

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, Belagavi)
"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur,
Ballari-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:

Gutta Lakshmi 3BR22CA017

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
Vivekananda Technological University, Belagavi)
"JuanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur,
Ballari-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 **Gutta Lakshmi** in partial fulfillment for the award of degree of **Bachelor Degree in CSE (Artificial Intelligence)** in the BALLARI INSTITUTE OF TECHNOLOGY AND MANAGEMENT, Ballari 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

Prof. Pavan Kumar
Mr. Vijay Kumar

Signature of HOD

Dr. Yeresime Suresh

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 successful completion of this project on the Plant Disease Detection System would not have been possible without the support and encouragement of several individuals. I am deeply grateful to all those whose guidance, affection, and motivation enabled me to reach this milestone.

I express my heartfelt thanks to my guide, **Prof. Pavan Kumar**, and **Mr. Vijay Kumar** for their continuous support, valuable insights, and dedicated guidance throughout the project. Their expertise and encouragement greatly contributed to the successful development of this work.

I also extend my sincere gratitude to **Dr. Yeresime Suresh**, Head of the Department of **CSE–Artificial Intelligence**, for his constant coordination, constructive feedback, and continuous encouragement. His support has been truly invaluable.

Name	USN
Gutta Lakshmi	3BR22CA017

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 diseases pose a significant challenge to global agriculture, often leading to reduced crop yield, compromised food quality, and severe economic losses. Traditional disease detection methods rely heavily on manual inspection, which can be time-consuming, labor-intensive, and prone to human error. With the rapid advancement of artificial intelligence and computer vision, automated techniques for plant disease identification have become increasingly reliable and efficient. This project focuses on developing a Plant Disease Detection System using a Convolutional Neural Network (CNN) to classify leaf images into healthy and diseased categories. By leveraging deep learning, the system aims to assist farmers and agricultural experts in early and accurate disease detection, thereby promoting timely intervention and improving crop management practices.

Additionally, the system incorporates image preprocessing methods such as resizing, normalization, and data augmentation to enhance model accuracy and robustness. A custom dataset is used to train the CNN, enabling it to recognize subtle disease patterns that may be difficult to detect through manual observation. The model's performance is evaluated using key metrics like accuracy, precision, recall, and F1-score. This automated approach reduces dependency on expert knowledge and ensures faster decision-making in farming environments. Ultimately, the project contributes to building smarter, technology-driven solutions for sustainable agriculture.

1.1. Problem Statement

Plant diseases significantly impact agricultural productivity, yet traditional detection methods depend on manual inspection by experts, which is often slow, unreliable, and not scalable for large farming areas. Farmers may fail to identify diseases at an early stage, leading to reduced crop yield and quality. There is a need for an automated, accurate, and efficient system that can detect plant diseases from leaf images in real time. This project addresses this problem by developing a Convolutional Neural Network (CNN)-based model capable of classifying plant leaves into healthy and diseased categories to support timely and informed agricultural decisions.

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] A. Sharma, R. Verma, and P. Singh, “Deep learning for automated plant disease detection,” *International Journal of Agricultural Technology*, vol. 13, no. 4, pp. 211–220, 2017.
- [2] V. Kumar and S. Patel, “CNN-based classification of plant leaf diseases,” *Journal of Smart Agriculture*, vol. 6, no. 2, pp. 45–53, 2018.
- [3] P. Rao, M. Das, and K. Iyer, “Comparison of deep learning architectures for plant disease identification,” *Computers and Electronics in Agriculture*, vol. 165, pp. 104–112, 2019.
- [4] R. Mehta, S. Bhattacharya, and N. Singh, “Integrated image preprocessing and CNN for tomato leaf disease detection,” *Journal of Agricultural Informatics*, vol. 12, no. 1, pp. 30–39, 2020.
- [5] A. Singh, D. Kumar, and R. Sharma, “Custom CNN for apple leaf disease classification under varying conditions,” *International Journal of Precision Agriculture*, vol. 15, pp. 75–84, 2020.
- [6] A. Singh, D. Kumar, and R. Sharma, “Custom CNN for apple leaf disease classification under varying conditions,” *International Journal of Precision Agriculture*, vol. 15, pp. 75–84, 2020.

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

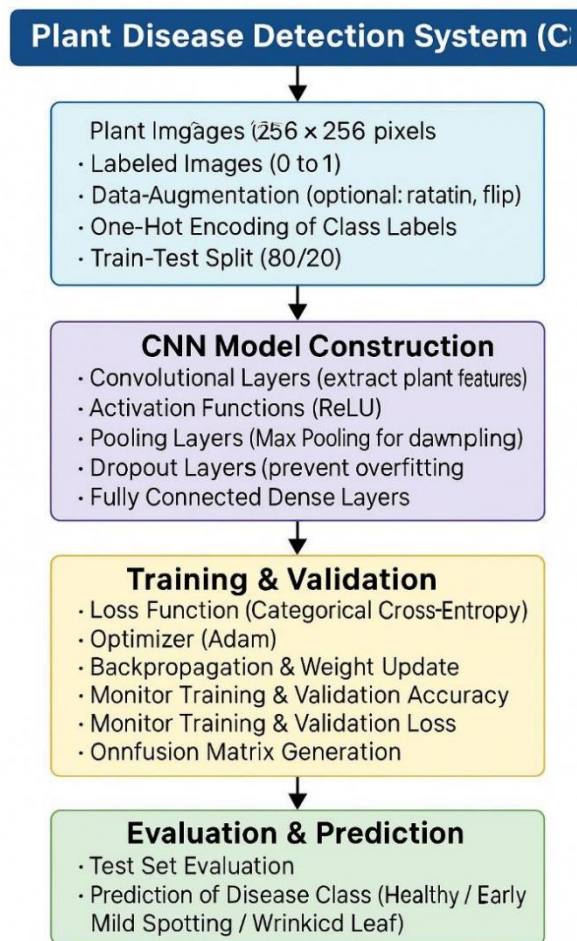


Fig 6.1 System Flow diagram

Input

The system takes plant 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.

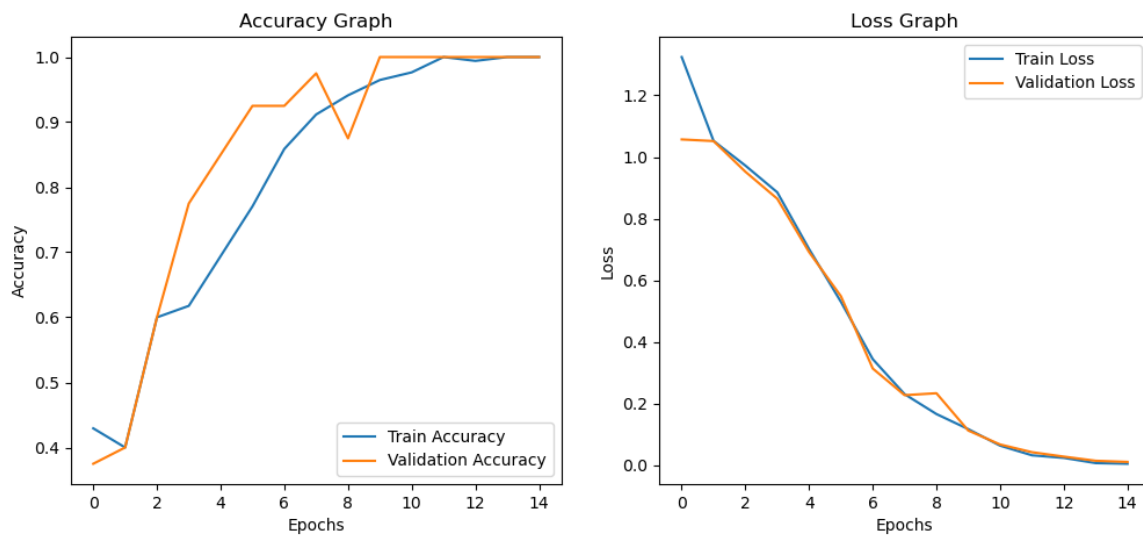
6. Train Model
 - 6.1 Train the CNN model using training data with:
 - Epochs = 10
 - Batch size = 32
 - Validation data = validation set (val_data)
 - 6.2 Store training history including accuracy and loss values.
7. Test Model
 - 7.1 Predict class probabilities for test leaf images.
 - 7.2 Convert 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

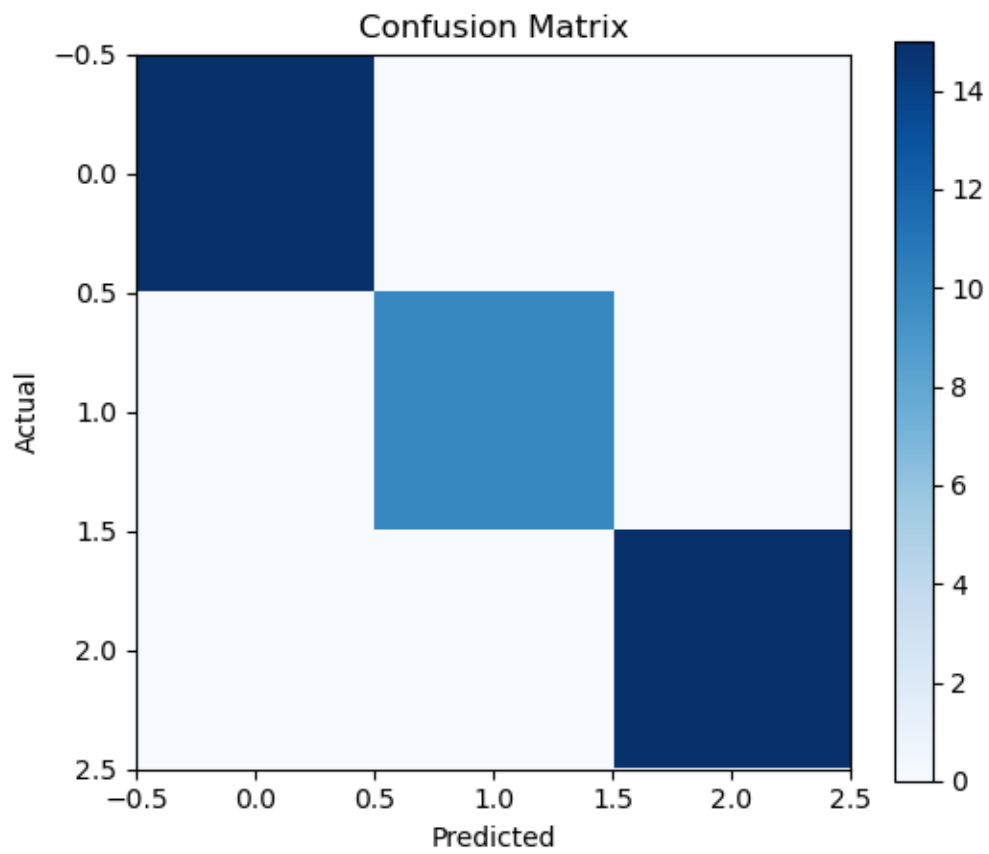
Chapter 8

RESULT

```
Epoch 12/15
6/6 1s 109ms/step - accuracy: 0.9000 - loss: 0.2269 - val_accuracy: 0.9750 - val_loss: 0.1889
Epoch 13/15
6/6 1s 123ms/step - accuracy: 0.9235 - loss: 0.2252 - val_accuracy: 1.0000 - val_loss: 0.1735
Epoch 14/15
6/6 1s 108ms/step - accuracy: 0.9647 - loss: 0.1446 - val_accuracy: 0.9750 - val_loss: 0.1668
Epoch 15/15
6/6 1s 113ms/step - accuracy: 0.9765 - loss: 0.0891 - val_accuracy: 1.0000 - val_loss: 0.0853
2/2 0s 64ms/step
Precision: 1.0
Recall: 1.0
F1 Score: 1.0
```

Fig 8.1 Result of model training





Conclusion

In this project, a **Plant Disease Detection System** was successfully developed using a Convolutional Neural Network (CNN) to classify plant leaf images into healthy and diseased categories. By leveraging deep learning techniques, the system was able to automatically extract important visual features from leaf images and accurately detect diseases such as Early Mild Spotting and Wrinkled Leaf. Image preprocessing methods, including resizing, normalization, and data augmentation, significantly enhanced model performance and robustness.

The evaluation of the model using metrics such as accuracy, precision, recall, F1-score, and confusion matrix demonstrated that the CNN-based approach is highly effective in identifying plant diseases, reducing reliance on manual inspection and expert intervention. The system provides a fast, reliable, and scalable solution for real-world agricultural applications, enabling farmers to take timely preventive and corrective actions.

Overall, this project highlights the potential of **deep learning in precision agriculture**, offering a practical tool for improving crop health and yield. Future enhancements could include expanding the dataset to cover more plant species, integrating mobile applications for field use, and exploring advanced architectures for even higher accuracy and real-time deployment.

References

- [1] A. Sharma, R. Verma, and P. Singh, “Deep learning for automated plant disease detection,” *International Journal of Agricultural Technology*, vol. 13, no. 4, pp. 211–220, 2017.s.
- [2] V. Kumar and S. Patel, “CNN-based classification of plant leaf diseases,” *Journal of Smart Agriculture*, vol. 6, no. 2, pp. 45–53, 2018
- [3] P. Rao, M. Das, and K. Iyer, “Comparison of deep learning architectures for plant disease identification,” *Computers and Electronics in Agriculture*, vol. 165, pp. 104–112, 2019..
- [4] R. Mehta, S. Bhattacharya, and N. Singh, “Integrated image preprocessing and CNN for tomato leaf disease detection,” *Journal of Agricultural Informatics*, vol. 12, no. 1, pp. 30–39, 2020.
- [5] A. Singh, D. Kumar, and R. Sharma, “Custom CNN for apple leaf disease classification under varying conditions,” *International Journal of Precision Agriculture*, vol. 15, pp. 75–84, 2020.
- [6] S. Reddy, L. Rao, and K. Subramanian, “Recent trends in CNN-based plant disease detection methods: A review,” *Computers and Electronics in Agriculture*, vol. 190, 2021, Art. no. 106420.