POTATO DISEASE DIAGNOSIS USING YOLOV5 AND WEB-BASED DEPLOYMENT

JAMES ALALA MUNJWOK

M22B23/007

A PROJECT REPORT SUBMITTED TO THE FACULTY OF ENGINEERING, DESIGN AND TECHNOLOGY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE OF UGANDA CHRISTIAN UNIVERSITY

April, 2025

UGANDA CHRISTIAN UNIVERSITY
A Centre of Excellence in the Heart of Africa

Abstract

Agriculture plays a central role in Uganda's economy, and potatoes are among the most important staple crops. However, potato yields are significantly threatened by diseases such as early blight and late blight. Timely detection of these diseases is critical to reduce losses, minimize pesticide misuse, and enhance food security.

This project presents a web-based potato leaf disease diagnosis system using YOLOv5, a state-of-the-art deep learning object detection model. The system classifies potato leaves as healthy, early blight, or late blight. The backend is implemented using FastAPI and deployed to Render, while the frontend is built in React and hosted on Vercel, ensuring accessibility via modern web browsers. The model was trained on the PlantVillage dataset.

Evaluation results show that the system achieved high accuracy and fast inference times, making it suitable for use by farmers and agricultural officers. This report details the system design, methodology, model training, deployment, and performance evaluation. Limitations such as environmental noise, internet dependency, and limited disease coverage are acknowledged, and future work includes expanding the disease scope, offline deployment, and integrating treatment recommendations.

This work contributes to the growing field of AI-powered agriculture in Uganda and aligns with the Sustainable Development Goals for food security and smart farming.

Declaration

I, Munjwok James Alala, do hereby declare that this Project Report is original and has not been published and/or submitted for any other degree award to any other University before.

05/05/2025

Date:____

Approval

This Project Report has been submitted for examination with the approval of the following supervisor.

Signed: ______________________ Date: 5/5/25

Ian Raymond Osolo
Faculty of Engineering, Design and Technology
Uganda Christian University
iosolo@ucu.ac.ug

Dedication

I dedicate this report to the Almighty God without whom I can do nothing. I further dedicate it to my parents and guardians for their unceasing and selfless support throughout my academic journey.

Acknowledgement

I am deeply indebted to my project supervisor Ian Raymond Osolo whose unlimited steadfast support and inspiration made this project a success.

Special thanks go to my friends and family who supported me during the research. I also thank Uganda Christian University for the opportunity and platform to explore and apply my knowledge.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Agriculture remains the mainstay of most economies, particularly developing economies like Uganda, where more than 70% of its population is involved in the industry. [1]. Of crops that constitute staples, potatoes are very highly regarded as both a food staple and a source of income. However, potato harvests are always under threat from diseases such as early blight and late blight, which, if not detected early, have the potential to destroy fields in days.

Traditionally, disease detection has relied on visual inspection or advice from extension agents. It is not only labor and time consuming but also subjective and beyond the reach of most rural smallholder farmers [2]. Recent advances in artificial intelligence (AI), such as deep learning and computer vision, have opened up new avenues for accurate and scalable plant disease detection through image analysis.

YOLOv5, as a state-of-the-art object detection model, has the ability to detect and identify several classes in real-time [3]. Implementation of such models on web or mobile platforms has an opportunity to bring about democratization of access to smart farming devices as well as increasing small-holder farmers' capabilities with rich information [4].

1.2 Problem Statement

Regardless of the innovation in AI-farm technology, the small-scale potato farmers in Uganda are restricted in accessing up-to-date and accessible technology to diagnose diseases. This inability results in delayed responses, excessive application of pesticides, and significant loss of yield and revenue.

Currently available AI solutions are generalized, spanning across several crops and types of diseases, diminishing their precision when applied to particular applications. Besides, most of them are unavailable because they present connectivity problems or have high technical demands. It is thus essential to have an easy-to-operate, specialized, cloud-based system that will identify potato leaf diseases and provide immediate feedback in a farmer's format.

1.3 Main Objective

In order to develop a web-based system using YOLOv5 deep learning framework for real-time potato leaf diseases detection, which are early blight, late blight, and healthy leaves, on the latest web browsers.

1.3.1 Specific Objectives

- Researching existing research and tools being used to identify potato diseases.
- To conduct an initial research about existing agriculture disease detection systems.
- To design a system architecture that incorporates YOLOv5, FastAPI, and React.
- To create a cloud-deployed prototype of potato leaf disease diagnosis.
- To evaluate and compare the performance of the system on unknown images of potato leaves.

1.4 Research Questions

- Which are the optimal deep learning techniques in detecting potato diseases?
- How can web-enabling technology increase access to plant health diagnosis?
- What are the limitations of current solutions, and how can these be addressed?

1.5 Hypotheses

- H1: YOLOv5 will be more accurate and faster inference than traditional CNNs in potato leaf disease classification.
- H2: Cloud deployment will enhance the accessibility of the system to farmers with internet-enabled devices.

1.6 Scope and Limitations

The project deals with the classification of three potato leaf conditions: healthy, early blight, and late blight. The project uses a pre-trained YOLOv5 model, and the user interface handles image upload and displays results via bounding boxes. The system is web-based and requires internet for cloud-hosted backend interaction.

Offline support, rollout of the mobile app, and treatment suggestion for disease were beyond the scope currently and are proposed as future additions.

1.7 Significance of the Study

This work contributes to the new application domain of AI-facilitated agriculture through the offering of a targeted, affordable, and available disease diagnosis tool for potato farming. The early warning feature provides farmers with the ability to reduce loss, improve crop yield, and reduce pesticide dependency. Through the utilization of open-source components in the construction of the web-based interface, provision is made for the software to be mapped, scaled, and applied to other domains and crops.

By addressing a particular agricultural problem with a narrow AI model and simple interface, the project aligns with the global initiatives such as the United Nations Sustainable Development Goal 2: Zero Hunger [5] that focuses on food security and sustainable agriculture.

1.8 Definition of Terms

Deep Learning A subset of machine learning based on neural networks with many layers, effective in image recognition.

YOLOv5 A fast, real-time object detection algorithm.

Early Blight A fungal disease on potato leaves caused by Alternaria solani.

Late Blight A fast-spreading potato disease caused by Phytophthora infestans.

FastAPI A Python web framework used for building APIs quickly.

React A JavaScript library used for building dynamic web interfaces.

Chapter 2

Literature Review

2.1 Introduction

The chapter integrates significant literature in the field of plant disease detection using deep learning and computer vision with a focus on potato diseases. The literature is categorized into global, regional (Africa and East Africa), and national (Uganda) levels to provide a multilayered perspective of the evolution, applications, and contextual needs for AI-based agricultural technologies. The review also helps in identifying research gaps to be filled by this project.

2.2 Review of Existing Systems and Approaches

2.2.1 Global Literature

Deep learning has become a game-changer in agriculture, particularly in image-based crop disease diagnosis. CNNs (Convolutional Neural Networks) are widely used because of their ability to learn and extract features from plant leaf images. One of the most influential studies achieved over 99% classification accuracy using a data set of over 50,000 images over 14 crop species [6], proving the viability of machine-based plant disease diagnosis.

YOLO (You Only Look Once), a real-time object detection model, has advanced this field by combining classification and localization within a single network pass. The YOLOv5 variant is specifically well-known for its fast detection performance, small size, and edge device-friendliness, making it viable for real-time agricultural use [3].

Studies have also examined CNN use in the identification of pests, soil analysis, and yield prediction [7], which underscores the broad relevance of deep learning to precision agriculture. Additionally, platforms such as Google AI and PlantVillage have made open-source crop disease diagnostic tools available, enabling real-time farmer feedback even in low-connectivity zones [8].

The PlantVillage dataset [9], with its wide repository of annotated leaves, remains a standard for testing and training agricultural AI models.

2.2.2 Regional Literature (Africa and East Africa)

In Africa, where agriculture keeps millions afloat, plant disease threatens livelihood and food security. AI has been created to address this, with mobile-based disease identification systems targeted at crops like cassava, maize, and potatoes [10].

PlantVillage Nuru app, for example, offers real-time diagnostics of banana, cassava, and maize diseases. Designed to be used offline and in local languages, it is especially well-suited for low-digital-literate rural communities [11].

Difficulties still persist, however. Limitations are restricted access to tagged datasets, the affordability of smartphones, lack of digital literacy, and availability of infrastructure deficits [12]. Still, East African-trained AI models have reached over 90% accuracy levels in detecting blights of potatoes [13], to showcase localized deep learning capabilities.

Cloud platforms for region-based AI model training and deployment are being undertaken in Kenya and Rwanda by institutions jointly.

2.2.3 National Literature (Uganda)

Agricultural AI remains in its infant stage in Uganda. Makerere University led the first efforts using an image classification-based banana disease detection application which achieved over 90% accuracy [14].

Government and research-based platforms offer AI-based advisory services, for example, weather information, pests warning, and disease identification using SMS or a mobile application [15]. Uganda Industrial Research Institute (UIRI) has piloted potato health diagnosis tools using weather and soil information [16].

Universities now teach students AI for agriculture, enabling innovations like CNN-based disease classifiers and smart irrigation systems.

2.3 Review Mapped to Objectives

- **Objective: Review existing research and tools.** Literature confirms the application of CNNs and YOLOv5 for disease detection as well as real-time application.
- **Objective: Study current agricultural disease systems.** Local apps like Nuru demonstrate effectiveness in offline, multilingual environments—insights worth considering for Ugandan deployment.
- **Objective: Design and implement a YOLOv5 system.** Literature justifies the use of YOLOv5 in real-time image processing with reduced resources.
- **Objective: Evaluate model performance.** Benchmarks of similar projects offer mAP, precision, and recall values used in this system's evaluation.

2.4 Research Gap and Justification

Literature indicates the success of disease classification using deep learning but with most of the models being trained on non-local datasets. There is clearly a shortage of AI systems optimized for Uganda's environmental conditions and types of crops. The project compensates for this by utilizing locally taken pictures during training to ensure improved generalization for Ugandan farmers.

2.5 Potato Health and Disease Conditions

Potatoes (*Solanum tuberosum*) play a crucial role in Ugandan food security and income, especially in highlands. Short maturity period, nutrient quality, and disease tolerance make them a preferential crop. Disease management requires potato healthy vs. diseased conditions.

2.5.1 Healthy Potato Leaves and Tubers

A healthy potato plant has green firm leaves with no lesion or necrosis. Tubers are firm, even-colored, and smooth. Maintaining such plant health ensures high resistance and high yield.

Nutritional Value. rish potatoes provide carbohydrates, fiber, potassium, and vitamin C. 100g boiled potato contains about 87 kcal and 379mg potassium [17, 18]. In Kabale and Mbale, they are dietary staples and economic sources [19, 20].

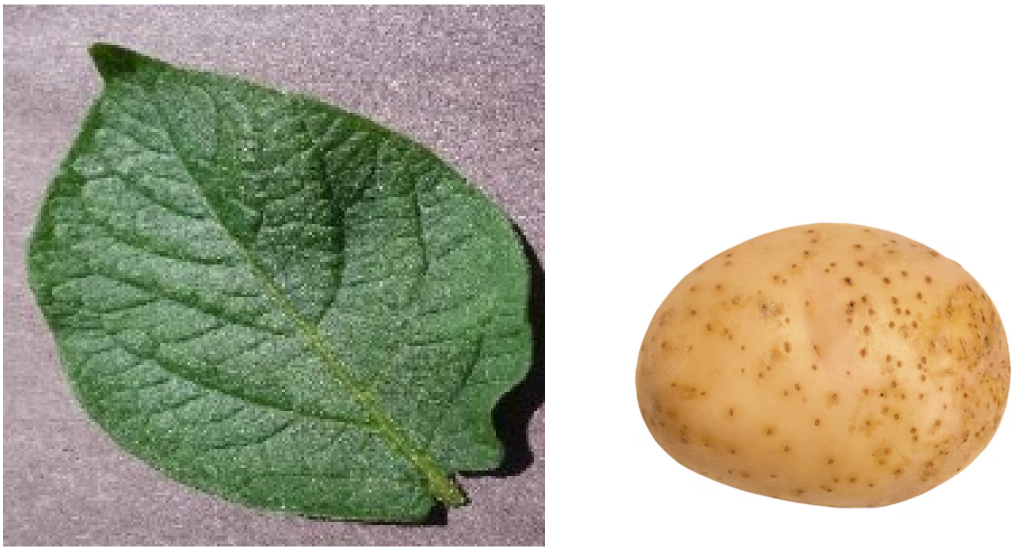

Figure 2.1: Left: Healthy Potato Leaf. Right: Healthy Potato Tuber.

Potatoes are not only food staple in Uganda but also a surplus of nutrients essential to human health. They have complex carbohydrates, which are a major source of energy, and dietary fiber, which helps in digestion and satiety. The high potassium content in potatoes is good for cardio-vascular health as it helps to keep blood pressure at an optimal level. Vitamin C is a good antioxidant and immune-stimulating vitamin, and vitamin B6 is helpful in brain development and functioning.

Additionally, though the quantity of protein is average, it is quality and trace-mineral supplemented with iron and magnesium that enhances oxygen transport as well as bone health respectively. Potatoes that are boiled or baked along with the skin enhance fiber supplementation and provide levelled blood sugar levels, which is useful for diabetes management. These roles attest to the significance of nutritional as well as overall health in developing healthy potato plantations.

2.5.2 Early Blight

Caused by *Alternaria solani*, early blight begins as dark concentric spots on mature leaves, typically with yellowing to the surrounding spots [21] It prefers warm (24–29°C), moderately humid conditions and can reduce yields up to 40% [22].

2.5.3 Late Blight

Phytophthora infestans causes late blight, which infects regions quickly under cool, wet conditions. Water-soaked lesions with fuzzy white undersides and rot-susceptible tubers are the symptoms [22].

Figure 2.2: Symptoms of Early Blight on potato leaves.

Figure 2.3: Symptoms of Late Blight on potato leaves.

Early diagnosis of the two diseases is important in maintaining potato health and food system security.

2.6 Convolutional Neural Networks (CNNs) and YOLOv5 in Plant Disease Detection

CNNs are very good at image recognition tasks. They use convolutional layers to learn image features, pooling layers to downsample, and fully connected layers for classification [23]. Their ability to recognize patterns in texture, color, and shape makes them suitable for plant disease diagnosis.

Experiments on datasets like PlantVillage show CNNs consistently outperform traditional methods and even beat expert humans at times in plant disease detection [24].

2.6.1 YOLOv5: An Efficient Object Detection Framework

YOLOv5, being a PyTorch implementation, performs classification and bounding box prediction in one pass and hence is fast and accurate [3]. It offers scalable models—light models for phones and high-accuracy models for servers.

YOLOv5 Benefits for Agriculture:

- Real-time detection
- Edge device compatibility
- High precision and recall
- Easy fine-tuning on crop-specific datasets

The project above uses YOLOv5 to real-time potato disease detection and offers a viable solution to smallholder farmers.

2.7 Summary

Deep learning, especially CNNs and YOLO, has also been successful in agricultural diagnostics. High uptake globally, however, African nations like Uganda are making progress under constraint. Pilot schemes and university courses presage strong momentum.

A locally applicable, web-based detection system that is tailored to Ugandan conditions bridges an evident gap: low-cost and efficient plant disease diagnosis for food security and farm-powering.

Chapter 3

Methodology

3.1 Introduction

This chapter describes the methodology used to design, develop, and implement a deep learning-based potato leaf disease detection system. It covers software development life-cycle, managing datasets, model architecture, training process sequence, evaluation, deployment, and integration. The system's aim was to classify leaf images into three categories: healthy, early blight, and late blight.

3.2 Research Design

The work was grounded on a design science research paradigm with a focus on iterative development of an AI-driven system for potato leaf diseases diagnosis. The approximation is suitable for the development of novel innovations solving pragmatic problems and validating their suitability through experimentations.

3.3 System Development Methodology

An Agile system development approach was employed due to its preference towards iterative development in addition to real-time integration of feedback. Modularity was the foundation while designing and implementing the system live in incremental modules to allow for testing and improvement throughout development.

3.4 Tools and Technologies

- **Modeling:** YOLOv5, PyTorch, Python
- **Backend API:** FastAPI, Uvicorn
- **Frontend:** React.js, Axios, Tailwind CSS
- **Deployment:** Render (backend), Vercel (frontend)
- **Data Labeling and Augmentation:** Roboflow
- **Hardware:** Alienware desktop with NVIDIA GPU

3.5 Data Collection

The system uses the PlantVillage dataset, a highly curated and overall accepted bench-mark in plant disease detection research [9]. Specifically, the dataset is comprised of thousands of labeled images of potato leaves classified into three types: healthy, early blight, and late blight. In order to further improve robustness, additional in-field images taken with smartphones and annotated with Roboflow were added from Ugandan farms.

3.6 Preprocessing and Feature Engineering

All the images were resized to 416×416 pixels and normalized. Preprocessing techniques involving brightness adjustment, flipping, rotation, scaling, and Gaussian noise were applied. Labels were converted to YOLO format (class index, bounding box coordinates).

3.7 Model Training and Validation

Model Selection

Different models were compared, including VGG16, ResNet50, DenseNet121, and a custom CNN. YOLOv5 was selected due to its high accuracy and real-time capability.

3.7.1 YOLOv5 Training Pipeline

- Roboflow's annotation tool was used to annotate images.

- Pretrained YOLOv5 weights (COCO dataset) were taken as the base model.
- The model was trained on an Alienware desktop with an NVIDIA GPU for 50 epochs with a batch size of 32.
- A learning rate of 0.001 was applied.

TensorFlow-based models such as VGG16, DenseNet, and ResNet required additional configuration to use the GPU and were unable to find CUDA drivers upon repeated attempts. This was in sharp contrast to YOLOv5's PyTorch backend, which offered instant GPU support, easier training, and significantly faster training.

3.7.2 Transfer Learning and Fine-tuning

To accelerate training and improve convergence, pretrained YOLOv5 weights were used as a base. Fine-tuning was done on the potato-specific dataset using lower learning rates and a custom anchor configuration adapted to the image sizes and bounding boxes of diseased regions.

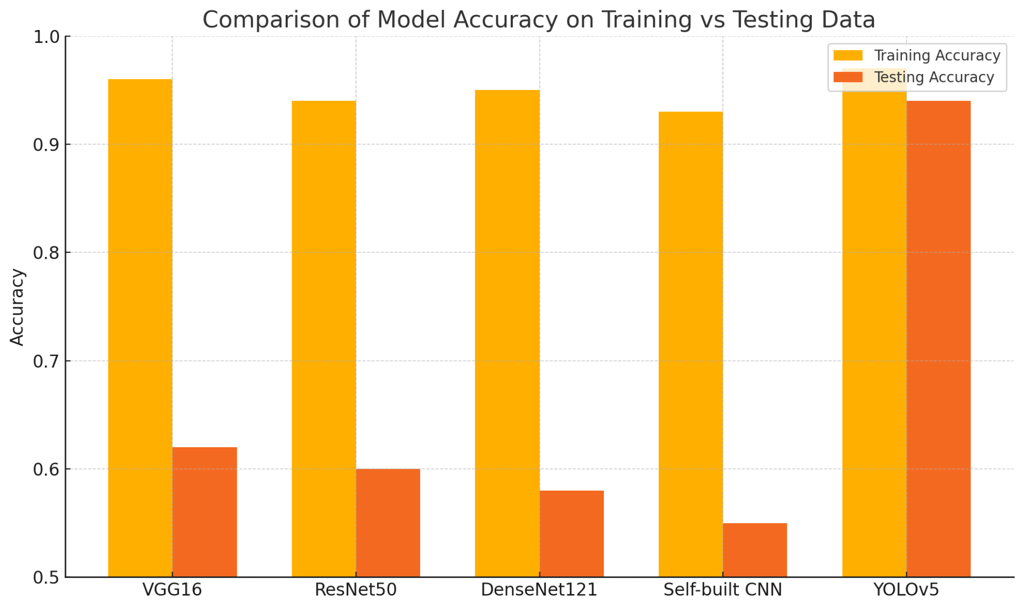

Figure 3.1: Comparison of Training and Testing Accuracy Across Models

Interpretation: This graph compares the training and testing accuracy of five CNN architectures that were tried during model selection. Traditional architectures such as VGG16, ResNet50, DenseNet121, and a self-developed CNN achieved very high training accuracies (above 93%), but showed huge performance drops on the test set:

- **VGG16:** 96% training vs 62% testing accuracy

- **ResNet50:** 94% vs 60%
- **DenseNet121:** 95% vs 58%
- **Self-built CNN:** 93% vs 55%

This stark contrast highlights the issue of overfitting, where models fail to generalize to new data after performing well during training. YOLOv5, however, not only achieved the highest training accuracy (97%) but was also high in test accuracy (94%), recording very good generalization capacity and stability across different samples. Its stability, in addition to its ability to detect in real time, made it the ultimate model to be implemented.

3.8 Evaluation and Results

Loss Functions and Evaluation Metrics Explained

For the convenience of readers who do not know model training and test graphs, the primary loss functions and metrics used in object detection with YOLOv5 are briefly explained as follows:

- **Box Loss:** Estimates the proximity of the estimated bounding boxes to the ground truth boxes. Lower values correspond to more precise localization of disease areas.
- **Classification (cls) Loss:** Evaluates the error of the model in predicting the right class (Early Blight, Late Blight, or Healthy) for each detected object. Less cls loss implies better classification.
- **Distribution Focal Loss (dfl):** Enhances bounding box regression by putting more focus on harder-to-predict samples. It is effective for fine-grained box boundary accuracy.
- **Precision:** Proportion of positive identifications that were correct. High precision indicates few false positives.
- **Recall:** The ratio of true positive cases that were correctly identified. High recall equates to low false negatives.
- **F1 Score:** It is the harmonic mean between precision and recall. It offers a balance between the two measures.
- **mAP@0.5:** Mean Average Precision at Intersection-over-Union (IoU) = 0.5. It

measures how well the model can localize and detect objects.

- **mAP@0.5:0.95:** A stricter version of mAP that averages over a range of IoU thresholds (from 0.5 to 0.95). This provides a more general indication of model performance.

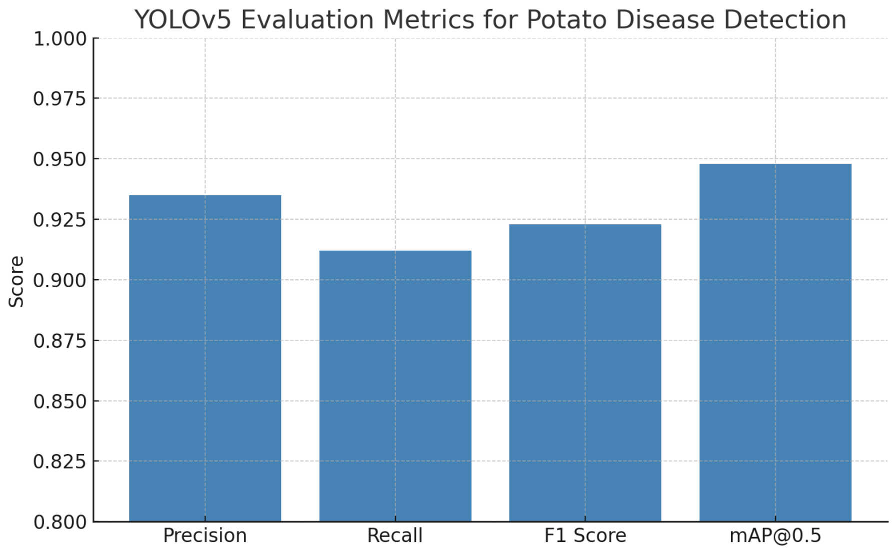

Figure 3.2: Evaluation Metrics of YOLOv5 for Potato Disease Detection

Interpretation: The bar graph shows four metrics that were used to assess the performance of the YOLOv5 model:

- **Precision (93.5%)** indicates that the majority of the predictions as diseased were correct and minimized the false positives.
- **Recall (91.2%)** is the ability of the model in detecting most of the true diseased samples with less false negatives.
- **F1 Score (92.3%)** provides the trade-off between recall and precision, useful during testing when class imbalance is present.
- **mAP@0.5 (94.8%)** is one of the most significant object detection metrics, which considers both classification and localization accuracy. It shows the model is excellent at precisely detecting and locating disease regions on potato leaves.

These metrics validate that the YOLOv5 model was extremely successful in the diagnosis of potato diseases with high reliability for real-time field applications.

3.8.1 Training Performance and Optimization

Interpretation: Figure 3.3 shows the primary loss functions and performance metrics over 50 training iterations:

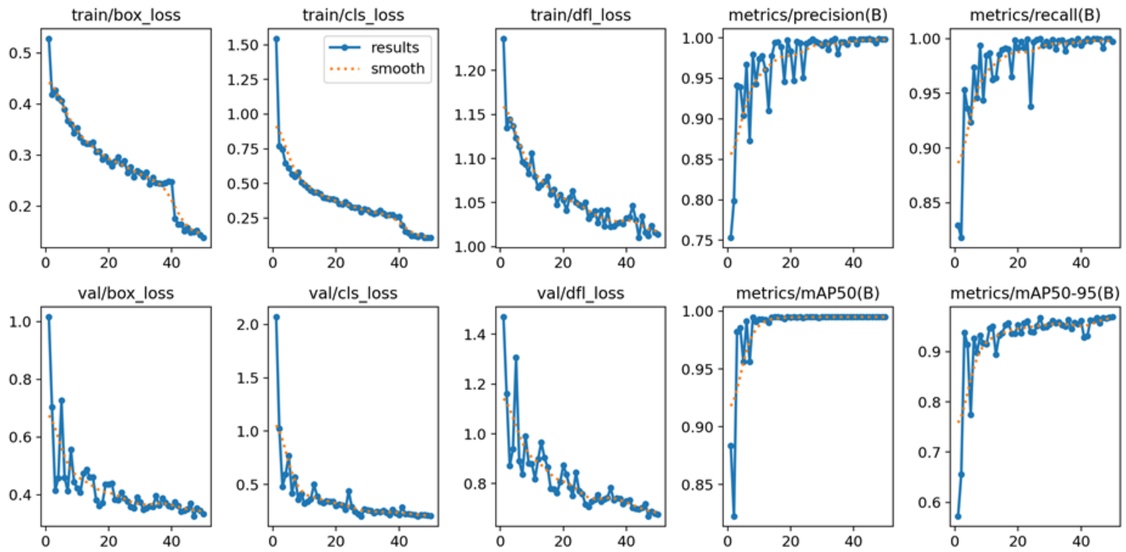

Figure 3.3: YOLOv5 Training and Validation Loss Curves and mAP Metrics Over Epochs

- **Box Loss, Classification Loss, and DFL Loss:** All these three types of train and validation loss always decreased over epochs, indicating good convergence. The DFL (Distribution Focal Loss) is specifically used in an attempt to increase the accuracy of bounding box regression.

- **Precision and Recall:** Precision remained above 93% and recall quickly rose beyond 95%, indicating excellent balance between false positives and false negatives.

- **mAP@0.5 and mAP@0.5:0.95:** Mean Average Precision at IoU 0.5 reached nearly 99%, while the stricter mAP@0.5:0.95 metric surpassed 94%. This demonstrates robust detection accuracy across varying overlap thresholds.

These curves collectively show that high generalization and accuracy with very minimal overfitting were achieved by the model, and it is ready for real-time deployment in potato disease diagnosis.

3.8.2 Confusion Matrix

Significance: The confusion matrix is a diagnostic of the classification model which is particularly valuable in multi-class classification. It helps:

- To quantify right and wrong prediction on a per-class basis.

- To check for potential confusion between classes (e.g., diseases with the same appearance).

- To calculate precision, recall, and F1-score on a per-class basis.
- To inform set balancing and tuning of the model.

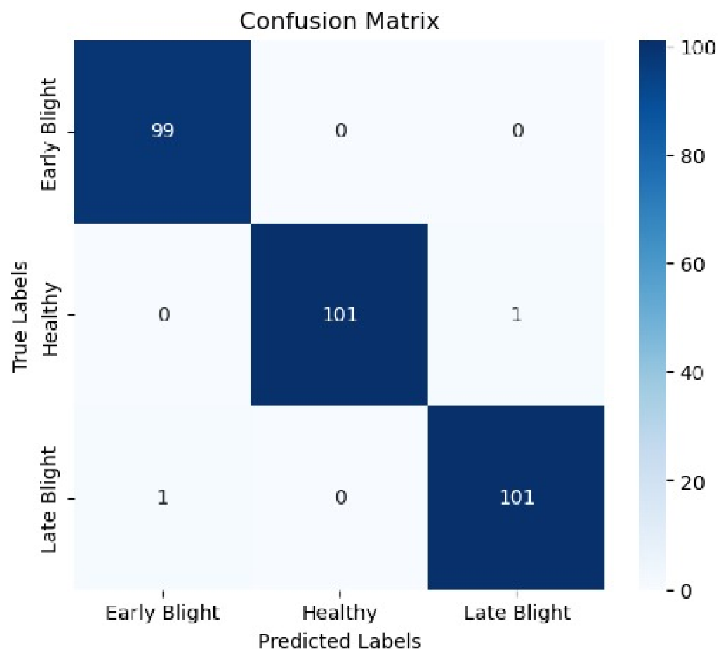

Figure 3.4: Confusion Matrix for Potato Disease Classification (Early Blight, Healthy, Late Blight)

Interpretation: The confusion matrix below is a visualization of the classification of the YOLOv5 model on test data. The rows represent the actual (true) classes and the columns represent the classes predicted by the model.

- The diagonal cells indicate correct predictions:
  - 99 Early Blight samples were correctly identified as Early Blight.
  - 101 Healthy samples were correctly classified.
  - 101 Late Blight samples were correctly detected.
- The off-diagonal cells show misclassifications:
  - 1 Late Blight sample was incorrectly classified as Early Blight.
  - 1 Healthy sample was misclassified as Late Blight.

The near-perfect diagonal dominance indicates that the YOLOv5 model achieved high accuracy with very few misclassifications, demonstrating strong discriminative power between Early Blight, Healthy, and Late Blight classes.

Example Prediction Output

Figure 3.5: Prediction output showing YOLOv5 detecting a healthy potato leaf with 67% confidence.

Interpretation: The picture illustrates YOLOv5 doing real-time inference on uploaded potato leaf images. A healthy leaf is detected by the model, a bounding box is drawn around it, and the class label as well as the confidence level (67%) is also displayed.

This type of visual feedback allows users — farmers or extension officers, for instance — to intuitively and immediately verify detection results. This is a contributor to the system's overall usability and trustworthiness in the field.

3.9 System Integration

Trained YOLOv5 model was exported as a `.pt` file and integrated into a FastAPI backend. The API was rendered stateless, efficient, and capable of handling concurrent image uploads.

API Endpoint:

- `POST /predict`: Accepts image files via multipart form-data.
- Returns a JSON object with class labels, confidence scores, and bounding boxes.

Frontend Interaction:

- React interface allows image upload.

- Axios is used to send the request to the backend.
- Bounding boxes and predictions are rendered on the uploaded image.

3.10 Deployment

- **Frontend:** Deployed to Vercel with automatic GitHub integration and HTTPS.
- **Backend:** Containerized using Docker and deployed to Render.
- **CI/CD:** GitHub Actions automate builds and deployments on every push.

This cloud-based deployment enables real-time usage from both mobile and desktop browsers.

3.11 Limitations and Future Work

- **Dataset bias:** Heavily dependent on PlantVillage; may miss rare variants seen in local farms.
- **Internet connectivity:** The model is currently dependent on live cloud access.
- **Limited disease spectrum:** Currently handles only three categories.

Future work will involve:

- Training with a broader, localized dataset from Ugandan farms.
- Offline inference deployment on Android devices using TensorFlow Lite.
- Integrating farmer advice and agrochemical recommendations.

Chapter 4

System Study, Analysis and Design

4.1 Overview of the System

The system is planned to enable users to upload pictures of potato leaves and get real-time feedback on early blight, late blight, or healthy status. It includes a React-based web interface, a FastAPI backend, and a YOLOv5 model for prediction. The design is made for responsiveness, scalability, and ease of deployment, which is highly required for remote agricultural areas.

4.2 System Architecture

System architecture has three large layers:

- **Frontend:** There is a client interface in React.js where the users are able to input potato leaf images and view the prediction outputs.
- **Backend API:** With the existence of the FastAPI process, the layer receives the images from the frontend and passes them to the inference model. It also returns the results in the form of bounding boxes and confidence levels.
- **Model Inference Engine:** YOLOv5 processes the input image, identifies disease areas, and classifies them. YOLOv5 was optimized for performance and speed. Modularized architecture such as this avoids updates to frontend, backend, or model affecting the whole system.

This modular approach allows for independent updates to the frontend, backend, or model without affecting the rest of the system.

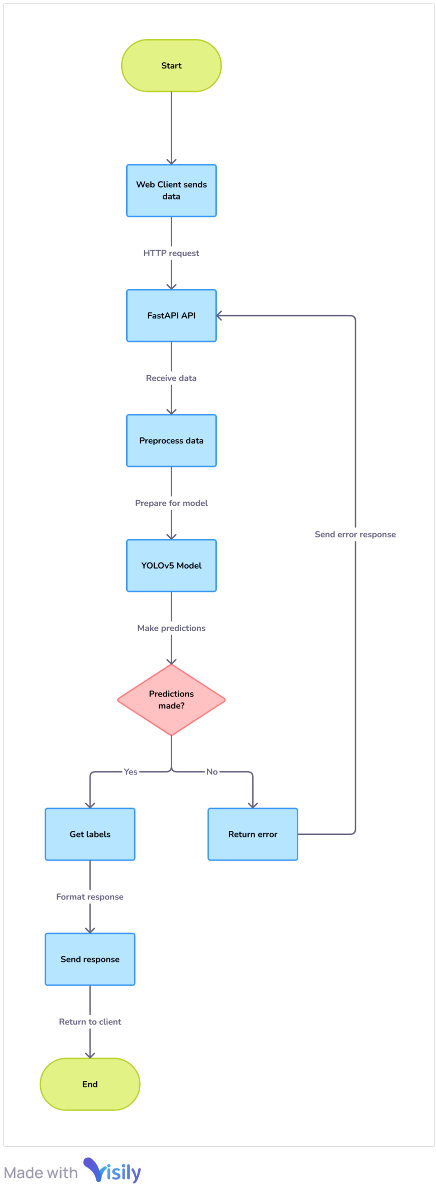

Figure 4.1: System Architecture

4.3 Use Case Diagram

Actors and Use Cases:

- **User:** Uploads potato leaf images, receives diagnostic feedback.
- **Admin/Developer:** Model retraining, system update, and deployment are their duties.

Principal(Main) Use Case: A user posts an image via the web interface, which is transferred over to the backend for classification. The backend runs YOLOv5 on the image and responds back with the output to the frontend to be displayed.

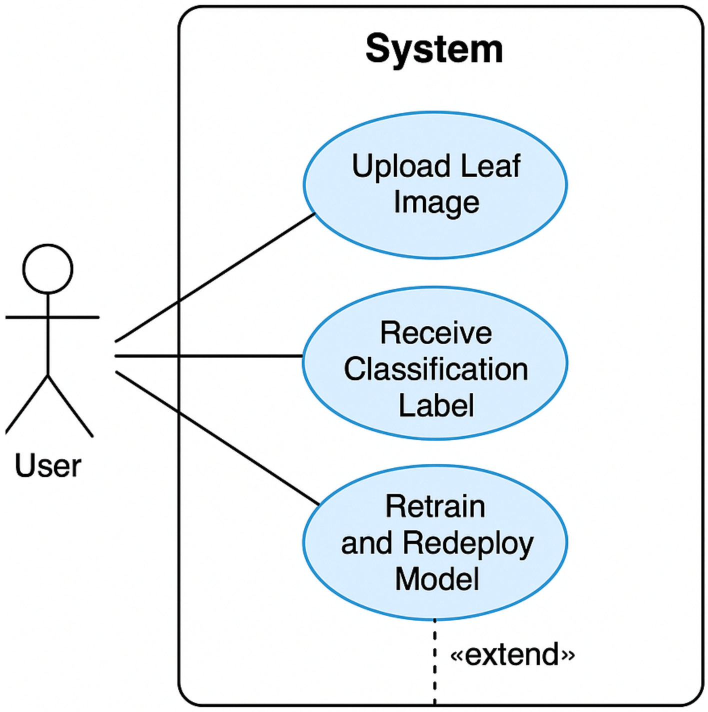

Figure 4.2: Use Case Diagram

4.4 UI Mockups

The interface was mobile-supporting and easy to use. It contains the following features:

- Image upload panel
- Preview of selected leaf image
- Show Prediction Outcome
- Option for uploading a new image

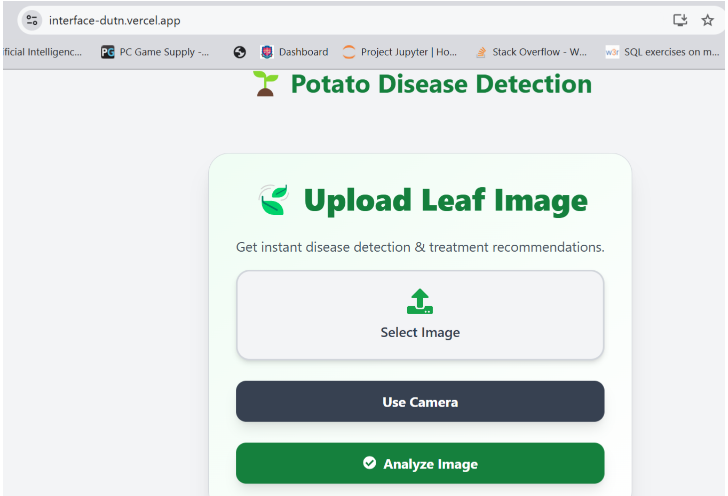

Figure 4.3: Interface Mockup of Image Upload Page

4.5 Example Prediction Output

Interpretation: The following picture shows an uploaded potato leaf being detected as ”Late Blight” with a confidence of 87%. Certain recommendations on actions to be taken are also being shown. This design for the system emphasizes usability, reliability, and deployment modularity to offer effective operation within actual agricultural settings.

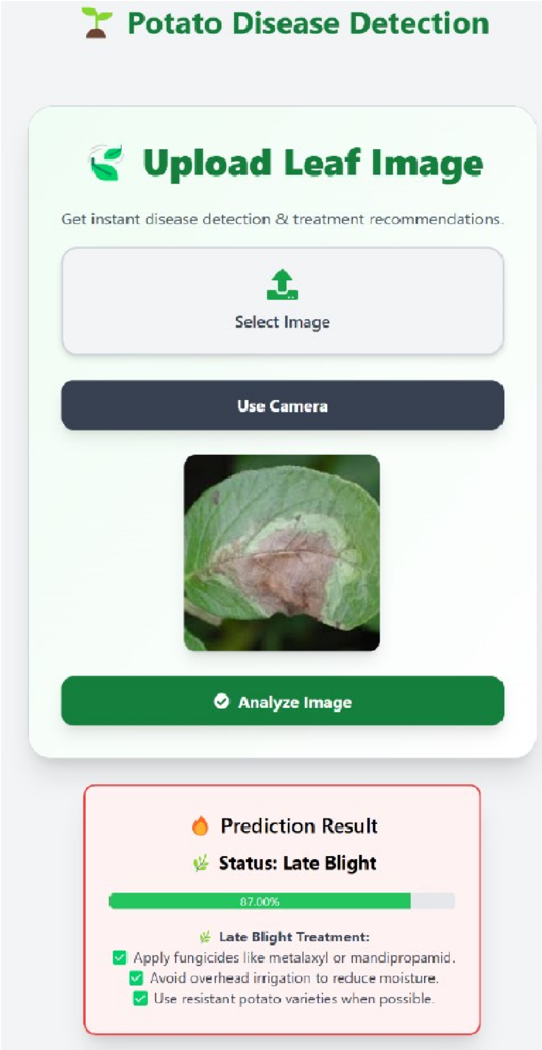

Figure 4.4: Example Output Showing Predicted Class and Confidence Level

4.6 Deployment Environment

The system has been tested on a browser environment on laptop and mobile devices. Image inputs have been captured under varying lighting to test robustness. Frontend was served on Vercel, and backend (YOLOv5 model and FastAPI) was served on Render. With this cloud configuration, real-time prediction was possible using regular internet connectivity.

Chapter 5

Results, Discussion, and Evaluation

5.1 Introduction

In this chapter, the principal results of the project are emphasized, the principal constraints faced during the development and testing process are underlined, and strategic recommendations to enhance the system's functionality, scalability, and impact in future versions are offered.

5.2 Limitations

Despite the strength of the system's performance in laboratory environments, certain limitations were observed:

- **Limited dataset diversity:** The majority of training images were drawn from the PlantVillage dataset, which may not cover the entire range of potato leaf varieties in diverse Ugandan agricultural conditions.
- **Environmental inconsistencies:** The model can fail in difficult environment conditions such as low light, occluded vegetation, or cluttered backgrounds, which were not well represented in the dataset.
- **Dependence on stable internet:** Since model inference and backend are cloud-based, there needs to be stable internet connectivity—something rural or low-connectivity areas might not offer.
- **Limited disease coverage:** The system still only supports classifying three diseases: early blight, late blight, and healthy. Other common potato diseases are still not covered.

- **Lack of persistent data storage:** No integrated database to store user history of diagnosis, which does not make it convenient to return customized recommendations or monitor long-term disease development.
- **Latency during prediction:** System has some delay time between uploading the image and response, particularly on lower networks and under backend stress.

5.3 Achievements vs Objectives

- **Objective 1:** Reviewed related tools and approaches – *Achieved*
- **Objective 2:** Studied agricultural diagnostic systems – *Achieved*
- **Objective 3:** Designed architecture combining YOLOv5, FastAPI, and React – *Achieved*
- **Objective 4:** Implemented a working prototype – *Achieved*
- **Objective 5:** Evaluated performance on test images – *Achieved*

5.4 Recommendations

To make the system more applicable in real-life situations and ease of use to the users, the following improvements are recommended to be implemented:

- **Collect localized data:** Create a Ugandan-based dataset by collaboration between Ugandan farmers and agro-institutions to produce real-field variation.
- **Develop offline access:** Develop an Android app on TensorFlow Lite or ONNX Runtime to enable prediction without internet connectivity..
- **Expand disease classification:** Include more disease categories within the model (e.g., bacterial wilt, viral) to increase the value of diagnosis.
- **Provide actionable feedback:** Incorporate professional-approved disease management suggestions and treatment options into the app directly.
- **Enable user feedback loop:** Allow users to validate or correct predictions in order to allow continuous retraining of the model and increasing accuracy over time.

- **Integrate a backend database:** Include a secure and scalable database (e.g., PostgreSQL or MongoDB) to store previous diagnoses, timestamps, image metadata, and disease confidence levels. This can help users and administrators view history, observe trends, and offer more advanced decision support.

- **Improve prediction latency:** Lower backend response latency by using faster inference engines (e.g., ONNX), batching requests, or executing the model closer to the user with edge computing platforms.

5.5 Conclusion

This project was successful in developing and deploying an end-to-end potato leaf disease detection system using YOLOv5 for object detection, FastAPI for back-end APIs, and React for front-end UI. The model worked exceptionally well to detect early blight, late blight, and healthy leaves and was made available via a straightforward web interface.

Using artificial intelligence and contemporary web technology, the system will facilitate the realization of precision agriculture in Uganda and complement the United Nations' Sustainable Development Goals, i.e., food security and digital inclusiveness. The designed proposed project has a robust framework for future potential improvement capability like model enhancement, off-line assistance, database integration, and regionally scalable implementations.

Chapter A

Project Links

Source Code Repository

The complete source code and documentation for the potato disease detection system can be accessed via GitHub:

- GitHub Repository: https://github.com/munji74/interface

Live Web Application

The deployed version of the system is available online and can be accessed through modern web browsers:

- Web App Link: https://interface-dutn-munjwok-james-alalas-projects.vercel.app/

Chapter B

Glossary of Terms

CNN Convolutional Neural Network – a type of deep learning model used for image analysis.

YOLOv5 ”You Only Look Once” Version 5 – a real-time object detection framework.

mAP Mean Average Precision – a metric used to evaluate object detection models.

IoU Intersection over Union – measures the overlap between predicted and actual bounding boxes.

Precision The proportion of true positive predictions among all positive predictions.

Recall The proportion of actual positives that are correctly predicted.

DFL Loss Distribution Focal Loss – improves bounding box regression accuracy.

FastAPI A modern web framework for building APIs with Python.

Vercel/Render Cloud platforms used for frontend and backend deployment, respectively.

Chapter C

YOLOv5 Configuration Summary

Table C.1: YOLOv5 Training Configuration

Parameter	**Value**
Model Architecture	YOLOv5s (small)
Image Size	416 x 416 pixels
Epochs	50
Batch Size	16
Learning Rate	0.001 (cosine schedule)
Optimizer	SGD
Loss Functions	Box, Classification, DFL
Pretrained Weights	COCO dataset
Augmentation Techniques	Flip, Rotation, Brightness, Noise
Annotation Format	YOLO (bounding boxes, class index)
Training Platform	Alienware desktop with NVIDIA GPU

Chapter D

React Component: FileUpload.js

Below is the complete source code for the React component used in the frontend of the web application.:

```
import React, { useState } from 'react';
import axios from 'axios';
import { FaUpload, FaCheckCircle, FaTimesCircle, FaSpinner } from 'react-icons/fa';
import Webcam from 'react-webcam'; // Import Webcam library

const FileUpload = ({ setPrediction }) => {
  const [file, setFile] = useState(null);
  const [preview, setPreview] = useState(null);
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState(null);
  const [webcamEnabled, setWebcamEnabled] = useState(false); // State for enabling webcam
  const [capturedImage, setCapturedImage] = useState(null); // State for captured image

  const webcamRef = React.useRef(null);

  const handleFileChange = (e) => {
    const selectedFile = e.target.files[0];
    if (selectedFile) {
      setFile(selectedFile);
      setPrediction(null);
      setError(null);
      setPreview(URL.createObjectURL(selectedFile));
    }
  };

  const handleCapture = () => {
```

```
    const imageSrc = webcamRef.current.getScreenshot();
    setCapturedImage(imageSrc);
    setPreview(imageSrc); // Set the captured image as the preview
    setWebcamEnabled(false); // Disable webcam after capturing
  };

  const handleUpload = async () => {
    const fileToUpload = file || capturedImage;

    if (!fileToUpload) {
      setError('Please select or capture an image first');
      return;
    }

    setLoading(true);
    setError(null);

    const formData = new FormData();

    if (file) {
      // If it's a regular file, append as-is
      formData.append("file", file);
    } else if (capturedImage) {
      // Convert base64 to Blob and append
      const blob = await fetch(capturedImage).then(res => res.blob());
      formData.append("file", blob, "captured_image.jpg"); // Ensure the filename is valid
    }

    try {
      const response = await axios.post(
        'https://interface-c00d.onrender.com/predict/', // Ensure the endpoint is correct
        formData,
        { headers: { 'Content-Type': 'multipart/form-data' } }
      );

      if (response.data.predictions && response.data.predictions.length > 0) {
        const prediction = response.data.predictions[0];
        let recommendation = 'No recommendation available';
        let borderColor = 'border-gray-300';
        let bgColor = 'bg-gray-100';
        let icon = ' ';

        switch (prediction.class) {
          case 'Early Blight':
```

```
            recommendation = ` <strong>Early Blight Treatment:</strong><br /> Use fungicides containing chlorothalonil, mancozeb, or copper-based compounds.<br /> Rotate crops to prevent soil contamination.<br /> Remove and destroy infected plants to reduce spread.`;
            borderColor = 'border-orange-500';
            bgColor = 'bg-orange-50';
            icon = ' ';
            break;
          case 'Late␣Blight':
            recommendation = ` <strong>Late Blight Treatment:</strong><br /> Apply fungicides like metalaxyl or mandipropamid.<br /> Avoid overhead irrigation to reduce moisture.<br /> Use resistant potato varieties when possible.`;
            borderColor = 'border-red-500';
            bgColor = 'bg-red-50';
            icon = ' ';
            break;
          case 'Healthy':
            recommendation = ` <strong>Your potato plant appears healthy!</strong><br/>Continue monitoring and maintaining good agricultural practices.`;
            borderColor = 'border-green-500';
            bgColor = 'bg-green-50';
            icon = ' ';
            break;
          default:
            recommendation = ` <strong>Unable to determine disease.</strong> Please upload a clear image of a potato leaf.`;
        }

        setPrediction({
          class: prediction.class,
          confidence: (prediction.confidence * 100).toFixed(2),
          recommendation,
          borderColor,
          bgColor,
          icon
        });
      } else {
        setError('No␣predictions␣returned.␣Please␣upload␣a␣valid␣potato␣leaf␣image.');
      }
    } catch (err) {
      console.error('Error␣uploading␣file:', err);
      setError('Error␣during␣file␣upload.␣Please␣try␣again.');
    } finally {
```

```
      setLoading(false);
    }
  };

  return (
    <div className="p-8 bg-gradient-to-br from-green-50 to-white shadow-xl rounded-3xl max-w-lg mx-auto mt-12 border border-gray-300">
      <h1 className="text-4xl font-extrabold text-center mb-6 text-green-700" > Upload Leaf Image</h1>
      <p className="text-gray-600 text-center mb-4">Get instant disease detection & treatment recommendations.</p>

      {/* File Upload Button */}
      <label className="w-full cursor-pointer bg-gray-100 p-5 rounded-2xl border-2 border-gray-300 hover:bg-green-100 hover:border-green-500 transition-shadow shadow-md flex flex-col items-center">
        <FaUpload className="text-green-600 text-3xl mb-2 transition-transform transform hover:scale-110" />
        <span className="text-gray-700 font-semibold">Select Image</span>
        <input type="file" accept="image/*" onChange={handleFileChange} className="hidden" />
      </label>

      {/* Camera Button */}
      <button
        onClick={() => setWebcamEnabled(true)}
        className="mt-6 w-full bg-gray-700 hover:bg-gray-800 text-white font-bold py-3 rounded-xl transition duration-200 shadow-lg flex items-center justify-center gap-2"
      >
        Use Camera
      </button>

      {/* Webcam */}
      {webcamEnabled && (
        <div className="mt-6 flex justify-center">
          <Webcam
            audio={false}
            ref={webcamRef}
            screenshotFormat="image/jpeg"
            width="100%"
            videoConstraints={{
              facingMode: 'environment', // Use back camera
            }}
          />
          <button
            onClick={handleCapture}
```

```
            className="mt-4 bg-green-700 hover:bg-green-800 text-white font-bold py-3 rounded-xl transition duration-200"
          >
            Capture Image
          </button>
        </div>
      )}

      {/* Preview Image */}
      {preview && (
        <div className="mt-6 flex justify-center">
          <img
            src={preview}
            alt="Selected Leaf"
            className="w-48 h-48 object-cover rounded-xl shadow-lg border border-gray-300 transform transition-all ease-in-out duration-200 hover:scale-105"
          />
        </div>
      )}

      {/* Analyze Button */}
      <button
        onClick={handleUpload}
        className="mt-6 w-full bg-green-700 hover:bg-green-800 text-white font-bold py-3 rounded-xl transition duration-200 shadow-lg flex items-center justify-center gap-2"
        disabled={loading}
      >
        {loading ? <FaSpinner className="animate-spin" /> : <><FaCheckCircle /> Analyze Image</>}
      </button>

      {/* Error Message */}
      {error && (
        <p className="mt-4 text-red-500 text-center font-bold flex items-center justify-center gap-2">
          <FaTimesCircle /> {error}
        </p>
      )}
    </div>
  );
};

export default FileUpload;
```

Listing D.1: React Component for Image Upload and Webcam Capture

This route handles image input, performs inference using YOLOv5, and returns the class label, confidence score, and bounding box data as JSON.

Bibliography

[1] Food and Agriculture Organization of the United Nations, "Uganda at a glance," 2020, accessed: 2025-04-11. [Online]. Available: https://www.fao.org/uganda/fao-in-uganda/uganda-at-a-glance/en/

[2] CGIAR, "Ai for diagnosing potato diseases will benefit millions of rural farmers," 2021, accessed: 2025-04-11. [Online]. Available: https://www.cgiar.org/news-events/news/ai-for-diagnosing-potato-diseases-will-benefit-millions-of-rural-farmers/

[3] G. Jocher *et al.*, "YOLOv5 by ultralytics," https://github.com/ultralytics/yolov5, 2020, accessed: 2025-04-10.

[4] PlantVillage, "Nuru: Ai-powered crop disease detection," 2023, accessed: 2025-04-11. [Online]. Available: https://plantvillage.psu.edu/nuru

[5] United Nations, "Sustainable development goal 2: Zero hunger," https://sdgs.un.org/goals/goal2, 2023, accessed: 2025-04-10.

[6] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016.

[7] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.

[8] G. AI, "Ai for social good: Agriculture," https://ai.google/social-good/agriculture/, 2023, accessed: 2025-04-10.

[9] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv preprint arXiv:1511.08060*, 2015.

[10] K. Nji and J. Oduol, "Ai-based early maize disease detection in kenya using mobile phones," *Agricultural Information Worldwide*, vol. 12, no. 1, pp. 10–16, 2020.

[11] PlantVillage and IITA, "Ai mobile app for cassava disease detection," https://plantvillage.psu.edu, 2019, accessed: 2025-04-10.

[12] C. Okeke, F. Akinyi, and J. Mbabazi, "Ai adoption in african agriculture: A review of challenges and opportunities," *African Journal of Agricultural Research*, vol. 18, no. 5, pp. 150–160, 2022.

[13] J. Habimana and S. Tadesse, "Deep learning for potato blight detection in east africa," *Journal of Precision Agriculture*, vol. 9, no. 2, pp. 45–52, 2021.

[14] M. A. Lab, "Mobile-based banana disease detection," https://air.ug, 2018, accessed: 2025-04-10.

[15] E. Mugisha and R. Katende, "Digital farming tools for smallholder farmers in uganda," *Uganda Journal of Agricultural Innovation*, vol. 3, no. 1, pp. 22–30, 2021.

[16] U. I. R. Institute, "Ai in local potato farming: A uiri case study," https://www.uiri.go.ug, 2023, accessed: 2025-04-10.

[17] Uganda Bureau of Statistics, "Uganda national household survey: Nutrition and food security report," 2022, accessed: 2025-04-11. [Online]. Available: https://www.ubos.org

[18] FAO, "Nutrition country profile: Uganda," 2019, accessed: 2025-04-11. [Online]. Available: https://www.fao.org/nutrition/en/

[19] National Agricultural Research Organisation, "Irish potato research strategy for uganda," 2021, accessed: 2025-04-11. [Online]. Available: https://www.naro.go.ug/

[20] CGIAR, "Potato production and impact in uganda," 2020, accessed: 2025-04-11. [Online]. Available: https://www.cgiar.org/research/program-platform/potato/uganda

[21] D. Hansen and R. Lehmensiek, "Early blight of potato: Epidemiology, management and resistance breeding," *Crop Protection*, vol. 144, 2021.

[22] S. Nadeem, Y. Shahzad, and A. Javaid, "Late blight of potato: Disease management strategies and resistance breeding," *Plant Pathology Journal*, vol. 36, no. 3, pp. 193–206, 2020.

[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[24] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.