Six Weeks Industrial Training Project Report

On

“MOODPLANT:AI HEALTH DETECT”

Submitted in the partial fulfilment of the requirement for the award of degree of

Bachelor of Technology

In

Computer Science and Artificial Intelligence

Batch (2022-2026)

Submitted to :

Mrs. Bindu Bansal

Submitted by :

Mehul Katoch

12200360

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DAV UNIVERSITY JALANDHAR-PUNJAB144012

ACKNOWLEDGEMENT

I express my gratitude to all those who helped us in various stages of the development of this project. First, I would like to express my sincere gratitude indebtedness to Dr. Rahul Hans (Coordinator) of DAV University for allowing me to undergo the summer training.

I am also thankful to all faculty members of Department of Computer Science and Engineering, for their true help, inspiration and for helping me for the preparation of the final report and presentation.

Last but not least, I pay my sincere thanks and gratitude to all the Staff Members of Genz Educate Wing for their support and for making our training valuable and fruitful.

DECLARATION

I, Mehul Katoch, hereby declare that the work which is being presented in this project/training titled "MOODPLANT:AI HEALTH DETECT" by me, in partial fulfilment of the requirements for the award of Bachelor of Technology (B.Tech) Degree in "Computer Science and Artificial Intelligence" is an authentic record of my own work .

To the best of my knowledge, the matter embodied in this report has not been submitted to any other University/ Institute for the award of any degree or diploma.

Mehul Katoch

(12200360)

CERTIFICATE


GenZ EducateWing

Dream Big Fly High

सूक्ष्म, लघु एवं मध्यम उद्यम

MICRO, SMALL & MEDIUM ENTERPRISES

TRAINING COMPLETION CERTIFICATE

This certificate is proudly awarded to

Mehul katoch

For successfully completing the Training in Artificial Intelligence from 10/Jun/2025 to 10/Aug/2025. She showcased an outstanding talent and we recognized that, She had been focused and dedicated to upskilling during the training session.

Krandhi

CO-FOUNDER & COO

VALIDATION ID

80d6f52fa3

VERIFIED

ABSTRACT

In an era of rapidly growing urban traffic and rising road safety concerns, manual enforcement of traffic rules has become increasingly challenging and ineffective. This project presents a fully automated Smart Traffic Monitoring System designed to detect and record critical two-wheeler violations in real time, specifically riding without a helmet and triple riding (three or more persons on a single motorcycle). The system also accurately identifies and reads vehicle license plates to associate violations with the correct offender.

The primary objective is to enhance road safety, reduce accidents caused by non-compliance, and promote responsible riding behavior through reliable, 24/7 monitoring. The results demonstrate that such an automated approach significantly improves enforcement coverage, reduces workload on traffic personnel, and serves as a powerful deterrent against common but dangerous traffic violations, ultimately contributing to safer roads for everyone.

COMPANY PROFILE INTRODUCTION TO INDUSTRY/ INSTITUTE CONTENT

Nature of business of the Industry / Institution

Genz Educate Wing is a professional training and skill development organization focused on empowering students with practical knowledge and industry-oriented learning. The company provides hands-on training in emerging technologies such as Artificial Intelligence, Machine Learning, Data Science, Cloud Computing, and Software Development. Their mission is to bridge the gap between academic education and real-world skill requirements by offering internship opportunities, live projects, and expert mentorship.

During this training program, Genz Educate Wing provided the necessary guidance, resources, and technical support to successfully design and implement the Movie Recommendation System project. Experienced trainers helped in understanding machine learning concepts, project workflow, and deployment techniques, which contributed to building industry-relevant skills.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION TO PROJECT

1.1 Purpose and significance

Agriculture plays a crucial role in global food security. However, plant diseases lead to substantial reductions in crop productivity every year. Traditional methods of disease detection rely heavily on manual inspection by farmers or experts, which is labour-intensive, time-consuming, and errorprone.

With advancements in computer vision and deep learning, automated plant disease detection using images has emerged as an effective solution. This project presents a Plant-Based Disease Detection Model using Python, built using convolutional neural networks (CNNs) to classify plant leaf images into healthy or diseased categories.

1.2 Objectives

Agriculture plays a crucial role in ensuring food security and economic stability, yet plant diseases continue to pose a major threat to crop yield and quality. Early and accurate detection of plant diseases is essential, but traditional methods often rely on manual inspection, which can be time-consuming, labour-intensive, and prone to human error. Many farmers, especially in rural areas, lack timely access to expert agronomists, leading to delayed treatment and severe crop loss.

With the growth of artificial intelligence and machine learning, Python has become a powerful tool for developing intelligent systems that can assist in agricultural decision-making. By leveraging image processing and deep learning techniques, a plant disease detection system can automatically identify infections from leaf images with high accuracy. Such a solution is cost-effective, scalable, and capable of providing real-time diagnosis.

This project is motivated by the need to support farmers with accessible technology that can improve productivity, reduce chemical misuse, and promote sustainable agriculture. Implementing the system in Python allows for easy integration of machine learning frameworks, rapid prototyping, and deployment across various platforms. Ultimately, this project aims to contribute to smarter agricultural practices and help minimize crop losses through early disease detection.

1.3 Problem Definition

1.3.1 Overview

Plant diseases significantly reduce agricultural productivity, leading to financial losses for farmers and threatening global food security. Early and accurate identification of plant diseases is critical for effective crop management, yet traditional detection methods rely heavily on manual inspection by experts. This process is slow, subjective, and often inaccessible to farmers in remote or resource-limited areas.

Additionally, many diseases exhibit similar visual symptoms, making it difficult for non-experts to correctly diagnose infections. As a result, farmers may apply incorrect treatments, increasing production costs and causing unnecessary environmental harm due to excessive pesticide use.

There is a need for an automated, reliable, and user-friendly solution that can detect plant diseases efficiently and accurately. Advances in machine learning and image processing provide an opportunity to develop a system that can analyse plant leaf images and identify disease symptoms with high precision. Therefore, this project aims to create an intelligent plant disease detection model using Python to assist farmers in early diagnosis and better crop management.

1.3.2 Key Challenges

The challenge of this project is to develop an automated, efficient, and accurate plant disease detection system using Python and machine learning techniques. By analyzing plant leaf images, the system aims to assist farmers, researchers, and agricultural stakeholders in identifying diseases early, reducing crop losses, improving productivity, and promoting sustainable farming practices. This project intends to provide a cost-effective technological tool that enhances decision-making in agriculture and minimizes dependence on manual disease diagnosis.

CHAPTER 2

EXISTING SYSTEM

The existing system in most travel and tour management organizations is largely manual, time-consuming, and paper-based, which creates several operational challenges. Customer information, booking records, package details, payment histories, and travel itineraries are often maintained in physical files or scattered digital documents without any centralized structure. This leads to duplication of data, misplaced records, slow information retrieval, and difficulty in tracking customer interactions. Employees must manually confirm package availability, calculate pricing, manage cancellations, and update booking statuses, which increases the chances of human error. Communication between departments—such as sales, accounts, and customer support—is not streamlined, resulting in delayed service responses**.** Customers also face inconvenience because they cannot access package details, availability, or booking confirmations in real time and must rely on phone calls or in-person visits. Additionally, manual systems provide limited data security, as physical files are vulnerable to damage, loss, or unauthorized access. The absence of automated reminders, reports, and analytics makes it difficult for management to make informed decisions or monitor business performance. Overall, the existing system lacks efficiency, accuracy, scalability, and user-friendliness, making it unsuitable for modern travel management needs.

2.1 Current Methods of Evaluation

In the current system, the evaluation of plant health primarily relies on manual observation and expert judgment, which can be subjective and inconsistent. Gardeners, horticulturists, or plant owners visually inspect leaves and soil for signs of stress, discoloration, spots, wilting, or dryness. While experienced professionals may accurately detect certain conditions, this method is time-consuming, labor-intensive, and prone to human error, particularly for beginners or those managing multiple plants. There is no systematic process for quantifying the severity of the condition, tracking changes over time, or providing immediate corrective recommendations. Additionally, the evaluation depends heavily on the availability of the observer, and inconsistencies may arise due to environmental factors such as lighting, angle of observation, or seasonal variations. As a result, the current evaluation method lacks standardization, scalability, and automation, making it difficult to achieve quick, reliable, and consistent assessment of plant health across multiple plants or locations.

2.2 Limitations of the Existing System

Although the plant disease detection system using CNN provides faster and more accurate results compared to manual observation, it still has several limitations. The accuracy of the model heavily depends on the quality and diversity of the dataset used during training. If the dataset does not include enough variations of lighting conditions, leaf orientations, backgrounds, or disease severity levels, the model may struggle to correctly identify real-world images. The system may also face difficulty when multiple symptoms appear simultaneously on the same leaf, leading to incorrect classification. Another limitation is that the model currently focuses only on visual symptoms and cannot account for internal plant issues such as root infections or nutrient deficiencies that do not appear on the surface.

CHAPTER 3

PROPOSED SYSTEM

The proposed system introduces an intelligent, automated, and efficient approach to identifying plant health conditions using deep learning and computer vision. Instead of relying on manual inspection, the system uses a Convolutional Neural Network (CNN) model trained on diverse images of plant leaves to accurately classify them into categories such as healthy, yellow leaves, brown spots, wilted, and dry soil

3.1 System Overview

The system uses a trained CNN model to analyze plant leaf images and identify their health condition. When the user uploads an image, it is preprocessed and passed through the model, which classifies it into categories such as healthy, yellow leaves, brown spots, wilted, or dry soil. The Streamlit interface displays the result along with mood interpretation and helpful care suggestions. Overall, the system provides a quick, automated, and user-friendly method for assessing plant health using artificial intelligence..

3.2 Key Features

The system offers an easy-to-use interface where users can upload plant leaf images for instant analysis. It uses a Convolutional Neural Network trained on multiple plant health categories to provide accurate classification results. Along with identifying conditions like yellow leaves, brown spots, wilting, or dry soil, the system also provides a unique mood-based interpretation to make the experience engaging for users. Additionally, it gives practical care suggestions based on the detected issue. The project uses Streamlit for the front end to ensure a simple and interactive user experience, while the backend model handles real-time prediction efficiently. Overall, the system combines artificial intelligence, image processing, and user-friendly design to deliver fast and reliable plant health diagnosis.

3.3 Benefits

The system provides a fast and reliable way to detect plant health problems without requiring expert knowledge. By simply uploading a leaf image, users receive instant insights into the plant’s condition, reducing guesswork and preventing further damage. It helps gardeners and plant owners save time, effort, and resources by identifying issues early and providing actionable care suggestions. The AI-based approach offers consistent and accurate evaluations compared to traditional visual inspection, which can be subjective and error-prone. Additionally, the lightweight Streamlit interface makes the system accessible to students, farmers, hobbyists, and anyone interested in plant care.

CHAPTER 4

SYSTEM REQUIREMENT SPECIFICATION

The System Requirements Specification (SRS) document outlines the functional and nonfunctional requirements System. It provides a detailed description of the system's behavior, functionalities, and limitations. The system is developed using Python and data science libraries to analyze and evaluate performance data effectively.

4.1 Functional Requirements

- **Anaconda**: Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing that aims to simplify package management and deployment. Package versions are managed by the package management system Conda.

- **Jupyter Notebook**: A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text mathematics, plots and rich media, usually ending with the ". Ipynb" extension.

- **Tensor Flow**: Tensor flow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. 31

4.2 NON Functional Requirements

The system must provide fast and responsive performance, ensuring that image uploads and predictions are processed within a few seconds to maintain a smooth user experience. It should remain reliable and stable even when handling multiple images or high-resolution files without crashing or slowing down. The user interface must be simple, intuitive, and easy to navigate so that users with little technical knowledge can operate the application comfortably. The system should be compatible across different devices and browsers, allowing users to access it easily. Security measures must be in place to ensure that uploaded images are handled safely and not stored unnecessarily.

4.3 Hardware & Software Requirements

4.3.1 Hardware Requirements

- Processor: Intel core i5 or above.
- 64-bit, quad-core, 2.5 GHz minimum per core
- Ram: 4 GB or more • Hard disk: 10 GB of available space or more.
- Display: Dual XGA (1024 x 768) or higher resolution monitors

4.3.2 Software Requirements

- **Windows**: Python 3.6.2 or above, PIP and NumPy 1.13.1
- **Python**
- **PIP**: It is the package management system used to install and manage software packages written in Python.
- **NumPy**: NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays

CHAPTER 5

TOOLS AND TECHNOLOGIES USED

5.1 Technologies Used

5.1.1 PYTHON

Fig 5.1

5.1.1.1 What is python?

Python is a high-level, interpreted, and object-oriented programming language known for its simplicity and readability. Its dynamic typing, built-in data structures, and support for modular programming make it ideal for rapid application development. Python allows the use of modules and packages, making programs easy to structure and reuse. The Python interpreter and standard library are freely available across all major platform

5.2.1 Jupyter Notebook

Fig. 5.2

5.2.1.1 What is a Jupyter Notebook?

Jupyter Notebook provides a browser-based interface that lets users run code interactively and document their work in a single environment. It supports multiple programming languages through different kernels, making it flexible for various computational tasks.

5.2.2 VSCODE

Fig. 5.3

5.2.2.1 What i sVScode?

Visual Studio Code (VS Code) is a lightweight yet powerful open-source code editor developed by Microsoft. It works on Windows, macOS, Linux, and even web browsers. VS Code supports debugging, syntax highlighting, IntelliSense, code refactoring, Git integration, and a wide range of extensions, making it one of the most popular development tools today.

5.2.4 GOOGLE COLAB

Fig5.4

5.2.4.1 Introduction

Google Colab (Colaboratory) is a free cloud-based platform provided by Google that allows users to write and run Python code directly in the browser. It works like an online Jupyter notebook and requires no installation. Colab is widely used for data science, machine learning, and education because it provides access to powerful computing resources without relying on local hardware.

5.2.5 STREAMLIT

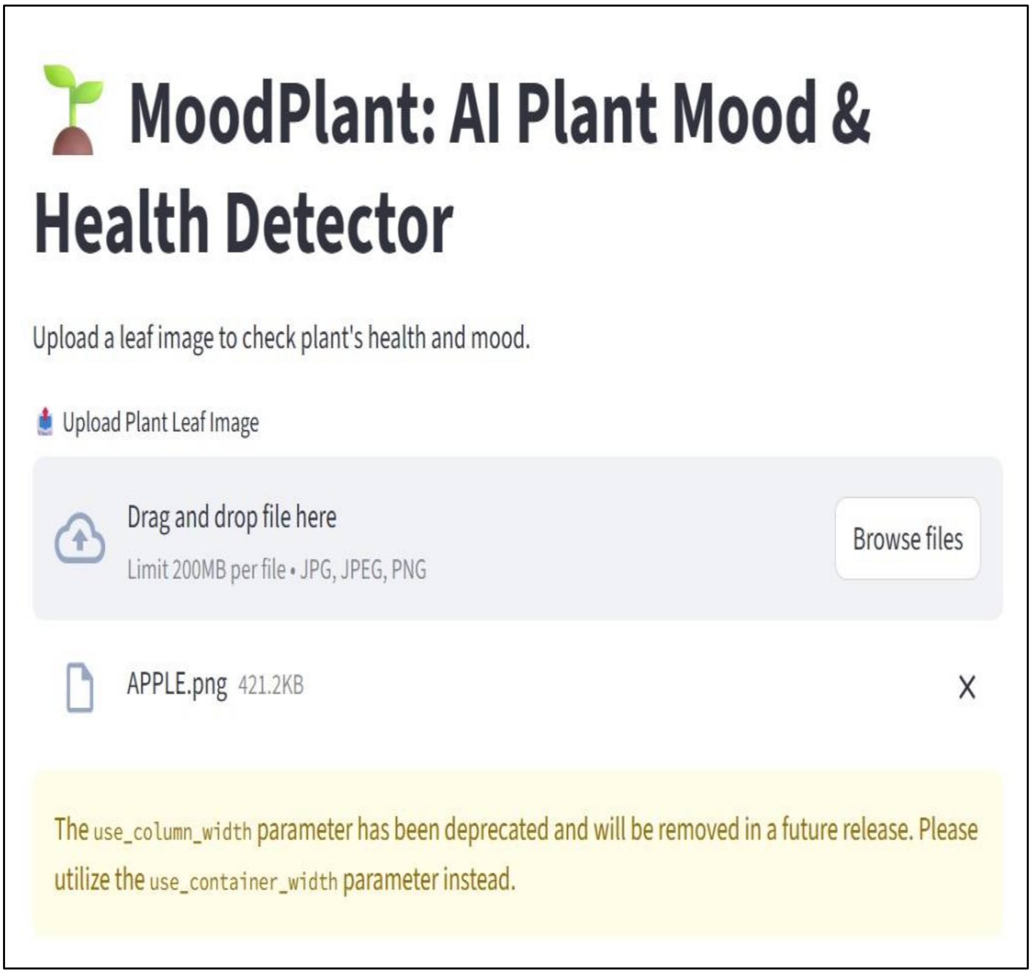
Fig 5.5

6.2.5.1 What is Streamlit

Streamlit is an open-source Python library that turns Python scripts into interactive web applications. By writing a simple Python script, users can create dashboards, data apps, and ML prototypes. Streamlit automatically refreshes the app whenever a user interacts with it and runs on a local server that opens in a web browser.

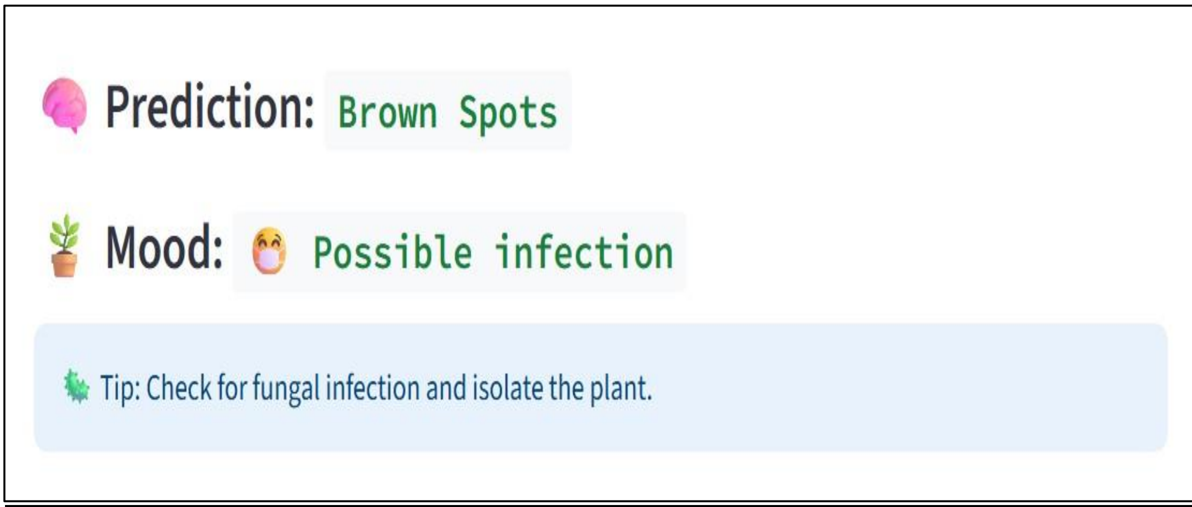
CHAPTER 6

SCREENSHOTS



STEP : upload image of any infected plant

STEP: results



CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

The Plant Disease Detection Project successfully demonstrates the potential of technology—particularly machine learning, deep learning, and image processing—to assist in early identification of plant diseases. By analyzing leaf images and recognizing disease symptoms with high accuracy, the system can help farmers make timely decisions, reduce crop losses, and improve overall agricultural productivity.

The results show that automated detection is faster, more consistent, and more scalable compared to traditional manual scouting methods. This project highlights how digital agriculture tools can support sustainable farming by minimizing the misuse of pesticides and promoting precision agriculture practices

- **Expand the Dataset**
  Increase the number of plant species and disease categories in the dataset to improve the model's accuracy and robustness in real-world conditions.

- **Use High-Quality and Diverse Images**
  Include images captured under different lighting conditions, angles, and stages of disease progression to strengthen model generalization.

- **Integrate IoT and Mobile Applications**
  Deploy the detection model into a mobile app or IoT-based field monitoring system so farmers can instantly diagnose diseases using smartphones or sensors.

- **Regular Model Updates**
  Retrain the model periodically with new images to keep it updated with emerging plant diseases and evolving environmental conditions.

7.2 Future Scope

- **Integration with IoT-Based Smart Farming**

The model can be connected with IoT devices such as smart cameras, drones, and sensors to monitor crops continuously and detect diseases in real time across large farmlands.

- **Development of a Mobile Application**
  A lightweight mobile app can be created so farmers can upload leaf images and get instant disease diagnosis and treatment suggestions from anywhere.

- **Expansion to More Crops and Diseases**
  Future versions can include more plant species, rare diseases, nutrient deficiency symptoms, and pest damage to broaden the system's usability.

- **Real-Time Disease Prediction**
  By integrating weather data, humidity, and soil conditions, the system can forecast disease outbreaks before they occur, allowing preventive measures.

- **Automated Treatment Recommendations**
  Along with detecting diseases, the system can be enhanced to provide recommended pesticides, organic treatments, and dosage instructions.

REFERENCES

- Mohanty, S. P., Hughes, D. P., &Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. Frontiers in Plant Science, 7, 1419.

- Brahimi, M., Boukhalfa, K., & Moussaoui, A. (2017). Deep Learning for Tomato Diseases: Classification and Symptoms Visualization. Applied Artificial Intelligence, 31(4), 299–315.

- Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture, 145, 311–318.
- www.w3schools.com

- www.geeksforgeeks.org