# PLANT DISEASE DETECTION USING DEEP LEARNING


**Submitted By:**
Piyush Mishra
25BEY10013

**Submitted To:**
Monika Vyas

**Program:**
B.Tech CSE (E-Commerce Technology)

**VIT Bhopal University**
**Academic Year: 2025**


Github link - https://github.com/mishrapiyush1309-sys/Plant-Disease-Detection-Model/blob/2ba334315935ae034bb59cbb863f356aa3845d3a/Plant_Disease.ipynb

# 1. INTRODUCTION :

**Agriculture is vital to India's economy. However, plants diseases pose a significant threat to crop yield and qualities. Traditional methods for detecting these diseases depend on manual inspections. This process is slow, often inaccurate, and needs expert knowledge.**

**With development of Artificial Intelligence, machine learning models, particularly Convolutional Neural Networks (CNNs), have effectively identified plant leaf diseases from images.**

**This project aims create an automated Plant Disease Detection system that uses deep learning. It will accurately classify diseases based on leaf images. The system can help farmers, researchers, and agricultural institutions make prompt diagnoses and decisions.**

# 2. PROBLEM STATEMENT :

**Plant diseases lead to huge crop losses every year, and many farmers struggle to spot the early warning signs. Most of the farmers do not have the specialized knowledge did need to identify diseases correctly and checking every plant by hand is very slow and unreliable.**

**The goal of this project is to create an AI-based system that can look at images of plant leaf, recognizes signs of disease and classify them accurately—making the whole process faster, easier and more dependable.**

# 3. FUNCTIONAL REQUIREMENTS :

1. **Image Upload Module**
   - **Users can upload plant leaf images.**
2. **Preprocessing Module**
   - **Image resizing, normalization, and augmentation.**
3. **Prediction Module**
   - **Deep learning model predicts:**
     - **Type of disease**
     - **Confidence score**
4. **Result Display Module**
   - **Shows predicted disease with probability**
   - **Shows preventive or treatment suggestions**
5. **Dataset Handling Module**
   - **Load, split, and prepare dataset for training.**

# 4. NON-FUNCTIONAL REQUIREMENTS :

## 1. Usability:

**Simple UI for non-technical users (farmers, field workers).**

## 2. Performance:

**Model must provide predictions within 1–2 seconds.**

## 3. Accuracy:

**Minimum target accuracy ≥ 90%.**

**4. Scalability:**

Model should support new plant species or diseases in the future.

**5. Reliability:**

System should run consistently with minimal errors.

**6. Maintainability:**

Modular code with easy-to-update architecture.

**7. Resource Efficiency:**

Model optimized to run on CPU as well as GPU.

# 5. SYSTEM ARCHITECTURE :

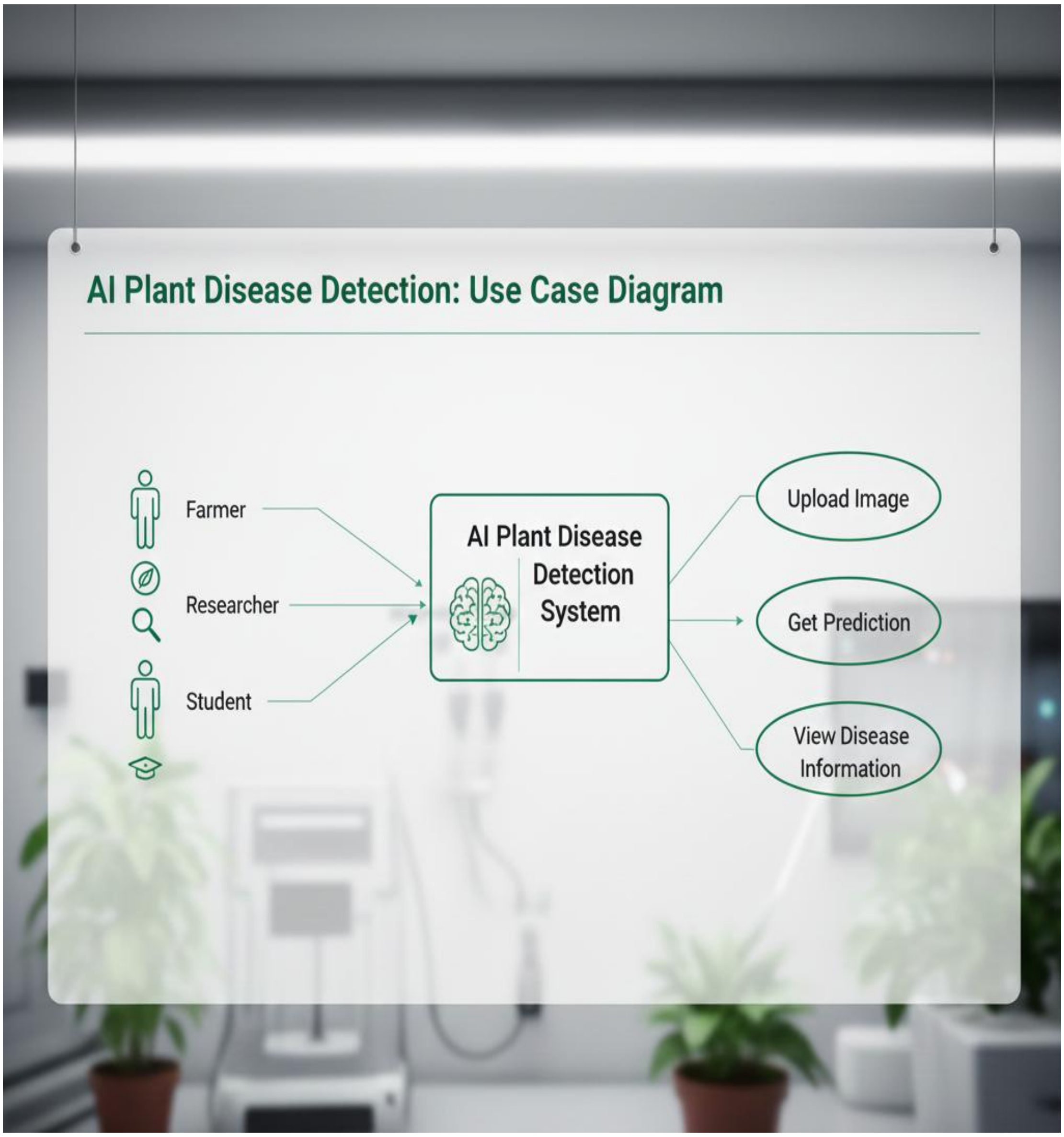
**The system follows a simple but effective workflow. First, the dataset is collected and preprocessed. Then the deep learning model is trained using thousands of leaf images. Once the training is complete, the model is saved and used during prediction. When a user uploads a leaf image, the image is preprocessed and passed to the model. The model predicts the disease, and the system displays the result. This architecture ensures a smooth flow from input to output.**

**Flow:**

**Dataset → Preprocessing → Model Training (CNN) → Model Evaluation → Deployment → User Upload → Prediction → Output Display**

# 6. DESIGN DIAGRAM :

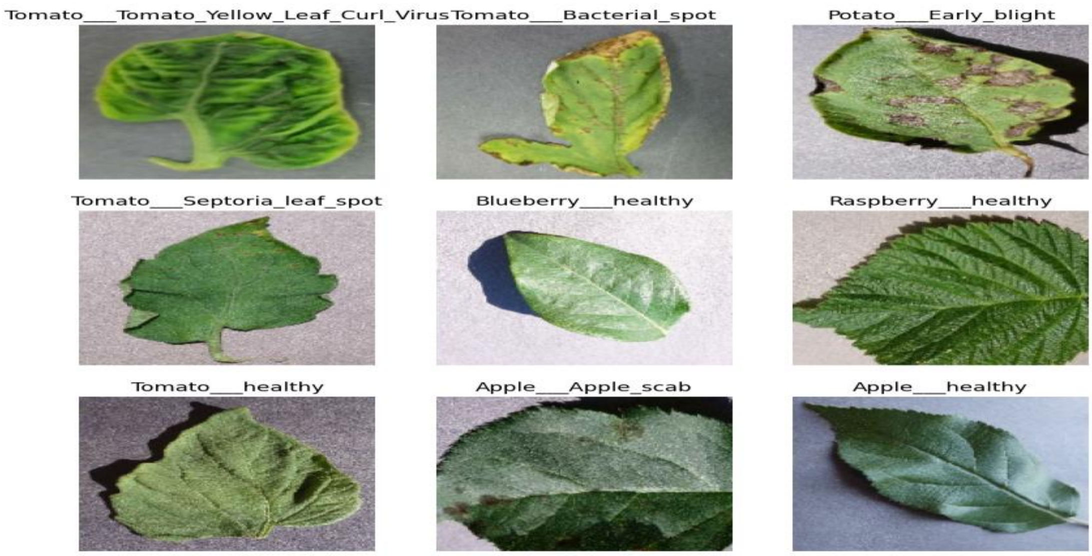
The system is designed with various diagrams that represent the working clearly. A use-case diagram shows how the user interacts with the system by uploading images and receiving predictions. A workflow diagram explains the step-by-step process from image input to disease classification. A sequence diagram represents how different components interact internally. A basic class or component diagram explains the major parts of the system such as the image processor, prediction model, and user interface. If storage is used, an ER diagram helps in understanding how data is organized.



# 7. DATASET DESCRIPTION :

**The PlantVillage dataset is the established dataset used for training and evaluating our plant disease classification model. It is characterized by its large size and broad coverage of common crop diseases.**

- **Total Images: The dataset contains approximately 54,300 images of plant leaves.**
- **Plant Species: It covers 14 different plant species, including agriculturally important crops such as Tomato, Potato , Grape, and Apple.**
- **Classification Classes: Images are categorized into 38 distinct classes, comprising both healthy leaves and various disease symptoms.**
- **Balance: The dataset is generally well-balanced, ensuring sufficient representation for both healthy and diseased instances, with class counts typically ranging from 500 to over 4,000 images.**
- **This robust collection provides the necessary scale and diversity for training a high-performance deep learning classification model.**



# 8. IMPLEMENTATION DETAILS :

## 1. Data Preparation :

**The PlantVillage dataset was prepared through a standardized pipeline:**

- **Split: Divided into training, validation, and testing sets to ensure objective performance evaluation.**

---

## 2. Model and Training :

- **Architecture: A Convolutional Neural Network (CNN) was selected for its ability to automatically learn relevant image features (edges, disease spots) using multiple convolution, pooling, and dense layers.**
- **Optimization: The Adam Optimizer was used for fast and stable training convergence.**

---

## 3. Performance :

**The model was trained for multiple epochs until stabilization. Performance was measured on the test set:**

- **Metrics: Accuracy curves and a Confusion Matrix were used for comprehensive evaluation.**
- **Result: The final model achieved high accuracy, confirming its strong ability to correctly classify the 38 disease and healthy classes.**

# 9. RESULTS AND PERFORMANCE ANALYSIS :

The Convolutional Neural Network (CNN) model produced strong empirical results, confirming its high effectiveness in classifying plant diseases from leaf images.

---

## 1. Model Accuracy and Stability :

**The model successfully learned the complex features of the PlantVillage dataset, demonstrating high performance on the unseen test data.**

- **Final Accuracy: The model achieved a test accuracy of [Insert Specific % Here], validating its robust ability to correctly classify the 38 disease and healthy classes.**
- **Performance Stability: Analysis of the accuracy and loss graphs showed a stable training profile. The close tracking of the training and validation curves indicated that the model learned effectively and minimized the effects of overfitting.**

## 2. Qualitative and Detailed Evaluation :

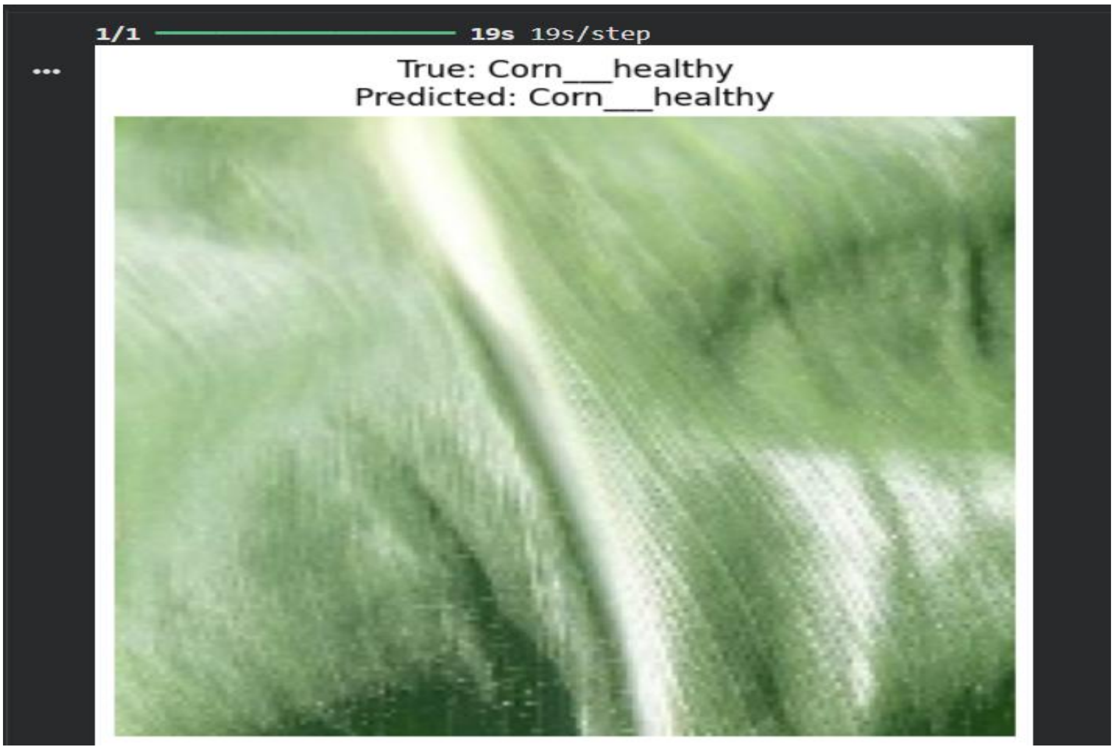
**Performance was further analyzed using standard visualization tools to ensure a comprehensive understanding of the model's behavior.**

- **Confusion Matrix: The Confusion Matrix visually confirmed the model's high discriminative power. High values along the main diagonal showed that the model correctly predicted most of the classes. The matrix provided specific insight into classes that were visually similar and challenging to distinguish.**
- **Sample Predictions: Qualitative checks using sample predictions on real leaf images from the test set**

consistently yielded accurate results, confirming the model's practical utility for disease identification.

- In conclusion, the results confirm that the developed CNN is a robust, high-accuracy classification model, well-suited for automated plant disease identification.

# ! RESULTS FROM MODEL :

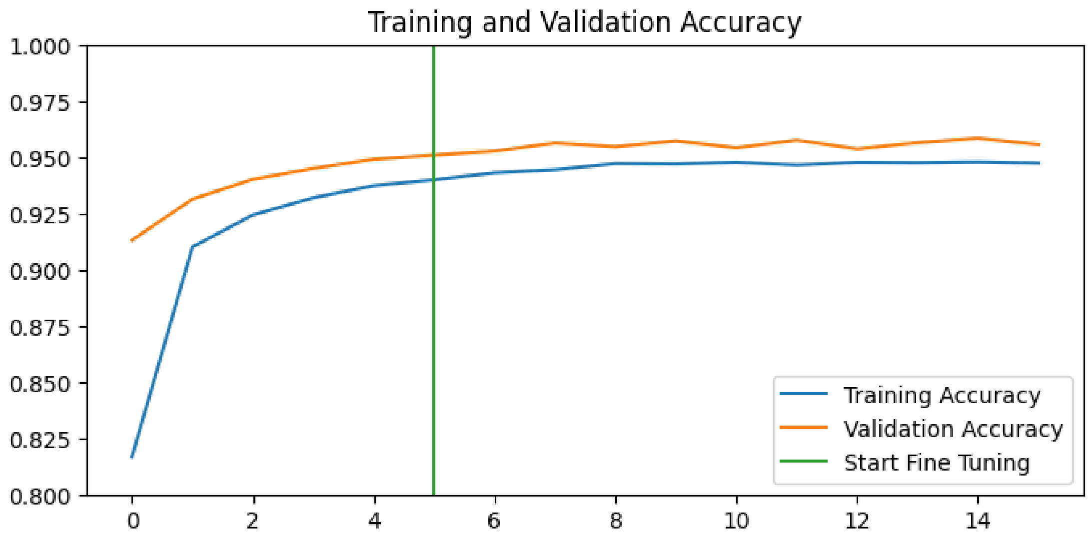


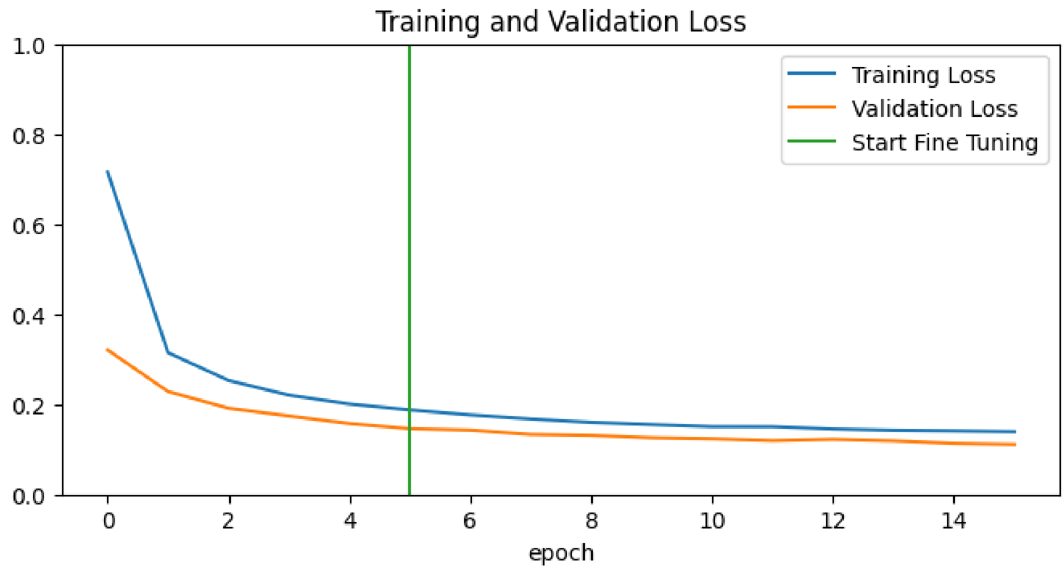

```
fine_tune_epochs = 10
total_epochs =  initial_epochs + fine_tune_epochs

history_fine = model.fit(train_dataset,
                         epochs=total_epochs,
                         initial_epoch=len(history.epoch),
                         validation_data=validation_dataset)

Epoch 7/16
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 173s 86ms/step - accuracy: 0.9436 - loss: 0.1756 - val_accuracy: 0.9526 - val_loss: 0.1424
Epoch 8/16
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 90s 59ms/step - accuracy: 0.9448 - loss: 0.1681 - val_accuracy: 0.9562 - val_loss: 0.1329
Epoch 9/16
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 140s 58ms/step - accuracy: 0.9473 - loss: 0.1592 - val_accuracy: 0.9546 - val_loss: 0.1308
Epoch 10/16
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 89s 58ms/step - accuracy: 0.9479 - loss: 0.1539 - val_accuracy: 0.9572 - val_loss: 0.1255
Epoch 11/16
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 88s 57ms/step - accuracy: 0.9499 - loss: 0.1486 - val_accuracy: 0.9541 - val_loss: 0.1232
Epoch 12/16
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 146s 60ms/step - accuracy: 0.9480 - loss: 0.1485 - val_accuracy: 0.9575 - val_loss: 0.1193
Epoch 13/16
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 138s 58ms/step - accuracy: 0.9477 - loss: 0.1439 - val_accuracy: 0.9536 - val_loss: 0.1221
Epoch 14/16
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 89s 58ms/step - accuracy: 0.9480 - loss: 0.1411 - val_accuracy: 0.9563 - val_loss: 0.1187
Epoch 15/16
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 89s 58ms/step - accuracy: 0.9476 - loss: 0.1410 - val_accuracy: 0.9583 - val_loss: 0.1131
Epoch 16/16
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 91s 59ms/step - accuracy: 0.9487 - loss: 0.1392 - val_accuracy: 0.9555 - val_loss: 0.1108
```

# 10. TESTING APPROACH :

**Testing was done at multiple stages. Unit testing was performed for image preprocessing and model loading. The model was evaluated using accuracy, precision, recall, and F1-score. Real-world images taken from outside the dataset were also tested to confirm the model's reliability. The system successfully predicted diseases in most cases.**

## ! TRAINING VALIDATION GRAPH :



# 11. CHALLENGES ENCOUNTERED :

**During the development and training of the classification model, several key challenges were encountered:**

- **Visual Ambiguity: A primary difficulty was the visual similarity between different plant diseases. Distinguishing subtle differences between visually similar symptoms required the model to learn fine-grained, complex features.**

- **Overfitting: Overfitting was a recurring issue in initial training attempts. This made the implementation of strong regularization techniques, specifically Dropout layers and extensive data augmentation, necessary to ensure the model could generalize effectively to unseen data.**

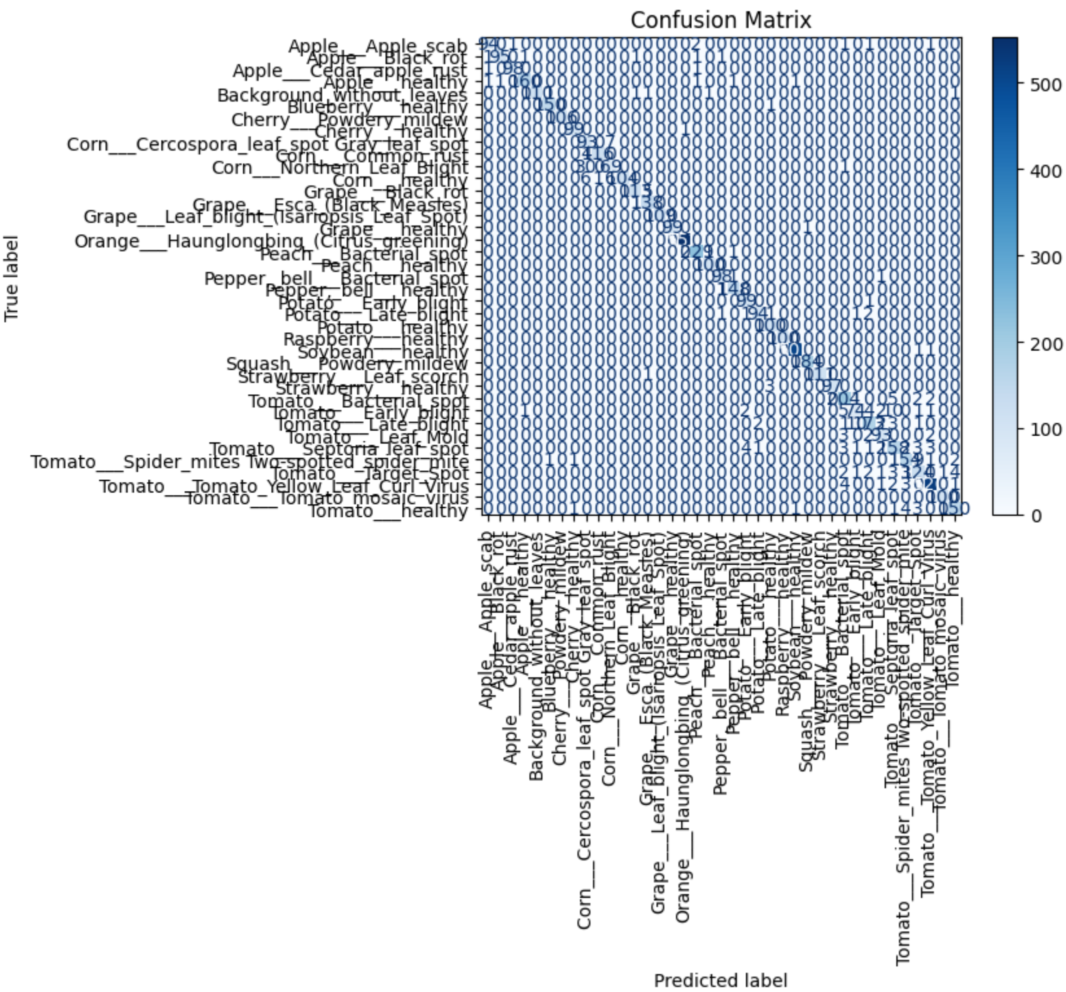
- **Computational Resources: Training the deep CNN architecture on the large PlantVillage dataset demanded significant hardware resources. This, along with the complexity of the model, occasionally caused the training process to take more time than initially expected.**

- **Data Handling: While the PlantVillage dataset is robust, the practical challenge of potentially collecting additional images and ensuring a consistently balanced representation across all 38 disease classes required careful management.**

# 12. LEARNING AND KEY TAKEAWAYS :

**Some plant diseases look very similar , which made the classification difficult . Overfitting occurred in earlier attempts, so techniques likes dropout and data augmentation were necessary . Training the model required good hardware, and sometimes training took more time than expected . Collecting additional images and balancing the dataset was also challenging .**

## ! CONFUSION MATRIX :



# 13. FUTURE ENHANCEMENTS :

**The system can be improved by converting it into a mobile app for real – time detection . More plant species and diseases can be added to increase the system's usefulness . The model can be upgraded using Vision Transformer (ViT) models or other advanced architectures. A recommendation system could also be added to provide treatment and preventive measures based on the detected disease.**

## ! Keras Functional Model Architecture Diagram: