac{TP}{TP+FP} \quad (6)$$

Recall indicates number of actual positive samples was correctly identified as defined in Eq. (7).

$$\text{Recall(R)} = \frac{TP}{TP+FN} \quad (7)$$

F1 score combines recall and precision together as defined in Eq. (8)

$$\text{F1-Score (F)} = 2 * \frac{P*R}{P+R} \quad (8)$$

Where:

True Positive (TP) The model accurately predicts the positive class. True Negative (TN) The negative class is correctly classified by the model. False Positive (FP), The model inaccurately predicts the positive class. False Negative (FN), The model inaccurately predicts the negative class.

Table 4 shows the performance evaluation of the proposed approach with fine-tuned Resnet50, MobilenetV3 small, and proposed model with a dataset distribution of an 8:1:1 ratio and a dataset distribution of a 7:2:1 ratio. [Figure. 3: Figure.14] displays the loss, accuracy, validation loss, and validation accuracy for experiments one and two during different epochs in the training. The confusion matrix of all experiments is shown in [Figure. 15 : Figure. 20]. As shown in Table 5, the proposed model was also compared with five existing methodologies, and it was found to outperform the existing approaches in terms of accuracy, precision, recall, and F1 score.

Table 4 Performance Measures for Experiment One and Two

Models	Classes	Precision	Recall	F1_score	Precision	Recall	F1_score	Support
ResNet50	-	Train 80% Validation 10% Test 10% (experiment 1)			Train 70% Validation 20% Test 10% (experiment 2)			-
	Pepper_bell_healthy	1.00	0.99	1.00	0.99	1.00	0.99	192
	Potato_Late_blight	1.00	1.00	1.00	1.00	0.99	0.99	200
	Tomato_Early_blight	0.99	1.00	0.99	0.99	0.99	0.99	195
	Tomato_Late_blight	1.00	0.99	0.99	1.00	0.94	0.94	195
	Potato_Early_blight	0.99	1.00	0.99	0.97	1.00	1.00	184
	Tomato_Bacterial_spot	0.97	0.99	0.98	0.99	0.99	0.99	170
	Tomato_healthy	1.00	0.92	0.96	1.00	0.96	0.96	192
	Potato_healthy	0.95	1.00	0.98	0.93	1.00	1.00	186
	Pepper_Bacterial_spot	0.99	1.00	1.00	0.99	1.00	1.00	193
MobileNetV3 Small	Pepper_bell_healthy	0.99	0.98	0.99	0.96	1.00	0.98	192
	Potato_Late_blight	0.96	0.99	0.98	0.99	0.99	0.99	200
	Tomato_Early_blight	0.96	0.98	0.97	0.99	0.99	0.99	195
	Tomato_Late_blight	1.00	0.83	0.91	0.96	0.97	0.96	195
	Potato_Early_blight	0.96	0.99	0.97	0.95	0.99	0.97	184
	Tomato_Bacterial_spot	0.87	0.96	0.91	0.93	0.90	0.91	170
	Tomato_healthy	0.97	0.79	0.87	0.89	0.90	0.89	192
	Potato_healthy	0.85	0.97	0.91	0.99	0.90	0.94	186
	Pepper_Bacterial_spot	0.95	1.00	0.97	0.98	1.00	0.99	193
Proposed Model	Pepper_bell_healthy	1.00	0.98	0.99	1.00	0.99	1.00	192
	Potato_Late_blight	0.98	1.00	0.99	1.00	1.00	1.00	200
	Tomato_Early_blight	1.00	1.00	1.00	1.00	0.99	1.00	195
	Tomato_Late_blight	1.00	0.96	0.98	1.00	0.99	1.00	195
	Potato_Early_blight	0.98	1.00	0.99	0.99	1.00	1.00	184
	Tomato_Bacterial_spot	0.99	0.99	0.99	0.99	0.99	0.99	170
	Tomato_healthy	1.00	0.96	0.98	1.00	0.97	0.98	192
	Potato_healthy	0.96	1.00	0.98	0.97	0.99	0.98	186
	Pepper_Bacterial_spot	0.98	1.00	0.99	0.99	1.00	0.99	193

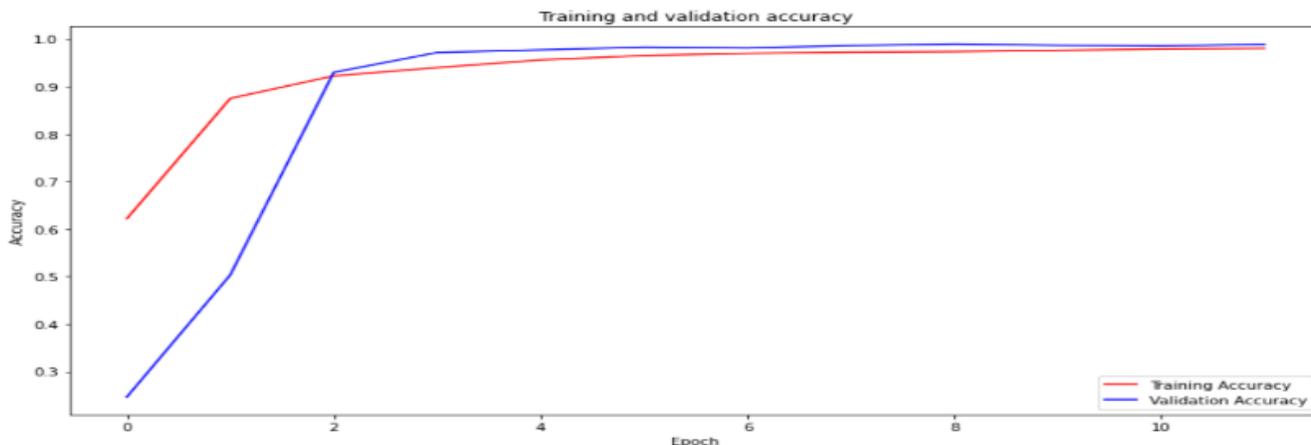


Figure. 3: ResNet Training and Validation Accuracy for Experiment 1.

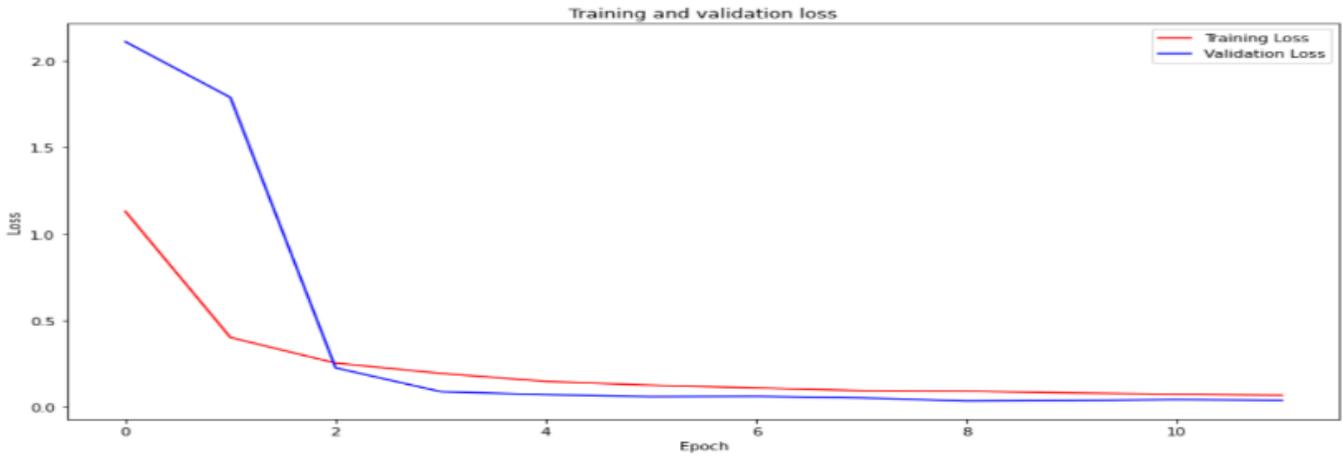


Figure. 4: ResNet Training and Validation Loss for Experiment 1.

Figure. 3 and Figure. 4 Observation: The accuracy and loss during training and validation remain stable over epochs, indicating that the model has learned the most important features of the training data and can generalize well to new data. We used early stopping technique and patience is set to 3, the model stopped in epochs 12, indicating that there was no improvement in its performance after epochs 12.

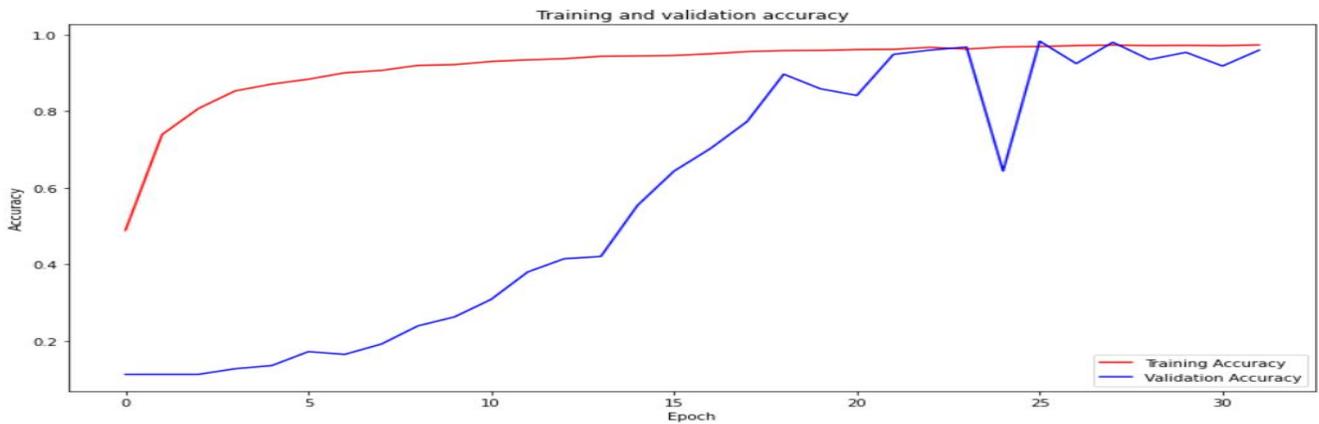


Figure. 5: MobileNet Training and Validation Accuracy for Experiment 1.

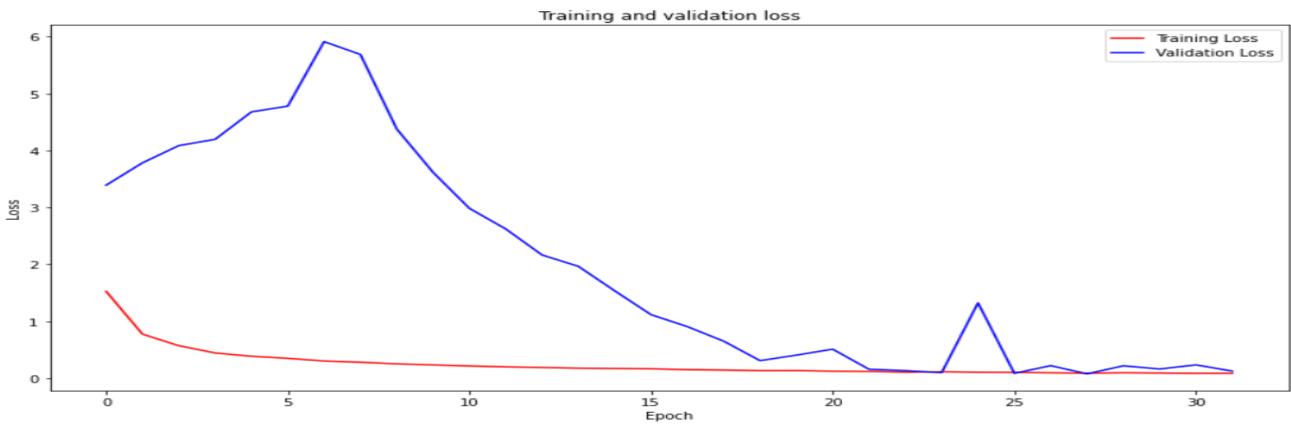


Figure. 6: MobileNet Training and Validation Loss for Experiment 1.

Figure. 5 and Figure. 6 Observation: The graph of the fine-tuned MobileNet small model shows considerable variations, which means that the model is still learning and has not yet reached a stable solution. The loss generally decrease over time as the model improves its performance. We used early stopping technique and patience is set to 3, the model stopped in epochs 35, indicating that there was no improvement in its performance after epochs 35. Although the model used many epochs, it did not achieve good results.

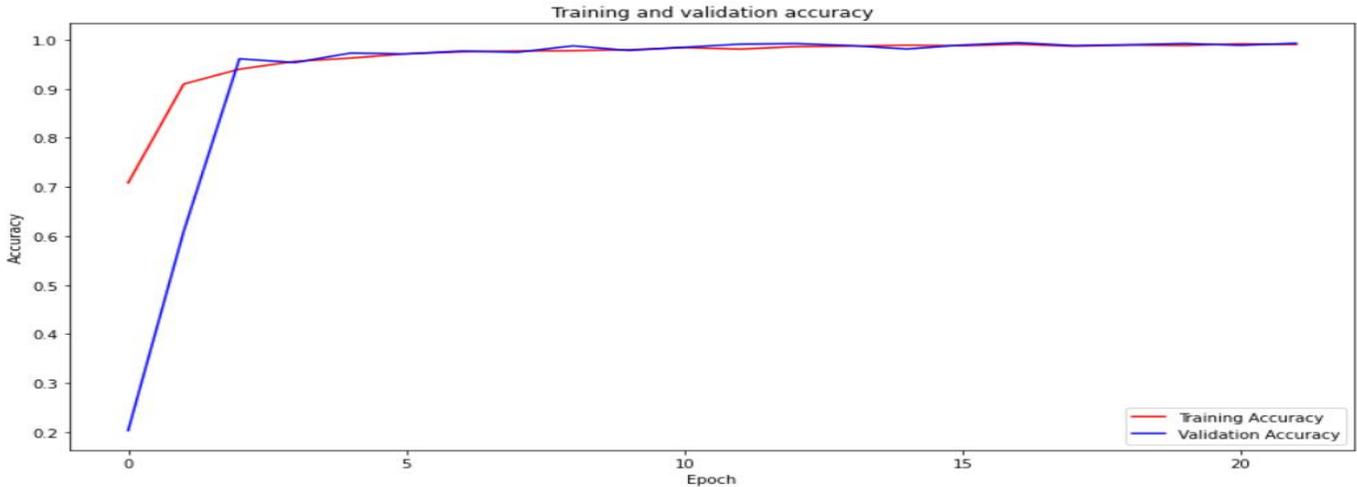


Figure. 7: Proposed Model Training and Validation Accuracy for Experiment 1.

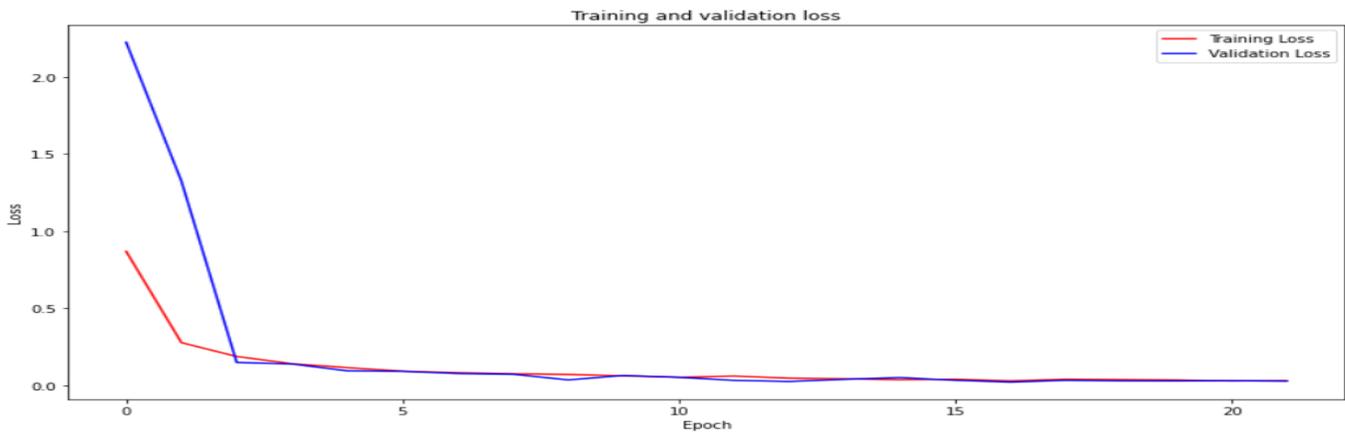


Figure. 8: Proposed Model Training and Validation Loss for Experiment 1.

Figure. 7 and Figure. 8 Observation: A stable and consistent accuracy and loss graph during training and validation is a sign that the model is working effectively and can make accurate predictions. Used early stopping technique and patience is set to 3, the model stopped in epochs 25, indicating that there was no improvement in its performance after epochs 25.

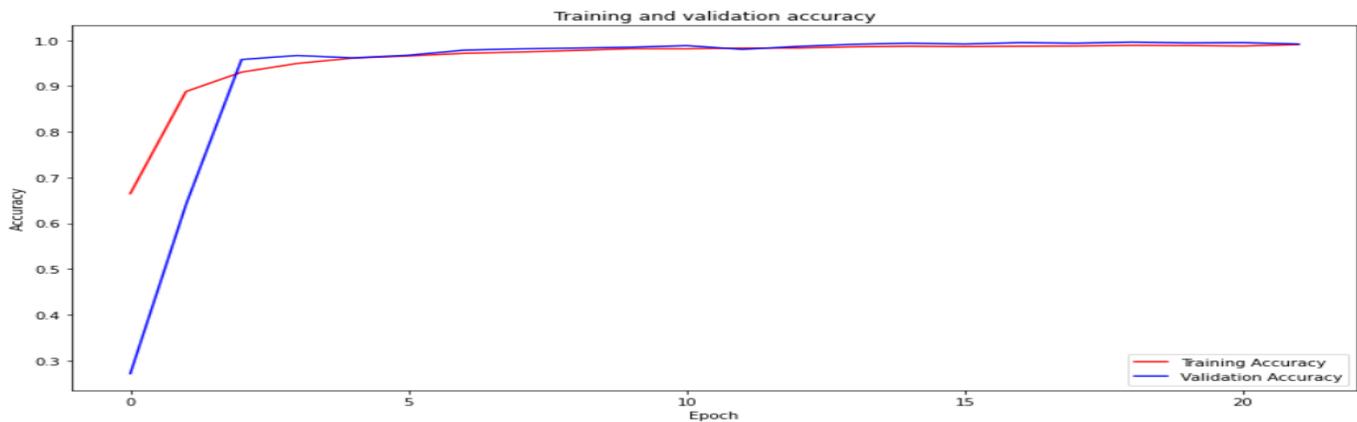


Figure 9: ResNet Training and Validation Accuracy for Experiment 2.

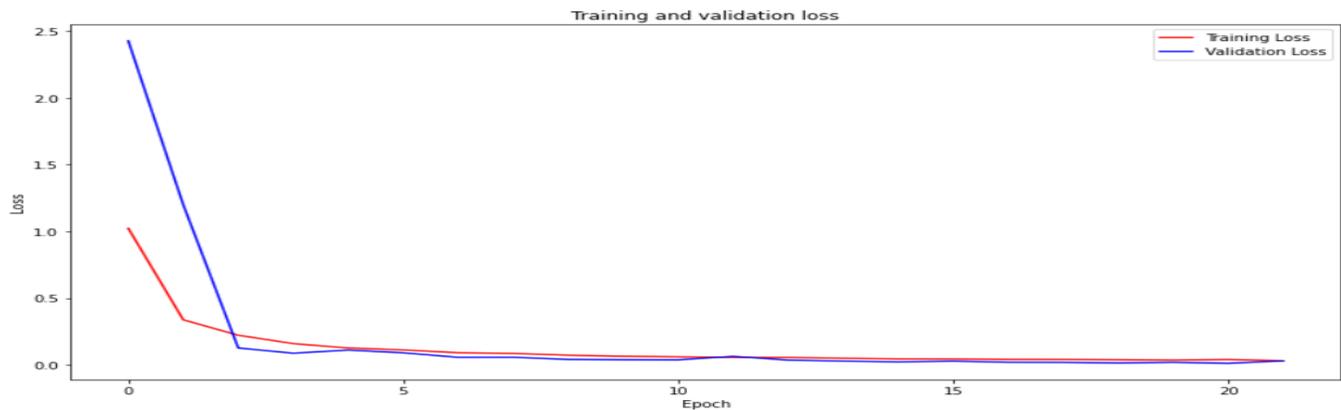


Figure 10: ResNet Training and Validation Loss for Experiment 2.

Figure 9 and Figure 10 Observation: The accuracy and loss during training and validation remain stable over epochs, indicating that the model has learned the most important features of the training data and can generalize well to new data. We used early stopping technique and patience is set to 3, the model stopped in epochs 25, indicating that there was no improvement in its performance after epochs 25.

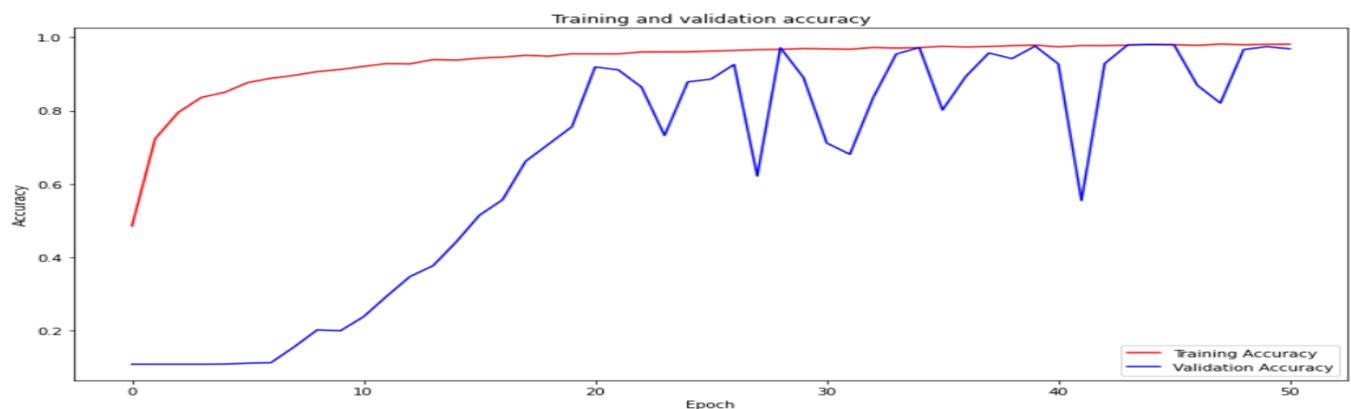


Figure 11: MobileNet Training and Validation Accuracy for Experiment 2.

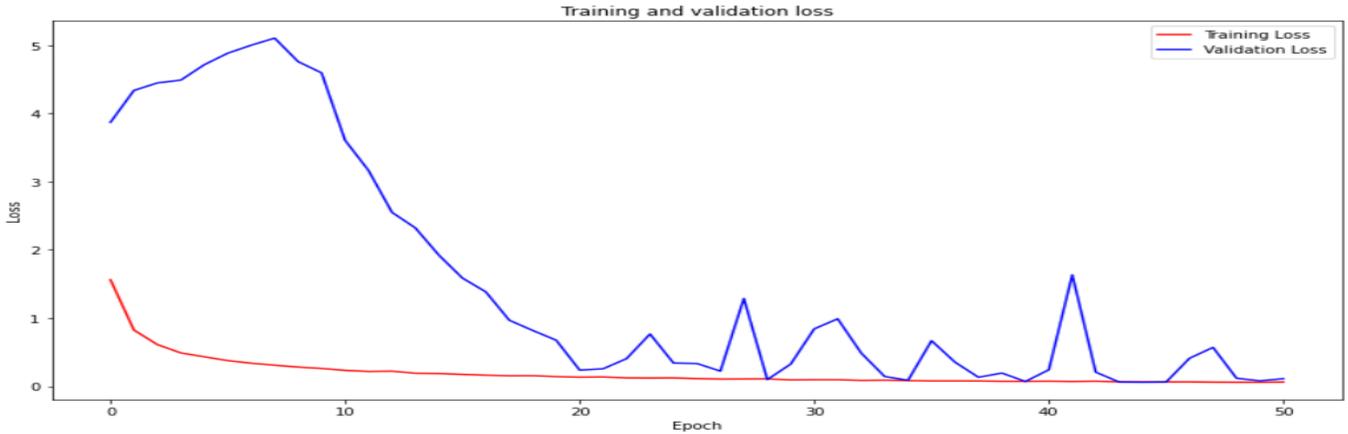


Figure. 12: MobileNet Training and Validation Loss for Experiment 2.

Figure. 11 and Figure. 12 Observation: The graph shows considerable variations, which means that the model is still learning and has not yet reached a stable solution. We used early stopping technique and patience is set to 3, the model stopped in epochs 50, indicating that there was no improvement in its performance after epochs 50. Although the model used a large number of epochs, it did not achieve good results.

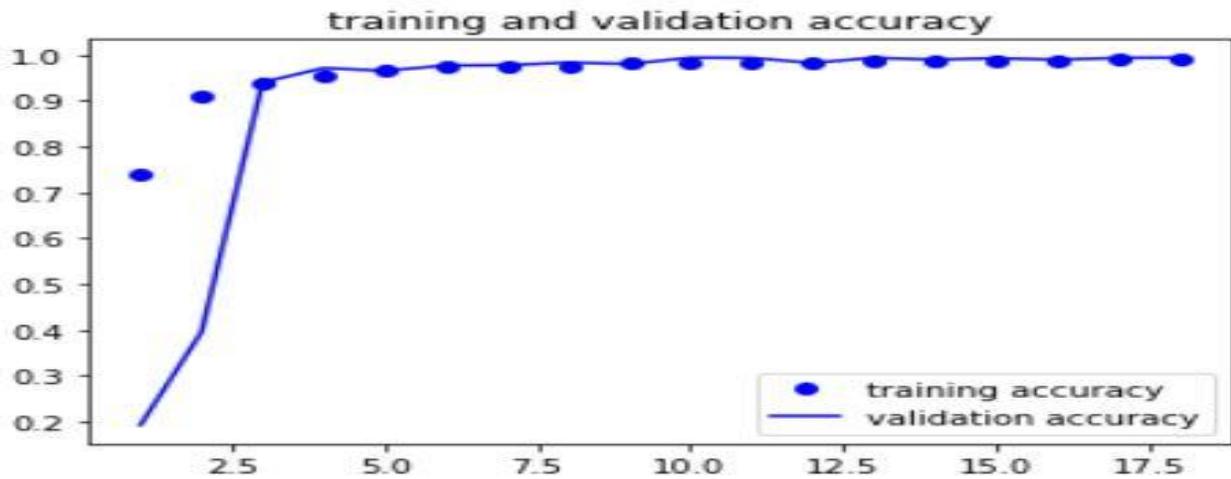


Figure. 13: Proposed Model Training and Validation accuracy for Experiment 2.

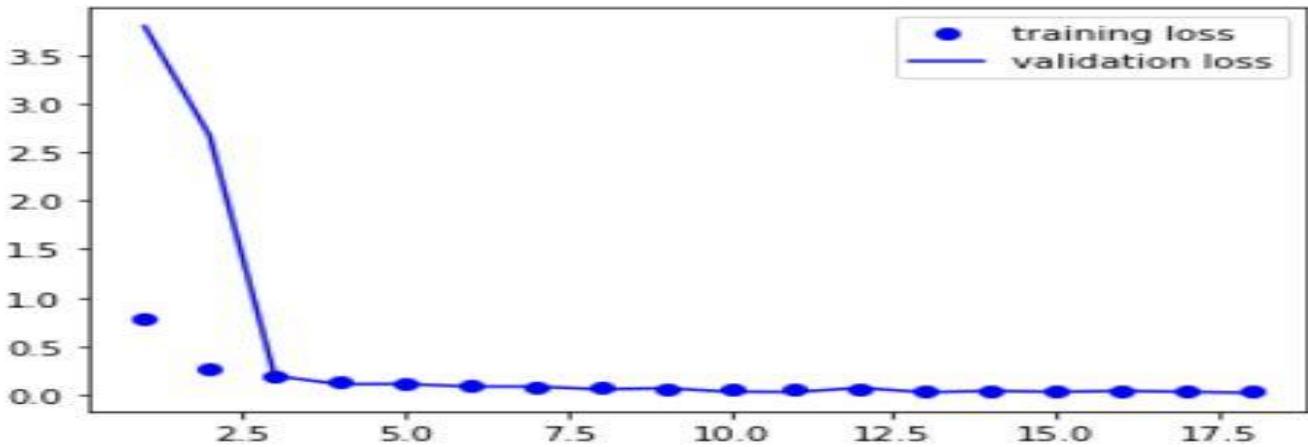


Figure.14: Proposed Model Training and Validation loss for Experiment 2.

Figure. 13 and Figure.14 Observation: The accuracy and loss during training and validation remain stable over epochs, indicating that the model has learned the most important features of the training data and is capable of generalizing well to new data. We used early stopping technique and patience is set to 3, the model stopped in epochs 18, indicating that there was no improvement in its performance after epochs 18.

Output Classes	PepperHealthy	191	1	0	0	0	0	0	0	99.47%	
	PotatoLateBlight	0	200	0	0	0	0	0	0	100%	
	TomatoEarlyBlight	0	0	195	0	0	0	0	0	100%	
	TomatoLateBlight	0	0	0	193	2	0	0	0	98.97%	
	PotatoEarlyBlight	0	0	0	0	184	0	0	0	100%	
	TomatoBacterial	0	0	0	0	0	169	0	1	99.41%	
	TomatoHealthy	0	0	2	0	0	5	176	8	91.66%	
	PotatoHealthy	0	0	0	0	0	0	0	186	100%	
	PepperBacterial	0	0	0	0	0	0	0	0	193	100%
											98.82%

Figure. 15: ResNet50 Confusion Matrix Experiment 1.

Figure. 15 Observation: Potato Late Blight (FN) = 1, Tomato Early Blight (FN) = 2, Potato Early Blight (FN) = 2, Tomato Bacterial (FN) = 5, Potato Healthy (FN) = 9, and Pepper Bacterial (FN)= 1.

Output Classes	PepperHealthy	189	3	0	0	0	0	0	0	98.43%	
	PotatoLateBlight	0	199	0	0	1	0	0	0	99.5%	
	TomatoEarlyBlight	1	0	191	0	0	0	0	3	97.94%	
	TomatoLateBlight	0	0	6	162	4	5	0	17	83.07%	
	PotatoEarlyBlight	0	2	0	0	182	0	0	0	98.91%	
	TomatoBacterial	0	1	0	0	3	163	1	1	95.88%	
	TomatoHealthy	1	2	1	0	0	19	151	10	8	78.64%
	PotatoHealthy	0	0	0	0	0	1	4	181	0	97.31%
	PepperBacterial	0	0	0	0	0	0	0	0	193	100%
											94.37%

Figure. 16: MobileNet Confusion Matrix Experiment 1.

Figure. 16 Observation: Pepper Healthy (FN) =2, Potato Late Blight (FN) =8, Tomato Early Blight (FN) = 7, Potato Early Blight (FN) = 8, Tomato Bacterial (FN)= 25, Tomato Healthy = 5, Potato Healthy (FN) = 31, and Pepper Bacterial (FN)= 10.

Output Classes	PepperHealthy	188	4	0	0	0	0	0	0	97.91%	
	PotatoLateBlight	0	200	0	0	0	0	0	0	100%	
	TomatoEarlyBlight	0	0	195	0	0	0	0	0	100%	
	TomatoLateBlight	0	0	0	187	4	0	0	4	95.89%	
	PotatoEarlyBlight	0	0	0	0	184	0	0	0	100%	
	TomatoBacterial	0	0	0	0	0	169	0	1	99.41%	
	TomatoHealthy	0	0	0	0	0	2	185	2	3	96.35%
	PotatoHealthy	0	0	0	0	0	0	0	186	0	100%
	PepperBacterial	0	0	0	0	0	0	0	0	193	100%
											98.82%

Figure. 17: Proposed Model Confusion Matrix Experiment 1.

Figure. 17 Observation: Potato Late Blight (FN) = 4, Potato Early Blight (FN) = 4, Tomato Bacterial (FN) = 2, Potato Healthy (FN) = 7, and Pepper Bacterial (FN)= 3.

Output Classes	PepperHealthy	192	0	0	0	0	0	0	0	100%	
	PotatoLateBlight	2	198	0	0	0	0	0	0	99%	
	TomatoEarlyBlight	0	0	194	0	1	0	0	0	99.48%	
	TomatoLateBlight	0	0	1	183	4	0	0	7	93.84%	
	PotatoEarlyBlight	0	0	0	0	184	0	0	0	100%	
	TomatoBacterial	0	0	0	0	0	169	0	1	99.41%	
	TomatoHealthy	0	0	0	0	0	2	184	5	1	95.83%
	PotatoHealthy	0	0	0	0	0	0	0	186	0	100%
	PepperBacterial	0	0	0	0	0	0	0	0	193	100%
											98.59%

Figure. 18: ResNet50 Confusion Matrix Experiment 2.

Figure. 18 Observation: Pepper Healthy (FN) = 2, Tomato Early Blight (FN) =1, Potato Early Blight (FN) = 5, Tomato Bacterial (FN) = 2, Potato Healthy (FN) = 13, and Pepper Bacterial (FN)= 1.

Output Classes	PepperHealthy	192	0	0	0	0	0	0	0	100%	
	PotatoLateBlight	0	198	0	0	2	0	0	0	99%	
	TomatoEarlyBlight	0	0	193	0	2	0	0	0	98.97%	
	TomatoLateBlight	0	0	2	189	1	0	3	0	96.92%	
	PotatoEarlyBlight	0	0	0	1	183	0	0	0	99.45%	
	TomatoBacterial	3	0	0	0	4	153	10	0	90%	
	TomatoHealthy	5	1	0	2	0	8	172	1	3	89.58%
	PotatoHealthy	1	0	0	5	0	4	9	167	0	89.78%
	PepperBacterial	0	0	0	0	0	0	0	0	193	100%
											96.07%

Figure. 19: MobileNet Confusion Matrix Experiment 2.

Figure. 19 Observation: Pepper Healthy (FN) = 9, Potato Late Blight (FN) = 1, Tomato Early Blight (FN) = 2, Tomato Late Blight (FN) = 8, Potato Early Blight (FN) = 9, Tomato Bacterial (FN) = 12, Tomato Healthy = 22, Potato Healthy (FN) = 1, and Pepper Bacterial (FN)= 3.

Output Classes	PepperHealthy	191	1	0	0	0	0	0	0	99.47%	
	PotatoLateBlight	0	200	0	0	0	0	0	0	100%	
	TomatoEarlyBlight	0	0	194	0	1	0	0	0	99.48%	
	TomatoLateBlight	0	0	0	194	0	0	0	1	99.48%	
	PotatoEarlyBlight	0	0	0	0	184	0	0	0	100%	
	TomatoBacterial	0	0	0	0	0	169	0	1	99.41%	
	TomatoHealthy	0	0	0	0	0	1	186	4	1	96.87%
	PotatoHealthy	0	0	0	0	0	0	0	185	1	99.46%
	PepperBacterial	0	0	0	0	0	0	0	0	193	100%
											99.35%

Figure. 20: Proposed Model Confusion Matrix Experiment 2.

Figure. 20 Observation: Potato Late Blight (FN) = 1, Potato Early Blight (FN) = 1, Tomato Bacterial (FN) = 1, Potato Healthy (FN) = 6, and Pepper Bacterial (FN) = 2.

Table 5 Comparison of existing methodologies with proposed model

Authors name and year	Methodologies	Dataset Used	Accuracy (%)	Loss	No. parameters
Hema et al. [13] (2021)	VGG16 and Resnet34 CNN	Plant Village	97.77 % and 97.58% respectively	0.0857 and 0.0875	---
Tiwari, D. [14] (2020)	VGG19 with logistic regression	Potato Plant village	97.8%	---	143 million
Sanga et al. [17] (2020)	ResNet-152	banana leaf images (Arusha and Mbeya regions in Tanzania)	99.2	0.0539	60 million
Andrew, et al. [16] (2022)	Ensemble (VGG16, VGG19, and Xception)	grape leaf diseases Plant Village	99.82%	0.0216	73 million
M. Bhagat. [18] (2020)	CNN	pepper leaves	96.78%.	---	----
Proposed Model	Ensemble transfer learning (ResNet50 and MobileNetV3)	(Tomato, potato, and pepper) Plant Village	99.35%	0.019	37 million

5.2 Results Discussion

According to experiments, it was found that the performance of the proposed model in the second experiment, provided better accuracy with 99.38% and a loss of 0.0211 for training and an accuracy of 99.35 % and a loss of 0.019 for testing. This is likely due to the fact that the second experiment had fewer training sets than the first experiment, which allowed the model to learn more effectively and generalize better to new data, which goes beyond the testing accuracy of the studies done by (Tiwari, Divyansh, et al., [14]), (S. L. Sanga et al. [17]), and (Hema et al. [13]) as shown in Table 5 With an accuracy of 99.82%, the proposed model's accuracy is just somewhat lower than that of the work of Andrew, et al. [16]. However, only 37 million training parameters are utilized in the proposed work, compared to the number of training parameters used in [16]. This means that the proposed model requires fewer computational resources to train, can be trained faster, and reduce overfitting. In the plant disease classification problem, misclassifying a diseased plant as healthy can have devastating consequences, such as the spread of the disease to other plants. The proposed model successfully reduced this risk, with the highest false negative value of 6, which is a good percentage compared to previous works. This indicates that the proposed model is effective in accurately classifying plant diseases. The proposed model has been trained for 18 epochs with a learning rate of $2e-5$. The time during training the model is about $70 \text{ min} \pm 6 \text{ min}$ and the time during testing is $2 \text{ min} \pm 1 \text{ min}$. The number of errors was 11 images out of 1,707. This indicates that the proposed model is effective in accurately classifying plant diseases and can play a significant role in preventing the spread of plant diseases.

6. Conclusion

Classification of plant diseases at their earliest stages plays an essential role in increasing agricultural crops and ability the fulfilling of people's food needs. In recent years, deep learning and machine learning have gained popularity for automating this process. However, current studies often rely on handcrafted features in machine-learning approaches, which may not be suitable for diagnosing complex plant diseases. Additionally, using too many training parameters can lead to overfitting, and the overall accuracy may have an inverse relationship with dataset size. Due to disease similarities and a lack of background information, it can be challenging to identify plant leaf diseases solely from visual observation accurately.

The proposed model is classifying the most consumable crops (tomato, potato, and pepper) as healthy or diseased using deep ensemble transfer learning with lightweight networks (MobileNetV3 small and ResNet50), the proposed model achieved a performance of about 99.50 % with only 37 million training parameters, without any feature engineering, segmentation, or preprocessing. The reduced number of parameters in the proposed model represents a more efficient use of computational resources, significantly reduces the time required for classification, and reduces overfitting. Additionally, by eliminating the need for feature engineering, segmentation, or preprocessing, the proposed model is better able to handle the complexities and variations associated with plant diseases.

Overall, this work represents a general plant disease classification model with significant improvement in efficiency and performance. Also, it has the potential to increase crop yields and prevents the spread of plant diseases.

References

- [1] L. Jun and X. Wang, "Plant Diseases and Pests Detection Based on Deep Learning: A Review," *Plant Methods*, vol. 17, no. 1, pp. 22, Jan. 2021.
- [2] P. K. Sethy et al., "Deep feature based rice leaf disease identification using support vector machine," *Computers and Electronics in Agriculture*, vol. 175, p. 105527, 2020.
- [3] M. A. Ebrahimi et al., "Vision-based pest detection based on SVM classification method," *Comput. Electron. Agricult.*, vol. 137, pp. 52-58, 2017.
- [4] E. D. de Wolf and S. A. Isard, "Disease Cycle Approach to Plant Disease Prediction," *Annu Rev Phytopathol*, vol. 45, no. 1, pp. 203–220, Sep. 2007, doi: 10.1146/annurev.phyto.44.070505.143329.
- [5] J. Isleib, "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
- [6] J. Chen et al., "Using deep transfer learning for image-based plant disease identification," *Comput. Electron. Agricult.*, vol. 178, p. 105739, 2020.
- [7] T. S. Xian and R. Ngadiran, "Plant diseases classification using machine learning," *J. Phys. Conf. Ser.*, vol. 1962, no. 1, p. 012024, 2021, doi: 10.1088/1742-6596/1962/1/012024.
- [8] M. A. Khan, "Detection and classification of plant diseases using image processing and multiclass support vector machine," *Int. J. Comput. Trends Technol.*, vol. 68, no. 4, pp. 5-11, 2020, doi: 10.14445/22312803/ijctt-v68i4p102.
- [9] K. Thaiyalnayaki and C. Joseph, "Classification of plant disease using SVM and deep learning," *Mater. Today: Proc.*, vol. 47, pp. 468-470, 2021, doi: 10.1016/j.matpr.2021.05.029.
- [10] C. Janiesch et al., "Machine learning and deep learning," *Electron. Markets*, vol. 31, no. 3, pp. 685-695, 2021, doi: 10.1007/s12525-021-00475-2.

- [11] A. S. Paymode and V. B. Malode, "Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG," *Artif. Intell. Agric.*, vol. 6, pp. 23-33, 2022, doi: 10.1016/j.iaia.2021.12.002.
- [12] M. Hussain et al., "A study on CNN transfer learning for image classification," in *Advances in Intelligent Systems and Computing*, Springer, vol. 840, pp. 191-202, 2019.
- [13] H. Hema et al., "Plant disease prediction using convolutional neural network," *EMITTER Int. J. Eng. Technol.*, vol. 9, no. 2, pp. 283-293, 2021, doi: 10.24003/emitter.v9i2.640.
- [14] D. Tiwari et al., "Potato leaf diseases detection using deep learning," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, 2020, pp. 967-972.
- [15] A. Khamparia et al., "Seasonal crops disease prediction and classification using deep convolutional encoder network," *Circuits Syst. Signal Process.*, vol. 39, no. 2, pp. 818-836, 2020, doi: 10.1007/s00034-019-01041-0.
- [16] A. Nader et al., "Grape leaves diseases classification using ensemble learning and transfer learning," *Int. J. Adv. Comput. Sci. Appl. (IJACSA)*, vol. 13, no. 7, 2022, doi: 10.14569/ijacsa.2022.0130767.
- [17] S. L. Sanga et al., "Mobile-based deep learning models for banana disease detection," *Technol. Appl. Sci. Res.*, vol. 10, no. 3, pp. 5674-5677, 2020.
- [18] M. Bhagat, D. Kumar, R. Mahmood, B. Pati, and M. Kumar, "Bell pepper leaf disease classification using CNN," in *2nd International Conference on Data, Engineering and Applications (IDEA)*, 2020, pp. 1-5, doi: 10.1109/IDEA49133.2020.9170728.
- [19] A. Ali, "PlantVillage Dataset [Data set] [Online]. Available: <https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset>," 2019.
- [20] K. He et al., "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.
- [21] A. Howard et al., "Searching for MobileNetV3," 2019, doi: 10.48550/ARXIV.1905.02244.
- [22] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," 2017, doi: 10.48550/ARXIV.1712.04621.
- [23] R. Yamashita, M. Nishio, R.K.G. Do, et al., "Convolutional neural networks: an overview and application in radiology," *Insights Imaging*, vol. 9, pp. 611-629, 2018, doi: 10.1007/s13244-018-0639-9.
- [24] A. F. Agarap, "Deep learning using rectified linear units (ReLU)," 2018, doi: 10.48550/ARXIV.1803.08375.
- [25] H. Gholamalinezhad and H. Khosravi, "Pooling methods in deep neural networks, a review," *arXiv preprint arXiv:2009.07485*, 2020.
- [26] C. Nwankpa et al., "Activation functions: comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378*, 2018.
- [27] X. Li et al., "Understanding the disharmony between dropout and batch normalization by variance shift," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2019.
- [28] E. Bisong, "Google Colaboratory," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, Apress, Berkeley, CA, 2019, doi: 10.1007/978-1-4842-4470-8_7.
- [29] Keras Team, "Getting started," *Keras.io*. [Online]. Available: https://keras.io/getting_started/.
- [30] I. Markoulidakis et al., "Multiclass confusion matrix reduction method and its application on Net Promoter Score classification problem," *Technologies*, vol. 9, no. 4, p. 81, 2021, doi: 10.3390/technologies9040081.