# Performance Analysis Report

**Plant Disease Detection Using Deep Learning**

Idhant Joshi, Mayank Kundalwal, Dr. Deepak Mishra

*Department of Computer Science and Engineering, IIT Jodhpur*

**

**Abstract**

*Food production must rise by 70% by 2050 to fulfil the demands of a population that is predicted to surpass 9 billion people, posing a serious challenge to humanity. However, infectious illnesses already cause an average 40% reduction in agricultural output, with some farmers, especially those in impoverished nations suffering total losses. With over 5 billion cell phones expected to be in use by 2020, agricultural farmers present a special potential. These gadgets have the potential to be used as plant disease diagnostic tools with machine learning and crowdsourcing. We can develop strong machine-learning models to help farmers all across the world by utilising a dataset like PlantVillage, which has more than 50,000 photos of both healthy and damaged crop leaves.*

**

# Contents:

1. [Introduction 1](#_TOC_250013)
   1. [Background 1](#_TOC_250012)
   2. [Objectives and Metrics 1](#_TOC_250011)
2. [Data Handling 2](#_TOC_250010)
   1. [PlantVillage Dataset 2](#_TOC_250009)
   2. [Data Pre-processing 3](#_TOC_250008)
3. Performance Metrics 3
   1. [Pipeline Implementation 3](#_TOC_250007)
      1. [Feature extraction with MobileNetV2 3](#_TOC_250006)
      2. [Support Vector Machine (SVM) 3](#_TOC_250005)
      3. [Comparison Models 3](#_TOC_250004)
4. [Key Findings 3](#_TOC_250003)
5. [Future Enhancements 4](#_TOC_250002)
6. [Conclusion 4](#_TOC_250001)

[……...References 4](#_TOC_250000)

……...Github Repository 4

# Introduction

## Background

Infectious illnesses pose serious problems for the agricultural industry since they can lower crop yields by 40% on average, with some farmers suffering losses of up to 100%. By 2050, it is expected that there will be more than 9 billion people on the planet, making it imperative to increase food production. With a sizable collection of 54,303 carefully chosen photos of both healthy and diseased plant leaves, the PlantVillage dataset serves as a basis for creating machine learning models that are intended to identify plant diseases. The performance of a Plant Disease Detection system using Support Vector Machine (SVM) classification on characteristics and features taken from MobileNetV2 are examined in this Performance Analysis Report. It further includes attributes such as: key results, performance evaluation versus goals, data handling, objectives, and future possibilities.

## Objectives and Metrics

### Objectives

* + - 1. To develop a plant disease detection system using transfer learning and machine learning techniques.
      2. To evaluate system performance against established metrics such as accuracy, precision, recall, and F1-score.
      3. To identify the most suitable classifier for the dataset by comparing SVM with other classifiers such as XGBoost and Random Forest on the abovementioned metrics.
      4. To develop a web interface to showcase the results.

### Metrics

* + - 1. Accuracy: The proportion of cases that were properly categorised out of all occurrences.
      2. Precision: An indicator of how well the classifier prevents erroneous positives.
      3. Recall: An indicator of how well the classifier detects true positives.
      4. F1-Score: A balanced mean of recall and accuracy that balances their trade-offs.
      5. True positives, true negatives, false positives, and false negatives are all represented visually in a confusion matrix.

# Data Handling

## PlantVillage Dataset

As each picture in the PlantVillage dataset, publicly available on Kaggle and the [PlantVillage website](https://plantvillage.psu.edu/), is classified as either healthy or ill, the dataset is strong enough to handle classification tasks.

Dataset Properties:

1. Number of Images: 54,303
2. Crop Species: 14
3. Disease Classes: 38
4. File Format: JPG
5. Image Dimensions: RGB, variable sizes
6. Total Size: ~2GB

Disease classes in the PlantVillage Dataset:

1. Apple Black Rot
2. Apple Cedar Rust
3. Apple Rust
4. Apple Healthy
5. Apple Scab
6. Blueberry Healthy
7. Cherry Powdery Mildew
8. Cherry Healthy
9. Corn Cercospora leaf spot - Gray leaf spot
10. Corn Common Rust
11. Corn Northern Leaf Blight
12. Corn Healthy
13. Grape Black Rot
14. Grape Esca (Black Measles)
15. Grape Leaf blight (Isariopsis Leaf Spot)
16. Grape Healthy
17. Orange Huanglongbing (Citrus greening)
18. Peach Bacterial Spot
19. Peach Healthy
20. Pepper, Bell Bacterial Spot
21. Pepper, Bell Healthy
22. Potato Early Blight
23. Potato Late Blight
24. Potato Healthy
25. Raspberry Healthy
26. Soybean Healthy
27. Squash Powdery Mildew
28. Strawberry Leaf Scorch
29. Strawberry healthy
30. Tomato Bacterial Spot
31. Tomato Early Blight
32. Tomato Late Blight
33. Tomato Leaf Mold
34. Tomato Septoria Leaf Spot
35. Tomato Spider Mites - Two-spotted Spider Mite
36. Tomato Target Spot
37. Tomato Yellow Leaf Curl Virus
38. Tomato Mosaic Virus

## Data Preprocessing

The pipeline for preprocessing included:

1. Image Resizing: To comply with MobileNetV2's input specifications, all images were downsized to 224x224.
2. Pixel values are normalised to fall between 0 and 1.
3. Data augmentation, the process of applying colour jitter, brightness modifications, random flips, cropping to increase the variability of a dataset et cetera was applied
4. Dataset splitting: The dataset was divided into 20% validation and 80% training sets.

# Performance vs Metrics

## Pipeline Implementation

### Feature extraction with MobileNetV2

* + - 1. For feature extraction, a pre-trained MobileNetV2 model (on ImageNet) was employed.
      2. To create 1280-dimensional feature vectors, a GlobalAveragePooling2D layer was included after the network's completely connected layers were eliminated.

### Support Vector Machine (SVM)

* + - 1. A support vector machine (SVM) is a type of supervised learning algorithm used in machine learning to solve classification and regression tasks. SVMs are great at solving binary classification problems, which require classifying the elements of a data set into two groups, as we have done in this scenario.
      2. The extracted features were classified using an SVM with a linear kernel.
      3. The classifier was optimized with grid search for hyperparameter tuning.

### Comparison Models

* + - 1. **Random Forest:** A random forest is a machine learning algorithm that uses multiple decision trees to reach a single result. It aggregates decisions from multiple decision trees.
      2. **XG Boost:** XGBoost is an optimized distributed gradient boosting library designed for efficient and scalable training of machine learning models. It is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction.

## Model Evaluation

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **SVM** | **Random Forest** | **XG Boost** |
| **Accuracy** | 95.6% | 92.4% | 89.6% |
| **Precision** | 95.9% | 92.8% | 89.3% |
| **Recall** | 95.5% | 92.3% | 89.7% |
| **F1-Score** | 95.7% | 92.5% | 89.5% |

# Key Findings

Given below is an holistic comparison of all three algorithms:

The Support Vector Machine (SVM) emerged as the most effective classifier for this task, achieving the highest accuracy (95.6%) and F1-score (95.7%). This is due to its ability to handle high-dimensional

feature spaces extracted from MobileNetV2. The linear kernel provided simplicity and efficiency, contributing to faster training and inference times compared to ensemble methods.

XGBoost fell short due to its over reliance on iterative boosting resulting in higher computational costs and longer training times. It did provide interpretability through feature importance, however the trade-offs in real-time application scenarios made it

less suitable.

Random Forest’s tendency to overfit on larger datasets and its challenges with rare class detection limited its effectiveness.

Even though the model's clear feature

importance metrics were advantageous for interpretability, its variance affected reliability compared to SVM.

To summarise, SVM was ideal for balanced, high-dimensional data with low inference latency.

# Future Enhancements

1. Extend Dataset Coverage: Additional plant species and disease types to

increase the system's versatility

across global agricultural scenarios can be added, such as fruits, tree barks, and branches.

1. Mobile Integration: The code to convert the SVM model to

TensorFlow Lite model for deployment on Android devices, enabling farmers to use the application offline is available on the GitHub repository linked below.

1. Exploring Hybrid Models: Combining SVM with ensemble methods to leverage their

complementary strengths, potentially improving rare disease detection.

1. Expanding Augmentation Techniques: Implementing more advanced augmentation methods such as mosaic augmentation can further improve the model’s robustness.

# Conclusion

This report explored a Plant Disease Detection system using the PlantVillage dataset. SVM has been identified as the most accurate classifier among the three evaluated, and MobileNetV2 was employed for feature extraction. The system was successfully integrated into a web application, showcasing its potential for aiding in the identification of plant diseases. For further progress, fine tuning is necessary to enhance the system's accuracy and

robustness across a wider range of plant species and disease types, which will be added in future commits on GitHub.

# References

1. [PlantVillage Dataset on Kaggle](https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset)
2. [PlantVillage Website](https://plantvillage.psu.edu/)

***GitHub Repository:*** The complete project, including source code on how to train the model, detailed documentation, and set up

instructions for the front end is available on GitHub: [Plant Disease Detector](https://github.com/idsjoe/Plant_Disease_Detector)