Software Requirements Specification

for

Potato Leaf Disease Classification

Version 1.0 approved

Prepared by

Dogiparthi Aasrith 21BAI1702, Kalyan Chakravarthy Y 21BAI1759

Hanuma Vihari G 21BRS1022

29-02-2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

Predicting potato leaf disease as soon as possible is crucial because it can have significant impacts on crop yield and quality. Early detection allows farmers to take prompt action to prevent the spread of the disease and reduce crop damage. Here are some reasons why predicting potato leaf disease early is essential:

- **Minimize crop loss:** Potato leaf disease can significantly reduce crop yield and quality. By predicting the disease early, farmers can take timely measures to control its spread, thereby minimizing crop loss.
- **Reduce cost:** Early detection of potato leaf disease can help farmers reduce costs associated with disease control measures, such as pesticides and other treatments. By identifying the disease early, farmers can target the specific area of the crop affected, which helps in reducing the overall cost of control measures.
- **Protect the environment:** The excessive use of pesticides and other control measures can have negative impacts on the environment. Early detection of potato leaf disease can help farmers target only the affected areas and minimize the use of pesticides, reducing their environmental impact.
- **Improve crop quality:** Potato leaf disease can affect the quality of the crop, making it less desirable to buyers. By predicting the disease early, farmers can take appropriate measures to prevent its spread, thereby improving the overall quality of the crop.

1.2 Document Conventions

This document follows the IEEE Software Requirements Specification (SRS) format. Acronyms and abbreviations utilized throughout the document are clearly defined in the glossary section.

1.3 Intended Audience and Reading Suggestions

Intended Audience:

Farmers and agricultural professionals seeking a user-friendly tool for identifying and managing potato crop diseases through advanced AI technology.

Reading Suggestions

Farmers: The user interface is designed with simplicity in mind, allowing farmers to easily upload images of their potato crops.
Data Scientists: The Deep Learning Model is built using Transfer Learning on Resnet50, and VGG19. The dataset contains over 6000 images classified into different diseases.

Investors and Decision-Makers: The potential benefits include, improved crop management, increased yield, and sustainable agricultural practices.
The technology aligns with industry trends and contributes to the overall advancement of agriculture.

1.4 Product Scope

The Potato Disease Classification System, integrated with a Flask Web Application, centers on providing farmers with accurate and timely identification of crop diseases through advanced deep learning models. The system aims to empower users by offering AI-driven insights into their potato crops, facilitating informed decisions for effective disease management and overall crop health.

1.5 References

1. https://www.cambridge.org/core/journals/advances-in-animal-biosciences/article/abs/potato-disease-classification-using-convolution-neural-networks/E9303F667377BD763C3054CB8488D36C

2. https://link.springer.com/chapter/10.1007/978-981-13-8406-6_37

3. https://ieeexplore.ieee.org/abstract/document/9231784

4. https://philpapers.org/rec/ELSPCU

5.https://ieeexplore.ieee.org/abstract/document/8905128

2. Overall Description

2.1 Product Perspective

The potato leaf disease classification system using deep learning is designed to provide an advanced solution for identifying and categorizing diseases affecting potato crops. The product caters to farmers, agricultural researchers, and agritech companies, aiming to enhance disease detection efficiency. Key features include image input for leaf disease identification, deep learning algorithms for classification, a user-friendly interface, compatibility with various image formats, and real-time or batch processing capabilities.

2.2 Product Functions

Image Upload:

- Easy for farmers with varying technical expertise to upload high-quality images of potato leaves.
- Multiple image capture: Take several pictures from different stages of disease progression for more accurate diagnosis.

Image Pre-processing:

- Automatic image resizing and cropping: Ensure consistency for model input.
- Colour and contrast correction: Adjust for lighting variations and enhance disease signatures.
- Noise reduction: Improve image quality and remove background distractions.
- Optional manual adjustments: To fine-tune pre-processing parameters.

Disease Classification:

- AI-powered model analysis: Leverage a machine learning model trained on a diverse dataset of labeled potato leaf images.
- Accurate disease identification: Recognize common potato leaf diseases like late blight, early blight, scab, and leaf roll with high confidence scores.
- Differential diagnosis: Distinguish between similar-looking diseases and potential environmental factors like nutrient deficiencies.
- Real-time results: Provide instant disease classification feedback after image analysis

Result Presentation:

- Clear and concise diagnosis: Display the identified disease name in bold, followed by the confidence score.
- Visual aids: Show corresponding diseased potato leaf images from the dataset for confirmation.
- Detailed information: Provide access to comprehensive information about the identified disease, including:
  - Symptoms and visual characteristics
  - Causes and contributing factors
  - Economic impact and crop loss risks
  - Treatment options and control measures
  - Preventative strategies and best practices

Shareable reports: Allow users to generate and share reports with extension agents, agricultural consultants, or fellow farmers.

2.3 User Classes and Characteristics

- The potato leaf disease classification system caters to a diverse range of user classes, each with distinct characteristics and requirements.
- Farmers, often possessing limited technical expertise, rely on a straightforward and intuitive interface for quick disease identification to inform their crop management decisions. Agricultural researchers, equipped with advanced knowledge, seek detailed disease classification reports and analytics, requiring the system to handle large datasets.
- Agritech companies, with technical proficiency, look for customization options, scalability, and real-time processing capabilities to integrate the system into broader precision agriculture solutions.
- General end users depend on the system for immediate disease identification and clear output reports to guide their agricultural practices.

2.4 Operating Environment

Deployment Platform:

- Localhost: If you're developing or testing the app locally, you can simply run it on your own computer using Python and Flask. You'll need Python 3 installed, along with necessary dependencies like OpenCV for image processing, TensorFlow or PyTorch for your chosen machine learning model, and Flask itself.
- Web Server: For wider accessibility, you can deploy the app on a web server like Apache or Nginx. This requires configuring the server to run Flask applications and potentially setting up virtual environments or containerization (e.g., Docker) for managing application dependencies.

Hardware Requirements:

- Processor: The app's CPU power will influence the speed of image processing and model inference. While basic functionality might run on low-power devices, consider a more powerful CPU for real-time analysis and handling larger image datasets.
- Memory: RAM capacity influences how many images the app can handle simultaneously and overall responsiveness. Aim for sufficient memory to avoid issues with image buffer overflow or slow processing.
- Storage: Disk space is needed for storing the app's code, model files, and potentially uploaded images. Depending on the dataset size and expected user activity, consider using a persistent storage solution like a database or cloud storage.

2.5 Design and Implementation Constraints

Technical Constraints:

- Image quality and variability: Real-world images can be affected by lighting, resolution, angle, background clutter, and partial occlusions, impacting model accuracy.
- Limited training data: Acquiring a large and diverse dataset of labeled potato leaf images representing various diseases and severities can be expensive and time-consuming.
- Model training and optimization: Choosing appropriate machine learning models, training them effectively, and optimizing for accuracy and efficiency can be challenging.
- Computational resources: Running complex AI models on mobile devices might require significant processing power and battery life, especially for offline functionality.
- Device compatibility: Balancing functionality with compatibility across diverse smartphone models or developing a dedicated device for wider reach can be complex.

Data and Algorithm Constraints:

- Data bias and generalizability: Training data could be biased towards specific regions, disease types, or image acquisition conditions, limiting generalizability to new scenarios.
- Algorithm accuracy and explainability: Achieving high accuracy while ensuring the model's decision-making process is transparent and understandable for users can be difficult.
- Evolving diseases and new strains: Continuously updating the model with new disease data and adapting to emerging strains is crucial for long-term effectiveness.

User and Adoption Constraints:

- Limited technical knowledge: Users, especially small-scale farmers, might require training and support to use the tool effectively.
- Language barriers: Localized versions and user interfaces addressing diverse languages and cultural contexts are important for wider adoption.
- Internet access and affordability: Offline functionality or data-efficient models are crucial for areas with limited or expensive internet access.
- Cost and value proposition: The product's cost should be balanced against its perceived value and affordability for the target audience.

Regulatory and Ethical Constraints:

- Data privacy and security: Ensuring user data privacy and security throughout image capture, analysis, and storage is essential.
- Liability and decision-making: Clearly defining the tool's limitations and recommendations to avoid misinterpretations and potential liability issues.
- Ethical considerations: Algorithmic bias and fairness need to be addressed to ensure equitable access and outcomes for different user groups.

2.6 User Documentation

Getting Started:

- Launch the website: Depending on the distribution method,Navigate to the web address where it's hosted.
- Ensure compatibility: Check if your device meets the minimum requirements for running the website smoothly.
- Permissions: Grant the necessary permissions to access your storage.

Using the App:

- Capture an image: Use the camera feature to take a clear picture of your potato leaf, focusing on the affected area. Adjust lighting and position for optimal results.
- Analyze the image: Click the "Analyze" button or follow the app's specific prompt to trigger the analysis process.
- Get results: The app will display the identified disease (if any), its confidence level, and information about the disease, including symptoms, causes, and control measures.
- Additional features: The app might offer features like saving disease histories, accessing additional resources, or sharing results. Explore these functions based on your needs.

Tips for accurate results:

- Take pictures in good lighting conditions, avoiding direct sunlight or shadows.
- Capture close-up images of the affected area, focusing on the most prominent symptoms.
- Ensure the entire leaf is visible in the frame.
- Avoid blurry or low-resolution images.
- If the app identifies an unknown disease or the confidence level is low, consult a professional agricultural advisor for a definitive diagnosis.

Additional Resources:

The app might provide links to relevant websites, articles, or other resources for further information about potato diseases and their management. Utilize these resources to learn more and make informed decisions about your potato crop

Troubleshooting:

- If the app encounters any issues, refer to the FAQ section or contact the app developer for support.
- Provide clear descriptions of the problem and any error messages encountered.

2.7 Assumptions and Dependencies

2.7.1 Assumptions

Deep Learning Model Accuracy: The project assumes that the deep learning model developed for potato disease classification using images will achieve a satisfactory level of accuracy.

Image Dataset Quality: It is assumed that the image dataset used for training the deep learning model is representative and comprehensive.

Flask Framework Compatibility: The project assumes compatibility with the Flask web framework. Any changes or updates to Flask that introduce incompatibilities may require adjustments to the web application.

2.7.2 Dependencies

Deep Learning Framework: The project is dependent on the stability and compatibility of the chosen deep learning framework (e.g., TensorFlow, PyTorch).

Web Application Dependencies: The Flask web application relies on various dependencies, including Flask itself, as well as additional libraries for user authentication and feedback form functionality.

User Feedback Handling: The feedback page assumes a mechanism for collecting and managing user feedback securely.

Server Infrastructure: The successful operation of the Flask web application depends on the availability and reliability of the server infrastructure.

3. External Interface Requirements

3.1 User Interfaces

The Potato Leaf Disease Classification System begins with a secure login page. Once logged in, users can upload potato leaf images for disease classification. The system provides immediate feedback on the upload status and displays classification results, specifying identified diseases. The output page also offers practical remedies for detected diseases. To enhance user engagement, a feedback page is included for users to share their thoughts and suggestions. This concise process ensures a user-friendly experience from login to upload, output, remedies, and feedback.

3.2 Hardware Interfaces

The Potato Leaf Disease Classification System interfaces with a camera or scanner for image input, a robust server for processing, and storage for data management. Output is displayed through monitors or mobile screens, with user interaction facilitated by input devices such as keyboards or touchscreens. Network connectivity ensures communication between components.

3.3 Software Interfaces

The Potato Leaf Disease Classification System (PLDCS) relies on versatile communication interfaces to facilitate seamless interactions. These include network interfaces for data transfer between the user interface and the server, ensuring efficient image uploads, processing, and result retrieval. The user feedback mechanism utilizes interfaces for submitting and receiving feedback, contributing to the continuous improvement of the system.

3.4 Communications Interfaces

The Potato Leaf Disease Classification System (PLDCS) is designed with robust software interfaces to ensure seamless functionality. The system incorporates user interfaces for effortless login, image uploads, and result display, enhancing the user experience. Integration with image processing algorithms and classification models forms crucial software interfaces, enabling accurate disease identification. Database interfaces facilitate efficient data storage and retrieval, while communication interfaces handle interactions between the server and user interfaces. Feedback mechanisms employ dedicated interfaces for users to submit and receive feedback, contributing to system refinement.

4. System Features

4.1 System Feature 1

The Software Requirements Specification (SRS) for the Potato Leaf Disease Classification System outlines key features and functionalities of the system. Here are some potential system features:

1. User Authentication: Secure login and authentication mechanisms to ensure authorized access.

2. Image Upload:Capability for users to upload potato leaf images for disease classification.

3. Image Processing: Integration of image processing algorithms and classification models for accurate disease identification.

4. Classification Results: Display of classification results, indicating whether the potato plant is healthy or diseased, along with specific disease identification.

5. Remedies Information: Presentation of practical remedies and guidance for identified diseases to assist users in addressing plant health issues.

6. Feedback Mechanism: User-friendly interface for users to provide feedback on the system's performance, report issues, or suggest improvements.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1.Performance

- The system should be able to classify images and display results within a reasonable timeframe, ideally within seconds.
- The system should be responsive and minimize delays, even on low-power devices like smartphones.

2.Accuracy

- The disease classification model should achieve high accuracy, ideally exceeding 90% for common potato leaf diseases.
- The system should perform well under diverse lighting conditions and image quality variations.

3.Usability

- The system should be easy to learn and use for farmers with no prior technical experience.
- The UI should be intuitive and minimize the need for training or complex instructions.
- The system should be available in multiple languages to reach a wider audience.

4.Security

- The system should protect user data, including captured images and classification results.
- Secure data storage and transmission protocols should be implemented.
- User privacy should be respected, and data should not be shared without explicit consent.

5.2 Safety Requirements

1. Data safety and security:
    - User data protection: Implement secure data storage and encryption practices to protect user information like location, farm data, and image records.
    - Privacy considerations: Obtain explicit user consent for data collection and usage, adhere to data privacy regulations (e.g., GDPR), and provide transparent information about data handling practices.
2. Safe disease identification and control:

- Accuracy and reliability: Ensure the disease classification model is trained on a diverse and high-quality dataset, and provides accurate diagnoses with clear confidence scores.
- Differential diagnosis: Differentiate between similar-looking diseases or environmental factors to avoid misdiagnosis and incorrect treatment recommendations.
- Safe control measures: Offer information and recommendations for disease control based on scientific evidence and best practices, avoiding promoting the use of harmful or unauthorized chemicals.

3. Environmental impact and sustainability:
    - Promote biocontrol and environmentally friendly methods: Encourage the use of organic and sustainable disease control methods whenever possible, minimizing reliance on chemical pesticides.
    - Data-driven insights for informed decision-making: Facilitate data analysis and trend identification to help farmers optimize resource allocation and minimize chemical use.
4. User safety and training:
    - Clear instructions and warnings: Provide clear instructions on app usage, potential limitations of the technology, and safety precautions when handling diseased plants or chemicals.
    - Limited access to sensitive information: Restrict access to information on highly infectious or dangerous diseases to authorized personnel with relevant expertise.
    - Training and education: Offer educational resources and training programs for users on disease identification, safe control methods, and responsible agricultural practices.

5.3 Security Requirements

1. Data Security:
    - Data encryption: Implement encryption at rest and in transit for sensitive user data such as location, farm records, and image captures.
    - Secure storage: Store user data on secure cloud platforms with strong access controls and intrusion detection systems.
    - Data access control: Implement role-based access control to restrict access to sensitive information based on user permissions.
    - Data anonymization and aggregation: When sharing or analyzing data, ensure user privacy by anonymizing or aggregating data sets.
    - Data deletion policies: Implement clear policies for data retention and deletion based on user consent and regulatory requirements.
2. User Education and Awareness:
    - Privacy policy and terms of service: Clearly explain user data collection, usage, and sharing practices in a transparent privacy policy and terms of service.
    - Security awareness training: Educate users about safe app usage, password hygiene, and phishing scams to avoid potential security risks.

- Reporting mechanisms: Create easy-to-use channels for users to report suspicious activity or potential security vulnerabilities.

5.4 Software Quality Attributes

1. Performance:
    - Accuracy and reliability: The app should consistently provide accurate disease diagnoses with high confidence scores, minimizing misdiagnoses that could lead to incorrect treatment and potential crop losses.
    - Response time: Image capture, disease classification, and result presentation should occur within a reasonable timeframe, ideally within seconds, to facilitate timely decision-making for farmers.
2. Usability:
    - User interface (UI) simplicity and intuitiveness: The UI should be clear, concise, and easy to navigate even for users with limited technical knowledge. Large icons, minimal text, and visual aids like pictures and videos can enhance usability.
    - Multilingual support: Support for diverse languages catering to a wider user base and ensuring information accessibility for various regions and populations.
    - Accessibility features: Compliance with accessibility standards for font size, colour contrast, and compatibility with assistive technologies.
3. Reliability and Availability:
    - Robustness and error handling: The app should be able to handle unexpected inputs, network outages, and device limitations gracefully, providing clear error messages and maintaining functionality as much as possible.
    - System uptime and stability: Minimize downtime and ensure consistent availability of the app's core functionalities even under heavy user load or challenging environmental conditions.
    - Data security and backups: Implement robust data security measures to protect user information and agricultural data from unauthorized access, breaches, or accidental loss. Regular backups and disaster recovery plans should be in place to ensure data resilience.
4. Maintainability and Scalability:
    - Modular design and clean code: The app's codebase should be well-organized, modular, and documented to facilitate future maintenance, updates, and feature enhancements.

- Testing and quality assurance: Implement comprehensive testing procedures to identify and address bugs, crashes, and performance issues before deployment and throughout the app's lifecycle.
- Scalability and adaptability: The app should be designed to accommodate future growth, handle increasing user numbers, and integrate with new technologies or data sources as needed.

5.5 Business Rules

1. Disease Classification Logic:
    - Confidence score thresholds: Establish minimum confidence scores for disease diagnoses to ensure sufficient accuracy and minimize misdiagnosis risks. For example, require a score of 90% or higher for confirmed disease identification.
    - Differential diagnosis protocols: Define criteria for identifying similar-looking diseases or environmental factors that could mimic disease symptoms. Implement additional analysis or user input when confidence scores are close or ambiguous.
    - Uncertainty handling: Determine how the app handles situations where the AI model is unable to reach a definitive diagnosis due to image quality, unclear symptoms, or limitations of the model itself. Offer alternative suggestions like consulting an agricultural expert.
2. Information and Recommendation Delivery:
    - Disease severity levels: Categorize disease severity based on image analysis and provide tailored recommendations for control measures based on the severity level.
    - Treatment efficacy and safety: Prioritize recommendations for safe and effective control measures based on scientific evidence and best practices. Avoid promoting unapproved or harmful chemicals or pesticides.
    - Organic and sustainable options: Encourage the use of organic and environmentally friendly control methods whenever possible, aligning with sustainable agricultural practices.
3. User Interaction and Support:
    - Transparency and user education: Provide transparent information about the app's limitations, accuracy rates, and data usage practices. Offer educational resources and training materials on disease identification, control methods, and safe agricultural practices.
    - Feedback mechanisms: Implement channels for users to provide feedback on app functionality, diagnosis accuracy, and recommendations. Utilize this feedback to improve the app and address user concerns.
    - Support and troubleshooting: Offer readily accessible support options for users encountering technical difficulties or needing assistance with app usage.

6. Other Requirements

Sustainability and Environmental Considerations:

- Promote integrated pest management (IPM): Encourage a holistic approach to disease control that combines biological, cultural, and mechanical methods alongside chemical application where necessary.
- Minimize chemical use: Promote awareness of potential environmental and health risks associated with excessive pesticide use and encourage alternative control methods.
- Data-driven insights for optimization: Integrate data analysis and disease trend information to recommend resource-efficient control strategies and minimize unnecessary chemical inputs.

Social and Economic considerations:

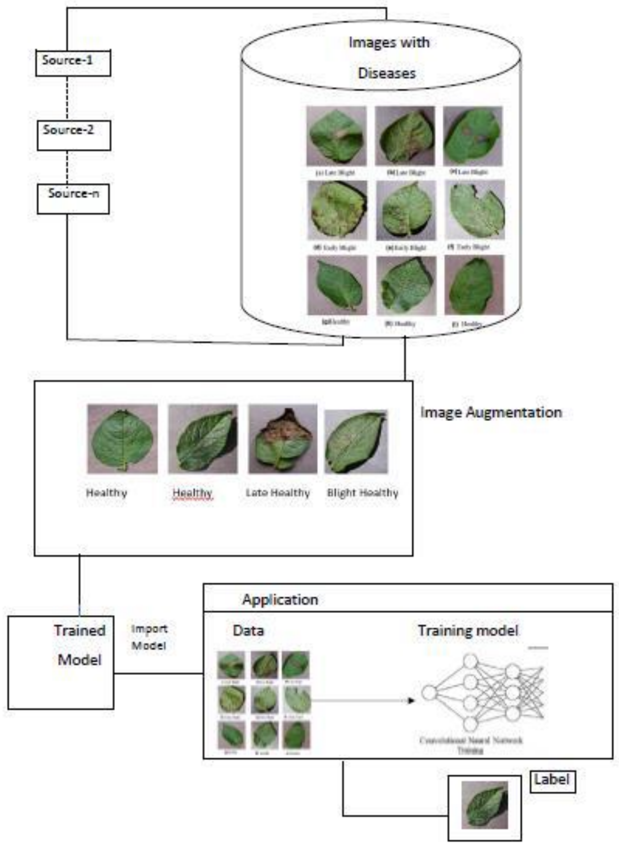
- Accessibility and affordability: Ensure the app is accessible and affordable for small-scale farmers in resource-limited settings. Consider alternative pricing models, partnerships with NGOs or agricultural organizations, and offline functionality to address affordability concerns.
- Capacity building and training: Offer educational resources and training programs for farmers on disease identification, safe control methods, and responsible agricultural practices. Partner with local extension agents and agricultural institutions to ensure effective knowledge dissemination.

Appendix A: Glossary

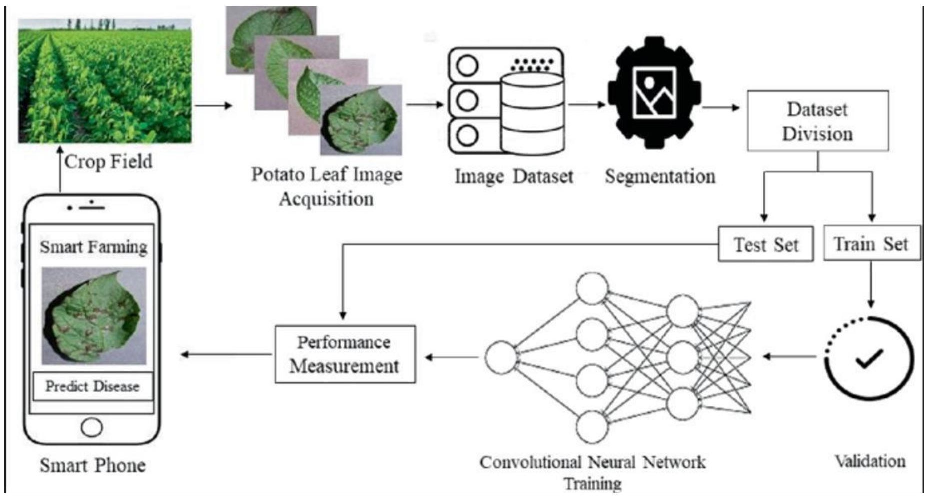
CNN Model: Convolution Neural Network

Appendix B: Analysis Models

Data Flow Diagram:



Model:


Crop Field
Potato Leaf Image Acquisition
Image Dataset
Segmentation
Dataset Division
Test Set
Train Set
Smart Farming
Predict Disease
Smart Phone
Performance Measurement
Convolutional Neural Network Training
Validation