KIET GROUP OF INSTITUTIONS

Connecting Life with Learning

www.kiet.edu

Delhi-NCR, Ghaziabad

Synopsis

on

Tomato Leaf Disease Classification Using Convolutional Neural Networks

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Harsh Kumar Singh (2100290100069)

Prateek Kumar (2100290100118)

Prince Rauniyar (2100290100119)

Under the supervision of

Prof. Pushpendra Kumar

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2025

1. INTRODUCTION

1. PURPOSE

Identification of the plant diseases is the key to preventing the losses in the yield and quantity of the agricultural product. The studies of the plant diseases mean the studies of visually observable patterns seen on the plant. Health monitoring and disease detection on plant is very critical for sustainable agriculture. It is very difficult to monitor the plant diseases manually. It requires tremendous amount of work, expertize in the plant diseases, and also require the excessive processing time. Hence, image processing and Machine learning techniques are used for the detection of plant diseases. Disease detection involves the steps like image acquisition, image pre-processing, image segmentation, feature extraction and classification

2. INTENDED AUDIENCE AND READING SUGGESTIONS

1.Farmers and Agricultural Workers:

Audience Description: Farmers, farm managers, agricultural workers, and individuals involved in crop cultivation who seek practical solutions for identifying and managing plant diseases to optimize crop health and productivity.

2.Reading Suggestions:

"Introduction to Plant Pathology" by George N. Agrios: Provides an overview of plant diseases, symptoms, and management strategies.

"Integrated Pest Management for Sustainable Intensive Agriculture" by Geoff M. Gurr et al.: Offers insights into sustainable pest and disease management practices for agricultural systems.

3.Agronomists and Crop Consultants:

Audience Description: Agronomists, crop consultants, and agricultural advisors who require in-depth knowledge and expertise in plant pathology, disease diagnosis, and crop management.

3. PROJECT SCOPE

The scope of the project encompasses the development of a software system capable of accurately identifying plant diseases through image classification techniques. The system will enable users, such as farmers and agronomists, to upload images of plant leaves and receive classification results indicating the presence of diseases. Additionally, the system will provide recommendations for disease management and treatment based on the identified diseases.

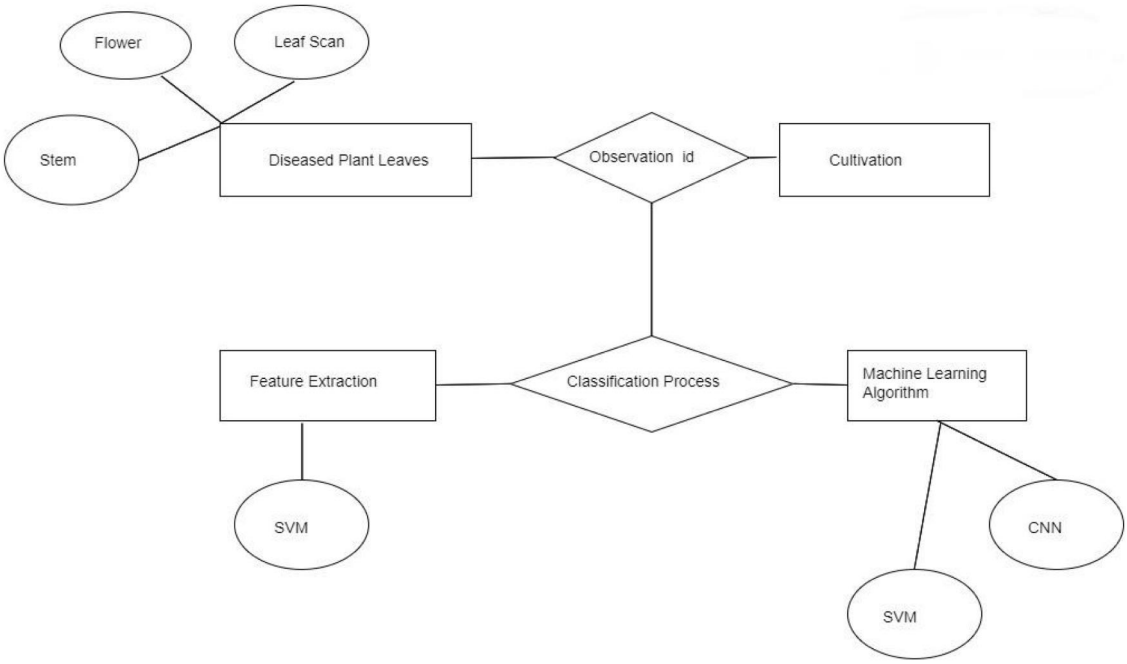
4. REFERENCES

- https://www.scribd.com/Plant-Leaf-Disease-Detection
- Fundamentals of dataset by ramez elmarsi and shamkant b.navathe

2. OVERALL DESCRIPTION

1. Product Perspective

This section describes the context in which the software system will operate, including its relationship to other systems or components. In the case of the "Plant Disease Classification" system, it is depicted as a standalone application with interfaces for user interaction, emphasizing its independent functionality.



ER Diagram of tomato disese detection

2. Product Functions

The "Plant Disease Classification" project incorporates several key features designed to provide users with a comprehensive tool for accurately identifying plant diseases and managing agricultural health. Here's a detailed description of the product features:

1. **Image Upload**: Users can easily upload images of plant leaves exhibiting symptoms of disease through the system's intuitive user interface. This feature allows farmers and agronomists to quickly capture and submit images using cameras or mobile devices, facilitating seamless interaction with the system.

2. **Disease Classification**: Upon image upload, the system employs advanced machine learning algorithms, such as convolutional neural networks (CNNs), to analyze the images and classify them into predefined disease categories. By leveraging image classification techniques, the system accurately identifies the presence of diseases based on the observed symptoms, enabling users to diagnose issues promptly and take appropriate action.

3. **Classification Result Display**: The system presents the classification results to users in a clear and accessible format, indicating the detected plant disease(s) along with probability scores for each disease class. This feature allows users to understand the severity and likelihood of different diseases affecting their crops, facilitating informed decision-making and prioritization of disease management efforts.

4. **Actionable Recommendations**: In addition to classification results, the system provides users with actionable recommendations for disease management and treatment based on the identified diseases. These recommendations may include suggested preventive measures, such as crop rotation or pest control strategies, as well as potential treatments, such as application of fungicides or pruning techniques. By offering practical guidance, the system empowers users to effectively address plant diseases and minimize crop damage.

5. **User Authentication and Management**: The system incorporates user authentication mechanisms to ensure security and accountability. Users are required to register for an account or log in using existing credentials before accessing the system's features. User management functionalities allow administrators to manage user accounts, permissions, and access levels, ensuring appropriate control and governance over system usage.

6. **User-Friendly Interface**: The system features a user-friendly graphical user interface (GUI) designed to streamline the image upload and classification process. Intuitive navigation and interactive elements guide users through the workflow, making it easy for individuals with varying levels of technical expertise to interact with the system effectively. Clear and informative feedback messages provide users with guidance and support throughout their interaction with the system.

3. USER CLASS and CHARACTERISTICS

In the "Plant Disease Classification" project, the user class encompasses individuals involved in plant cultivation, including farmers, agronomists, and researchers. Each user class possesses distinct characteristics and requirements tailored to their roles

and expertise levels. Here's a breakdown of the user classes and their characteristics:

Farmers:

Characteristics:

Typically have practical knowledge of crop cultivation and management. Often work in field settings and may have limited access to technical resources.

Require practical and actionable insights to address plant diseases and optimize crop yields.

Requirements:

User-friendly interface that simplifies the process of uploading images and interpreting classification results.

Actionable recommendations for disease management and treatment that are easy to understand and implement in field conditions.

Access to the system from various devices, including mobile phones and tablets, for on-the-go usage in agricultural settings.

Agronomists:

Characteristics:

Possess specialized knowledge of agronomy and plant pathology.Often work in advisory or consulting roles, providing expertise and guidance to farmers.

Require detailed insights and analysis to support decision-making and recommendations for disease management.

Requirements:

Access to advanced analysis tools and features for in-depth examination of classification results and disease trends.

Customizable settings for adjusting classification parameters and fine-tuning model performance based on specific crop varieties and environmental conditions.

Integration capabilities with other agricultural technologies and management systems to streamline workflow and data sharing.

1.Researchers:

Characteristics:

Engage in scientific research and experimentation to advance understanding of plant diseases and develop innovative solutions.

Require access to large datasets and analytical tools for conducting experiments and validating research findings.

2.Requirements:

Access to comprehensive data analytics capabilities for exploring trends, correlations, and patterns in disease data.

Collaboration features for sharing research findings, datasets, and methodologies with peers and colleagues.Integration with research databases and repositories for accessing relevant literature and resources to support research efforts.

Common Requirements Across User Classes:

- **Security:** All user classes require secure authentication mechanisms to protect user data and ensure privacy.

- **Reliability:** Users expect the system to be reliable and available whenever needed, especially during critical times such as disease outbreaks or crop monitoring periods.

- **Training and Support:** Users may require training and support resources to familiarize themselves with the system's features and capabilities, as well as ongoing assistance to address technical issues or questions that may arise during usage.

- **Accessibility:** The system should be accessible to users with varying levels of technical proficiency and accommodate different devices and network environments to ensure widespread adoption and usability.

4. OPERATING ENVIRONMENT

The operating environment of the "Plant Disease Classification" project refers to the system's technical infrastructure, including hardware, software, and network components, as well as the conditions under which the system operates. Here's an overview of the operating environment for the project:

1.Hardware Requirements:

Server Infrastructure: The project requires server infrastructure to host the web-based application and handle image processing tasks. This infrastructure may include physical servers or cloud-based services.

Client Devices: Users access the system through various client devices, including desktop computers, laptops, tablets, and smartphones. These devices should have internet connectivity and web browser support to interact with the system.

2.Software Requirements:

Web Server: The system operates on a web server that hosts the application and serves web pages to users. Common web servers include Apache, Nginx, or Microsoft IIS.

Database Management System (DBMS): The project utilizes a DBMS to store and manage data related to user accounts, uploaded images, classification results, and disease information. Popular options include MySQL, PostgreSQL, or MongoDB.

Programming Languages and Frameworks: The system is developed using programming languages such as Python, JavaScript, or Java, along with frameworks and libraries for web development (e.g., Flask, Django, Node.js, React.js).

Machine Learning Libraries: Machine learning algorithms for image classification are implemented using libraries such as TensorFlow, PyTorch, or scikit-learn.

Security Tools: Security tools and protocols are implemented to ensure data privacy, user authentication, and secure communication over the network. This may include SSL/TLS certificates, encryption algorithms, and firewall configurations.

1.Network Infrastructure:

Internet Connectivity: Both the server infrastructure and client devices require internet connectivity to access the web-based application and transmit data over the network.

Bandwidth and Latency: Sufficient bandwidth is needed to support data transmission between client devices and the server, while low latency ensures timely responses during image upload and classification processes.

1.Operating Systems:

The server infrastructure may run on various operating systems, including Linux distributions (e.g., Ubuntu, CentOS) or Windows Server.

Client devices may use different operating systems such as Windows, macOS, iOS, or Android, as long as they support modern web browsers for accessing the system.

1.Development and Deployment Environment:

Development of the system can be carried out using integrated development environments (IDEs) such as Visual Studio Code, PyCharm, or IntelliJ IDEA.

Deployment of the system to production environments may involve containerization technologies (e.g., Docker) and orchestration platforms (e.g., Kubernetes) for scalability and resource management.

Overall, the operating environment of the "Plant Disease Classification" project encompasses a range of hardware, software, network, and development components tailored to support the functionality, scalability, and usability of the system for users in agricultural settings.

5. DESIGN and IMPLEMENTATION CONSTRAINTS

The "Plant Disease Classification" project is subject to various design and implementation constraints that may impact its development and deployment. These constraints include technical limitations, resource availability, regulatory requirements, and project-specific considerations. Here are some common design and implementation constraints for the project:

1. **Data Availability and Quality:** The effectiveness of the disease classification model relies on the availability and quality of labeled datasets containing images of diseased plants. Constraints may arise if sufficient data is not readily available or if the data is of poor quality, leading to limitations in model accuracy and performance.

2. **Computational Resources:** Machine learning algorithms used for image classification, such as convolutional neural networks (CNNs), require significant computational resources for training and inference. Constraints may arise if the available hardware infrastructure is insufficient to support the computational demands of model training and real-time classification tasks.

3. **Model Complexity and Training Time:** Complex machine learning models with a large number of parameters may require extensive training time and computational resources. Constraints may arise if the project timeline or available resources limit the feasibility of training and optimizing complex models.

4. **Scalability:** The system must be designed to scale effectively to accommodate increasing volumes of image uploads and user interactions as its user base grows. Constraints may arise if the system architecture or infrastructure does not support horizontal scaling or if scalability considerations are not adequately addressed during the design phase.

5. **Integration with External Systems:** The system may need to integrate with external systems or APIs to access additional data sources, such as weather data or crop management information. Constraints may arise if compatibility issues or limitations in third-party APIs hinder seamless integration with external systems.

6. **Regulatory Compliance:** The project may be subject to regulatory requirements and data privacy laws governing the collection, storage, and processing of agricultural data and personal information. Constraints may arise if regulatory compliance requirements impose restrictions or additional overhead on system development and operation.

7. **User Interface Design:** The user interface must be designed to be intuitive, accessible, and responsive across different devices and screen sizes. Constraints may arise if design considerations, such as limited screen real estate on mobile devices or compatibility with assistive technologies for users with disabilities, are not adequately addressed.

8. **Security Considerations:** The system must implement robust security measures to protect user data, prevent unauthorized access, and mitigate risks such as data breaches or cyberattacks. Constraints may arise if security vulnerabilities are identified or if compliance with security standards and best practices imposes constraints on system design and implementation.

9. **Localization and Internationalization:** The system may need to support multiple languages, currencies, and regional preferences to accommodate users from different geographic regions. Constraints may arise if localization and internationalization requirements add complexity to the system architecture or impose additional development and maintenance overhead.

10. **Resource Constraints:** The project may be subject to constraints related to budget, time, and human resources. Constraints may arise if limited resources impact the scope, timeline, or quality of system development and implementation.

Addressing these design and implementation constraints requires careful consideration, strategic planning, and collaboration among stakeholders to ensure the successful development and deployment of the "Plant Disease Classification" project while effectively managing risks and constraints.

6. ASSUMPTION DEPENDENCIES

Assumption dependencies refer to the assumptions made during the planning and development of a project that are dependent on each other or impact one another's validity. In the case of the "Plant Disease Classification" project, several assumption dependencies may be relevant:

1. Data Availability and Quality: Assumption: Sufficient labeled datasets of plant images with accurate disease annotations are available. Dependency: The accuracy and effectiveness of the disease classification model depend on the availability of high-quality training data. If this assumption is not met, the model's performance may be compromised.

2. Hardware Infrastructure: Assumption: Adequate computational resources are available for model training and inference. Dependency: The scalability and performance of the system depend on the availability of sufficient hardware infrastructure to support machine learning tasks. If this assumption is not met, the system's efficiency may be limited.

3. User Interaction Patterns: Assumption: Users will interact with the system in expected ways, such as uploading images of plant leaves and reviewing classification results. Dependency: The system's design and functionality depend on assumptions about user behavior and interaction patterns. If these assumptions are incorrect, the user experience may be negatively impacted.

4. Model Accuracy and Performance: Assumption: The machine learning models used for disease classification are accurate and reliable. Dependency: The effectiveness of the system's disease classification functionality depends on the assumption that the machine learning models produce accurate results. If this assumption is not met, the system's reliability may be compromised.

5. Regulatory Compliance: Assumption: The system complies with relevant regulations and data privacy laws. Dependency: The legality and ethicality of the system's data collection, storage, and processing depend on assumptions about regulatory compliance. If these assumptions are incorrect, the system may face legal or ethical challenges.

6. Internet Connectivity: Assumption: Users have reliable internet connectivity to access the web-based application. Dependency: The accessibility and usability of the system depend on assumptions about users' internet connectivity. If users do not have reliable internet access, they may face difficulties in using the system effectively.

7. Crop Variability: Assumption: The symptoms of plant diseases manifest consistently across different crop varieties and environmental conditions. Dependency: The accuracy of the disease classification model depends on assumptions about the consistency of disease symptoms. If these assumptions are incorrect, the model's performance may be affected.

8. User Expertise: Assumption: Users have basic knowledge of plant diseases and agricultural practices. Dependency: The effectiveness of the system's recommendations and user interface design depends on assumptions about users' expertise levels. If users lack the necessary

knowledge, they may struggle to interpret classification results or implement recommended actions.

3. EXTERNAL INTERFACE REQUIREMENTS

1. USER INTERFACES

Description: The system's UI allows users to interact with the application, upload images, view classification results, and access recommendations.

Requirements:

The UI should be intuitive, user-friendly, and accessible across different devices and screen sizes.

It should include interactive elements such as buttons, forms, and dropdown menus for navigation and data input.

Visualizations and feedback mechanisms should be incorporated to enhance user experience and understanding.

2. HARDWARE INTERFACES

Description: The system may interface with external hardware devices such as cameras, sensors, or IoT devices for capturing images or collecting environmental data.

Requirements:

The system should support standard communication protocols (e.g., USB, Bluetooth, Wi-Fi) for interfacing with hardware devices.

Device drivers or libraries may be required to facilitate communication and data exchange between the system and external hardware.

Compatibility testing should be conducted to ensure seamless integration and interoperability with supported hardware devices.

3. SOFTWARE INTERFACES

Backend Services and APIs:

Description: Backend services and APIs handle business logic, data processing, and interaction with external systems, providing the core functionality of the system.

Interface Requirements:

Backend services should expose well-defined APIs for interacting with frontend components and external systems.

APIs should be designed following RESTful principles, with clear endpoints, request/response formats, and error handling mechanisms.

Services should enforce authentication, authorization, and data validation to ensure security and integrity of interactions.

Database Interface:

Description: The database interface handles data storage and retrieval operations, interacting with the underlying database management system (DBMS).

Interface Requirements:

The interface should abstract database operations using an ORM (Object-Relational Mapping) framework or query builder to ensure portability and flexibility.

Data access methods should be encapsulated within data access objects (DAOs) or repository classes, providing a layer of abstraction over database interactions.

Transactions should be managed efficiently to maintain data consistency and integrity across multiple database operations.

4.COMMUNICATION INTERFACES

Description: Communication protocols govern how different software components communicate and exchange data over the network.

Interface Requirements:

Standard protocols such as HTTP(S), WebSocket, or MQTT may be used for communication between frontend and backend components.

Messaging protocols (e.g., AMQP, MQTT) may be employed for asynchronous communication between distributed components or microservices.

Secure communication protocols (e.g., TLS/SSL) should be used to encrypt data transmissions and protect against unauthorized access or interception.

4. SYSTEM FEATURES

1. Image Upload Feature:

Description: This feature allows users to upload images of plant leaves exhibiting symptoms of disease through the system's user interface.

Priority: High

Action/Result: Users can select and upload images from their devices using the designated upload functionality in the system's interface.

2. Functional Requirements:

The system shall provide a user-friendly interface for image upload.

Users shall be able to select and upload images in common formats (e.g., JPEG, PNG).

The system shall validate uploaded images to ensure compatibility and quality.

Users shall receive feedback on successful image uploads and any errors encountered during the process.

3. Disease Classification Feature:

Description: This feature employs machine learning algorithms to analyze uploaded images and classify them into predefined disease categories.

Priority: High

Action/Result: Upon image upload, the system processes the images using pre-trained machine learning models to classify them accurately into disease categories.

Functional Requirements:

The system shall employ machine learning algorithms, such as convolutional neural networks (CNNs), for image classification.

The system shall classify images based on observed disease symptoms and assign probability scores to each disease class.

Classification results shall be presented to users in a clear and understandable format.

The system shall provide real-time feedback on the classification process and any errors encountered.

4. Actionable Recommendations Feature:

Description: This feature provides users with actionable recommendations for disease management and treatment based on the identified diseases.

Priority: Medium

Action/Result: Users receive practical guidance and recommendations for addressing identified plant diseases, including preventive measures and treatment options.

5. Functional Requirements:

The system shall generate actionable recommendations based on the identified diseases and associated symptoms.

Recommendations shall be tailored to specific crops, environmental conditions, and disease severity levels.

Users shall have access to a comprehensive database of disease management strategies and treatment options.

The system shall provide additional resources and references to support users in implementing recommended actions effectively.

User Authentication and Management Feature:

Description: This feature implements user authentication mechanisms to ensure security and accountability.

Priority: High

Action/Result: Users can register for an account or log in using existing credentials, enabling secure access to the system's features.

6. Functional Requirements:

The system shall implement user authentication mechanisms, such as username/password login or social media authentication.

Users shall be able to register for new accounts or recover/reset their passwords if forgotten.

Administrators shall have the ability to manage user accounts, permissions, and access levels.

The system shall enforce security best practices, such as password complexity requirements and account lockout policies.

User-Friendly Interface Feature:

Description: This feature ensures that the system features a user-friendly graphical user interface (GUI) that facilitates seamless interaction with the application.

7. Functional Requirements:

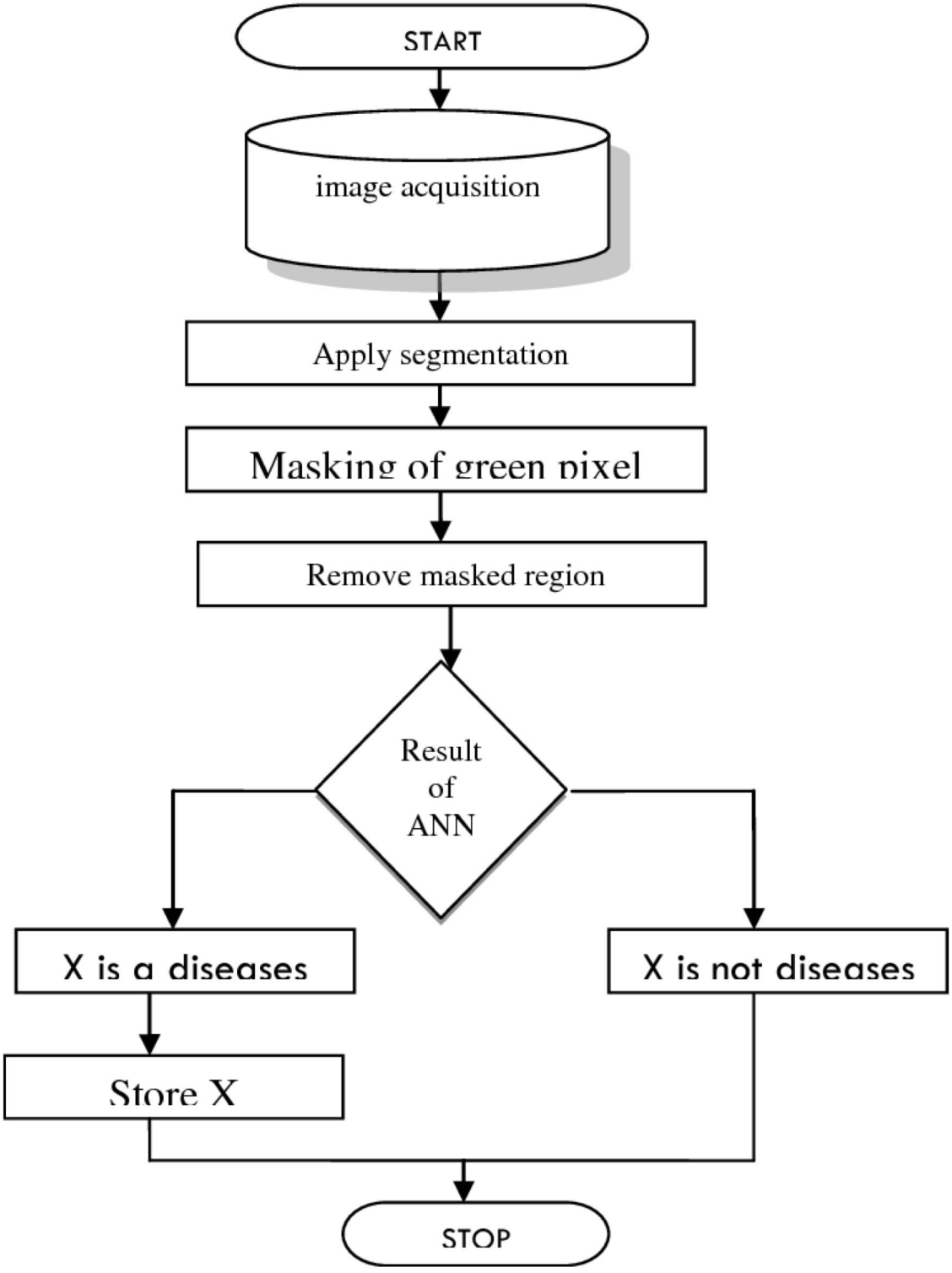
The system shall feature a visually appealing and intuitive GUI with consistent design elements.

Users shall be provided with clear instructions and guidance on using the system's features.

Interactive elements, such as buttons, dropdown menus, and form fields, shall be easily accessible and responsive.

The GUI shall be optimized for usability across different devices and screen sizes, including desktop computers, tablets, and smartphones.

These descriptions, priorities, action/results, and functional requirements provide a comprehensive overview of the core features of the "Plant Disease Classification" project and outline the specific functionalities and capabilities required to meet user needs and project objectives.



Data Flow Diagram of Plant Disease Classification.

5. NON FUNCTIONAL REQUIREMENTS

Non-functional requirements define the attributes of the system beyond its specific functionalities. They focus on aspects such as performance, usability, security, and maintainability. Here are some non-functional requirements for the "Plant Disease Classification" project:

1. Performance:

Response Time: The system should respond to user actions (e.g., image upload, classification) within a reasonable time frame, typically within a few seconds.

Throughput: The system should be capable of handling a large number of concurrent users and image processing tasks without experiencing significant performance degradation.

Scalability: The system architecture should be scalable to accommodate increasing loads and data volumes as the user base grows.

2 . Usability:

User Interface Design: The user interface should be intuitive, visually appealing, and easy to navigate, ensuring a positive user experience for users of varying technical backgrounds.

Accessibility: The system should adhere to accessibility standards (e.g., WCAG) to ensure that users with disabilities can access and use the application effectively.

Multilingual Support: The system should support multiple languages to cater to users from diverse linguistic backgrounds.

3. Security:

Data Privacy: The system should adhere to data privacy regulations (e.g., GDPR) and implement measures to protect user data from unauthorized access, disclosure, or misuse.

Authentication and Authorization: User authentication mechanisms should be secure and robust, with options for multi-factor authentication and role-based access control to ensure proper authorization levels.

Data Encryption: Sensitive data should be encrypted during transmission and storage to prevent interception or unauthorized access.

Security Auditing: Regular security audits should be conducted to identify and address vulnerabilities, with logs maintained for audit trail purposes.

4. Reliability:

Availability: The system should be highly available, with minimal downtime and maintenance windows to ensure continuous access for users.

Fault Tolerance: The system architecture should incorporate fault-tolerant mechanisms to mitigate the impact of hardware failures or network disruptions.

Backup and Recovery: Regular backups of system data should be performed, with procedures in place for data recovery in the event of data loss or corruption.

5. Maintainability:

Code Maintainability: The system's codebase should be well-structured, documented, and modular to facilitate ease of maintenance and future enhancements.

Version Control: Source code should be managed using version control systems (e.g., Git), with clear branching and merging strategies to track changes and collaborate effectively.

Documentation: Comprehensive documentation should be provided for system architecture, APIs, deployment procedures, and troubleshooting guides to assist developers and administrators.

6. Compatibility:

Browser Compatibility: The system should be compatible with major web browsers (e.g., Chrome, Firefox, Safari) to ensure consistent performance and functionality across different platforms.

Device Compatibility: The system should be responsive and optimized for various devices, including desktop computers, laptops, tablets, and smartphones.

7. Regulatory Compliance:

The system should comply with relevant industry standards and regulations governing agricultural technology, data privacy, and security.

Compliance certifications (e.g., ISO 27001) may be obtained to demonstrate adherence to best practices and regulatory requirements.