**A Major Project Report on**

**PLANT DAMAGE DETECTION USING MACHINE LEARNING ALGORITHM**

Submitted to

Jawaharlal Nehru Technological University, Hyderabad in partial fulfilment of

the academic requirements for the award of the degree

# BACHELOR OF TECHNOLOGY

in

**COMPUTER SCIENCE AND ENGINEERING**

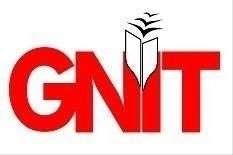
By

|  |  |
| --- | --- |
| **BATHULA SHIRISHA** | **(19831A0519)** |
| **PREETHAM ROY BHUPATHI** | **(19831A0522)** |
| **JADI SANDEEP** | **(19831A0558)** |

Under the esteemed guidance of

# Ch. KOTESWARA RAO

Assistant Professor



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**GURU NANAK INSTITUTE OF TECHNOLOGY**

(Affiliated to JNTUH-Hyderabad)

Ranga Reddy District -501506

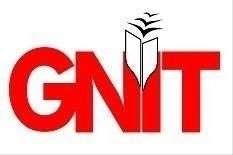
January 2023

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**GURU NANAK INSTITUTE OF TECHNOLOGY**

(Affiliated to JNTUH-Hyderabad)

Ranga Reddy District -501506



**CERTIFICATE**

This is to certify that the project entitled “PLANT DAMAGE DETECTION USING MACHINE LEARNING ALGORITHM” is being presented with report by BATHULA SHIRISHA (19831A0519), PREETHM ROY BHUPATHI (19831A0522), JADI SANDEEP (19831A0558), in partial fulfillment for the award of Degree of Bachelor of Technology in Computer Science and Engineering, to Jawaharlal Nehru Technological University, Hyderabad.

|  |  |
| --- | --- |
| INTERNAL GUIDE | PROJECT COORDINATOR |
| (Mr.Ch. KOTESWARA RAO) | (Ms. M. RADHIKA) |
|  |  |
| HEAD OF DEPARTMENT | PRINCIPAL |
| (Dr. B. SANTHOSH KUMAR) | (Dr. S. SREENATHA REDDY) |

i

**GURU NANAK INSTITUTE OF TECHNOLOGY**

Ibrahimpatnam, R.R. Dist.– 501510

## VISION OF GNIT

To be a world class educational and research institution in the service of humanity by promoting high-quality Engineering and Management Education.

## MISSION OF GNIT

* Imbibe soft skills and technical skills.
* Develop the faculty to reach international standards.
* Maintain high academic standards and teaching quality that promotes critical thinking

and independent judgment.

* Promote research, innovation and Product development by collaboration with reputed

foreign universities.

* Offer collaborative industry programs in emerging areas and the spirit of enterprise.

ii

**GURU NANAK INSTITUTE OF TECHNOLOGY**

Ibrahimpatnam, R.R. Dist. – 501506.

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**VISION**

To be recognized as a leading department of Computer Science and Engineering in the region by students, employers and be known for leadership, Ethics, and commitment to fostering quality teaching–learning, research, and innovation.

**MISSION**

* Nurture young individuals into knowledgeable, skill-full and ethical professionals in their pursuit of Computer Science and Engineering.
* Nurture the faculty to expose them to world-class infrastructure.
* Sustain high performance by excellence in teaching, research and innovations.
* Extensive partnerships and collaborations with foreign universities for technology upgradation.
* Develop Industry-Interaction for innovation and product development.

iii

 **GURU NANAK INSTITUTE OF TECHNOLOGY**

Ibrahimpatnam, R.R. Dist. – 501506.

### **DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Program Educational Objectives (PSO’s)**

PSO-1: Graduates shall have the ability to apply knowledge and technical skills in emerging areas of Computer Science and Engineering for higher studies, research, employability, product development and handle realistic problems.

PSO-2: Graduates shall maintain ethical conduct, sense of responsibility to serve the society and protect the environment

PSO-3: Graduates shall possess academic excellence with innovative insight, soft skills, managerial skills, leadership qualities, knowledge of contemporary issues for successful professional career.

**Program Outcomes (PO’s)**

PO-1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO-2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO-3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet specified needs with appropriate consideration for public health and safety, and the cultural, societal, and environmental considerations.

iv

PO-4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO-5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO-6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO-7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO-8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO-9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO-10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO-11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

v

PO-12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**PROGRAM SPECIFIC OUTCOMES**

PSO1- Professional Skills: The ability to understand the principles and working of computer systems. Students can assess the hardware and software aspects of computer systems.

PSO2- Problem Solving Skills: The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.

PSO3- Successful Career and Entrepreneurship: The ability to employ modern computer languages, environments, and platforms in creating innovative career paths to be an entrepreneur, and a zest for higher studies.

vi

**GURU NANAK INSTITUTE OF TECHNOLOGY**

Ibrahimpatnam, R.R. Dist. – 501506.

### **DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

### **MAPPING WITH PO’S AND PSO’S**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.No. | Content | PO’s | PSO’s |
| 1 | Introduction | PO1, PO2 | PSO1 |
| 2 | Abstract | PO1, PO2 | PSO1 |
| 3 | System Specifications | PO3, PO5 | PSO1 |
| 4 | Architecture Design | PO3, PO4 | PSO1, PSO2 |
| 5 | HDFS CLI Commands | PO1 | PSO3 |
| 6 | Implementation | P09, PO10, PO11 | PSO2, PSO3 |
| 7 | Results | PO4, PO11, PO12 | PSO2, PSO3 |

# vii

# DECLARATION

We hereby declare that the mini project report entitled “PLANT DAMAGE DETECTION USING MACHINE LEARNING ALGORITHM” is the work done by BATHULA SHIRISHA, PREETHAM ROY BHUPATHI, JADI SANDEEP, bearing the roll no’s 19831A0519, 19831A0522, 19831A0558, towards the fulfilment of the requirement for the award of the Degree of Bachelor of Technology in Computer Science and Engineering, to Jawaharlal Nehru Technological University, Hyderabad, is the result of the work carried out under the guidance Mr.Ch. KOTESHWARA RAO, Guru Nanak Institute of Technology, Hyderabad.

We further declare that this project report has not been previously submitted either in part or full for the award of any degree or diploma by any organization or university.

BATHULA SHIRISHA

PREETHAM ROY BHUPATHI

JADI SANDEEP

viii

# ACKNOWLEDGEMENT

“Task successful” makes everyone happy. But the happiness will be gold without glitter if we didn’t state the persons who have supported us to make it a success.

We would like to express our sincere thanks and gratitude to our Principal, Dr. S. SREENATHA REDDY and Head of the Department Dr. B. SANTHOSH KUMAR, Department of Computer Science and Engineering, Guru Nanak Institute of Technology for having guided me in developing the requisite capabilities for taking up this project.

We thank Project Coordinator CSE, GNIT for providing seamless support and right suggestions that are given in the development of the project.

We specially thank our internal guide Mr. Ch. KOTESWARA RAO for his constant guidance in every stage of the project. We would also like to thank all our lecturers for helping me in every possible way whenever the need arose.

On a more personal note we thank our beloved parents and friends for their moral support during the course of our project.

ix

**ABSTRACT:**

Agriculture accepts a basic part by virtue of the quick improvement of the general population and extended interest in food in India. Hence, it is required to increase harvest yield. One serious cause of low collect yield is an infection brought about by microorganisms, infection, and organisms. Plant disease investigation is one of the major and essential tasks in the part of cultivating. It tends to be forestalled by utilizing plant disease detection techniques. To monitor, observe or take care of plant diseases manually is a very complex task. It requires gigantic proportions of work, and moreover needs outrageous planning time; consequently, image processing is utilized to distinguish diseases of plants. Plant disease classification can be done by using machine learning algorithms which include steps like dataset creation, load pictures, pre-preparing, segmentation, feature extraction, training classifier, and classification. The main objective of this research is to construct one model, which classifies the healthy and diseased harvest leaves and predicts diseases of plants. In this paper, the researchers have trained a model to recognize some unique harvests and 26 diseases from the public dataset which contains 54,306 images of the diseases and healthy plant leaves that are collected under controlled conditions. This paper worked on the ResNets algorithm. A residual neural network (ResNet) is a subpart of the artificial neural network (ANN). ResNet algorithm contains a residual block that can be used to solve the problem of vanishing/exploding gradient. ResNet algorithm is also used for creating Residual Network. For the image classification, ResNets achieve a much well result. The ResNets techniques applied some of the parameters like scheduling learning rate, gradient clipping, and weight decay. Using the ResNet algorithm, the researchers expect high accuracy results and detecting more diseases from the various harvests.

**Plant Disease Detection and Classification Using Machine Learning Algorithm**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT**  **LIST OF FIGURES**  **LIST OF SYMBOLS** | i  v  vii |
| 1. | **CHAPTER 1 : INTRODUCTION**   * 1. GENERAL   2. SCOPE OF THE PROJECT   3. OBJECTIVE   1.4 EXISTING SYSTEM  1.4.1 EXISTINGSYSTEM DISADVANTAGES  1.5 LITERATURE SURVEY  1.6 PROPOSED SYSTEM  1.6.1 PROPOSED SYSTEM ADVANTAGES |  |
| 2. | **CHAPTER 2 :PROJECT DESCRIPTION**  2.1 GENERAL  2.2 METHODOLOGIES  2.2.1 MODULES NAME  2.2.2 MODULES EXPLANATION  2.3 TECHNIQUE OR ALGORITHM |  |
| 3. | **CHAPTER 3 : REQUIREMENTS**  3.1 General  3.2 Hardware REQUIREMENTS  3.3 Software REQUIREMENTS |  |
| 4. | **CHAPTER 4 :SYSTEM DESIGN**  **4.1 general**  **4.2 uml diagrams**  4.2.1 USE CASE DIAGRAM  4.2.2 CLASS DIAGRAM  4.2.3 OBJECT DIAGRAM  4.2.4 STATE DIAGRAM  4.2.5 activity diagram  4.2.6 SEQUENCE DIAGRAM  4.2.7 COLLABORATION DIAGRAM  4.2.8 COMPONENT DIAGRAM  4.2.9 DATA FLOW DIAGRAM  4.2.10 DEPLOYMENT DIAGRAM  4.2.11 SYSTEM ARCHITECTURE |  |
| 5. | **CHAPTER 5 : DEVELOPMENT TOOLS**  5.1 general  5.2 History of Python  5.3 Importance of Python  5.4 Features of Python  5.5 Libraries used in python |  |
| 6. | **CHAPTER 6 :IMPLEMENTATION**  6.1 GENERAL  6.2 IMPLEMENTATION |  |

|  |  |  |
| --- | --- | --- |
| 7. | **CHAPTER 7 :SNAPSHOTS**  7.1 GENERAL  7.2 VARIOUS SNAPSHOTS |  |
| 8. | **CHAPTER 8 :SOFTWARE TESTING**  8.1 GENERAL  8.2 DEVELOPING METHODOLOGIES  8.3 TYPES OF TESTING |  |
| 9. | **CHAPTER 9 :**  **FUTURE ENHANCEMENT**  9.1 FUTURE ENHANCEMENTS |  |
| **10** | **CHAPTER 10 :**  10.1CONCLUSION  10.2 REFERENCES |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **NAME OF THE FIGURE** | **PAGE NO.** |
|  |  |  |
| 4.1 | Use case Diagram |  |
| 4.2 | Class diagram |  |
| 4.3 | Object diagram |  |
| 4.4 | State Diagram |  |
| 4.5 | Activity Diagram |  |
| 4.6 | Sequence diagram |  |
| 4.7 | Collaboration diagram |  |
| 4.8 | Component Diagram |  |
| 4.9 | Data flow diagram |  |
| 4.10 | Deployment Diagram |  |
| 4.11 | Architecture Diagram |  |

**LIST OF SYSMBOLS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NOTATION**  **NAME** | **NOTATION** | **DESCRIPTION** |
| 1. | Class |  | Represents a collection of similar entities grouped together. |
| 2. | Association | name | Associations represents static relationships between classes. Roles represents the way the two classes see each other. |
| 3. | Actor |  | It aggregates several classes into a single classes. |
| 4. | Aggregation | Interaction between the system and external environment |

|  |  |  |  |
| --- | --- | --- | --- |
| 5. | Relation  (uses) | uses | Used for additional process communication. |
| 6. | Relation  (extends) | extends | Extends relationship is used when one use case is similar to another use case but does a bit more. |
| 7. | Communication |  | Communication between various use cases. |
| 8. | State |  | State of the processes. |
| 9. | Initial State |  | Initial state of the object |
| 10. | Final state |  | Final state of the object |
| 11. | Control flow |  | Represents various control flow between the states. |
| 12. | Decision box |  | Represents decision making process from a constraint |
| 13. | Use case |  | Interact ion between the system and external environment. |

|  |  |  |  |
| --- | --- | --- | --- |
| 14. | Component |  | Represents physical modules which are a collection of components. |
| 15. | Node |  | Represents physical modules which are a collection of components. |
| 16. | Data Process/State |  | A circle in DFD represents a state or process which has been triggered due to some event or action. |
| 17. | External entity |  | Represents external entities such as keyboard, sensors, etc. |
| 18. | Transition |  | Represents communication that occurs between processes. |
| 19. | Object Lifeline |  | Represents the vertical dimensions that the object communications. |
| 20. | Message | Message | Represents the message exchanged. |

**CHAPTER-1**

**INTRODUCTION**

**1.1 INTRODUCTION:**

Agribusiness has grown to be a crucial supply of financial improvement and about 80% of the population depends on farming in India. Rancher chooses the proper harvest dependent on the type of soil, climate circumstance of the place, and financial value. In most cases, farmers commit suicide due to production loss because they are not able to pay the bank loans taking for farming purposes. We have noticed in present times that the climate is changing persistently which is harmful to the crops and leading farmers towards debts and suicide. These risks can be minimized when various mathematical or statistical methods are applied to data and by using these methods, we can identify the disease of plants to the farmer for his agriculture land so that it helps him to get maximum profit.

Due to the growing population, changes in climate, and weakness in governmental issues, the agricultural businesses initiated searching for new methods to increase the production of food. Researchers are trying to develop new capable and unique technologies for generating high-efficiency results. In this methodology, one of the cnn related technique is utilized for plant diseases detection and image processing technique is utilized for plant diseases classification. In the cnn learning method is InceptionV3, used for detection of plant disease training & testing and InceptionV3 & image processing technique, used to understand user input related data to give detection & classification for plants.

**1.2 SCOPE OF THE PROJECT**

The goal of this project is to develop a system that can instantly identify plant disease detection as well as at a same time to classify what type of plant it is. We will employ neural network methods. Via a through cnn techniques like InceptionV3, the study will show highest accuracy detection’s for two types at a time.

**1.3 OBJECTIVE**

This project's primary goal is to create a reliable and effective system for classifying and detecting plant diseases using CNN (Convolutional Neural Network) method, specifically InceptionV3. The device aims to classify the type of afflicted plant while also instantly and accurately identifying plant illnesses. The ultimate objective is to help farmers identify plant diseases in a timely manner so that production losses may be avoided and the danger of farmer suicide due to crop loss debts can be reduced. In order to efficiently analyse user input data and deliver reliable detection and classification results for different plant varieties, the system also seeks to make use of image processing techniques. The system attempts to use advanced neural network technique to contribute to improving food production efficiency and addressing the challenges posed on agricultural issues in India.

**1.4 EXISTING SYSTEM:**

* Crop disease is a serious concern for safety of food, but its fast detection still remains difficult in different parts of the world because of the lack of proper infrastructure. Automatic identification of plant diseases is necessary for food security, yield loss estimation and management of disease. With the worldwide increase in digital cameras and continuous improvement in computer vision domain, the automated techniques for detection of disease are highly in demands in precision agriculture, highly productive plant phenotype, smart greenhouse and much more.
* Working on an open dataset which includes 15200 images of crop leaves, a Residual Network (ResNet34) was trained to perform this task of classification. The proposed ResNet34 model accomplished a highest accuracy on a test set, illustrating the viability of the proposed model.

**1.4.1 EXISTINGSYSTEM DISADVANTAGES:**

* Increased complexity of architecture.
* Implementation of Batch normalization layers since ResNet heavily depends on it.
* Adding skip level connections for which you have take into account the dimensionality between the different layers which can become a problem.

**1.5 LITERATURE SURVEY**

**Title:** A review on machine learning classification techniques for plant disease detection **Author:** Shruthi, U., V. Nagaveni, and B. K. Raghavendra

**Year:** 2019.

**Description:** In India, Agriculture plays an essential role because of the rapid growth of population and increased in demand for food. Therefore, it needs to increase in crop yield. One major effect on low crop yield is disease caused by bacteria, virus and fungus. It can be prevented by using plant diseases detection techniques. Machine learning methods can be used for diseases identification because it mainly apply on data themselves and gives priority to outcomes of certain task. This paper presents the stages of general plant diseases detection system and comparative study on machine learning classification techniques for plant disease detection. In this survey it observed that Convolutional Neural Network gives high accuracy and detects more number of diseases of multiple crops.

**Title:** Plant disease classification using soft computing supervised machine learning.

**Author:** Sehgal, Aman, and Sandeep Mathur.

**Year:** 2019.

**Description**: Plants are always concerned about the diseases introduced by pathogens For example infections, microorganisms and parasites in the plant bodies. It is globally recognized that, pathogens tends to cause huge yield misfortunes. Various researchers have explored how to diminish the harmfulness of pathogens in plants. A few analysts have explored some opposition qualities in plants and attempts to improve the obstruction of plants to pathogens. Meanwhile, different analysts have created ID and s coring framework for monitoring and examining the advancement or quality and also by anticipating the infection bolstered leaves. The reason for this Review work is to display the use of AI in the revelation of plant opposition.

**Title:** Image processing techniques for detecting and classification of plant disease: a review

**Author:** Hungilo, Gilbert Gutabaga, Gahizi Emmanuel, and Andi WR Emanuel

**Year:** 2019.

**Description:** Agriculture is the main contributor to Tanzania Economy. Apart from climate change, disease acts as one of contributing factors which results in the poor production of the most important staple foods like maize and cassava. This leads to economic loss and food insecurity in the area. Preventive action is needed for early detection of the diseases. Image processing techniques to detect disease on plant leaves can be a promising solution to the farmer. The current way of detecting disease using naked eyes done by an expert is a time-consuming and cumbersome task to implement in a large farm. This paper presents a survey of current studies in the area of image processing, by checking techniques used to detect disease on plants leaves or fruits and machine learning model used to classify the disease. The main aim of the paper is to show the current state of the art and clarify step taken during the image processing stage and check merit and demerit of each technique used also the performance of the machine learning model used to classify the disease. This review paper will be of important to other researchers working in the area of image processing for detecting and classification of plant -- leaves/fruit diseases to know the current state of the art in the field.

**Title:**  Automated plant disease analysis (APDA): performance comparison of machine learning techniques

**Author:** Akhtar, Asma, Aasia Khanum, Shoab A. Khan, and Arslan Shaukat

**Year:** 2013

**Description**: Plant disease analysis is one of the critical tasks in the field of agriculture. Automatic identification and classification of plant diseases can be supportive to agriculture yield maximization. In this paper we compare performance of several Machine Learning techniques for identifying and classifying plant disease patterns from leaf images. A three-phase framework has been implemented for this purpose. First, image segmentation is performed to identify the diseased regions. Then, features are extracted from segmented regions using standard feature extraction techniques. These features are then used for classification into disease type. Experimental results indicate that our proposed technique is significantly better than other techniques used for Plant Disease Identification and Support Vector Machines outperforms other techniques for classification of diseases.

**Title:** Plant disease classification using image segmentation and SVM techniques.

**Author**: Elangovan, K., and S. Nalini

**Year:** 2017**.**

**Description:** For preventing the losses in the yield and quantity of the agricultural product, Classification is performed, if proper analysis is not taken in this approach or classification, then it produces serious effects on plants and due to which respective product quality or productivity is affected. Disease classification on plant is very critical for supportable agriculture. It is very difficult to monitor or treat the plant diseases manually. It requires huge amount of work, and also need the excessive processing time, therefore image processing is used for the detection of plant diseases. Plant disease classification involves the steps like Load image, pre-processing, segmentation, feature extraction, svm Classifer.

**1.6 PROPOSED SYSTEM**

The main objective of this research is simple and clear which is to build a model that can be classified between healthy and unhealthy harvest leaves and also if the crop has any disease, predict which disease is it. This paper worked on 54,306 images which include unhealthy and healthy plant leaves images and trained a model with the help of a convolution neural network to recognize different crop species, different diseases, and various classes. This trained model achieves accuracy greater on a held-out test set. In this methodology, a InceptionV3 Architecture model are used. The types of the leaf can be recognized by the first layer and possible diseases that could happen in the plant can be checked by the accompanying layer. Deep learning gives better accuracy results that could help for diagnosis of diseases in crops and analyze down to the smallest unit of an image, a pixel.

**1.6.1 PROPOSED SYSTEM ADVANTAGES:**

* It has a deeper network compared to the Inception V1 and V2 models, but its speed isn't compromised.
* It has higher efficiency.

**CHAPTER 2**

**PROJECT DESCRIPTION**

**2.1 GENERAL:**

This project is aimed at creating a machine learning-based system for plant disease detection and classification. The system will utilize advanced image processing and neural network techniques to achieve high accuracy in disease detection and plant classification. The project aims to benefit the agriculture industry by providing a reliable and efficient tool for early disease detection and prevention, ultimately helping to increase crop yields and food production.

**2.2 METHODOLOGIES**

**2.2.1MODULES NAME:**

* **Dataset**
* **Importing the necessary libraries**
* **Retrieving the images**
* **Splitting the dataset**
* **Building the model**
* **Apply the model and plot the graphs for accuracy and loss**
* **Accuracy on test set**
* **Saving the Trained Model**

**2.2.2 MODULES DESCSRIPTION:**

**1) Dataset:**

In the first module, we developed the system to get the input dataset for the training and testing purpose. Dataset is given in the model folder. The dataset consists of 70,295 Plant images of Apple, Blueberry, Cherry, Corn (maize), Grape, Orange, Peach, Pepper bell, Potato, Raspberry, Soybean, Strawberry and Tomato.

**2) Importing the necessary libraries:**

In the next module, we will be importing the necessary libraries for our plant disease detection system. We will be using Python language for this. First we will import the necessary libraries such as keras for building the main model, sklearn for splitting the training and test data, PIL for converting the images into array of numbers and other libraries such as pandas, numpy, matplotlib and tensorflow.

**3)Retrieving the images:**

We will retrieve the images and their labels. Then resize the images to (224,224) as all images should have same size for recognition. Then convert the images into numpy array.

**4) Splitting the dataset:**

Split the dataset into train and test. 80% train data and 20% test data.

**5) Building the model:**

In this module, we will be building t InceptionV3| CNN model

Inception v3:mainly focuses on burning less computational power by modifying the previous Inception architectures. In comparison to VGGNet, Inception Networks (GoogLeNet/Inception v1) have proved to be more computationally efficient, both in terms of the number of parameters generated by the network and the economic cost incurred (memory and other resources).

**6) Apply the model and plot the graphs for accuracy and loss:**

We will compile the model and apply it using fit function. The batch size will be 10. Then we will plot the graphs for accuracy and loss. We got average validation accuracy of 91.00%.

**7)Accuracy on test set:**

We got an accuracy of 89.00% on test set.

**8) Saving the Trained Model:**

Once you’re confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .h5 or .pkl file using a library like pickle.

Make sure you have pickle installed in your environment.

Next, let’s import the module and dump the model into .h5 file.

**2.3 TECHNIQUE USED OR ALGORITHM USED**

**2.3.1 EXISTING TECHNIQUE: -**

* **SSVEP and EOG**
* A residual neural network (ResNet) is an artificial neural network (ANN). It is a gateless or open-gated variant of the HighwayNet, the first working very deep feedforward neural network with hundreds of layers, much deeper than previous neural networks. Skip connections or shortcuts are used to jump over some layers (HighwayNets may also learn the skip weights themselves through an additional weight matrix for their gates).
* Typical ResNet models are implemented with double- or triple- layer skips that contain nonlinearities (ReLU) and batch normalization in between. Models with several parallel skips are referred to as DenseNets. In the context of residual neural networks, a non-residual network may be described as a plain network.

**2.3.2 PROPOSED TECHNIQUE USED OR ALGORITHM USED:**

* **InceptionV3 Architecture**
* Inception-v3 is a convolutional neural network that is 48 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database [1]. The pretrained network can classify images into 1000 object categories Inception v3 is an image recognition model that has been shown to attain greater accuracy on the ImageNet dataset.
* The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concatenations, dropouts, and fully connected layers. Batch normalization is used extensively throughout the model and applied to activation inputs. Loss is computed using Softmax.

**CHAPTER 3**

**REQUIREMENTS ENGINEERING**

**3.1 GENERAL**

We can see from the results that on each database, the error rates are very low due to the discriminatory power of features and the regression capabilities of classifiers. Comparing the highest accuracies (corresponding to the lowest error rates) to those of previous works, our results are very competitive.

**3.2 HARDWARE REQUIREMENTS**

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

* PROCESSOR : DUAL CORE 2 DUOS.
* RAM : 4GB DD RAM
* HARD DISK : 250 GB

**3.3 SOFTWARE REQUIREMENTS**

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team’s progress throughout the development activity.

* Operating System : Windows 7/8/10
* Platform : Spyder3
* Programming Language : Python
* Front End : HTML, CSS

**3.4 FUNCTIONAL REQUIREMENTS**

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, Firstly, the system is the first that achieves the standard notion of semantic security for data confidentiality in attribute-based deduplication systems by resorting to the hybrid cloud architecture.

**3.5 NON-FUNCTIONAL REQUIREMENTS**

**The major non-functional Requirements of the system are as follows**

**Usability**

The system is designed with completely automated process hence there is no or less user intervention.

**Reliability**

The system is more reliable because of the qualities that are inherited from the chosen platform python. The code built by using python is more reliable.

**Performance**

This system is developing in the high level languages and using the advanced back-end technologies it will give response to the end user on client system with in very less time.

**Supportability**

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is built into the system.

**Implementation**

The system is implemented in web environment using Jupyter notebook software. The server is used as the intellignce server and windows 10 professional is used as the platform. Interface the user interface is based on FLASK provides server system.

**CHAPTER 4**

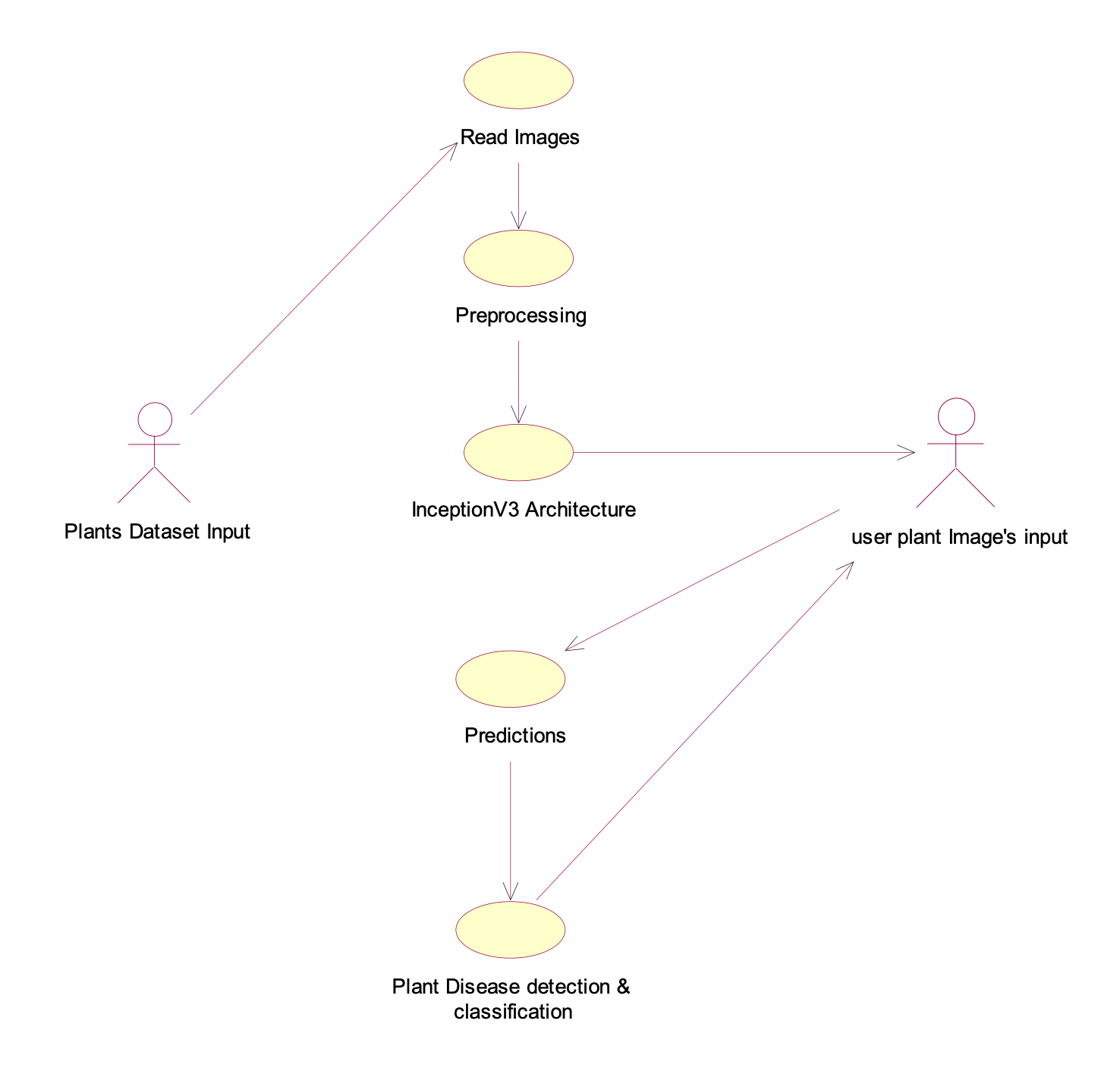
**DESIGN ENGINEERING**

**4.1 GENERAL**

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering.

**4.2 UML DIAGRAMS**

**4.2.1 USE CASE DIAGRAM**



**EXPLANATION:**

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

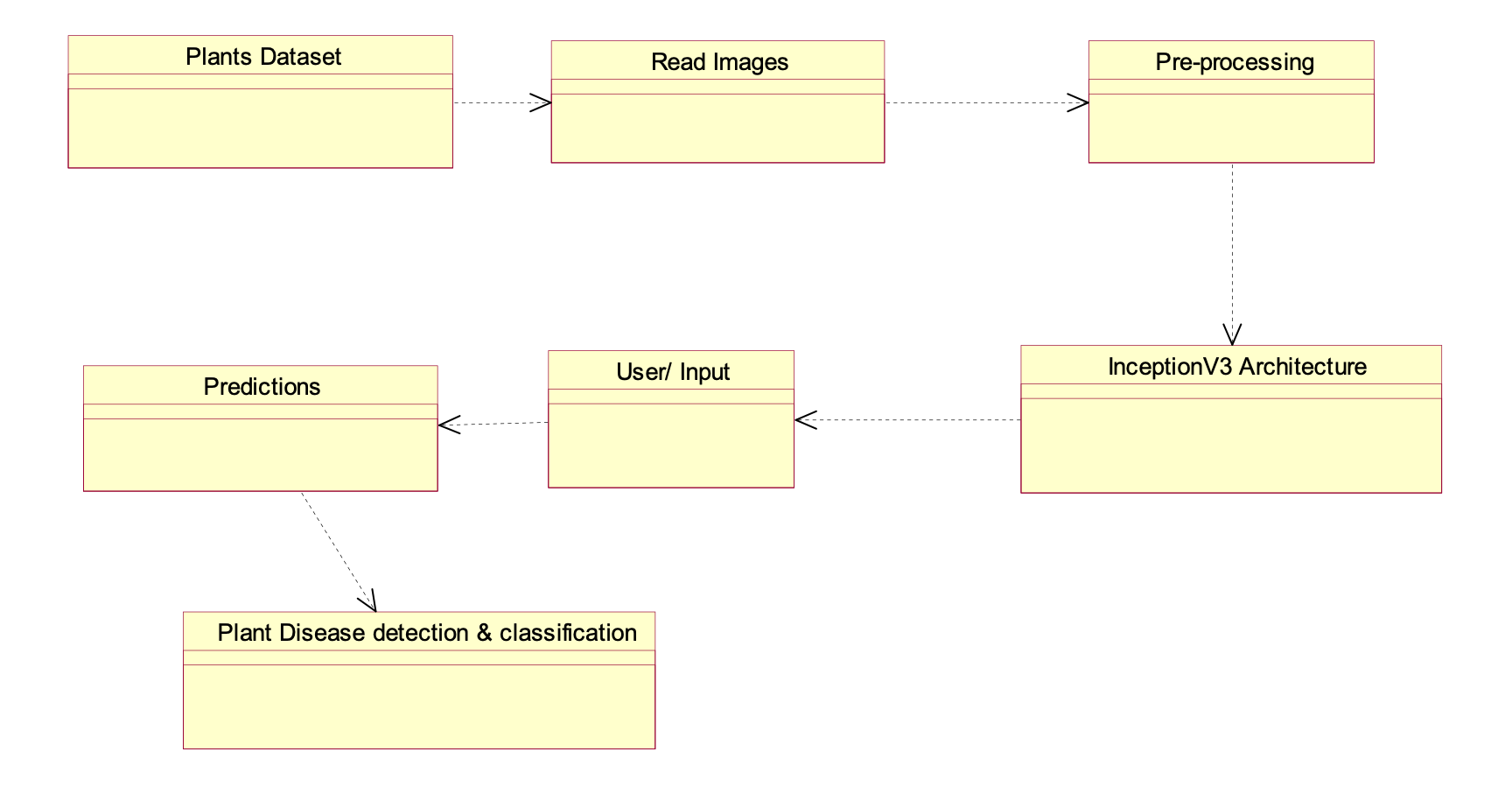
**4.2.2 CLASS DIAGRAM**

****

**EXPLANATION**

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project.

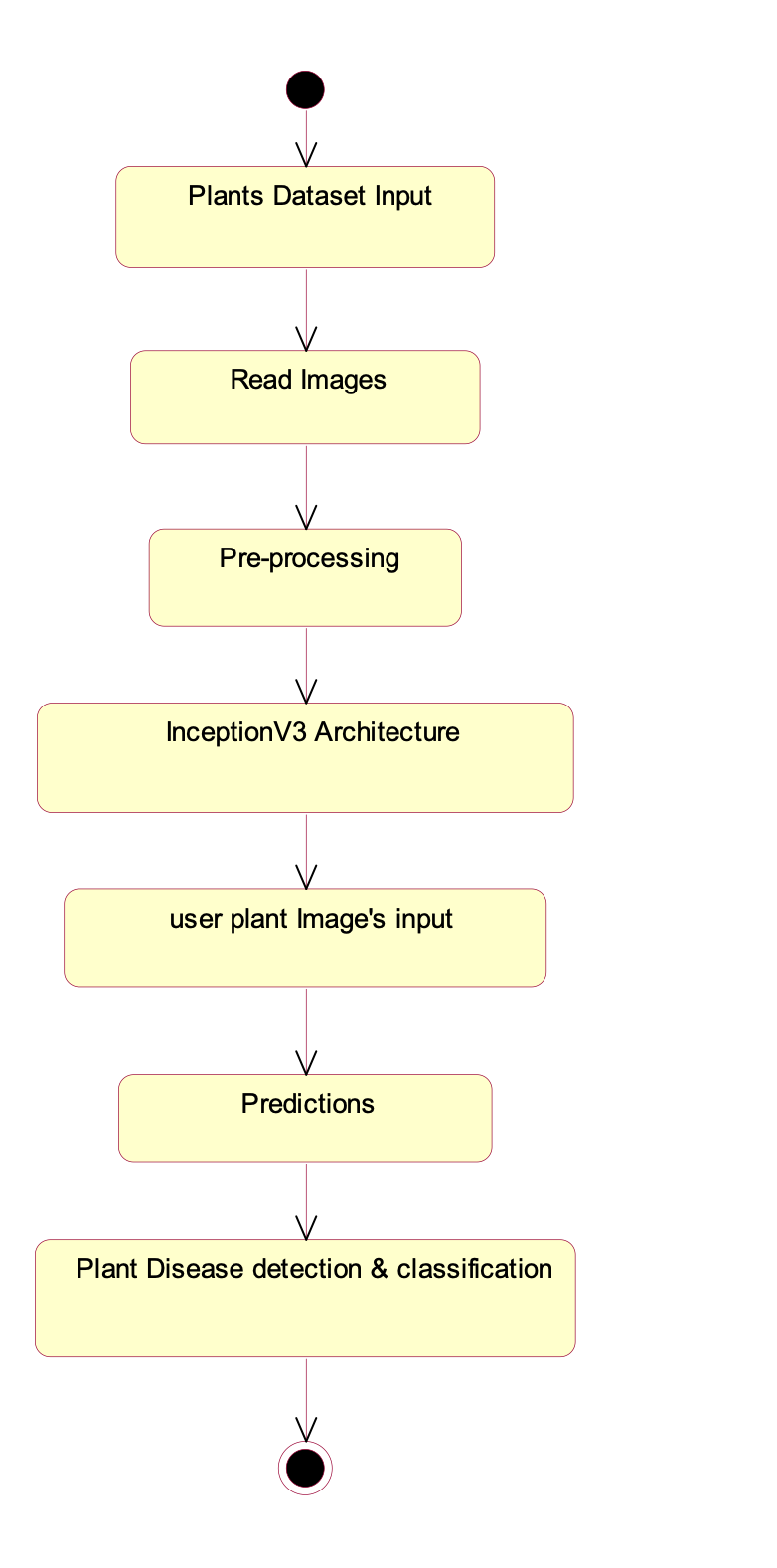
**4.2.3 OBJECT DIAGRAM**



**EXPLANATION:**

In the above digram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

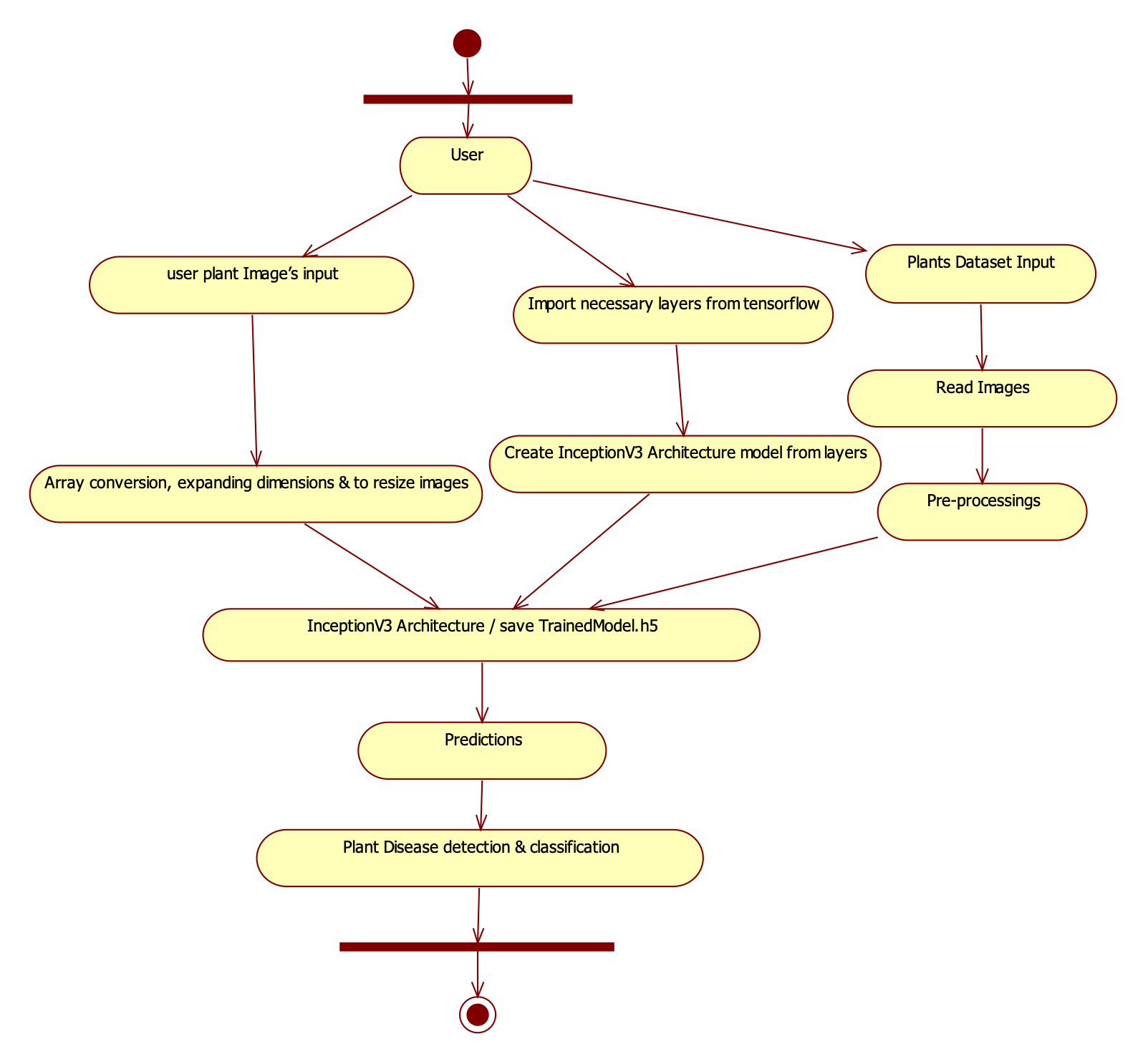
**4.2.4 STATE DIAGRAM**

****

**EXPLANATION:**

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

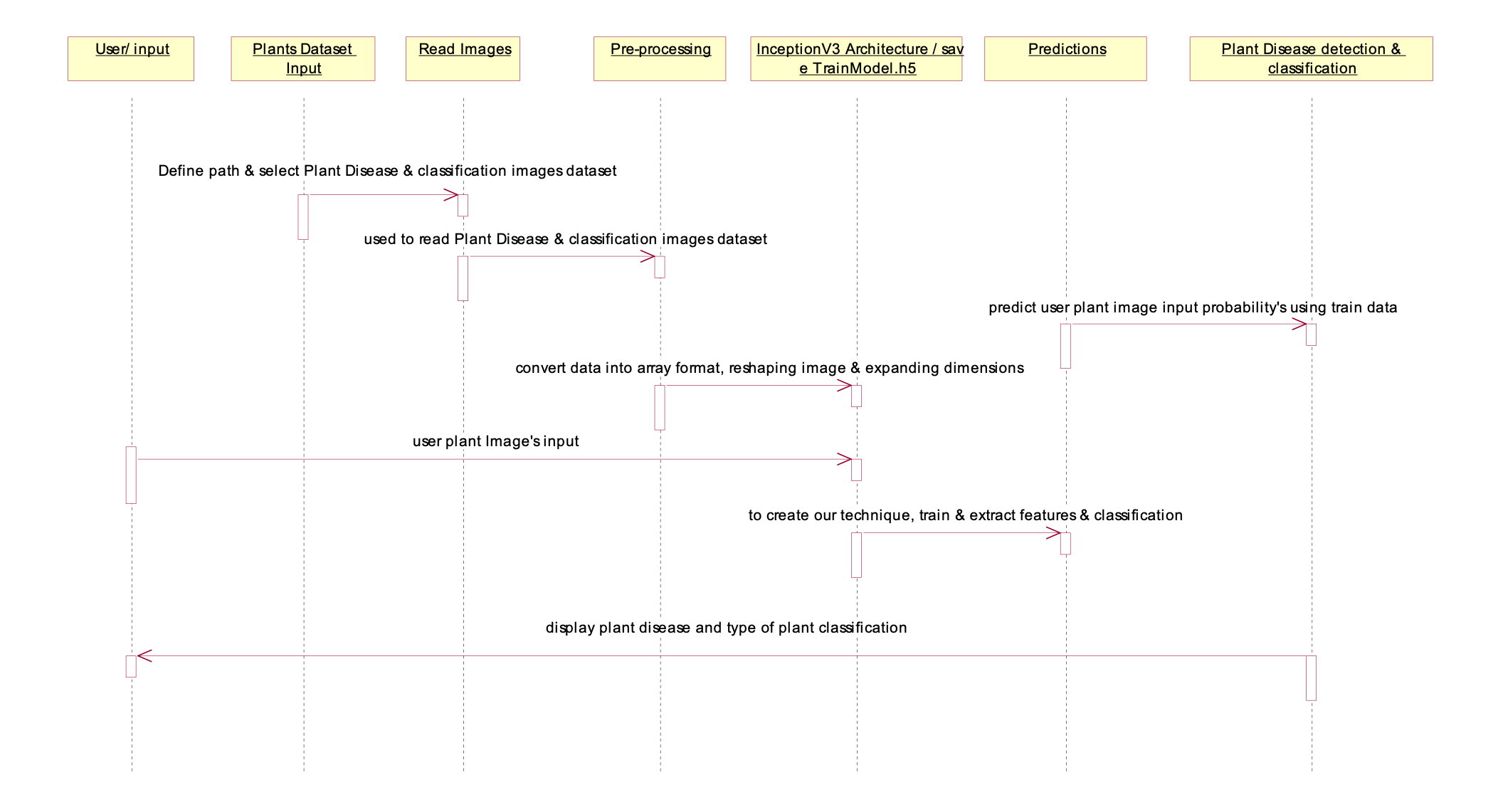
**4.2.5 ACTIVITY DIAGRAM**

****

**EXPLANATION:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

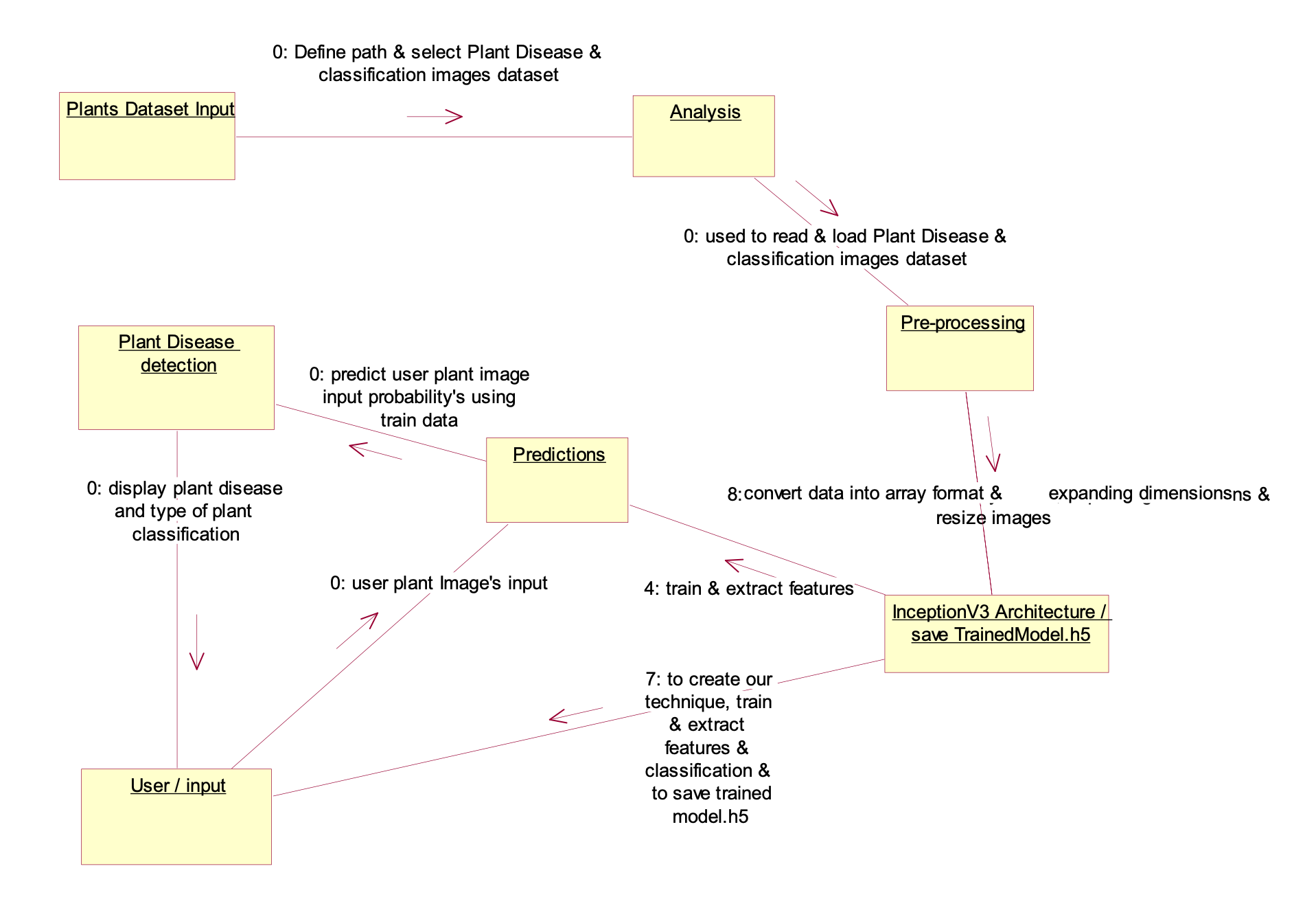
**4.2.6 SEQUENCE DIAGRAM**

****

**EXPLANATION:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

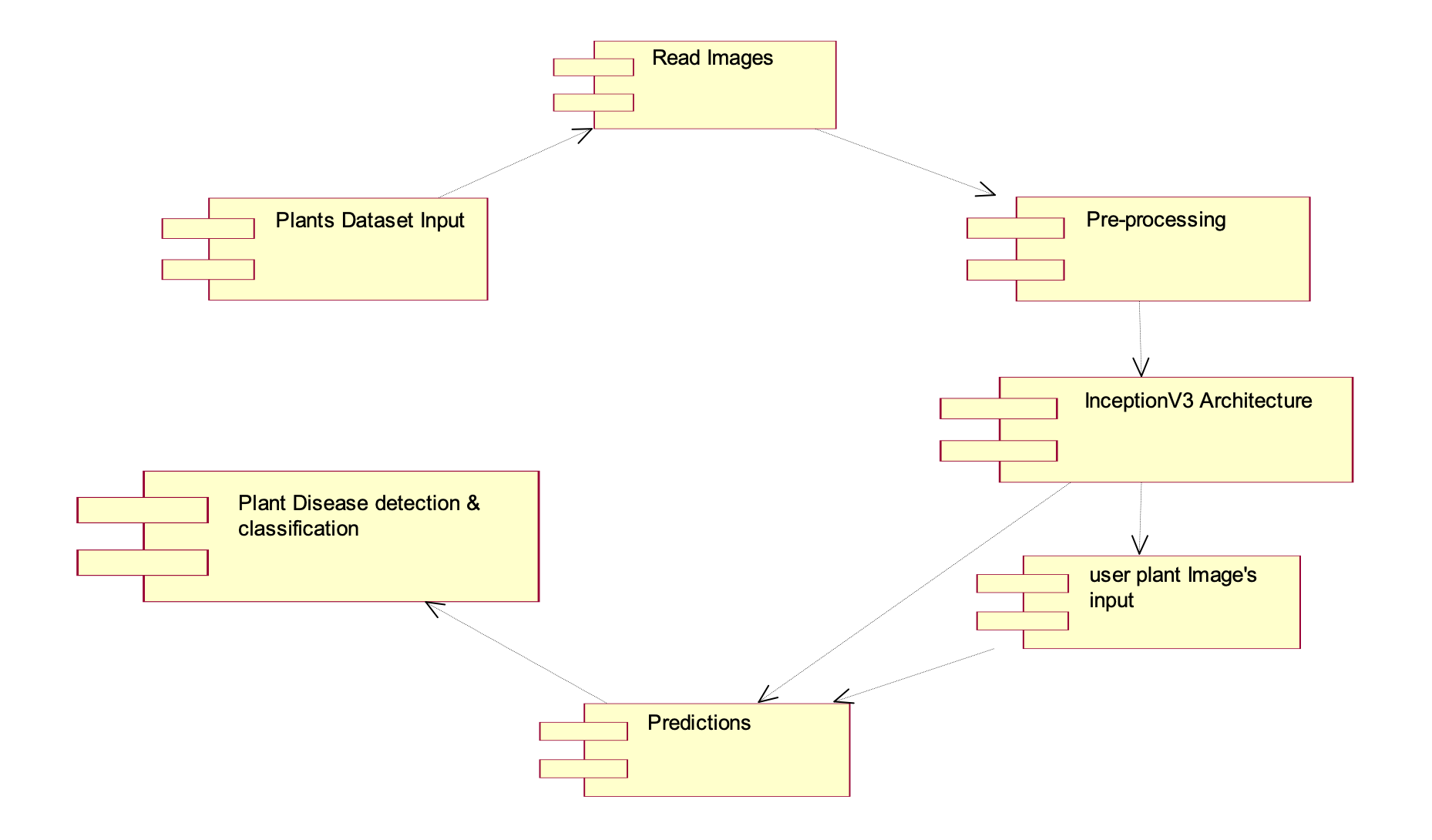
**4.2.7 COLLABORATION DIAGRAM**



**EXPLANATION:**

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

**4.2.8 COMPONENT DIAGRAM**



**EXPLANATION**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

**4.2.9 Data Flow Diagram**

**Level 0:**

Read Images & Pre-processing

Apply InceptionV3 Architecture & save trained file like TrainedModel.h5

Plants Dataset Input

User

**Level 1:**

Create InceptionV3 Architecture using tensor-flow layers

Understand about layers

Import necessary layers from tensorflow

User

**Level 2:**

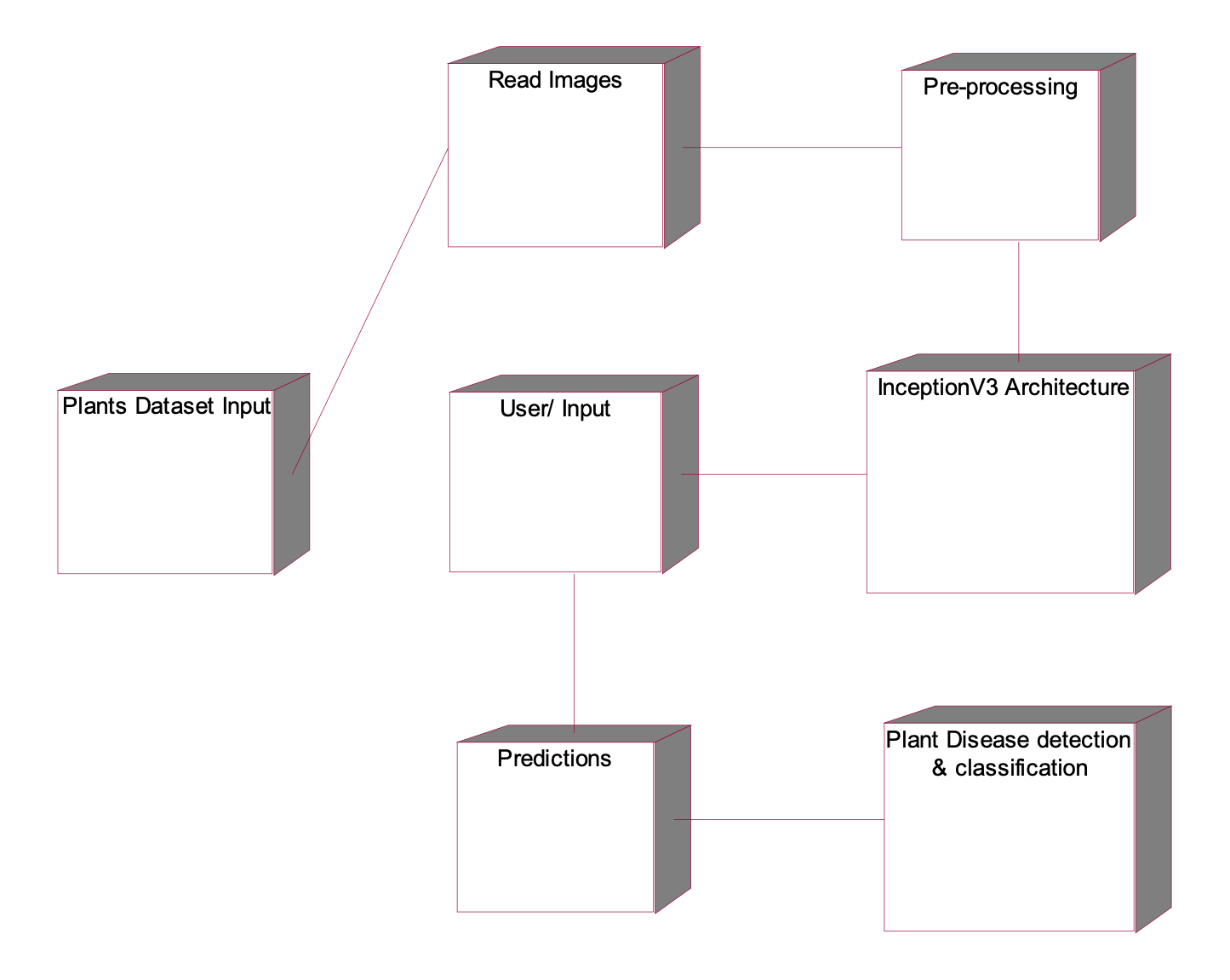
user plant Image’s input

Loads related TrainedModel.h5 & apply converted data

Plant Disease detection & classification

Array conversion, expanding dimensions & resize images

**4.2.10 DEPLOYMENT DIAGRAM**

****

**EXPLANATION:**

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

**SYSTEM ARCHITECTURE:**

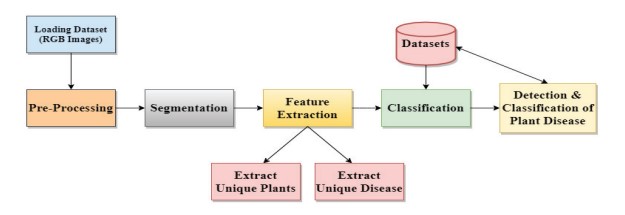
****

Fig 4.11: System Architecture

**CHAPTER 5**

**DEVELOPMENT TOOLS**

**5.1 Python**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

## 5.2 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

#### 5.3 Importance of Python

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

#### 5.4 Features of Python

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* IT supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

**5.5 Libraries used in python**

* numpy - mainly useful for its N-dimensional array objects.
* pandas - Python data analysis library, including structures such as dataframes.
* matplotlib - 2D plotting library producing publication quality figures.
* scikit-learn - the machine learning algorithms used for data analysis and data mining tasks.



Figure : NumPy, Pandas, Matplotlib, Scikit-learn

**CHAPTER 6**

**IMPLEMENTATION**

**6.1 GENERAL**

**Coding:**

from flask import Flask, render\_template, request

import tensorflow as tf

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing import image

from tensorflow.keras.metrics import AUC

import numpy as np

app = Flask(\_name\_)

dependencies = {

'auc\_roc': AUC

}

verbose\_name = {

0:":Apple\_\_\_Apple\_scab",

1:'Apple\_\_\_Black\_rot',

2:'Apple\_\_\_Cedar\_apple\_rust',

3:'Apple\_\_\_healthy',

4:'Blueberry\_\_\_healthy',

5:'Cherry\_(including\_sour)\_\_\_Powdery\_mildew',

6:'Cherry\_(including\_sour)\_\_\_healthy',

7:'Corn\_(maize)\_\_\_Cercospora\_leaf\_spot Gray\_leaf\_spot',

8:'Corn\_(maize)\_\_Common\_rust',

9:'Corn\_(maize)\_\_\_Northern\_Leaf\_Blight',

10:'Corn\_(maize)\_\_\_healthy',

11:'Grape\_\_\_Black\_rot',

12:'Grape\_\_Esca(Black\_Measles)',

13:'Grape\_\_Leaf\_blight(Isariopsis\_Leaf\_Spot)',

14:'Grape\_\_\_healthy',

15:'Orange\_\_Haunglongbing(Citrus\_greening)',

16:'Peach\_\_\_Bacterial\_spot',

17:'Peach\_\_\_healthy',

18:'Pepper,bell\_\_Bacterial\_spot',

19:'Pepper,bell\_\_healthy',

20:'Potato\_\_\_Early\_blight',

21:'Potato\_\_\_Late\_blight',

22:'Potato\_\_\_healthy',

23:'Raspberry\_\_\_healthy',

24:'Soybean\_\_\_healthy',

25:'Squash\_\_\_Powdery\_mildew',

26:'Strawberry\_\_\_Leaf\_scorch',

27:'Strawberry\_\_\_healthy',

28:'Tomato\_\_\_Bacterial\_spot',

29:'Tomato\_\_\_Early\_blight',

30:'Tomato\_\_\_Late\_blight',

31:'Tomato\_\_\_Leaf\_Mold',

32:'Tomato\_\_\_Septoria\_leaf\_spot',

33:'Tomato\_\_\_Spider\_mites Two-spotted\_spider\_mite',

34:'Tomato\_\_\_Target\_Spot',

35:'Tomato\_\_\_Tomato\_Yellow\_Leaf\_Curl\_Virus',

36:'Tomato\_\_\_Tomato\_mosaic\_virus',

37:'Tomato\_\_\_healthy'}

model = load\_model('plants.h5')

def predict\_label(img\_path):

test\_image = image.load\_img(img\_path, target\_size=(224,224))

test\_image = image.img\_to\_array(test\_image)/255.0

test\_image = test\_image.reshape(1, 224,224,3)

predict\_x=model.predict(test\_image)

classes\_x=np.argmax(predict\_x,axis=1)

return verbose\_name[classes\_x[0]]

@app.route("/")

@app.route("/first")

def first():

return render\_template('first.html')

@app.route("/login")

def login():

return render\_template('login.html')

@app.route("/index", methods=['GET', 'POST'])

def index():

return render\_template("index.html")

@app.route("/submit", methods = ['GET', 'POST'])

def get\_output():

if request.method == 'POST':

img = request.files['my\_image']

img\_path = "static/tests/" + img.filename

img.save(img\_path)

predict\_result = predict\_label(img\_path)

return render\_template("prediction.html", prediction = predict\_result, img\_path = img\_path)

@app.route("/performance")

def performance():

return render\_template('performance.html')

@app.route("/chart")

def chart():

return render\_template('chart.html')

if \_name\_ =='\_main\_':

app.run()

**CHAPTER 7**

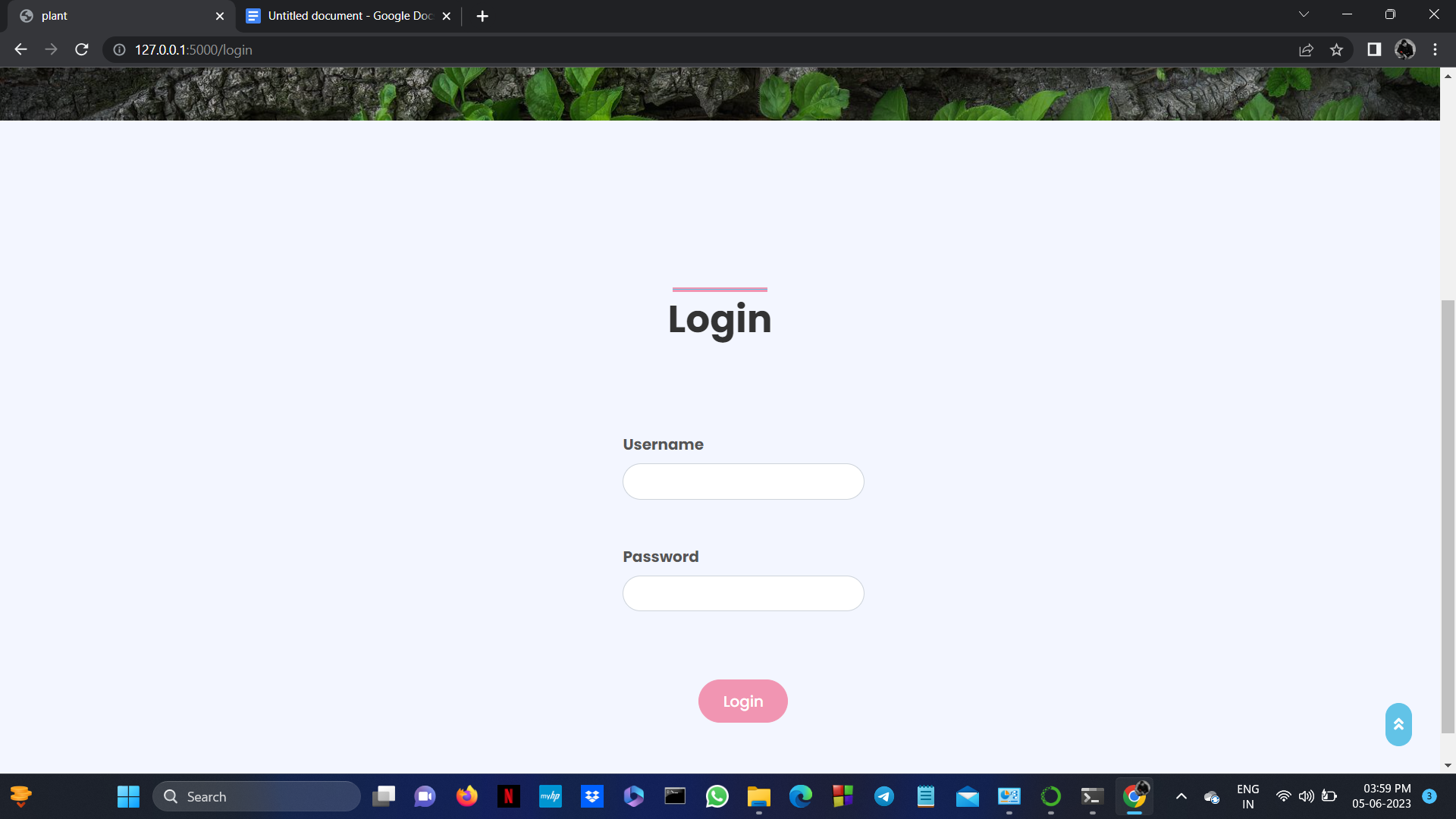
**SNAPSHOTS**

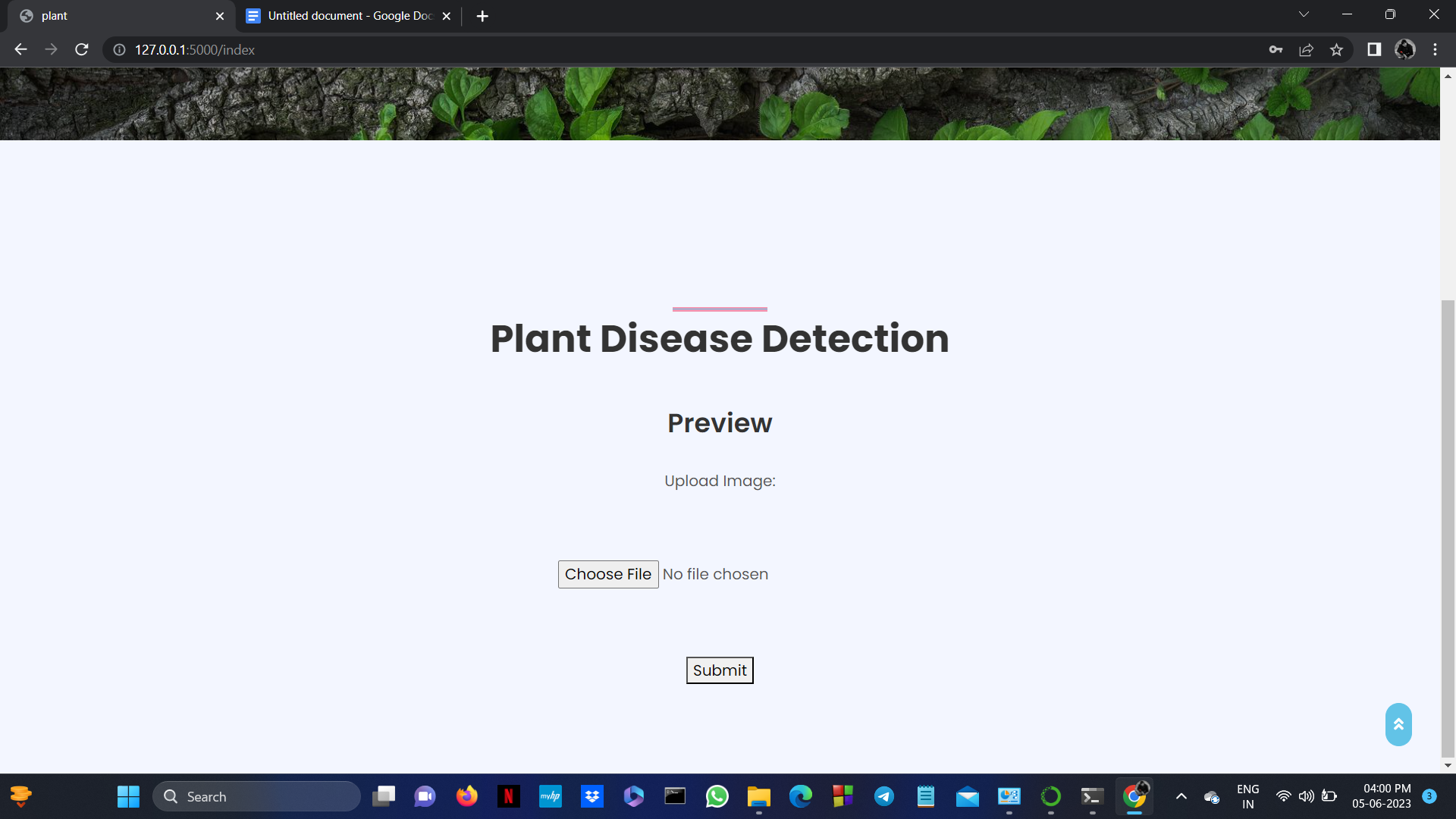
**General:**

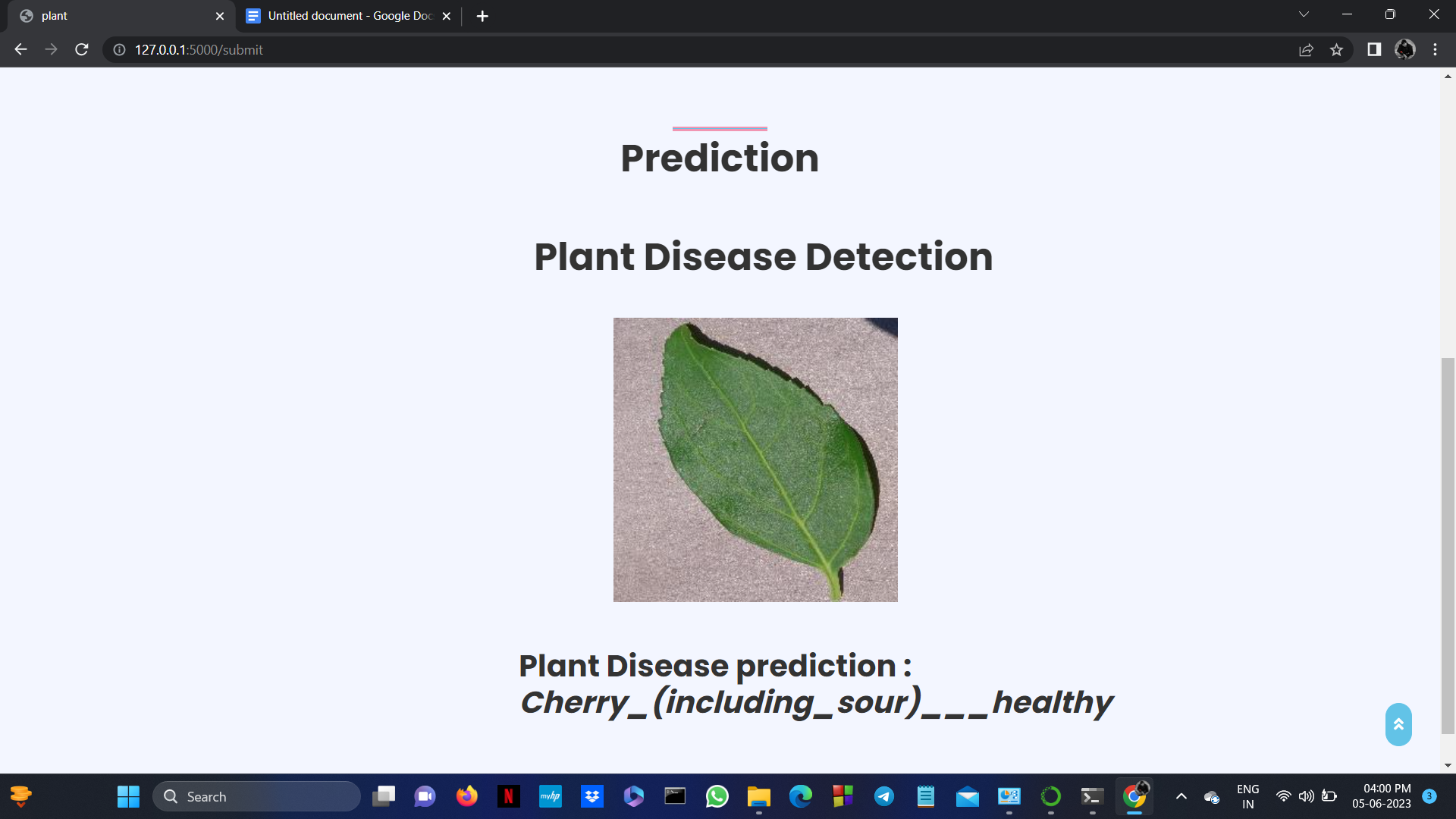
This project is implements like application using python and the Server process is maintained using the SOCKET & SERVERSOCKET and the Design part is played by Cascading Style Sheet.

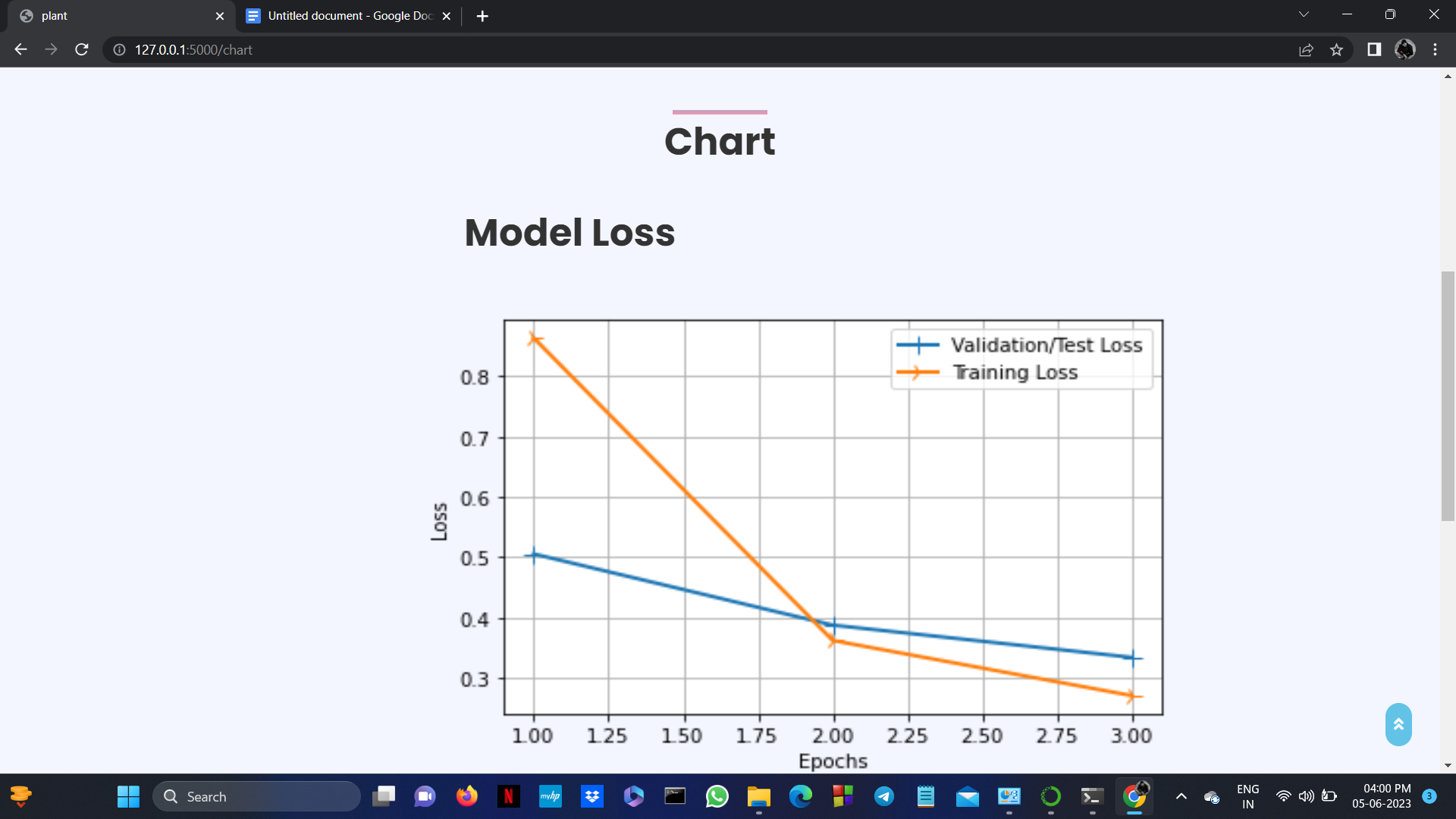
**SNAPSHOTS**

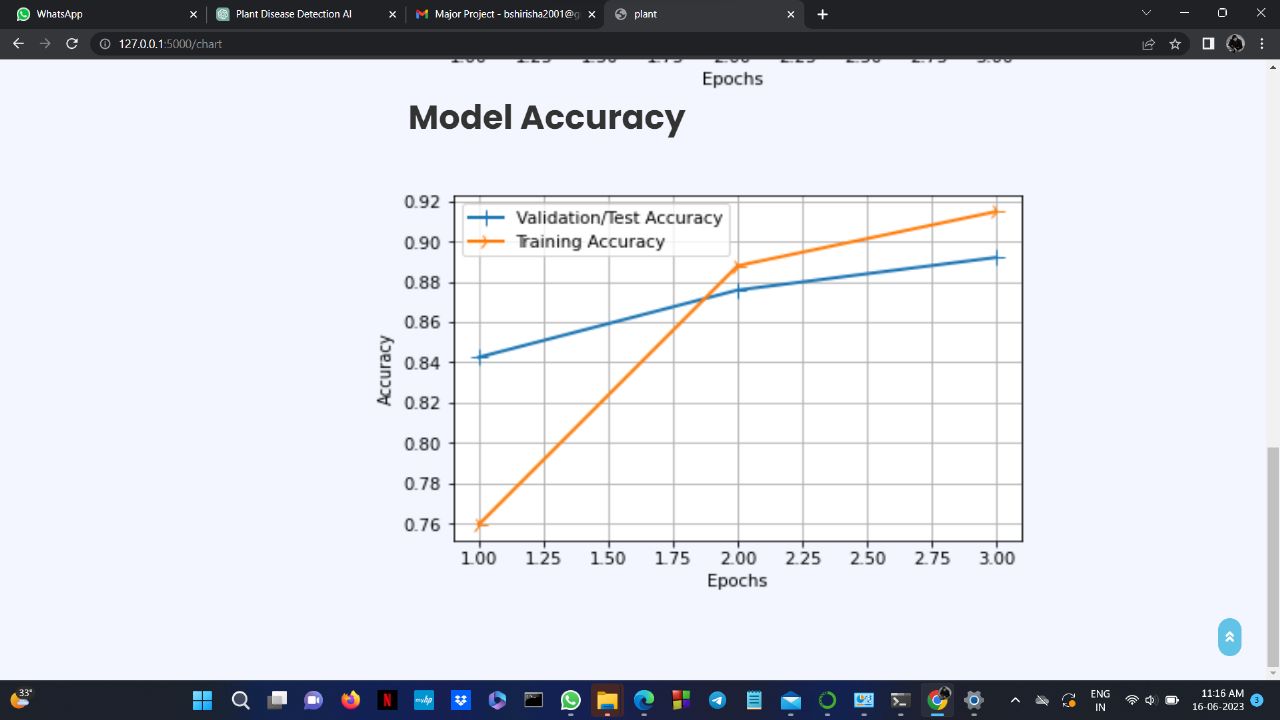










**CHAPTER 8**

**SOFTWARE TESTING**

**8.1 GENERAL**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**8.2 DEVELOPING METHODOLOGIES**

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

**8.3Types of Tests**

**8.3.1 Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**8.3.2 Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

**8.3.3 System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**8.3.4 Performance Test**

The Performance test ensures that the output be produced within the time limits,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

**8.3.5 Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**8.3.6 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Acceptance testing for Data Synchronization:**

* The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
* The Route add operation is done only when there is a Route request in need
* The Status of Nodes information is done automatically in the Cache Updation process

**8.2.7 Build the test plan**

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

**CHAPTER 9**

**FUTURE ENHANCEMENT**

**9.1 FUTURE ENHANCEMENTS:**

In future work will involve increasing the performance of DL in the detection of plant diseases having a multiclass subcategory. As the severity of the disease is supposed to be changing with time, so the DL should be improved to enhance detection and classification of the diseases throughout the whole development cycle of plant leaves.

**CHAPTER 10**

**CONCLUSION AND REFERENCES**

**10.1 CONCLUSION**

Plants make up more than 80 percent of the human diet. As a result, they are critical for food security and ensuring that we all have access to enough, cheap, clean, and nutritional food to lead healthy and active living. This research focuses on plant diseases as they create a major threat to food security. Plant protection in organic agriculture is not a simple matter. It depends on a thorough knowledge of the plants grown and their likely pests, pathogens and weeds. In our system InceptionV3 Architecture was used for the detection of plant diseases through leaves images of healthy or diseased plants. Our experimental results are able to successfully recognize different categories of diseases in various plants. Pests/diseases are generally not a significant problem in organic systems, since healthy plants living in good soil with balanced nutrition are better able to resist pest/disease attack. We hope our proposed

**10.2 REFERENCES**

[1] Shruthi, U., V. Nagaveni, and B. K. Raghavendra. "A review on machine learning classification techniques for plant disease detection." In 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), pp. 281-284. IEEE, 2019.

[2] Sehgal, Aman, and Sandeep Mathur. "Plant disease classification using soft computing supervised machine learning." In 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 75-80. IEEE, 2019.

[3] Hungilo, Gilbert Gutabaga, Gahizi Emmanuel, and Andi WR Emanuel. "Image processing techniques for detecting and classification of plant disease: a review." In Proceedings of the 2019 international conference on intelligent medicine and image processing, pp. 48-52. 2019.

[4] Yang, Xin, and Tingwei Guo. "Machine learning in plant disease research." European Journal of BioMedical Research 3, no. 1 (2017): 6-9.

[5] Akhtar, Asma, Aasia Khanum, Shoab A. Khan, and Arslan Shaukat. "Automated plant disease analysis (APDA): performance comparison of machine learning techniques." In 2013 11th International Conference on Frontiers of Information Technology, pp. 60-65. IEEE, 2013.

[6] Elangovan, K., and S. Nalini. "Plant disease classification using image segmentation and SVM techniques." International Journal of Computational Intelligence Research 13, no. 7 (2017): 1821-1828.

[7] Mohanty, Sharada P., David P. Hughes, and Marcel Salathé. "Using deep learning for image-based plant disease detection." Frontiers in plant science 7 (2016): 1419.

[8] Ramesh, Shima, Ramachandra Hebbar, M. Niveditha, R. Pooja, N. Shashank, and P. V. Vinod. "Plant disease detection using machine learning." In 2018 International conference on design innovations for 3Cs compute communication control (ICDI3C), pp. 41-45. IEEE, 2018.

[9] Venkataramanan, Aravindhan, Deepak Kumar P. Honakeri, and Pooja Agarwal. "Plant disease detection and classification using deep neural networks." Int. J. Comput. Sci. Eng 11, no. 9 (2019): 40-46.