Plant Disease Detection System for Sustainable Agriculture

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning
with
TechSaksham – A joint CSR initiative of Microsoft & SAP

by

Shalu Choudhary, 22cse107@gweca.ac.in

Under the Guidance of

P.Raja, Master Trainer, Edunet Foundation

ACKNOWLEDGEMENT

This is an opportunity to express my heartfelt words for the people who were part of this Internship in numerous ways, people who gave me unending support right from the beginning of the industrial training. This experience has been both enriching and rewarding, and I am deeply thankful to all those who made it possible.

I want to give sincere thanks to my supervisor Mr. P.Raja, for being a great mentor and Edunet Foundation for the valuable support and advice. His guidance, insights, and continuous encouragement have helped me grow both professionally and personally. His talks and lessons not only help in project work and other activities of the program but also make me a good and responsible professional. I am also extremely grateful to TechSaksham – A joint CSR initiative of Microsoft and SAP or Edunet foundation.

……...

.....

……

….

……

ABSTRACT

The Plant Disease Detection System for Sustainable Agriculture aims to develop an automated, efficient, and reliable approach for identifying plant diseases in crops to promote sustainable agricultural practices and it also help farmers to find out the disease and increase their production.

The primary objective of this project is to design and implement a system capable of detecting plant diseases through image processing and machine learning algorithms. Machine learning models are trained on these images to classify plant diseases accurately. The system uses dataset which include all images of plant disease and CNN (Convolutional Neural Networks) is used as machine learning algorithm which is a classification algorithm.

The results show that the proposed system is able to identify a wide variety of plant diseases with high accuracy. It offers a promising solution to enhance agricultural sustainability by detecting diseases at an early stage, reducing chemical pesticide usage, and improving crop yield.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

Introduction

1.1Problem Statement:

The primary problem addressed in this project is the lack of awareness among farmers about crop diseases, which often results in delayed detection and ineffective management. When plant diseases are not identified early, they can spread rapidly across crops, leading to severe yield losses. This not only reduces the quantity and quality of the harvest but also negatively impacts overall agricultural production. As a result, farmers face significant economic losses, which affect their livelihoods and contribute to broader issues of food security. Furthermore, the lack of timely intervention puts strain on farming sustainability and hinders long-term agricultural development.

1.2Motivation:

This project was chosen due to the increasing challenges faced by farmers in detecting and managing plant diseases, which significantly affect agricultural productivity. The motivation stems from the need to create a sustainable solution that empowers farmers with the tools to detect crop diseases early and efficiently, ultimately improving crop yield and quality. As agricultural practices continue to face the pressures of climate change, growing populations, and limited resources, the need for innovative solutions in plant disease management has never been more critical.

The potential applications of this project are vast, particularly in the realm of precision agriculture. By using advanced image processing and machine learning techniques such as CNN (convolutional neural network), the system can provide farmers with a tool that helps in diagnosing plant diseases . This will not only help in identifying a wide range of diseases but also enable early interventions, reducing the overuse of pesticides and minimizing environmental damage and also help economically. If the farmer know about the disease in early stage then the production of yield can be increased.

The impact of this project is significant in both economic and environmental terms. For farmers, it offers the potential to increase crop yields, reduce production costs, and improve their overall economic stability. It contributes to sustainable farming practices by reducing pesticide use, promoting healthier ecosystems, and enhancing food security.

1.3Objective:

The objective of the Plant Disease Detection System for Sustainable Agriculture includes classification of the disease and the Recommendation system which give suggestion to the farmers from which they can take action in early stages and increase the productivity of the yield.

To develop the Plant Disease Detection System for Sustainable Agriculture system capable of accurately classifying various plant diseases based on images of plant leaves, using image processing and machine learning techniques. To enable early detection of plant diseases at their initial stages, allowing farmers to take timely action and prevent the spread of diseases across the crop.

To implement a suggestion-making feature that provides farmers with recommended actions for managing identified plant diseases, including potential treatments and preventive measures.

To ensure that the system is user-friendly and accessible, allowing farmers to easily upload images using devices for quick disease diagnosis and suggestions.It support sustainable agricultural practices by reducing the overuse of pesticides and encouraging healthier, eco-friendly crop management strategies.

1.4Scope of the Project:

The project Disease Detection System for Sustainable Agriculture system includes the development and deployment of an advanced technological solution designed to detect plant diseases early, prevent their spread from which the yield can be produced more. By focusing on early detection and real-time intervention, the system enables farmers to manage crop health more effectively, reducing the likelihood of widespread disease outbreaks and minimizing crop loss.

The scope of the project includes some key aspect such as :

1.4.1 Early detection to prevent disease spread - The system's ability to identify diseases in their early stages ensures that farmers can take immediate action to prevent further spread, thereby minimizing crop damage and potential yield loss.

1.4.2 Improved crop yield - By reducing the impact of diseases on crops, farmers can maintain more production levels, increased yield.

1.4.3 Reduction of wasting resources - The system promotes efficient use of resources such as water, fertilizers, and pesticides by providing data-driven insights into crop health.

CHAPTER 2

Literature Survey

2.1 Review relevant literature or previous work in this domain.

2.1.1 Disease detection -

Disease attacks are constant threats to agriculture and cause heavy losses in the country's economy. Therefore, early detection can mitigate the severity of diseases and protect crops. However, manual disease identification is both time-consuming and error prone, and requires a thorough knowledge of plant pathogens. Instead, automated methods save both time and effort.

optical imaging techniques are among the indirect methods that are able to identify diseases and predict the health of the crop through different parameters such as morphological change and transpiration rate. Fluorescence and hyperspectral imaging are some of the most widely used indirect methods for early disease identification. Although hyperspectral images are a valuable source of data and contain more information than ordinary photos, hyperspectral devices are very expensive, bulky, and difficult to obtain for low-income farmers.

Technologies such as machine learning, IoT, and remote sensing have been increasingly applied to disease detection systems in agriculture to enhance efficiency and sustainability.

2.1.1.1 Technology that are used – Machine learning , IOT, image processing, sensor networking.

2.1.1.2 Background - This section provides a review of different techniques applied in the identification of crop diseases, presents the taxonomy of various crop diseases, and describes the concept of image processing and machine learning. It also demonstrates the application of hyperspectral imagery, the Internet of Things, and deep and transfer learning in the field of disease recognition.

Taxonomy of Crop Diseases and Their Symptoms - The leaves of crops are highly prone to diseases, which are a natural phenomenon . However, if corrective measures are not taken at the right time to stop the spread of the disease. Crops are affected by various pathogens such as viruses, bacteria, fungi, and deficiencies. The symptoms of the disease adversely affect the development and growth of crops and are easily visible. There are some type of plant disease :- virus disease, Fungal disease , Bacterial disease.

2.1.1.3 Application of Machine Learning and Image Processing in Disease Identification-

Foliar images are an excellent and rich source of data on plant pathology and morphological behavior; thus, these data must be thoroughly extracted and analyzed. Image processing plays a crucial role in the diagnosis and analysis of leaf diseases.

The primary step in identifying diseases is the acquisition of images. In most cases, images can be fetched either from a digital camera or an imaging system. As raw images tend to contain noise, removing these impurities is required. the second step is known as image pre-processing, and involves the removal of unwanted distortions, in addition to contrast enhancement, to clarify and brighten the image features. the third step in which the image is segmented from its background, whereas the region of interest (ROI) is partitioned to emphasize the prominent features. The fourth step is feature extraction ,which unveils the information and details of an image. The leaf features usually include shape, texture, and color, which are used to diagnose the crop. Thus, these chosen features form an input feature vector which is then fed into the classifier. he final step is classification.

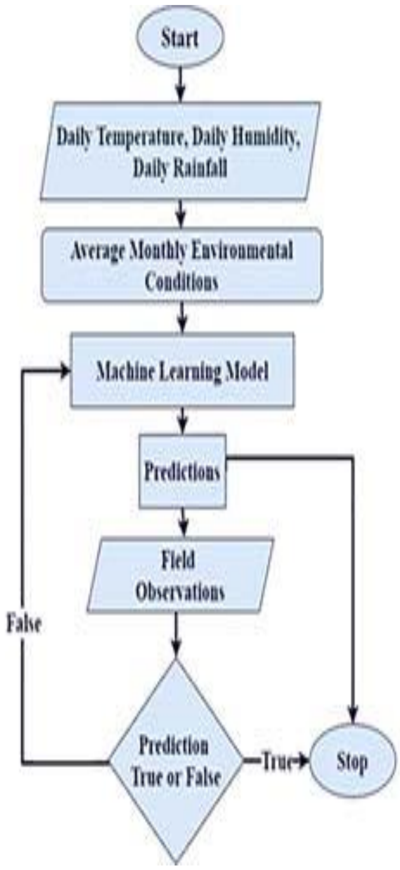
References - https://www.mdpi.com/2073-4395/12/10/2395

Ahsan, Md Manjurul, Shahana Akter Luna, and Zahed Siddique. "Machine-learning-based disease diagnosis: A comprehensive review." *Healthcare*. Vol. 10. No. 3. MDPI, 2022.

Ahsan, M. M., Luna, S. A., & Siddique, Z. (2022, March). Machine-learning-based disease diagnosis: A comprehensive review. In *Healthcare* (Vol. 10, No. 3, p. 541). MDPI.

Ahsan, Md Manjurul, Shahana Akter Luna, and Zahed Siddique. "Machine-learning-based disease diagnosis: A comprehensive review." In *Healthcare*, vol. 10, no. 3, p. 541. MDPI, 2022.

2.1.2 IOT and Machine learning algorithm in plant disease detection – This aims to propose a Machine Learning (ML) approach for the early prediction of the probability of disease attack based on Internet of Things (IoT) directly sensed crop field environmental conditions. The crop field environmental conditions are used to predict the occurrence of plant diseases. The Multiple Linear Regression (MLR) is applied as the ML model due to the existence of a linear relationship between disease attack and environmental conditions. Internet of Things (IoT) based crop field environmental conditions help to accurately predict the occurrence of plant diseases using the ML approach.



References - https://ieeexplore.ieee.org/abstract/document/9761267/,

Liu, Zhiyan, et al. "Internet of Things (IoT) and machine learning model of plant disease prediction–blister blight for tea plant." *Ieee Access* 10 (2022): 44934-44944.

Liu, Z., Bashir, R. N., Iqbal, S., Shahid, M. M. A., Tausif, M., & Umer, Q. (2022). Internet of Things (IoT) and machine learning model of plant disease prediction–blister blight for tea plant. *Ieee Access*, *10*, 44934-44944.

Liu, Zhiyan, Rab Nawaz Bashir, Salman Iqbal, Malik Muhammad Ali Shahid, Muhammad Tausif, and Qasim Umer. "Internet of Things (IoT) and machine learning model of plant disease prediction–blister blight for tea plant." *Ieee Access* 10 (2022): 44934-44944.

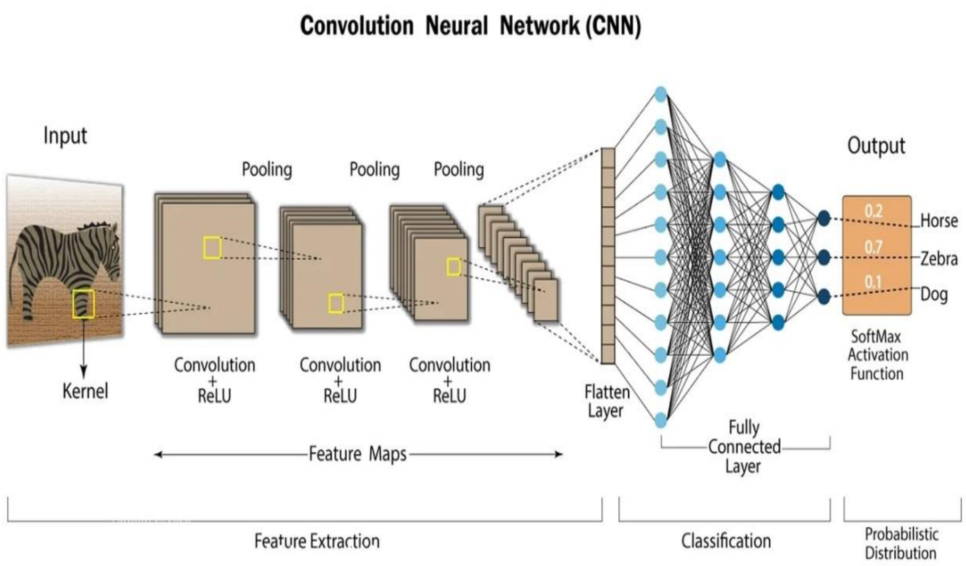
2.2 Mention any existing models, techniques, or methodologies related to the problem.

2.2.1 CNN (Convolution neural network) - Deep Convolutional Neural Network is utilized in this study to identify infected and healthy leaves, as well as to detect illness in afflicted plants. The CNN model is designed to suit both healthy and sick leaves; photos are used to train the model, and the output is determined by the input leaf. A Convolutional Neural Network has three layers: a convolutional layer, a pooling layer, and a fully connected layer.

2.2.1.1 Convolutional layer : produces an activation map by scanning the pictures several pixels at a time using a filter.

2.2.1.2 Pooling layer : reduces the amount of data created by the convolutional layer so that it is stored more efficiently.

2.2.1.3 Fully connected input layer – The preceding layers' output is "flattened" and turned into a single vector which is used as an input for the next stage. The first fully connected layer – adds weights to the inputs from the feature analysis to anticipate the proper label. Fully connected output layer – offers the probability for each label in the end.



References - Shelar, Nishant, et al. "Plant disease detection using CNN." *ITM Web of Conferences*. Vol. 44. EDP Sciences, 2022.

Shelar, N., Shinde, S., Sawant, S., Dhumal, S., & Fakir, K. (2022). Plant disease detection using CNN. In *ITM Web of Conferences* (Vol. 44, p. 03049). EDP Sciences.

Shelar, Nishant, Suraj Shinde, Shubham Sawant, Shreyash Dhumal, and Kausar Fakir. "Plant disease detection using CNN." In *ITM Web of Conferences*, vol. 44, p. 03049. EDP Sciences, 2022.

Shelar N, Shinde S, Sawant S, Dhumal S, Fakir K. Plant disease detection using CNN. InITM Web of Conferences 2022 (Vol. 44, p. 03049). EDP Sciences.

2.2.2 SVM (Support vector machine) – The SVM (HRF-MCSVM) design for plant foliar disease detection. To improve the computation accuracy, the image features are preprocessed and segmented using Spatial Fuzzy C-Means prior to the classification process. Finally, the performance metrics like accuracy, F-measure, specificity, sensitivity, and recall value were evaluated to determine the effectiveness of the system. The proposed HRF-MCSVM method is compared with a few existing techniques to determine its efficiency.

The diseased plants display their symptoms mainly in the plant’s growth and leaf such as irregular leaf shapes, and reduced growth color variations. Nowadays, machine learning and deep learning techniques are used for data processing resources for detecting and diagnosing plant disease efficiently . The random forest Multiclass SVM (HRF-MCSVM) model for plant foliar disease detection include some objectives -

an HRF-MCSVM that accurately classifies the diseases in crops and rapidly improves the quality and production rate. The HRF-MCSVM model is optimized using the TDO algorithm.

The TDO algorithm optimizes the width of the Gaussian kernel and the boundary parameters to enhance the True Positive Rate of the HRF-MCSVM classifier.

The Random Forest (RF) and Multi-Class Support Vector Machine (MCSVM) classifiers are hybridized to offer higher classification accuracy for the leaf disease classes with convenience and speed. The MCSVM classifier minimizes the error rate of the RF classifier by normalizing the data vectors and maximizing the hyperplane margin.

The image features are preprocessed and segmented using the Spatial Fuzzy C-Means approach to increase computation accuracy and provide accurate and effective data analysis.

References - Sahu, Santosh Kumar, and Manish Pandey. "An optimal hybrid multiclass SVM for plant leaf disease detection using spatial Fuzzy C-Means model." *Expert systems with applications* 214 (2023): 118989.

Sahu, S. K., & Pandey, M. (2023). An optimal hybrid multiclass SVM for plant leaf disease detection using spatial Fuzzy C-Means model. *Expert systems with applications*, *214*, 118989.

Sahu, Santosh Kumar, and Manish Pandey. "An optimal hybrid multiclass SVM for plant leaf disease detection using spatial Fuzzy C-Means model." *Expert systems with applications* 214 (2023): 118989.

Sahu SK, Pandey M. An optimal hybrid multiclass SVM for plant leaf disease detection using spatial Fuzzy C-Means model. Expert systems with applications. 2023 Mar 15;214:118989.

2.2.3 Deep Learning - Deep learning is powerful machine learning approach which have mitigated the traditional machine learning headache of feature engineering. The core of deep learning is artificial neural network (ANN). Artificial neural networks are mathematical models that replicate with their neurons and synapses interconnecting them the general principles of brain function [9].To implement neural network one of the most standard library is Tensorflow. It provides all libraries related to artificial neural network. With the help of Tensorflow one can perform classification tasks on text as well as images.

The model is able to detect several diseases from plants using pictures of their leaves. [Methodology] Plant disease detection model is developed using neural network. First of all augmentation is applied on dataset to increase the sample size. Later Convolution Neural Network (CNN) is used with multiple convolution and pooling layers. PlantVillage dataset is used to train the model. After training the model, it is tested properly to validate the results. [Results] We have performed different experiments using this model. 15% of data from PlantVillage data is used for testing purpose that contains images of healthy as well as diseased plants. Proposed model has achieved 98.3% testing accuracy.

The advantage of deep learning over machine learning is that one does not need to worry about domain expertise as no feature engineering is required in this, unlike traditional machine learning approaches .

References -Chohan, Murk, et al. "Plant disease detection using deep learning." *International Journal of Recent Technology and Engineering* 9.1 (2020): 909-914.

Chohan, M., Khan, A., Chohan, R., Katpar, S. H., & Mahar, M. S. (2020). Plant disease detection using deep learning. *International Journal of Recent Technology and Engineering*, *9*(1), 909-914.

Chohan, Murk, Adil Khan, Rozina Chohan, Saif Hassan Katpar, and Muhammad Saleem Mahar. "Plant disease detection using deep learning." *International Journal of Recent Technology and Engineering* 9, no. 1 (2020): 909-914.

2.3 Highlight the gaps or limitations in existing solutions and how your project will address them.

2.3.1 Limited Generalization Across Plant Species : Many existing models, particularly deep learning models like CNNs, are often trained on datasets that focus on specific plant species. As a result, these models may not generalize well to other plant species or crops, especially when there is limited data for training.

Project Contribution - This project will focus on developing a more generalized model that can effectively detect diseases across multiple plant species and ensure its usability in diverse agricultural environments.

2.3.2 Limited Detection of Early Disease Symptoms : Many current systems rely heavily on visual symptoms such as spots, discoloration, or wilting, which are often visible only after the disease has progressed. This limits the ability to detect diseases in their early, more manageable stages.

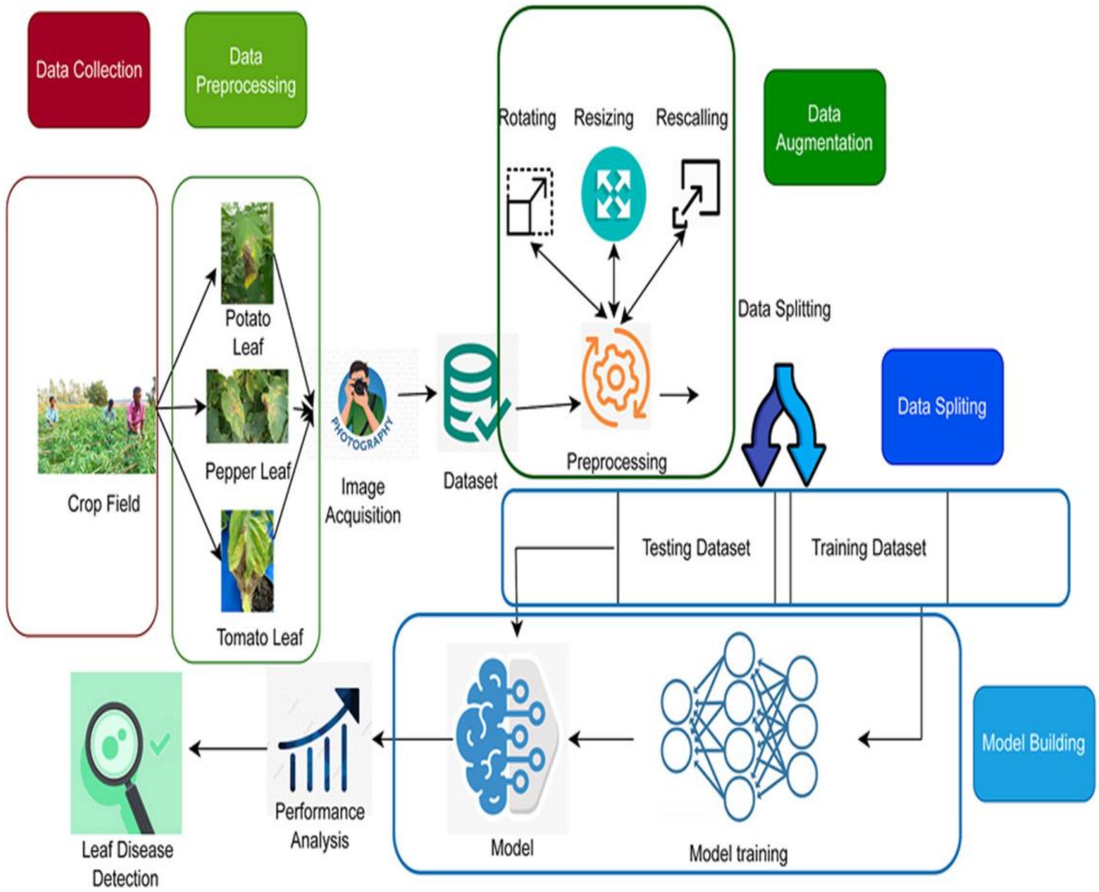
Project Contribution: We aim to develop a solution that incorporates multispectral or hyperspectral imaging techniques, which can detect subtle physiological changes in plants even before visible symptoms appear. By integrating these advanced sensor-based methods with machine learning, our project will focus on improving early disease detection, enabling timely intervention.

2.3.3 **Scalability :** Many plant disease detection systems, particularly those based on image processing or machine learning, can be computationally intensive and may not be scalable or practical for real-time use on large farms.

Project contribution : We plan to design a lightweight, cloud-based system that can process disease detection data in real time while requiring minimal computational resources.

CHAPTER 3
Proposed Methodology

3.1 System Design



3.1.1 Data Collection - This stage involves collecting data from various sources. The proposal starts with collecting the input images representing different types of leaves like potatoes, tomatoes, and peppers. These raw images can be collected using a real-time camera or mobile. For our deployment, the deep learning model was trained using a publicly accessible dataset during the framework's testing and training phases.

3.1.2 Data Preprocessing - The raw images collected from the dataset might contain noises and it is essential to preprocess them before fitting them into the learning module. We apply rotation, resizing, and shearing to preprocess the image during the preprocessing phase. Relevant features such as color patterns, leaf texture, and disease spots are extracted from the images, or patterns from the sensor data are analyzed. This step is crucial as it reduces the complexity of the input data, focusing on what matters most for disease detection.

3.1.3 Training the Model - The training process allows the model to learn from labeled data, where the system can understand which features (like leaf patterns or environmental factors) correspond to specific plant diseases. This step has two main phases. The TL models are trained using a training image dataset during the first phase. During the later phase, the architecture is validated using test images reserved for performance evaluation.

3.1.4 Model construction - To build the predictive model, we apply the following steps:

- Collecting images from the dataset.
- Pre-process image data by resizing and rotating images.
- Creating convolute feature connect into Fully Connected Layers. Usually, it is flattened, converted to a one-dimensional (1D) array (or vector), and then joined to one or more completely connected layers.
- Finally, extract the features for different classes of the input.

3.1.5 Model evaluation - From an ideal dataset, 80% of photos are taken for training and 20% for testing. Then Validation data is used to check accuracy by applying the predict function and accurately extracting features. Images are taken for confirming detection once validation provides good results. Finally, characteristics are retrieved to determine whether or not the leaves are infected.

3.1.6 Performance evaluation - In this phase, we obtain the best model based on the performance of the extensive experiments. We used accuracy, precision, recall, f1score, training accuracy, training loss, validation accuracy, and validation loss. This will help to build the smart web application with deep learning guidance.

3.1.7 Result - The machine learning model developed for plant disease detection achieved promising results in terms of accuracy and performance. After training on a dataset containing over many labeled images of various plant species and diseases.

3.1.7.1 Training score – After trained the model the model give training accuracy of 0.9904971718788147 .

3.1.7.2 Validation score – The model give the testing accuracy of 0.9655702114105225.

3.1.7.3 Accuracy – 97%

3.1.7.4 Precision – 92%(macro avg), 97%(weighted avg)

3.1.7.5 Recall – 93% (macro avg) , 97% (weighted avg)

3.1.7.6 F1- score- 94 % (macro avg) , 97% (weighted avg)

The plant disease detection application successfully achieved a high level of performance with an accuracy of 92% on the test dataset, significantly reducing the identification time for common diseases in crops like tomatoes, potato, corn etc.

3.2 Requirement Specification

For implement plant disease detection system for agriculture sustainable the hardware or software resources are essentials.

3.2.1 Hardware Requirements:

The hardware requirements specify the physical devices and configurations needed to support the development and deployment of the application. These include the computing devices used for model training, testing, and deployment on user devices.

3.2.1.1 Processor – For implement a machine learning model the processor should be more efficient and give the better performance.

3.2.1.2 RAM - A minimum of 16 GB of RAM is required to handle large datasets, machine learning models, and data processing tasks without performance degradation.

3.2.1.3 Storage - At least 500 GB of hard disk drive (HDD) space is required, though a solid-state drive (SSD) is preferred for faster data access and retrieval.

3.2.1.4 Internet Connection - A stable internet connection is needed for downloading large datasets, frameworks, libraries, and updates.

3.2.1.5 Graphical processing unit (GPU) - For training deep learning models, the GPU should be efficient.

3.2.2 Software Requirements:

The software requirements encompass the programming languages, frameworks, libraries, IDLE, and development tools necessary for creating the Plant Disease Detection Application.

3.2.2.1 Operating System - I used Windows 10 as my primary operating system during development. which offered more flexibility for running certain machine learning libraries. I deployed the application on my operating system making sure that the application is working fine when the user enter the image and give the correct result.

3.2.2.2 Programming Language – It is the core component of the application. From the help of programming language I able to created efficient application.

Python - I chose Python as the main programming language because it is well-suited for machine learning and data science. It has extensive libraries for handling image processing and deep learning, which are integral to this project.

Libraries – Python Libraries are the collection of pre written code that provide reusable function , classes. They can help programmers to Speed up development. There are some libraries which are -

- **Pandas -** It is essential for data manipulation, particularly for handling large datasets of plant images. Pandas is a powerful Python library for data manipulation and analysis. It was primarily used for handling and processing the large datasets of plant images and labels that I used during model training. Pandas provided a user-friendly interface for cleaning, transforming, and analyzing the data.

  I used DataFrames to store, manipulate, and organize datasets efficiently. I used Pandas for tasks such as data preprocessing and cleaning like handling missing data, removing outliers, and normalizing image metadata or associated information (like plant disease labels). Pandas simplified the data preprocessing pipeline, which was a crucial step before feeding the data into the machine learning model.

- **NumPy -** NumPy is a library that provides support for numerical operations, including large, multi-dimensional arrays and matrices. I used NumPy for efficient numerical operations during the data preparation and training processes, especially when dealing with pixel data from images and performing operations like reshaping and resizing. NumPy's functions allowed me to perform essential matrix operations required for processing image data, which is crucial for image-based machine learning tasks.
- **Scikit- learn -** Scikit-learn is a Python library used for traditional machine learning tasks. While deep learning was the primary approach for plant disease detection, I used scikit-learn for comparison purposes, to experiment with traditional machine learning algorithms like Decision Trees, Random Forest, and Support Vector Machines (SVM). I used scikit-learn to test different algorithms and compare their performance against the deep learning model.
- **Matplotlib and Seaborn** – These techniques are used for data visualization with the help of some charts, blocks etc.
  Matplotlib is a library in python that enable user to generate visualization like histogram, scatter plot, bar charts, pie charts and much more. While Seaborn is a library that is built on top of matplotlib. It provide data visualization that are typically more aesthetic and statistically sophisticated.

3.2.2.3 Framework - A framework is a set of pre-written code that provides a foundation to help develop applications, saving time by offering a structured way to handle common tasks. Frameworks often define a specific structure and coding style, ensuring that developers can focus on the core features of their application without reinventing the wheel for routine tasks. They abstract away repetitive code and allow you to leverage pre-built solutions for common problems like image processing, API creation, or machine learning model deployment. It include :

OpenCV - A highly efficient library used for real-time computer vision tasks. OpenCV provides functionalities to process and analyze images and video streams. In plant disease detection, OpenCV can be used to handle image pre-processing, such as resizing, color correction, or feature extraction before passing images into your deep learning models.

Flask: A micro web framework for Python, Flask is lightweight and easy to use, making it ideal for developing simple web applications and APIs. If your application includes a web-based interface for users to upload images and receive disease predictions, Flask can serve as the backend framework. It allows the creation of RESTful APIs to connect your machine learning model with the frontend interface.

TensorFlow and **Keras -** One of the core frameworks used for this project was TensorFlow, an open-source deep learning framework that provides tools to design, train, and deploy machine learning models. I utilized TensorFlow because it supports a variety of neural network architectures, making it ideal for tasks like image classification, which is essential in detecting plant diseases based on leaf images. TensorFlow is highly optimized for performance and can run on multiple platforms, including mobile devices.

TensorFlow's ability to train complex deep learning models was essential in classifying various plant diseases accurately.

Its comprehensive ecosystem also allowed for easy integration with other tools and frameworks, making it suitable for end-to-end application development.

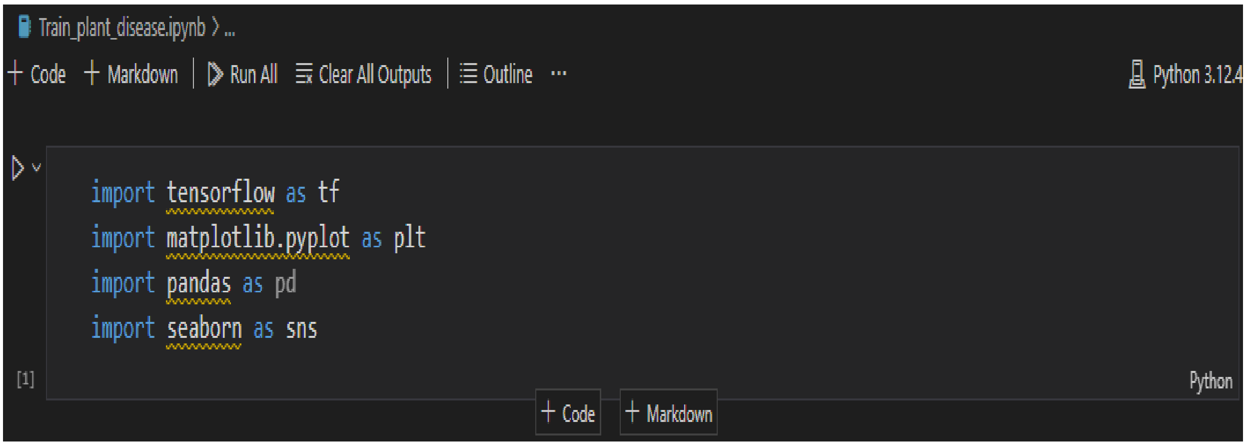
3.2.2.4 Integrated Development Environments (IDEs) - An Integrated Development Environment (IDE) **i**s a software application that provides a comprehensive set of tools to help developers write, test, and debug code. It typically includes a code editor, a debugger, build automation tools, and a compiler or interpreter. IDEs are designed to simplify the development process by providing a cohesive platform to manage code and related tasks.

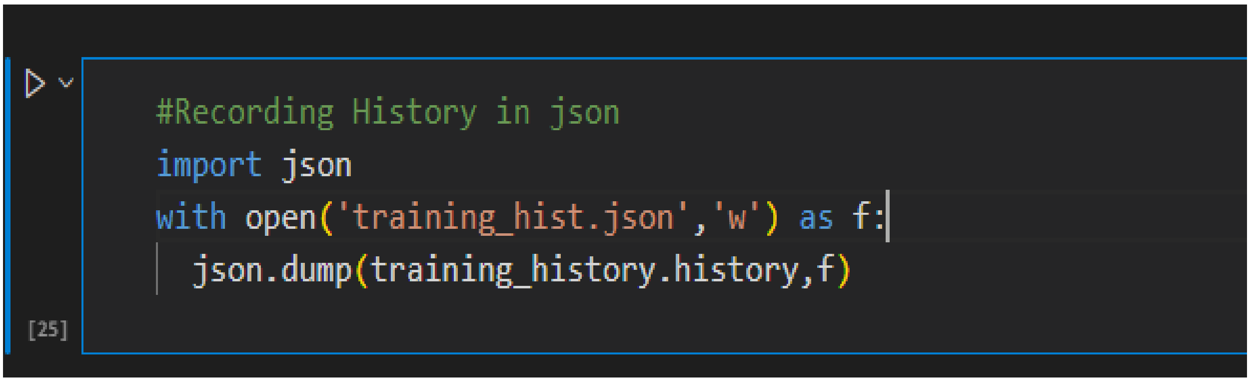
Visual Studio Code (VSCode) - A lightweight and highly customizable code editor that supports Python development through extensions. VSCode is known for its fast startup and integrated terminal, making it ideal for quick development cycles.

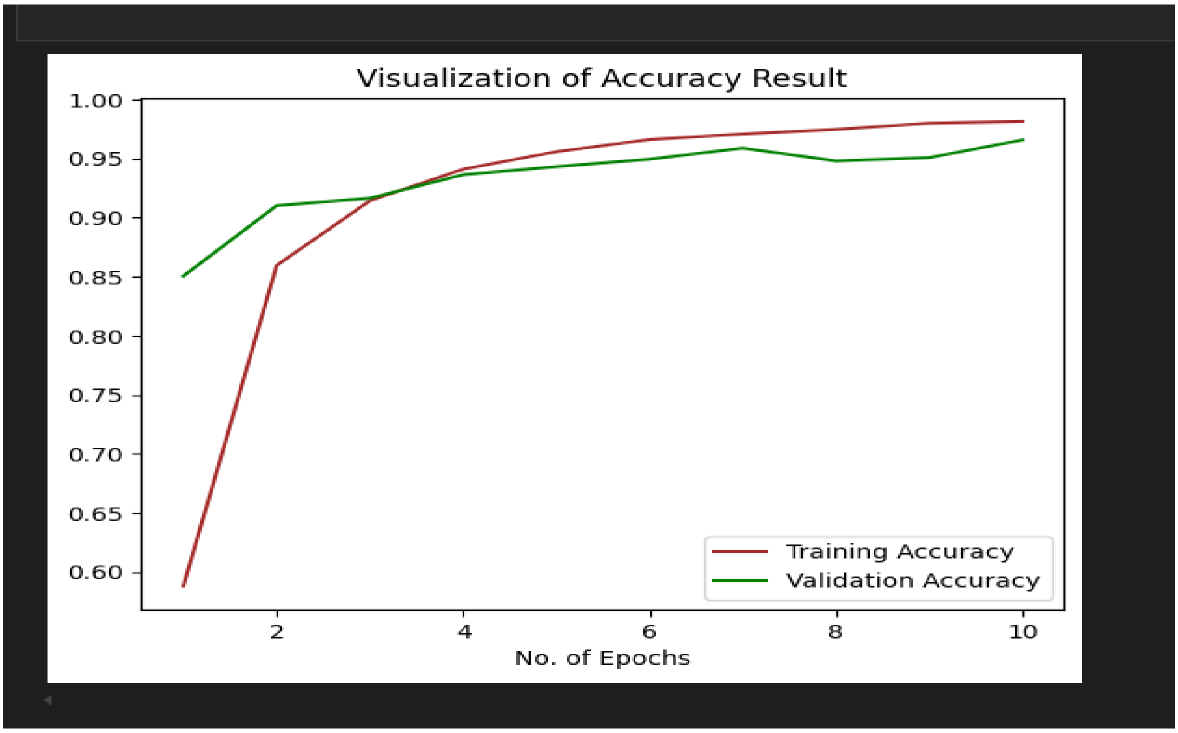
Jupyter Notebooks: While not a traditional IDE, Jupyter is incredibly popular in data science and machine learning. It provides an interactive environment to write code, visualize data, and test machine learning models in real-time. For tasks like training models on plant images, Jupyter can be a very effective tool.

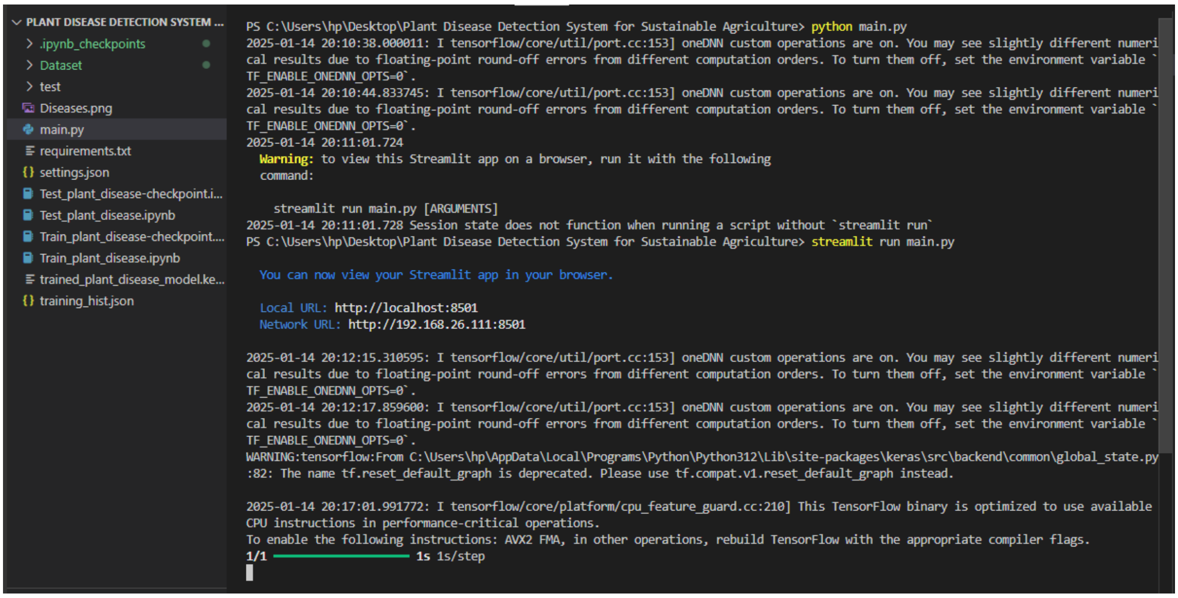
3.2.2.5 DataBase - I used JSON for backend work.

JSON (JavaScript object notation) - JSON is a lightweight, text-based format used to represent structured data in a human-readable way. It is widely used for data exchange between systems and is a standard for APIs, configuration files, and data storage due to its simplicity and ease of use. JSON has a simple syntax that is easy to read and write. It requires less bandwidth compared to other data formats such as XML. JSON represents data as key-value pairs, arrays, or nested objects. This makes it highly flexible and adaptable for storing structured data in various types of applications.






Visualization of Accuracy Result
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
2
4
6
8
10
No. of Epochs
Training Accuracy
Validation Accuracy


PLANT DISEASE DETECTION SYSTEM ...
.ipynb_checkpoints
Dataset
test
Diseases.png
main.py
requirements.txt
settings.json
Test_plant_disease-checkpoint.i...
Test_plant_disease.ipynb
Train_plant_disease-checkpoint....
Train_plant_disease.ipynb
trained_plant_disease_model.ke...
training_hist.json
PS C:\Users\hp\Desktop\Plant Disease Detection System for Sustainable Agriculture> python main.py
2025-01-14 20:10:38.000011: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numeri
cal results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `
TF_ENABLE_ONEDNN_OPTS=0`.
2025-01-14 20:10:44.833745: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numeri
cal results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `
TF_ENABLE_ONEDNN_OPTS=0`.
2025-01-14 20:11:01.724
Warning: to view this Streamlit app on a browser, run it with the following
command:
streamlit run main.py [ARGUMENTS]
2025-01-14 20:11:01.728 Session state does not function when running a script without `streamlit run`
PS C:\Users\hp\Desktop\Plant Disease Detection System for Sustainable Agriculture> streamlit run main.py
You can now view your Streamlit app in your browser.
Local URL: http://localhost:8501
Network URL: http://192.168.26.111:8501
2025-01-14 20:12:15.310595: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numeri
cal results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `
TF_ENABLE_ONEDNN_OPTS=0`.
2025-01-14 20:12:17.859600: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numeri
cal results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `
TF_ENABLE_ONEDNN_OPTS=0`.
WARNING:tensorflow:From C:\Users\hp\AppData\Local\Programs\Python\Python312\Lib\site-packages\keras\src\backend\common\global_state.py
:82: The name tf.reset_default_graph is deprecated. Please use tf.compat.v1.reset_default_graph instead.
2025-01-14 20:17:01.991772: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available
CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
1/1 1s 1s/step

CHAPTER 4

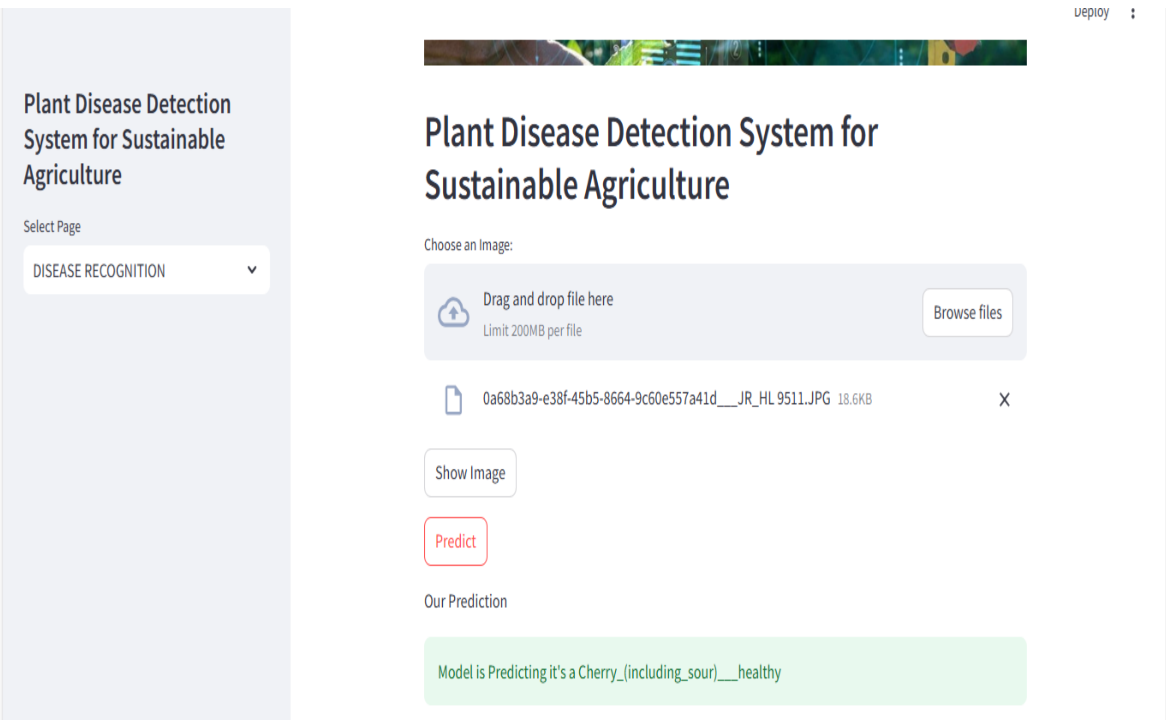
Implementationand Result

4.1Snap Shots of Result:

These snapshots illustrates the web interface of the application as it appears when the user first opens it. It shows the home page, which serves as the main entry point to the system. From here, users can upload images, view results, and interact with various features of the plant disease detection application.

These snapshots shows the image uploaded by the user, but no result has been generated yet. The system is awaiting processing of the provided image to detect the plant disease. At this stage, the image is displayed on the interface, and the user will see the prediction results once the system finishes processing the image

This snapshot shows the output of the plant disease detection system when a user uploads an image of a plant leaf for analysis. It shows the result after the system has processed the uploaded image. The model has successfully identified the plant as a Cherry (including sour), and the prediction indicates that the plant is healthy. The system confidently classifies the plant as disease-free, which suggests no signs of infection or damage. The confidence level for this prediction is high, providing assurance to the user that the plant is in good health. This result reflects the model's ability to accurately differentiate between healthy and diseased plants based on image analysis.



4.2GitHub Link for Code:

https://github.com/Shalu-choudhary/plant_disease_detection

CHAPTER 5

Discussion and Conclusion

5.1 Future Work:

future work, the dataset should be expanded by incorporating a wider variety of plant species and diseases. More images from diverse geographical regions and varying environmental conditions would help the model generalize better. Additionally, acquiring a balanced dataset (equal representation of healthy and diseased samples) can address issues of bias. Future improvements could involve advanced image augmentation techniques like rotation, flipping, or color variation to create more diverse training data. Implementing better normalization techniques could also help in achieving better performance. Consider using more advanced Convolutional Neural Networks (CNNs) or Transformer-based models ,which have shown great potential in image classification tasks like recommendation of crop according to the land , grow the crop according to the weather etc.

5.2 Conclusion:

The Plant Disease Detection Application for Agriculture makes a significant contribution to the field of agricultural technology by providing an automated, accessible, and efficient tool for identifying and diagnosing plant diseases. The project leverages machine learning techniques, particularly Convolutional Neural Networks (CNNs)**,** to analyze plant images and accurately detect diseases, empowering farmers and agricultural professionals to take timely actions.

The application allows users to quickly assess the health of their plants by uploading images for analysis. By detecting diseases early, the system helps prevent the spread of infections, ensuring healthier crops and better yields.

By accurately identifying plant diseases, the system reduces the reliance on pesticides and other chemical treatments. This supports more sustainable farming practices, reducing chemical use and its environmental impact.