A

Mini Project-II Report on

Sugarcane Disease RedRot Detection

by

GROUP NO: A2(C)

Name of Student	Roll No.
Ms. Snehal Ashok Mankar	29
Ms. Sangita Madhukar Naik	33
Ms. Madhura Sanjay Patil	38
Ms. Rutika Sachin Patil	40

Bachelor of Technology

In

Computer Science & Engineering

Under the Guidance of

Prof. B.A.Chougule

-Associate Professor

SWVSM's

Tatyasaheb Kore Institute of Engineering and Technology,Warananagar

(An Autonomous Institute)

Department of Computer Science & Engineering

Academic Year 2023-24

Tatyasaheb Kore Institute of Engineering and Technology, Warananagar

(An Autonomous Institute)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled

"Sugarcane Disease RedRot Detection"

Submitted by

Name of Student	Roll No.
Ms. Snehal Ashok Mankar	29
Ms. Sangita Madhukar Naik	33
Ms. Madhura Sanjay Patil	38
Ms. Rutika Sachin Patil	40

of Third Year B. Tech in partial fulfilment of the completion of the of B. Tech in Computer Science and Engineering. It is also certified that all corrections and suggestions indicated for internal assessment have been incorporated into the report. This Mini-Project-II work is a record of students' own work, carried out by them under our supervision and guidance during the academic year 2023-2024.

Prof. B.A.Chougule
(Project Guide)

Prof. V. C. Patil
(MP-II Co-ordinator)

Prof. R. B. Patil
(Head of Department)

Dr. S. V. Anekar
(Principal)

(External Examiner)

Place:Warananagar

Date: ____________

Tatyasaheb Kore Institute of Engineering and Technology, Warananagar.

(An Autonomous Institute)

Department of Computer science and Engineering

ACKNOWLEDGEMENT

We are greatly indebted to our guide **Prof. B. A. Chougule** **MP-**II Co-ordinator **Prof. V. C. Patil** for his unstinted support and valuable suggestions. We are grateful to them not only for the guidance, but also for their unending patience and keeping our spirits high throughout. We express our sincere thanks to our beloved Head of the Department, **Prof. R. B. Patil** and Principal, **Dr. S. V. Anekar** for being source of inspiration and providing us the opportunity to work on this project.

We extend heartfelt thanks to all the **teaching** and **non-teaching staff** of the department of computer science and system and network support of TKIET for their assistance and cooperation

Finally, we would like to thank our parents and friends for their moral support and encouragement throughout our academics.

Name of Student	Roll	Signature
Ms. Snehal Ashok Mankar	**29**	
Ms. Sangita Madhukar Naik	**33**	
Ms. Madhura Sanjay Patil	**38**	
Ms. Rutika Sachin Patil	**40**	

ABSTRACT

Sugarcane disease is a major challenge for the sugar industry in Maharastra, often leading to significant crop destruction and financial losses. Early detection and treatment of these diseases are crucial, but farmers may lack the expertise to identify them. This study explores the use of machine learning, specifically image processing and deep learning techniques (CNN), as a potential solution to this problem. By training a deep learning model on a dataset of disease-infected sugarcane images, the study successfully develops a model capable of detecting and classifying sugarcane diseases. This research offers a promising approach to assist farmers in detecting and classifying sugarcane diseases using deep learning algorithms.

INDEX

CHAPTER ONE
INTRODUCTION

1. Introduction:

Western Maharashtra, particularly regions like Kolhapur, Sangli, Satara, Pune, and Solapur, is well-known for sugar cane cultivation in India. The climate and soil conditions in this region are favorable for sugar cane cultivation, making it one of the major sugarcane-producing areas in the country. The sugar industry plays a significant role in the economy of Western Maharashtra, providing employment opportunities and contributing to the region's agricultural output. Given the importance of sugar cane cultivation in this area, the development of tools and technologies, such as the proposed web application for disease identification and management, can be particularly beneficial for farmers and stakeholders involved in sugar cane production in Western Maharashtra.

2. Motivation:

Detecting diseases like RedRot in sugarcane is crucial for farmers to prevent its spread and minimize crop losses.

1.Economic Impact: RedRot can severely reduce sugarcane yield and quality, leading to significant financial losses for farmers and the sugar industry as a whole. Early detection can help farmers take timely action to mitigate these losses.

2.Crop Health Management: Timely detection allows for targeted intervention strategies, such as applying fungicides or removing infected plants, to prevent the disease from spreading to healthy plants.

3.Resource Optimization: By identifying infected plants early, farmers can optimize the use of resources such as water, fertilizers, and pesticides, reducing waste and environmental impact.

3.Objectives:

- A user-friendly web application to identify disease in sugarcane crop
- After disease detection provide a solution.

4. Purpose:

The purpose of detecting RedRot disease in sugarcane serves multiple vital functions:

1.Early Identification: Detecting RedRot at its early stages enables farmers to take timely preventive measures. Early identification aids in controlling the spread of the disease, minimizing crop damage, and maximizing yield.

2.Economic Sustainability: RedRot detection contributes to the economic sustainability of sugarcane farming. By preventing the spread of the disease and minimizing crop losses, farmers can maintain profitability and ensure the sustainability of their agricultural operations.

CHAPTER TWO
LITERATURE SURVEY

2.1 Existing System:

1. Plantix And Plantora:

Plantix and Plantora are mobile app developed for Android and iOS platforms**.** It is designed to help farmers and gardener identify and manage crop diseases, pests, and nutrient deficiencies using image recognition technology.

Users take a picture of a plant with the app. Plantix analyzes the picture using artificial intelligence (AI) to figure out what's wrong with the plant. It gives suggestions on how to fix the problem, like which medicine or fertilizer to use.

Drawbacks in Existing System:

- **Storage/ Installation Requirement**:
  Users need to download and install mobile or iOS applications from app stores, which adds an extra step and may deter some users from accessing the application, especially if they have limited storage space on their devices.
- **Update Android App**:
  Updates to mobile or iOS applications need to be distributed through app stores, which may involve additional time and effort for developers. Users also need to manually download and install updates, which can lead to fragmentation if not all users update to the latest version promptly.
- **Platform Dependency**:
  Mobile or iOS applications are typically developed for specific platforms, such as iOS for Apple devices and Android for other smartphones. This requires separate development efforts for each platform, increasing development time and costs.

2.1.1 Referred Journal / Conference Papers:

1. Project Title: Sugarcane Disease RedRot Detection

- Objective: To detect RedRot disease in sugarcane using advanced technology.

- Methodology: Utilizing image processing and machine learning algorithms for early disease detection.
- Outcome: Enhancing crop yield by timely identification and treatment of RedRot, contributing to agricultural sustainability.
- Target Audience: Science students interested in agricultural science and technology.
- Significance: Demonstrates the application of technology in agriculture, fostering interest in research and innovation among students.

2.1.2 Elaborate on Existing System Applications/Examples:

1.Plantix:

Plantix is a mobile application designed to assist farmers and gardeners in diagnosing plant diseases and nutrient deficiencies.

Using image recognition technology, Plantix identifies plant health issues based on uploaded photos of affected plants.

The app provides detailed information about the identified problem, including potential causes and recommended solutions.

Plantix also offers a community platform where users can share their experiences, ask questions, and learn from each other.

Examples of Plantix's functionality include diagnosing diseases like powdery mildew in grapes, leaf spot in tomatoes, and nutrient deficiencies in various crops.

2.Plantora:

Plantora is an application that serves as a comprehensive plant care assistant, offering guidance on plant care, maintenance, and identification.

Users can access a vast database of plants and receive personalized care tips based on factors like location, climate, and plant type.

Plantora provides information on watering schedules, sunlight requirements, pruning techniques, and pest management strategies for different plant species.

The app also offers a plant identification feature, allowing users to upload photos of plants for accurate species identification and care advice.

Examples of Plantora's functionality include helping users care for indoor plants like succulents, outdoor plants like roses, and even providing guidance on growing vegetables in home gardens.

2.1.3 Limitations or challenges in Existing System:

1. **Limited Plant Database**:

   Both Plantix and Plantora rely on databases of known plant diseases, pests, and care information. However, these databases may not cover every plant species or every possible issue that users encounter. This limitation can affect the accuracy of diagnoses and care recommendations, especially for less common plants or emerging diseases.

2. **Dependency on Image Quality**:

   Plantix primarily relies on image recognition technology for diagnosing plant diseases. The accuracy of diagnoses is highly dependent on the quality of the images uploaded by users. Poor lighting conditions, image blur, or obscured plant parts can lead to misdiagnoses or inaccurate recommendations.

3. **Geographic and Environmental Variability**:

   Plant health issues and optimal care practices can vary significantly based on geographic location, climate, soil type, and other environmental factors. Existing systems like Plantix and Plantora may not always account for these variations, leading to generic recommendations that may not be suitable for all users.

4. **Limited User Feedback Incorporation**:

   While Plantix and Plantora may offer community platforms for users to shareexperiences and ask questions, there may be limitations in how effectively user feedback is incorporated into the systems. Continuous improvement based on user feedback is crucial for maintaining accuracy and relevance, but challenges may arise in managing and analyzing large volumes of user-generated content.

5. **Internet Connectivity Requirement**:

   Both Plantix and Plantora typically require internet connectivity to access their databases, upload images, and receive updates. This dependency on internet access may pose challenges for users in remote or rural areas with limited connectivity, hindering their ability to utilize these tools effectively.

2.2 Problem Statement:

Sugarcane diseases are hard to detect visually, farmers are unsure about the specific disease affecting their crops. This uncertainty makes it challenging for them to effectively treat the disease, potentially resulting in lower sugarcane yields and reduced income for the farmers.

2.3 Proposed System:

Our system helps sugarcane farmers to find and deal with diseases better. By taking pictures of their fields, farmers can quickly find out if their crops are sick. We give them advice on what to do next, like using the right treatments or preventing the disease from spreading. It's all about helping sugarcane farming stay healthy and successful.

2.4 Feasibility study:

1. Market Analysis:

Understanding the market demand for a RedRot disease detection system in sugarcane cultivation regions is essential. This involves identifying potential users, their needs, and preferences, as well as assessing competition from existing solutions or traditional methods of disease detection.

2. Risk Assessment:

Identifying potential risks and challenges associated with implementing the disease detection system, such as technical barriers, regulatory issues, data privacy concerns, and environmental factors, is crucial for informed decision-making.

3. Technological Feasibility:

This involves assessing the availability of suitable technology for image processing and machine learning algorithms to accurately detect RedRot disease in sugarcane plants. It includes evaluating the feasibility of developing or leveraging existing software solutions for disease detection.

CHAPTER 3 PROJECT SCOPE AND REQUIREMENT ANALYSIS

3.1 Project Scope:

1. Objective: The primary goal of the project is to develop a system for early detection of Red Rot disease in sugarcane crops.
2. Technology: The project will utilize image processing and machine learning algorithms for disease detection.
3. Platform: The system will be designed as a mobile application for ease of use by farmers and agricultural workers.
4. Geographic Scope: Initially, the focus will be on regions where sugarcane cultivation is prevalent and Red Rot disease poses a significant threat.
5. Collaboration: Collaboration with agricultural experts, researchers, and sugarcane farmers will be essential for data collection, validation, and real-world testing.

3.2 Requirement gathering and analysis:

3.2.1 Functional Requirements:

1.User Interface for Data input:

Develop a user-friendly interface accessible computers, allowing farmers to easily upload images of their sugarcane fields. The interface should guide farmers through the process of capturing and submitting images, ensuring data quality and consistency.

2.Image Recognition and Analysis:

The system should be able to analyze images of sugarcane plants captured by farmers to identify symptoms of diseases.It should employ image recognition algorithms to detect patterns, discoloration, lesions, or other visual indicators associated with sugarcane diseases.

3.Disease Detection:

a system capable of accurately detecting sugarcane diseases using advanced imaging techniques, such as multispectral or hyperspectral imaging, to capture subtle changes in plant health. This system should be able to analyze images of sugarcane fields and identify potential disease hotspots.

4.Treatment Recommendations:

Provide farmers with detailed treatment recommendations based on the diagnosed disease, including information on recommended pesticides, fungicides, or cultural practices to mitigate disease spread and minimize yield losses.

3.2.2 Non-Functional Requirements:

1.Performance:

The system should be capable of processing and analyzing sugarcane images rapidly, providing real-time or near-real-time feedback to farmers.Response time for disease detection and diagnosis should be minimized to ensure timely intervention.

2.Scalability:

The solution should be scalable to accommodate a large number of users and handle varying levels of image data.It should be able to expand seamlessly to support additional sugarcane farms and geographic regions as needed.

3. Reliability:

The system should be highly reliable, with minimal downtime or service interruptions. It should be resilient to failures and capable of recovering gracefully from errors to ensure uninterrupted operation.

4.Accuracy:

Disease detection algorithms should achieve high levels of accuracy and reliability in identifying sugarcane diseases.The system should minimize false positives and false negatives to prevent misdiagnosis and ineffective treatment.

5.Usability:

The user interface should be intuitive and easy to navigate, catering to users with varying levels of technological proficiency.The system should provide clear instructions and guidance to farmers on how to use the platform effectively.

6.Data Privacy:

- User data should be kept confidential and only accessible to authorized personnel.
- Compliance with data protection regulations such as GDPR should be ensured.
- Users should have the option to provide feedback on the e-waste facility centers they visit.

3.2.3 Data Requirements:

1. Input Data:
    - Image.

2. Output Data:
    - RedRot Disease.

3.2.4 Software and Hardware requirements:

- **Software requirements:**

    Windows 10 and Above

    Frontend: HTML, CSS, JS

    Backend: Python and Frameworks

    Database: MySQL

    Image Recognition: Python Libraries

    Web Server: Google Chrome

- **Hardware requirements**

    2 GB required 4 GB recommended.

    Processor: Intel or AMD

CHAPTER 4
PROJECT DESIGN AND MODELLING DETAILS

4.1 Software requirement specification:

1. Windows 10 and Above:

   Windows 10 is a widely used and supported operating system, ensuring compatibility with a broad range of hardware and software. It provides a stable and secure environment for running the proposed system.
2. Frontend: HTML, CSS, JS:

   HTML, CSS, and JavaScript are fundamental technologies for building web-based user interfaces. They are well-supported across browsers and platforms, ensuring consistent and accessible user experiences for farmers accessing the system.
3. Backend: Python and Frameworks:

   Python is a versatile and widely adopted programming language known for its simplicity and readability. It offers numerous frameworks such as Django and Flask, which provide robust tools for building scalable and maintainable web applications. Python's ease of use and extensive libraries make it suitable for rapid development and integration with other components.
4. Database: MySQL:

   MySQL is a popular open-source relational database management system known for its reliability, performance, and ease of use. It provides features such as ACID compliance, data integrity, and scalability, making it well-suited for storing and managing the data required for the proposed system.
5. Image Recognition:

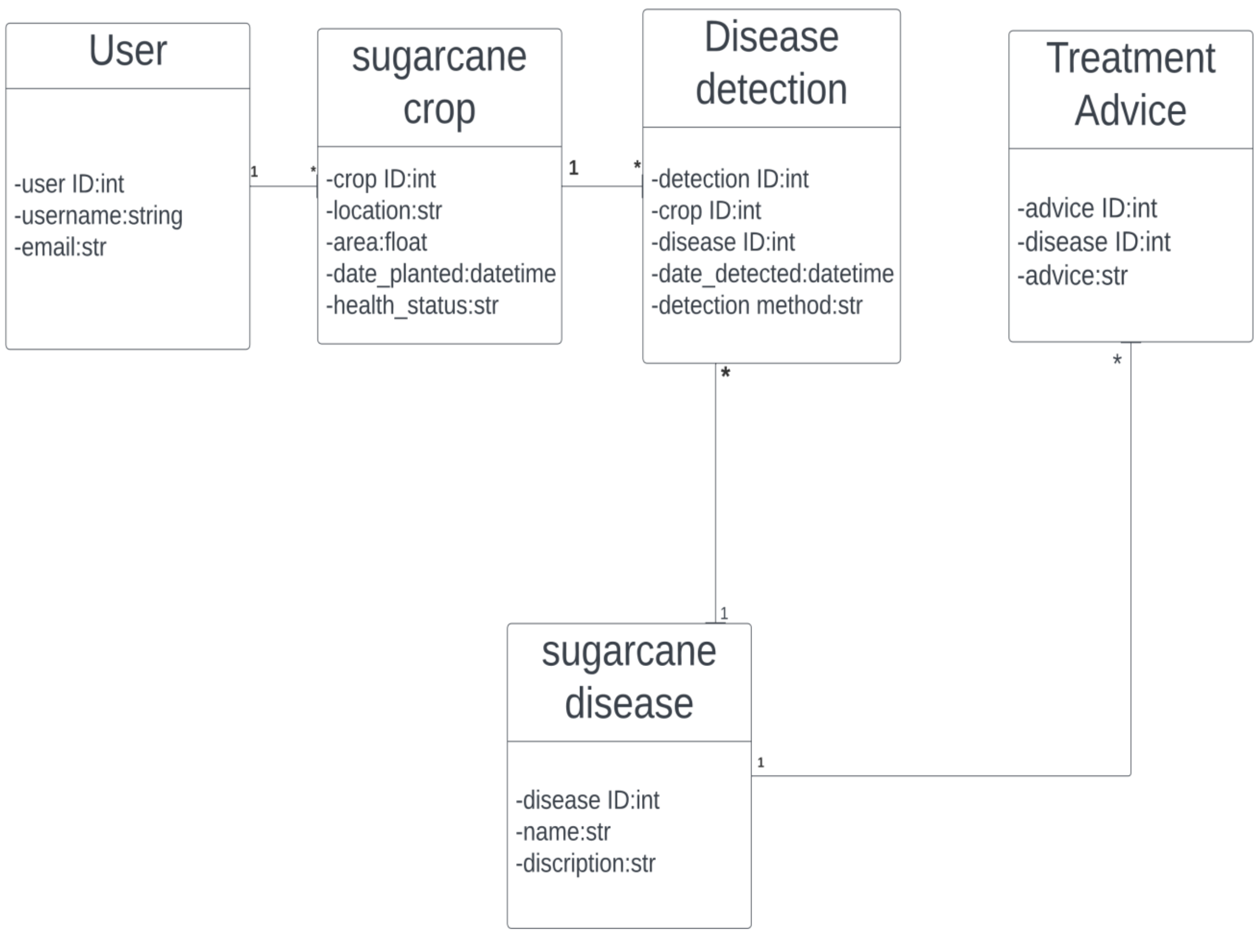
   Python Libraries: Python libraries such as OpenCV, TensorFlow, and Keras offer powerful tools and algorithms for image recognition and machine learning tasks. These libraries enable the system to analyze images captured by farmers and identify signs of sugarcane diseases accurately.

6. Web Server: Google Chrome:

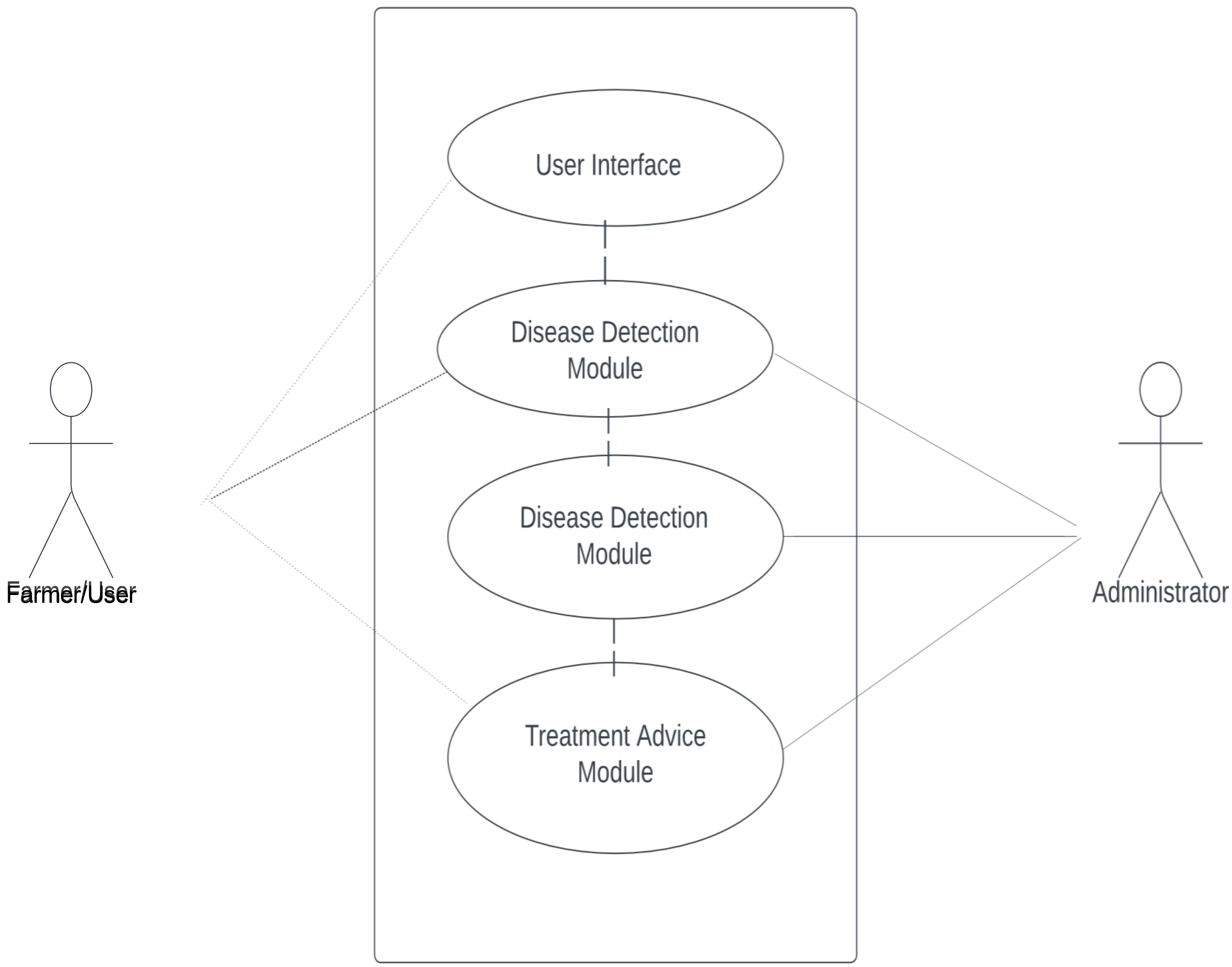
   Google Chrome is a widely used web browser known for its speed, security, and compatibility with modern web standards. While it is not typically used as a web server, it can be used for local development and testing purposes, providing a familiar environment for frontend development and debugging.

4.2 System Modeling and Design:

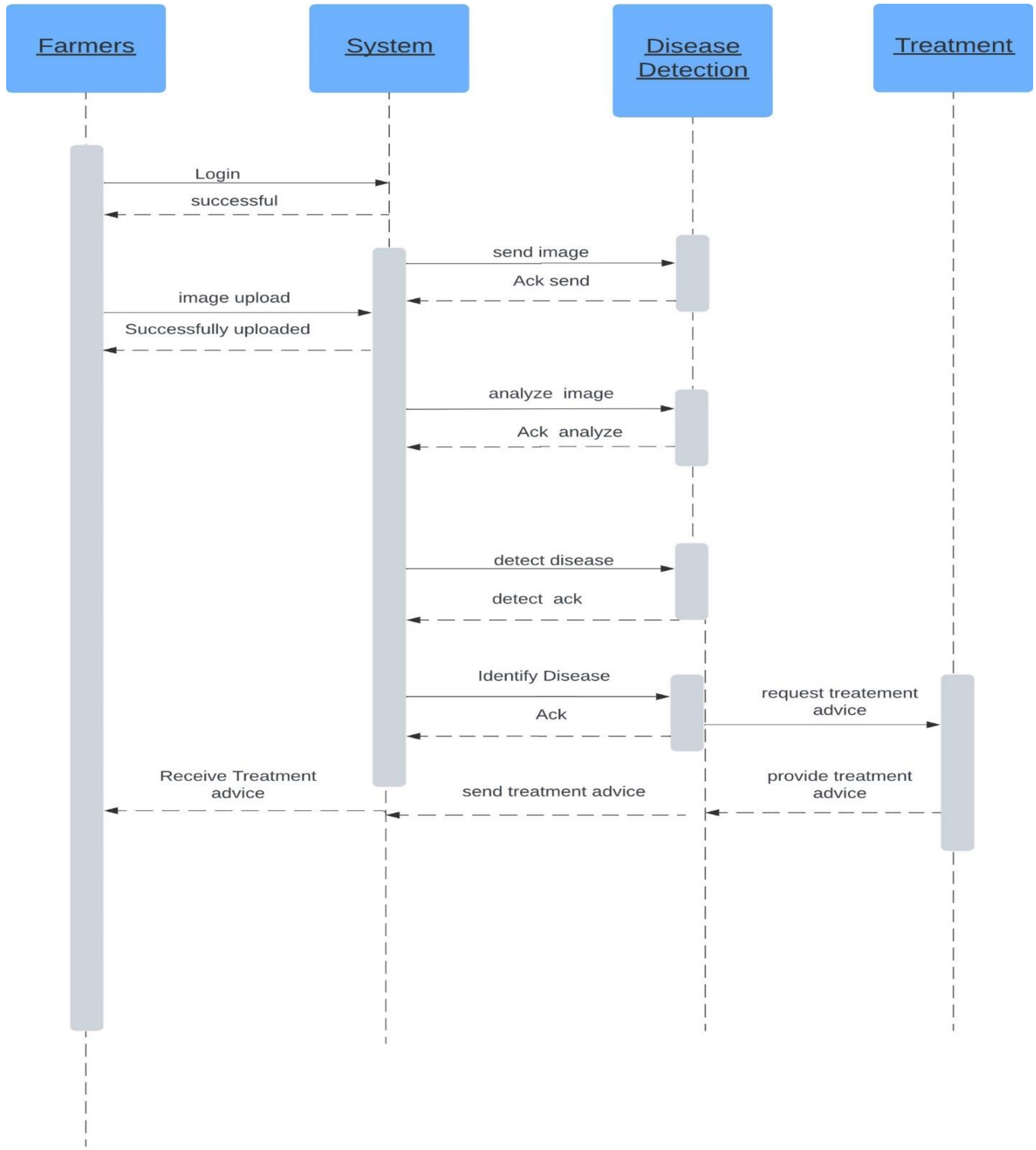
Class diagram:



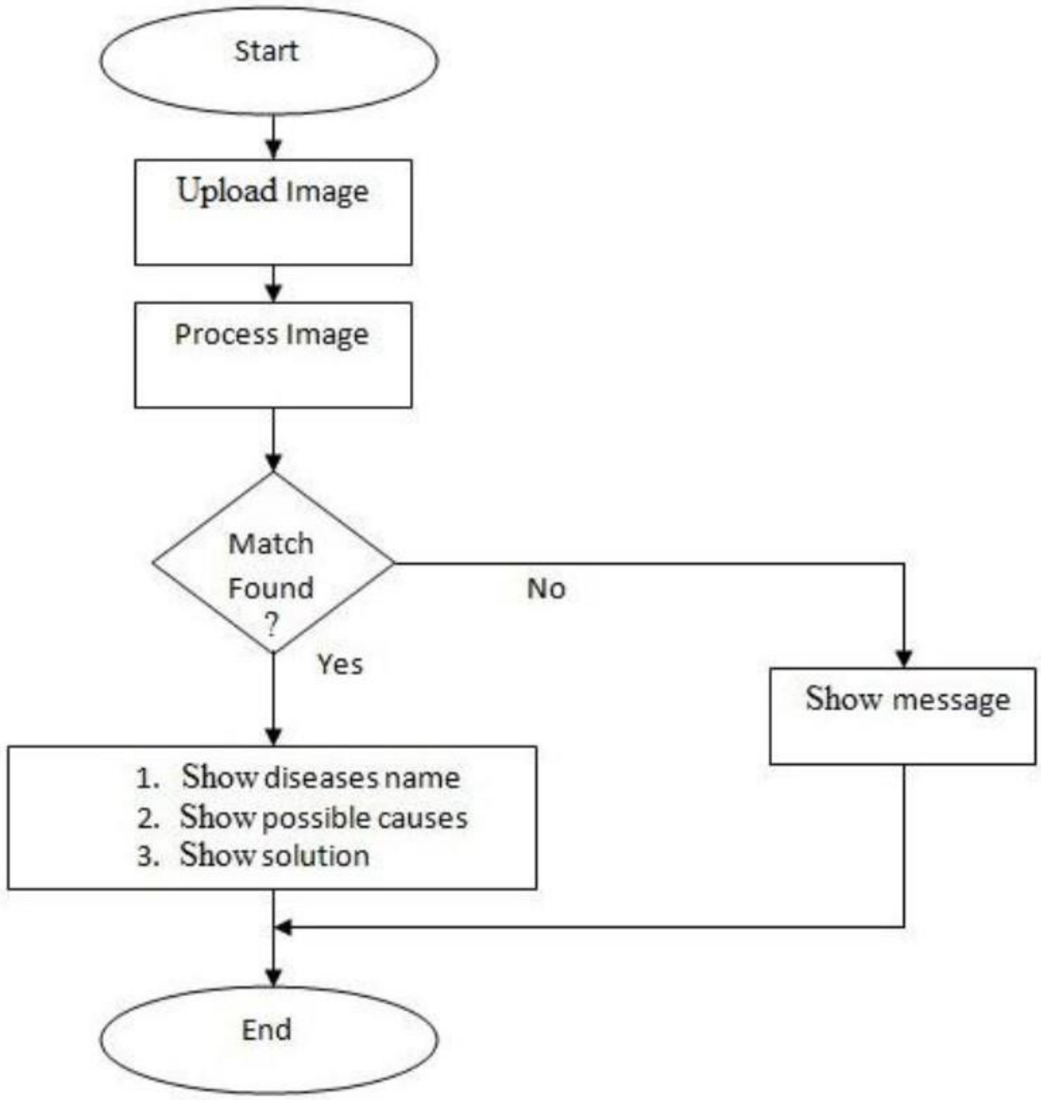
Use Case Diagram:



Sequence Diagram:



Data Flow Diagram :



CHAPTER 5 IMPLEMENTATION AND CODING

5.1 Algorithms:

1. **Data Collection and Preprocessing:**

Gather a diverse dataset of images containing healthy sugarcane leaves as well as leaves affected by various diseases.

Preprocess the images to ensure consistency in size, color, and orientation. Techniques such as resizing, normalization, and data augmentation can be applied to enhance the dataset's quality and diversity.

2. **Model Development:**

Design a deep learning model for image classification, particularly suited for detecting sugarcane diseases.

Choose an appropriate architecture such as Convolutional Neural Networks (CNNs) due to their effectiveness in image-related tasks.

Experiment with different CNN architectures, varying the number of layers, kernel sizes, and activation functions to optimize performance.

Incorporate techniques like transfer learning if available pre-trained models exist,which can leverage knowledge learned from other similar tasks or datasets.

3. **Training and Validation:**

Split the preprocessed dataset into training, validation, and testing sets to train and evaluate the model's performance.

Train the model on the training set while monitoring its performance on the validation set to prevent overfitting and fine-tune hyperparameters.

Utilize appropriate metrics such as accuracy, precision, recall, and F1-score to evaluate the model's performance on both the validation and testing sets.

4. **Model Evaluation and Interpretation:**

Evaluate the trained model on the testing set to assess its real-world performance in detecting sugarcane diseases.

Analyze the model's predictions and misclassifications to understand its strengths and limitations.

Interpret the model's decisions using techniques like class activation maps or attention mechanisms to provide insights into which regions of the image contributed most to the predictions.

5. **Deployment and Integration:**

Deploy the trained model as a user-friendly tool accessible to sugarcane farmers, possibly as a mobile application or web-based platform.

Integrate the model with an intuitive interface that allows farmers to upload images of sugarcane leaves for disease diagnosis.

Provide actionable recommendations based on the model's predictions, including suggested treatments or interventions for identified diseases.

6. **Feedback and Iteration**:

Gather feedback from users, particularly farmers and agricultural experts, to continually improve the model's accuracy and usability.

Incorporate new data and insights into the model's training process to adapt to evolving patterns of sugarcane diseases and treatment effectiveness.

Iterate on the model and deployment platform based on user feedback and emerging technologies to ensure ongoing relevance and effectiveness.

5.2 Software Requirements with relevant justifications:

Windows 10 and Above: This refers to the operating system needed to run the software. Windows 10 and later versions provide the necessary support for modern software applications, security updates, and advanced features needed to support contemporary software developments.

Frontend: HTML, CSS, JS:

HTML (HyperText Markup Language): The standard markup language used to create and design the structure of web pages.

CSS (Cascading Style Sheets): Used for styling HTML elements on web pages. It allows for the separation of presentation and content, including layout, colors, and fonts.

JavaScript (JS): A scripting language used to create dynamic content on websites, handling interactions, animations, and complex features that improve user experience.

Backend: Python and Frameworks:

Python: A high-level, interpreted programming language known for its readability, succinctness, and object-oriented features. It’s widely used for web development, data analysis, artificial intelligence, and more.

Frameworks (e.g., Django, Flask): These are collections of modules and libraries that help in developing robust web applications. Django provides a high-level framework that encourages rapid development and clean, pragmatic design. Flask is a micro-framework that provides the essentials to build a web app but remains lightweight and modular.

Database: MySQL:

MySQL: A popular relational database management system known for its reliability and ease of use. It uses structured query language (SQL) for accessing and managing the data stored in relational databases.

Image Recognition: Python Libraries:

Libraries such as OpenCV, TensorFlow, and PyTorch are commonly used for image recognition tasks. These libraries provide tools to process images and implement machine learning algorithms for tasks such as image classification, object detection, and more.

Web Server: Google Chrome:

Although not a web server, Google Chrome is a web browser. A web server software requirement might instead be something like Apache or Nginx. Google Chrome is used to access and interact with web applications developed using the aforementioned technologies.

5.3 Hardware Requirements with relevant justifications:

RAM: 2 GB required, 4 GB recommended.

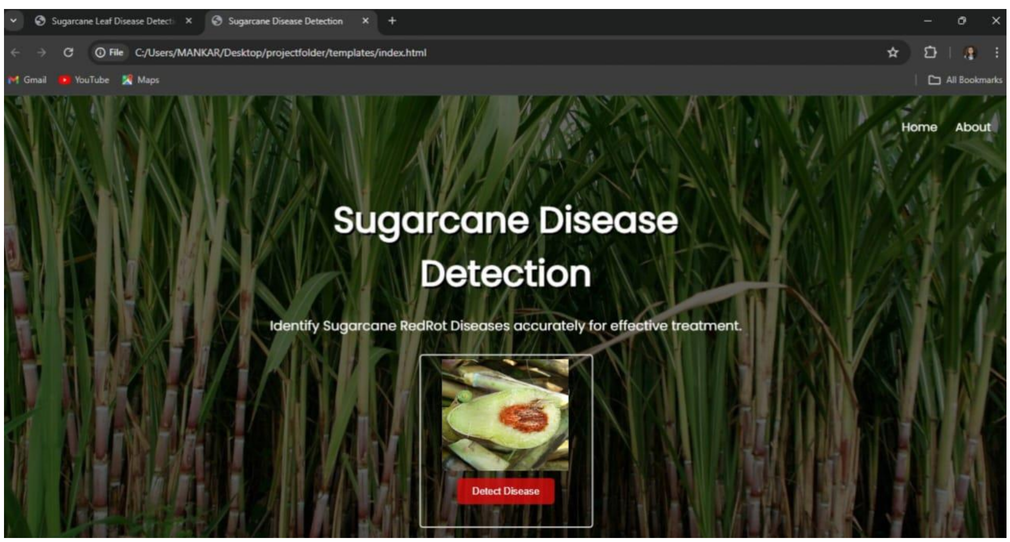
RAM (Random Access Memory): This is the hardware within a computer that temporarily stores data, serving as the computer's "working" memory. More RAM allows for more processes to be executed simultaneously, crucial for running complex applications smoothly.

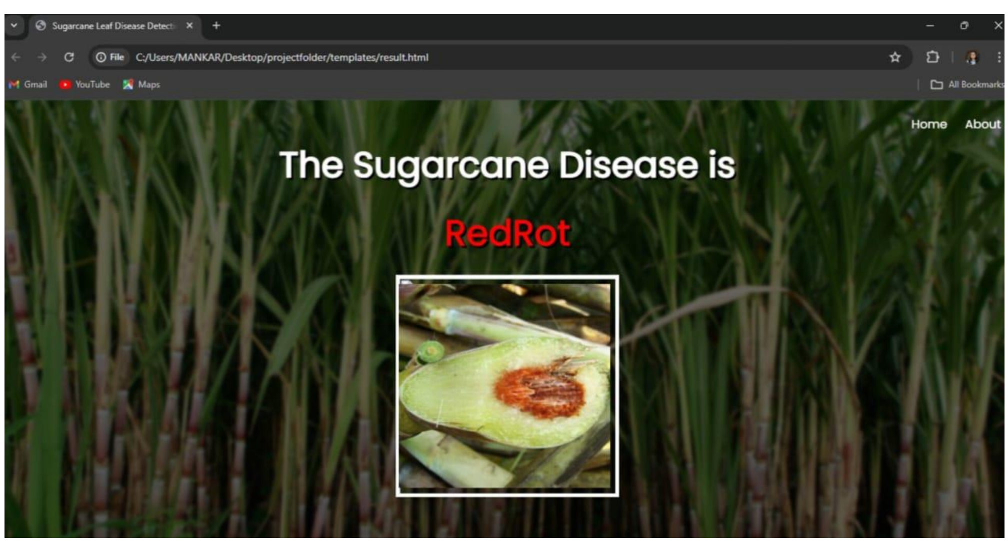
Processor: Intel or AMD:

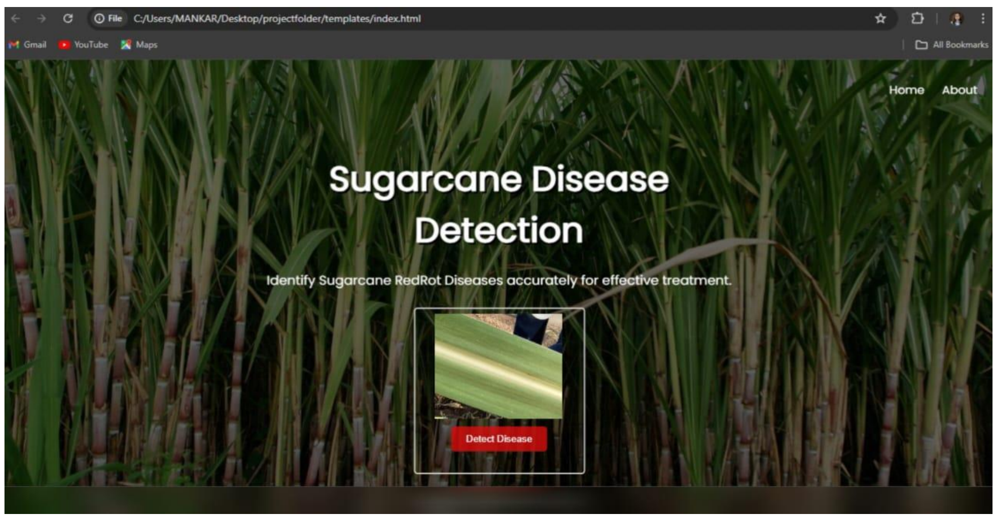
Intel and AMD: These are the two leading manufacturers of processors. Both offer a range of processors that support general computing needs to high-end processing requirements for gaming, video editing, and more. The choice between Intel and AMD depends on specific performance needs and budget considerations.

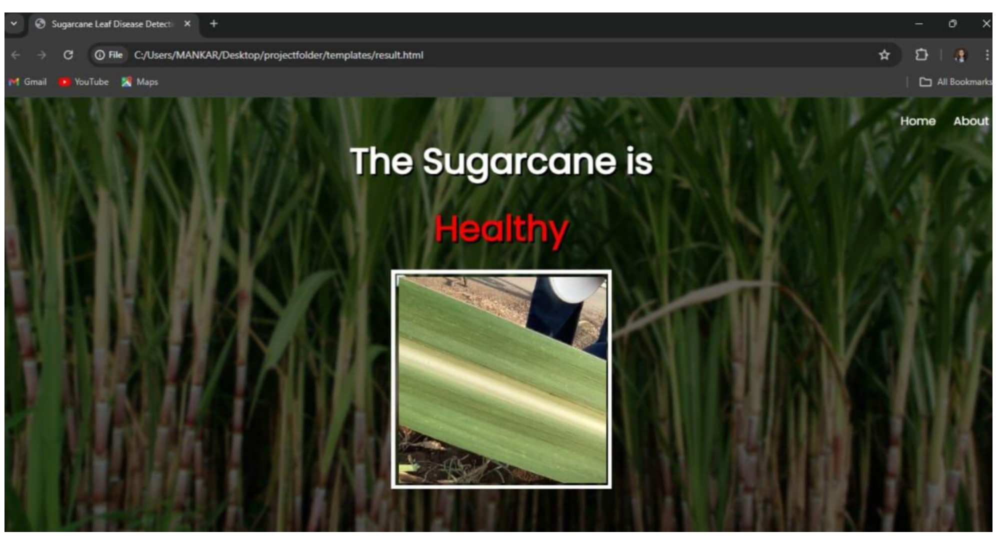
5.4 Implementation:

Screenshots






C:/Users/MANKAR/Desktop/projectfolder/templates/index.html
Gmail
YouTube
Maps
All Bookmarks
Home
About
Sugarcane Disease Detection
Identify Sugarcane RedRot Diseases accurately for effective treatment.
Detect Disease


Sugarcane Leaf Disease Detecti
C:/Users/MANKAR/Desktop/projectfolder/templates/result.html
Gmail
YouTube
Maps
Home
About
The Sugarcane is
Healthy

CHAPTER 6 TEST SPECIFICATION

6.1 Fundamental of testing:

i. **Purpose:** The primary purpose of testing is to identify defects or errors in software systems and ensure that they meet the specified requirements. Testing helps to improve the quality, reliability, and performance of the software.

ii. **Testing Process:** Testing follows a systematic process that includes test planning, test design, test execution, and test evaluation. This process ensures that all aspects of the software are thoroughly tested and verified.

iii. **Test Objectives:** Testing aims to achieve various objectives, including finding defects, validating the functionality of the software, verifying if it meets the requirements, ensuring usability, reliability, and performance, and assessing the overall quality of the software.

iv. **Test Levels:** Testing is carried out at multiple levels to cover different aspects of the software. Common test levels include unit testing (testing individual components), integration testing (testing the interaction between components), system testing (testing the entire system), and acceptance testing (testing against user requirements).

v. **Test Techniques:** Various test techniques are employed to design effective test cases and scenarios. These include black-box testing (testing without knowledge of internal code structure), white-box testing (testing with knowledge of the internal code structure), and gray-box testing (a combination of black-box and white-box testing).

vi. **Test Types:** Different types of tests are performed to address specific aspects of software quality. These include functional testing (testing the functionality of the software), performance testing (testing the system under specific workloads), security testing (testing for vulnerabilities and threats), usability testing (testing the user-friendliness of the software), and regression testing (retesting after modifications to ensure existing functionality is not affected).

vii. **Test Documentation:** Documentation is an essential aspect of testing. It includes test plans, test cases, test scripts, test data, and test reports. This documentation helps in tracking the progress of testing, reproducing issues, and providing evidence of testing performed.

viii. **Test Automation:** Test automation involves using specialized tools and scripts to automate repetitive and time-consuming testing tasks. Automation can improve efficiency, reduce human error, and enable frequent testing in agile development environments.

ix. **Test Environment:** A suitable test environment, including hardware, software, and network configurations, is essential for accurate and reliable testing. The test environment should mimic the production environment as closely as possible. Effective testing plays a vital role in ensuring the reliability and quality of software systems. By following the fundamentals of testing, organizations can mitigate risks, improve customer satisfaction, and deliver robust and reliable software solutions.

6.2 Test Cases:

1.Test Case: Disease Detection Accuracy

Description: Validate the accuracy of the model in detecting sugarcane diseases.

Steps:

Provide the model with images of sugarcane leaves affected by different diseases.

Verify that the model correctly identifies the type of disease present in each image.

Compare the model's predictions with expert diagnoses or ground truth data.

Calculate metrics such as precision, recall, and accuracy to assess the performance of disease detection.

2.Test Case: Disease Misclassification Rate

Description: Determine the rate of misclassification or false positive/negative detections.

Steps:

Evaluate the model's performance on a set of test images containing both diseased and healthy sugarcane leaves.

Identify instances where the model incorrectly classifies a healthy leaf as diseased (false positive) or a diseased leaf as healthy (false negative).

Calculate the misclassification rate to understand the frequency of incorrect diagnoses.

3. **Test Case: Treatment Recommendation**

Description: Assess the effectiveness of treatment recommendations provided based on disease detection.

Steps:

Use the model to identify the type of disease affecting a sample of sugarcane crops.

Implement treatment strategies recommended by the model for each detected disease.

Monitor the progression of the treated crops over time.

Evaluate the success rate of treatments in mitigating the impact of the detected diseases on sugarcane yields.

3.Test Case: Farmer Training and Support

Description: Measure the impact of providing training and support to farmers on disease detection and management.

Steps:

Conduct training sessions to educate farmers about common sugarcane diseases and their visual symptoms.

Provide farmers with access to the disease detection model and guidance on interpreting its output.

Monitor farmers' ability to use the model effectively for identifying and managing sugarcane diseases.

Collect feedback from farmers regarding the usefulness and usability of the model and associated training materials.

4.Test Case: Yield Improvement

Description: Evaluate the impact of disease detection and management on sugarcane yields and farmers' income.

Steps:

Implement disease detection and management interventions based on the model's recommendations in a real-world farming context.

Track the yield of sugarcane crops over multiple growing seasons.

Compare the yield and income generated from treated crops with historical data or untreated crops.

Analyze the financial benefits and return on investment (ROI) of using the disease detection model and associated management practices.

These test cases aim to assess the effectiveness, accuracy, and real-world impact of using technology-based solutions for detecting and managing sugarcane diseases, ultimately improving agricultural outcomes for farmers.

CHAPTER 7 CONCLUSION

7.1 Conclusion:

Sugarcane farmers have a tough time figuring out if their crops are sick because it's hard to spot diseases just by looking at them. This uncertainty makes it tricky for farmers to treat the problems effectively. As a result, they might end up with fewer sugarcane plants and less money from their harvests. To help them out, we need easy-to-use tools that use smart technology to spot and diagnose sugarcane diseases. These tools can give farmers the right advice on how to treat the sick plants and prevent the diseases from spreading. By working together and using these tools, farmers can protect their crops better and make more money in the long run.

7.2 References:

7.2.1 Journal/Conference Papers:

1. Web Based Sugarcane RedRot Disease Detection.
2. Nearby Farmers Collection Information About Sugercane
3. Web Based Sugarcane Disease Detection Web Application.

7.2.2 Books References:

- IEEE Paper

7.2.3 Web References:

- https://www.researchgate.net/profile/Arpan-Kumar-3/publication/336037806_Detection_of_Sugarcane_Disease_and_Classification_using_Image_Processing/links/5ee64984a6fdcc73be7b9797/Detection-of-Sugarcane-Disease-and-Classification-using-Image-Processing.pdf
- https://ieeexplore.ieee.org/abstract/document/8942690
- https://www.turcomat.org/index.php/turkbilmat/article/view/8769

MP-2 GUIDE	**MP-2 Co-ordinator**	**HOD[CSE]**
Prof.A.S.Phalle	**Prof.V.C.Patil**	**Prof.R.B.Patil**