Crop Disease Detection with Machine Learning

Samay Jain[1] and Bradford[#]

[1]Mission San Jose High School, USA
[#]Advisor

ABSTRACT

Plant disease, a significant threat, harms a great deal of farmers each year. A study done by the USDA, (U.S. Department of Agriculture), stated that $220 billion worth of crops are lost each year from plant disease. These diseases can be difficult to discern as the visible symptoms can be overlooked by the human eye. To address this problem, using CNN, (Convolutional Neural Networks), this research identifies crop disease through a dataset of images. The program employs feature extraction techniques using a pre-trained MobileNetV2 model to enhance classification accuracy. After training, the model resulted in 87% testing accuracy. On top of that, a system that recommends certain medication to cure the disease was also implemented. In all, the proposed method displays a way to accurately and efficiently identify plant disease.

Introduction

Plants are critical organisms that uphold the lives of all humans and animals; however, these vital living beings see many populations decline due to preventable diseases. A study from the Caribbean Plant Health Directors stated that 14.1% of plants die via illness. Treating these plants quickly and efficiently is important, for these crops generate a great share of our GDP. The US Department of Agriculture stated that food and plant related industries earned nearly $1.530 trillion in 2023, contributing to 5.6% of the US GDP. Moreover, that same year, American farms alone produced a staggering $203.5 billion which was 0.7% of the GDP. To identify and cure infected crops, farmers call inspectors also known as pathologists. During the time before the inspectors arrive, the disease might grow to a point till it is incurable. Hence, the aim of this project is to build code that can efficiently and accurately identify illnesses in plants. The experiment uses the Plant Village dataset and applies the MobileNetV2 architecture. The data is then trained and tested to see the outcome of the model.

Background

Previous Research

Previous studies have demonstrated the potential of deep learning models in plant disease detection. However, most focus on model accuracy without addressing the practical application of treatment recommendations. This research aims to fill this gap by providing a recommendation for a solution.

Underlying Science

A broad science, plant pathology, includes detecting and treating plant diseases. These pathologists spot plant diseases through unique methods such as using light and electron microscopy to determine miniscule fungi and bacteria on a plant.

Using Transfer Learning in Plant Disease Detection

Transfer learning involves using pre-trained models on large datasets to leverage learned features for specific tasks. This approach is particularly useful in scenarios with limited labeled data. MobileNetV2, a lightweight CNN architecture, is commonly used in mobile and embedded vision applications due to its efficiency and accuracy.

Dataset

In the project, the data imported from Kaggle is named Plant Village Dataset by Adil Mubashir Chaudhry. The data collected consisted of a gamut of photos and was already split into training, testing, and validation sets; specifically, 14438 images were allotted to train, while 4128 and 2073 images went for testing and validation respectively. Images of some of the healthy and disease leaves can be seen in Figure 1.

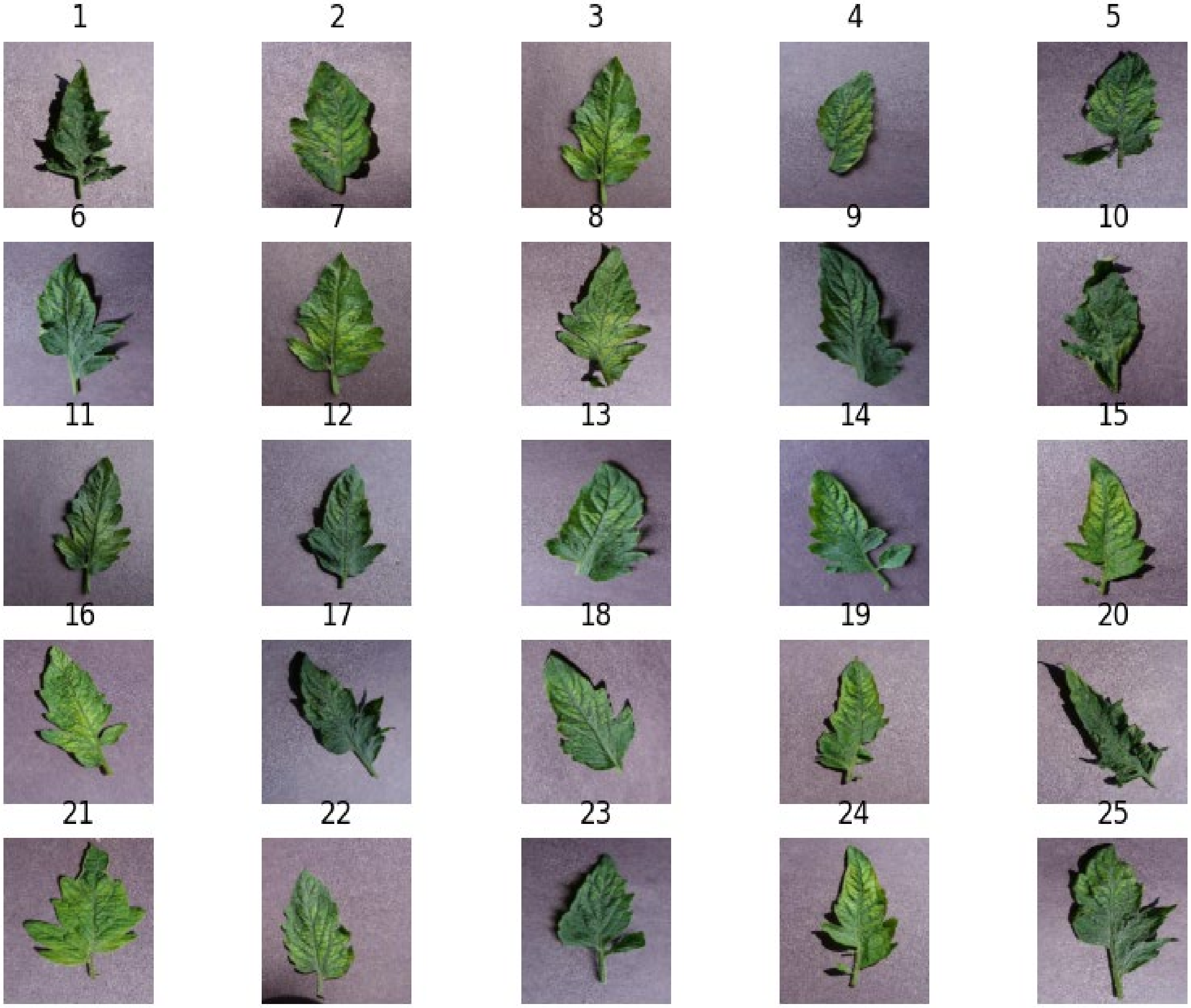


Figure 1. Images of 25 healthy or diseases leaves

The first step is to visualize the data. Next, the distributions of each of the classes in the training, testing, validation folders were plotted (Figure 2, 3 and 4) – the directories contained multiple sub-directories with images in them – to understand the structure. Each set had identical structure where in the figures below, the training is displayed first, then testing, and lastly validation.

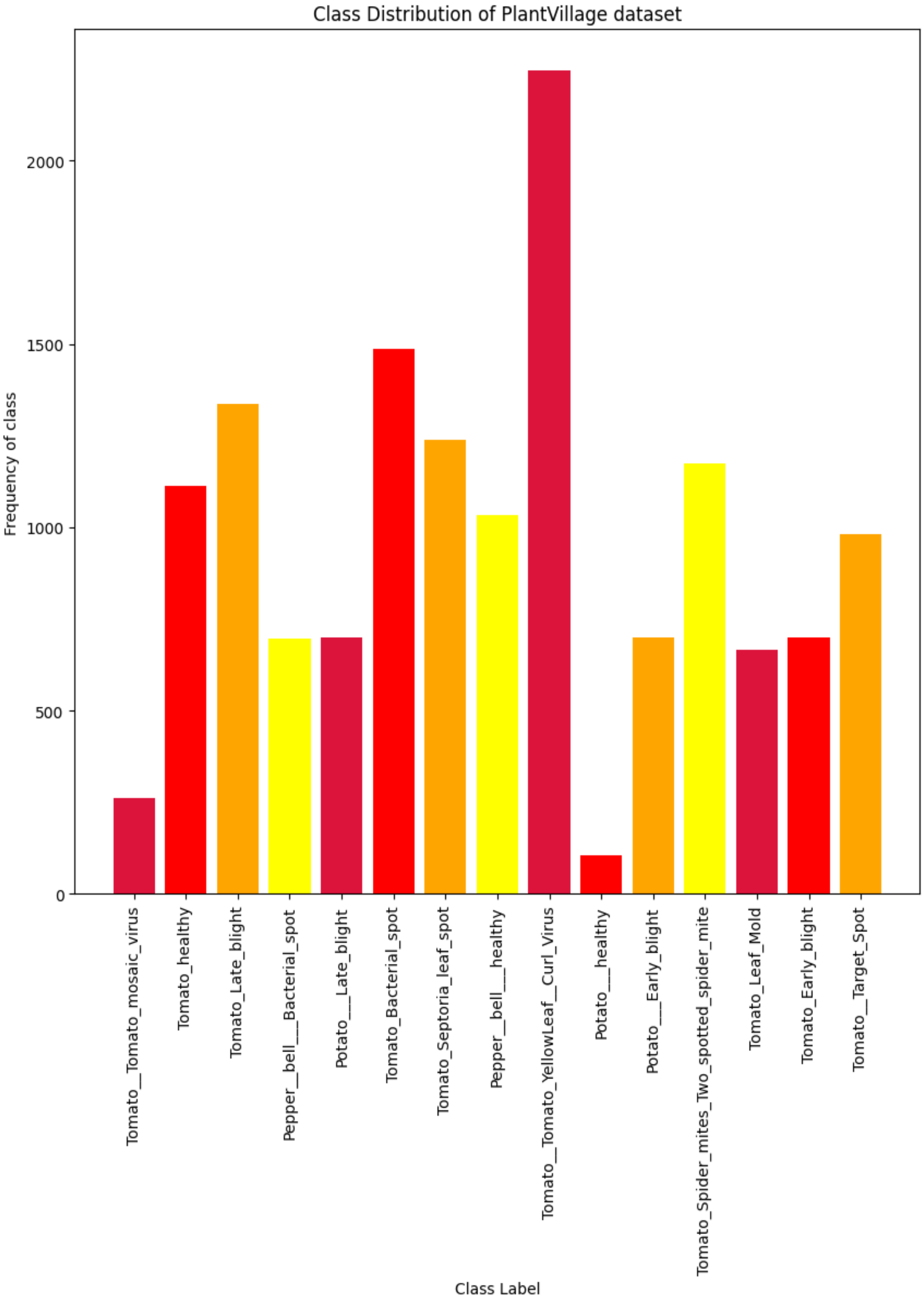


Figure 2. Class distributions of the healthy set

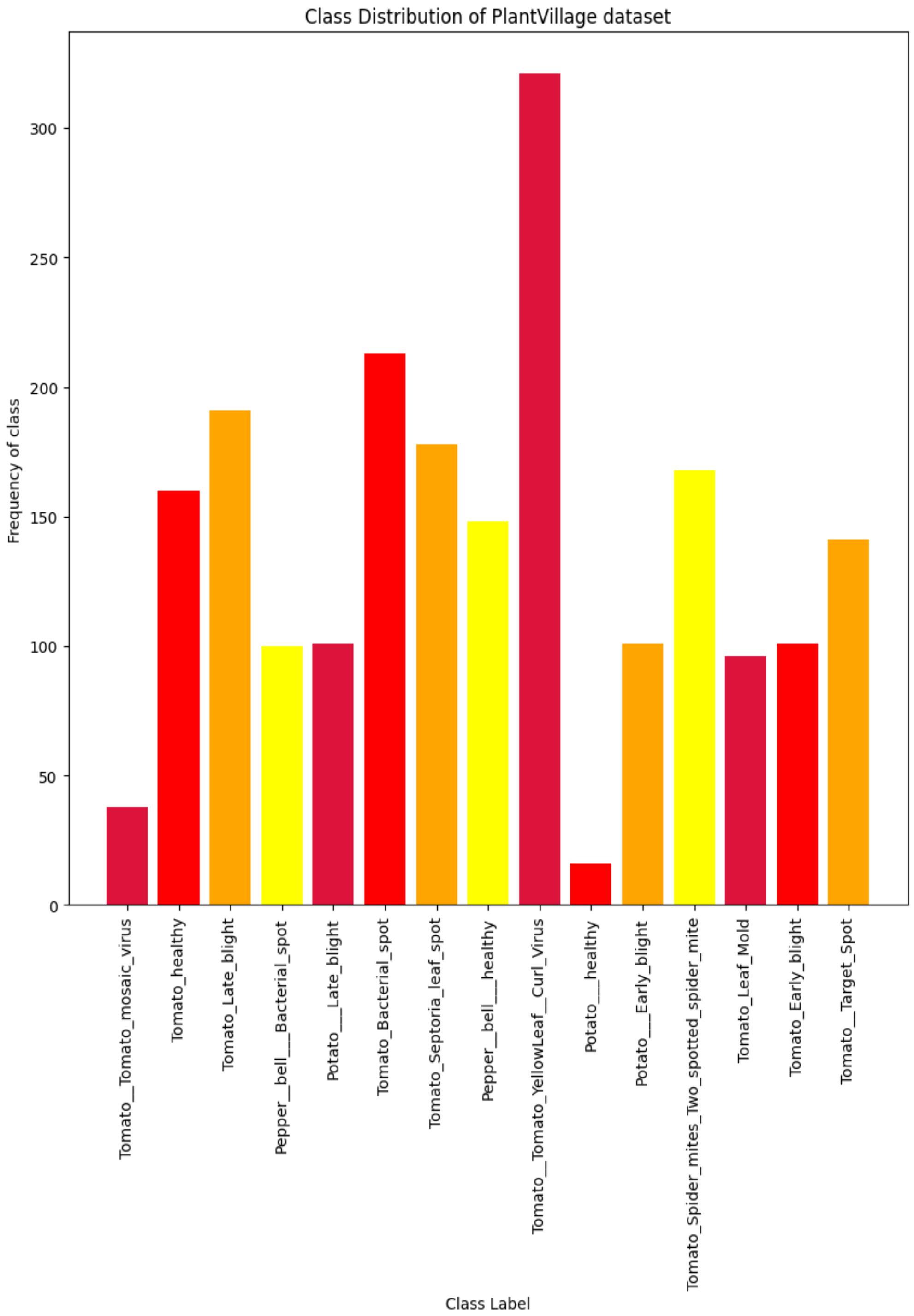


Figure 3. Class distributions of the diseased set

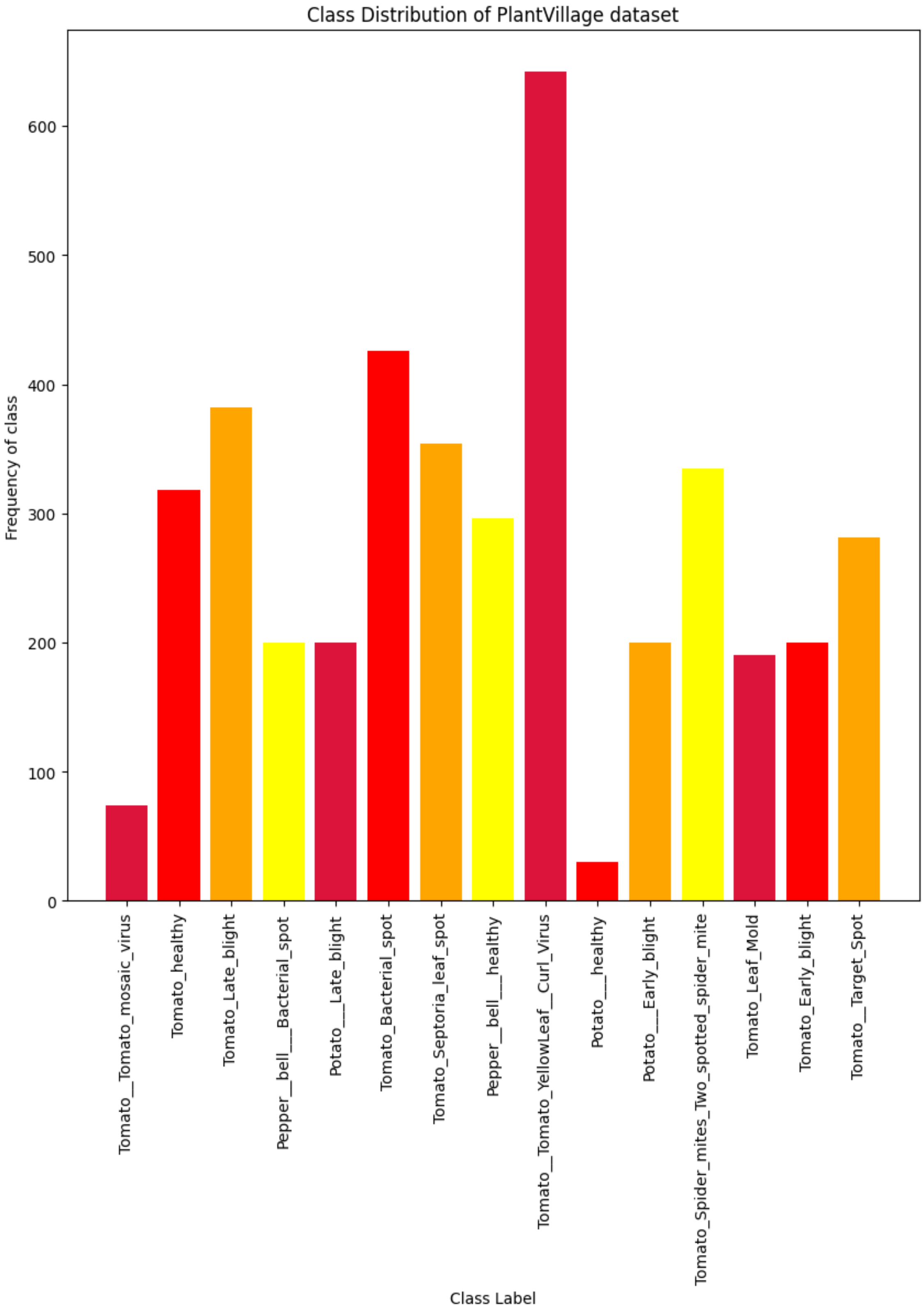


Figure 4. Class distributions of the validation set

The images in the database have a different background, different shape and uneven light, which affects the accuracy of the application. To increase the model's detection and accuracy, using ImageDataGenerator from keras, the images were resized to 224 x 224 and normalized. In addition, for an enhanced quality, data augmentation methods such as rotation, flipping, zooming, etc. were used. Lastly, the program used the VGG16 model pre-trained on ImageNet for feature extraction. The convolutional layers of VGG16 were used to extract high-level features from the images, which were then fed into a custom-built dense neural network for classification. The features extracted include fine details, textures, shapes, edges, etc (Figure 5).

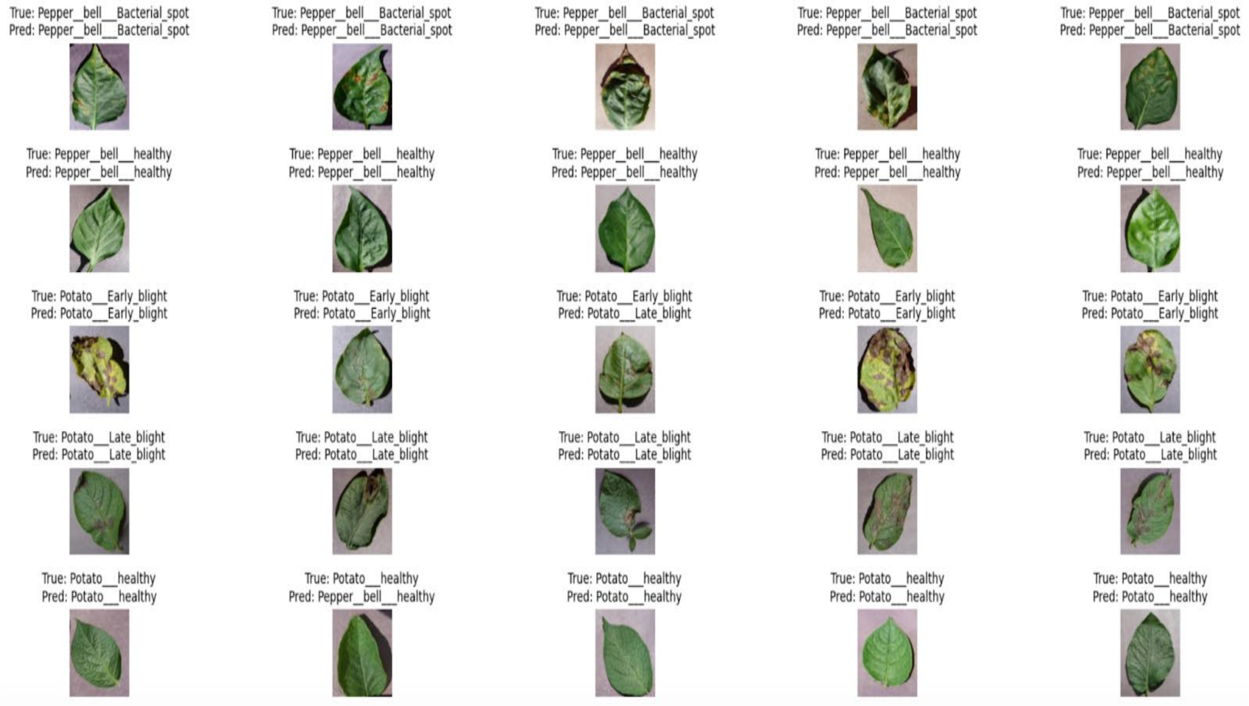

True: Pepper__bell__Bacterial_spot
Pred: Pepper__bell__Bacterial_spot
True: Pepper__bell__Bacterial_spot
Pred: Pepper__bell__Bacterial_spot
True: Pepper__bell__Bacterial_spot
Pred: Pepper__bell__Bacterial_spot
True: Pepper__bell__Bacterial_spot
Pred: Pepper__bell__Bacterial_spot
True: Pepper__bell__Bacterial_spot
Pred: Pepper__bell__Bacterial_spot
True: Pepper__bell__healthy
Pred: Pepper__bell__healthy
True: Pepper__bell__healthy
Pred: Pepper__bell__healthy
True: Pepper__bell__healthy
Pred: Pepper__bell__healthy
True: Pepper__bell__healthy
Pred: Pepper__bell__healthy
True: Pepper__bell__healthy
Pred: Pepper__bell__healthy
True: Potato__Early_blight
Pred: Potato__Early_blight
True: Potato__Early_blight
Pred: Potato__Early_blight
True: Potato__Early_blight
Pred: Potato__Late_blight
True: Potato__Early_blight
Pred: Potato__Early_blight
True: Potato__Early_blight
Pred: Potato__Early_blight
True: Potato__Late_blight
Pred: Potato__Late_blight
True: Potato__Late_blight
Pred: Potato__Late_blight
True: Potato__Late_blight
Pred: Potato__Late_blight
True: Potato__Late_blight
Pred: Potato__Late_blight
True: Potato__Late_blight
Pred: Potato__Late_blight
True: Potato__healthy
Pred: Potato__healthy
True: Potato__healthy
Pred: Pepper__bell__healthy
True: Potato__healthy
Pred: Potato__healthy
True: Potato__healthy
Pred: Potato__healthy
True: Potato__healthy
Pred: Potato__healthy

Figure 5. Plot of sample prediction vs real disease

Methodology

Main Overview

The project was commenced by importing the dataset from Kaggle which was soon evaluated and preprocessed. Afterwards, a model was made using a pre-trained architecture which was compiled and trained.

Model Architecture

The MobileNetV2 architecture was used, initialized with ImageNet pre-trained weights. GlobalAveragePooling2D and dropout with dense layers for classification were then added to the network. Here is the overview of the model created.

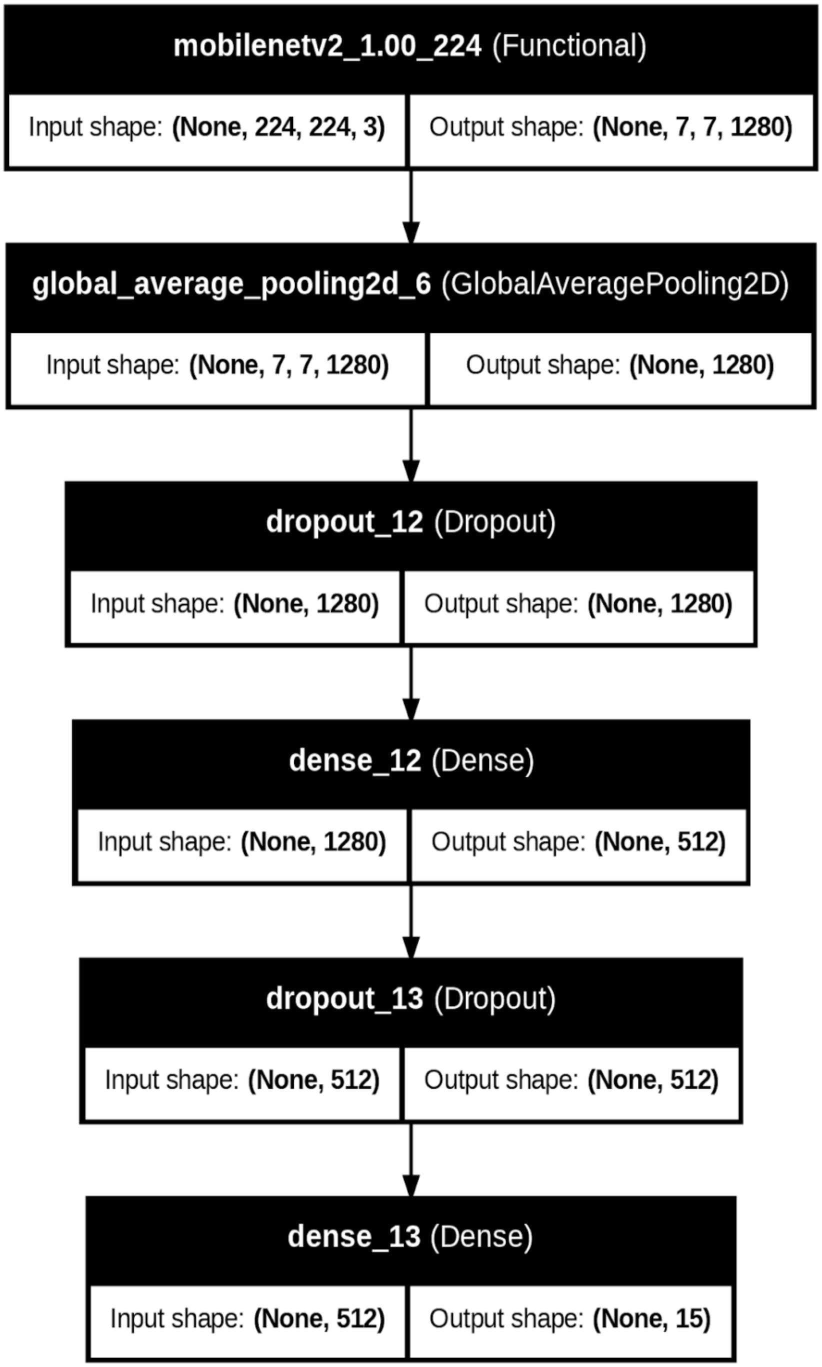

mobilenetv2_1.00_224 (Functional)
Input shape: (None, 224, 224, 3)
Output shape: (None, 7, 7, 1280)
global_average_pooling2d_6 (GlobalAveragePooling2D)
Input shape: (None, 7, 7, 1280)
Output shape: (None, 1280)
dropout_12 (Dropout)
Input shape: (None, 1280)
Output shape: (None, 1280)
dense_12 (Dense)
Input shape: (None, 1280)
Output shape: (None, 512)
dropout_13 (Dropout)
Input shape: (None, 512)
Output shape: (None, 512)
dense_13 (Dense)
Input shape: (None, 512)
Output shape: (None, 15)

Figure 7. Model overview

Data Preprocessing

First, a plot for the class distributions, as seen in figures 2-4, was made to see the scope of the images. After, images were loaded from the folder, resized to 224x224 and normalized for MobileNetV2 (using preprocess_input). With the help of data augmentation techniques-rotations, height and width shifting, horizontal shift and vertical shifts including zoom were taken.

Training and Evaluation

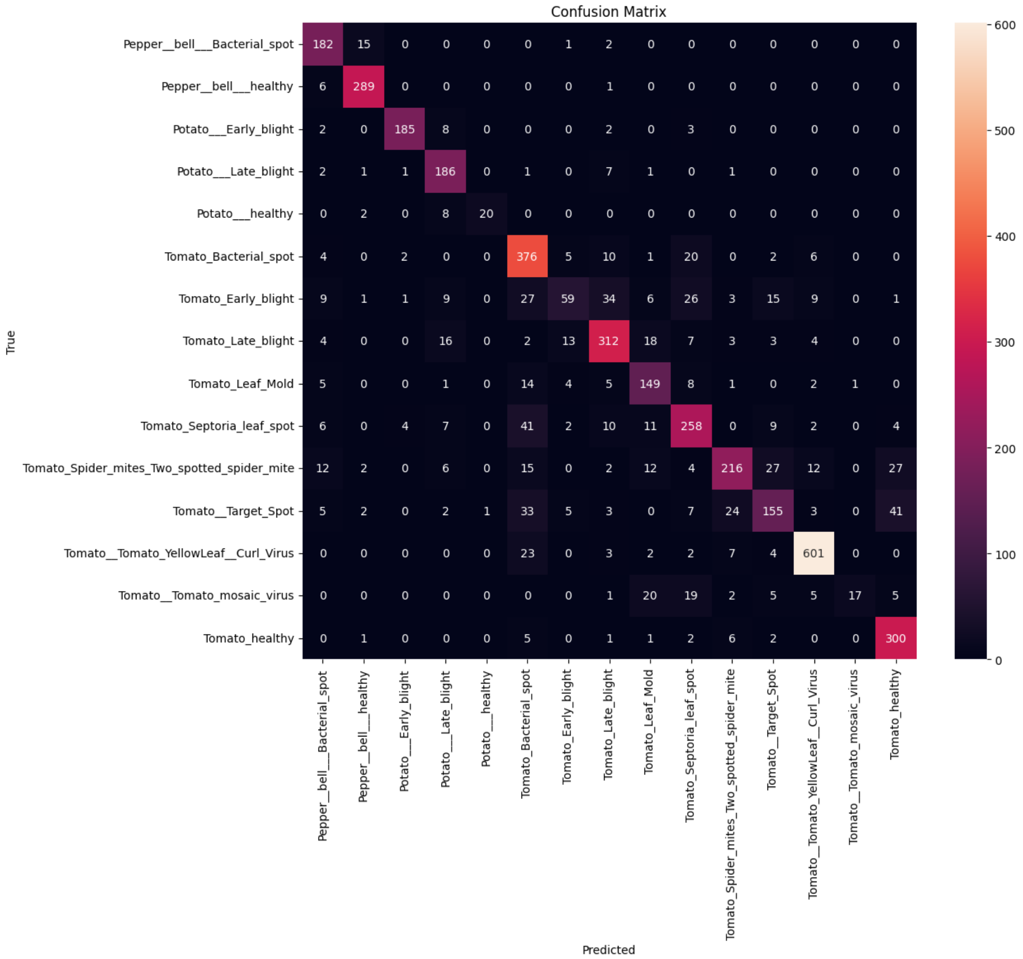
The model was compiled using the Adam optimizer with a learning rate of 0.0001. It was trained for 5 epochs with a batch size of 32. This can be seen in figure 7. The training process was monitored using accuracy and loss metrics, and the model's performance was evaluated on the validation and test sets.

Graphs

A confusion matrix, figure 8, was generated to analyze the model's predictions across different classes. The confusion matrix helped in identifying any misclassifications and assessing the model's quality. Moreover, a plot that mapped out the training, testing, and validation set accuracies which is seen in the results category was also created.

Results

The trained model achieved a test accuracy of 87%, indicating its effectiveness in detecting various plant diseases (Figure 7 and 8). The confusion matrix revealed that most diseases were correctly classified, with only a few instances of misclassification, primarily between similar disease categories.



	Pepper__bell___Bacterial_spot	Pepper__bell___healthy	Potato___Early_blight	Potato___Late_blight	Potato___healthy	Tomato_Bacterial_spot	Tomato_Early_blight	Tomato_Late_blight	Tomato_Leaf_Mold	Tomato_Septoria_leaf_spot	Tomato_Spider_mites_Two_spotted_spider_mite	Tomato__Target_Spot	Tomato__Tomato_YellowLeaf__Curl_Virus	Tomato__Tomato_mosaic_virus	Tomato_healthy
Pepper__bell___Bacterial_spot	182	15	0	0	0	0	1	2	0	0	0	0	0	0	0
Pepper__bell___healthy	6	289	0	0	0	0	0	1	0	0	0	0	0	0	0
Potato___Early_blight	2	0	185	8	0	0	0	2	0	3	0	0	0	0	0
Potato___Late_blight	2	1	1	186	0	1	0	7	1	0	1	0	0	0	0
Potato___healthy	0	2	0	8	20	0	0	0	0	0	0	0	0	0	0
Tomato_Bacterial_spot	4	0	2	0	0	376	5	10	1	20	0	2	6	0	0
Tomato_Early_blight	9	1	1	9	0	27	59	34	6	26	3	15	9	0	1
Tomato_Late_blight	4	0	0	16	0	2	13	312	18	7	3	3	4	0	0
Tomato_Leaf_Mold	5	0	0	1	0	14	4	5	149	8	1	0	2	1	0
Tomato_Septoria_leaf_spot	6	0	4	7	0	41	2	10	11	258	0	9	2	0	4
Tomato_Spider_mites_Two_spotted_spider_mite	12	2	0	6	0	15	0	2	12	4	216	27	12	0	27
Tomato__Target_Spot	5	2	0	2	1	33	5	3	0	7	24	155	3	0	41
Tomato__Tomato_YellowLeaf__Curl_Virus	0	0	0	0	0	23	0	3	2	2	7	4	601	0	0
Tomato__Tomato_mosaic_virus	0	0	0	0	0	0	0	1	20	19	2	5	5	17	5
Tomato_healthy	0	1	0	0	0	5	0	1	1	2	6	2	0	0	300

Figure 7. Confusion matrix of correct classification

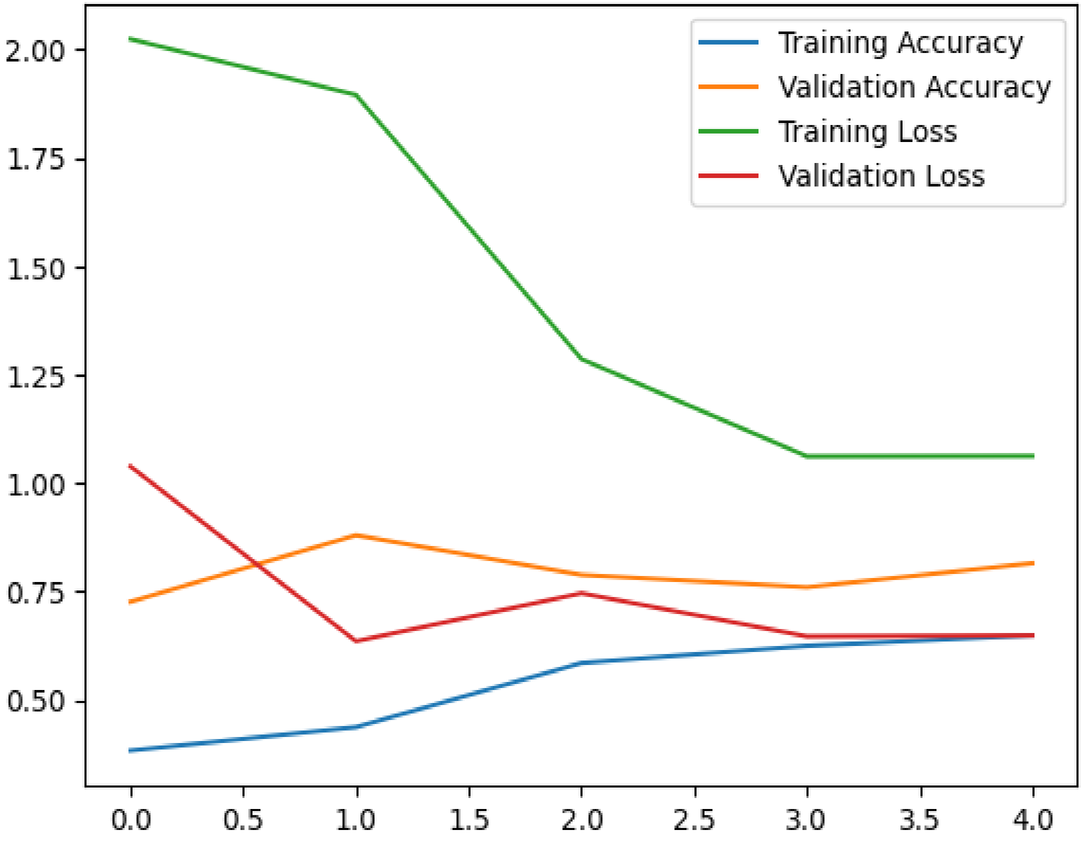


Figure 8. Accuracy of the data over training and validation periods

Compared to previous studies, the use of transfer learning with MobileNetV2 provided a balance between accuracy and efficiency. The model's performance aligns with existing methods, demonstrating the value of transfer learning. Although the training accuracy reached 70%, the testing accuracy was 88%, shown in figure 8.
Some misclassifications were observed in diseases with similar visual symptoms. This highlights the need for further refinement in distinguishing between closely related diseases. Potential sources of error include variations in image quality and lighting conditions.

The model also included a treatment recommendation system based on the predicted disease. Using if statements, the program would print a treatment according to the output of the model. For example, for "Tomato__Tomato_YellowLeaf__Curl_Virus," the recommendation is to use insecticides to control whiteflies. This practical application makes the model useful beyond mere detection, offering medicinal insights.

Conclusion

In this paper, an AI model was developed with the testing accuracy was 88%. In the future with hyperparameter tuning or with the use of different pre-trained models, the program could be tuned to reach an accuracy of 90% for testing. In addition, a reason the training accuracy is lower than the testing is because the training data is more diverse than the testing data, causing the model to falter. This was confirmed after checking the data.

Acknowledgments

I would like to thank my advisor for the valuable insight provided to me on this topic.

References

[1] A. M. Chaudhry, “Plant Village Dataset,” *Kaggle*, 2022. [Online]. Available: https://www.kaggle.com/datasets/adilmubashirchaudhry/plant-village-dataset. [Accessed: Sep. 6, 2024].
[2] CPHD Forum, “Plant Disease – Crop Loss,” *CPHD Forum*, May 26, 2022. [Online]. Available: https://www.cphdforum.org/index.php/2022/05/26/plant-disease-crop-loss/. [Accessed: Aug. 17, 2024].
[3] USDA Economic Research Service, “Ag and Food Sectors and the Economy,” *USDA ERS*, 2024. [Online]. Available: https://www.ers.usda.gov/data-products/ag-and-food-statistics-charting-the-essentials/ag-and-food-sectors-and-the-economy/. [Accessed: Sep. 3, 2024].
[4] Y. Wu, A. A. Abdellatif, S. Zhang, S. K. Das, and Z. Xue, “Image-based crop disease detection with federated learning,” *Scientific Reports*, vol. 13, no. 1, 2023. [Online]. Available: https://www.nature.com/articles/s41598-023-46218-5. [Accessed: Sep. 6, 2024].