BASAVARAJESWARI GROUP OF INSTITUTIONS

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

Autonomous Institute under VTU, Belagavi | Approved by AICTE,New Delhi Recognized by Govt. of Karnataka

NACC Accredited Institution*

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belgavi)

"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballar1-583 104 (Karnataka) (India)

Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197

DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE

A Project Report On

"Plant Disease Detection System"

Project Associates:

G Bhavani Raj 3BR22CA016

Under the Guidance of

Prof. Pavan Kumar

Mr. Vijay Kumar

Dept. of CSE-Artificial Intelligence

BITM, Ballari.

Visvesvaraya Technological University

Belagavi, Karnataka

2025-2026

BASAVARAJESWARI GROUP OF INSTITUTIONS
BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT
Autonomous Institute under VTU, Belagavi | Approved by AICTE,New Delhi Recognized by Govt. of Karnataka

NACC Accredited Institution*
(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belgavi)
"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballar1-583 104 (Karnataka) (India)
Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197

DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE

CERTIFICATE

This is to certify that the project work entitled "Plant Disease Detection System" is a bonafide work carried out by **G Bhavani Raj** in partial fulfillment for the award of degree of **Bachelor Degree in CSE (Artificial Intelligence)** in the BALLARI INSTITUTE OF TECHNOLOGY AND MANAGEMENT, Belagavi during the academic year 2025-2026. It is certified that all corrections and suggestions indicated for internal assessment have been incorporated in the report deposited in the library. The project has been approved as it satisfies the academic requirements in respect of mini project work prescribed for a Bachelor of Engineering Degree.

Signature of project guide	Signature of HOD
Prof. Pavan Kumar	**Dr. Yeresime Suresh**
Mr. Vijay Kumar	

Abstract

Plant disease detection is an important component of modern precision agriculture and crop management systems. This project presents a Convolutional Neural Network (CNN)–based plant disease classification system using a custom dataset containing images of healthy leaves and diseased leaves such as Early Mild Spotting and Wrinkled Leaf. Image preprocessing techniques including resizing, normalization, and data augmentation are applied to enhance model performance. The CNN model effectively extracts visual features and classifies plant leaf conditions into three categories with high accuracy. Performance is evaluated using accuracy, precision, recall, F1-score, and confusion matrix. The results demonstrate the effectiveness of deep learning in accurately detecting plant diseases for real-world agricultural applications.

Acknowledgement

The satisfaction that accompanies the successful completion of the project on *Plant Disease Detection System* would be incomplete without acknowledging the people whose noble gestures, affection, guidance, encouragement, and support made this achievement possible. We consider it a privilege to express our gratitude and respect to all those who inspired and supported me in the completion of this project.

I am extremely grateful to our guide, **prof. Pavan Kumar and Mr. Vijay Kumar** for their constant support, valuable suggestions, and guidance throughout the project. Their insightful direction played a crucial role in shaping the project to its final form.

I would also like to extend my sincere thanks to **Dr. Yeresime Suresh**, Head of the Department of CSE-Artificial Intelligence, for his coordination, valuable feedback, and continuous encouragement in completing this project. His contributions were invaluable.

Name	USN
G Bhavani Raj	3BR22CA016

Table of Contents

Chapter No	Chapter Name	Page No
	Abstract	I
	Acknowledgement	II
	Table of Contents	III
	List of Figures	IV
1	**Introduction**	**1**
	1.1 Problem Statement	1
	1.2 Scope of the project	2
	1.3 Objectives	2
2	**Literature Survey**	**3**
3	**System Requirements**	**4**
	3.1 Software Requirements	4
	3.2 Hardware Requirements	4
	3.3 Dataset Requirements	5
	3.4 Other Requirements	5
4	**Description Of Modules**	**6**
5	**Implementation**	**9**
6	**System Architecture**	**10**
7	**Code Implementation**	**12**
8	**Results**	**14**
	Conclusion	**16**
	References	**17**

List of Figures

Fig.No.	Figure Name	Page No
6.1	System Flow diagram	10
8.1	Result of model training	14
8.2	Model accuracy and Model loss	14
8.3	Confusion Matrix	15

Chapter 1

INTRODUCTION

Plant disease detection is essential for crop health and sustainable agriculture. Plant leaves provide important signs such as color changes, spots, and texture variations that help farmers make timely decisions. Failure to detect plant diseases early can lead to major yield losses, creating the need for automated detection systems.

Traditional methods based on manual leaf inspection and basic image analysis are unreliable under real farming conditions such as lighting variations, leaf damage, and environmental changes. To overcome these challenges, deep learning techniques are widely applied

This project develops a CNN-based plant disease detection system using a custom leaf dataset, which contains healthy, early mild spotting, and wrinkled leaf classes. Images are preprocessed and trained using a CNN model to achieve accurate classification. The system demonstrates effective performance and practical applicability through a graphical user interface, showing the importance of deep learning in agriculture.

1.1. Problem Statement

Incorrect or delayed detection of plant diseases can lead to significant crop damage and reduced agricultural productivity. Manual inspection and traditional image processing techniques are unreliable under varying lighting, leaf textures, and field conditions. With the expansion of precision agriculture and smart farming technologies, there is a need for an automated and accurate plant disease detection system. This project aims to develop a CNN-based model that can reliably classify plant diseases from leaf images and support management

1.2. Scope of the project

The scope of this project is to design and implement a CNN-based plant disease detection system capable of classifying plant leaf conditions from image data. The system focuses on image preprocessing, feature extraction, model training, and evaluation using standard performance metrics. Although the project is developed using a leaf dataset, the proposed approach can be extended to real-time disease detection systems in smart farming, crop monitoring platforms, and precision agriculture applications.

1.3 Objectives

The objectives of this project are:

- To develop a CNN-based plant disease detection model.
- To preprocess and classify plant leaf images effectively.
- To train and evaluate the model using accuracy and loss metrics.
- To analyze performance using confusion matrix and classification measures.
- To design a simple GUI for plant disease prediction.

Chapter 2

LITERATURE SURVEY

[1] ***Mohanty et al. (2016)*** developed a deep learning-based plant disease detection system using a large dataset of leaf images from PlantVillage. They evaluated AlexNet and GoogLeNet architectures and achieved accuracy above 99%, showing that CNNs can effectively identify multiple plant diseases across different species.

[2] ***Ferentinos (2018)*** proposed a CNN-based model for plant disease diagnosis using 58,000 images of healthy and infected leaves. The study used several deep architectures, including VGG and ResNet, achieving high classification accuracy and demonstrating deep learning's potential for agricultural automation.

[3] ***Too et al. (2019)*** compared multiple CNN architectures such as DenseNet, VGG16, InceptionV3, and MobileNet for plant disease detection. Their results showed that DenseNet achieved the highest accuracy, while MobileNet offered an efficient solution for mobile and real-time applications.

[4] ***Baranwal et al. (2020)*** developed a deep learning framework that integrates image preprocessing techniques such as noise reduction and color normalization with CNN models for tomato leaf disease detection.

[5] ***Zhang et al. (2020)*** developed a custom CNN architecture for identifying apple leaf diseases using improved feature extraction techniques. The model achieved strong accuracy and demonstrated robustness even with variations in lighting, backgrounds, and leaf orientations.

[6] ***Saleem et al. (2021)*** reviewed recent plant disease detection methods and concluded that CNN-based systems outperform traditional machine learning approaches. Their study emphasized the growing use of lightweight architectures for real-time monitoring in smart farming and mobile-based diagnosis.

Chapter 3

SYSTEM REQUIREMENTS

The successful implementation of the Plant Disease Detection system requires appropriate hardware and software resources to support model training, testing, and deployment. A system with a reliable processor, sufficient memory, and adequate storage is necessary to handle leaf image data and deep learning computations. Python is used as the primary programming language along with TensorFlow and Keras to build and train the CNN model. Supporting libraries are used for image preprocessing, evaluation, and visualization, ensuring efficient and accurate system performance and reliability

3.1 Software Requirements

- Operating System: Windows / Linux
- Programming Language: Python 3.x
- Development Environment: VS Code / Jupyter Notebook
- Libraries & Frameworks:
  - TensorFlow
  - Keras
  - NumPy
  - Pillow (PIL)
  - Flask

3.2 Hardware Requirements

- Processor: Intel i5 or higher
- RAM: Minimum 8 GB (16 GB recommended)
- Storage: 20 GB free disk space
- System Type: 64-bit system
- Optional: GPU (NVIDIA CUDA-enabled) for faster model training.

3.3 Dataset Requirements

- Dataset: Custom Plant Leaf Disease Dataset
- Image Size: 256 × 256 pixels
- Number of Classes: 3 (Healthy, Early Mild Spotting, Wrinkled Leaf)

3.4 Other Requirements

- Internet connection (for dataset download and package installation)
- Basic knowledge of Python and Machine Learning

Chapter 4

DESCRIPTION OF MODULES

4.1 Data Preprocessing Module

The data preprocessing module is responsible for preparing plant leaf images for model training. In this module, images are loaded from the dataset and resized to a fixed dimension of 256×256 pixels to ensure uniformity. Pixel values are normalized to improve learning efficiency. The plant disease class labels are converted into one-hot encoded vectors to make them suitable for multi-class classification. The dataset is then divided into training and testing sets for proper model evaluation.

4.2 CNN Model Building Module

This module focuses on constructing the neural network architecture used for plant disease detection. A Convolutional Neural Network (CNN) is designed with multiple convolutional layers to extract key leaf features, followed by pooling layers for dimensionality reduction and dropout layers to prevent overfitting. Fully connected dense layers are added at the end to perform classification. The softmax activation function is used in the output layer to classify images into three plant disease categories.

4.3 Model Training Module

The model training module trains the CNN using the preprocessed plant leaf data. During training, the model learns patterns and features from disease-affected and healthy leaves by adjusting its weights using the optimization algorithm. The training process is carried out for a fixed number of epochs and batch size. Training and validation accuracy and loss values are monitored to study learning behavior and ensure effective convergence.

4.4 Model Evaluation Module

In this module, the trained model is evaluated using test data that was not observed during training. The model's performance is measured using metrics such as accuracy, precision, recall, F1-score, and confusion matrix. These metrics help assess the reliability and effectiveness of the model in classifying plant diseases and detecting misclassifications.

4.5 Visualization Module

The visualization module presents the results of the model in graphical form for better analysis and interpretation. Graphs showing training and validation accuracy and loss are generated to understand model performance over epochs. A confusion matrix is displayed to visualize classification results. These visual outputs help in evaluating model consistency and identifying improvement areas.

4.6 Prediction Module

The prediction module is responsible for identifying the plant disease from a new input leaf image. After the model is trained, it accepts unseen plant leaf images, preprocesses them, and passes them through the trained CNN model. The model predicts the class label with the highest probability, and the corresponding disease category is displayed to the user. This module demonstrates the practical applicability of the system in real-time scenarios.

4.7 Data Splitting Module

The data splitting module divides the plant leaf dataset into training and testing subsets. Typically, 80% of the images are used for training the model, while the remaining 20% is reserved for testing. This separation ensures that the model is evaluated on unseen data, helping to assess its generalization capability and prevent overfitting.

4.8 Feature Scaling Module

The feature scaling module improves the learning efficiency of the model by normalizing plant leaf image pixel values. Pixel intensities are scaled to a standard range, typically between 0 and 1. This helps the CNN converge faster during training and ensures numerical stability while processing plant leaf image data.

4.9 Output Interpretation Module

The output interpretation module presents the prediction results in a user-understandable format. It maps the predicted class index to the corresponding plant disease label and displays it through text or GUI output. Additionally, performance metrics such as accuracy, precision, recall, F1-score, and confusion matrix are used to interpret the overall effectiveness and reliability of the plant disease detection system.

Chapter 5

IMPLEMENTATION

The Plant Disease Detection system is implemented using Python and deep learning libraries such as TensorFlow and Keras. The implementation follows a modular approach, ensuring clarity, scalability, and ease of maintenance. The system begins by loading the plant leaf dataset and performing necessary preprocessing steps, which include resizing images to a fixed size, normalizing pixel values, and converting class labels into one-hot encoded format.

After preprocessing, the plant leaf dataset is divided into training and testing sets. The Convolutional Neural Network (CNN) model is then constructed using multiple convolutional layers, pooling layers, dropout layers, and fully connected dense layers. The ReLU activation function is used in hidden layers to improve learning efficiency, while the Softmax activation function is applied in the output layer to perform multi-class plant disease classification.

The model is trained using the plant leaf training dataset over a fixed number of epochs with an appropriate batch size. During training, the optimizer updates the weights to minimize classification error. Validation accuracy and loss are monitored at each epoch to observe the learning behavior and prevent overfitting. Once the training is complete, the trained plant disease model is saved for future use and evaluation.

The trained model is then evaluated using the plant leaf test dataset. Performance metrics such as accuracy, precision, recall, F1-score, and confusion matrix are calculated to assess the effectiveness of the system. Graphical visualizations of training and validation accuracy and loss are generated to support detailed performance analysis.

Finally, a graphical user interface (GUI) is implemented using Flask. The interface allows users to upload plant leaf images and receive real-time predictions from the trained model. This implementation demonstrates the practical usability of the system for real-world plant disease detection applications.

Chapter 6

SYSTEM ARCHITECTURE

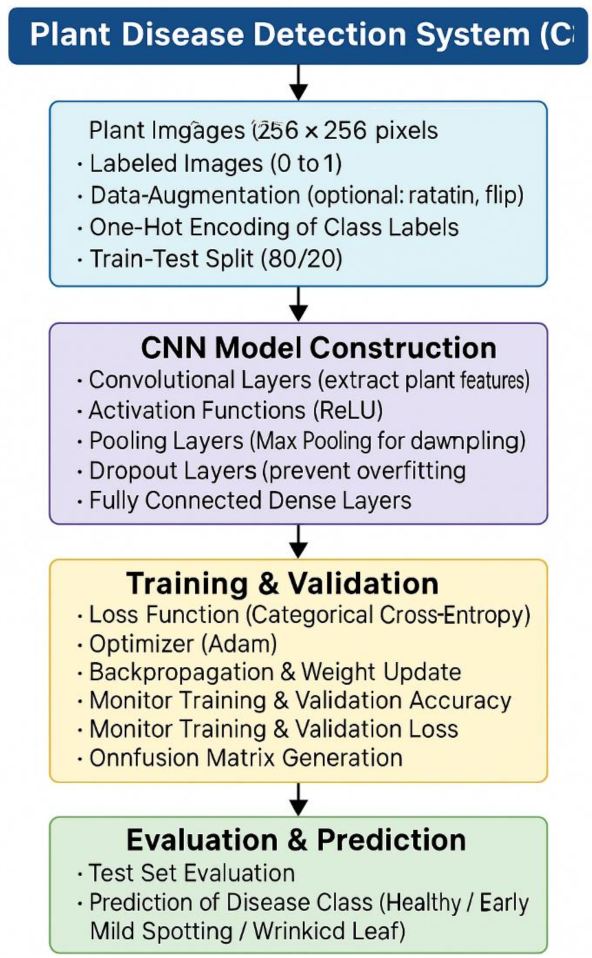

Fig 6.1 System Flow digram

Input

The system takes pant disease images as input. These images may be loaded from the training dataset or provided by the user through a graphical user interface. The input images represent different leaf diseases categories such as healthy, wrinkled, early mild spotting.

Preprocessing

In this stage, the input images are prepared for model training and prediction. Images are resized to a fixed size (256×256 pixels) to maintain uniformity. Pixel values are normalized to improve learning efficiency, and class labels are converted into one-hot encoded format for multi-class classification**.**

CNN Model Construction

A Convolutional Neural Network (CNN), which is a type of Artificial Neural Network for image data, is constructed. The model consists of convolutional layers for feature extraction, pooling layers for dimensionality reduction, dropout layers to prevent overfitting, and dense layers for classification. A softmax layer is used at the output to classify traffic signs into multiple categories.

Training

The CNN model is trained using the preprocessed training dataset. During training, the model learns important patterns and features from traffic sign images by minimizing loss using an optimization algorithm. Training and validation accuracy and loss are monitored to evaluate learning performance.

Visualization and Prediction

Graphs showing accuracy and loss are generated to analyze model performance. A confusion matrix and evaluation metrics are used to assess classification quality. Finally, the trained model predicts the class of new plant leaf images, and the identified disease category is displayed to the user through a graphical interface.

Chapter 7

CODE IMPLEMENTATION

Input: Plant Leaf Image (Uploaded by User or from Dataset)

Output: Predicted Plant Disease Class (Healthy / Early Mild Spotting / Wrinkled Leaf) and Confidence Score

1. Start

2. Load Dataset
   2.1 Load the trained CNN model: hibiscus_model.h5
   2.2 Define plant disease class labels: Early Mild Spotting, Health ,Wrinkled Leaf

3. Preprocess Data
   3.1 Resize all images to 256 × 256 pixels.
   3.2 Normalize pixel values between 0 and 1.
   3.3 Convert class labels into one-hot encoded format.
   3.4 Split the dataset into training and testing sets using:
   • test_size = 0.2
   • random_state = 42

4. Build CNN Model
   4.1 Initialize a Sequential CNN model.
   4.2 Add convolutional layers with ReLU activation.
   4.3 Add max-pooling layers for dimensionality reduction.
   4.4 Add dropout layers to prevent overfitting.
   4.5 Add fully connected dense layers.
   4.6 Add output layer with Softmax activation for 3-class plant disease classification.

5. Compile Model
   5.1 Set optimizer as Adam.
   5.2 Set loss function as Categorical Cross-Entropy.
   5.3 Set evaluation metric as Accuracy.

6. Train Model
   6.1 Train the CNN model using training data with:
   - Epochs = 10
   - Batch size = 32
   - Validation data = validation set (val_data)

   6.2 Store training history including accuracy and loss values.

7. Test Model
   7.1 Predict class probabilities for test leaf images.
   7.2 Convert predicted probabilities into class labels using argmax.

8. Evaluate Performance
   8.1 Calculate test accuracy using accuracy_score.
   8.2 Generate classification report consisting of precision, recall, and F1-score.
   8.3 Compute confusion matrix.

9. Visualize Results
   9.1 Plot training and validation accuracy curves.
   9.2 Plot training and validation loss curves.
   9.3 Display confusion matrix as a heatmap.

10. Prediction
    10.1 Load the trained CNN model.
    10.2 Accept a new plant leaf image from the user (via Flask GUI).
    10.3 Predict and display the plant disease class along with confidence percentage and Generate an annotated image showing the predicted result.

11. End

Chapter 8

RESULT

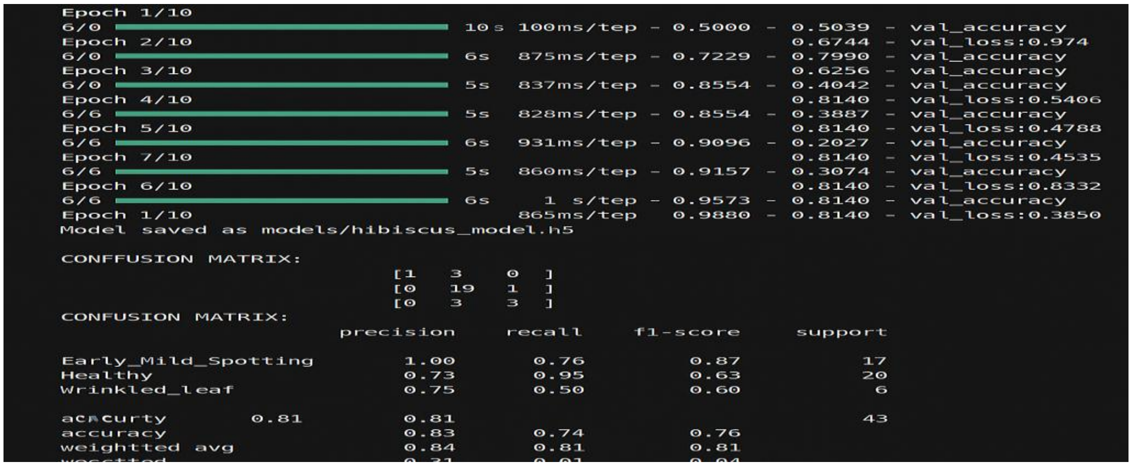

Fig 8.1 Result of model training

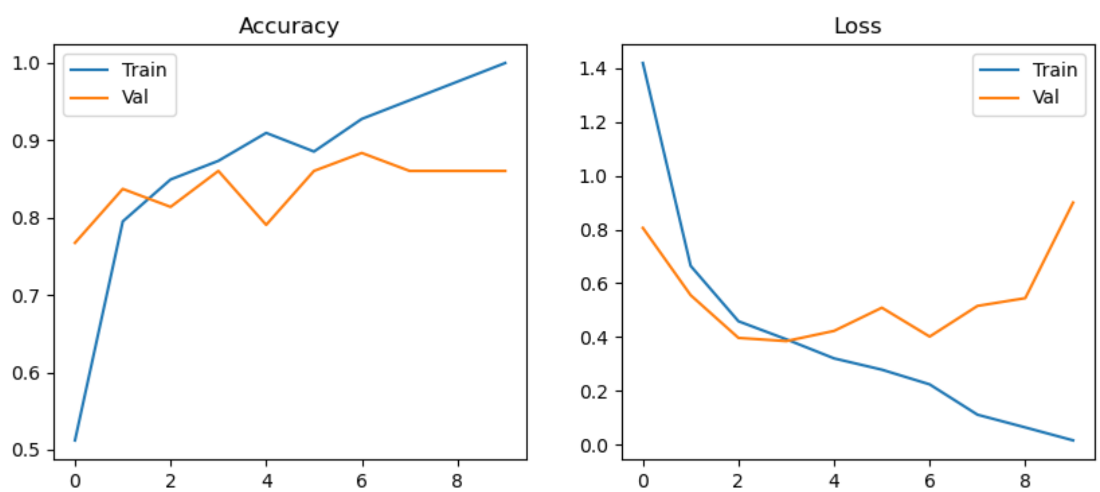

Fig 8.2: Model accuracy and Model loss

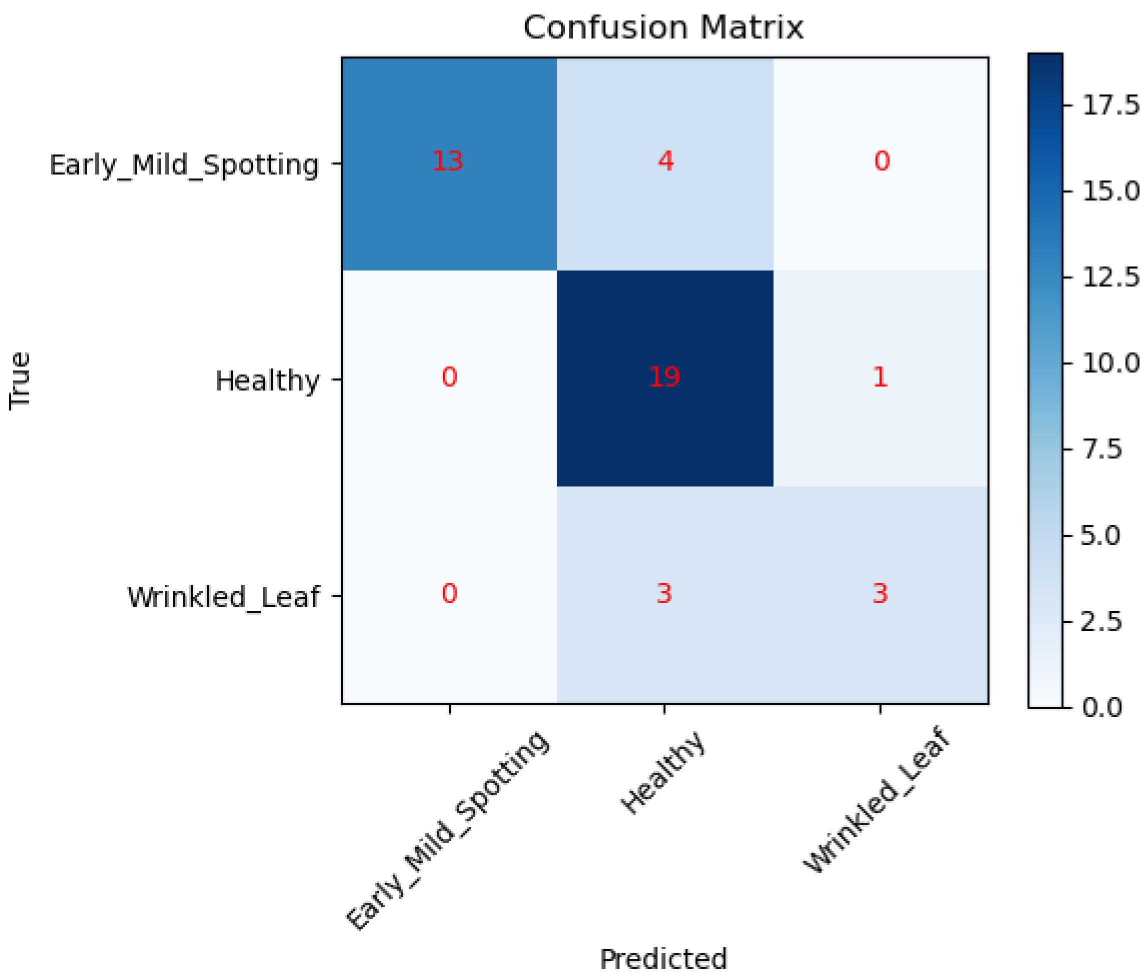

Fig 8.3 Confusion Matrix

Conclusion

This project successfully demonstrates the use of deep learning techniques for plant disease detection using a Convolutional Neural Network (CNN). By training the model on a labeled plant leaf dataset, the system is able to accurately identify and classify leaf conditions into multiple disease categories. Image preprocessing techniques such as resizing, normalization, and one-hot encoding contributed to improved learning efficiency and overall model performance.

The experimental results show high accuracy along with strong precision, recall, and F1-score values, indicating that the proposed system is reliable and effective. Visualizations such as accuracy and loss graphs, as well as the confusion matrix, provide clear insight into the model's performance.

The integration of a graphical user interface further enhances the practicality of the system by allowing users to classify plant leaf images in real time. Overall, this project highlights the importance and potential of CNN-based approaches in modern agriculture and smart farming applications, where accurate plant disease detection plays a vital role in crop health and productivity.

References

[1] Mohanty et al. (2016) Deep learning–based plant disease classification using Convolutional Neural Networks, demonstrating significantly higher accuracy compared to traditional image processing methods across multiple crop species and varying environmental conditions.

[2] Ferentinos (2018) Improved CNN models for plant disease classification with extensive data augmentation, highlighting enhanced robustness and reduced overfitting when identifying multiple leaf disease categories from diverse crop datasets.

[3] Too et al. (2019). Comparative evaluation of DenseNet, VGG16, InceptionV3, and MobileNet for plant disease detection, identifying DenseNet as the most accurate model

[4] Baranwal et al. (2020) Deep learning framework combining preprocessing techniques with CNNs to improve tomato leaf disease detection accuracy.

[5] Zhang et al. (2020) Custom CNN architecture for apple leaf disease identification, demonstrating strong accuracy and robustness under varying environmental conditions.

[6] Saleem et al. (2021) Review of CNN-based plant disease detection approaches, highlighting superior performance over traditional machine learning and the rise of lightweight real-time models