**Analysis and identification of crop**

**diseases by machine learning**

*A*

***ProjectReport***

*submitted*

*in partial fulfilment*

*for the award of the Degree of*

***Bachelor of Technology***

***in Department of*** ***Computer Science and Engineering***

****

Supervisor Submitted By:

Mr. Deepak Gupta  Pratham ubana ( 20EEACS076)

(Assistant Professor) Yashika Saini (20EEACS107)

Signature: Kajol khatwani (20EEACS045)

Sakshi khandelwal (20EEACS090)

**Department of Computer Science & Engineering**

Engineering College Ajmer

**June , 2024**

***Candidate’s Declaration***

I hereby declare that the work, which is being presented in the Project, entitled **“Analysis and identification of crop diseases by machine learning”** in partial fulfilment for the award of Degree of “*Bachelor* of Technology” in Department of **Computer Science and Engineering** **,** Engineering College Ajmer, Bikaner Technical University is a record of my own investigations carried under the Guidance of **Mr. Deepak Gupta**, Department of Computer Science & Engineering**,** Engineering College Ajmer. I have not submitted the matter presented in this report anywhere for the award of any other Degree.

**Pratham ubana**

(20EEACS076)

**Yashika Saini**

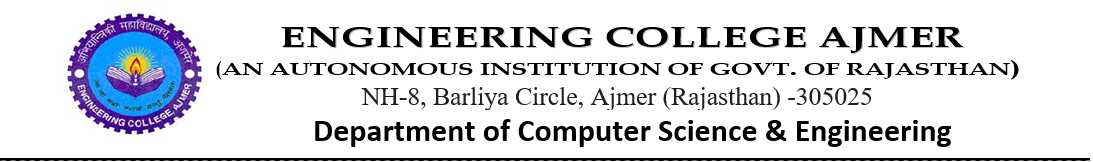
(20EEACS107)

**Kajol khatwani**

(20EEACS045)

**Sakshi Khandelwal**

(20EEACS090)



***CERTIFICATE***

This is to certify that ***Pratham ubana, Yashika Saini, Kajol khatwani, Sakshi khandelwal*** of VIII Semester, B.Tech (CSE) 2023-24, has submitted the Project titled “**Analysis and identification of crop diseases by machine learning**” in partial fulfilment for the award of the degree of Bachelor of Technology under Bikaner Technical University, Kota.

Date: 15-05-2024

Mr. Deepak Gupta

(Supervisor)

**ACKNOWLEDGMENET**

I take this opportunity to express my gratitude to all those people who have been directly and indirectly with me during the competition of this Project.

I pay thank to **Mr. Deepak Gupta** who has given guidance and a light to me during this major project. His versatile knowledge about **“Analysis and identification of crop diseases by machine learning”** has eased me in the critical times during the span of this Project.

I acknowledge here out debt to those who contributed significantly to one or more steps. I take full responsibility for any remaining sins of omission and commission.

B.Tech VIII Year

(CSE)

**ABSTRACT**

Crop diseases are a significant threat to global food security, causing substantial yield losses and economic hardships for farmers. Timely detection and accurate identification of these diseases are crucial for implementing effective management strategies and minimizing agricultural losses This project aims to leverage ML algorithms for the analysis and identification of crop diseases, with the goal of providing farmers and agricultural stakeholders with a reliable tool for early disease diagnosis and management.

The project begins with an extensive review of existing literature on crop disease detection methodologies and the application of ML in agriculture. Building upon this foundation, the methodology section outlines the data collection process, preprocessing techniques, and feature extraction and selection methods utilized in the study.

A diverse dataset comprising images of diseased crops is curated, ensuring data quality and representativeness across different crop types and disease severities. Various ML algorithms, but we mainly focus on Convolutional Neural Networks (CNN), are implemented and evaluated using a range of performance metrics such as accuracy, precision, recall, and F1-score. Rigorous cross-validation techniques are employed to assess model robustness and generalizability.

The results demonstrate the efficacy of ML algorithms in accurately identifying crop diseases based on visual symptoms extracted from images. Despite promising results, limitations such as dataset size and computational resources are acknowledged, suggesting avenues for future research and development.

In conclusion, this project underscores the transformative potential of ML techniques in revolutionizing crop disease management practices. By providing farmers with automated tools for disease diagnosis and decision-making, the proposed approach has the potential to mitigate yield losses, optimize resource utilization, and contribute to the sustainability of agricultural production systems. Further research is warranted to address identified limitations and refine ML models for real-world deployment in agriculture.

CONTENTS

Certificate ..............................................................................................................i

Acknowledgement............................................................................................... ii

Abstract............................................................................................................... iii

List of Figures .......................................................................................................

List of Tables ........................................................................................................

Chapter 1: Introduction ........................................................................................1

* 1. Methodology…......................................................................................2
  2. Purpose ….……………………………………………………………5
  3. Scope …………………………………………………………………6
  4. Tools used ….………………………………………………………...7
  5. Overview …….……………………………………………………….8

Chapter 2: Literature Review ..........................................................................10

Chapter 3: System Design ..................................................................................12

3.1 ER-Diagram ………………………………………………………..12

3.2 Architecture Design ………………………………………………..13

3.3 Activity Diagram …………………………………………………..14

Chapter 4: Implementation ……………………………………………………15

4.1 Data collection and prepocessing …..…………………………….15

4.2 Feature extraction ….……………………………………………...16

4.3 Model training ….…………………………………………………18

4.4 Model evaluation ….………………………………………………..19

4.5 Software interface …..……………………………………………..20

Chapter 5: Conclusion ……………………………………………………...…24

Chapter 6 : Future Enhancements ……………………………………………..27

References .........................................................................................................30

LIST OF FIGURES

Fig. 1.1 images used in different 38 sets …............................................................................5

Fig 3.1 ER- diagram …………………………………..……………………………………..12

Fig 3.2 Architecture design……………………………………………………………….….13

Fig 3.3 Activity diagram……………………………………………………………………..14

Fig. 4.2. The architecture of a typical [CNN](https://www.sciencedirect.com/topics/computer-science/convolutional-neural-network) …………………………………………………16

Fig 4.3 Convolutional layer …………………………………………………………………16

Fig 4.4 Pooling layer ……..…………………………………………………………………17

Fig. 4.5 Fully connected layer ………………………………………………………………17

Figures in chapter 4 and 5 are mostly the screen shorts of program used in model training

LIST OF TABLES

Table 1.1 Information of the dataset…......................................................................................3

Table 4.1 Prams of the CNN layers….....................................................................................18

Table 5.1 Performance of model ……....................................................................................24

**Chapter 1**

**INTRODUCTION**

Crop diseases pose a significant threat to global food security, agricultural sustainability, and socio-economic stability. As the world's population continues to grow, the demand for food increases, placing additional pressure on agricultural systems to produce higher yields. However, the presence of crop diseases undermines these efforts by causing substantial yield losses, estimated to range from 20% to 40% globally each year. These losses not only impact farmers' incomes but also have broader implications for food availability, prices, and access, particularly in vulnerable regions with limited resources and infrastructure.

Historically, the management of crop diseases has relied heavily on manual observation and diagnosis by trained agronomists and plant pathologists. While effective to some extent, this approach is labor-intensive, time-consuming, and often subjective, leading to delays in disease detection and suboptimal treatment decisions. Furthermore, with the emergence of new and more virulent pathogens, traditional disease management strategies are increasingly challenged, necessitating innovative solutions to enhance disease surveillance, diagnosis, and control.

In recent years, advances in technology, particularly in the field of machine learning (ML), have opened up new possibilities for improving crop disease management practices. ML techniques, which enable computers to learn from data and make predictions or decisions without being explicitly programmed, offer the potential to automate and streamline various aspects of disease detection and identification. By leveraging large datasets of crop images, sensor data, and other agricultural parameters, ML algorithms can learn to recognize patterns and symptoms associated with different diseases, allowing for faster and more accurate diagnosis.

This project seeks to harness the power of ML for the analysis and identification of crop diseases, with the overarching aim of providing farmers and agricultural stakeholders with a reliable tool for early disease detection and management. By developing ML-based models capable of accurately detecting and classifying crop diseases based on visual symptoms, this research endeavors to enhance the efficiency and effectiveness of disease surveillance efforts.

The remainder of this report is organized as follows: Section 2 provides a comprehensive review of existing literature on crop disease detection methodologies and the application of ML in agriculture. Section 3 outlines the methodology employed in this study, including data collection, preprocessing, and ML model development. Sections 4 and 5 present the results and discussion, respectively, while Section 6 offers concluding remarks and suggestions for future research directions.

**1.1 Methodology**

The methodology employed in this study involves several key steps, including data collection, preprocessing, feature extraction, model development, and evaluation. A diverse dataset comprising images of diseased crops is collected from various sources, ensuring representativeness across different crop types and disease severities. Preprocessing techniques such as image augmentation, normalization, and noise reduction are applied to enhance data quality and facilitate subsequent analysis. Feature extraction methods, including both handcrafted features and deep learning-based approaches, are utilized to capture relevant information from the images.

ML models are developed using a variety of algorithms, including Decision Trees, Random Forest, Support Vector Machines (SVM), and Convolutional Neural Networks (CNN). These models are trained on the preprocessed dataset using appropriate training/validation splits and optimization techniques. Model performance is evaluated using standard metrics such as accuracy, precision, recall, and F1-score, with rigorous cross-validation methods employed to assess robustness and generalizability.

***1.1.1 Data Collection:***

The first step in the methodology involves the collection of a diverse and representative dataset comprising images of diseased crops. Various sources are utilized to gather images depicting different types of crop diseases across multiple crop species and geographical regions. Efforts are made to ensure the quality and diversity of the dataset, with a focus on capturing a wide range of disease symptoms and severities.

***1.1.2 Data Preprocessing****:*

Once the dataset is assembled, preprocessing techniques are applied to prepare the data for model training and evaluation. This involves several steps, including image augmentation to increase the size and diversity of the dataset, normalization to standardize pixel values, and noise reduction to improve image quality. Additionally, techniques such as resizing and cropping may be employed to ensure consistency in image dimensions and aspect ratios.

***1.1.3 Feature Extraction:***

Feature extraction plays a crucial role in capturing relevant information from the image data and facilitating the learning process of ML models. Both handcrafted features and deep learning-based approaches may be utilized for this purpose. Handcrafted features, such as color histograms, texture descriptors, and shape features, capture domain-specific knowledge about disease symptoms. Deep learning-based feature extraction, on the other hand, involves the use of pre-trained convolutional neural networks (CNNs) to automatically learn discriminative features from raw image data.

***1.1.4 Model Development:***

With preprocessed data and extracted features in hand, ML models are developed using a variety of algorithms suited to the task of disease classification. Decision Trees, Random Forest, Support Vector Machines (SVM), and CNNs are among the algorithms considered for model development. Each algorithm is implemented and trained on the preprocessed dataset using appropriate training/validation splits and hyperparameter tuning techniques to optimize model performance.

***1.1.5 Model Evaluation:***

Model evaluation is conducted to assess the performance of the developed ML models in classifying crop diseases. Standard evaluation metrics such as accuracy, precision, recall, and F1-score are computed to quantify the performance of each model. Additionally, confusion matrices and receiver operating characteristic (ROC) curves may be analyzed to gain insights into model behavior and performance across different disease classes.

***1.1.6 Cross-Validation:***

To ensure the robustness and generalizability of the ML models, cross-validation techniques are employed during the training and evaluation process. K-fold cross-validation is commonly used, where the dataset is partitioned into k subsets, and each model is trained and evaluated k times, with each subset used as the validation set once. This helps mitigate the risk of overfitting and provides more reliable estimates of model performance.

***1.1.7 Model Deployment:***

Once the ML models are trained and evaluated, they can be deployed for practical use in disease surveillance and management systems. Deployment involves integrating the models into user-friendly interfaces or applications accessible to farmers and agricultural stakeholders. Cloud-based deployment options may also be explored to facilitate scalability and accessibility of the models.

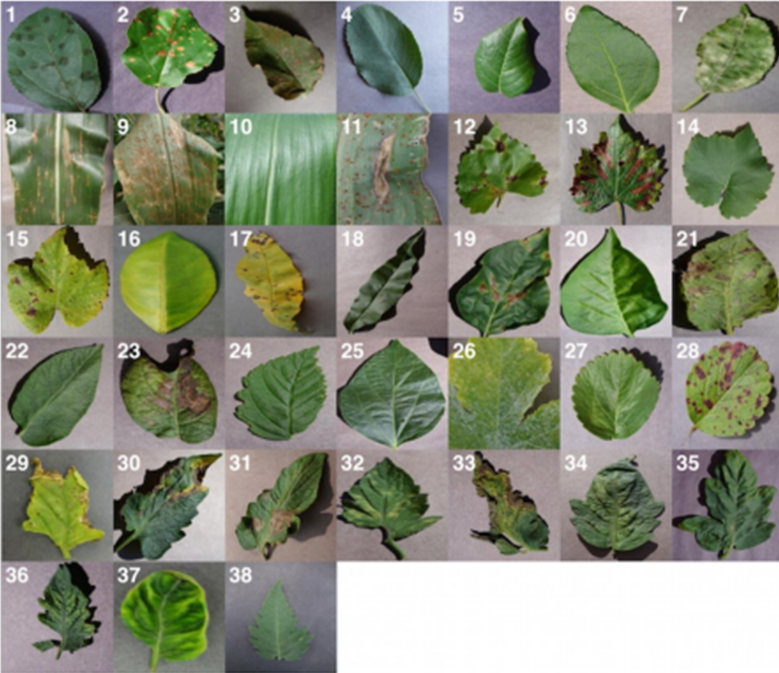
**Datasets :**

• Plant Village datasheet: The Plant Village datasheet be one for the widely use datasheets for plant disease detect. It have thousands of labeled pictures for diseased and healthy leaves from plant over different crop genus.

• Kraggle Documents: This datasheets have around 87K rgb pictures for healthy and diseased crop leafs that split into 38 diverse categories. <https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset>

**Table 1.1 – Information of the dataset**

|  |  |  |
| --- | --- | --- |
| **Plant** | **Disease Name** | **No. of Image** |
| **Apple** | Healthy | 2008 |
|  | Diseased Scab | 2016 |
|  | Diseased Diseased: Black rot | 1987 |
|  | Diseased: Cedar apple rust | 1760 |
| **Corn** | Healthy | 1859 |
|  | Diseased: Cercospora leaf spot | 1642 |
|  | Diseased Common rust | 1907 |
|  | Diseased: Northern Leaf Blight. | 1908 |
| **Grapes** | Healthy | 1692 |
|  | Diseased: Black rot | 1888 |
|  | Diseased: Esca (Black Measles) | 1920 |
|  | Diseased: Leaf blight (Isariopsis) | 1722 |
| **Potato** | Healthy | 1824 |
|  | Diseased: Early blight | 1939 |
|  | Diseased: Late blight | 1939 |
| **Tomato** | Healthy | 1926 |
|  | Diseased: Bacterial spot | 1702 |
|  | Diseased: Early blight | 1920 |
|  | Diseased: Late blight | 1851 |
|  | Diseased: Leaf Mold | 1882 |
|  | Diseased: Septoria leaf spot | 1745 |
|  | Diseased: Two-spotted spider mite | 1741 |
|  | Diseased: Target Spot | 1827 |
|  | Diseased: Yellow Leaf Curl Virus | 1961 |
|  | Diseased: Tomato mosaic virus | 1790 |

****

*Fig 1.1 Some examples of images used in each different 38 sets*

**1.2 Purpose**

The purpose of this project is multi-faceted, addressing several key objectives aimed at leveraging machine learning (ML) techniques to enhance crop disease management and contribute to sustainable agriculture. The overarching goal is to provide farmers and agricultural stakeholders with a reliable tool for early disease detection, accurate diagnosis, and effective management strategies. Below are the main purposes of the project:

1. ***Enhance Disease Surveillance:*** The primary purpose of this project is to improve disease surveillance capabilities in agriculture. By developing ML-based models capable of analyzing large datasets of crop images, sensor data, and other relevant parameters, the project aims to enable more efficient and accurate monitoring of disease outbreaks in agricultural fields. This enhanced surveillance can help farmers detect diseases early, allowing for timely intervention and mitigation strategies to minimize crop losses.
2. ***Facilitate Early Detection and Diagnosis:*** Early detection and accurate diagnosis of crop diseases are crucial for implementing effective management strategies and minimizing yield losses. By automating the process of disease detection using ML algorithms, this project seeks to provide farmers with timely and accurate information about the presence and severity of crop diseases in their fields. Early detection enables farmers to take proactive measures such as targeted spraying, crop rotation, or disease-resistant crop varieties to mitigate the impact of diseases on crop yields.
3. ***Optimize Resource Allocation:*** Effective disease management requires judicious allocation of resources such as pesticides, fungicides, water, and labor. By providing farmers with actionable insights based on ML-based disease detection models, this project aims to optimize resource allocation and reduce input costs associated with disease management. By targeting interventions only where and when they are needed most, farmers can minimize waste and maximize the efficiency of their agricultural practices.
4. ***Support Sustainable Agriculture****:* Sustainable agriculture aims to balance the needs of current and future generations by maximizing productivity while minimizing environmental impact. By empowering farmers with tools for more precise and targeted disease management, this project contributes to the broader goal of sustainability in agriculture. By reducing reliance on chemical inputs and optimizing resource use, farmers can minimize the environmental footprint of their operations and promote more ecologically sustainable farming practices.
5. ***Advance Technological Innovation****:* Finally, this project aims to advance technological innovation in agriculture by harnessing the power of ML and data analytics for disease management. By demonstrating the feasibility and effectiveness of ML-based disease detection models, the project not only provides practical benefits to farmers but also contributes to the ongoing evolution of agricultural technologies. This fosters a culture of innovation and continuous improvement in agricultural practices, paving the way for future advancements in the field.

**1.3 Scope**

The scope of this project delineates the boundaries and extent of the research, specifying the focus areas, objectives, and limitations. It encompasses the following aspects:

1. ***Disease Identification and Analysis****:* The primary focus of the project is on the analysis and identification of crop diseases using machine learning (ML) techniques. The scope includes developing ML models capable of accurately classifying crop diseases based on visual symptoms observed in images of diseased crops. These symptoms may include leaf discoloration, lesions, wilting, deformities, and other indicators of disease presence.
2. ***Crop Types and Diseases****:* The project aims to address a range of crop types and prevalent diseases that have significant economic impact on agricultural production. While the focus may vary depending on the availability of data and resources, efforts are made to cover a diverse range of crops and diseases..
3. ***Data Sources and Availability***: The scope includes utilizing diverse datasets of crop images sourced from various sources, including publicly available repositories, research institutions, and collaborative partnerships with agricultural organizations. The availability and quality of data may influence the scope of the project, with efforts made to ensure representativeness and relevance to real-world agricultural scenarios.
4. ***Model Development and Evaluation****:* The project involves developing ML models using a variety of algorithms, but mainly focuses on Convolutional Neural Networks (CNN). These models are evaluated using standard performance metrics such as accuracy, precision, recall, and F1-score, with rigorous cross-validation techniques employed to assess robustness and generalizability.
5. ***Practical Application and Deployment:*** While the primary focus is on model development and evaluation, the project also considers the practical application and deployment of ML-based disease detection systems in real-world agricultural settings. This may involve integrating the developed models into user-friendly interfaces or applications accessible to farmers and agricultural stakeholders, with considerations for scalability, usability, and performance.
6. ***Limitations:*** Despite the comprehensive scope outlined above, the project may be subject to certain limitations. These may include constraints related to data availability, computational resources, and expertise. Additionally, the scope may be influenced by external factors such as time constraints and project objectives. Efforts are made to acknowledge and address these limitations within the context of the project.

**1.4 Tools Used**

In this project, a variety of tools are utilized to facilitate different stages of the machine learning (ML) workflow, from data preprocessing to model development and evaluation. These tools include programming languages, libraries, frameworks, and software applications that provide essential functionalities for tasks such as data manipulation, model training, visualization, and deployment. Below are the main tools used in the project:

1. Python: Python is a versatile programming language widely used in data science, machine learning, and scientific computing. Its simplicity, readability, and extensive ecosystem of libraries make it well-suited for ML projects.
2. NumPy: NumPy is a fundamental library for numerical computing in Python. It provides support for multidimensional arrays, mathematical functions, and linear algebra operations, making it essential for data manipulation and preprocessing tasks.
3. Pandas: Pandas is a powerful library for data manipulation and analysis in Python. It provides data structures such as DataFrames and Series, along with functions for indexing, merging, reshaping, and aggregating data, making it invaluable for working with structured datasets.
4. TensorFlow: TensorFlow is an open-source deep learning framework developed by Google. It provides a flexible and scalable platform for building and training deep neural networks, including convolutional neural networks (CNNs) commonly used for image classification tasks.
5. Matplotlib: Matplotlib is a popular plotting library for creating static, interactive, and publication-quality visualizations in Python. It offers a wide range of plotting functions and customization options, making it suitable for visualizing data distributions, model performance metrics, and other relevant information.
6. Seaborn: Seaborn is a statistical data visualization library based on Matplotlib. It provides additional high-level functions for creating attractive and informative statistical graphics, including heatmaps, violin plots, and pair plots, enhancing the aesthetics and interpretability of visualizations.
7. Jupyter Notebook: Jupyter Notebook is an open-source web application that allows users to create and share documents containing live code, equations, visualizations, and narrative text. It provides an interactive environment for conducting data analysis, exploring ML algorithms, and documenting project workflows.
8. Git: Git is a distributed version control system used for tracking changes in software development projects. It allows multiple developers to collaborate on a project simultaneously, track changes over time, and manage code repositories efficiently.

These tools collectively provide the necessary infrastructure and resources for implementing the ML-based approach to crop disease detection, from data preprocessing and feature engineering to model development, evaluation, and deployment. They enable efficient workflow management, code development, and collaboration, contributing to the success and reproducibility of the project.

**1.5 Overview**

The overview section provides a high-level summary of the project, outlining its objectives, methodology, and expected outcomes. It serves as a roadmap for readers, providing context and setting the stage for the subsequent sections of the project report. Below are the key components of the overview:

* Objectives: The project aims to develop and evaluate machine learning (ML) models for the analysis and identification of crop diseases. The primary objectives include enhancing disease surveillance, facilitating early detection and diagnosis, optimizing resource allocation, supporting sustainable agriculture, and advancing technological innovation in agriculture.
* Methodology: The methodology section outlines the systematic approach employed in the project, including data collection, preprocessing, feature extraction, model development, evaluation, and deployment. It highlights the use of diverse datasets, ML algorithms, and evaluation metrics to achieve the project objectives.
* Expected Outcomes: The project expects to deliver ML-based models capable of accurately detecting and classifying crop diseases based on visual symptoms observed in images of diseased crops. These models are expected to provide farmers and agricultural stakeholders with a reliable tool for early disease detection, accurate diagnosis, and effective management strategies.
* Significance: The project addresses a critical need in agriculture by leveraging ML techniques to improve disease surveillance and management practices. By providing farmers with timely and accurate information about crop diseases, the project aims to mitigate yield losses, optimize resource use, and promote sustainable agricultural practices.
* Challenges and Limitations: The overview may also mention potential challenges and limitations that could impact the project, such as data availability, computational resources, and technical expertise. By acknowledging these challenges upfront, the project demonstrates transparency and sets realistic expectations for the reader.

**Chapter 2**

**Literature Review**

Review existing literature on plant disease detection using machine shows a growth body of research focus on leverage machine learning algorithms, specially convolutional neural networks (CNNs), for automated disease identify in corps. Studies have show the effective of ML models in accurately detect many different diseases among various plant species and environment. Major themes include dataset creation, model architect optimization, and metrics for evaluation of performance. Challenges like dataset scarcity, class imbalance, and model interpretability also get highlighted, indicating roads for future research and innovation in this area. Some of the articles we have seen are:

1. **Smart Farming: Pomegranate Disease Detection Using Image Processing, 2015**

A web-based tool has been developed to identify fruit diseases by uploading fruit

image to the system. Feature extraction has been done using parameters such as colour, morphology and CCV (colour coherence vector). Clustering has been done using the k-means algorithm. SVM is used for classification as infected or non-infected. This work achieved an accuracy of 82%to identify pomegranate disease.

1. **Leaf Disease Detection and Recommendation of Pesticides using Convolution Neural Network, 2018**

Crop production problems are common in India which severely effect rural farmers, agriculture sector and the country’s economy as a whole. In Crops leaf plays an important role as it gives information about the quantity and quality of agriculture yield in advance depending upon the condition of leaf. In this paper we proposed the system which works on pre-processing, feature extraction of leaf images from plant village dataset followed by convolution neural network for classification of disease and recommending Pesticides using Tensor flow technology. The main two processes that we use in our system is android application with Java Web Services and Deep Learning. We have use Convolution Neural Network with different layers five, four& three to train our model and android application as a user interface with JWS for interaction between these systems. Our results show that the highest accuracy achieved for 5-layer model with 95.05% for 15 epochs and highest validation accuracy achieved is for 5- layer model with89.67% for 20 epochs using tensor flow.

1. **An Artificial Intelligence and Cloud Based Collaborative Platform for Plant Disease Identification, Tracking and Forecasting for Farmers, 2018**

This paper presents an automated, low cost and easy to use end-to-end solution to one of the biggest challenges in the agricultural domain for farmers –precise, instant and early diagnosis of crop diseases and knowledge of disease outbreaks - which would be helpful in quick decision making for measures to be adopted for disease control. This proposal innovates on known priorart with the application of deep Convolutional Neural Networks (CNNs) for disease classification, introduction of social collaborative platform for progressively improved accuracy, usage of geocoded images for disease density maps and expert interface for analytics. High performing deep CNN model “Inception” enables real time classification of diseases in the Cloud platform via a user facing mobile app.

1. **CNN based Leaf Disease Identification and Remedy Recommendation System, 2019**

This paper focus upon plant disease detection using image processing approach. This work utilizes an open dataset of 5000 pictures of unhealthy and solid plants, where convolution system and semi supervised techniques are used to characterize crop species and detect the sickness status of 4 distinct classes. Convolution neural network is used to detect and classify plant diseases. The Network is trained using the images taken in the natural environment and achieved 99.32% classification ability. This shows the ability of CNN to extract important features in the natural environment which is required for plant disease classification. [6]

1. **Plant Leaf Diseases Detection and Classification Using Image Processing and Deep Learning Techniques, 2020**

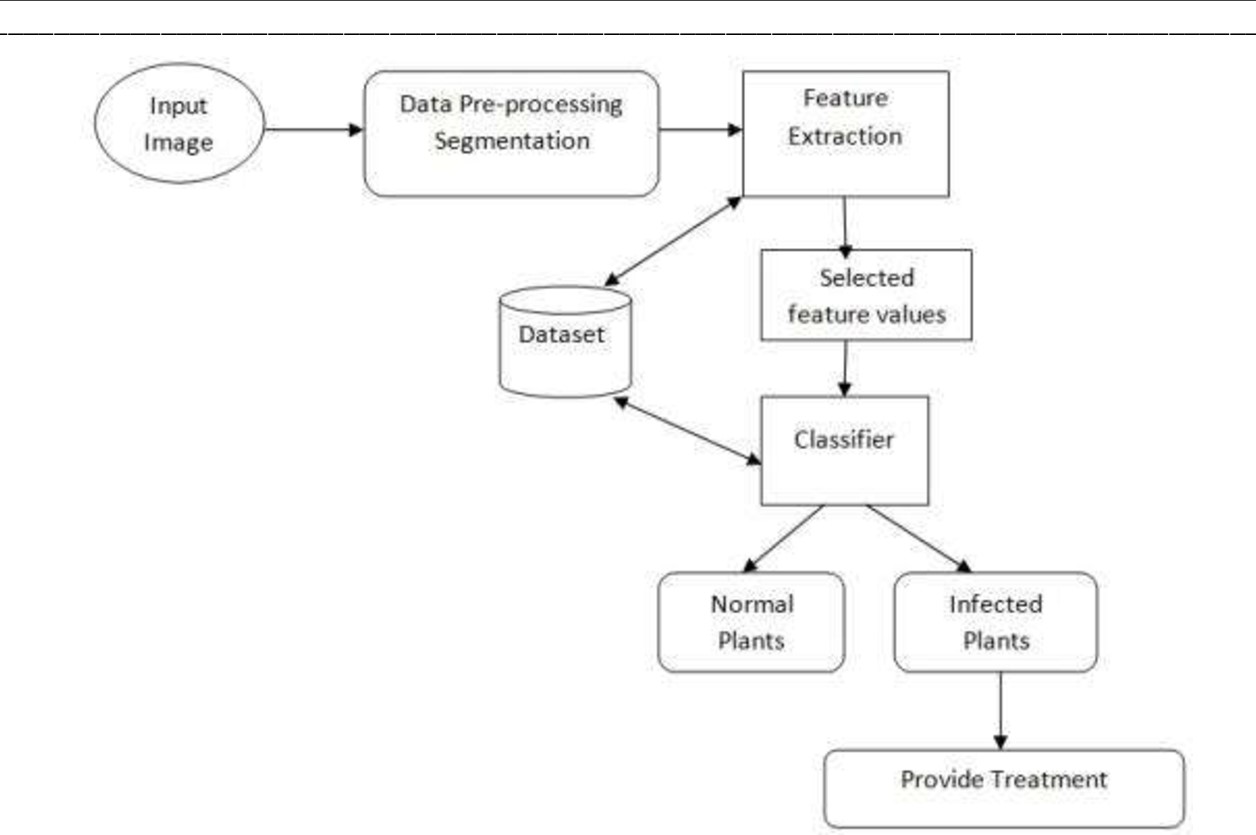
This paper presents a system that is used to classify and detect plant leaf diseases using deep learning techniques. The used images were obtained from (Plant Village dataset) website. In our work, we have taken specific types of plants; include tomatoes, pepper, and potatoes, as they are the most common types of plants in the world and in Iraq in particular. This Data Set contains 20636 images of plants and their diseases. In our proposed system, we used the convolutional neural network (CNN), through which plant leaf diseases are classified, 15classes were classified, including 12 classes for diseases of different plants that were detected, such as bacteria, fungi, etc., and 3 classes for healthy leaves. As a result, we obtained excellent accuracy in training and testing, we have got an accuracy of (98.29%) for training, and(98.029%) for testing for all data set that were used.[7]

**Chapter 3**

**SYSTEM DESIGN**

**3.1 ER-Diagram:**

The Entity-Relationship (E-R) diagram encapsulates the structural blueprint of the plant leaf disease recognition software, delineating the interconnections between key entities and their attributes. At its core, the diagram depicts four primary entities: "Images," "Diseases," "Predictions," and "Plants," each encapsulating distinct aspects of the software's functionality. Images serve as the repository for uploaded plant leaf images, complemented by Diseases, which catalog various plant diseases and their characteristics. Predictions bridge the gap between images and diseases, capturing the system's predictive outputs alongside confidence levels. Lastly, Plants encompass the diversity of plant species impacted by diseases, grounding the software in real-world botanical taxonomy. Through this cohesive schema, the E-R diagram illuminates the intricate relationships and dependencies that underpin the software's data model, facilitating a holistic understanding of its structural design and operational dynamics.



*Fig. 3.1 ER-diagram*

**3.2 Architecture Design:**

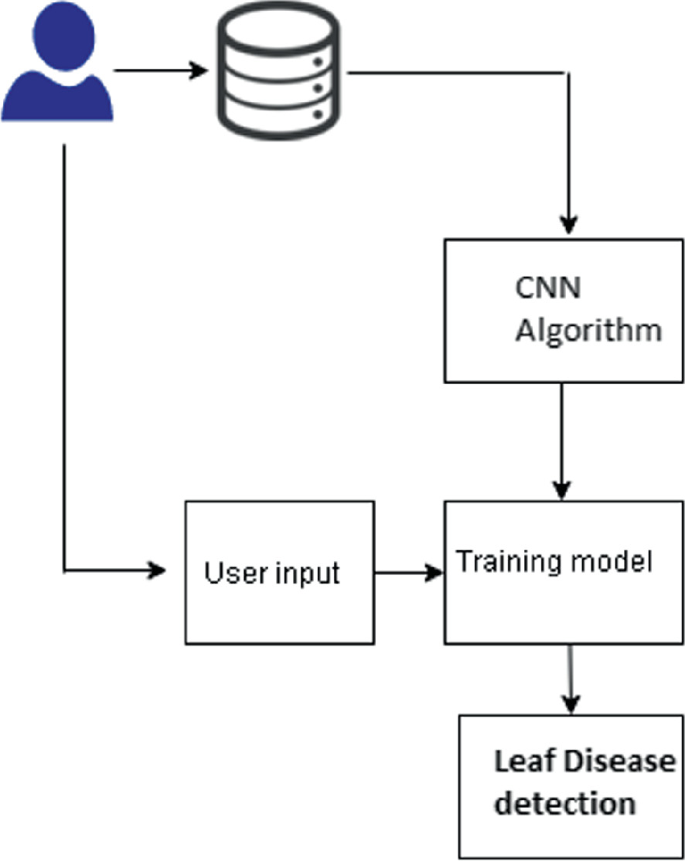


Fig. 3.2 Architecture design

**Image Description:**

The image presented in the project report portrays a comprehensive architectural depiction of the Plant Leaf Disease Recognition System. At the forefront of the image is a user icon, symbolizing user interaction within the system. This user is intricately connected with two essential elements: the database and user input. This linkage underscores the user's pivotal role in providing input data and accessing stored information crucial for disease recognition tasks.

**Database Connectivity:**

The database component forms the backbone of the system's data management infrastructure. It serves as a centralized repository for storing and organizing various types of data, including images, metadata, and model parameters. In the architectural depiction, the database is seamlessly linked with the Convolutional Neural Network (CNN) algorithm, illustrating the flow of data from storage to processing.

**CNN Algorithm and Training Model Integration:**

The CNN algorithm represents the core engine driving the disease recognition capabilities of the system. Trained on extensive datasets of labeled images, the CNN algorithm possesses the ability to extract meaningful features indicative of plant diseases. In the architectural diagram, the CNN algorithm is intricately connected with the training model, highlighting the iterative process of model refinement and optimization through continuous training **iterations**.

**Leaf Disease Detection:**

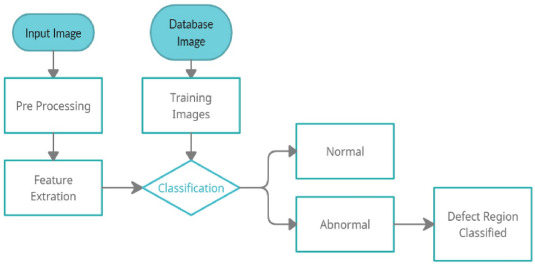
At the final stage of the architectural flow, the leaf disease detection component signifies the culmination of the system's efforts in identifying and diagnosing plant diseases. Leveraging the knowledge acquired through the CNN algorithm and training model, the system accurately detects and classifies various disease types, providing valuable insights for disease management and mitigation strategies.

**3.3 Activity diagrams**

Activity diagrams depict the flow of activities or processes within the system, illustrating the sequence of actions, decisions, and interactions between different components. In the context of the crop disease detection system, activity diagrams may illustrate processes such as image preprocessing, model training, disease classification, and result presentation. Each diagram includes:

* Activities: The tasks or actions performed within the system.
* Decisions: Points where the flow diverges based on certain conditions or criteria.
* Transitions: Arrows indicating the flow of control between activities.
* Start and End Points: Nodes representing the beginning and end of the process.

Example Activity Diagram: "Disease Diagnosis Process”

* Activities:

1. Upload Image
2. Preprocess Image
3. Classify Disease
4. Display Results

* Decisions: Fig. 3.3 Activity diagram

1. If Image Quality is Poor: Redirect to "Upload Image" activity.
2. If Disease Identified: Display Results; Else, Display Error Message.

* Transitions: Arrows connecting activities and decisions.
* Start Point: "Upload Image"
* End Point: "Display Results"

**Chapter 4**

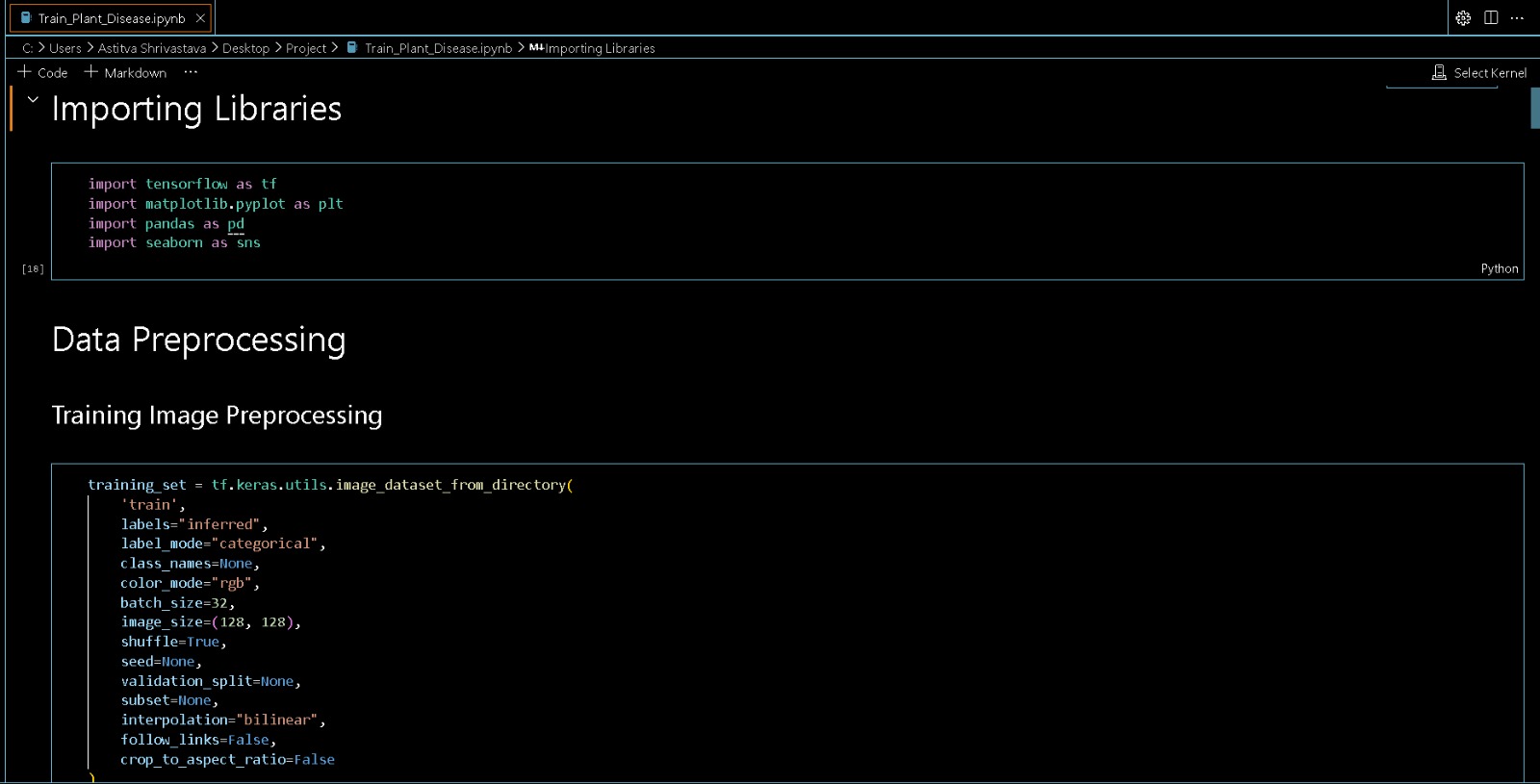
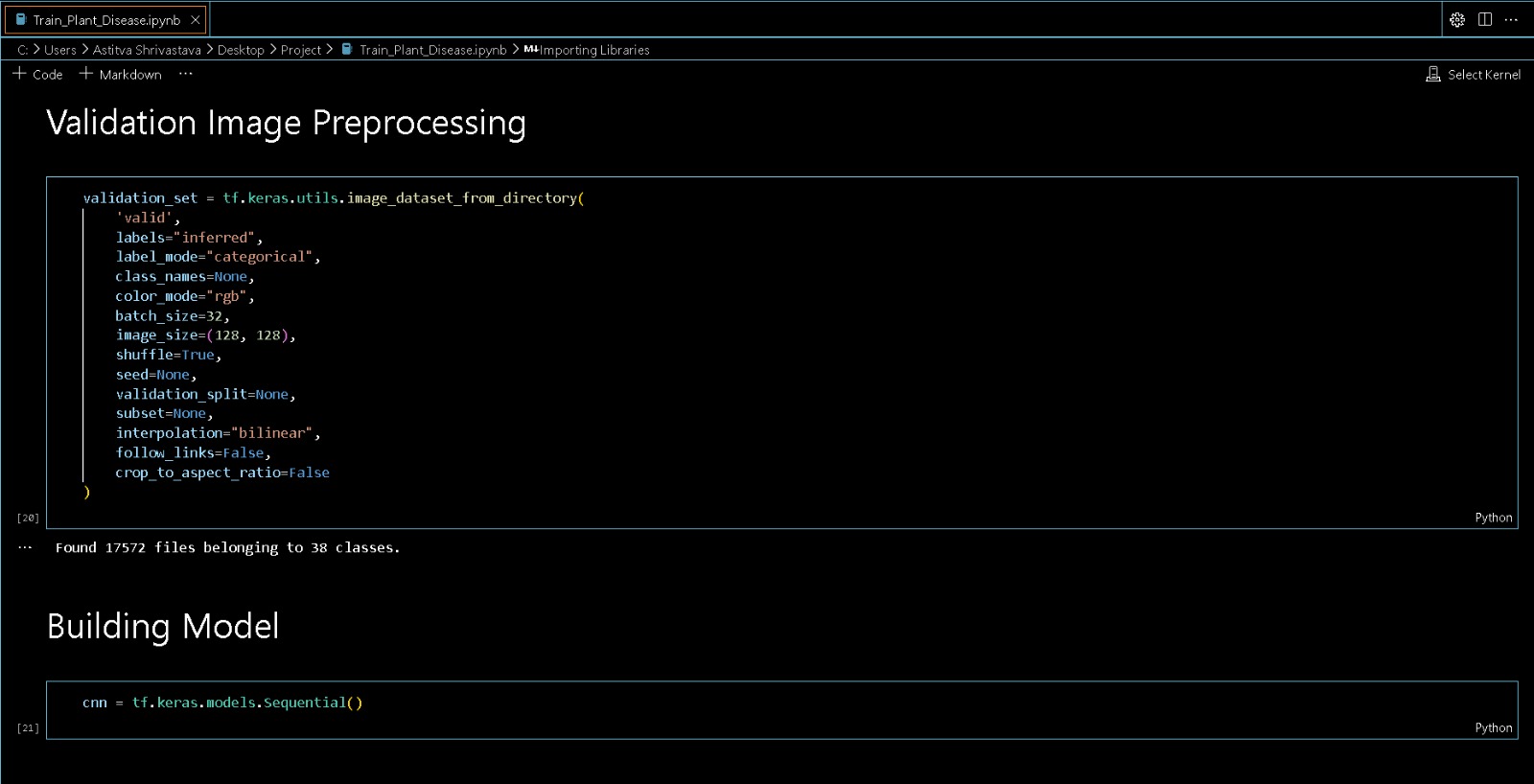
**IMPLEMENTATION**

The experimental setup for plant disease detection using machine learning typically involves several key steps, including dataset preparation, model training, validation, testing and GUI. Here's an outline of the experimental setup:

* 1. **Data Collection and Preprocessing:**

Images of healthy plants and plants affected by various diseases are collected to create a dataset. Utilize a publicly available dataset of plant leaf images containing healthy and diseased samples. The dataset source is **kraggle** and the number of classes it covers is **38**.

For doing the data preprocessing and other works in this project we want to import some important libraries like **tensorflow, keras, matplotlib, seaborn and etc**. Data preprocessing techniques such as resizing, normalization, and augmentation applied using the tensorflow and keras library and arguments(**tf.keras.utils.image\_dataset\_from\_directory**).



**Fig 4.1 Figures showing the imported libraries and image preprocessing steps**

**4.2 Feature Extractions:**

For image-based models, Convolutional Neural Networks (CNN) are using to automatically extract relevancy features from images. CNN is a types of neural network that commonly used for image recognition tasks. They works by extraction features from images using filters. These features are then uses to classify image. They contain multiple layers, includes convolutional, pooling, and connected fully layers. CNNs using convolutional operations for extract features from input images, pooling layers for decrease spatial dimensions, controls overfitting, and fully connected layers for classification or regression tasks. Through training on labeled data, CNNs learns hierarchical representations of visual features, enable them for recognize patterns and objects in images. This makes CNNs high effective for plant disease detections in agriculture settings.

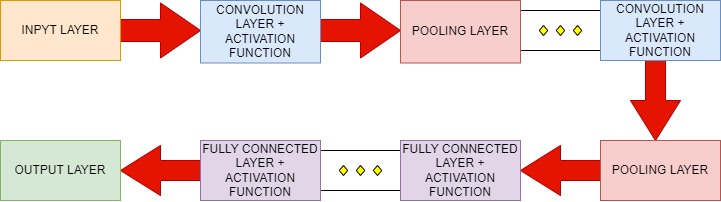
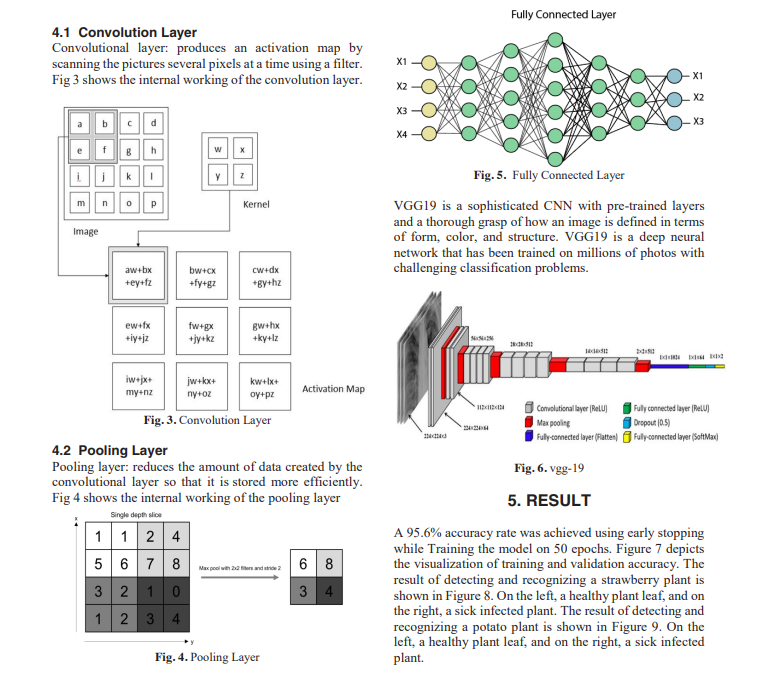
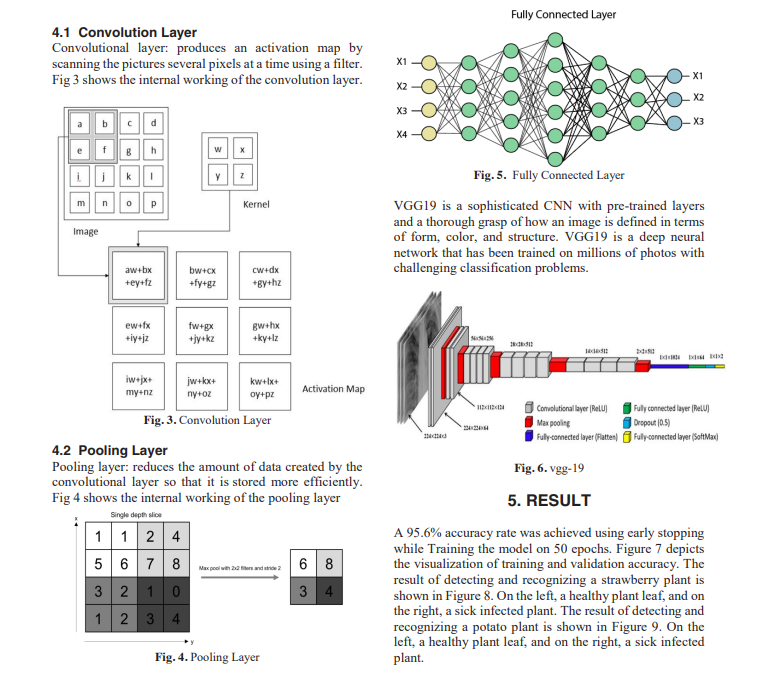


Fig. 4.2. The architecture of a typical [CNN](https://www.sciencedirect.com/topics/computer-science/convolutional-neural-network)

The key steps of an Convolutional Neural Network (CNN) algorithm for featuring extraction in plant diseases detection, along with relevant formulas:

* + 1. ***Input Layer:***
* Received imaged of size W×H×C, where W are the width, H is the height, and C are an number of channels (e.g., RGB coloring).
* A channel is a specific feature map result from applying filters to the input details, typically represent different learned patterns or featuring.
  + 1. ***Convolutional Layer:***
* Apply convolution an operation to the input image using a set of learnable filters (kernels). This layer is responsible for extracting features from an input image. We specify the numbers of filters to use in the convolution layer, as well as the kernel size and padding (The kernel size is an size of the filter that will be applied to an image. Padding refers to how the edges of an image are handling during the convolution operation).

*Fig 4.3 Convolutional layer*

* Output feature map size:Wo=, Ho​=, where *F* is the filter size, *P* is the padding, and *S* is the stride.
* Output feature map depth: Number of filters.
* **Activation Function:** Activation Function: Apply an activation function (e.g., ReLU) element-wise to introducing non-linearity: fun(x)=max(0,xi).
  + 1. *****Pooling Layer:***
* Downsample an feature maps to reduce spatial dimensions and control overfitting. This layer is used to reduce an size of an output from an

*Fig 4.4 Pooling layer*

convolution layer. we specify an pool size and stride for an pooling layer. an pool size is the size of an window that is used to slid over an output from an convolution layer. an stride is the number of pixies that an window is moved after each step.

* Output feature map size: Wo=SWi-F+1, Ho=SHi-F+1, where F is the pooling size and S is an stride.
  + 1. ***Flattening:***
* Flatten an pooled feature maps into an 1D vector to be fed into an fully connected layers. This layer flattens an output from an previous layer into an one-dimensional vector. an vector is then fed to an dense layer, which is an fully connected layer. an dense layer is used to classify an image.
  + 1. ***Fully Connected Layers:***
* Connect every neuron in one layer to every neuron in an next layer.
* Number of parameters for each fully connected layer: Np=(Ni+1)×No, where an Ni and No ares the numbers of an input and output neurons, respectively.

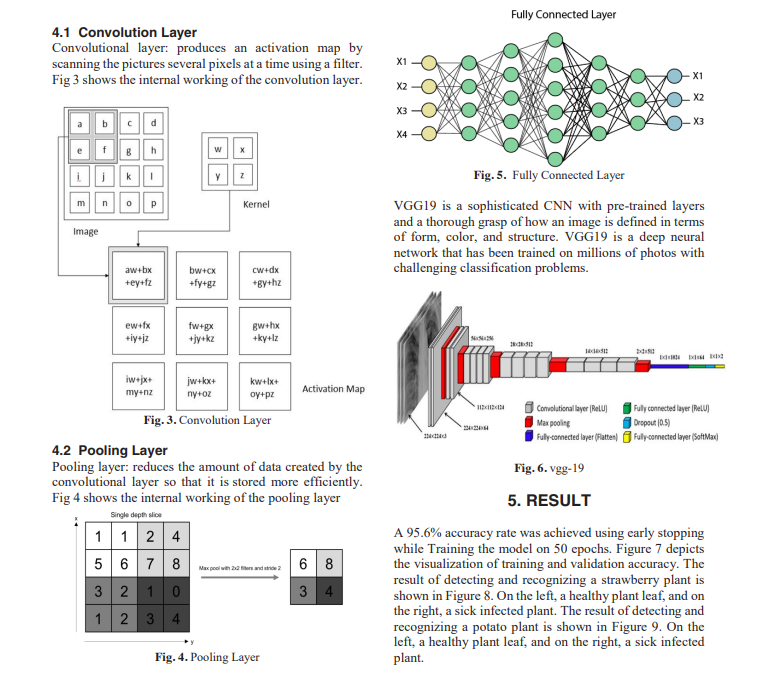


Fig. 4.5 Fully connected layer

* + 1. ***Output Layer:***
* Apply an appropriate activation function (e.g., softmax for classification) to obtain class probabilities.
* SoftMax function: SoftMax(*xi*​)= ​​, where *xi*​ is the raw output score for class *i*, and an *N* is the number of classes.
  1. **Model Training:**

an extracted features are an fed into an classification model, typically an fully connected neural network.an model is trained using an labeled dataset, adjusting the model's parameters to minimize an classification errors.

Here we face some problems like overshooting loss functions during training and overfitting.To address this problem, we use-

* Decreased the learning rate
* Increased the number of neurons in the dense layer
* Added more convolutional layers to extract more features from the image
* Dropout is used to avoid overfitting.

Padding is used to make an feature map size that same as an input size in an first convolutional layer.

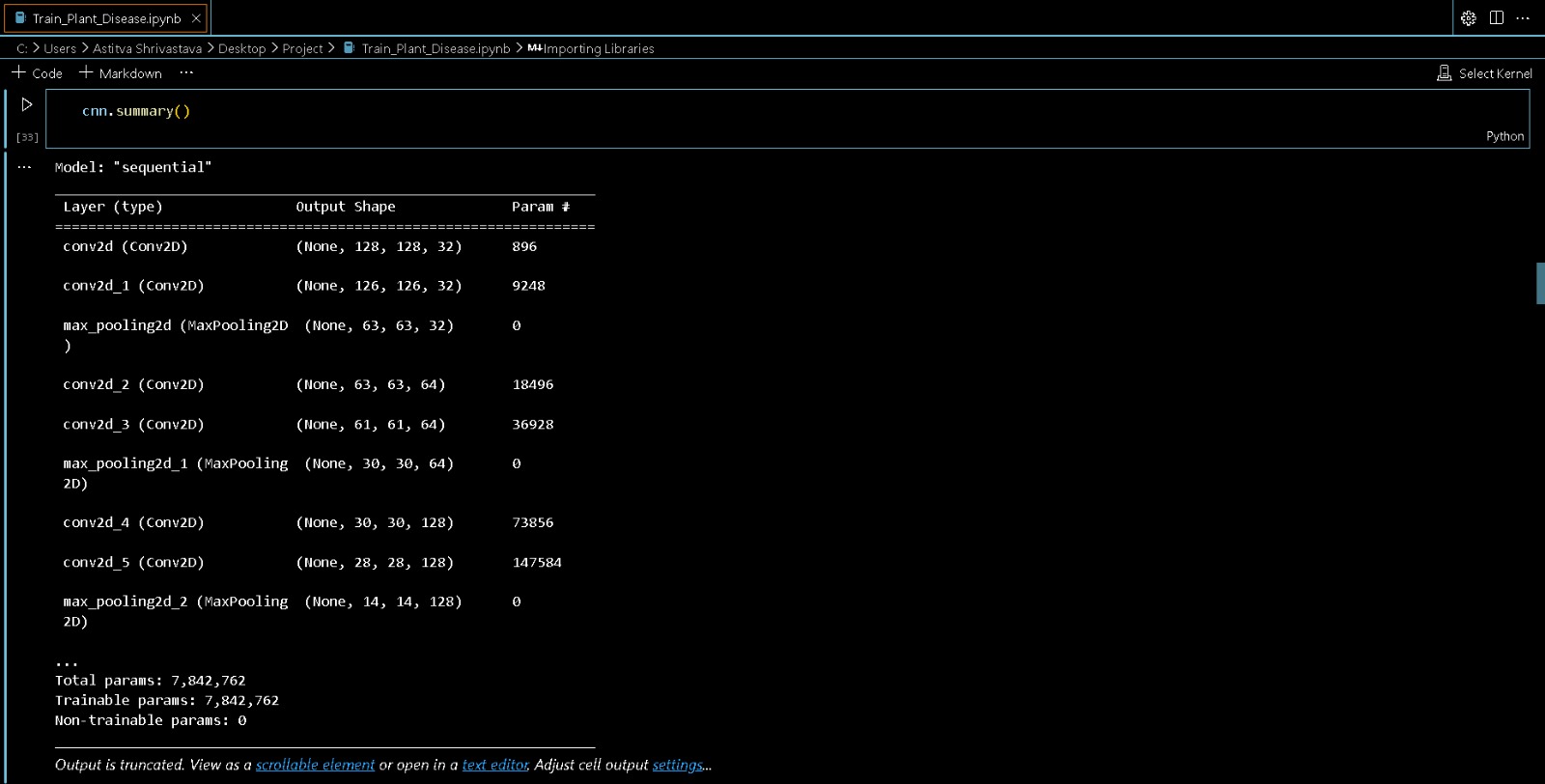


Table 4.1

After training the model, the speaker achieved 96.84% training accuracy and 95.52% validation accuracy.

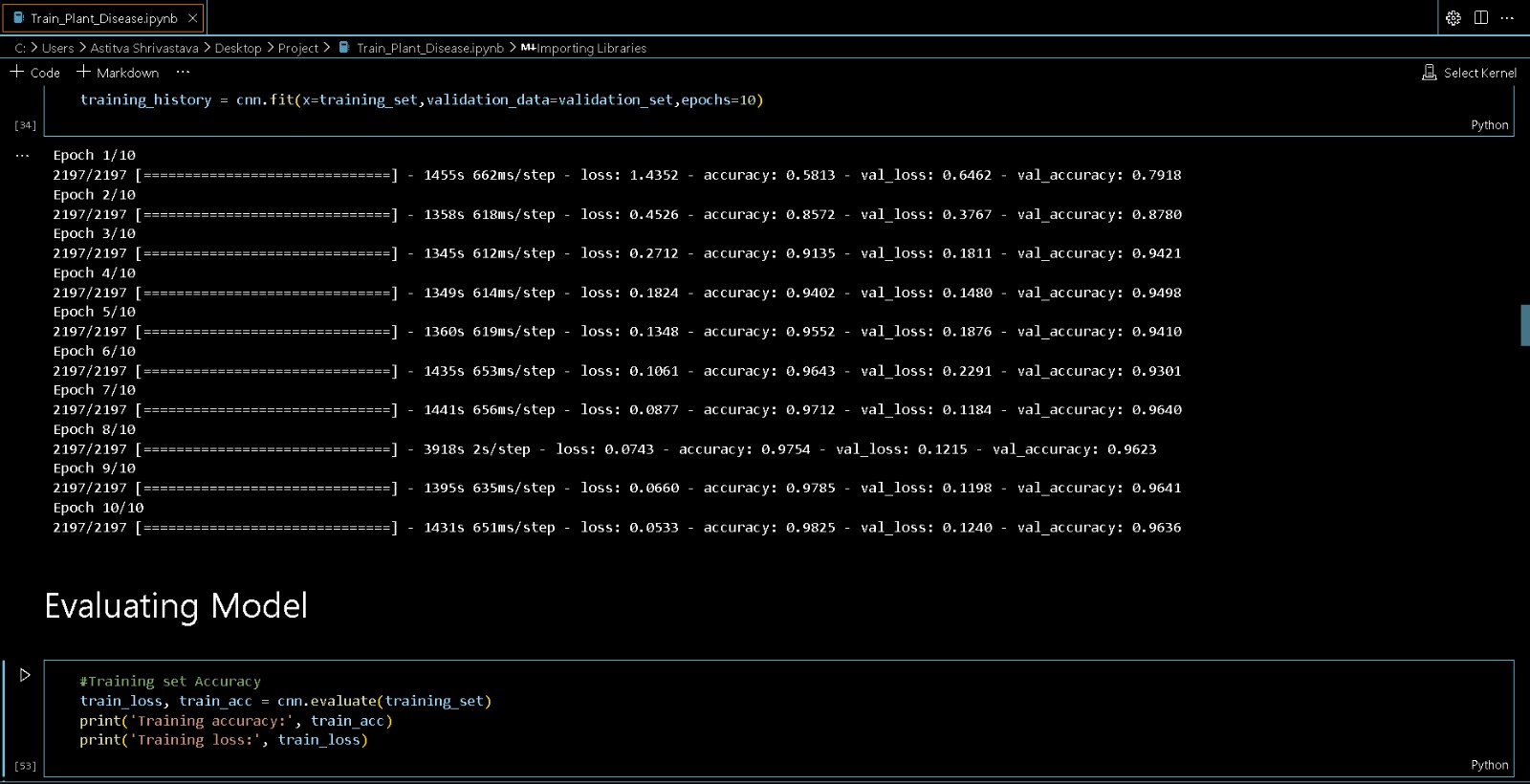


Fig. 4.7

* 1. **Model Evaluation:**

Model evaluation on an training set and validation set – first we defines an functions to evaluate an model on an training set and validation set. Then runs an functions and gets an training loss, an training accuracy, an validation loss, and an validation accuracy.

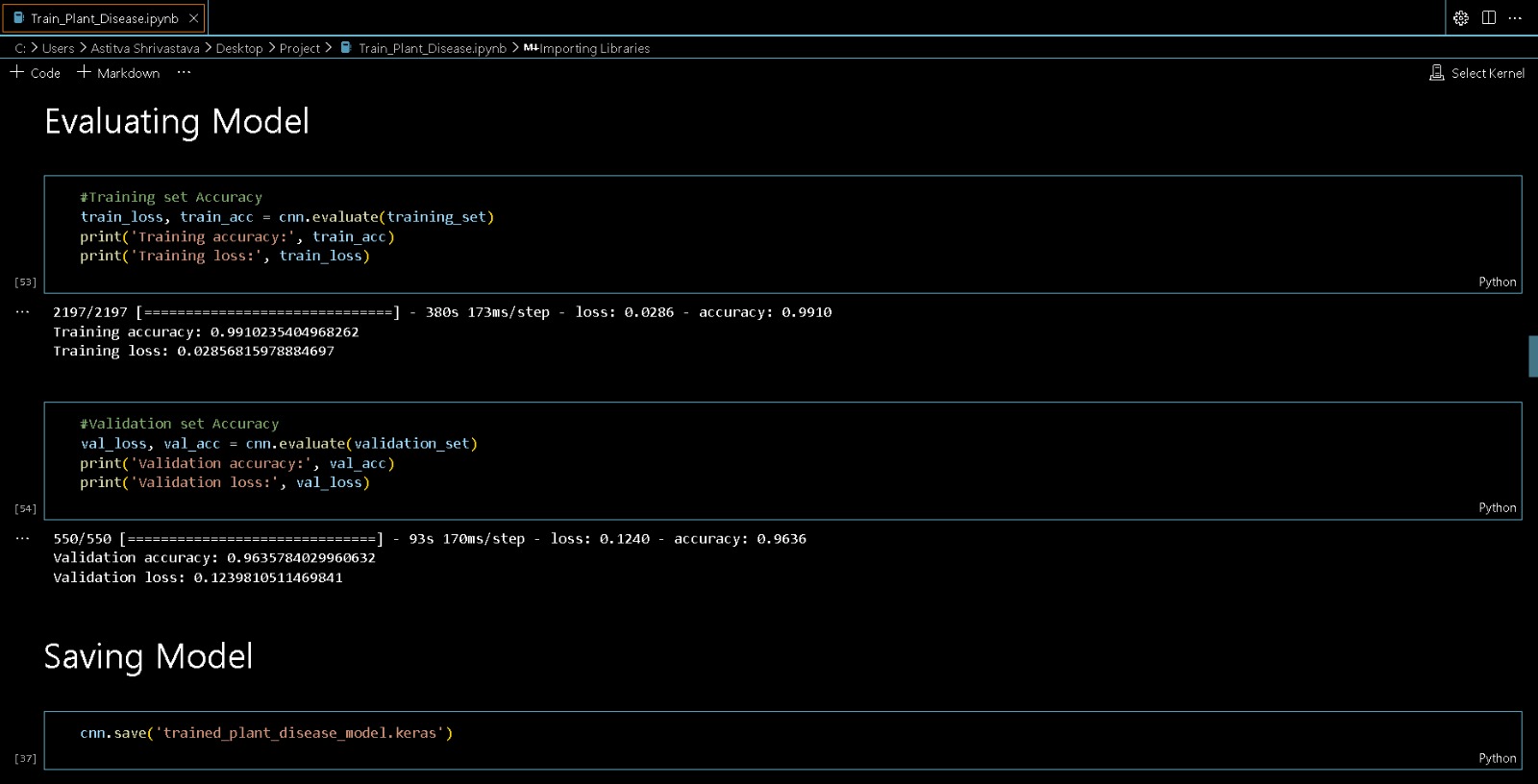


Fig. 4.8

Saving an model- there are two ways to save an model: H5 format and Caris format. Caris format is a compressed format that reduces the file size.

Recording model history- it include training loss, training accuracy, validation loss, and validation accuracy, into an Json file.

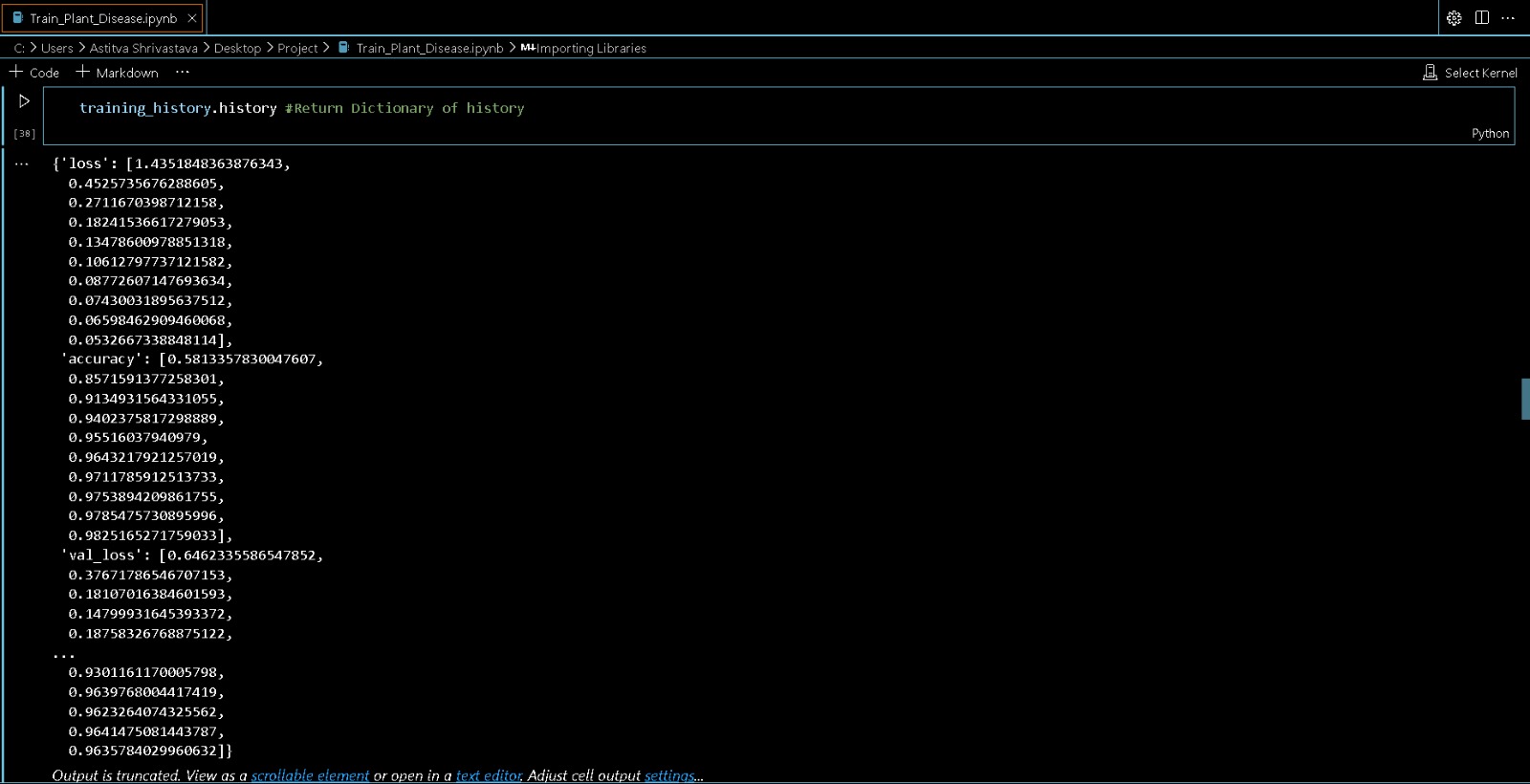


Fig. 4.9

Model accuracy visualization- Here the code to visualise the training accuracy and validation accuracy vs epoch.



*No of epochs* Fig.4.10

* 1. **Software Interface**

The software interface for the ML-based crop disease detection system is designed to be user-friendly, intuitive, and accessible to farmers, agricultural experts, and other stakeholders. The interface includes various components and features to facilitate the upload of images, disease diagnosis, visualization of results, and access to recommendations. Below is an elaborate description of the key components and functionalities of the software interface, including the homepage.

*Key Components:*

1. Homepage
2. Image Upload Section
3. Disease Diagnosis and Result Display

**1. Homepage**

Purpose:

* Serve as the entry point for the application, providing an overview and guiding users to key functionalities.

Features:

* Welcome Message: Introduction to the application and its purpose.
* Main Navigation: Links to Login/Registration, About, Features, Contact, and other key sections.
* Features Overview: Brief descriptions of main features such as disease diagnosis, historical data, and recommendations.
* Footer: Contact information, social media links, and additional resources.

User Flow:

* User visits the homepage.
* User reads about the application and its features.

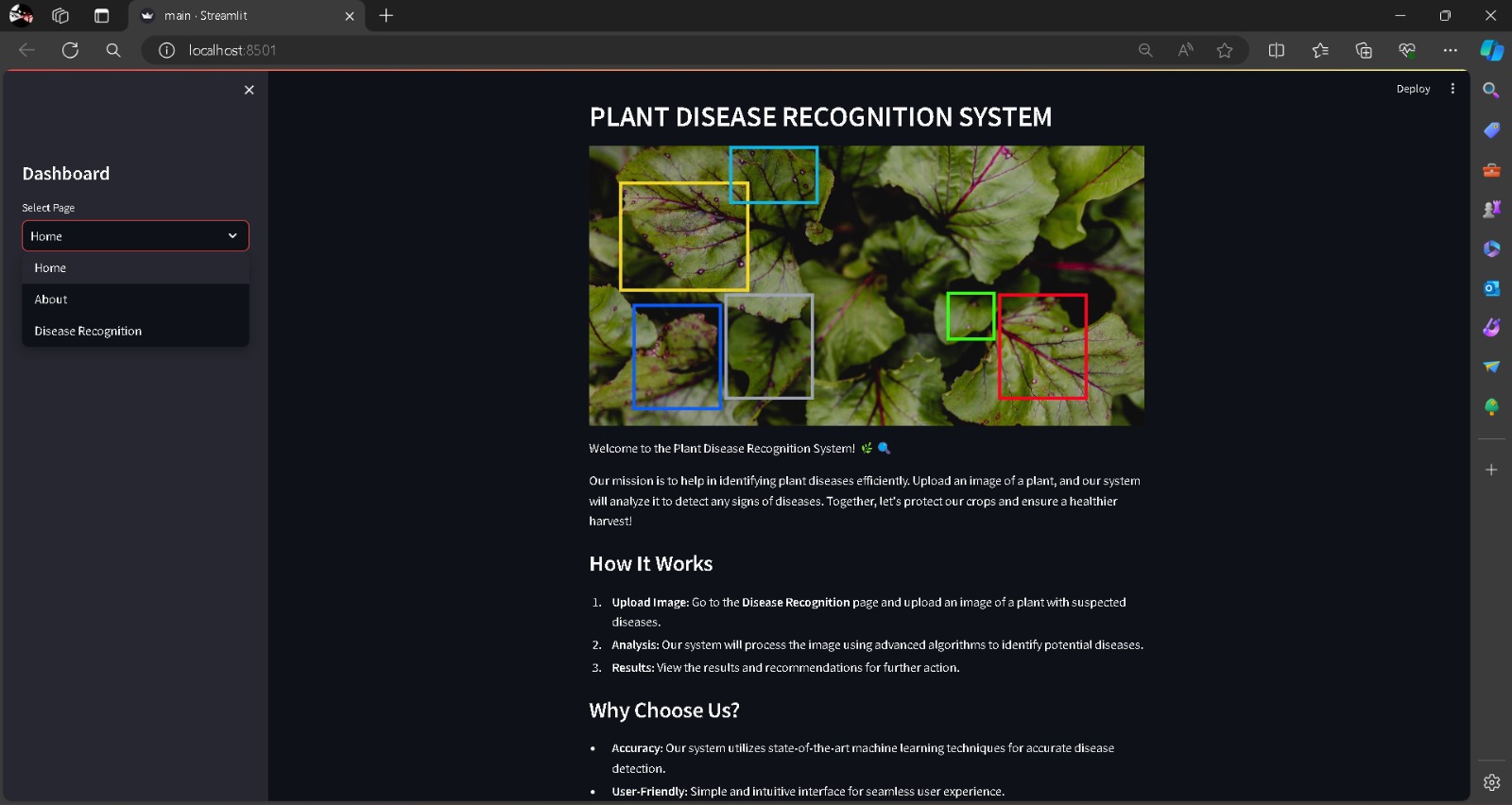


Fig. 4.11 Homepage

**2****. Image Upload Section**

Purpose:

* Allow users to upload images of crops for disease diagnosis.

Features:

* Upload Button: Option to select and upload images from the user's device.
* Drag and Drop Area: Alternative method for image upload.
* Image Preview: Display the uploaded image for user confirmation.
* Submit Button: Initiate the disease diagnosis process.

User Flow:

* User clicks the upload button or drags and drops an image.
* The image is previewed, and the user confirms the upload.
* User clicks the submit button to send the image for analysis.

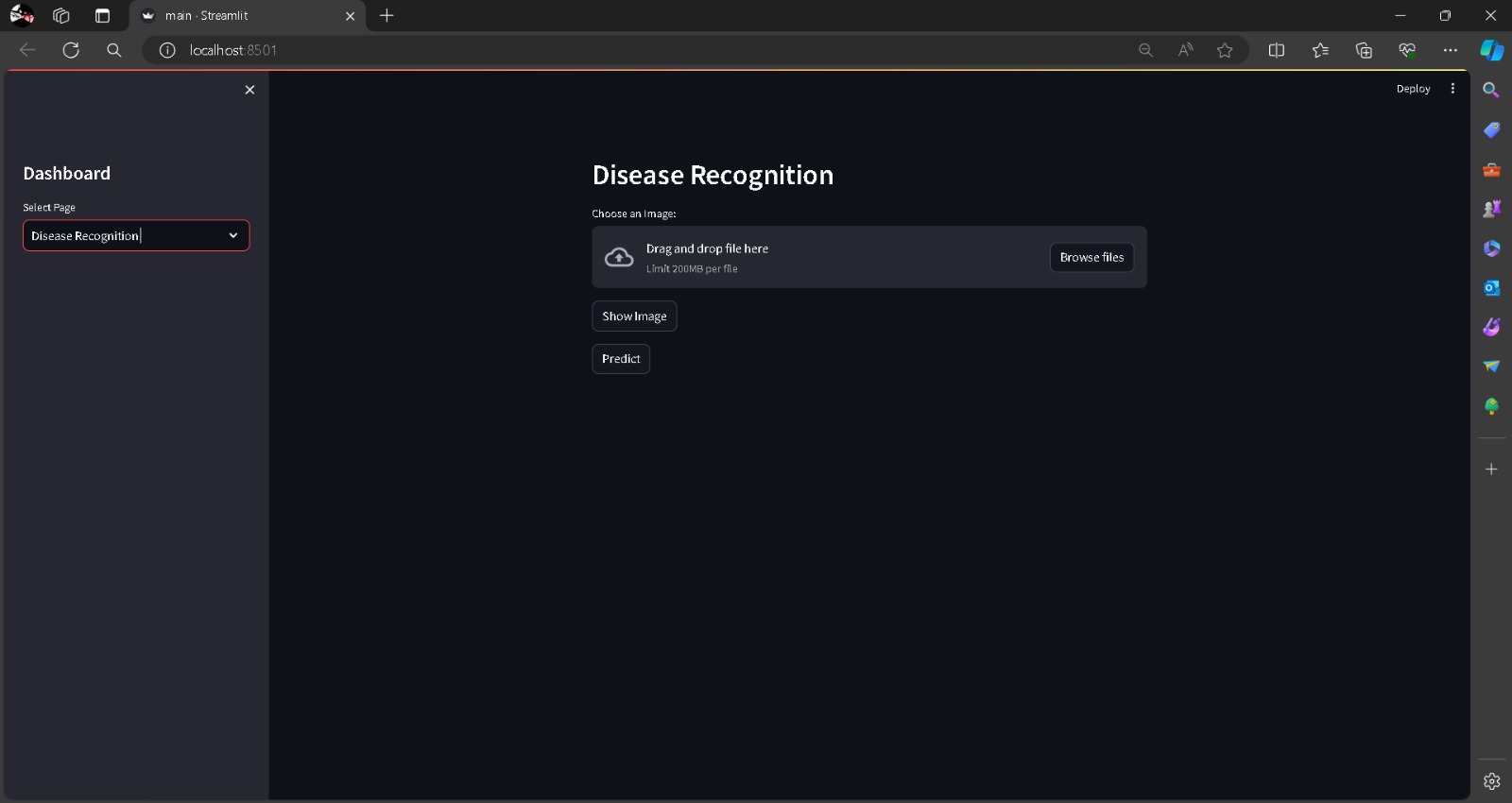


Fig4.12 Image Upload Section

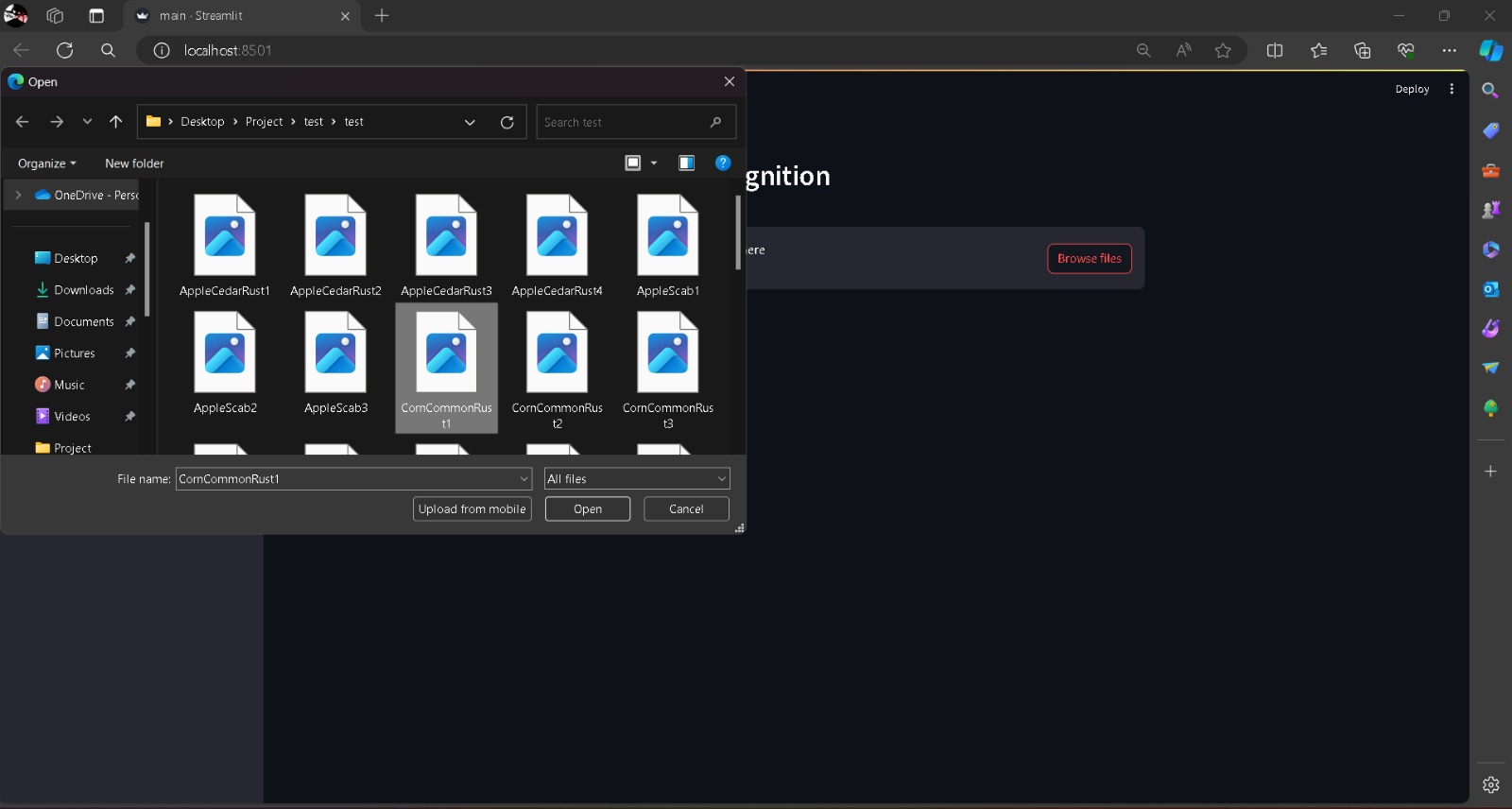


Fig. 4.13

**3.** **Disease Diagnosis and Result Display**

Purpose:

* Provide users with the diagnosis results of the uploaded crop images.

Features:

* Loading Indicator: Shows the progress of the analysis.
* Diagnosis Result: Display the identified disease with a confidence score.

User Flow:

* After submitting an image, the system analyzes the image.
* The result is displayed .

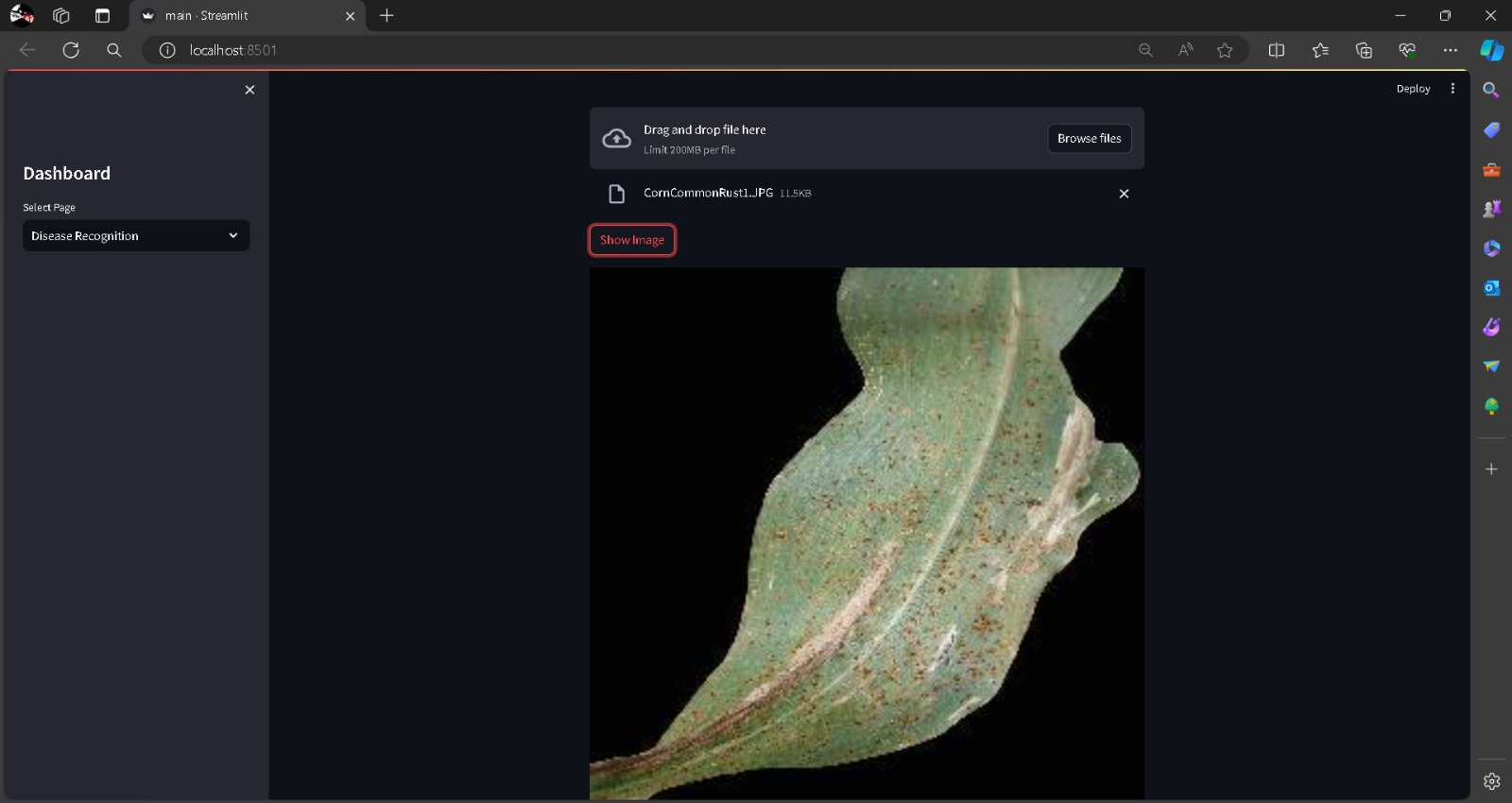


Fig.4.14 Disease Diagnosis and Result Display



Fig4.15

**Chapter 5**

**RESULT AND CONCLUSIONS**

The results of experiments in plant disease detection using machine learning are typically presented by reporting various performance metrics and evaluation outcomes.

These metrics provide insights into an effectiveness and reliability of an trained models.

* 1. **Performance Metrics:**
  + **Accuracy:** an percentage of correctly classified instances (both diseased and healthy plants) out of an total instances.
  + **Precision:** an ratio of true positive predictions to an total number of positive predictions, indicating an model's ability to avoid false positives.
  + **Recall (Sensitivity):** an ratio of true positive predictions to an total number of actual positive instances, reflecting an model's ability to capture all positive instances.
  + **F1-score:** an harmonic mean of precision and recall, providing an balanced measure of an model's performance.

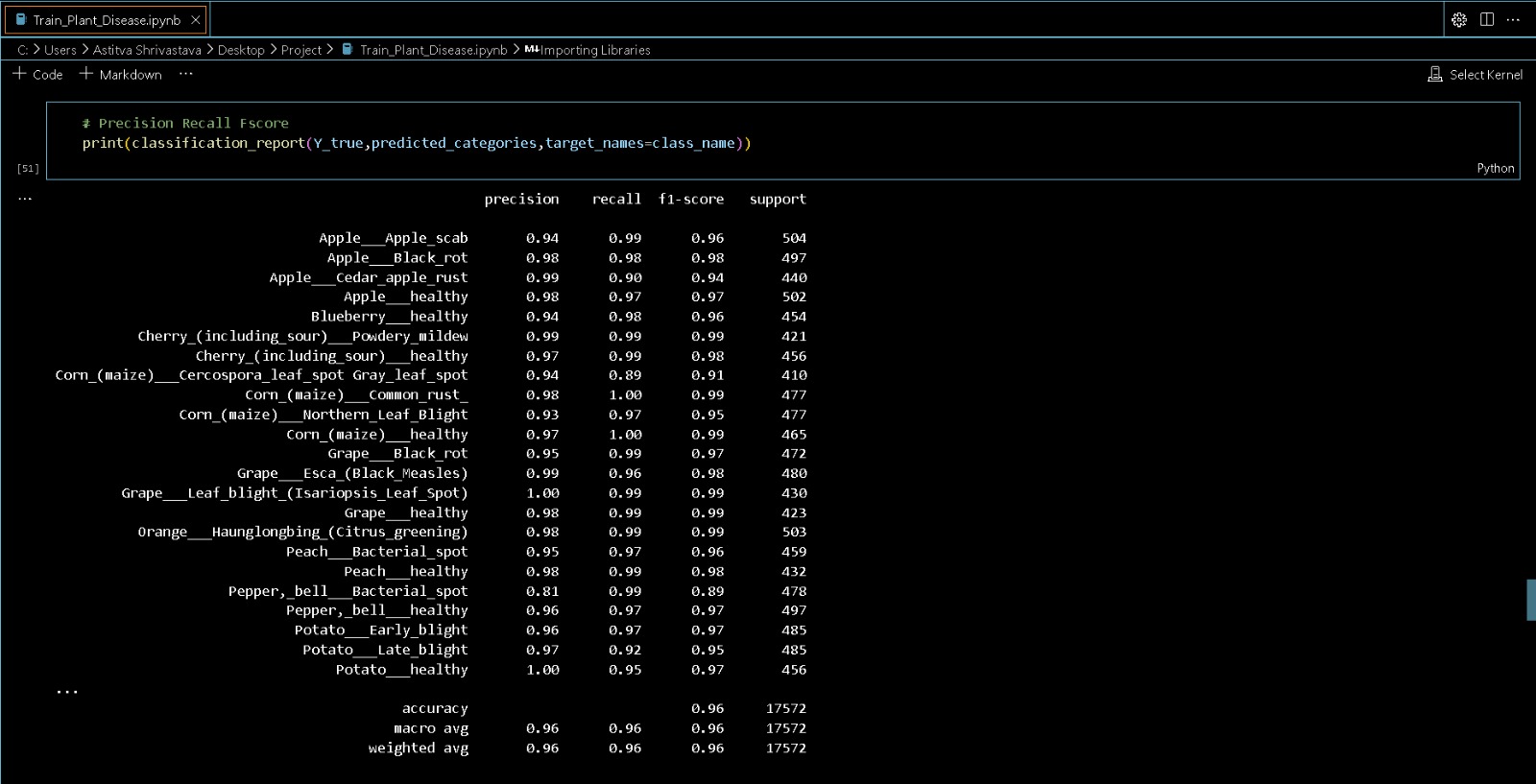


Table 5.1Performance matrix

* 1. **Evaluation Outcomes:**
* **Confusion Matrix:** an table presenting an actual versus predicted class labels, allowing for a detailed analysis of true positives, true negatives, false positives, and false negatives.
* **ROC Curve:** an graphical representation of an true positive rate (sensitivity) against an false positive rate (1-specificity) at various threshold settings, illustrating an model's performance across different decision thresholds.
* **Precision-Recall Curve**: an graphical representation of precision against recall at various threshold settings, providing insights into an trade-off between precision and recall.

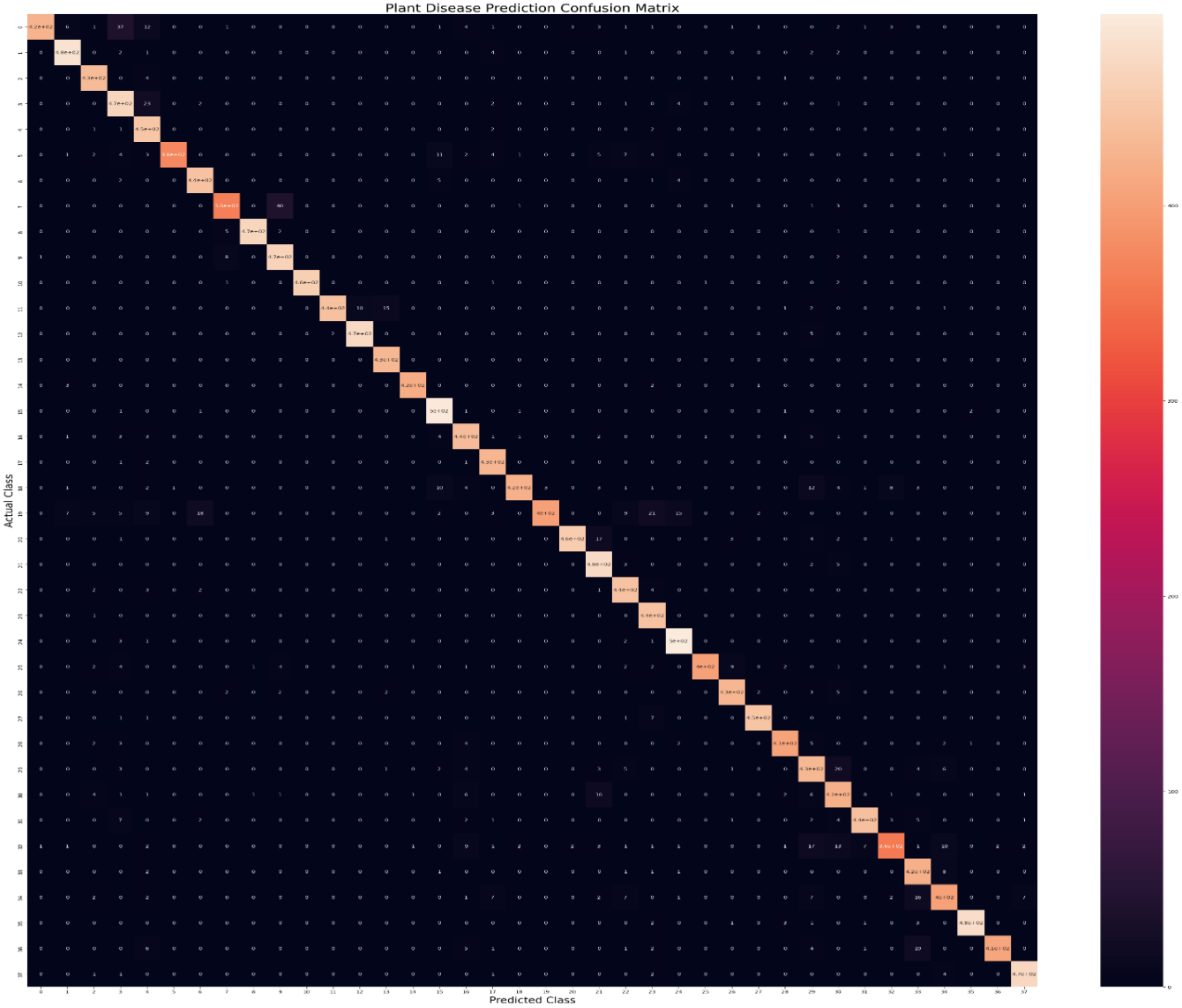
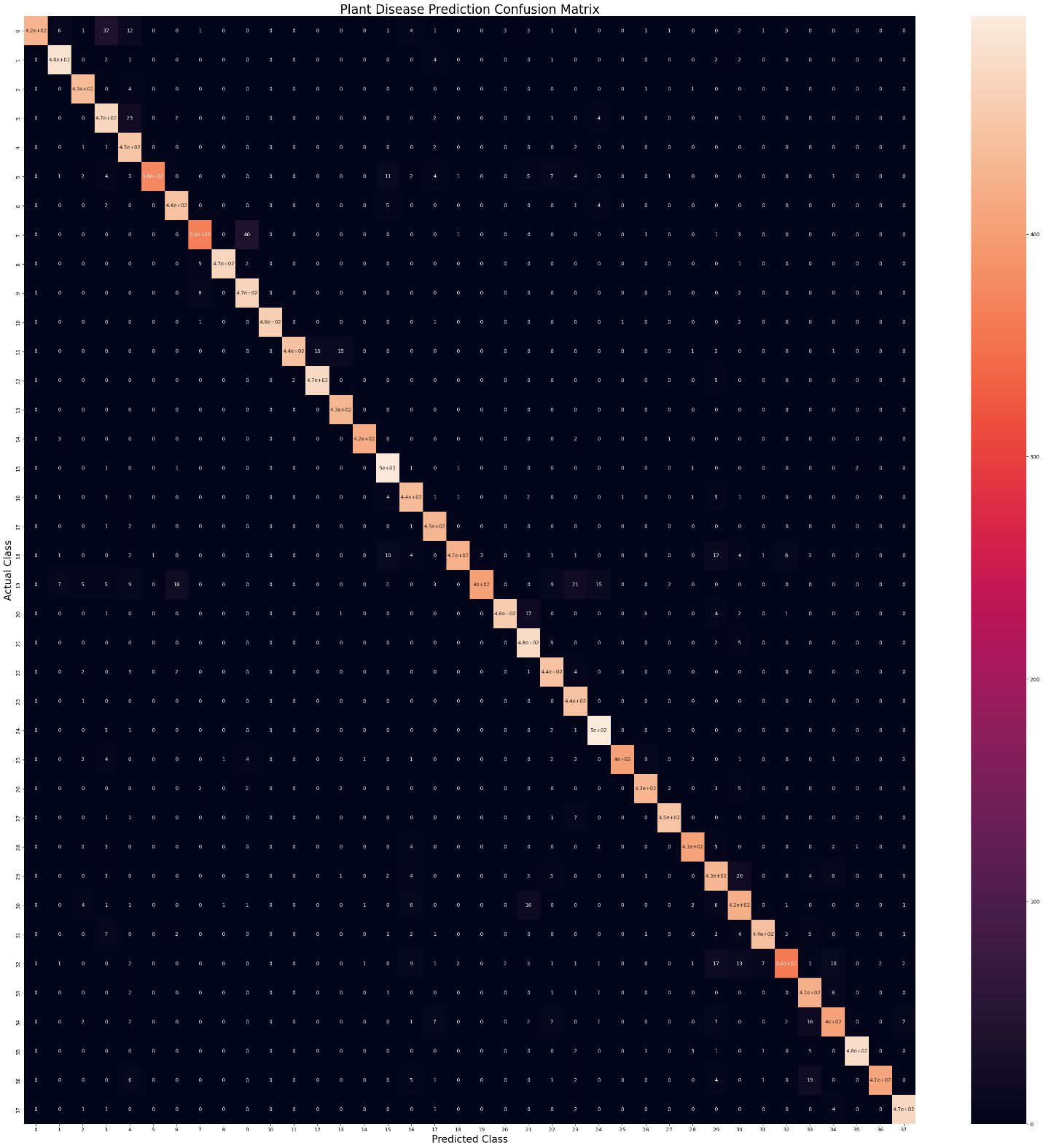


Fig.5.2

**Top of Form**

**5.3 Final testing of model:**

For final testing we take an unknown picture of an infected plant and try to predict the disease using an model.

* Firstly we created an new notebook and import all libraries, load the saved model and visualize for the image.
* Now, we pre-process the image by resizing it and converting it into a batch.
* Now perform a prediction on the image using the model and display the prediction result.

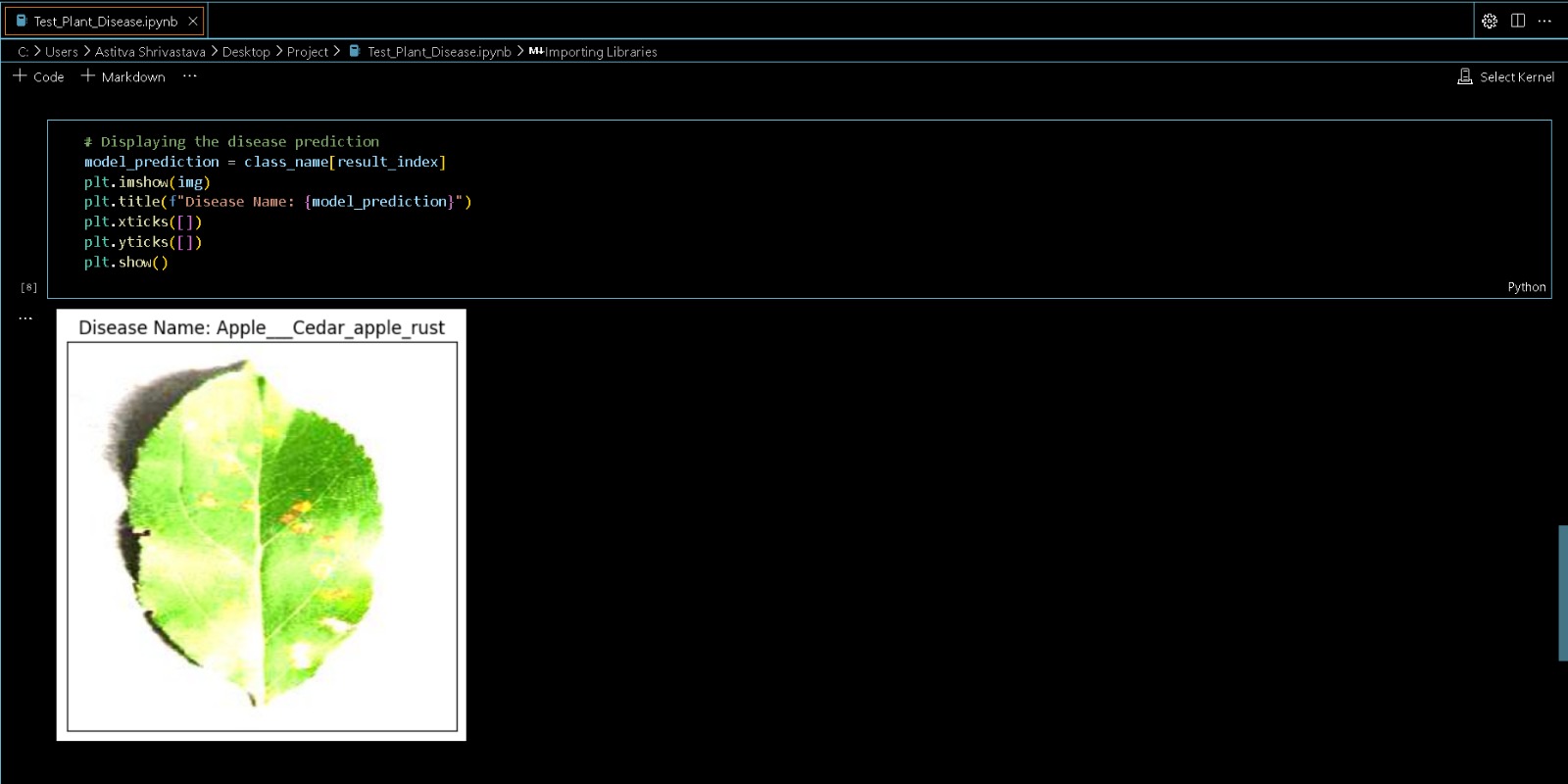


Fig. 5.3Final testing model

**Chapter 6**

**FUTURE ENHANCEMENTS**

As technology advances and user requirements evolve, various potential enhancements can be explored for the ML-based crop disease detection system. These improvements aim to boost the system's accuracy, expand its functionalities, and enhance user experience.

**1. Advanced Machine Learning Models**

Objective: Increase the accuracy and dependability of disease identification.

Enhancements:

* *Deep Learning Integration:* Implement advanced deep learning architectures like Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) to improve image analysis capabilities.
* *Transfer Learning:* Use pre-trained models on extensive and diverse datasets to enhance performance with limited agricultural data.
* *Continuous Learning:* Develop a system that continually learns from new data, ensuring it remains updated with emerging disease patterns and variations.

**2. Expanded Disease Database**

Objective: Broaden the system’s scope to include more crop types and diseases.

Enhancements:

* *Collaborative Data Collection*: Partner with agricultural institutions, universities, and research centers to gather extensive and diverse datasets.
* *Crowdsourced Data:* Allow users to contribute images and data on new diseases, enriching the database.
* *Automated Data Augmentation:* Use techniques to generate synthetic data, expanding the dataset and improving model training.

**3. Multi-Language Support**

Objective: Make the system accessible to a wider audience across different regions.

Enhancements:

* *Language Localization:* Translate the user interface and notifications into multiple languages.
* *Voice Commands:* Implement voice recognition technology to support users who are not literate or prefer audio instructions.

**4. Integration with IoT Devices**

Objective: Provide real-time, continuous monitoring and data collection.

Enhancements:

* *Smart Sensors:* Integrate IoT sensors that monitor environmental conditions such as temperature, humidity, and soil moisture, providing additional data for disease prediction.
* *Drones and UAVs:* Utilize drones equipped with cameras and sensors for large-scale monitoring and early detection of diseases across extensive farm areas.
* *Automated Alerts:* Set up automatic alerts based on real-time data from IoT devices, notifying farmers of potential disease outbreaks.

**5. Enhanced User Experience**

Objective: Improve the usability and functionality of the system.

Enhancements:

* *Interactive Tutorials:* Provide guided tutorials and walkthroughs to help new users understand how to use the system effectively.
* *Personalized Dashboards*: Customize dashboards based on user preferences and needs, highlighting the most relevant information.
* *Mobile App Enhancements:* Improve the mobile application for better offline functionality, ensuring users in remote areas can still use the system without constant internet access.

**6. Predictive Analytics and Decision Support**

Objective: Help farmers make informed decisions by predicting potential disease outbreaks and suggesting proactive measures.

Enhancements:

* *Predictive Models:* Develop models that predict disease outbreaks based on historical data, weather conditions, and crop lifecycle stages.
* *Decision Support System (DSS):* Create a DSS that offers tailored advice on crop management, irrigation scheduling, and pest control based on predictive analytics.

**7. Data Analytics and Visualization**

Objective: Provide deeper insights and better understanding of crop health trends.

Enhancements:

* *Advanced Analytics:* Implement advanced data analytics tools that offer insights into disease trends, treatment efficacy, and overall crop health.
* *Customizable Reports:* Allow users to generate customized reports based on specific parameters and timeframes.
* *Interactive Visualizations*: Develop interactive charts and maps that visually represent disease spread, environmental conditions, and treatment outcomes.

**8. Blockchain for Data Security and Transparency**

Objective: Ensure data integrity, security, and transparency in data sharing.

Enhancements:

* *Secure Data Storage*: Use blockchain technology to securely store and manage data, ensuring it is tamper-proof and traceable.
* *Transparent Data Sharing:* Enable transparent and secure data sharing among stakeholders, ensuring data privacy and ownership rights.

**9. Community and Collaboration Features**

Objective: Foster a community of users who can share knowledge and support each other.

Enhancements:

* *Forums and Discussion Boards:* Create online forums where users can discuss issues, share experiences, and offer advice.
* *Expert Consultations:* Provide access to agricultural experts for consultations and second opinions.
* *Collaborative Projects:* Enable users to participate in collaborative research projects and field trials.

**10. Regulatory Compliance and Certifications**

Objective: Ensure the system meets industry standards and regulatory requirements.

Enhancements:

* *Compliance with Agricultural Standards:* Ensure the system complies with local and international agricultural standards and regulations.
* *Certifications:* Obtain relevant certifications for the technology and processes used, enhancing credibility and user trust.

REFERENCES: