GROUP ID: GB13

A PROJECT REPORT ON

FRUIT DISEASE DETECTION AND PESTICIDE RECOMMENDATION

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

SAMVEDYA JEDHEDESHMUKH	**B190334315**
SAHIL KACHOLE	**B190334322**
NISHANT PARAKH	**B190334405**
UMESH CHAUDHARI	**B190334239**

DEPARTMENT OF COMPUTER ENGINEERING

PCET'S PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

Sector No. 26, Pradhikaran, Nigdi, Pimpri-Chinchwad, PUNE 411044

SAVITRIBAI PHULE PUNE UNIVERSITY
2022-2023

CERTIFICATE

This is to certify that the project report entitled

"FRUIT DISEASE DETECTION AND PESTICIDE RECOMMENDATION"

Submitted by

SAMVEDYA JEDHEDESHMUKH	**B190334315**
SAHIL KACHOLE	**B190334322**
NISHANT PARAKH	**B190334405**
UMESH CHAUDHARI	**B190334239**

are bonafide students of this institute and the work has been carried out by them under the supervision of **Prof. Madhura Kalbhor** and it is approved for the partial fulfilment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

(Prof. Madhura Kalbhor)
Guide
Department of Computer Engineering

(Prof. Dr. K. Rajeswari)
Head,
Department of Computer Engineering

(Prof. Dr. G.N. Kulkarni)
Director,
Pimpri Chinchwad College of Engineering Pune – 44

Place: Pune
Date:

ACKNOWLEDGEMENT

We express our sincere thanks to our **Guide Prof. Madhura Kalbhor** for her constant encouragement and support throughout our project, especially for the useful suggestions given during the project and having laid down the foundation for the success of this work.

We would also like to thank our **Project Coordinator, Prof. Sushma R. Vispute** for her assistance, genuine support and guidance from early stages of the project. We would like to thank **Prof. Dr. K. Rajeswari, Head of Computer Department** for her unwavering support during the entire course of this project work. We are very grateful to our **Director, Prof. Dr. G.N. Kulkarni** for providing us with an environment to complete our project successfully. We also thank all the staff members of our college and technicians for their help in making this project a success.

We also thank all the web committees for enriching us with their immense knowledge. Finally, we take this opportunity to extend our deep appreciation to our family and friends for all that they meant to us during the crucial times of the completion of our project.

SAMVEDYA JEDHEDESHMUKH
SAHIL KACHOLE
NISHANT PARAKH
UMESH CHAUDHARI

ABSTARCT

Fruit crops are a crucial component of agriculture worldwide because they offer important nutrients and financial advantages. However, a number of diseases can seriously harm fruit orchards, resulting in a decrease in output and quality. Manual examination and treatment is a time-consuming, labour-intensive, and sometimes unsuccessful procedure. Therefore, the creation of an automated system for fruit disease diagnosis and pesticide recommendation has the potential to completely alter how we cultivate fruits. This method reliably identifies and categorises illnesses impacting fruit crops using cutting-edge technology including machine learning, image processing, and data analysis. In addition to lowering the use of toxic pesticides and encouraging environmentally friendly and sustainable agricultural methods, it may provide the best pesticide treatment depending on the disease's symptoms. which can provide farmers access to real-time monitoring of crop health and environmental factors.

Among the many advantages of a fruit disease detection system with pesticide recommendation are increased food safety, increased productivity, and less environmental impact. This technology has the potential to transform how we cultivate fruits and help to the conservation of biodiversity and the health of ecosystems. Therefore, to improve the system's precision and efficacy and make it more available to farmers throughout the world, research and development in this field are crucial.

Keywords: CNN**,** ResNet50, Deep Learning, Image Processing

TABLE OF CONTENTS

LIST OF ABBREVATIONS

ABBREVIATION	ILLUSTRATION
CNN	Convolutional Neural Network
SVC	Support Vector Classifier
KNN	K-Nearest Neighbors Algorithm
DL	Deep Learning

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

In India, the livelihood of 58% of the Indian population is based on agriculture. So, the constantly changing climatic conditions and also some diseases have a high effect on crops and are leading to less crop yield. And India stands second in the list of highly populated countries and it is still increasing. Agriculture research aims to increase food production and quality while lowering costs and boosting products. Fruit trees play an important role in any state's economic development. One of the most well-known fruit plant species is the citrus plant, which is high in vitamin C and widely used in the Indian sub-Continent, the Middle East and Africa. Citrus

plants are associated with many health advantages, as well as being used as a raw material in the agricultural industry for the production of several types of other agri-products, including jams, sweets, ice cream, and confectionery, etc.

On account of that, food consumption will automatically increase and this will lead us to the situation where people have to produce more food. India not only produces and exports food crops but also fruits. Here the classification of good and bad fruits is barely done manually in most of the places. This leads to more errors in the grading of fruits while export ing. So, to overcome the faults that happen during the manual classification, here also, researchers have proposed an image detection method to classify the diseased fruits from good fruits to improve the quality of classification while exporting fruits.

Here this approach is using CNN(Convolutional Neural Networks) which detects the quality of the fruit layer by layer. CNN approach will be detecting the diseases which affect the fruits and can even identify some types of diseases which attacks fruits based on some comparisons. CNN is a deep learning algorithm that is where input is taken as images, and those images were differentiated based on various aspects and parameters taken from it and is most commonly applied to analyzing visual imagery. This will be definitely helpful for the farmers to enhance the growth of the crops in the mere future.

The paper is organized into the subsequent sections. Very first section gives a brief introduction to the importance of plant disease detection. After that Second section discusses the existing work carried out recently in this area and reviews the techniques used. Third section includes basic methodology followed for developing disease detection system. Fourth section concludes this paper along with future directions.

1.1 OVERVIEW

Agricultural products are used to cater the daily needs of animals as well as human beings. Agriculture has been a part of everyone's life directly or indirectly. It is the way of crop production which ends up in providing food, the building block of each person. Whether a human resides during a metro city or lives during a village everyone survives on this crop production. The main objective of this review paper is to review various techniques of disease detection for multiple fruits and discuss in terms of various parameters.

1.2 MOTIVATION:

- Improved crop yields: Detecting fruit diseases early can help farmers take appropriate measures to control the spread of the disease, which can lead to improved crop yields.
- Reduced crop losses: Fruit diseases can cause significant crop losses, leading to financial losses for farmers. Early detection and treatment can help reduce these losses.
- Food safety: Some fruit diseases can be harmful to human health. Detecting these diseases early can help prevent the spread of contaminated fruit to consumers.
- Environmental sustainability: The use of pesticides to control fruit diseases can have negative environmental impacts. Early disease detection can help reduce the need for pesticides and promote more sustainable agricultural practices.
- Cost savings: Early detection of fruit diseases can lead to cost savings for farmers by reducing the need for expensive treatments and reducing crop losses.

1.3 PROBLEM DEFINITION AND OBJECTIVE

- **Problem Definition:** The problem we are addressing is the detection of fruit diseases in crops and suggesting appropriate pesticides to control the spread of the disease. Fruit diseases can cause significant damage to crops, resulting in reduced yields and financial losses for farmers.
- **Objective:**
    - Classify the different types of fruit images.
    - Propose a system that can predict the fruit diseases through image processing.
    - Preparing or acquiring required datasets.
    - Recommend pesticides for identified fruit diseases.

1.4 PROJECT SCOPE:

The scope of the fruit disease detection and pesticide suggestion project includes:

1. Image acquisition: The system should be able to acquire images of fruit from different sources, including digital cameras, smartphones, or drones.

2. Image preprocessing: The system should preprocess the acquired images to enhance their quality, including noise reduction, contrast enhancement, and image segmentation.

3. Disease detection: The system should analyze the preprocessed images to detect any signs of disease, including lesions, discoloration, or deformations. The system should be able to detect a range of diseases that affect different types of fruit, including apples, oranges, grapes, and others.

4. Pesticide suggestion: The system should suggest appropriate pesticides to treat the detected disease. The system should take into account factors such as the type of disease, the stage of the disease, the type of fruit, and the local regulations on pesticide use.

5. User interface: The system should provide an intuitive and user-friendly interface for farmers to upload images, view disease detection results, and receive recommendations for pesticides.

6. Database management: The system should maintain a database of fruit diseases, pesticides, and regulations on pesticide use. The database should be updated regularly to ensure that the system provides accurate and up-to-date recommendations.

7. Performance evaluation: The system should be evaluated on a set of performance metrics, including accuracy, speed, and usability. The evaluation should be conducted on a dataset of images with known disease labels.

1.5 LIMITATION:

- Limited data: The availability of high-quality and diverse data is essential for developing accurate models for fruit disease detection and pesticide suggestion. However, there may be limited data available for specific types of fruit or diseases, which can limit the accuracy of the system.

- Complexity of the problem: The detection of fruit diseases and the suggestion of appropriate pesticides is a complex problem that requires expertise in both agriculture and machine learning. Developing a system that can accurately detect multiple diseases and suggest appropriate treatments may be challenging.

- Variability in fruit appearance: The appearance of fruits can vary widely depending on factors such as lighting, orientation, and ripeness. This variability can make it difficult to develop accurate models for fruit disease detection.

- Limited access to hardware: As mentioned earlier, the system may require specialized hardware such as high-resolution cameras or processing units to achieve optimal performance. Limited access to these resources may limit the system's accessibility and effectiveness.

- User acceptance: The system's success depends on its acceptance and adoption by farmers. Therefore, it is important to consider farmers' needs and preferences when designing the system's user interface and functionality.

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY

In paper [1], The author considered apple diseases like apple scab, apple blotch, and apple rot; these are fungal diseases. The dataset of the apples was collected from the local market; from that sample, they picked the apples which were already infected. they have considered different dividing ratios of training and testing data sets. For every ratio, data set is divided randomly, and model is trained and tested for ten times. The complete data set consists of images of four classes, namely, Apple blotch, Apple rot, Apple scab and Healthy apples. Each class consists of 800 images. Different models based on convolutional neural network are used for the classification of healthy apples and identifies the diseases apple. All the models showed good classification accuracy on more than 90% on testing images. The best accuracy was achieved by model5 which is 99.17% when the training data set was 90% and the testing data set was 10%. For other training ratios, model5 outperforms other models in term of classification accuracy and time complexity.

The author proposed a model in reference [2] to find how much percent the fruit is affected and recognize the fruit in the given image. To get better results in the classification and identification of fruit diseases Inception v3 model and Transfer Learning are used. In this experiment, the results are shown in the two ways 1. Classification of fruits 2. Percent of disease identification. By taking a sample of fruits like apple, banana, and cherry. For calculating the percent of disease identification in this model fruit has been classified into four grades they are A, B, C and D. The grade A says the given image is generally excellent for eating, grade B says that a little piece of the fruit is ruined, grade C says the half of the fruit is ruined and D indicates that the fruit is totally spoiled and isn't consumable. Grade A varies from 0 to 25 percent, the grade B varies from25 to 50 percent, the grade C varies from the 50 to 75 percent and the grade D varies from the 75 to 100 percent.

The proposed CNN-based leaf disease identification model in paper [3] is capable of distinguishing between healthy and diseased Citrus fruits and leaves. We used the CNN model to tackle the problem of classifying diseases from citrus fruit and leaf images in this study. The modules in our proposed model are as follows: i) Data acquisition, ii) Data preprocessing, and iii) CNN model application. Two convolutional layers were used in the suggested CNN model. The first convolutional layer separates low-level features from the picture, while the second convolutional layer collects highlevel attributes, yielding disease classification of citrus fruit/leaves into Black spot, canker, scab, greening, and Melanose classes. On plant disease datasets, we tested a variety of machine and deep learning models and reported our findings. The suggested CNN outperformed other classifiers in terms of accuracy, scoring 95.65% for citrus fruit/leaf disease classification experiments.

According to authors in paper [4], the applications of Lab color space model and Convolutional Neural Networks (CNNs) have been used for the process of segmentation and classification of the diseases that affect the lemon plant's leaves. Recognizing the disease and classifying it is the main purpose of the introduced approach. The Multiclass SVM algorithm and the CNN Algorithm were trained and tested on a healthy lemon leaf and on three diseases which influences the lemon leaves. The model was able to detect the presence of leaf and distinguish between the healthy leaves and the leaves with different diseases, which can be visually diagnosed. The diseases are Anthracnose, Citrus Canker and Greasy Spot. The experimental results indicate that both the approaches can support the detection and classification of the diseases on the leaf. The level of accuracy in classification of the leaves by Multiclass SVM was 83.6%, whereas it was 93.8% when the leaves were classified using the CNN algorithm. Hence the accuracy in classification of the leaves using CNN algorithm showed better results.

A machine learning-based intelligent system, which can detect the papaya diseases has been presented through the proposed research work in paper [5]. This research work has used random forest, kmeans clustering, SVC, CNN and obtained a good accuracy in CNN (98.4%). The proposed model can help the farmer to find out the problem easily and take proper steps to reduce the diseases. In the future, the present research work will be extended to work on a large dataset to predict the factor that is mainly responsible for the papaya diseases.

In Paper [7], The proposed system has built to detect 4 diseases named Anthracnose, Powdery Mildew, Black Banded, Red Rust and to provide solution over them. For image enhancement “Imadjust” function is particularly used for contrast enhancement. This is a inbuilt function in MATLAB. Contrast enhanced is done by linearly scaling pixel values and features used are Contrast, Correlation, Energy, Entropy, Homogeneity, Cluster prominence, Cluster shade, Variance and Dissimilarity Classifier used is SVM. The GUI and programming is made using MATLAB. The proposed system gives accuracy of 90% while testing on 92 samples. This system will increase productivity and improve quality of mango fruit. In given system processing speed of segmentation is 3 seconds and results will be displayed after classification in 0.1 seconds. The overall system will worked in 5 seconds.

Vani Ashok & D. S. Vinod [8] proposed a system that uses a pre-trained model of deep convolutional neural network (CNN) architecture, to retrain on images of mango fruits with the goal of identifying diseases (or absence thereof) and also on images that the model had not seen before. The system retrains the model on the principle of transfer learning (TL) using a dataset of diseased and healthy mango images acquired under controlled conditions. The system also incorporates functionality of listing out the causes and certain common symptoms of the diseases. The proposed system embeds the prediction model within the smartphone application so as to completely eliminate the necessity of internet access for mango disease detection. The maximum achieved training accuracy is 98.6%, and maximum validation accuracy is 96.4%.

Another study was conducted by Avinash Kumar, Sobhangi Sarkar and Chittaranjan Pradhan using the technology of recommendation system in which they recommended the crops suitable for growing [9]. SVM classification algorithm, SVM classification algorithm, Decision Tree algorithm and Logistic Regression Algorithm were the techniques used and the rules induced from these models helps in building Recommendation System. In this model an accuracy of 89.66% was achieved .

Other research about crop diseases was conducted by Zeel Doshi, Rashi Agrawal, Subhash Nadkarni, Prof. Neepa Shah[10]. In this paper an intelligent system, called Agro Consultant, is presented, which intends to assist the Indian farmers in making an informed decision about which crop to grow depending on the sowing season, his farm's geographical location, soil characteristics as well as environmental factors such as temperature and rainfall.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3. SOFTWARE REQUIREMENTS SPECIFICATION

3.1 ASSUMPTIONS AND DEPENDENCIES

3.1.1 Assumptions:

1. Image quality: The images of the fruit provided by the farmers are of sufficient quality and contain the necessary features for disease detection.
2. Expertise: The users of the system have a basic understanding of fruit diseases and pesticide use.
3. Pesticide availability: The suggested pesticides are available and legal in the region where the system is deployed.
4. Disease labels: The training dataset used for building the disease detection model is accurate and comprehensive.
5. Homogeneity: The fruit samples used in the project are similar in terms of their variety, ripeness, and growing conditions.
6. No additional factors: The system's accuracy in detecting diseases and suggesting pesticides is not affected by additional factors such as weather conditions, soil quality, or pest infestation.

3.1.2 Dependencies:

1. Data availability: The system's performance depends on the availability of a diverse and comprehensive dataset of fruit images with associated disease labels. The availability and quality of this data can significantly affect the system's accuracy.
2. Hardware requirements: The system may require specialized hardware, such as high-resolution cameras or processing units, to achieve optimal performance. The availability and accessibility of such hardware can affect the system's feasibility and effectiveness.
3. Expertise: The development and implementation of the project require expertise in both agriculture and machine learning. The availability of such expertise can affect the project's quality and success.
4. Pesticide regulations: The project's success may be affected by pesticide regulations in the region where it is deployed. Compliance with these regulations may limit the types of pesticides that can be recommended by the system.
5. User acceptance: The system's success depends on its acceptance and adoption by farmers. Therefore, it is crucial to consider farmers' needs and preferences when designing the system's user interface and functionality.

6. Ethical considerations: The use of pesticides raises ethical concerns related to the environmental impact, health hazards, and regulation compliance. The system should take into account these considerations and provide responsible recommendations.

3.2 FUNCTIONAL REQUIREMENTS

Functional requirements for an disease detection and pesticide suggestion system may include the following:

1. Upload Image: The system should be able to upload images of the environment using the camera.
2. Disease classification: The system should be able to classify the disease for different classes.
3. Image processing: The system should be able to process the images uploaded by the user to detect disease in fruits.
4. Disease detection: The system should be able to detect Disease in the fruits by analysing the existing dataset.
5. User interface: The system should include a user interface, such as a GUI, to display the output of the system.
6. Dataset: The system should have a dataset to train the machine learning models used in the image processing and fruit disease detection modules.
7. System performance: The system should perform accurately and reliably in different lighting conditions and environments.
8. System maintenance: The system should be easy to maintain and update as required.
9. System integration: The system should be able to integrate with other hardware and software components as required.
10. System scalability: The system should be scalable to accommodate larger environments and more complex images.

These are some of the functional requirements that an fruit disease detection and pesticide suggestion system.

3.3 EXTERNAL INTERFACE REQUIREMENTS

External user interface requirements for an fruit disease detection and pesticide suggestion system may include the following:

3.3.1 User Interfaces

- The system should have a user-friendly graphical user interface (GUI) to display the output of the system.
- The GUI should be intuitive and easy to use, even for users with little technical knowledge.
- The GUI should display real-time footage captured by the camera.
- The GUI should display the disease detected by system.
- The GUI should allow users to start and stop the system.

3.3.2 Hardware Interfaces

- The system should have hardware interfaces to connect the camera, and computer system used to run the software.
- The system should be compatible with a range of images with different types.
- The system should be designed to work with the minimum hardware specifications required to achieve accurate and reliable performance.

3.3.3 Software Interfaces

- The system should have network interfaces to communicate with internet.
- The system should be designed to work with a range of software tools and libraries commonly used in image processing and shape detection.

3.3.4 Communication Interfaces

- The system should have communication interfaces to allow the system to integrate with other hardware or software components as required.
- The system should be designed to work with a range of communication protocols commonly used in the industry, such as TCP/IP, USB, and Ethernet.
- The system should be able to communicate with other systems or devices, such as drones or robots, to enable remote operation or control.

3.4 NON-FUNCTIONAL REQUIREMENTS:

3.4.1 Performance Requirements:

- The system should be able to process images having different types.
- The system should be able to detect fruit disease accurately with a false positive.
- The system should be able to detect disease for different types of fruits.
- The system should be able to recommend the appropriate pestiocides.

3.4.2 Safety Requirements:

- The system should not emit any harmful radiation or pose a risk to human or animal health.
- The system should not interfere with other devices or systems in the environment, such as drones or communication networks.
- The system should be designed with fail-safe mechanisms to ensure that any malfunctions do not result in accidents or injuries.

3.4.3 Security Requirements:

- The system should be designed with secure authentication mechanisms to prevent unauthorized access or tampering.
- The system should be designed with secure data transmission protocols to ensure that images and data captured by the camera are not intercepted or tampered with during transmission.
- The system should be designed with secure storage mechanisms to prevent unauthorized access or tampering of stored data.

3.4.4 Software Quality Attributes:

- The system should be designed with a modular architecture that allows for easy maintenance and scalability.
- The system should be designed with high code quality and adherence to coding standards to ensure easy maintenance and reliability.
- The system should be designed with comprehensive error handling and logging mechanisms to facilitate debugging and maintenance.
- The system should be designed with efficient memory management to ensure that the system operates with minimal memory usage.

3.5 SYSTEM REQUIREMENTS

3.5.1 Database Requirements

- The system should have a database to store information about disease detected by the system.
- The database should be designed to handle large amounts of data and should be scalable to accommodate future growth.
- The database should be designed to ensure data integrity and security, with proper backup and recovery mechanisms.
- The database should be accessible by authorized users only and should be designed with secure authentication and access control mechanisms.

3.5.2 Software Requirements:

- The system should be developed using a programming language suitable for real-time image processing such as Python.
- The system should be developed using a suitable software framework for computer vision, such as OpenCV, keras, Pandas.
- The system should be developed to run on a suitable platform, such as a Linux-based operating system or Windows.

3.5.3 Hardware Requirements:

- The system should include a high-performance computer for processing huge image data.
- The system should include a camera for capturing images of fuits.
- The system should include a computer or microcontroller with sufficient processing power and memory to run the software and perform real-time image processing.
- The system should include sufficient storage for captured images, as well as for system logs and other data.
- The system should be designed with suitable power supply and backup mechanisms to ensure uninterrupted operation.

3.6 ANALYSIS MODEL: AGILE MODEL

Agile Methodology is used to adapt to changes fast and efficiently. Its main goal is to facilitate quick project completion. In Agile model the requirements are decomposed into small parts that are developed incrementally.

These are the following phases:

- **Concept**

First is concept phase. Here we determine the scope of the project. We discussed key requirements and prepare documentation to outline them, including what features will be supported and the proposed end results. We kept the requirements to a minimum as they can be added to in later stages. This detailed analysis helped us to decide whether or not a project is feasible.

- **Inception**

Once the concept is outlined, we started with software development planning. We started the design process. We planned and drew some sample mockup user interface and build the project architecture. The inception stage helped us determine the product functionality.

- **Iteration**

Next up is the iteration phase. It is the longest phase as the bulk of the work is carried out here. We will work on UX to combine all product requirements and turn the design into code. The goal is to build the bare functionality of the product by the end of the first iteration or sprint. Additional features and tweaks can be added in later iterations.

- **Release**

The product is almost ready for release. But for quality assurance needs to perform some tests to ensure the software is fully functional. The team members will test the system to ensure the code is clean — if potential bugs or defects are detected, the developers will address them swiftly.

- **Maintenance**

The software will now be fully deployed and made available to customers. This action moves it into the maintenance phase. During this phase, the software development team will provide ongoing support to keep the system running smoothly and resolve any new bugs.

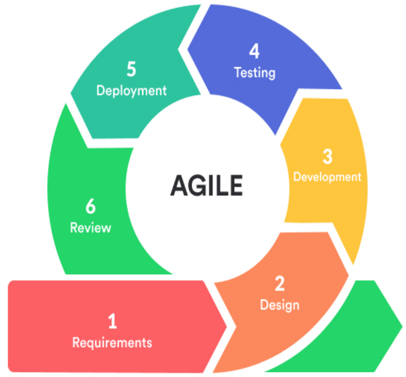


Figure 3.1 Agile Model

CHAPTER 4

SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

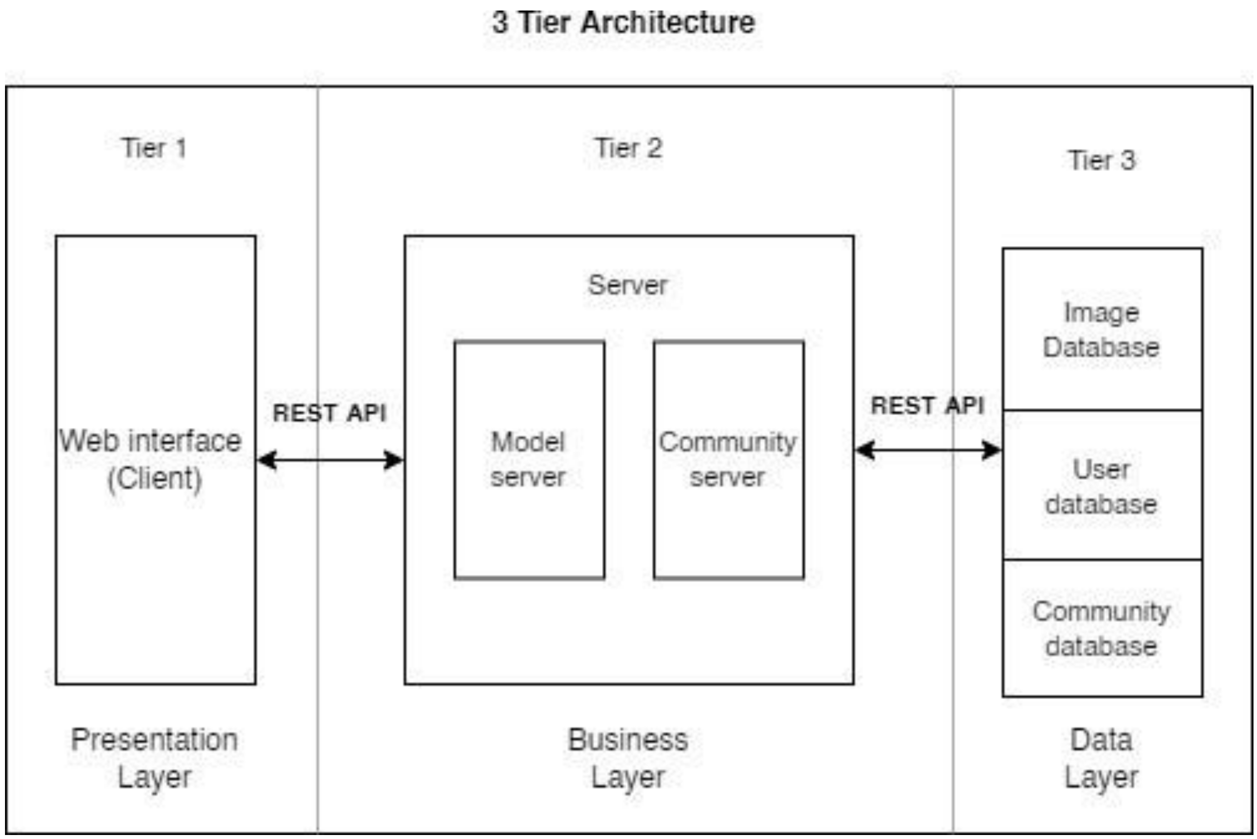


Figure 4. 1 System Architecture

4.2 UML DIAGRAMS:

4.2.1 Flow Chart:

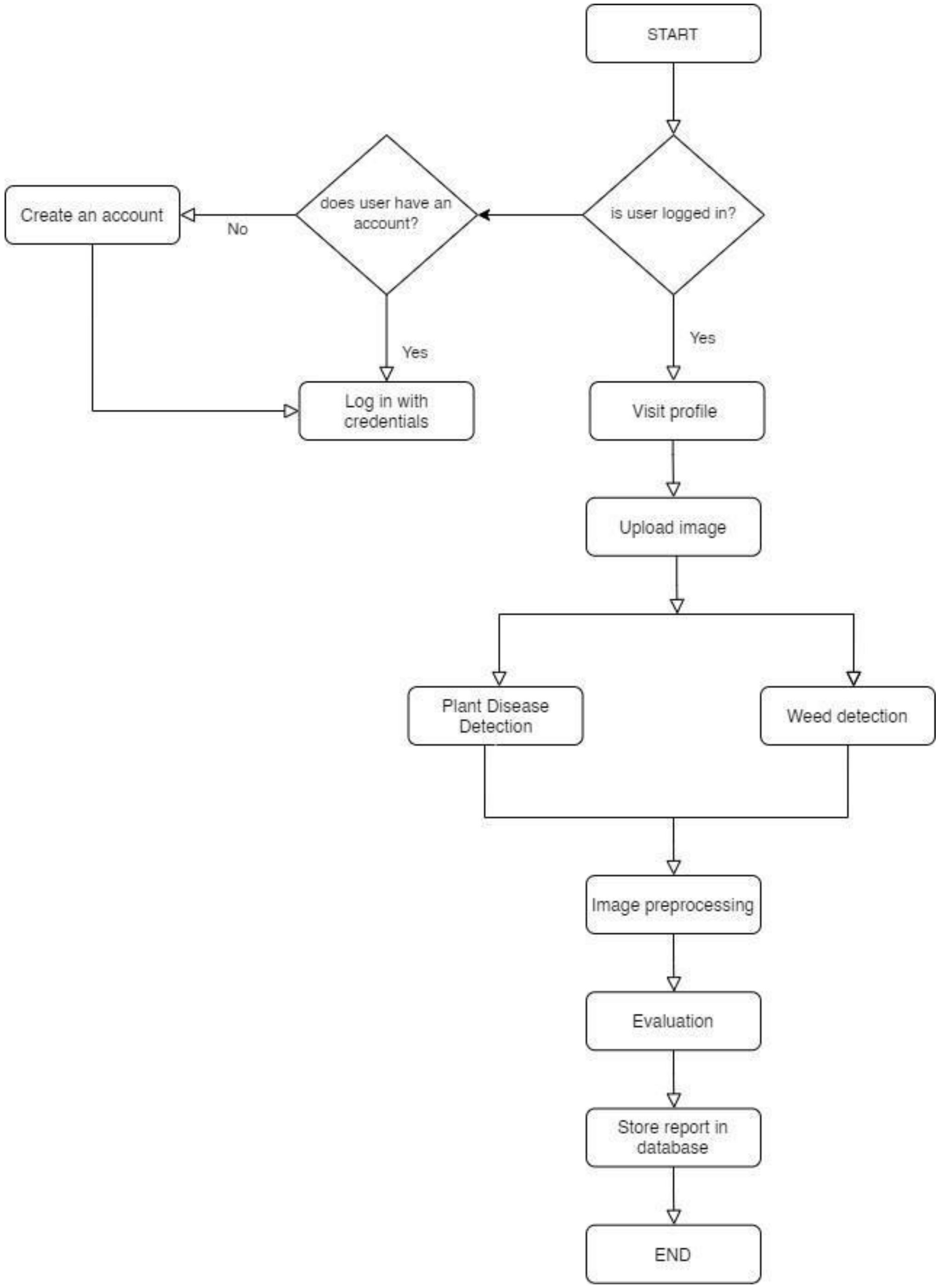


Figure 4. 2 Flow Chart Diagram

4.2.2 Use Case Diagram:

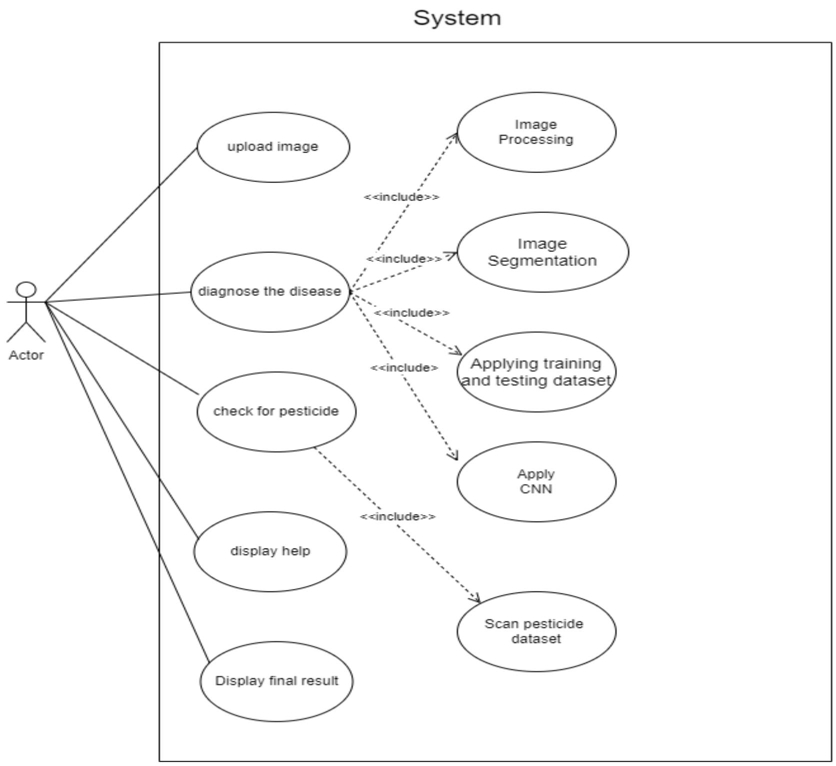


Figure 4. 3 Use Case Diagram

4.2.3 Data Flow Diagram:

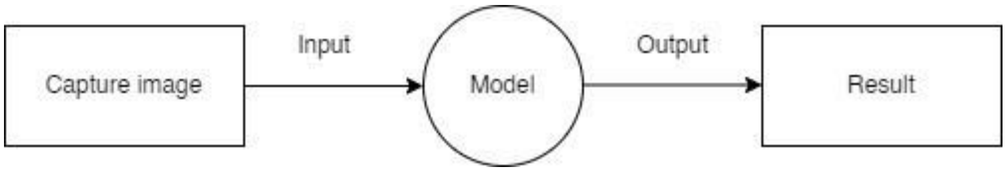


Figure 4. 4 DFD level – 0

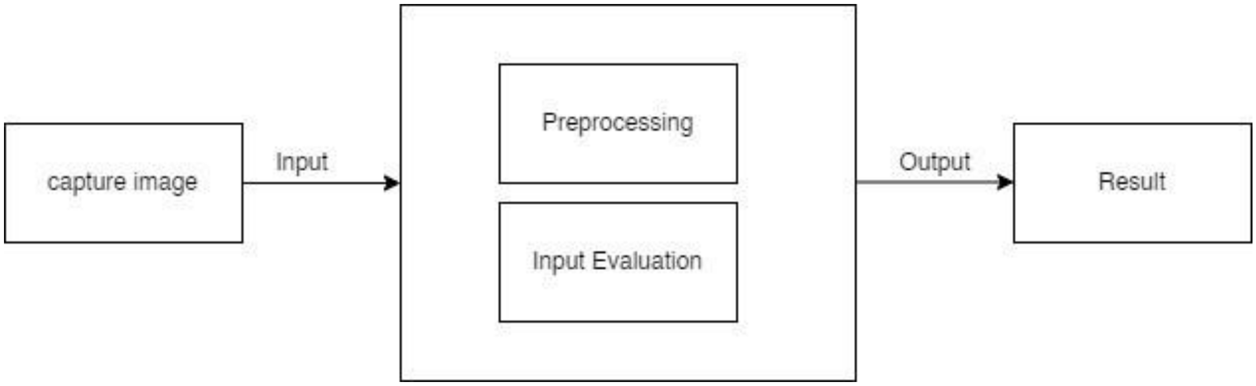


Figure 4. 5 DFD level – 1

4.2.4 Activity Diagram:

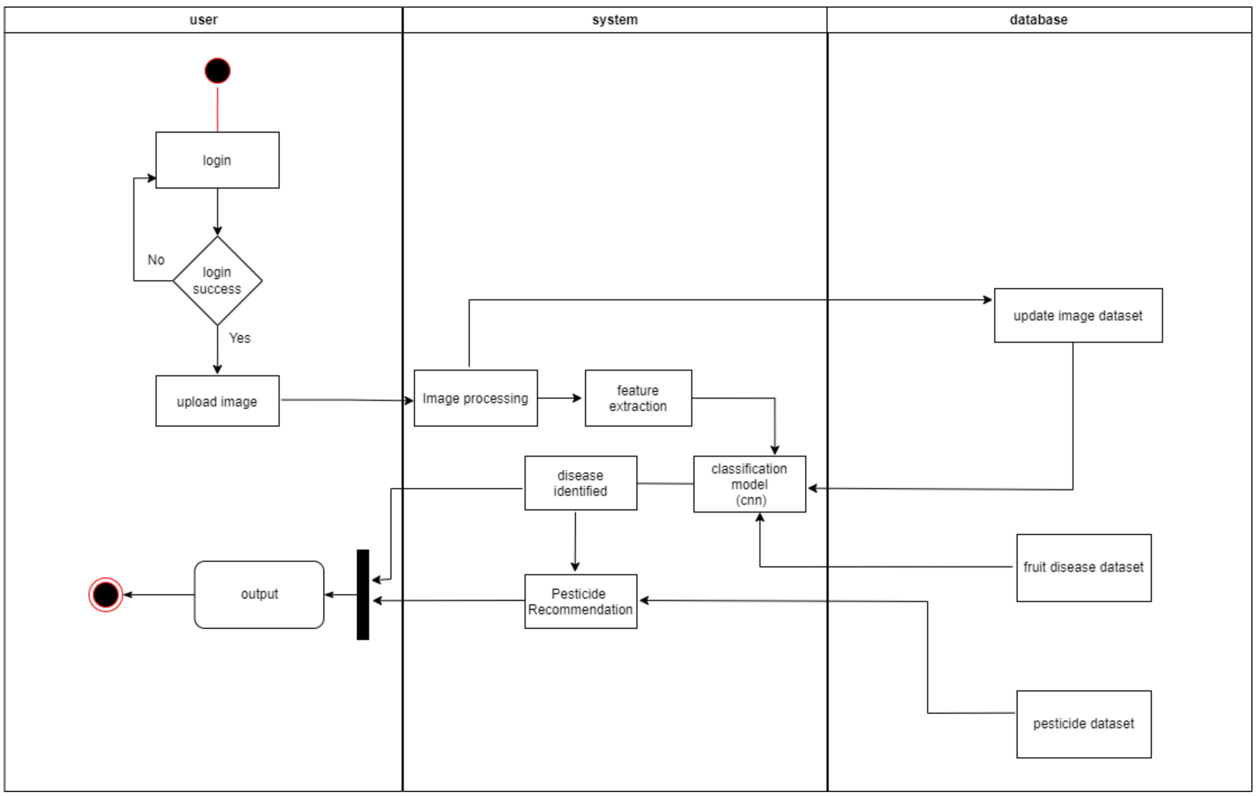


Figure 4. 6 Activity Diagram

4.2.5 Sequence Diagram:

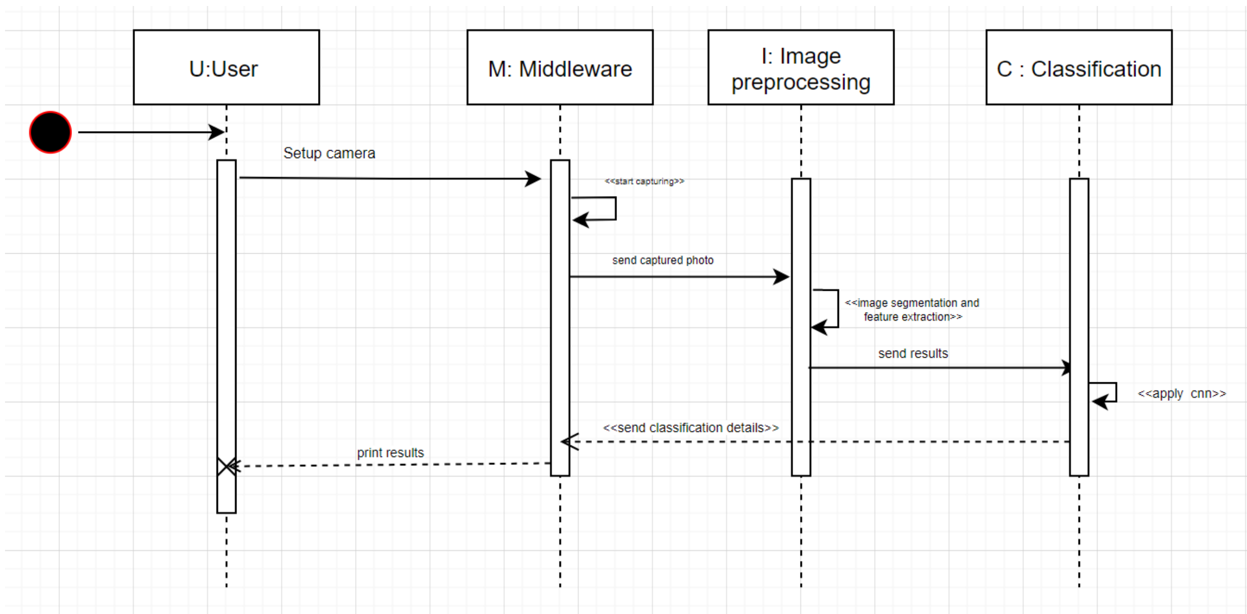


Figure 4. 7 Sequence Diagram

4.2.6 Class Diagram:

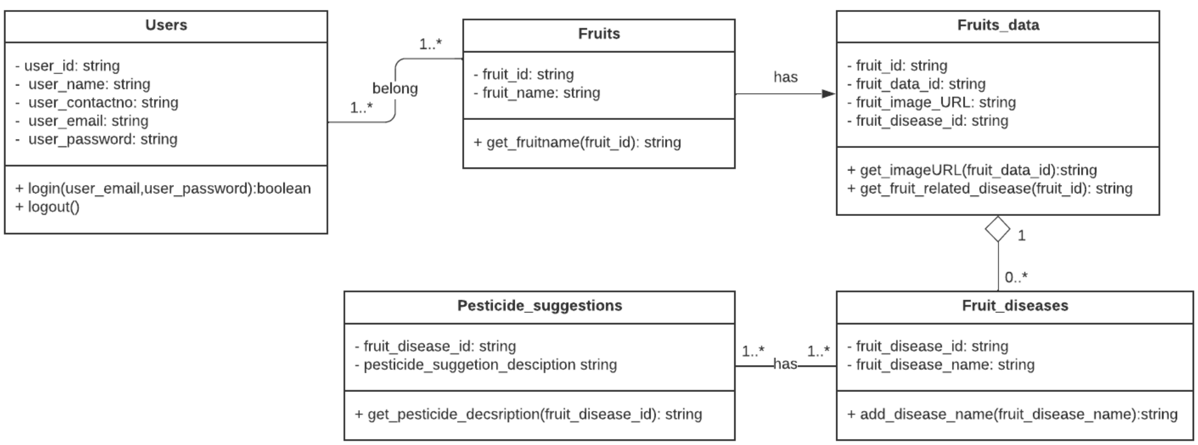


Figure 4. 8 Class Diagram

4.2.7 Object Diagram:

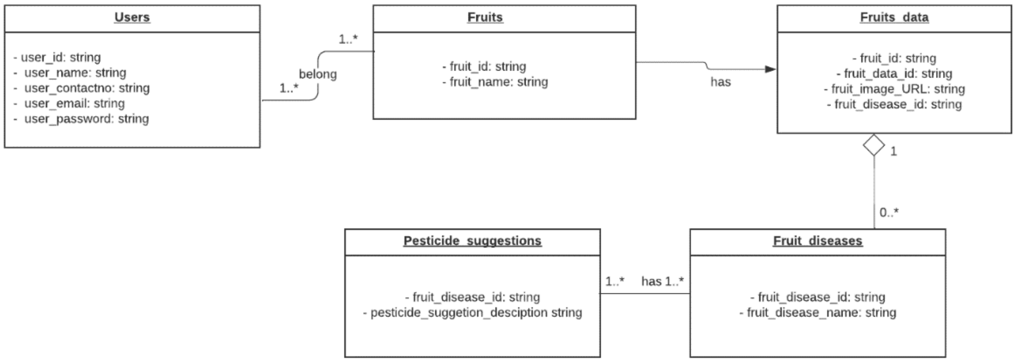


Figure 4. 9 Object Diagram

4.2.8 State Diagram:

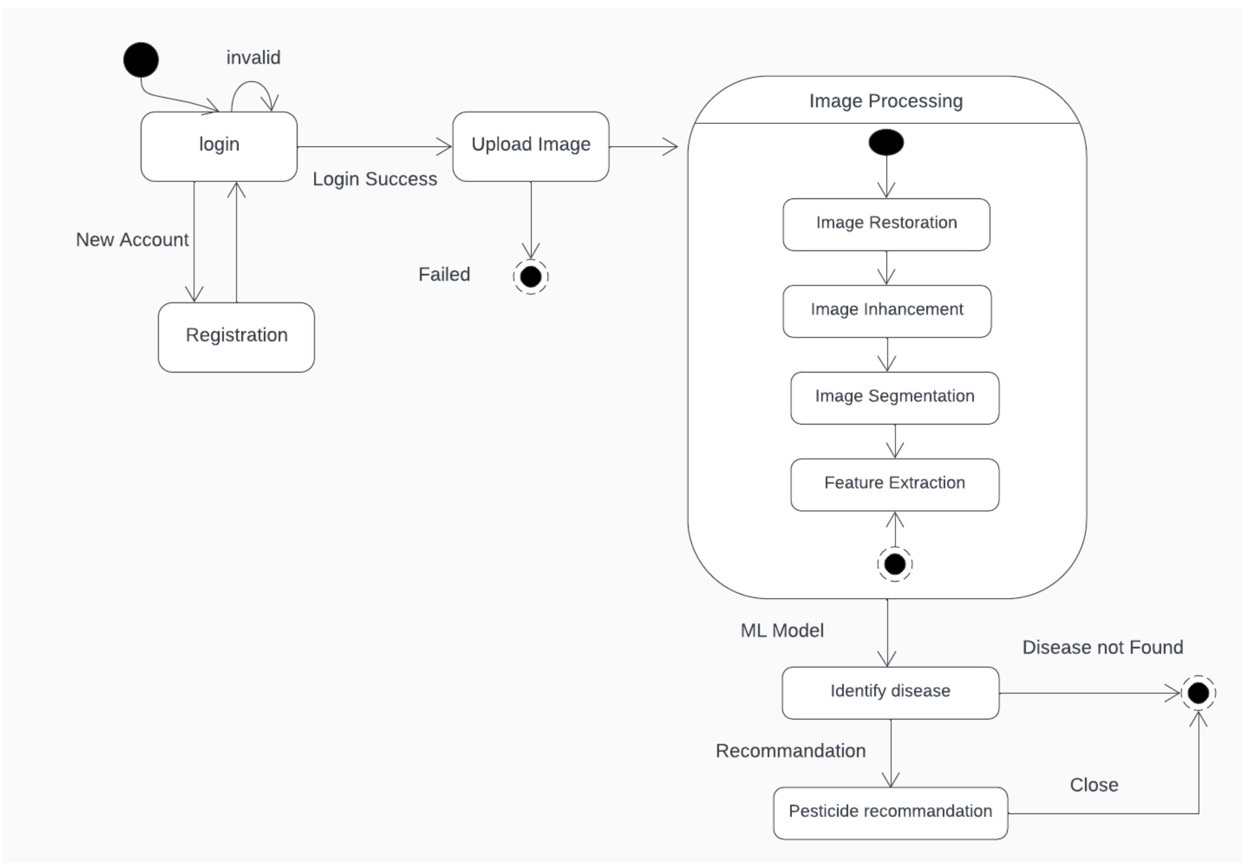


Figure 4. 10 State Diagram

CHAPTER 5

PROJECT PLAN

5. PROJECT PLAN

5.1 PROJECT ESTIMATE

- The scope of the project for fruit disease detection and pesticide suggestion would involve developing a system that uses a CNN algorithm and then analysing the resulting image to detect any disease in fruit.
- A feasibility study for an fruit disease detection and pesticide suggestion using a CNN is an evaluation process that determines the technical and economic viability of the project. This study aims to identify whether the proposed system is able to detect the disease.
- Estimating the project duration, cost, and resources required for fruit disease detection using a CNN algorithm involves several factors that need to be considered. It's important to note that the duration, cost, and resources required for the project will depend on various factors, such as the size of image, the complexity of an algorithm. Therefore, it's essential to carefully assess these factors and adjust the estimate accordingly.

5.1.1 Reconciled Estimates:

1. Conduct a detailed analysis of project requirements and resources needed: This involves a thorough understanding of the project scope and requirements. The project team has analysed the technical specifications of the fruit disease detection and any other necessary hardware and software and also reviewed the project deliverables, goals, and objectives. This analysis help to identify the resources needed for the project, including personnel, equipment, and software.
2. Estimate the cost and duration of each project activity: Once the project requirements and resources were identified, the project team estimated the cost and duration of each activity. This includes estimating the time and resources required for the design, development, testing, and deployment of the fruit disease detection system. The project team has also identify any potential risks and developed a contingency plan to mitigate these risks.
3. Develop a project schedule with timelines for each activity: Based on the estimated duration of each activity, the project team created a project schedule with timelines for each activity. The schedule include key milestones and deliverables, as well as the estimated start and end dates for each activity. This helped to ensure that the project is completed on time and within budget.
4. Identify any constraints that may affect project execution: During the project planning phase, the team has identify any potential constraints that may affect project execution. These include budget constraints, resource availability, technical limitations, or legal and regulatory requirements. The team

also develop a plan to manage these constraints and ensure that they do not impact the project timeline or budget.

5. Adjust estimates based on new information or changes in project scope: Throughout the project, the team may receive new information or changes to the project scope. As a result, it may be necessary to adjust the estimates for the project duration, cost, and resources required. The team should carefully evaluate any changes and adjust the project plan accordingly. They should also communicate any changes to stakeholders to ensure that everyone is aware of the impact on the project timeline and budget.

5.1.2 Project Resources:

Identify and secure necessary resources, such as hardware, software, and personnel.

- ✓ **Hardware:**
  - Processor – i5
  - RAM - 8 GB
  - Hard Disk - 2 TB
- ✓ **Software:**
  - Image processing software
  - Programming tools (Python)
  - Integrated development environment (IDE) : Jupiter note book
  - Python Libraries: Pandas, NumPy, Keras

5.2 RISK MANAGEMENT

5.2.1 Risk Identification

1. **Technical Risks:**

- Difficulty in achieving accurate fruit image due to noise or interference in the camera.
- Inability to achieve real-time performance due to limitations in hardware or software processing capabilities.
- Challenges in network issues in ruler areas.

2. **Operational Risks:**

- User error or mishandling of the equipment, resulting in damage to the hardware.
- System failure or downtime due to hardware or software issues, leading to delays or disruption in operations.
- Inability to adapt to changing lighting or environmental conditions, leading to reduced accuracy in while capturing images.

3. **Cost and Resource Risks:**

- Cost overruns due to unexpected expenses, such as additional hardware or software requirements.
- Regulatory or compliance requirements that could increase costs or limit the scope of the project.

5.2.2 Risk Analysis

1. **Technical Risks:**

   - Accuracy of disease detection: One of the most significant risks in a fruit disease detection project is the accuracy of the system in detecting diseases. A model that is not accurate enough can result in incorrect recommendations and ineffective pesticide use.
   - Generalization to new fruits: The system may not generalize well to new types of fruits that were not included in the training dataset. The accuracy of the model may be limited to the specific fruits that were used to train it.
   - Limited training data: Limited training data can lead to poor model performance and low accuracy in detecting fruit diseases.

2. **Operational Risks**

   - User acceptance: The system's success depends on its acceptance and adoption by farmers. Therefore, it is crucial to consider farmers' needs and preferences when designing the system's user interface and functionality.
   - Training and support: Farmers may require training and support to use the system effectively. Insufficient training and support can lead to low adoption rates and ineffective use of the system.
   - Regulatory compliance: The use of pesticides raises ethical concerns related to the environmental impact, health hazards, and regulation compliance. The system should take into account these considerations and provide responsible recommendations.

- Security and privacy: The system may collect sensitive information about farmers and their crops. The system must have robust security measures to protect this information from unauthorized access or theft.

3. **Cost and Resource Risks:**

- Data acquisition: Acquiring a diverse and comprehensive dataset of fruit images with associated disease labels can be expensive and time-consuming.
- Hardware and software requirements: The system may require specialized hardware, such as high-resolution cameras or processing units, and software licenses, which can be costly.
- Expertise: Developing and implementing the project requires expertise in both agriculture and machine learning. Hiring experts in both fields can be costly.
- Deployment and maintenance: The deployment and maintenance of the system may require ongoing costs, such as server hosting, electricity, and software updates.
- Training and support: Providing training and support to farmers can be costly, particularly in remote or rural areas.
- Time constraints: The development and implementation of the project can be time-consuming, delaying its effectiveness and increasing its cost.

5.2.3 Overview of Risk Mitigation, Monitoring, Management

Risk mitigation, monitoring and management strategies can help to address the risks identified for the fruit disease detection and pesticide suggetion. Here are some possible approaches:

1. **Technical Risks:**

To mitigate, monitor, and manage technical risks in a fruit disease detection project, the following strategies can be considered:

- Accuracy of disease detection: The accuracy of disease detection can be improved by using a larger and more diverse dataset for training and validation, applying data augmentation techniques to increase dataset variability, and performing rigorous testing on a diverse range of fruits and environmental conditions.

- Generalization to new fruits: To improve generalization to new fruits, transfer learning techniques can be applied. This involves training the model on a large and diverse dataset of fruits and then fine-tuning the model on the target fruit.

- Limited training data: To overcome the limitation of limited training data, data augmentation techniques such as rotation, scaling, and flipping can be used to increase the dataset's size and variability. Additionally, transfer learning can be used to leverage pre-trained models on similar tasks.

- Hardware requirements: To address hardware requirements, alternative hardware solutions can be considered, such as using low-resolution cameras or cloud-based processing units. The system can also be designed to work on a range of hardware options to increase accessibility and affordability.

- Robustness to variability: The robustness of the system can be improved by incorporating uncertainty quantification techniques and multi-modal approaches, such as combining visual and spectral analysis.

- Pesticide suggestions: To ensure accurate and responsible pesticide suggestions, the system can be designed to comply with relevant regulations and safety standards. Additionally, involving domain experts, such as agronomists and plant pathologists, in the development of the system can increase its accuracy and effectiveness.

2. Operational Risks:

To mitigate, monitor, and manage operational risks in a fruit disease detection project, the following strategies can be considered:

- Accessibility: The system should be accessible to farmers, regardless of their location or technical expertise. To ensure accessibility, the system can be designed with a user-friendly interface and made available through mobile applications or web-based platforms. Additionally, the system should be designed to work on a range of hardware options, such as smartphones and low-cost tablets.

- Internet connectivity: To address internet connectivity issues in remote or rural areas, the system can be designed to work offline or with limited internet connectivity. This can be achieved through techniques such as edge computing and local storage.

- User training: To ensure farmers can use the system effectively, training programs and materials can be developed, covering topics such as image capture, data input, and interpretation of results. Additionally, providing ongoing technical support can help farmers address any issues that may arise.

- User adoption: To ensure user adoption, the system should be designed with the needs and preferences of farmers in mind. This can be achieved through user-centered design techniques, such as conducting user research and involving farmers in the design process.

- Ethical and legal considerations: The system should comply with relevant ethical and legal considerations, such as data privacy regulations and ethical guidelines for machine learning. Additionally, involving stakeholders, such as farmers and regulatory bodies, in the development and implementation of the system can help ensure its compliance with relevant ethical and legal standards.

3. Cost and Resource Risks:

To mitigate, monitor, and manage cost and resource risks in a fruit disease detection project, the following strategies can be considered:

- Data acquisition: Collecting and annotating a large and diverse dataset can be costly and time-consuming. To address this, crowdsourcing and community-based data annotation platforms can be used to reduce the cost and time required to acquire and annotate data.

- Hardware requirements: The hardware required for capturing and processing images can be expensive. To address this, alternative hardware solutions can be considered, such as using low-cost cameras or smartphones. Additionally, cloud-based processing units can be used to reduce the hardware requirements of the system.

- Model development and training: Developing and training deep learning models can be computationally expensive. To address this, cloud-based services can be used to leverage high-performance computing resources. Additionally, transfer learning can be used to reduce the amount of training required for the model to achieve high accuracy.

- Implementation and maintenance: Implementing and maintaining the system can be costly. To address this, open-source tools and frameworks can be used to reduce the cost of software development. Additionally, involving local communities and stakeholders in the maintenance of the system can reduce the maintenance costs and increase the sustainability of the project.

- Pesticide suggestions: Providing accurate and responsible pesticide suggestions requires access to a large and diverse database of pesticides and their effectiveness against specific diseases. To address this, partnerships can be established with relevant stakeholders, such as agricultural research institutions and pesticide manufacturers, to access their databases and expertise.

5.3 PROJECT SCHEDULE

5.3.1 Project Task Set:

1. Data collection: Gather a diverse dataset of images of healthy and diseased fruit.
2. Data preprocessing: Resize and normalize the images to ensure consistency and enhance model performance.
3. Data augmentation: Use techniques such as image flipping, rotation, and scaling to increase the diversity of the dataset and improve model generalization.
4. Model selection: Choose a deep learning model architecture that is suitable for image classification, such as a convolutional neural network (CNN).
5. Transfer learning: Use transfer learning techniques to leverage pre-trained models and reduce the amount of training required for the model to achieve high accuracy.
6. Model training: Train the deep learning model on the annotated dataset using a suitable loss function and optimizer.
7. Model evaluation: Evaluate the trained model on a holdout dataset to measure its accuracy and performance.

8. Model optimization: Adjust the model hyperparameters, such as learning rate and regularization, to improve its performance.
9. Deployment: Implement the trained model as an application that can take in images of fruit and output disease predictions.
10. User testing: Conduct user testing to evaluate the usability and effectiveness of the application.
11. Iterative improvement: Use feedback from users and model performance metrics to continuously improve the application and model performance.
12. Pesticide suggestion integration: Integrate a database of pesticides and their effectiveness against specific diseases to provide responsible pesticide suggestions based on the model's predictions.
13. System maintenance: Perform regular maintenance tasks, such as updating software dependencies and monitoring system performance, to ensure the long-term sustainability of the project.

5.3.2 Task Network:

Research and study of different fruit disease --> gather dataset for different fruit disease --> analyse different fruits disease data --> Build Deep learning model --> develop graphical user interface --> connect model with graphical user interface --> deploy developed software--> Test our software in production --> Create documentation.

5.3.3 Timeline Chart:

Task	Duration (weeks)
Research and study of different fruit disease	1
gather dataset for different fruit disease	2
analyse different fruits disease data	2
Build Deep learning model	4
develop graphical user interface	1

connect model with graphical user interface	3
deploy developed software	2
Integrate system	2
Test our software in production	2
Create documentation	1

Table 5. 1 Timeline Chart

5.4 TEAM ORGANIZATION

5.4.1 Team structure

- Our project team consists of four members :
    - Samvedya Jedhedeshmukh
    - Sahil Kachole
    - Nishant Parakh
    - Umesh Chaudhari
- We all team members are Computer Engineering students at Pimpri Chinchwad College of Engineering, Pune.
- Prof. Madhura Kalbhor is the project guide and will be mentoring the team during entire project duration.
- Nishant is the Team Lead and will be responsible for communication with project guide as well as college.
- All team members have decided to work collaboratively for requirement identification, Planning of the project, System Design, Perform testing and validation.
- Sahil and Umesh will be coordinating among all team members for setting up team meetings.
- We all will work on developing a fruit detection module as well as GUI for the Project.
- We all will work on developing machine learning model.
- Sahil and Samvedya will work on a research paper about project and all team members will assist him in this task.
- Sahil will assist Samvedya and Umesh for coding.
- All team members will collectively work for developing documentation of the project.

5.4.2 Management Reporting and Communication:

- The Project Manager will be responsible for managing the reporting and communication within the team and with stakeholders.
- For effective communication between team members there should be a communication channel where each team member can communicate so we created a WhatsApp group.
- Dividing tasks among team members was one of the important tasks so that project outcomes could be achieved within defined duration. Team Lead was the one who divide the task among the team members.
- For sharing the weekly updates and milestones, setting up weekly meetings with team members was one of the effective ways of communication.
- All the communications and decisions to be made during team meeting only after approval of all team members and should be well documented.
- Project team will have a weekly meeting with project guide to discuss the progress of the project.

CHAPTER 6

PROJECT IMPLEMENTATION

6. PROJECT IMPLEMENTATION

6.1 OVERVIEW OF PROJECT MODULES

1. Image Processing Module: This module is responsible for capturing images of fruit crops and processing them to identify the presence of diseases or pests. It may use various image processing techniques like segmentation, feature extraction, and classification to detect the diseases accurately.

2. Disease Detection Module: This module uses machine learning algorithms to identify the specific disease affecting the fruit crops based on the images captured by the image processing module. The algorithms may use a pre-trained deep neural network to classify the image into different classes, each representing a particular disease.

3.Pesticide Recommendation Module: Once the disease is identified, this module recommends the most appropriate pesticide to treat the specific disease. The recommendation is based on the type of disease, crop type, and other factors like the environmental conditions and the stage of crop growth.

4.User Interface Module: This module provides a graphical user interface for the end-users to interact with the system. It displays the results of the image processing and disease detection modules and provides recommendations for the appropriate pesticide to be used.

5.Database Module: This module stores the data related to the different types of diseases that can affect fruit crops, the pesticides that can be used to treat these diseases, and the crop types that are vulnerable to different diseases. It also stores the images captured by the image processing module for further analysis and training the machine learning algorithms.

6.Reporting Module: This module generates reports on the status of the crop, the presence of any diseases or pests, and the recommended course of action. These reports may be used by the farmers or other stakeholders to take appropriate actions to protect the crops and maximize the yield.

6.2 TOOLS AND TECHNOLOGIES USED

The tools and technologies used in an fruit disease and recommendation system can vary depending on the specific imlementation, but some common ones include:

Tools:

1. Machine Learning Frameworks: Popular machine learning frameworks such as TensorFlow, PyTorch, and scikit-learn can be used to develop the disease detection and pesticide recommendation modules. These frameworks provide pre-trained models and algorithms that can be customized to fit the specific requirements of the fruit crop domain.

2. Computer Vision Libraries: Computer vision libraries such as OpenCV and scikit-image provide a wide range of image processing functions that can be used to preprocess the images captured by the system. These libraries can be used for tasks such as image segmentation, feature extraction, and object detection.

3. Web Frameworks: Web frameworks such as Flask and Django can be used to develop the user interface module of the system. These frameworks provide a simple and intuitive way to create web applications that can be accessed from any device with an internet connection.

4. Cloud Services: Cloud services such as Amazon Web Services (AWS) and Microsoft Azure can be used to store the large amount of data generated by the system and to deploy the machine learning models and web applications. These services provide scalable and cost-effective solutions for hosting and managing the system.

5. Programming Languages: The system can be developed using programming languages such as Python, Java, and C++. Python is a popular choice for developing machine learning and web applications, while Java and C++ can be used for developing high-performance modules.

6. Database Management Systems: Database management systems such as MySQL and PostgreSQL can be used to store and manage the data generated by the system. These systems provide efficient and reliable ways to store and retrieve data.

Techniques:

1. **Image processing:**

Image processing is a field of computer science and engineering that deals with the analysis and manipulation of digital images. It involves developing algorithms and techniques that allow computers to extract useful information from digital images and to enhance their quality, clarity, and detail. The goal of image processing is to improve the visual appearance of images, extract relevant information, and transform images into more useful formats. Image processing is used to convert the RGB image into grayscale so that laser line is analysed so that obstacle is detected.

2. **Computer vision:**

   Computer vision is an incredibly important field of study within computer science and engineering that involves the development of algorithms and techniques that allow computers to interpret and understand visual data from the world around them. This includes the analysis and processing of images, videos, and other types of visual data, with the goal of enabling computers to recognize objects, detect patterns, and make decisions based on what they "see". Computer vision has a wide range of applications, from industrial automation and robotics to medical imaging and surveillance. Its continued development has the potential to revolutionize many aspects of our daily lives, making it a field that commands great respect and admiration.

3. **Hyperparameter tuning:**

   Hyperparameter tuning is a critical process in machine learning that involves searching for the best combination of hyperparameters to optimize the performance of a model. Hyperparameters are variables that are set before the training of a machine learning model and control its learning process. They determine the model's architecture, such as the number of hidden layers and nodes, learning rate, regularization strength, and batch size, among others. The process of hyperparameter tuning involves selecting the best hyperparameter values that lead to the best performance of the model, such as the highest accuracy or the lowest error rate. This process requires experimenting with different combinations of hyperparameters and evaluating the model's performance on a validation set to determine which combination provides the best results. Hyperparameter tuning is a crucial step in machine learning as it can significantly impact the performance of a model. By selecting the optimal hyperparameters, a model can achieve better accuracy, lower error rates, and faster convergence.

Technologies:

cv2 (OpenCV):

Python statement: import cv2

Description: OpenCV is a popular computer vision library used for image processing, video analysis, and computer vision tasks. It provides a wide range of functions and algorithms for manipulating and analyzing images and videos.

Dense:
Python statement: from tensorflow.keras.layers import Dense
Description: Dense is a layer type in deep learning models that represents a fully connected neural network layer. It connects each neuron in the current layer to every neuron in the previous and next layers, enabling information flow between them.

Dropout:
Python statement: from tensorflow.keras.layers import Dropout
Description: Dropout is a regularization technique used in deep learning models to prevent overfitting. It randomly sets a fraction of the input units to 0 at each training update, which helps to reduce the dependence on specific input features and encourages the model to learn more robust and generalized representations.

Flatten:
Python statement: from tensorflow.keras.layers import Flatten
Description: Flatten is a layer type used to convert multidimensional data into a one-dimensional vector. It is often used as a transition layer between convolutional layers and fully connected layers in convolutional neural networks. It reshapes the input tensor into a flat vector format, which can be fed into subsequent fully connected layers.

GlobalAveragePooling2D:
Python statement: from tensorflow.keras.layers import GlobalAveragePooling2D
Description: GlobalAveragePooling2D is a layer type used for spatial data reduction in convolutional neural networks. It applies average pooling across spatial dimensions (height and width) of the input feature maps, resulting in a fixed-size output regardless of the input size. This technique helps in reducing the number of parameters and computational complexity while retaining important spatial information.

GridSearchCV:
Python statement: from sklearn.model_selection import GridSearchCV
Description: GridSearchCV is a technique for hyperparameter tuning in machine learning models. It exhaustively searches over a predefined set of hyperparameters to find the best combination that maximizes model performance. It evaluates and compares models trained on different hyperparameter configurations using cross-validation.

ImageDataGenerator:
Python statement: from tensorflow.keras.preprocessing.image import ImageDataGenerator
Description: ImageDataGenerator is a utility in TensorFlow/Keras used for data augmentation and image preprocessing. It generates augmented versions of images on-the-fly during model training, enabling the model to generalize better by seeing variations of the original images. It can perform operations like rotation, zooming, flipping, and normalization.

KNeighborsClassifier:
Python statement: from sklearn.neighbors import KNeighborsClassifier

Description: KNeighborsClassifier is a classification algorithm based on the k-nearest neighbors approach. It assigns labels to unseen data points based on the labels of their k nearest neighbors in the training set. The value of k is a hyperparameter that determines the number of neighbors considered.

matplotlib:
Python statement: import matplotlib.pyplot as plt
Description: Matplotlib is a widely used data visualization library in Python. It provides a comprehensive set of functions for creating various types of plots, charts, and graphs. It is often used to visualize data distributions, relationships, trends, and other patterns.

np (NumPy):
Python statement: import numpy as np
Description: NumPy is a fundamental library for numerical computing in Python. It provides efficient data structures, high-performance multidimensional arrays, and a collection of mathematical functions for performing computations. NumPy is widely used for array manipulation, mathematical operations, linear algebra, random number generation, and more.

os:
Python statement: import os
Description: The os module is a part of the Python standard library and provides a way to interact with the operating system. It offers functions for performing operations such as file and directory manipulation, path handling, environment variables, and more. It is commonly used to navigate and manipulate file systems, create or delete directories, check file existence, and manage paths.

PIL (Python Imaging Library):
Python statement: import PIL
Description: PIL is a widely used library in Python for image processing tasks. It provides a variety of functions and methods to open, manipulate, and save different image formats. PIL allows tasks such as resizing, cropping, rotating, filtering, and applying various transformations to images. It is often used in computer vision projects and for preprocessing images before feeding them into machine learning models.

pickle:
Python statement: import pickle
Description: The pickle module in Python provides a way to serialize (convert objects into a byte stream) and deserialize (reconstruct objects from the serialized byte stream) Python objects. It is commonly used for saving and loading machine learning models, saving intermediate results, or storing complex data structures. Pickle allows objects to be stored in a file or transmitted over a network.

RandomForestClassifier:
Python statement: from sklearn.ensemble import RandomForestClassifier
Description: RandomForestClassifier is an ensemble learning method based on decision trees for classification tasks. It combines multiple decision trees and uses averaging or voting to make predictions. Random forests are

known for their effectiveness in handling high-dimensional data, capturing complex relationships, and reducing overfitting.

ResNet50:
Python statement: from tensorflow.keras.applications.resnet50 import ResNet50
Description: ResNet50 is a pre-trained convolutional neural network architecture that is commonly used for image classification tasks. It consists of 50 layers and has been trained on a large dataset (e.g., ImageNet) to recognize a wide range of objects. ResNet50 has achieved excellent performance in various computer vision challenges and is often used as a feature extractor or fine-tuned for specific tasks.

seaborn:
Python statement: import seaborn as sns
Description: seaborn is a statistical data visualization library based on matplotlib. It provides a high-level interface for creating informative and visually appealing statistical graphics. seaborn offers various types of plots, including scatter plots, bar plots, histograms, heatmaps, and more. It is often used to explore and analyze data, uncover patterns, and communicate insights.

Sequential:
Python statement: from tensorflow.keras.models import Sequential
Description: Sequential is a model type in the Keras API, which is a high-level deep learning framework. The Sequential model is a linear stack of layers, where each layer is added sequentially. It is commonly used for building deep learning models with a single input and a single output, such as feedforward neural networks.

SVC (Support Vector Classifier):
Python statement: from sklearn.svm import SVC
Description: SVC, or Support Vector Classifier, is a classification algorithm based on support vector machines (SVM). It aims to find the best hyperplane that separates different classes in the input space. SVC is effective in handling both linear and non-linear classification tasks and can handle high-dimensional data efficiently.

tensorflow:
Python statement: import tensorflow as tf
Description: tensorflow is an open-source deep learning framework widely used for building and training machine learning models. It provides a comprehensive ecosystem of tools, libraries, and APIs for developing various types of deep learning models, including neural networks. tensorflow supports both CPU and GPU computations and offers high-level abstractions for easier model development and deployment.

Tree:
Python statement: from sklearn.tree import DecisionTreeClassifier
Description: Tree refers to the DecisionTreeClassifier, which is a classification algorithm based on decision tree learning. It creates a model that predicts the class label of an input based on a series of binary decisions at each internal node. Decision trees are interpretable and can handle both categorical and numerical features.

VotingClassifier:
Python statement: from sklearn.ensemble import VotingClassifier
Description: VotingClassifier is a model that combines multiple individual classifiers to make predictions. It aggregates the predictions from each classifier and uses majority voting or weighted voting to determine the final prediction. VotingClassifier is useful when leveraging the strengths of different classifiers and can often improve the overall predictive performance.

Flask:
Python statement: from flask import Flask, request, render_template
Description: Flask is a micro web framework written in Python that allows developers to build web applications quickly and easily. It provides simple and easy-to-use tools for routing HTTP requests, rendering HTML templates, handling form data, and more. Flask is lightweight, flexible, and modular, making it a popular choice for building small to medium-sized web applications.

HTML and CSS:
HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) are two essential technologies for web development. HTML is used to structure content on a web page, while CSS is used to style and visually enhance the appearance of HTML elements. HTML is a markup language that uses tags to define the structure of a web page. Tags are used to create headings, paragraphs, links, images, and other elements on a web page. HTML is the foundation of any web page and is used to provide structure and semantic meaning to the content.

CSS, on the other hand, is used to style the content created using HTML tags. It allows developers to define how HTML elements are displayed on a web page, such as their color, font size, margin, padding, and layout. CSS can be used to create visually appealing web pages and make them more user-friendly. Together, HTML and CSS form the basis of modern web development, and both are essential skills for any web developer. By using HTML and CSS effectively, developers can create beautiful and functional web pages that provide a great user experience.

6.3 ALGORITHM DETAILS

Transfer learning:

In the realm of deep learning, transfer learning is a potent method that enables models to use knowledge obtained from pre-training on substantial datasets to enhance performance on particular tasks. For transfer learning in this study, we used the well-known ResNet50 architecture. With regard to a variety of computer vision tasks, such as picture classification, object recognition, and image segmentation, ResNet50, a deep convolutional neural network, has produced state-of-the-art results. We took use of ResNet50's learned feature representations, which capture high-level visual patterns and semantic information, by using it as a pre-trained model. We improved performance while using less training time and computer resources by fine-tuning the pre-trained ResNet50 model on our target dataset.We demonstrate the effectiveness and efficiency of transfer learning with ResNet50 in enhancing the precision and convergence rate of our deep learning model through numerous experiments and evaluations.

Convolutional Neural Networks (CNNs):

Convolutional Neural Networks (CNNs) have emerged as a dominant deep learning architecture for analyzing and processing visual data, revolutionizing several research areas including computer vision, image recognition, and object detection. CNNs are specifically designed to handle grid-like data, such as images, by leveraging a hierarchical structure of convolutional layers. These layers consist of small filters that slide across the input data, capturing local patterns and features through convolution operations. By stacking multiple convolutional layers, CNNs can learn increasingly complex representations, enabling them to capture intricate visual patterns and hierarchies of features. The network's architecture is typically composed of alternating convolutional and pooling layers, which facilitate translation equivariance and reduce spatial dimensions, respectively. The final layers often include fully connected layers and a softmax activation function for classification tasks. One of the major strengths of CNNs is their ability to automatically learn and extract relevant features from raw data, eliminating the need for manual feature engineering. This data-driven approach has significantly improved the accuracy and efficiency of various image-related research tasks, such as image classification, object detection, and semantic segmentation. CNNs have also been successfully applied in domains beyond computer vision, including natural language processing and speech recognition, demonstrating their versatility and wide-ranging impact in research.

Ensemble Learning:

Ensemble learning is a powerful approach in machine learning where multiple models, called base models or weak learners, are combined to form a stronger and more accurate model, known as the ensemble model. The idea behind ensemble learning is that by aggregating the predictions of multiple models, the ensemble can often outperform any individual model.

In the specific case you mentioned, the ensemble is created using the VotingClassifier, which is a type of ensemble method. It combines the predictions of different classifiers, including a Decision Tree (classifier_1), Random Forest (classifier_2), K-Nearest Neighbors (classifier_3), and Support Vector Machine (classifier_4), using a voting scheme.

The voting scheme can be either "hard" or "soft." In the case of "hard" voting, each classifier in the ensemble gives a single vote for the predicted class label. The class label that receives the majority of votes is selected as the final prediction. In "soft" voting, the classifiers provide a probability distribution over the class labels, and the predicted class label is determined by averaging these probabilities and selecting the one with the highest average probability.

By combining different classifiers with diverse strengths and weaknesses, ensemble learning can improve the overall performance of the model. It helps to reduce overfitting, increase stability, and handle noise or uncertainty in the data. Ensemble learning is widely used in various research areas and has demonstrated its effectiveness in improving prediction accuracy and robustness in many real-world applications.

Random Forest:

Random Forest is a powerful and widely used machine learning algorithm that has gained significant popularity due to its ability to handle complex classification and regression tasks. It belongs to the ensemble learning family, where multiple decision trees are combined to create a robust and accurate model. The algorithm operates by constructing a multitude of decision trees during the training phase, each independently grown on a random subset of the training data and features. This process introduces randomness and diversity into the model, mitigating overfitting and improving generalization performance. During prediction, the Random Forest aggregates the predictions of all individual trees, typically using majority voting for classification problems or averaging for regression problems. This ensemble approach enhances the model's accuracy, stability, and resistance to noise and outliers. Moreover, Random Forest provides valuable insights into feature importance, enabling researchers to assess the relative contribution of different variables in the predictive process. These characteristics make Random Forest an effective and versatile tool for various applications in research, such as bioinformatics, finance, and natural language processing.

Decision trees:

Decision trees are widely used and interpretable machine learning models that excel in both classification and regression tasks. They mimic the human decision-making process by constructing a tree-like structure, where each internal node represents a feature or attribute, and each leaf node corresponds to a class label or predicted value. The tree is built through a recursive process, where at each step, the algorithm selects the most informative feature to split the data based on certain criteria, such as information gain or Gini impurity. This recursive splitting continues until a stopping condition is met, such as reaching a maximum depth or minimum number of samples in a leaf node. Decision trees offer several advantages in research applications, including their simplicity, interpretability, and ability to handle both numerical and categorical data. They can capture complex relationships and interactions between features, enabling the identification of important variables and decision rules. Decision trees can also handle missing data and are robust to outliers. However, they can suffer from overfitting when the tree becomes too deep and specialized to the training data. To mitigate this issue, ensemble methods like Random Forests or Gradient Boosting can be employed to combine multiple decision trees and improve overall performance. Decision trees have found applications in diverse fields such as healthcare, finance, and social sciences, providing valuable insights and predictive capabilities to researchers.

Support Vector Machines (SVMs):

Support Vector Machines (SVMs) are powerful and versatile machine learning models widely used for both classification and regression tasks. SVMs are particularly effective in solving complex and high-dimensional problems. The main objective of an SVM is to find the optimal hyperplane that separates different classes in the feature space. The hyperplane is determined by selecting a subset of training examples called support vectors, which are the data points closest to the decision boundary. SVMs can handle linearly separable as well as non-linearly separable data by leveraging kernel functions. These functions transform the input data into a higher-dimensional space, where it becomes easier to find a linear decision boundary. Common kernel functions include linear, polynomial, Gaussian radial basis function (RBF), and sigmoid. SVMs have several advantages, including their ability to handle high-dimensional data, their resistance to overfitting, and their ability to handle both linear and non-linear classification tasks. Moreover, SVMs provide a unique property called the "kernel trick," which allows implicit mapping of data into higher-dimensional feature spaces without explicitly computing the transformed feature vectors. This capability makes SVMs computationally efficient, even with large datasets. However, SVMs can be sensitive to the choice of hyperparameters, such as the penalty parameter C and the kernel parameters. Therefore, careful parameter tuning is essential for optimal performance. SVMs have been successfully applied in various research domains, such as image classification, text classification, bioinformatics, and finance, making them a valuable tool for researchers in a wide range of disciplines.

K-Nearest Neighbors (KNN):

K-Nearest Neighbors (KNN) is a popular and versatile machine learning algorithm used for both classification and regression tasks. KNN is a non-parametric algorithm, meaning it does not make any assumptions about the underlying data distribution. Instead, it relies on instance-based learning, where predictions are made based on the similarity between the new data point and its k nearest neighbors in the training set. In the case of classification, the class label of the majority of the k nearest neighbors is assigned to the new data point. For regression tasks, the predicted value is typically computed as the average or weighted average of the target values of the k nearest neighbors. One of the main advantages of KNN is its simplicity and ease of implementation. It is a lazy learning algorithm, meaning it does not involve an explicit training phase but directly uses the training data during prediction. KNN is effective in capturing complex relationships in the data and can handle non-linear decision boundaries. However, its performance may be sensitive to the choice of the number of neighbors (k) and the distance metric used to measure similarity. Proper selection of these parameters is crucial to ensure optimal performance. KNN is often used in research areas such as image recognition, recommender systems, and anomaly detection. Additionally, KNN can be easily combined with other machine learning algorithms and techniques to form more advanced models and improve overall performance.

Steps involved in generating the Model are as follows:

In this study, we developed a fruit disease detection system using the ResNet50 pre-trained model in the TensorFlow/Keras framework. The methodology can be divided into the following steps:

Data Preprocessing:

The input images were pre-processed using the ImageDataGenerator class from TensorFlow/Keras. The pixel values of the images were scaled down by a factor of 1/252.

Dataset Preparation:

The training and validation datasets were prepared using the ImageDataGenerator class, which loaded the images from the respective directories. The dataset was resized to a uniform size of 180x180 pixels to ensure consistency. The dataset was batched with a batch size of 23 for efficient training.

Exploratory Data Analysis:

A subset of images from the training dataset was visualized to gain insights into the dataset and understand the fruit disease classes.

Model Architecture:

We used the ResNet50 pre-trained model as the backbone of our fruit disease detection system. The pre-trained model was initialized with weights trained on the ImageNet dataset. We froze the weights of the pre-trained layers to prevent them from being updated during training. The output of the pre-trained model was flattened, followed by a fully connected layer with 512 units and ReLU activation. The final output layer consisted of 4 units with softmax activation, representing the different fruit disease classes.

Model Training:

The model was compiled with the Adam optimizer and a learning rate of 0.001. The loss function used was sparse categorical cross-entropy, suitable for multi-class classification tasks. The model was trained for 11 epochs using the training dataset, with validation performed using the separate validation dataset.

Evaluation and Performance Metrics:

The accuracy and loss curves were plotted to assess the model's performance during training. The accuracy and loss values were recorded for both the training and validation sets at each epoch.

Model Testing and Performance Evaluation:

The trained model was evaluated on a separate test dataset obtained from the same fruit disease dataset. The test accuracy and loss were calculated to measure the generalization performance of the model.

Results Analysis:

The obtained accuracy, loss, and other relevant metrics were analyzed to assess the effectiveness of the fruit disease detection system. The performance of the developed model was compared with other existing approaches and benchmarks, if applicable. The presented methodology provides a comprehensive framework for fruit disease detection using the ResNet50 pre-trained model. The next section will discuss the experimental results and provide an in-depth analysis of the model's performance.

CHAPTER 7

SOFTWARE TESTING

7. SOFTWARE TESTING

The purpose of testing is to identify flaws. Searching for flaws or vulnerabilities in a work product is the process of testing. It is possible to test individual parts, subassemblies, assemblies, and/or a finished product. It is the process of ensuring that software fulfills user expectations, adheres to standards, and does not malfunction in a way that is unacceptable. There are a variety of tests available.

7.1 TYPE OF TESTING

White-box testing is a sort of software testing that looks at an application's internal architecture rather than its functionality. It is also known as clear box testing, glass box testing, transparent box testing, and structural testing (i.e. black-box testing). White-box testing creates test cases by fusing a knowledge of the system's internal workings with programming skills. The tester selects inputs to explore potential code routes and identify suitable outputs. In-circuit testing is comparable to circuit node testing (ICT).

It is possible to perform white-box testing at any stage of the software development process. However, this is typically carried out at the unit level. You can check the paths inside a unit, the paths between units during integration, and the paths between subsystems during a system-level test. While this method of test design can find a variety of errors or issues, it might overlook requirements that haven't been met or specifications that haven't been applied.

Unit Testing :

Analyzing each unit or component of the software application. Check to see if software components meet the specified requirements. developer performed. We tested each individual algorithm in its unit. This sort of testing uses white boxes.

Integration Testing:

The test engineer performs group tests on the software's components or modules to evaluate the application's overall functionality. Testing using a black box. Each algorithm is assembled and put to the test.

Functional Testing:

Functional testing is mostly used to determine whether a programme meets customer requirements. Black-box testing is the category in which this belongs. All of the algorithms have been tested for functionality.

Performance Testing :

- Load testing - Testing under load involves gradually increasing the number of users using the application while monitoring its speed. (where exactly breaking application that benchmark need to find if it is breaking frequently then tell the developer to fix it).
- Stress testing involves abruptly increasing or decreasing load and evaluating application speed.
- Scalability testing is the process of testing an application's performance by increasing or decreasing the load on particular scales.
- Stability: Testing an application's performance by applying a load to it for a predetermined amount of time. Black-box testing is the category it falls under. We looked at how well the model worked from the user's point of view.
- Usability testing looks at how easy it is for the end user to use and understand the application. This is considered to be black-box testing. Our model's usability and ease of use were evaluated.

Black Box Testing:

Testing in the dark Instead of examining an application's core structures or processes, black-box testing looks at its functionality (what it accomplishes) (see whitebox testing). Each stage of software testing can use this test approach. incorporating acceptance testing as well as system, integration, and unit testing. Unit testing may be substituted. Nonetheless, it actually includes most higher-level testing, if not all of it.

7.2 TEST CASES & TEST RESULTS

System testing is a type of black-box testing. Every functionality is tested during system testing in accordance with client specifications. Software testing is the act of examining the execution of software components to find defects, mistakes, or flaws while considering all of the software's properties (reliability, scalability, portability, re-usability, and usability).

Test ID	Description	Expected Result	Passed/Failed
1	Fruit image uploaded	Disease detected	Pass
2	Fruit image not uploaded	No Disease detected	Pass
3	Fruit image with some noise	Disease detected	Pass

Table 7. 1 Test Case Table

CHAPTER 8

RESULTS

8. RESULTS

8.1 OUTCOMES:

The plants which are being cultivated should be disease free and pest free, so that people can contribute a good sum to global economy and can help farmers and agriculturalists to lead a good , wealthy as well as healthy life. These things can be literally achieved with the help of image processing and the proposed algorithm. The Use of CNN algorithms pave an easy way to detect the disease on the fruits and helps to classify the diseases from healthy fruit. From these methods and algorithms, this approach can easily identify and classify the fruits using image processing techniques. The leading objective of our project is to boost the worth of fruit disease detection.

CNNs are highly effective in image recognition and can accurately identify different types of fruit diseases. Therefore, a fruit disease detection system that uses CNNs can achieve higher accuracy than other traditional methods. Since CNNs are trained on large datasets, they are less prone to misclassification errors than human experts, who may miss subtle signs of disease. With the help of CNNs, a fruit disease detection system can quickly scan and analyze images of fruits, providing farmers with a faster and more efficient diagnosis. Early detection of diseases can prevent their spread to other crops and ultimately improve crop yield. Early detection of diseases can save farmers from incurring heavy losses caused by the spread of the disease. Additionally, the use of a fruit disease detection system can save farmers time and resources that would otherwise be spent on manual inspection. A fruit disease detection system that uses CNNs can be easily scaled up to analyze large quantities of fruits, making it suitable for commercial applications. Overall, a fruit disease detection system that uses CNNs can significantly improve the accuracy and efficiency of fruit disease detection, resulting in improved crop yield and reduced losses for farmers.

COMPARISON TABLE I

SR. NO	ALGORITHM	ACCURACY
1	Ensemble	63%
2	Decision Tree	53%
3	Random Forest Classifier	69%
4	KNN	59%
5	SVC	55%

Table 8. 1 comparison Table I

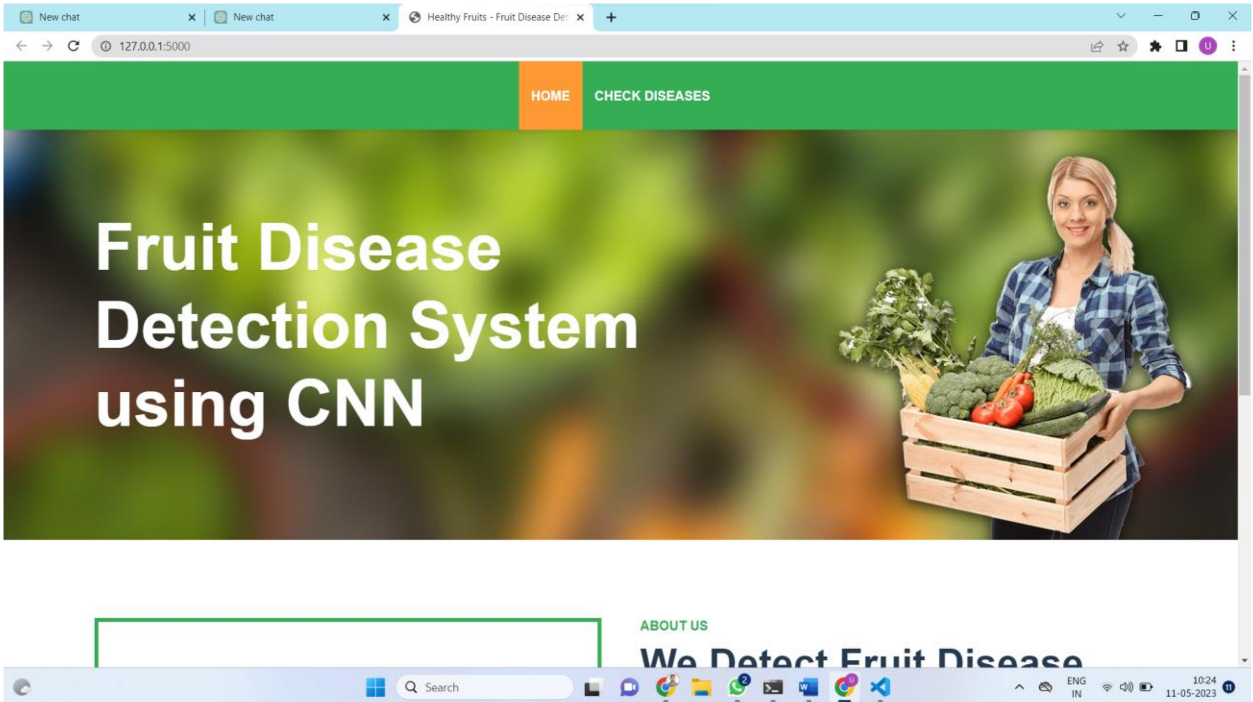
COMPARISON TABLE II

SR. NO	ALGORITHM	ACCURACY
1	ResNet50 (CNN)	80%
2	Random Forest	69%
3	Ensemble	63%
4	KNN	59%
5	SVC	55%
6	Decision Tree	53%

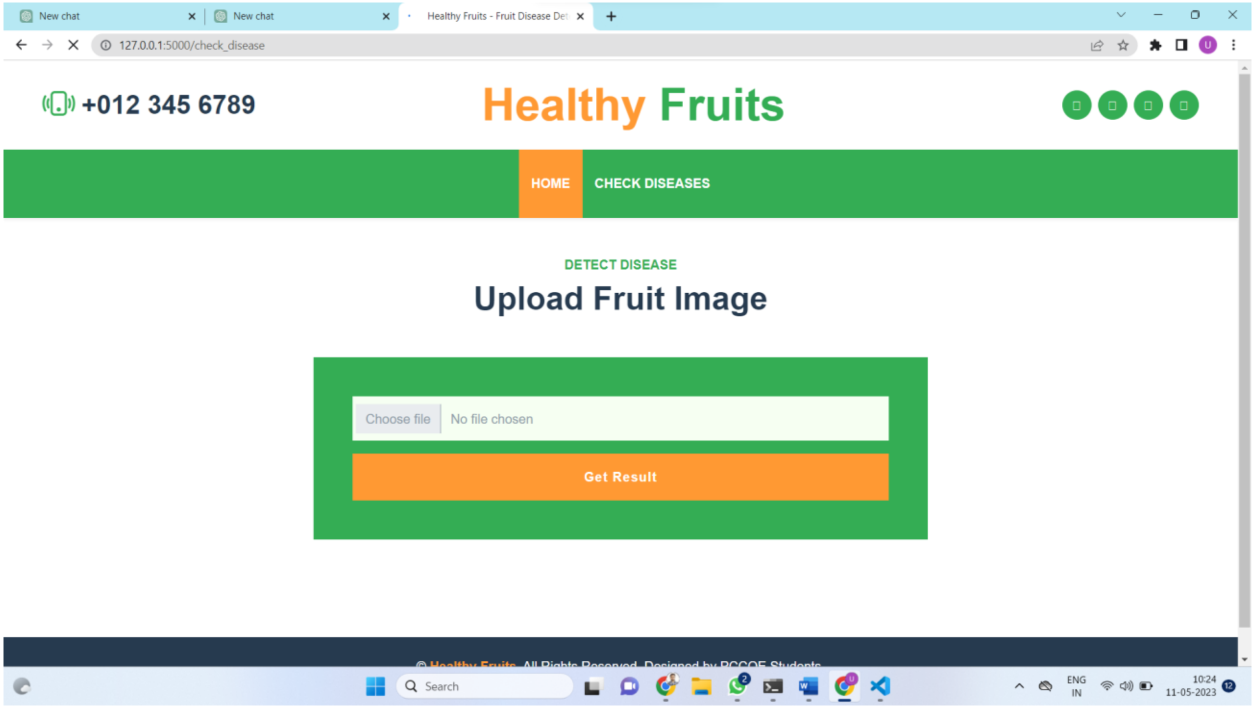
Table 8. 2 Comparison Table II

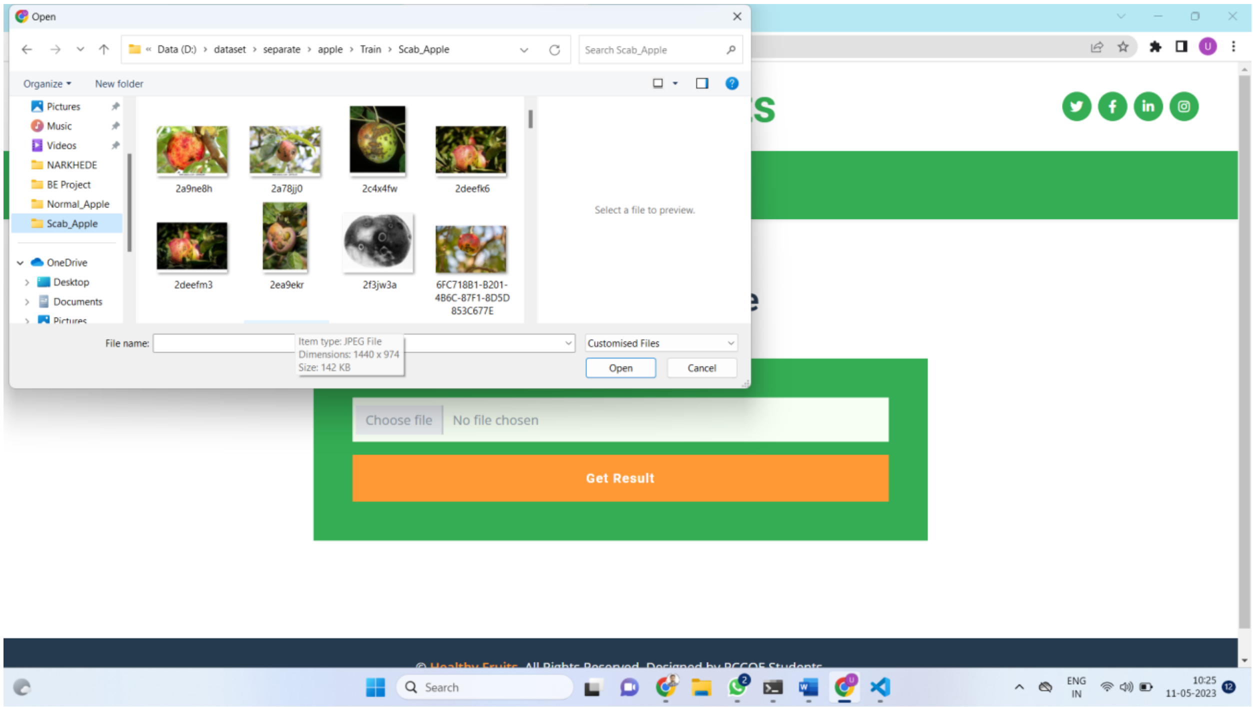
8.2 SCREEN SHOTS

1. Home Page:

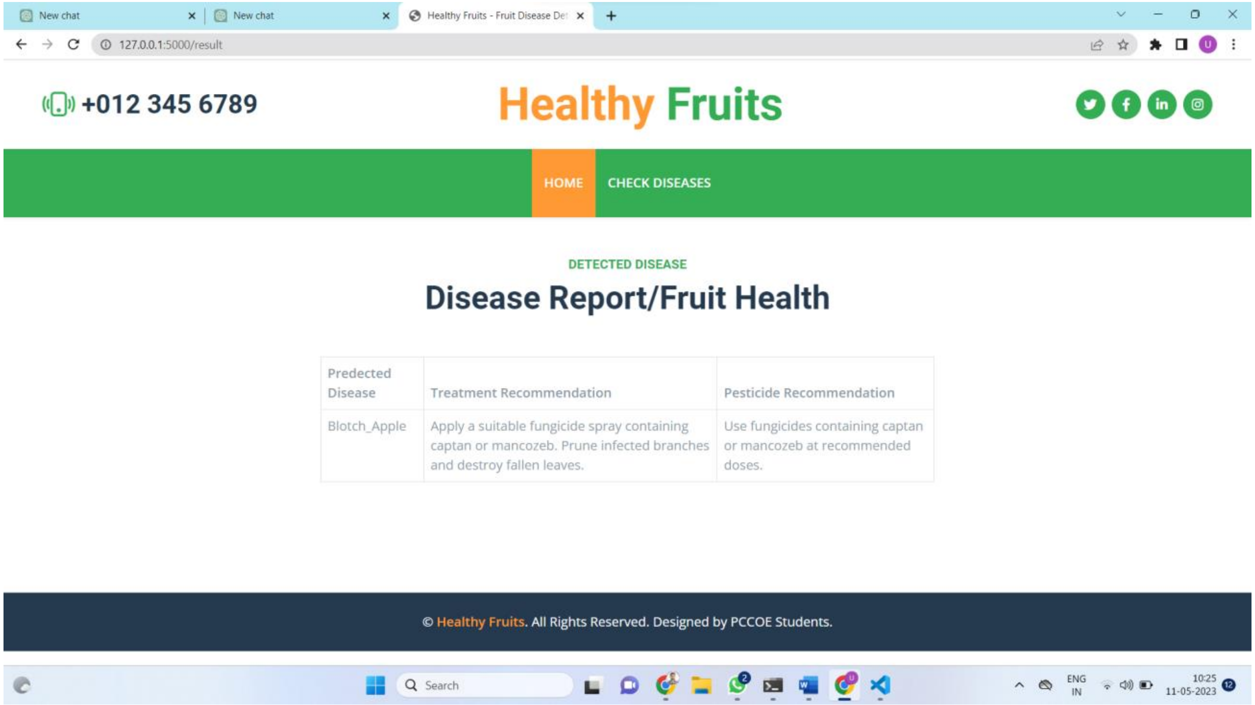


2. Interface for Uploading Fruit Image:





3. Detected Result:



CHAPTER 9

CONCLUSION

CONCLUSION

9.1 CONCLUSION:

In conclusion, a fruit disease detection system combined with pesticide suggestion can greatly benefit the agriculture industry. By utilizing advanced technologies such as machine learning, image processing, and data analysis, this system can accurately identify various diseases affecting fruits and suggest the appropriate pesticide treatment. This can save time and money for farmers by allowing them to quickly identify and address issues before they become severe. Furthermore, this system can promote the use of environmentally friendly and sustainable farming practices by recommending the use of less harmful pesticides. Overall, a fruit disease detection system with pesticide suggestion has the potential to revolutionize the way we approach fruit cultivation and help ensure the production of healthy and high-quality crops. Additionally, this system can also enhance food safety for consumers by ensuring that the fruits they consume are free of harmful diseases and toxins. By reducing the use of harmful pesticides, this system can also contribute to the preservation of biodiversity and ecosystem health. Furthermore, the integration of this system with other agricultural technologies such as precision agriculture and IoT devices can provide farmers with real-time monitoring of crop health and environmental conditions. This can further improve the efficiency and accuracy of disease detection and pesticide suggestion.

9.2 FUTURE WORK:

There are several areas of future work in fruit disease detection and pesticide recommendation systems, including:

Improving accuracy: There is a need to improve the accuracy of fruit disease detection systems and pesticide recommendation systems to reduce false positives and false negatives. This can be achieved by using more advanced machine learning algorithms and data sources, including satellite imagery, weather data, and soil data.

Integrating multiple data sources: There is a need to integrate multiple data sources, such as sensor data, remote sensing data, and weather data, to provide more comprehensive and accurate recommendations for pest management and disease control.

Developing customized solutions: There is a need to develop customized solutions for different crops, regions, and farming practices to provide tailored recommendations that take into account local conditions and farmer preferences.

Automating decision-making: There is a need to automate decision-making in fruit disease detection and pesticide recommendation systems to reduce the workload on farmers and improve the speed and accuracy of recommendations. This can be achieved by developing intelligent agents that can make decisions based on real-time data.

Addressing environmental concerns: There is a need to develop fruit disease detection and pesticide recommendation systems that address environmental concerns, such as reducing the use of pesticides, minimizing the impact on non-target species, and promoting sustainable farming practices.

Overall, the future work in fruit disease detection and pesticide recommendation systems is focused on improving the accuracy, efficiency, and sustainability of pest management and disease control in the agriculture and food industry.

Overall, the future work in fruit disease detection and pesticide recommendation systems is focused on improving the accuracy, efficiency, and sustainability of pest management and disease control in the agriculture and food industry.

9.3 APPLICATIONS

The fruit disease detection system has various applications in agriculture, food processing, and research. Some of the main applications of fruit disease detection systems are:

Early detection and diagnosis: Fruit disease detection systems can be used to detect and diagnose diseases in fruits at an early stage. Early detection can help farmers and food producers take timely action to prevent the spread of the disease and reduce crop losses.

Precision farming: Fruit disease detection systems can be integrated into precision farming systems to optimize the use of resources, reduce costs, and improve yields. By detecting diseases in fruits at an early stage, farmers can apply targeted treatments only where needed, reducing the use of pesticides and other inputs.

Quality control: Fruit disease detection systems can be used for quality control purposes. By identifying and removing diseased fruits from the production process, fruit disease detection systems can help in improving the overall quality of the final product.

Food safety: Fruit disease detection systems can help in ensuring the safety of fruits throughout the supply chain. By detecting and removing fruits that are contaminated with diseases or pathogens, fruit disease detection systems can help in preventing foodborne illnesses and other safety issues.

Research and development: Fruit disease detection systems can be used for research and development purposes, such as developing new treatments or management strategies for fruit diseases. By detecting and monitoring the development of diseases in fruits, researchers can better understand the biology and ecology of different pathogens and develop new tools and approaches for disease control.

Appendix A

Fruit disease detection and pesticide suggestion is an image classification task, which basically classify the fruit images based on disease. This task can be approached using various machine learning algorithms, including convolutional neural networks (CNNs) and other image classification techniques.

Fruit disease detection and pesticide suggestion is generally considered a P-type problem, as it can be solved in Polynomial time.

Convolutional Neural Network: Deep neural networks such as convolutional neural networks (CNNs) have excelled at image identification tasks. By combining input photos with various filters, CNNs are made to learn spatial hierarchies of information automatically and adaptively from the input images. Each filter takes a certain feature, like edges or corners, out of the input image. Convolutional, pooling, and fully linked layers are some of the layers that make up CNNs. Together, these layers convert the input image into a collection of feature maps, which are subsequently applied to prediction.

Image Pre-processing: Any activity involving picture recognition must begin with image pre-processing. In order to make the input photographs better suited for the model, it requires cleaning and improving them. Some typical picture pre-processing methods for handwritten digit recognition include:

- Reducing the dimensions of the photographs
- Bringing the pixel values within a predetermined range, such as 0 to 1
- Applying various filters on the image to enhance its properties, such as edge detection and gaussian blur
- Improving the image by removing noise or artefacts

Most image pre-processing operations can be performed in linear time, meaning the time required scales linearly with the number of pixels in the image. Therefore, the complexity is generally considered to be in the class P (polynomial time).

Appendix B

Paper 1:

Title: “A Review on Disease Detection and Pesticide suggestion”

Conference Name: ICCUBEA

Status: Submitted

Publisher Name: IEEE

Paper 2:

Title: “Fruit Disease Detection and Pesticide Recommendation”

Conference Name: ICCUBEA

Status: Submitted

Publisher Name: IEEE

Appendix C

by Sahil Kachole

General metrics

98,871	14,424	1098	57 min 41 sec	1 hr 50 min
characters	words	sentences	reading time	speaking time

Score

76

This text scores better than 76% of all texts checked by Grammarly

Writing Issues

908	289	619
Issues left	Critical	Advanced

Plagiarism

9 %

85 sources

9% of your text matches 85 sources on the web or in archives of academic publications

REFERENCES

[1] Asmaa Ghazi Alharbi, Muhammad Arif, “Detection and Classification of Apple Diseases using Convolutional Neural Networks”, 2020 2nd International Conference on Computer and Information Sciences (ICCIS). November 2020.

[2] Nikhitha M, Roopa Sri S, Uma Maheswari B, “Fruit Recognition and Grade of Disease Detection using Inception V3 Model”, 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA). June 2019

[3] Asad Khattak, Muhammad Usama Asghar, Ulfat Batool, Muhammad Zubair Asghar ,Hayat Ullah, Mabrook Al-Rakhami , (Member, Ieee), And Abdu Gumaei, “Automatic Detection of Citrus Fruit and leaves Diseases Using Deep Neural Network Model”, Institute of Electrical and Electronics Engineers(IEEE). July 2021

[4] Balambigai Subramanian, Jayashree.S,Kiruthika.S, Miruthula.S, “Lemon Leaf Disease Detection and Classification using SVM and CNN”, International Journal of Recent Technology and Engineering (IJRTE), Volume-8 Issue-4. November 2019

[5] Md. Ashiqul Islam, Md. Shahriar Islam, Md. Sagar Hossen, Minhaz Uddin Emon, Maria Sultana Keya, Ahsan Habib, “Machine Learning based Image Classification of Papaya Disease Recognition”, Fourth International Conference on Electronics, Communication and Aerospace Technology (ICECA-2020).

[6] Wahyuni Eka Sari, Yulia Ery Kurniawati, Paulus Insap Santosa. “Papaya Disease Detection Using Fuzzy Naïve Bayes Classifier”, 3rd International Seminar on Research of Information Technology and Intelligent Systems(ISRITI-2020).

[7] Prof. Shripad S.Veling ,Mr.Rohit ,S. Kalelkar ,Miss. Likhita V. Ajgaonkar3, Miss. Nivedita V. Mestry, Mr. Nilesh N.Gawade, “Mango Disease Detection by using Image Processing”, International Journal for Research in Applied Science & Engineering Technology (IJRASET). April 2019

[8] Vani Ashok & D. S. Vinod, “A Novel Fusion of Deep Learning and Android Application for RealTime

Mango Fruits Disease Detection”, Advances in Intelligent Systems and Computing book series (AISC,volume 1171). August 2020

[9] Avinash Kumar, Sobhangi Sarkar and Chittaranjan Pradhan “Recommendation System for Crop Identification and Pest Control Technique in Agriculture, : International Conference on Communication and Signal Processing, April 4-6, 2019, India.

[10] Zeel Doshi, Rashi Agrawal, Subhash Nadkarni, Prof. Neepa Shah “AgroConsultant: Intelligent Crop Recommendation System Using Machine Learning Algorithms”, : 2018 Fourth International Conference on Computing Communication Control and
Automation (ICCUBEA)

[11] Yousef Abbaspour-Gilandeh, Abdollah Aghabara, Mahdi Davari and Joe Mari Maja,” Feasibility of Using Computer Vision and Artificial Intelligence Techniques in Detection of Some Apple Pests and Diseases”, Multidisciplinary Digital Publishing Institute (MDPI) - Publisher of Open Access Journals. January 2022

[12] Ginne M. James and S. Sujatha,” Categorising Apple Fruit Diseases Employing Hybrid Neural Clustering Classifier”, Categorising Apple Fruit Diseases Employing Hybrid Neural Clustering Classifier. December 2020

[13] Ahmed R. Luaibi, Tariq M. Salman, Abbas Hussein Miry,” Detection of citrus leaf diseases using a deep learning technique”, International Journal of Electrical and Computer Engineering (IJECE). April 2021

[14] Ruhika Sharma and Sukhpreet Kaur,” Convolution Neural Network based Several Orange Leaves Disease Detection and Identification Methods”, International Conference on Smart Systems and Inventive Technology.2019

[15] Md. Tarek Habib, Anup Majumder, Rabindra Nath Nandi, Farruk Ahmed and Mohammad Shorif Uddin,” A Comparative Study of Classifiers in the Context of Papaya Disease Recognition”, Proceedings of International Joint Conference on Computational Intelligence.2020.

[16] K. Kurumatani, "Time Series Prediction of Agricultural Products Price Based on Time Alignment of

Recurrent Neural Networks," 2018 17th IEEE International
Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 2018, pp. 81-88, doi: 10.1109/ICMLA.2018.00020.

[17] Keke Zhang, Qiufeng Wu, Yiping Chen, Detecting soybean leaf disease from synthetic image using multi-feature fusion faster R-CNN, Computers and Electronics in Agriculture, Volume 183, 2021, 106064, ISSN 0168-1699

[18] Honghua Jiang, Chuanyin Zhang, Yongliang Qiao, Zhao Zhang, Wenjing Zhang, Changqing Song, CNN feature based graph convolutional network for pesticide and fruit recognition in smart farming,Computers and Electronics in Agriculture, Volume 174, 2020, 105450, ISSN 0168-1699

[19] Mohit Agarwal, Suneet Kr. Gupta, K.K. Biswas, Development of Efficient CNN model for Tomato fruit disease identification, Sustainable Computing: Informatics and Systems, Volume 28, 2020, 100407, ISSN 2210-5379

[20] Alessandro dos Santos Ferreira, Daniel Matte Freitas, Gercina Gonçalves da Silva, Hemerson Pistori, Marcelo Theophilo Folhes, Pesticide detection in soybean fruits using ConvNets, Computers and Electronics in Agriculture, Volume 143, 2017, Pages 314-324, ISSN 0168-1699

[21] Muhammad Hamza Asad, Abdul Bais, Pesticide detection in canola fields using maximum likelihood classification and deep convolutional neural network, Information Processing in Agriculture, Volume 7, Issue 4, 2020, Pages 535-545, ISSN 2214-3173

[22] Fenil Dankhara, Kartik Patel, Nishant Doshi, Analysis of robust pesticide detection techniques based on the Internet of Things (IoT), Procedia Computer Science, Volume 160, 2019, Pages 696-701, ISSN 1877-0509

[23] Aichen Wang, Wen Zhang, Xinhua Wei, A review on pesticide detection using ground-based machine vision and image processing techniques, Computers and Electronics in Agriculture, Volume 158, 2019, Pages 226-240, ISSN 0168-1699