DESIGN OF A DEEP LEARNING BASED CROP DISEASE DETECTION IN SMALL HOLDER FARMING SYSTEMS

BY

ORUCHE TOCHUKWU WISDOM

GOU/U22/CSC/875

DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY, GODFREY OKOYE UNIVERSITY, ENUGU STATE, IN PARTIAL FULFILLMENT OF THE AWARD OF BACHELOR OF SCIENCE (B.SC), DEGREE IN COMPUTER SCIENCE.

SUPERVISOR: ENGR. CAMILUS NJOKU

JUNE, 2026

APPROVAL PAGE

This research, titled "Design of a deep learning-based crop disease detection in small holder farming systems", has been assessed and approved by the Department of Computer Science and Mathematics Committees in the Faculty of Computing and Information Technology, Godfrey Okoye University, Enugu, Nigeria.

Mr. Camilus Njoku	....	....
Supervisor	**Signature**	**Date**
....	....	....
External Examiner	**Signature**	**Date**
Dr. Frank Okebanama	....	....
HOD, Department of Computer Science	**Signature**	**Date**
Prof. I. A. Ajah	....	....
Dean, Faculty of Computing And Information Technology.	**Signature**	**Date**

CERTIFICATION

I certify that I completed the research included therein and that it was primarily based on my observations and investigation. Consequently, I will fully accept the University's decision if I am found to have cheated on the assignment.

ORUCHE TOCHUKWU WISDOM

GOU/U22/CSC/875

DEDICATION

I dedicate this work to God and my beloved parents, Mr and Mrs Chinweuba Oruche, whose unwavering love, sacrifices, encouragement and steadfast belief in me have been the foundation of my academic journey. Your support, guidance, and countless acts of kindness have inspired me to persevere through challenges and remain committed to achieving my goals. This accomplishment is a reflection of your dedication and the values you have instilled in me.

ACKNOWLEDGEMENTS

I give thanks to Almighty God, who made all things possible and granted me the strength, wisdom, and grace to successfully carry out this project. Words alone cannot fully express my sincere and heartfelt appreciation to everyone who supported me materially, financially, morally, and academically throughout this period.

I am deeply grateful for the knowledge, guidance, and love I received during the course of this work. My special appreciation goes to my project supervisor, Mr Camilus Njoku, for his guidance and support.

I also appreciate the Head of Department, Dr Frank Okebanama, alongside the lecturers and staff of the Department of Computing and Information Technology, for their contributions and encouragement. May God bless you all abundantly.

Abstract

Crop diseases continue to pose a serious threat to agricultural productivity because timely diagnosis remains difficult when farmers rely solely on visual observation. The dependence on manual inspection often results in delayed identification and inaccurate disease recognition, which can reduce crop yield and increase economic losses. Although artificial intelligence has increasingly been explored for agricultural applications, there remains a need for practical systems capable of delivering reliable disease diagnosis under real-world conditions. This study introduces a multi-stage intelligent framework that automatically predicts and classifies crop diseases from digital images. The proposed framework first filters non-plant images using a plant verification model before identifying the crop species and subsequently detecting diseases specific to that crop. Model training was performed using the Crop Pest and Disease Detection (CCMT) dataset together with an additional manually compiled collection of non-plant images. Prior to training, image enhancement and augmentation procedures were implemented to improve model robustness and minimise the effects of dataset imbalance. EfficientNet-B0 was adopted as the feature-learning architecture because of its favourable balance between computational efficiency and predictive performance. Experimental evaluation demonstrated that the plant verification model achieved a classification accuracy of 96%, while the crop identification model produced an overall accuracy of 84.85%. The cassava disease classifier delivered the strongest predictive performance, whereas the maize classifier also achieved satisfactory results despite the visual similarity among several disease categories. Although the tomato disease model produced consistently high recall values, it was intentionally excluded from the deployed system following the established deployment criteria.

Table of Contents

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER ONE

INTRODUCTION

1.1 Background to the Study

Agriculture is one of the core foundations of Nigeria economy as it provides the rest of the population with employment and adds a good share to food security and livelihoods of the rural population. Cassava, maize, rice, yams, and tomatoes are some of the major crops prone to different kinds of plant diseases transmitted by fungi, bacteria, viruses, and pests (Jha and Kumar, 2025). The tropical climate in Nigeria frequently exacerbates these diseases, high humidity levels, and changes in temperature create favourable conditions of the proliferation of the pathogen. The detection of plant diseases in the initial and accurate way is thus critical to avoid mass losses in crop yield and maintain agricultural productivity sustainability (Ashurov et al., 2025).

Diseases disrupt normal structure or physiological functions within an organism (World Health Organization, 2020). This disruption is typically represented by identifiable symptoms and/or signs which are representative of overall health/fitness of a person (World Health Organization, 2020). There are many different types of organisms found in agriculture; therefore, there are many different types of diseases that can affect organisms in agricultural settings including fungal, bacterial, viral, nematode and other harmful microorganisms. For this reason, plant disease represents one of the most important threats to production agriculture since many plant diseases will either reduce yield or diminish quality and thus severely affect farmers' incomes and food security.

Manual visual observations by the farmers or agricultural extension officers continue to be the main conventional methods of plant disease detection in Nigeria. Although useful in a few instances, subjectivity, slow diagnosis, and unavailability of experts in rural communities restrict its range, and vast farmlands cannot be covered (Kumar et al., 2023). Moreover, the identification of them can be very difficult due to changes in light conditions, morphology of the leaves, and the presentation of the disease (Chen et al., 2023). These constraints outline the necessity in the use of the automated and intelligent systems that will be able to deliver timely and dependable detection of the disease in the varied agricultural settings (Li et al., 2024).

Recently, advances in artificial intelligence have fundamentally changed how crop diseases are managed through their use of machine learning and deep learning technologies. Specifically, these types of intelligent systems have improved the accuracy and consistency of disease diagnosis by gaining insight from the patterns contained in large organizations of agricultural data. Furthermore, deep learning (which is a more advanced subset of machine learning) has proven to be vastly more effective than traditional methods when working with complex multi-dimensional data (e.g., images of leaves). By allowing the system to automatically identify relevant visual features, the deep learning models are able to accurately diagnose disease (Li et al., 2020).

The latest developments in the field of deep learning and, in particular, CNNs have proven to be extremely successful in image classification, including in the domain of plant disease detection (Jothilakshmi and Sharanesh, 2020). The CNNs can automatically learn the hierarchical features of an image of leaves, including edges, textures, and lesion patterns and correctly classify the healthy crops and diseased crops (Kaur and Singh, 2021). Nonetheless, the CNN models are highly reliant on the effective hyperparameter adjustments such as the learning rate, the quantity of layers, the kernel size, and the dropout rate (Pai et al., 2025). Traditional methods of optimization like grid search or manual tuning are computationally costly and cannot find the optimal configurations in a complex search space (Sharma et al., 2024).

Overall, the application of deep learning and advanced machine learning algorithms in crop disease prediction offers a promising approach to address various challenges within the realm of contemporary agriculture. The aim of such intelligent systems is to help farmers detect plant diseases early and to take timely action based on these observations to prevent major damage to crops. Early detection can also help to optimize crop productivity, reduce unnecessary pesticide application, decrease production costs and reduce negative environmental impacts. Therefore, the implementation of smart disease detection solutions is a crucial step towards sustainable agriculture, enhanced food security, and environmentally sustainable farming practices.

1.2 Statement of the Problem

Today, diagnosis of crop disease still largely relies on manual observation and farmers' experience in many farming communities. Conventionally practiced methods are widely used, but tend to be inaccurate, laborious and unable to support timely disease management, especially for smallholder farmers. This means that diagnosis is often delayed or wrong and crops often are lost and farm productivity reduced. This study has several main challenges, namely:

1. Traditional visual diagnosis techniques are often inaccurate, and the farmers' response to the disease, often in terms of treatment, is inappropriate, leading to a failure to effectively contain disease outbreaks.
2. Current methods of collecting agricultural data are often inefficient, resulting in incomplete, inconsistent and low quality data which hinder effective disease diagnosis and prediction.
3. The inability to diagnose diseases accurately leads to unnecessary and/or inappropriate pesticide use, which reduces production costs, impacts environmental sustainability and human health.

4. Manual check-up of farmland is time consuming and labour intensive, which can cause lower productivity and make it hard to conduct normal check-up for crops, especially for farmers with large land holdings.

1.3 Aim and Objectives of the Study

The aim of this research is to develop a deep learning based crop disease detection system which can help diagnose crop diseases in a smallholder farming setup.

To achieve this, the research aims to:

1. Examine the shortcomings and operational limitations associated with existing conventional methods of crop disease detection.

2. Acquire, clean, and preprocess crop disease datasets suitable for training and validating deep learning models.

3. Design and develop a deep learning image classification model capable of accurately identifying crop diseases from plant images.

4. Develop and deploy a mobile application that integrates the trained model to provide users with real-time crop disease diagnosis.

5. Assess the performance of the proposed system by evaluating its diagnostic accuracy, its ability to minimize unnecessary pesticide application, and its effectiveness in reducing the time required for routine crop monitoring.

1.4 Significance of the Study

This research study is of great importance to the agriculture industry because it addresses some of the major challenges facing both smallholder traditional farmers and large-scale farmers (commercial), thereby reducing losses associated with crop production, improving yield and increasing crop quality. Moreover, this research will directly benefit farmers' income stability and greatly assist in ensuring the global food supply.

In addition, this research study is also important for the field of education, as it will support future research and development in agricultural technology; by using deep learning algorithms to forecast and prevent the incidence of crop disease, it will lead to more research and development related to the prevention and early identification of crop disease. The results of this study also offer a good basis for the study of intelligent agriculture in the future. The proposed framework can be further developed by future researchers to include more sophisticated deep learning algorithms, increased data sets, and further environmental factors for more accurate predictions. In addition, the developed approach is adaptable for the detection of other economically important diseases, apart from tomato, cassava and maize, which will help sustain agricultural production, increase food security and enhance the sustainability of agriculture.

1.5 Scope of the Study

This research work centres on the development of a system that leverages deep learning to identify and forecast crop diseases early in smallholder farming systems. The research specifically aims to tackle the need for timely disease detection for supporting farmers' decision on crop management before the disease can cause major damage. The developed system uses plant disease recognition methods with image data and tested with selected plant species namely tomato, cassava, and maize which is used because of their agricultural values and plant disease susceptibility. The main target group are smallholder farmers, but the developed system will also benefit agricultural researchers, extension officers, education and research institutions, and crop production and precision agriculture organizations.

1.6 Limitation of the Study

In the process of this research, several challenges were faced, that could affect the overall implementation and evaluation of the proposed system. These limitations include:

i. **Data available and quality:**

The success of the developed model greatly relies on the quality, completeness and consistency of the available dataset. The accuracy of the model predictions and their generalisation ability can be influenced by the inaccuracies in the image labels, the lack of environmental information, or lack of disease samples.

ii. **The findings will be generalised.**

Selected crop species and available datasets were used to evaluate the developed model; hence, its performance may differ under other crop species, geographical regions, and environmental conditions. Further adaptation of the model may be needed before the model can be widely used if farming practices, climate, and disease characteristics differ.

iii. **The third constraint is technical and resource-related.**

The third constraint is technical and resource. It was important to have knowledge about deep learning, computer vision, and software engineering for the development of the proposed system. The training time for the models was lengthy due to limited access to high-performance hardware and computational resources, and the ability to test more complex models was also hampered.

1.7 Operational Definition of Terms

The following terms are used throughout this study in the context of this study:

Accuracy: A performance metric which calculates the ratio of correct classified instances over the number of instances predicted by a machine learning model.

Classification: The process of identifying an input image or sample to one of a pre-defined set of categories (e.g. crop type, disease category) from learned patterns.

Crop Disease: A disease or disorder of a crop, usually caused by a pathogen (fungi, bacteria, viruses or other environmental factors) that affects the health and productivity of the crop.

Data Augmentation: A technique that is used before training a model to artificially expand and diversify the dataset to make the model more general.

Deep Learning: AI subfield using multilayered neural networks to extract complex patterns and representations from massive amounts of data without human intervention.

EfficientNet-B0: CNN architecture that allows for the high classification accuracy at low computation costs, through symmetric scaling of network depth, width, and resolution.

F1-Score: A performance measure that combines two metrics into a single; precision and recall, especially for imbalanced data sets.

Feature Extraction: A method in which input images can be automatically recognized and meaningful features learned from them that allow an accurate classification by a machine learning model.

Gate Model: The first filter model used in the proposed system to differentiate plant images and non-plant images before further classification.

Image Preprocessing: A series of processes that were applied to raw images to enhance the quality of the images and prepare them for model training, including resizing, normalization, and image enhancement.

Plant Disease Classification: Identification of specific pathogens of a crop from manually extracted visual features of plant images. Manual extraction of visual features from plant images for automated identification of specific diseases affecting the crop.

Precision: A measurement of how well a model correctly classifies positive cases when used to classify all positive cases.

Prediction: A machine learning model that has been trained to make a prediction about what class or outcome to label a given input.

Recall: A performance measure which represents the fraction of positive instances correctly classified by the classification model.

Training Dataset: A set of labeled data that is fed into a machine learning model to recognize patterns and to make accurate predictions.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter delves into the research and technological progress in crop disease detection, with a focus on image processing, machine learning, and deep learning methods. This chapter highlights the increasing importance of AI on crop health monitoring with special reference to smallholder farmers with low access to agricultural professionals. It follows the development of crop disease detection from manual observation to use of automated computer vision techniques like deep learning frameworks (particularly CNNs). Besides this, it evaluates the available research and outlines its strengths and weaknesses while identifying some of the current challenges such as lack of proper datasets, generalization problems, and applications in poor regions.

2.2 The Concept of Crop Disease Prediction Systems

Crop Disease Prediction Systems have become an integral part of modern precision agriculture, making it easier for farmers to detect diseases at an early stage and make informed management decisions. In the past, diagnosing crop health has relied primarily on the farmer's observations, practical experience and simple farming tools, resulting in somewhat subjective and vague diagnosis of diseases. The first prediction approaches were mostly statistical based on the way historical weather data and crop performance data were combined to derive an estimate of the risk of disease outbreak (Nourali & Shortridge, 2024). These techniques gave helpful information but were not sufficiently predictive due to their inability to effectively account for the complex interactions between environmental, biological and management variables. The advent of AI has revolutionized the prediction of crop disease. Recent machine learning and deep learning methods are now able to analyze vast amounts of agricultural data gathered from various sources such as satellite imagery, field sensors, weather stations and digital images of leaves. These smart algorithms automatically detect concealed patterns linked to the development of the disease so that the farmers can get accurate and timely predictions. This has therefore changed the crop monitoring into a more proactive intervention rather than a reactive crop disease management thereby cutting crop losses and enhancing farm productivity. One of the significant developments is the ability of deep learning models to detect disease symptoms in their early stages. Early diagnosis is especially useful as it enables farmers to take suitable control actions before significant infections become widespread within farms. Convolutional Neural Networks (CNNs), for example, are capable of accurately recognising diseases like early blight and late blight from the images of tomato leaves with fewer manual inspections required, with high classification accuracy (Sakkarvarthi et al., 2022). Early detection is also beneficial to sustainable agriculture because it helps reduce the amount of pesticides used unnecessarily and makes production more cost-effective.

In conclusion, the adoption of AI tools for disease prediction has significantly improved disease monitoring in agriculture, offering valuable benefits to farmers, researchers, and policymakers. The innovative systems offer quicker, more precise, and scalable crop health monitoring, enhancing productivity, minimizing economic losses, and fostering sustainable agriculture (Abbas et al., 2024). With the ongoing development of AI, crop disease prediction systems will become more precise, flexible, and contribute to climate-smart agriculture in the future.

2.2.1 Categories of Crop Disease Prediction System

Crop disease prediction systems can be classified on the basis of data used in the process of disease identification and prediction in to several categories. Below are some of the key categories discussed.

❖ **Image-Based Systems:**

Image-based prediction systems use images that capture characteristics from the leaves, stems, fruit, or other parts of the plant to identify plant diseases. Typically, these systems rely on computer vision and deep learning techniques, such as CNN, for the extraction of disease-related features from digital images. For instance, using deep learning methods, the TomatoDet model can classify disease conditions like early blight and late blight based on tomato leaf images with a high level of accuracy. These systems enable fast and accurate non-invasive disease detection, which is an important requirement in precision agriculture (Sakkarvarthi et al., 2022).

❖ **Environmental Data-Based Models:**

Environmental prediction systems use data on soil moisture, rainfall, soil parameters and humidity, and temperature to predict the occurrence of a disease. Environmental conditions greatly affect the growth of pathogens and the spread of diseases, and these models assess the link between environmental factors and disease to forecast potential outbreaks. Lee and Yun (2023) presented an environmental prediction model that is an example of this approach.

❖ **Hybrid Models:**

Multi-source prediction systems combine data from several sources such as plant images, weather data and previous disease history to increase the accuracy of the prediction. These systems use both machine learning and deep learning algorithms to leverage complementary information from multiple datasets, resulting in more accurate overall predictions. Sanida et al. (2023) proposed a hybrid framework that utilizes multiple sources to achieve better disease detection performance than single-source prediction models.

❖ **Genomic Data-Based Models:**

Genomic prediction systems use the genetic information from crop varieties to estimate their susceptibility or resistance to disease. These models use genetic markers linked to resistance to diseases to gain insights that can be used in crop breeding programs and disease management. While some practical application is yet to be achieved, research is still in progress on the use of genomic data to incorporate in intelligent crop disease prediction systems (Thangaraj et al., 2022).

2.2.2 Evolution of Crop Disease Prediction System:

The next few decades have seen a tremendous change in the way the crop disease prediction is done. In the beginning of agricultural practice, the diagnosis of diseases was virtually solely dependent on manual inspection in the field, where the farmers would look for observable symptoms in the crop. This was a traditional method which was simple but labour intensive, time consuming and sometimes led to results being inconsistent as a result of human error and delayed diagnosis (Gent et al., 2013). As agricultural research continued, the use of forecasting models, using past climate data and environmental factors, were adopted to help predict disease outbreaks before symptoms become severe. These early forecasting systems were able to enhance the understanding of crop-pathogen-environmental interactions, which will enable the development of scientific disease management strategies (Jafar et al., 2024; Basavant et al., 2024).

The emergence of machine learning and deep learning presented a significant advancement in forecasting crop diseases. These models utilize intelligent analysis of the large and heterogeneous datasets that consist of crop history, meteorological data, sensor data and images of the crop to identify the disease characteristics automatically with precise correctness (Wang & Liu, 2024). Among others, CNN has gained wider application in the context of the image classification due to its capability to learn astonishing visual features directly from the images. CNN-based models have been successful in the classification of diseases that affect tomato, potato and maize crop species including gray leaf spot, brown rot, late blight, potato blackleg, tar spot and southern rust disease (Thangaraj et al. 2021).

Recently, the advances in technologies made it possible to further enhance the operation of intelligent disease monitoring systems. Different frameworks, such as TomatoDet is capable to produce feature extraction and learn about different scales for better disease recognition experience. Besides, application of IoT devices, drones and remote sensing technologies make it possible to perform continuous monitoring and get timely data (Delfani et al., 2024). As a result, farmers can now detect the outbreaks of diseases much sooner than before, take control of the situation and control the spread of diseases as well as their negative economic implications.

2.2.3 The Future of Crop Disease Prediction Systems in the Field of Agriculture

The development of crop disease prediction systems have come a long way from field observation to data driven prediction technologies. Traditional approaches for detecting crop diseases were usually based on visual inspection and the farmer's personal experience, which could be unreliable and subject to the skill of the farmer. With machine learning and deep learning, disease management has been revolutionized, and faster, more accurate and proactive identification of the disease is possible. For instance, using a model like Convolutional Neural Networks (CNNs), images of crops are now able to detect diseases like grey leaf spot, brown rot in tomato, potato late blight, corn tar spot, and southern rust. Infection detection, losses and yield quality enhancement are enabled by these systems in the early stage of infection. The advent of Internet of Things (IoT) technologies has further enhanced prediction systems by providing the capability of real-time monitoring of crops and environmental conditions. Modern systems are capable of handling vast amounts of field data more efficiently and making more accurate predictions, thanks to developments in artificial intelligence. Hybrid models that are a combination of various machine learning techniques have also enhanced the accuracy of the classification of the diseases and timely intervention. In the future, new technologies like edge computing and blockchain will enhance agricultural disease forecasting by enabling quicker data handling, better security, and storage solutions. In addition, future AI models could be more adaptive, having fewer requirements for training data and changing to adapt to new farm conditions. In addition, prediction models using genomic data can help increase understanding of the resistance of disease and crop health. But, this will require the collaboration of technology developers, agricultural researchers, and farmers to ensure that such tools become useful, cost-effective, and available to them. Crop disease prediction systems can be an aid to more sustainable, resilient and productive agriculture if properly implemented.

2.3 Review of Related Works

There are a few studies that have investigated crop disease prediction/detection through machine learning, deep learning, image processing, and mobile diagnostic system. Below are some relevant works that are reviewed.

Prottasha et al. (2021) have created a lightweight CNN-based rice disease detection system in Bangladesh. These researchers gathered and enhanced images for 12 categories of rice diseases that were used to train the CNN model. However, when the model was tested on the real data, the mean accuracy was 95.4%. The study did have some inherent limitations due to the limited testing data.

Shoaib et al. (2022) presented a deep learning system to detect and classify tomato plant diseases using leaf images. The study involved over 18,000 segmented and non-segmented tomato leaf images and three architectures: InceptionNet, U-Net, and Modified U-Net. The Modified U-Net achieved a segmentation accuracy of 98.66%, InceptionNet had a binary segmentation accuracy of 99.95% and six-class segmentation accuracy of 99.12%.

Khan et al. (2024) presented a hybrid deep learning and machine learning approach to tackle the classification of ten tomato leaf classes comprising nine diseased classes and one healthy class. Feature extraction was done automatically with a customized CNN and the evaluation was carried out with the PlantVillage dataset. The highest accuracy and F1 score obtained by best CNN-stacked model was 98.27% and 98.53% respectively. The model was successful in identifying noisy images and recognizing the images rapidly, but it had no simple user friendly interface for the farmers.

Liu and Wang (2020) discussed deep-learning based object detection model, specifically YOLOv3, in crop disease detection. They found that YOLOv3 can handle raw images in an end-to-end system, thereby minimizing the error rate due to manual feature extraction and pre-processing. The model could be used for multi-scale disease detection and improved type and location identification of disease in natural field conditions.

To classify and analyze the tomato plant disease in real-time, Manchanda et al. (2024) introduced a CNN-based tomato plant disease predictor. Their target was to decrease the manual inspection, increase the detection accuracy, and facilitate sustainable farming by their system. The model demonstrated potential for proactive crop disease management; however, the degree of reliability was not thoroughly validated as it has not been widely tested against a variety of data.

Kaushik et al. (2023) proposed a mobile framework for tomato crop disease analysis called TomFusioNet. It employed a system that made use of a multilayer perceptron meta-learner, a modified transfer learning and late-fusion feature extraction. It was divided into two modules – DeepRec (for initial disease detection) and DeepPred (for disease classification). It recorded

99.93% accuracy for DeepRec as well as 98.32% for DeepPred with AUC value of 99.10%. The system still needs to be tested more broadly with different environmental conditions, though.

Palma et al. (2023) created a custom CNN model for tomato plant disease detection and classification. The architecture comprised of 3 convolutional layers and 3 fully connected layers and was less computationally expensive than models like AlexNet and VGG-16. The model has been tested on 10 tomato leaf classes and the accuracy obtained is comparable to other models with lesser processing cost. Its performance, however, has not been studied well enough in real world environmental variations.

Yildiz et al. (2024) have created an automatic tomato leaf disease detection system using computer vision and deep learning. The researchers employed the Kaggle datasets and applied machine learning algorithms like KNN, Logistic Regression, Neural Networks, SVM and Random Forest and compared them using Orange software. The algorithm using custom Python code was able to get the maximum accuracy of 96% which is higher than that of the traditional machine learning models with an accuracy of 95.6%. The system was implemented with AWS Lambda and was integrated into a mobile application on Flutter, but it was primarily tested with dummy data and not actual conditions on farms.

Jha et al. (2024) suggested an ensemble machine learning model for tomato, potato, and bell pepper disease detection. This study employed transfer learning with two models, namely, MobileNet and Inception models, which have been trained on 10,403 images from healthy and diseased leaves. The ensemble system resulted in 98.95% accuracy and helped in early detection of the disease. But the system was not implemented with an easy-to-use user interface for farmers.

Tan et al. (2021) compared the performance of the machine learning models and deep learning models on the PlantVillage dataset for the classification of tomato leaf diseases. They extracted

157 texture and colour features such as LBP, GLCM, colour moments and histograms. The COLOR+GLCM combination of the machine learning models was the most effective. But the accuracy, precision, recall and F1-score of the deep learning models were better than traditional methods with ResNet34 showing 99.7% accuracy, precision, recall and F1 score.

Badiger and Mathew (2023) created a deep learning model for tomato leaf disease detection, called DbneAlexNet. They applied anisotropic filtering and U-net segmentation with optimization techniques and image augmentation. The model's accuracy was 92.4%, true positive rate was 91.9%, and true negative rate was 92.2%. While the model exhibited good technical performance, the study did not take into account farmer acceptance and usability.

Ullah et al. (2023) introduced a lightweight deep learning model called DTomatoDNet that is capable of identifying tomato leaf diseases without requiring a large amount of computing resources. The model has 19 learnable layers and trained with 10,000 tomato leaf images with nine disease and one healthy class. It had a recognition rate of 99.34% and its use in mobile diagnosis was feasible. But it was not introduced in an accessible platform for farmers.

Verma et al. (2019) created an early stage tomato disease detection model based on CNN. The study was conducted on a total of 3,000 tomato leaf images from nine disease classes and healthy tomato leaves. It applied image preprocessing, segmentation and colour, texture and edge information feature extraction for the model. It was able to reach an accuracy of 98.49%, however, due to its location on Google Colab, it was not easy to access by non-technical farmers.

Tarek et al. (2022) tested some of the popular deep learning models, such as ResNet50, InceptionV3, AlexNet and MobileNet variants for detecting tomato leaf disease. MobileNetV3 outperformed the others, with MobileNetV3 Small scoring 98.99% accuracy, and MobileNetV3 Large scoring 99.81%. The feasibility of the models in crop disease detection for real-time application in IoT systems was confirmed through deployment tests using a workstation, and a Raspberry Pi 4.

Alzahrani and Alsaade (2023) used transfer learning with computer vision and deep learning to detect tomato leaf diseases at their initial stages. They tested DenseNet169, ResNet50V2 and Vision Transformer models. DenseNet121 outperformed the others with an accuracy of 99.88% and 99.00% during training and testing, respectively. The study demonstrated the feasibility of deep learning models in agricultural disease detection.

Agarwal et al. (2020) suggested a lightweight CNN based model for tomato leaf disease classification. They used the PlantVillage dataset to compare their 8-layer CNN with k-NN, decision trees, and VGG16. The model's accuracy is 98.4% and subsequently gets up to 98.7% on other data sets, which shows a good degree of generalization. But the system has not been tested using real data from the field.

Jasim (2021) studied the application of CNN models in the general detection of plant disease. The model was trained using 18,000 free available images of plant disease from 10 plant species. The best performing CNN had an accuracy of 97%, showing that deep learning can aid in scalable and accurate plant health diagnosis.

Table 2.3 Summary of Related Works

Author(s)	**Work Done**	**Technique Used**	**Result / Research Gap**
Prottasha et al. (2021)	Proposed a computationally efficient framework for recognising twelve categories of rice	Lightweight CNN architecture combined with image augmentation techniques.	The model obtained an accuracy of 95.4%; however, its evaluation was constrained by the limited availability of diverse field

	diseases common in Bangladesh.		images, reducing its generalisation capability.
Shoaib et al. (2022)	Presented a deep learning approach for automatic identification of tomato leaf diseases.	InceptionNet, U-Net, and Modified U-Net trained on a dataset containing more than 18,000 images.	The proposed models achieved 99.95% accuracy for binary classification and 99.12% for multi-class classification, although segmentation accuracy varied across different disease categories.
Gottemukkala et al. (2024)	Designed a framework for detecting tomato diseases at an early stage using visual image characteristics.	Image processing techniques integrated with Principal Component Analysis (PCA) and Ant Colony Optimization (ACO).	The framework enhanced detection accuracy and computational efficiency but depended significantly on high-quality, high-resolution image inputs.
Khan et al. (2024)	Introduced a hybrid deep learning solution for classifying ten categories of tomato leaves.	Custom Convolutional Neural Network (CNN) combined with CNN stacking.	The proposed model recorded an accuracy of 98.27% and an F1-score of 98.53%. Nevertheless, the system did not provide a

			user-oriented interface suitable for farmers.
Liu and Wang (2020)	Investigated conventional image-processing techniques alongside deep learning methods for plant disease identification.	YOLOv3 enhanced with an image pyramid strategy.	Their findings demonstrated superior disease localisation using deep learning, whereas conventional approaches remained susceptible to inaccuracies caused by manually engineered features.
Manchanda et al. (2024)	Developed a real-time intelligent system for predicting tomato diseases to support precision agriculture.	Fine-tuned Convolutional Neural Network (CNN).	The proposed solution showed promise for sustainable farming applications; however, validation using sufficiently diverse real-world datasets was not performed.
Kaushik et al. (2023)	Proposed TomFusioNet, a mobile-oriented framework for	Transfer learning, Multi-Layer Perceptron (MLP), NSGA-II	Experimental evaluation reported 99.93% accuracy for DeepRec, 98.32% for DeepPred, and an AUC of

	automated tomato disease diagnosis.	optimisation, and HSV-based background removal.	99.10%. Despite these results, testing under practical field conditions remained limited.
Aishwarya et al. (2023)	Developed a lightweight convolutional neural network for recognising tomato diseases.	CNN architecture consisting of three convolutional layers and three fully connected layers.	The network achieved competitive classification performance while maintaining low computational complexity, although its effectiveness in operational farming environments was not demonstrated.
Yildiz et al. (2024)	Implemented an automated tomato disease detection application with mobile deployment capability.	CNN integrated with RF, KNN, LR, SVM, Neural Network, AWS cloud services, and Flutter.	The proposed Python-based system achieved an overall classification accuracy of 96%; however, its performance was not validated using real agricultural field data.

2.4 Research Gap

The research study was carried out in a particular sector of the world, namely sub-Saharan Africa. The farmers in this region are still using the same old ways of detecting disease, which involves going through the crops painstakingly to find out those that have been infected. However, the intelligent systems that are currently available have not been able to lead to any desired results since some aspects of these systems have not been validated. This has happened because the systems utilize information that was collected in different areas of the world. Thus, the geographic condition has made it difficult for the use of these systems since the climatic conditions in other areas differ greatly with those existing in the sub-Saharan areas of Africa.

Nevertheless, the proposed system puts the scientific table into proper perspective and clearly identifies the gaps that need to be filled while contributing to the body of knowledge by coming up with a model through the use of a combination of both localized datasets and datasets from publicly available data.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.0 Introduction

This section of my research paper discusses the methodology and designing of my proposed plant disease detection system which is based on Artificial Intelligence. The problem of crop diseases has been a significant issue since long back and has been affecting the agricultural yield across the world. Thus, in order to maximize crop productivity and reduce the losses suffered by farmers, it is essential to recognize diseased crops as soon as possible. For the purpose of this chapter, I will make use of object-oriented techniques of systems analysis and design whereby I will be able to illustrate how machine learning algorithms can be used for diagnosing and classifying crop diseases.

3.1 Research Methodology

This research paper embraces an experimental research design and Object-Oriented Analysis and Design (OOAD) for the software development methodology in order to design and test an EfficientNet-B0 model integrated in a web application software to detect plant diseases better in the Nigerian farming landscape. Images of plant leaves are gathered by publicly available datasets and are supplemented by locally available farm images of Nigeria to provide environmental flexibility. Preprocessing is done to enhance generalisation and minimise overfitting in the dataset, and then it is divided into training, validation and testing sets.

3.2 System Analysis

The analysis generally refers to breaking down and understanding the working of an object, in terms of its parts and the relationships among them. This part of my research study focuses on the comprehensive system analysis of the system to be developed, which involves evaluation of the existing systems, identification of various limitations, and recommendations as to how to improve.

3.2.1 Weaknesses of Existing Systems

Other existing systems do have some drawbacks, though, such as the following:

- **Susceptibility to Overfitting**: It has to be noted that most of the systems available are simple, and this makes them susceptible to overfitting in case of a high-dimensional dataset or a noisy dataset as simple models that are not good at generalizing to unseen data may end up overfitting, and thus not reliable and not accurate.
- **Lack of Adaptability to Imbalanced Data**: Plant disease datasets often have an imbalanced class distribution, and straightforward algorithms might not perform well when dealing with imbalanced data, resulting in a skewed prediction and lower sensitivity to the minority classes.
- **Vulnerability to Missing Data**: xisting systems lack robust tools to deal with missing or incomplete data, which is prevalent in real world data sets. Poor data management of missing values can lead to biases and less credible model predictions.

3.2.2 Analysis of the Proposed System

The traditional agricultural decision-making process documented so far has had several drawbacks due to the lack of adequate knowledge, and it focused mainly on the eastern part of the country Nigeria as a case study, so this system is greatly a step forward to overcome such setbacks and inadequacies. The proposed system uses multi-stage classification architecture for accurate and efficient diagnosis of crop disease, rather than the system making prediction of the disease directly on all input images, architecture of the system consists of three stages: plant detection (gate model); crop classification; and crop-specific disease diagnosis. This smart system contains a wealth of features, most of which are intended to offer farmers a complete and intelligent crop disease detection system. It highlights the detection of cassava and maize crop diseases as a case study and provides a comprehensive solution through the use of a comprehensive crop disease dataset and deep learning algorithm predictions to transform all the activities involved in taking agricultural decisions and to encourage sustainable practices in the region.

3.2.4 Advantages of the Proposed System

The proposed system has the following advantages:

i. Improved Accuracy of Disease Detection: Deep learning models are extremely helpful in accurately detecting and identifying crop diseases using the images of leaves and weather information.

ii. Readability and Appropriateness: The system also integrates local climate and geographical data specific to the Eastern part of Nigeria, enabling more contextually and regionally relevant disease predictions that offer more actionable insights for farmers.

iii. Multi-Crop Support: The system is able to detect the diseases of various staple crops, which makes it more applicable to a wider group of farmers including tomatoes, cassava, and corn.

iv. Real-Time and Early Detection: The use of predictive analytics can help detect diseases in time and take action, thus minimizing losses caused by delayed response times.

v. User-Friendly Interface: The system features a simple graphical user interface (GUI) that allows it to be used by farmers with low technical skills, making it accessible to a wide range of smallholder farmers.

vi. Data-Driven Decision Making: The system minimizes guesswork such that farmers can now make decisions backed by data, which will result in better disease management.

vii. Support for Sustainable Agriculture: The system does so by providing accurate water monitoring and forecasting, which helps optimize inputs, eliminates unnecessary treatments, and maintains soil and crop health.

viii. Bridges Technological Gaps: The system's strengths lie in tackling some of the major limitations of current disease management tools, such as the lack of climate awareness in some and the failure to grow with the local agricultural needs.

3.3 Data Collection

In this work, the CCMT dataset (the Crop Pest and Disease Detection Data) was used, where high-resolution images of tomato, maize and cassava were taken under the real-field conditions in Ghana, having different lighting, backgrounds, and natural farm environments, comprising 24,881 raw images, 102,976 augmented images and 22 classes. The disease classes were taken into account as a part of this study. In order to increase the locality of the data, a farm in Ngwo, Udi LGA, Enugu (longitude 6.4128ºN, latitude 7.4184ºE) was sampled to capture common Nigerian crop diseases, including cassava mosaic Figure 1(a), bacterial blight, maize leaf blight in Figure 1(c), and tomato leaf blight in Figure 1(d) in real-field environments with shadows, occlusions, and mixed backgrounds and 152 images of plant leaves The combined data set was divided into the training and testing sets, and the images were reshaped to 224x 224 pixels and normalised so that the EfficientNet-B0 model would have the opportunity to learn both regionally on the West African data and locally on the Nigerian samples to achieve the maximum generalisation, and a wide applicability of smallholder farms.

3.3.1 Sample Dataset

Figure 3.1: Samples of collected dataset

In Figure 3.1, the locally gathered dataset is represented. Since the dataset has high-resolution images of crops like maize, tomato, and cassava that are very commonly used in the area under study, the dataset will be employed for training purposes. Each image is annotated properly to indicate whether the leaf in that image is affected by any disease or it is healthy. In addition to the above, the dataset shall additionally have any of the following attributes from the dataset itself. For instance, it includes;

i. Crop Type: It defines which crop is being represented in the image such as maize, cassava, tomato.

ii. Disease Classification: It indicates the health condition of the crop to find out whether the leaf is infected or healthy by naming the disease, such as maize streak virus in case of maize crops.

iii. Disease Severity: It describes the severity of the disease as mild, moderate, or severe, wherever applicable.

iv. Image Source: It indicates from where the image has been taken while conducting an experiment.

3.3.2 Data Preprocessing

All images of the CCMT dataset and the 152 local images of Enugu leaves were also pre-processed in a complete preprocessing pipeline before training to achieve consistency and maximise the model performance. To begin with, every image was downsized to 224x224 in order to fit the dimensions of the input of the CNN architecture, and the aspect ratio was maintained to prevent distortion of important features of the disease. It was then followed by normalising pixel values to the range (0-1) in order to standardise input values to enable quicker convergence and stabilised training. The oversampling and weighted loss functions were used to overcome the class imbalance within the disease categories, particularly within the locally collected images, where certain types of diseases were underrepresented. This assures that the model will not be biased to any of the majority classes and also the sensitivity of the model to all the disease categories will be high.

Also, both datasets were augmented with data augmentation algorithms, random rotations, flips, zooming, brightness changes and shearing, not only increasing the effective dataset size, but also modelling the natural variability of the real field commonly seen around Nigerian farms. The 152 local images have been well incorporated into the training and validation splits, which gives the model information about the local environmental factors, such as tropical light, heterogeneous backgrounds, and manifestations of field-level diseases. This enables the EfficientNet-B0 model to generalise well to the Nigerian farms outside the Ghanaian CCMT dataset. Lastly, all the processed images were sorted into disease-only classes, but not pests and loaded into PyTorch DataLoaders to train them in batches.

3.4 System Requirements

System requirements are a collection of the properties and behavior of the proposed crop disease prediction system that are required to make it work. The requirements are classified into functional requirements and non-functional requirements.

3.4.1 Functional requirements

1. Feature Extraction and Selection: The system should automatically be able to give useful features extracted from plant images associated with crop diseases. It should also employ feature selection methods to minimize the dimensionality of the data without losing important information needed for proper disease classification.

2. Model Development: The system should enable the development and training of deep learning models and support the detection of crop diseases. It should be flexible enough to support different machine learning algorithms such as RF, LR, SVM and Convolutional Neural Network (CNN) architectures.

3. Real-Time Disease Prediction: The application should classify the diseases from uploaded images of plant, and display the classification results with the relevant diagnosis information in a user-friendly interface.

3.4.2 Non-Functional requirements

1.Performance: The system should be able to process the images and make predictions within a reasonable time frame, to ensure quick diagnosis of diseases.

2. Scalability: The application should be scalable with respect to the number of users and the size of image sets it will be required to process.

3. Reliability: The system should have a regular working schedule with minimal downtime to allow farmers, researchers and agricultural extension workers to access the system at all times.

4. Accuracy: The prediction model should be accurate in classification, with low false positive and false negative predictions to improve the prediction accuracy.

5. Robustness: The system should work well under noise or missing information or different environmental conditions and should be robust against different varieties and images.

3.3.1 The Proposed EfficientNet-B0 model

For the proposed crop disease prediction system, EfficientNet-B0 is considered as main image classification model that could provide high prediction performance with low computation complexity. The EfficientNet-B0 is an effective member of the EfficientNet family of convolutional neural networks (CNNs) that provides a good trade-off between classification accuracy and computation in scenarios such as agriculture where computing resources can be limited. It simultaneously modifies the depth, width and input image resolution of the network via its compound scaling, which enhances feature learning without much computational overhead. The framework of the overall design for this study is shown in Figure 3.2. The EfficientNet-B0 network is built with the Mobile Inverted Bottleneck Convolution (MBConv) layers and Squeeze-and-Excitation (SE) blocks. Such architectural parts facilitate the model's ability to highlight discriminative visual attributes while minimizing the number of trainable parameters. Furthermore, the depthwise separable convolutions reduce computation cost by breaking down traditional convolutions into simpler modules, and inverted residual connections enhance the propagation of features and boost network efficiency. These design features enable the model to achieve high classification accuracy with reduced computational overheads compared to traditional CNN designs. One of the unique features of EfficientNet-B0 is that the dimensions of the network are scaled in a compound manner, rather than scaling each dimension individually. The model is designed to optimise all three dimensions – depth, width and the resolution of the image – with a compound coefficient (ϕ) instead of just changing one dimension, such as depth or width, or the image resolution. This relationship can be represented as shown in Equation (1).

$$d = \alpha\psi, w = \beta\psi, r = \gamma\psi \quad (1)$$

Here, d, w and r stand for the depth, width and resolution of the image of the input image into the network, respectively, and α, β, γ are scale constants > 1. They are learned by optimizing them via grid search, allowing the model to learn an optimal set of numbers with respect to the size of the network, for a balance of computational efficiency and predictive accuracy.

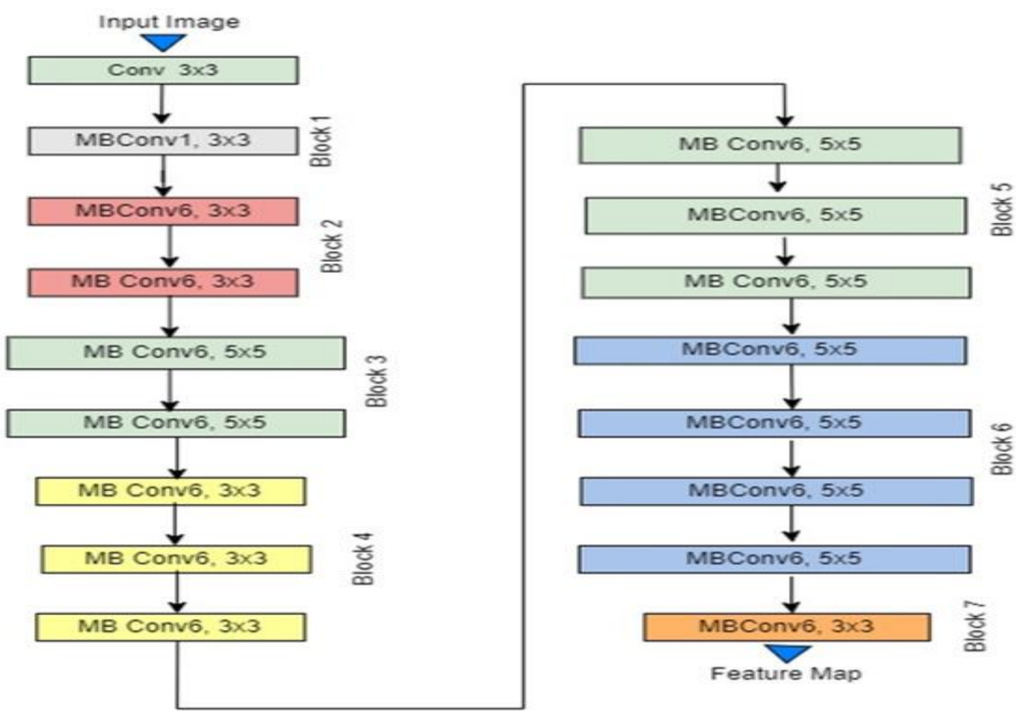


Figure 3.2: Architecture of the EfficientNet-B0 (Iqbal et al., 2024)

The Swish activation function is defined mathematically in Equation (2) as follows:

$$f\ swiss(x) = \frac{x}{1+e-\beta(x)\ (2)} \quad (2)$$

The classification head of EfficientNet-B0 acts as the last part of the architecture. This block serves the function of transforming the high level features extracted by previous convolution layers into the final result.

3.3.3 System Architecture

The developed crop disease prediction system is designed in a modular structure and designed so that it can be used in a single step by combining the acquisition, pre-processing, feature extraction, disease classification and result presentation of the system. The workflow starts with capturing/uploading plant images, then pre-processing the image, extracting features and finally applying the trained EfficientNet-B0 model for classification. The class of the disease is then predicted to the user along with its diagnosis, which helps in taking informed decision and management of the crop at the right time. The overall architecture of the proposed system is given in fig 3.3.

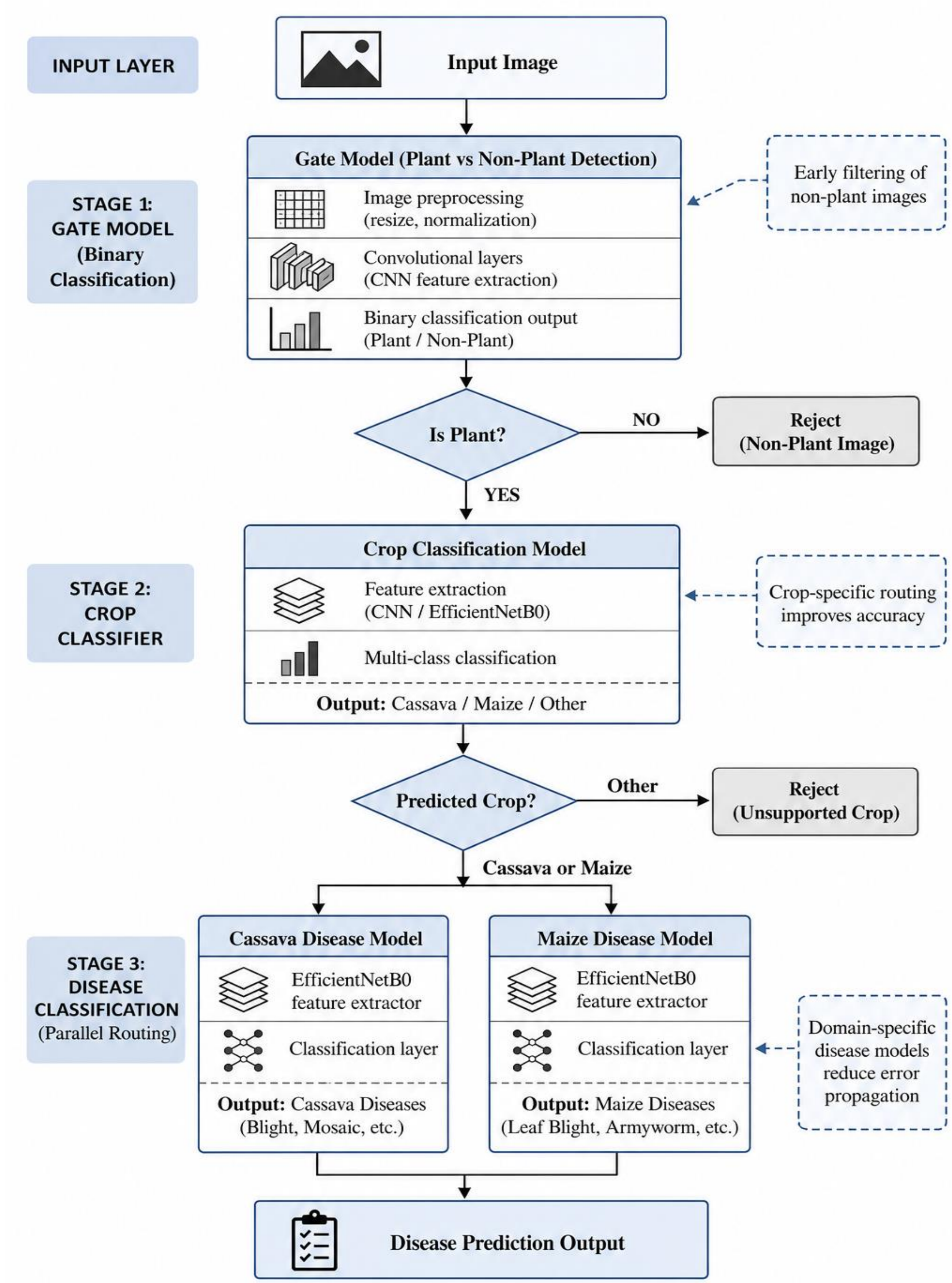


Figure 3.3 System Architecture diagram

Figure 3.3 illustrates the proposed plant disease detection system architectural diagram, where the system accepts a crop image as input, which is then passed to the input processing layer. The gate model first examines the image after the preprocessing steps, that involves image resizing and normalisation, to help improve image consistency and learning stability. Then, the CNN layers extract important visual features for the gate model to perform binary classification to check whether the image contains a plant; consequently, non-plant images are rejected immediately, while plant images continue to the next stage in order to reduce unnecessary disease classification errors.

The crop classifier receives accepted plant images for the EfficientNetB0 to extract deeper crop-related features in order for the model to perform multi-class crop classification, for example, predicting whether it is cassava, maize, or other (meaning unsupported crops will be rejected as "Other"). At this stage, cassava and maize proceed to disease routing, where crop-specific routing separates disease prediction paths, with cassava images entering the cassava disease model and maize images entering the maize disease model. Each disease model uses EfficientNetB0 classification feature layers to identify crop-specific diseases, such as Cassava diseases, including blight and mosaic, and maize diseases, including leaf blight and armyworm. And the parallel routing improves model specialisation and accuracy, meaning the domain-specific models will reduce wrong disease predictions. Finally, the predicted disease result is displayed by the system. This process demonstrates how the architecture supports accurate hierarchical decision-making, with the early filtering helping to improve the reliability and processing efficiency of the system.

3.4 Proposed Software Development Methodology

The Object-Oriented Analysis and Design (OOAD) is identified as a software development methodology that adopts the object-oriented concept. The OOAD methodology involves the application of object-oriented programming in designing and engineering implementations of the software system, since the UML diagrams present the way the interrelations of system parts are visually illustrated. Use cases, on the other hand, are widely used for defining different ways the end-users use the software system. The use of OOAD methodology is especially suitable for dealing with complex systems, which is the main reason for its use in such cases.

Unified Modelling Language

Unified Modeling Language (UML) is a standard way of modeling the structure and behaviour of software systems during analysis and design in a graphical manner. It allows system needs and implementation for software architects, developers, analysts and all other parties to be communicated in a common visual language. UML helps streamline complex system structures into well-defined diagrams, improving collaboration during the software development lifecycle. The use case diagram among the different UML diagrams is one of the most important, as it shows the interaction between the users or external entities with the system. It gives a very general overview of a system's functional needs and identifies services the application should offer. The activity diagram is a second use case diagram that supplements a use case diagram by modelling the operational flow of the system. It shows the sequence of activities, decision points and concurrent activities to execute different processes, instead of showing interactions between the actors and the system. This visualization helps developers understand the flow of tasks between different stages of the process and helps them identify potential bottlenecks, redundancies, and opportunities for optimizing the system. The other important UML modelling tool is the sequence diagram that shows the sequence of interaction of system objects in the course of their activities in a particular function or scenario. It shows the communication flow between various components that takes place over time in order to complete specific tasks. Therefore, sequence diagrams are used to help the developers understand the collaboration of the various components of the system and to ensure that the application's logic is correct. UML sequences diagrams consist of four main elements: lifelines, messages, activation bars and objects. The entities participating in the process are represented by lifelines, and the communication that is exchanged between them is represented by messages. Activation bars show the time period of activation of an object while executing a certain action on receiving a message. These components give a structural and temporal view of system interactions, allowing stakeholders to see not only what happens first in the system, but the timing of what happens in the system. Moreover, the UML sequence diagrams facilitate the understanding of the complex behavior of a system and explain it to technical and non-technical stakeholders in a visual and easy-to-understand way. They can clearly represent dynamic interaction, which facilitates collaboration among project participants, helps validate the system and aids developers in identifying design improvements and future system enhancements.

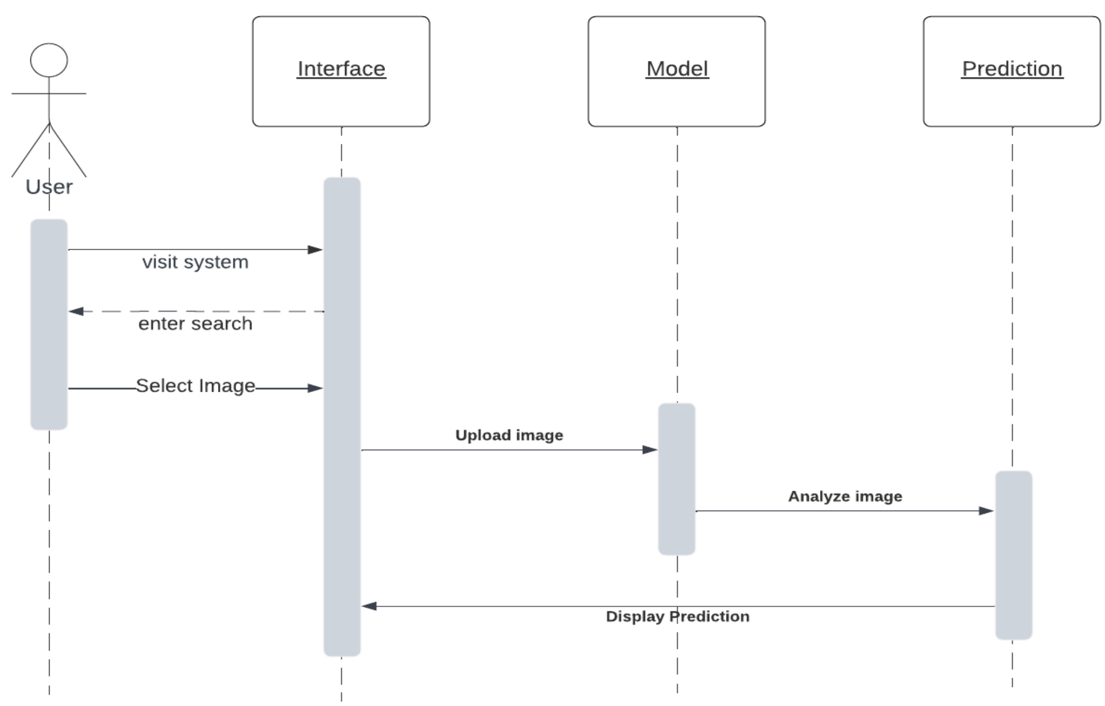


Figure 3.5 Sequence diagram of the proposed system

The sequence diagram is shown in Figure 3.5 and shows the flow of events in the proposed system. The diagram shows the sequence of interactions that start when a farmer logs into the application interface and the farmer uploads an image of an infected crop leaf. The uploaded image is then processed by the system, which identifies relevant features, makes predictions about the disease and provides diagnosis and treatment recommendations accordingly. This diagram shows a detailed view of how the various parts of the system communicate with each other and is useful for understanding the flow of operations of the proposed solution.

3.5 Use Case Diagram of Proposed System

Use case diagram is a UML modelling tool that is applied to represent the functional relationship of external actors with software system. It consists mainly of actors, use cases, associations and the system boundary. Actors are persons or other entities with which the application has to interact, Use Cases are the services the application offers to the user, Associations represent the relations between actors and the services the application needs to provide, the System Boundary indicates the scope of the application..

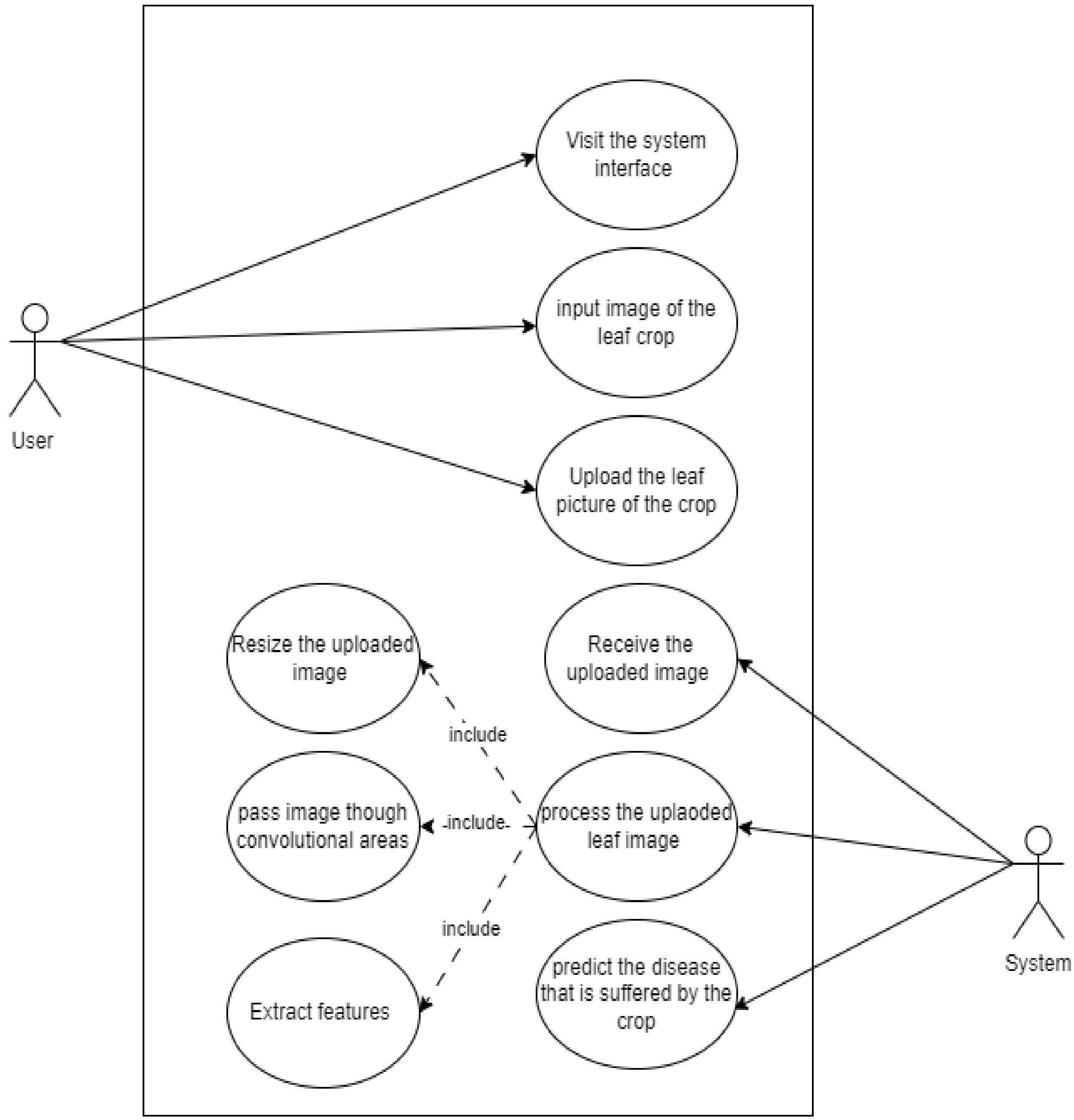


Figure 3.6: Use case diagram of the proposed system

From the user's perspective, the user interacts with the application by accessing the platform and uploads a crop leaf image as depicted in Figure 3.6. The system is then used to analyse the uploaded image, match the features with a corresponding disease and provide the user with recommendations for effective disease control and crop management.

3.6 Activity Diagram of the Proposed System

The activity diagram is another key diagram in UML that depicts the flow of activities of operations carried out in an application of software. It is a visual representation of the activities, decision points and process flows from start to finish of an operation. This modelling technique enables to have a whole picture on the operational behaviour of the system and to understand how the different processes are executed. The activity diagram shows the entire process of the proposed system, starting from when the farmer accesses the application and uploads an image of the crop leaf. The image uploaded is then analyzed by the system, after which a feature extraction process is performed, the system makes a disease prediction and presents the diagnostic results along with appropriate therapy suggestions.

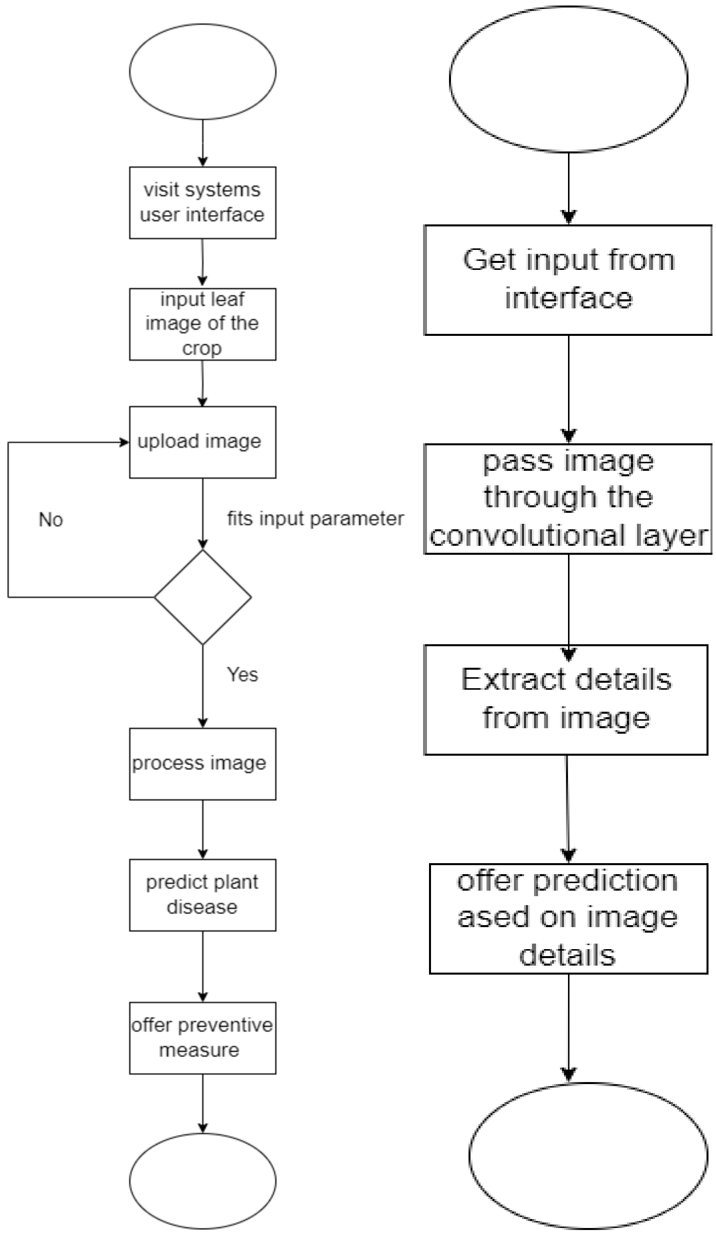


Figure 3.7: Activity diagram of the system

The activity diagram showing the operational flow of the proposed crop disease prediction system is presented in figure 3.7. The diagram shows each step of the system's execution beginning with the user's interaction, to the submission of the image, disease identification, and generation of recommendations. This representation allows a clear understanding of the sequence of the activities performed during system operation.

3.6.1 Class Diagram of the Proposed System

The Class Diagram of the proposed system is presented in this section. A UML class diagram illustrates the static structure of a software system, focusing on its classes, attributes, methods, and relationships between the elements. It offers a blueprint of system architecture, illustrating how various objects are arranged, and how they communicate with each other to facilitate the functionality of the application. The class diagram representation of the proposed Crop disease Prediction System is shown in Figure 3.8 and it shows the structure of the crop disease prediction system and their relationships in the application.

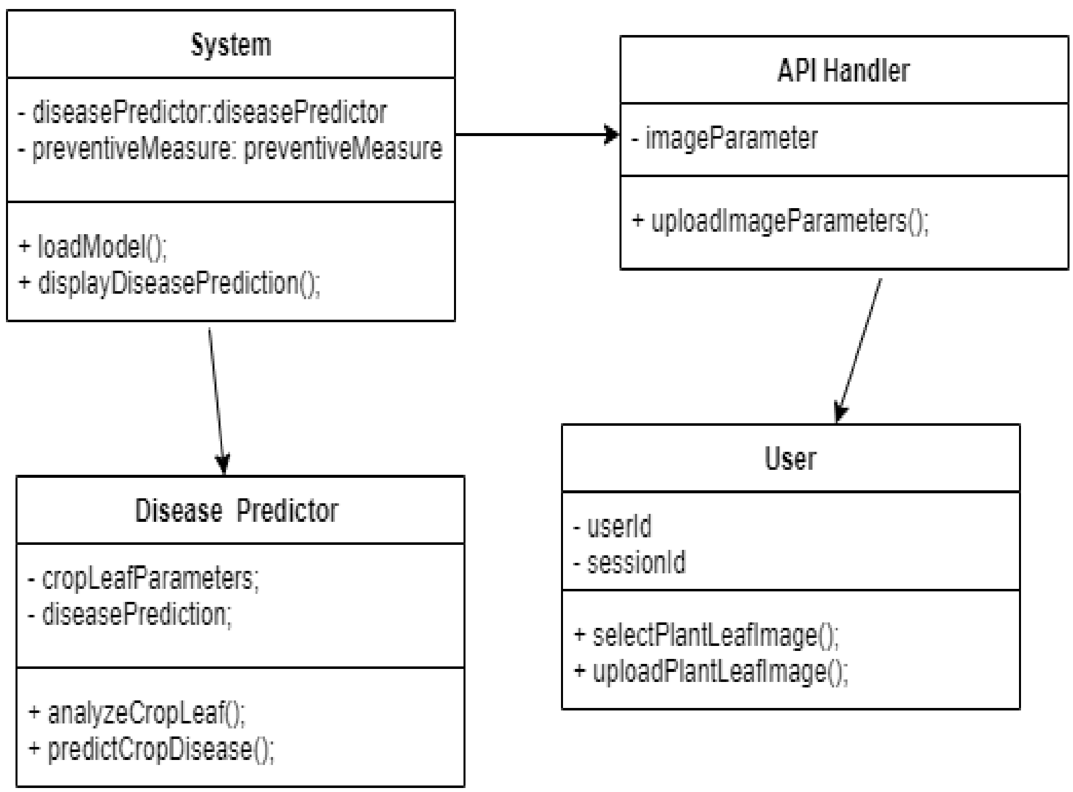


Figure 3.8: Class diagram of the proposed system

The diagram given above shows the clothing diagram of the proposed system. The diagram represents the various activities involved in the proposed system and also shows the different classes that actually exist within the system.

3.7 Database Design

The database design includes only a rough representation of the way in which database appears. The purpose of the database design is to provide a modeled view of the database of the proposed system.

Database schema for the proposed system

Table 3.1 User table for the proposed system

Attribute	Data Type	Description
Id	Int	Unique identifier and primary key for each user.
Username	Varchar	Stores the username associated with the user's account.
Email	Varchar	Stores the registered email address of the user.
Password	Varchar	Stores the encrypted password used for authentication.
Date_joined	Timestamp	Records the date and time the user account was created.

Table 3.2 Prediction table of the proposed system

Attribute	Data Type	Description
Id	Int	Primary key used to uniquely identify each prediction record.
Prediction	Varchar	Stores the disease identified by the prediction model.

User	Varchar	Records the user who generated the prediction.
Session	Timestamp	Captures the date and time when the prediction was performed.

Table 3.3 Suggestion table of the proposed system

Attribute	**Data Type**	**Description**
Id	Int	Unique identifier for each recommendation record.
Prediction	Varchar	Stores the detected crop disease associated with the recommendation.
Suggestion	Varchar	Contains the recommended treatment or management strategy for the identified disease.
Session	Timestamp	Records when the recommendation was generated by the system.

CHAPTER FOUR

SYSTEM IMPLEMENTATION

4.1 Introduction

System implementation is the stage in which the software is brought into functionality. This chapter provides the details of the system implementation along with the interface and other tools used in the system.

4.2 Development Environment and Tools

Python was chosen for its flexibility and support for deep learning model frameworks. It is an open-source ecosystem that accelerates AI model development by supporting multiple ML frameworks, enabling rapid prototyping, testing, and evaluation of AI systems. Additionally, the Python programming language has a strong community support and extensive documentation available.

4.2.1 Programming Languages and Libraries

Programming Language Justification

Python supports deep learning and scientific computing efficiently because it has a simple syntax that improves readability and faster implementation. The Image-to-Video generation system that will employ a deep learning technique will leverage Python's large ecosystem for AI, ML and data processing to build and implement an effective and efficient system. Additionally, Python has compatible GPU acceleration libraries, and it is widely used in research and industry applications.

Libraries Used

In implementing my model, I used a lot of Python libraries, including TensorFlow for the EfficientNetB0 model training, Keras for simplifying neural network model development, OpenCV for handling image preprocessing and transformations to the required format, and the NumPy library for supporting numerical computations and array operations. In addition to that, Matplotlib is used for visualisation and result plotting in accordance with the metrics considered. PIL handles image loading and basic processing, and Scikit-learn supports evaluation metrics and preprocessing of data utilised in training the EfficientNetB0 model. FastAPI was utilised for the backend API development and then a modular pipeline design for integration.

4.2.2 Development Platform and Hardware Requirements

The development platform or integrated development environment employed for this study includes the Visual Studio Code (VSCode), which was used for coding and debugging, the Google Colab, which was extensively used for cloud-based GPU training in order for fast and efficient model training, and finally, the Git for version control, collaboration and deployment.

The hardware requirements for the system implementation are:

i. Minimum 8GB RAM required for smooth processing
ii. Recommended 16GB RAM for deep learning tasks
iii. Intel i5 or higher processor recommended
iv. NVIDIA GPU used for faster model training
v. CUDA support required for GPU acceleration
vi. SSD storage improves data loading speed
vii. High VRAM GPU preferred for diffusion models
viii. Stable internet required for cloud-based training

4.3 Model Training and Optimisation

Data from the CCMT (Crop Pest and Disease Detection Data) dataset were used, and non-plant images were added to the gate model training. The dataset was cleaned, and mislabeled samples were removed, and images were resized to a fixed input dimension, along with normalising pixel values between 0 and 1. The dataset was split into training, validation, and test sets in a 70:15:15 ratio, and data augmentation was then applied to improve generalisation. The gate model was trained for a binary classification task, and the crop classifier, however, was trained for multi-class crop prediction, while disease models were trained separately for each crop type. The training followed a transfer-learning approach using the EfficientNetB0 backbone, with the pretrained weights initialised from the ImageNet dataset and final layers customised for classification outputs.

A. Hyperparameter Configuration

This explains the hyperparameters and the configuration of each hyperparameter along with the purpose of choosing the hyperparameter and setting its configuration to a level or adopting a type.

Table 4.1: Hyperparameter Configuration for EfficientNetB0 Model Training

Hyperparameter	Configuration Used	Purpose
Input image size	224 × 224 × 3	Matches EfficientNetB0 input format
Backbone model	EfficientNetB0	Feature extraction
Pretrained weights	ImageNet	Transfer learning initialisation
Batch size	16 or 32	Controls samples per training step
Learning rate	0.001 or 0.0001	Controls weight update speed
Optimizer	Adam	Adaptive model optimization
Loss function	Categorical cross-entropy	Multi-class disease classification
Activation function	Softmax	Produces class probabilities
Epochs	20–50	Defines training iterations
Dropout rate	0.3–0.5	Reduces overfitting

Validation split	20%	Monitors generalization
Data augmentation	Rotation, zoom, flip	Improves robustness
Model checkpoint	Best validation accuracy/loss	Saves best weights
Early stopping	Enabled	Prevents overtraining

B. Loss Functions and Optimisers

Binary cross-entropy is used for the gate model training, while for the crop classification and disease classification, we made use of categorical cross-entropy. Adam optimiser was selected for fast convergence during model training. In the same vein, gradient updates were performed using the backpropagation algorithm for effective and efficient training, as well as to obtain remarkable accuracy and precision. Finally, training loss was minimised iteratively during the training process.

4.4 System Implementation

We have successfully trained gate, crop, and disease models, saved as files, and loaded them during backend system initialisation, with each model assigned a specific prediction role and a shared preprocessing pipeline across all models. Model inference functions are defined for each stage output format, standardised across models. For pipeline design, the input image was first passed to the gate model, which checks for plant or non-plant, with non-plant images immediately rejected. Subsequently, the plant images are forwarded to the crop classifier, with the crop classifier predicting the crop category. Then, the unsupported crops are filtered out early, and the supported crops are routed to disease models; for example, cassava is routed to the cassava disease model, and maize is routed to the maize disease model.

For the backend implementation, the FastAPI framework was used for backend development, and a REST API endpoint was created for image prediction, while the images uploaded were handled using multipart form data and before processing occurs, input validation is performed. The models were invoked via API request functions, and a JSON response displayed the prediction results. When the user uploads an image using the web interface, the image is sent to the backend API endpoint for pre-processing the input image. The gate model performs initial classification, and the crop classifier processes valid plant images, while the disease model predicts a specific disease class as the final prediction, which will be compiled and displayed on the user interface.

4.5 Results and Evaluation

4.5.1 Gate Model Results

The plant gate model was evaluated using a binary classification approach.

Table 4.1: Gate model performance

Metric	**Non-Plant**	**Plant**
Precision	1.00	0.94
Recall	0.92	1.00
Accuracy	0.96	

Table 4.1 shows that the gate model achieved an overall accuracy of 96%, which demonstrates its strong capability in distinguishing plant from non-plant images. The non-plant class achieved perfect precision, meaning that all images classified as non-plant were actually non-plant. The plant class recorded perfect recall, ensuring that no valid plant images were rejected. Overall, the performance demonstrates that the gate model effectively prevents irrelevant inputs from entering the classification pipeline, thereby improving the reliability of the overall system.

4.5.2 Crop Classification Results (Crop gate model)

Table 4.2: Crop classification performance

Metric	**Value**
Accuracy	84.85%
Precision	87.00%
Recall	84.85%
F1-score	83.49%

Table 4.2 presents the crop classifier (crop gate model) performance, particularly for the target crops cassava and maize, which had remarkable results of 100% recall for both cassava and maize. This means that all cassava and maize samples were correctly identified, which is critical for the downstream disease prediction stage. However, the tomato class recorded a recall of 50%, indicating weaker performance and therefore was dropped and not included in the final deployment pipeline. Since the tomato was not included in the final deployment pipeline, this limitation does not affect the system's practical operation.

Figures 4.1 and 4.2 present the normalised and non-normalised confusion matrix for the crop gate model. After the image has passed the plant gate model and is identified as a crop, it proceeds to this model. The crop classifier model achieved the best validation accuracy of 0.997, final training accuracy of 0.996, and a validation loss of 0.0120 over 15 training epochs.

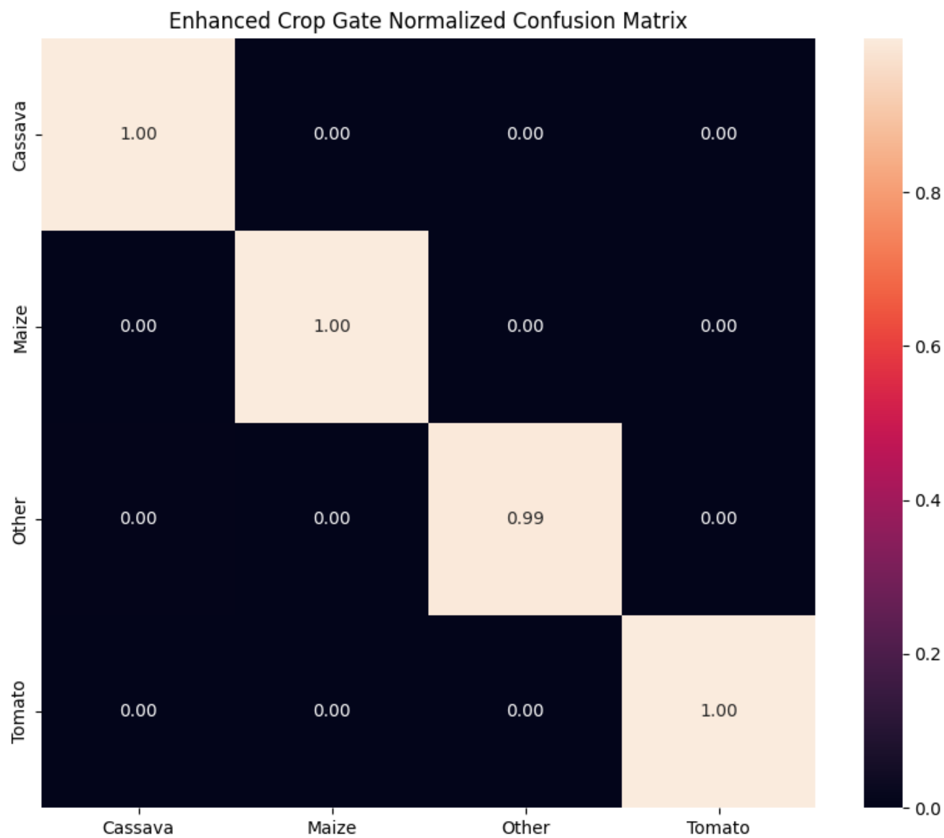


Figure 4.1: Crop classifier model normalised confusion matrix

Figure 4.1 illustrates the normalised confusion matrix of the crop classifier model, showing accuracy per class as percentages. According to the figure, cassava, maize, and tomato are perfectly classified with a value of 1.00, which is 100%, while Other had very slight miscalculations, possessing a value of 0.99 (99%).

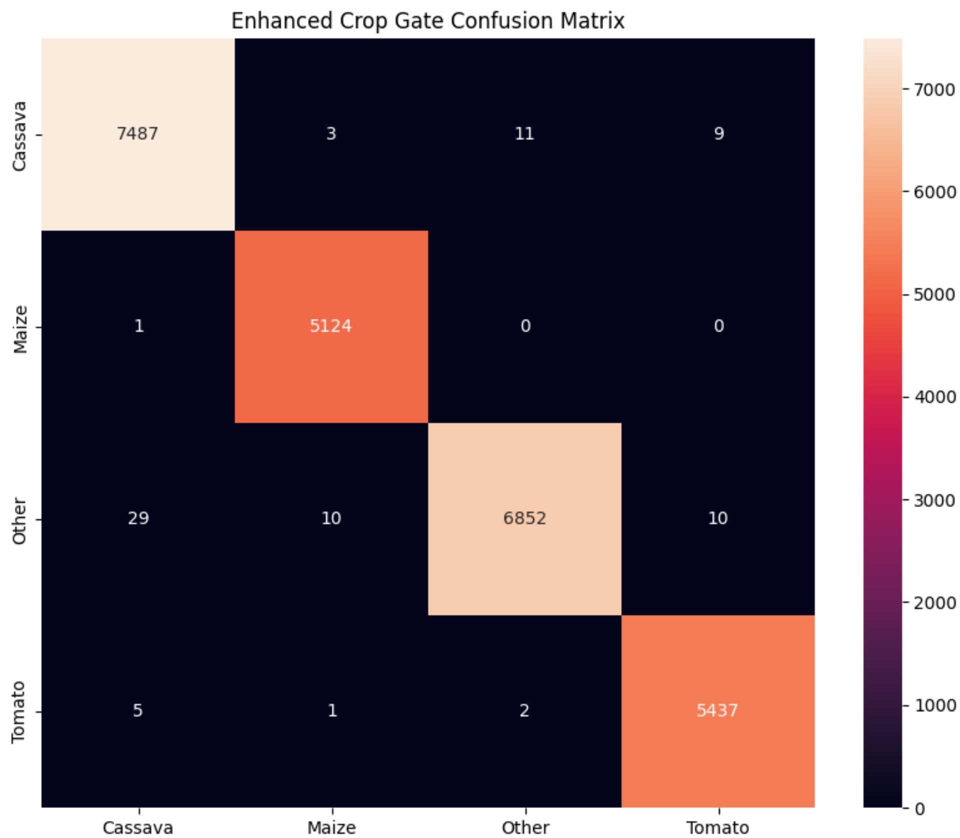


Figure 4.2: Crop classifier model confusion matrix

Figure 4.2 illustrates the confusion matrix of the crop classifier model, which shows the number of correctly and incorrectly classified samples. The non-normalised confusion matrix indicates that the cassava, maize, tomato, and other classes were correctly classified with very few errors and minor confusion or negligible misclassification with respect to the tomato class. Overall, errors are extremely low compared to correct predictions, indicating high reliability and strong generalisation of the model.

4.5.3 Cassava Disease Model Results

Table 4.3: Cassava model performance

Metric	Value
Accuracy	95.18%
Precision	95.31%
Recall	95.22%
F1-score	95.23%

The cassava model achieved excellent performance according to Table 4.3, with good differentiation between categories and low misclassification. Figure 4.3 presents the normalised confusion matrix of the cassava model.

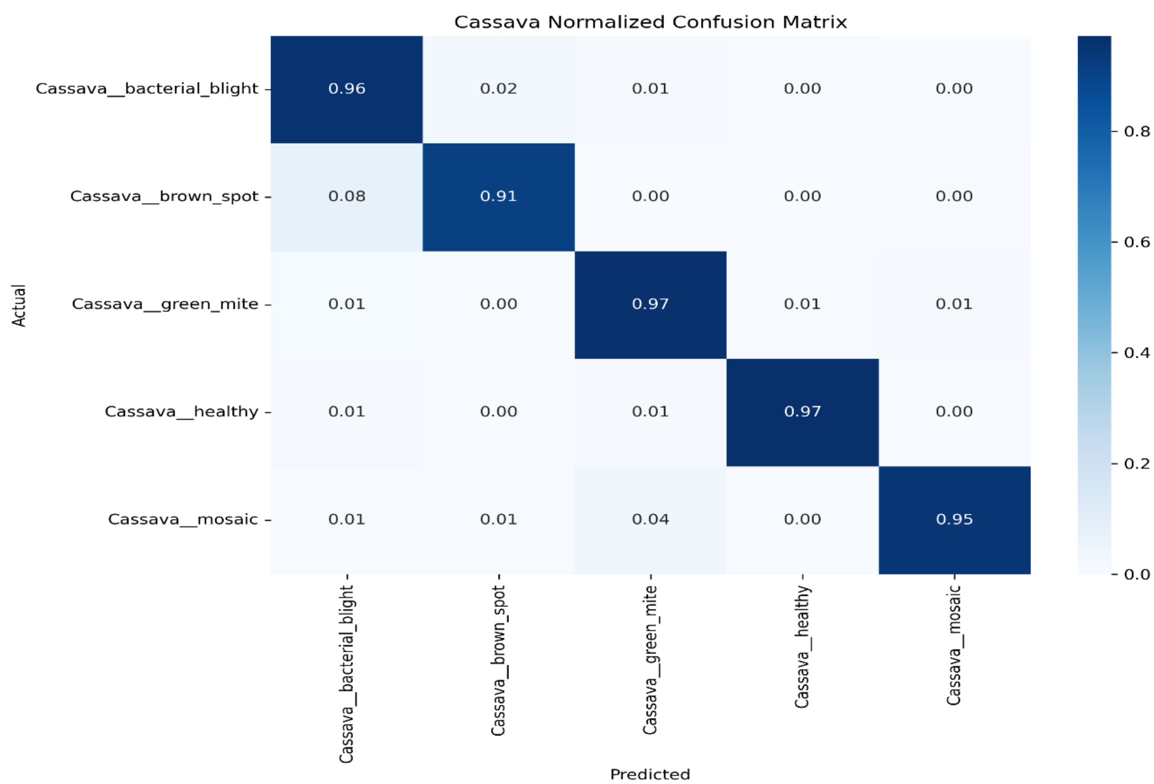


Figure 4.3: Cassava classifier model normalised confusion matrix

The diagonal values of the normalised confusion matrix range between 0.91 and 0.97, thus indicating a strong class accuracy with green mite and healthy as the best classified classes. Minor confusion occurred between brown spot and bacterial blight due to visual similarity, and the mosaic was misclassified as green mite by 4%, as illustrated in Figure 4.3. Overall, their off-diagonal values are very low, confirming minimal class overlap, thereby indicating good feature learning with slight visual similarity challenges.

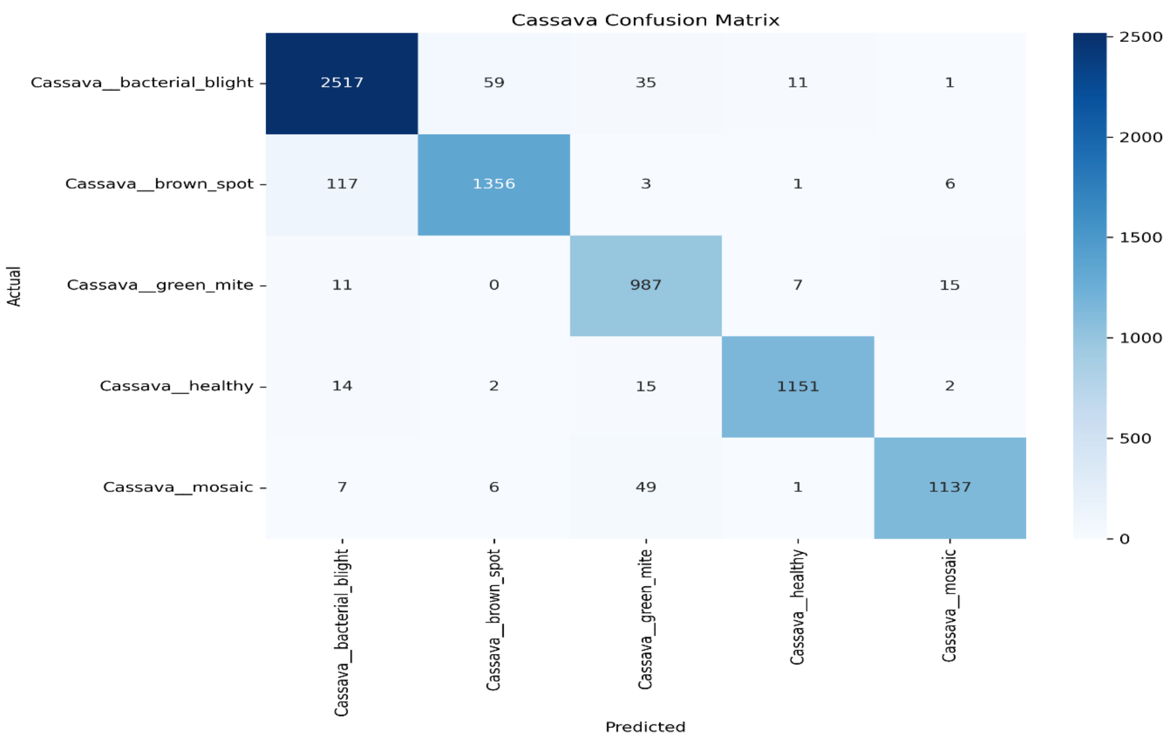


Figure 4.4: Cassava classifier model confusion matrix

Figure 4.4 presents the non-normalised confusion matrix of the cassava classifier model, with the majority of predictions appearing on the diagonal, that is correct classification. While the misclassifications are relatively small compared to the total sample.

Overall, the model demonstrates high accuracy and strong class separability, with the main errors occurring between visually distinct disease patterns. Also worth noting is that the performance remained consistent and reliable for practical deployment, supporting a high accuracy, precision, recall, and F1-score of all above 95%. Figure 4.5 presents a graphical plot illustrating the cassava model training and validation accuracy, while Figure 4.6 shows its training and validation loss of below 1.

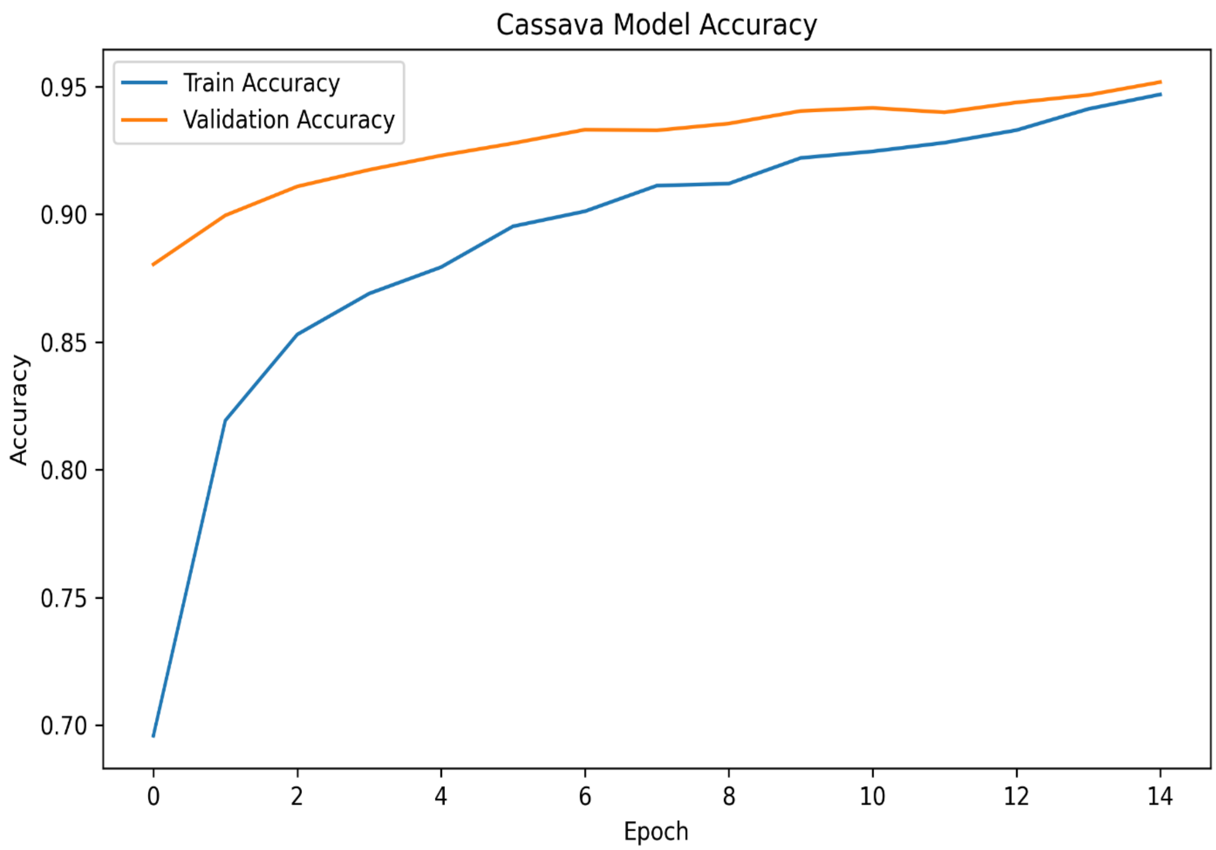


Figure 4.5: Cassava model training and validation accuracy

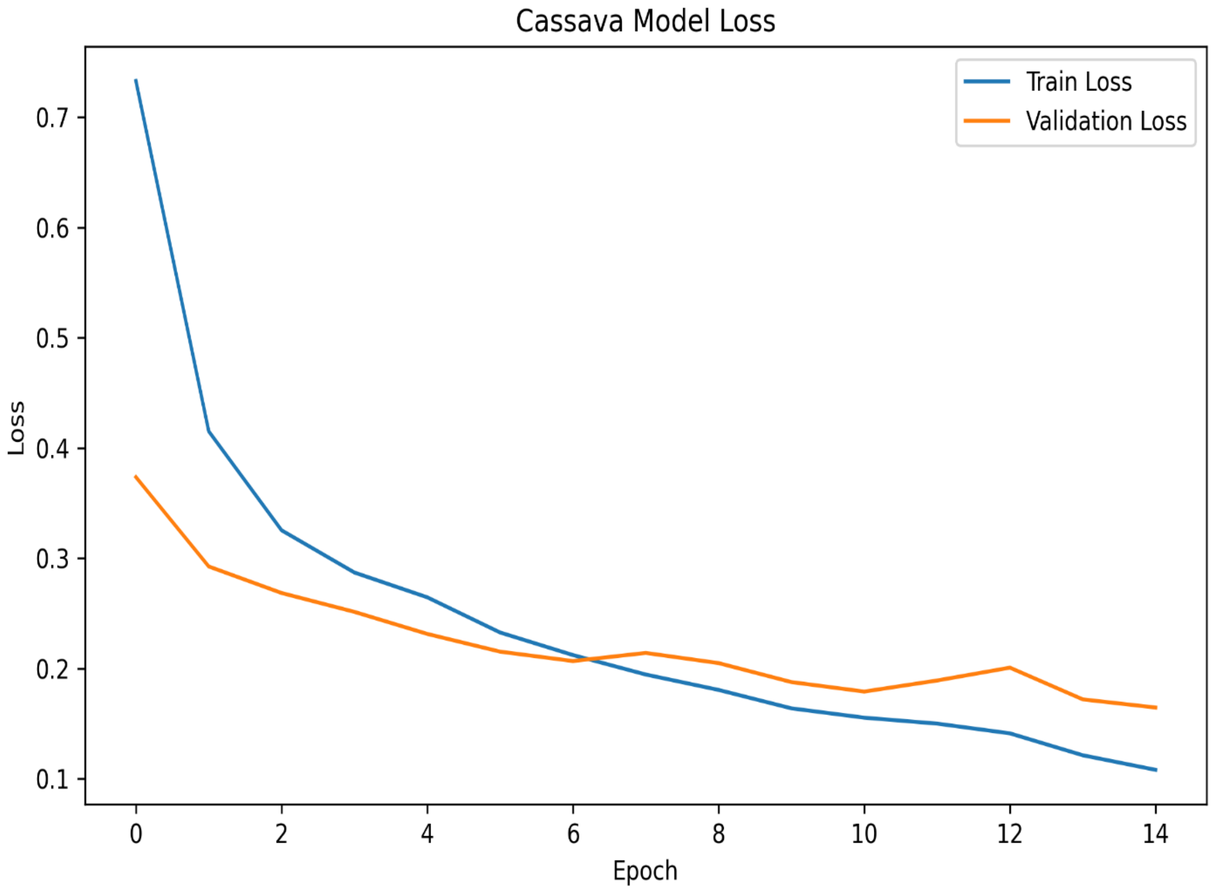


Figure 4.6: Cassava model training and validation loss

4.5.4 Maize Disease Model Results

Table 4.4: Maize model performance

Metric	**Value**
Accuracy	89.66%
Precision	89.42%
Recall	92.93%
F1-score	90.81%

Table 4.4 presents the results of the maize model, which demonstrated strong performance, especially in classes such as grasshopper and leaf beetle. Figure 4.7 presents the normalised confusion matrix of the maize model.

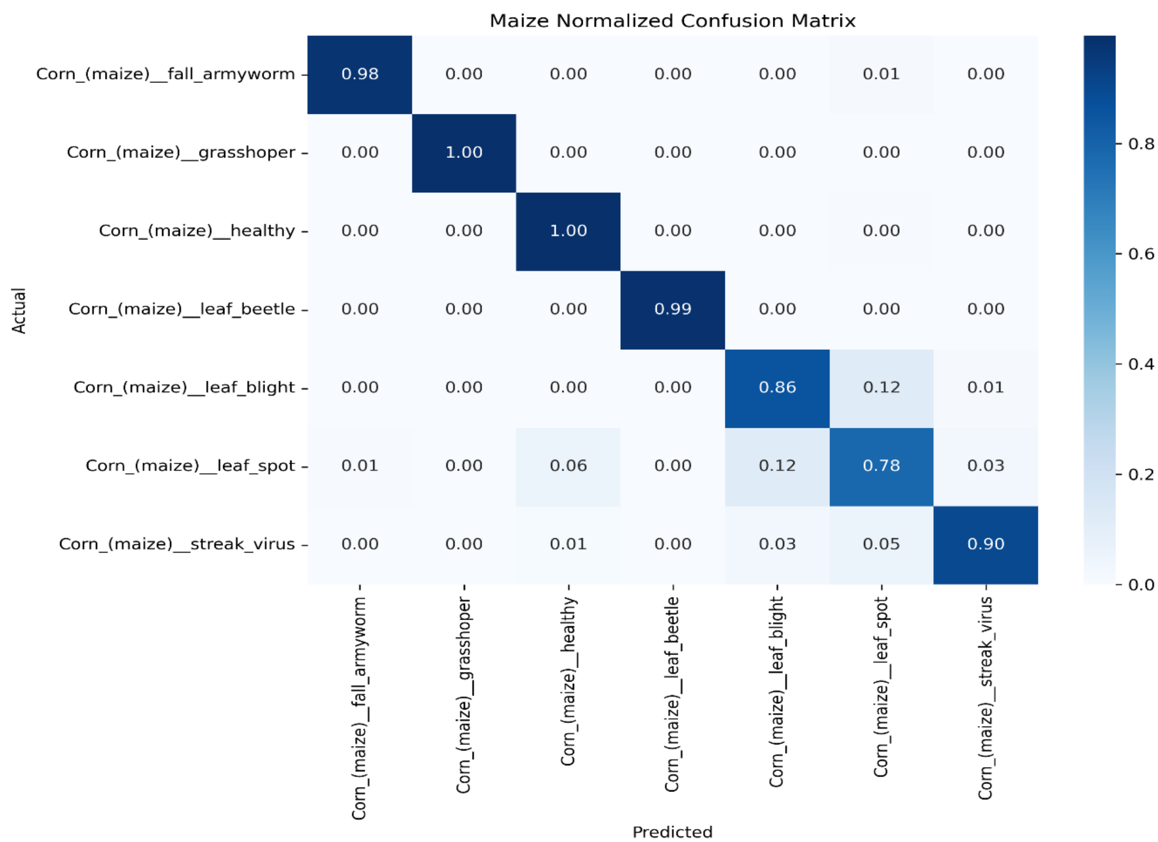


Figure 4.7 normalised confusion matrix of the maize model.

The diagonal values of the normalised confusion matrix range from 0.78 to 1, indicating strong class accuracy for the grasshopper, healthy, fall armyworm, leaf beetle, and streak virus classes. The leaf blight had noticeable misclassification, while the leaf spot had the weakest class performance, as the major confusion observed in these two was a result of Leaf blight having 12% misclassified as leaf spot, and Leaf spot having 12% misclassified as leaf blight and 6% misclassified as healthy

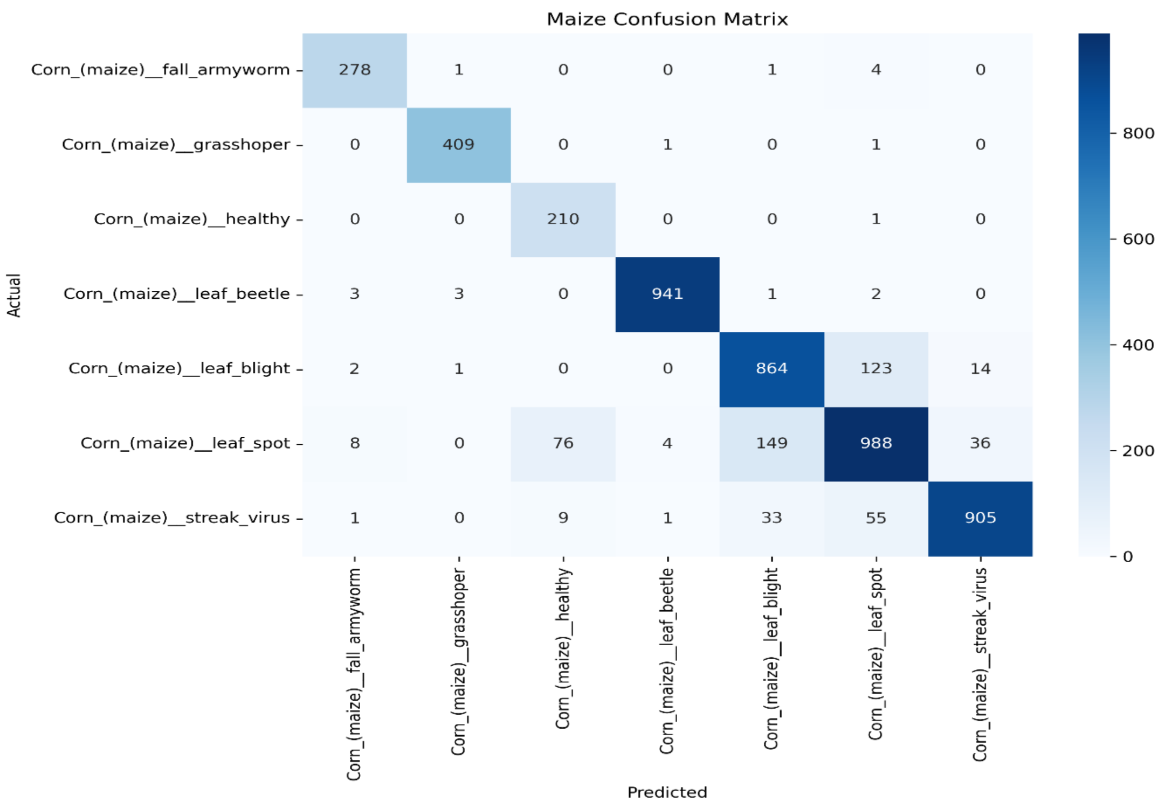


Figure 4.8 presents the confusion matrix of the maize model.

Figure 4.8 presents the non-normalised confusion matrix of the maize classifier model, with the majority of predictions appearing on the diagonal, that is correct classification. While the misclassifications are relatively small compared to the total sample, except for the streak virus and leaf blight, which had slight and significant confusion by misclassifying them as leaf spot (55) and blight (33), for the streak virus and leaf spot, of about 123 cases, only for the leaf blight. Additionally, the leaf spot model showed notable confusion, misclassifying 149 leaf spot cases as leaf blight and 76 as healthy.

Overall, the model demonstrates strong classification performance across most classes, with performance dropping mainly due to leaf blight vs leaf spot similarity. There is a high recall, which aligns with effective disease detection capability, thereby demonstrating that the model is reliable and suitable for real-world agricultural deployment. Figure 4.9 presents a plot of the maize model's training and validation accuracy, while Figure 4.10 shows its training and validation loss.

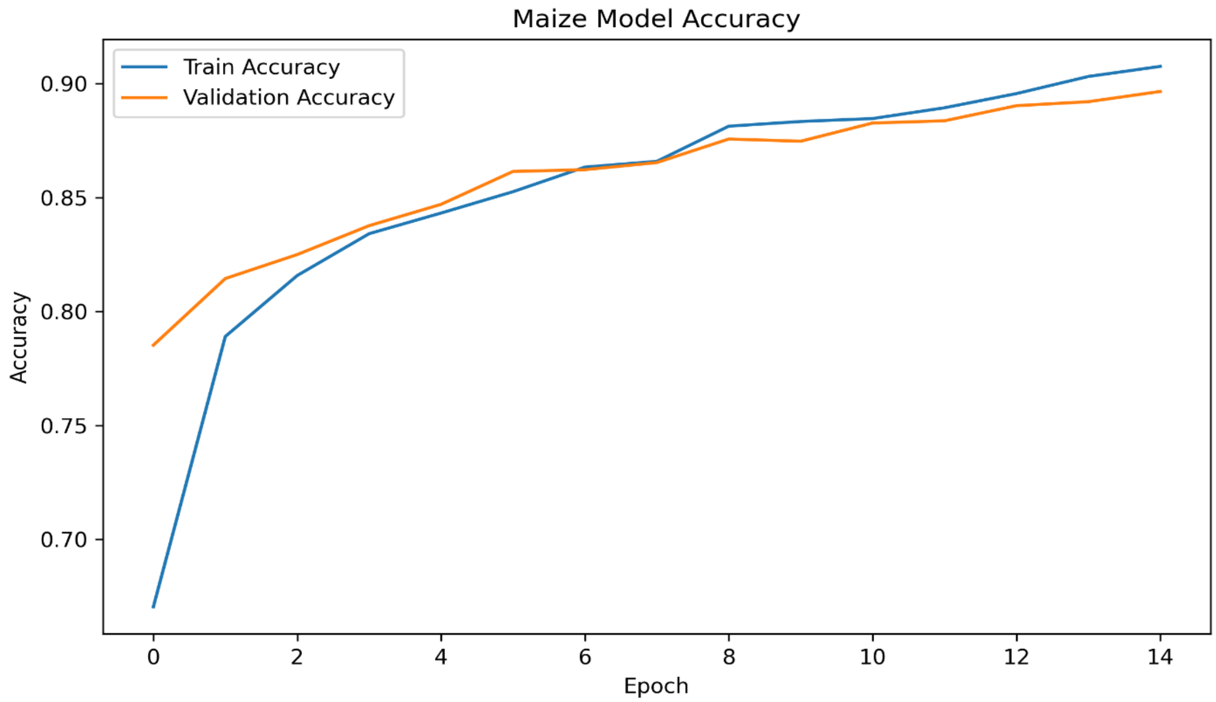


Figure 4.9 Maize model's training and validation accuracy

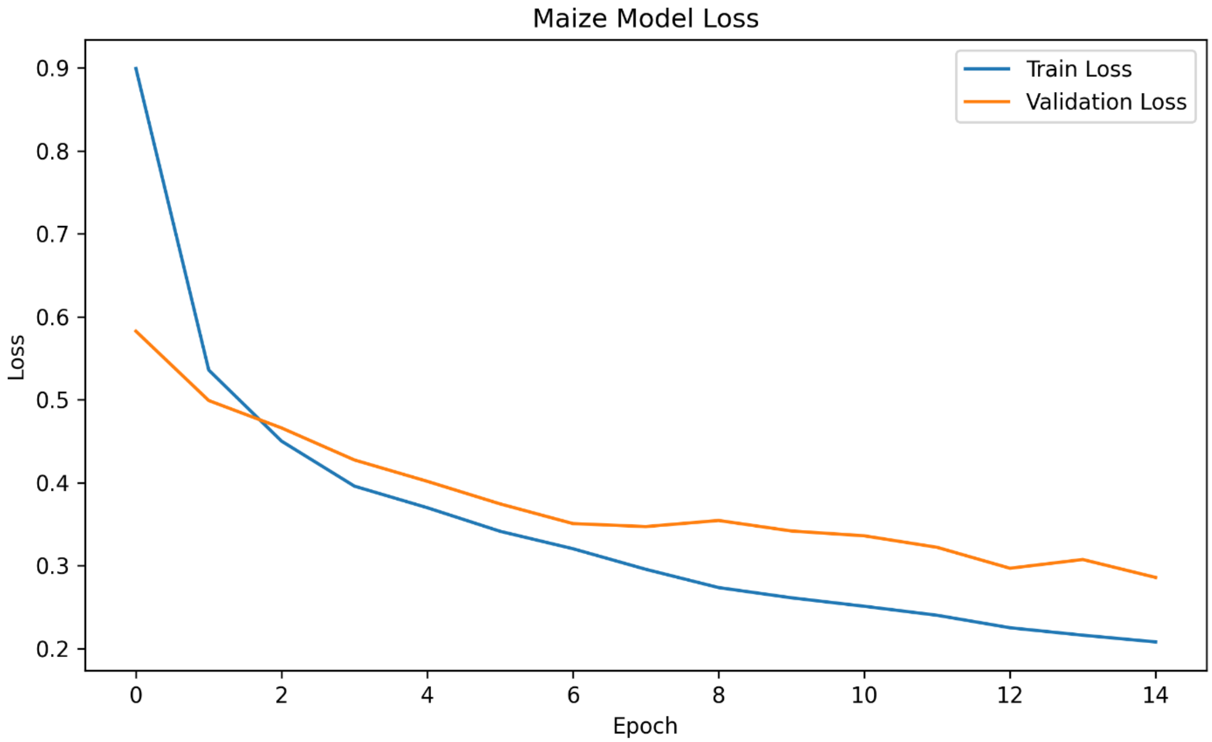


Figure 4.10 Maize model training and validation loss

4.6 Discussion of Results and Final Deployment Decision

The experimental results have shown that the proposed multi-stage architecture provides a practical and reliable solution for plant disease detection. By structuring the system into sequential stages, a gate model, a crop classifier, and crop-specific disease models, the overall performance was significantly enhanced. The gate model effectively eliminated irrelevant inputs at an early stage in order to reduce noise and prevent error propagation, which was then followed by accurate crop identification in order to ensure that only relevant samples were forwarded to the appropriate disease models. From the results, the cassava model achieved consistently high performance with strong class separability, while the maize model also demonstrated solid predictive capability across most classes. Although minor misclassifications were recorded, especially among visually similar disease categories, they did not affect the system's overall effectiveness.

In making the final deployment decision for my system, I placed emphasis on reliability and practical applicability of the system. The gate model, crop classifier, cassava disease model, and maize disease model were retained due to their stable and high-performing results across evaluation metrics. On the other hand, the tomato model was excluded, as its comparatively weaker performance could introduce inconsistencies during real-world use.

Consequently, the deployed system is focused on cassava and maize disease detection, in order to ensure a balance between accuracy and usability. This selective deployment not only strengthens confidence in the system's predictions, but it also reflects a design choice in which we prioritise dependable performance over broad but less reliable coverage.

4.7 System Interface Description

A. User Interface: The User interface is designed for simplicity and ease of use, with the login and get started for registration icon for user authentication.

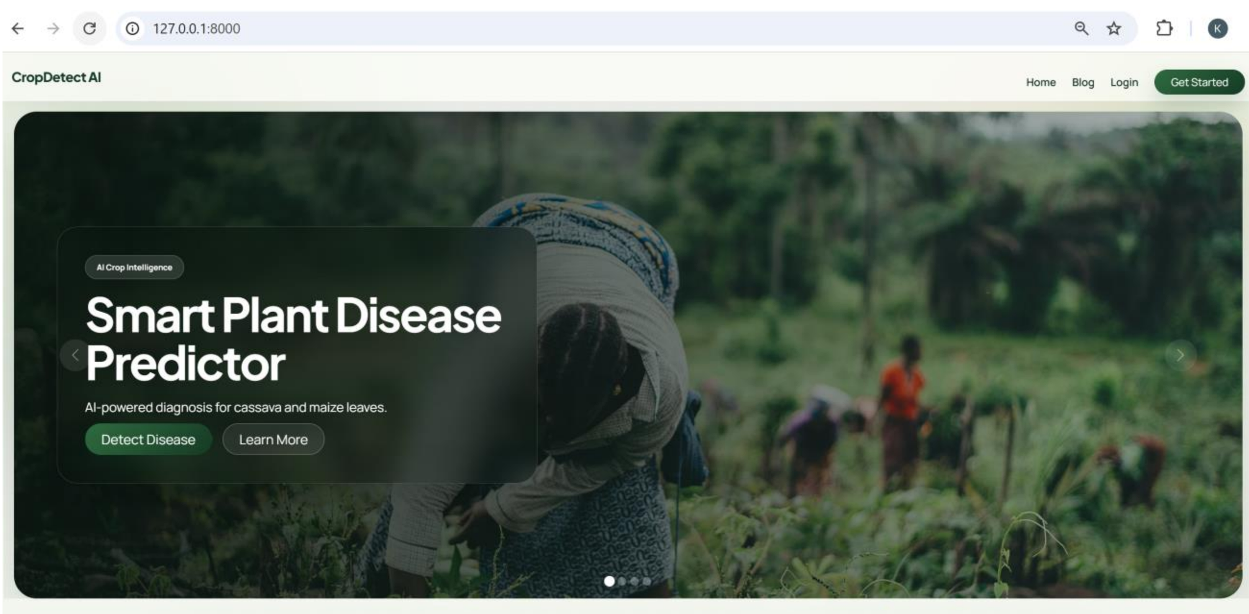


Figure 4.11 Landing page of the new system

B. Dashboard: The dashboard provides access to system functionalities.

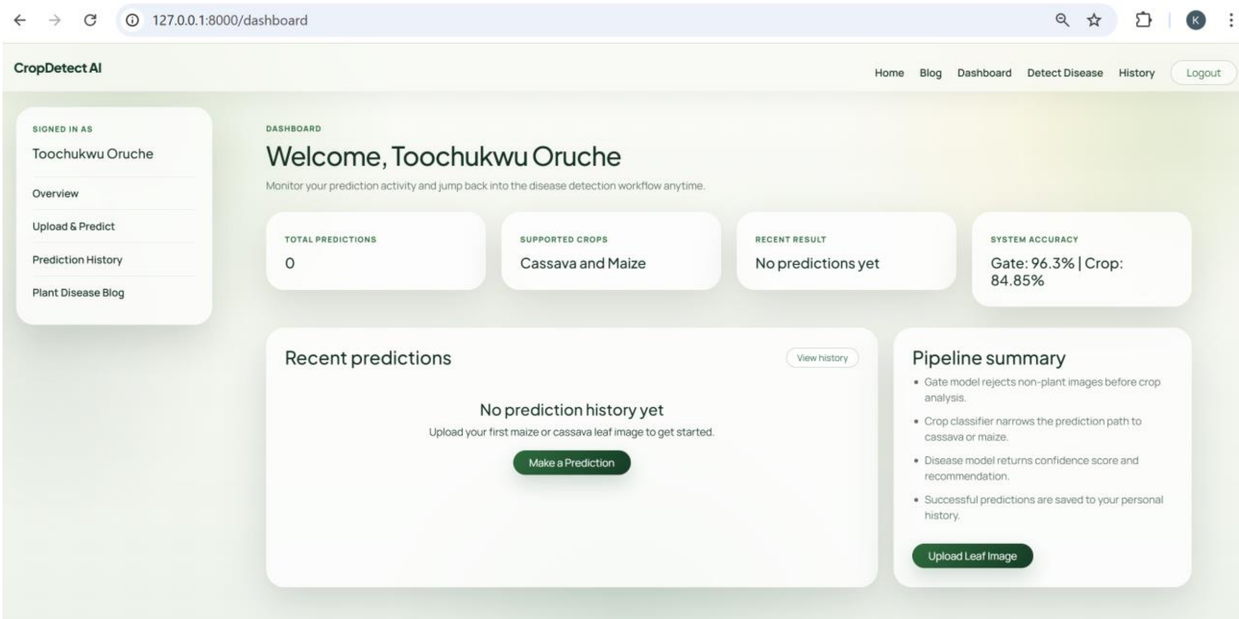


Figure 4.12 Dashboard of the new system

C. Image Upload Page: The image upload interface allows selecting an input image; the image will be previewed, and then the user will click on predictions to see results.

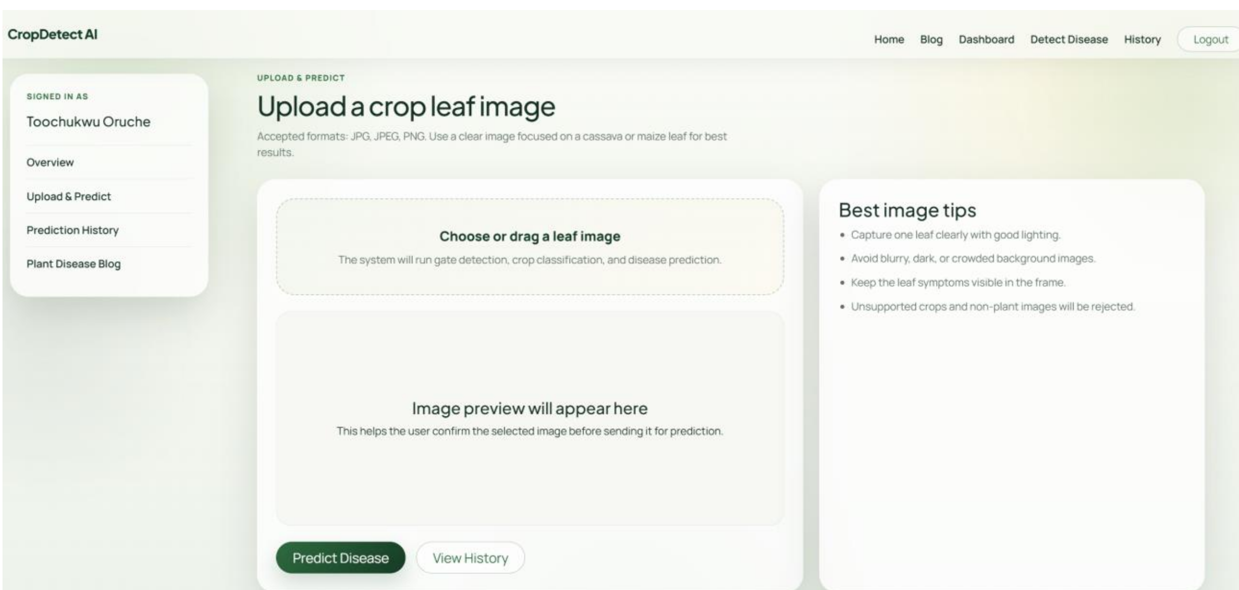


Figure 4.13 Upload page of the new system

D. Output Page: The output section displays the prediction done by the system, and in Figure 4.14, it shows that it has successfully detected the image as a plant and classified it as cassava with a confidence of 77.89%, while the disease was diagnosed as cassava-healthy. While in Figure 4.15, the gate model rejected the image as a non-plant image with a confidence score of 92.85.

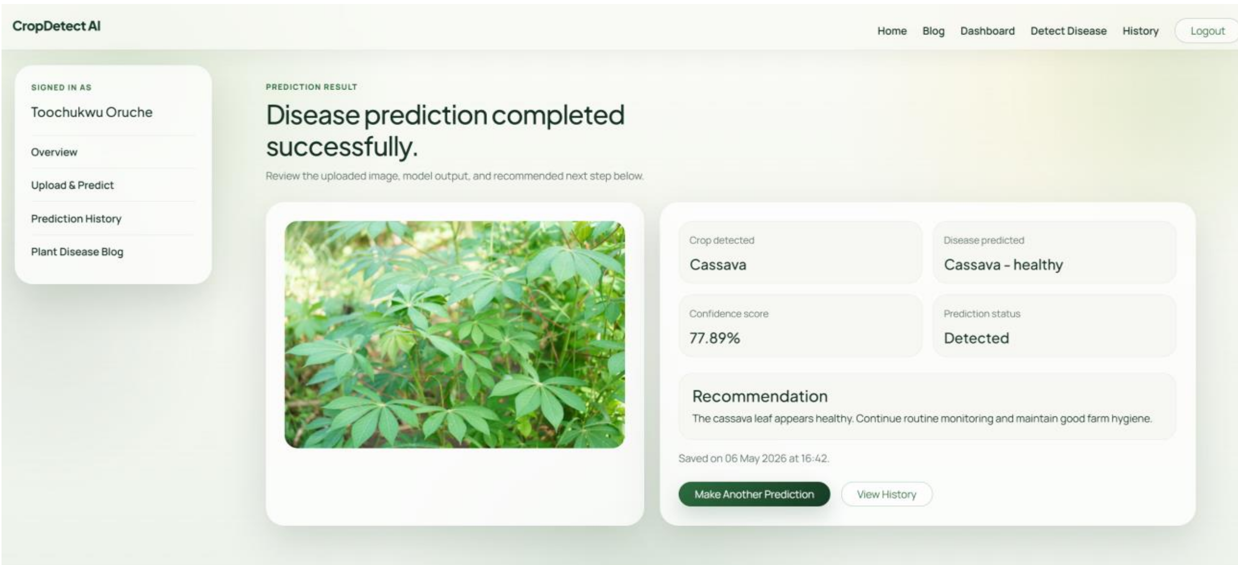


Figure 4.14 Output page of the new system

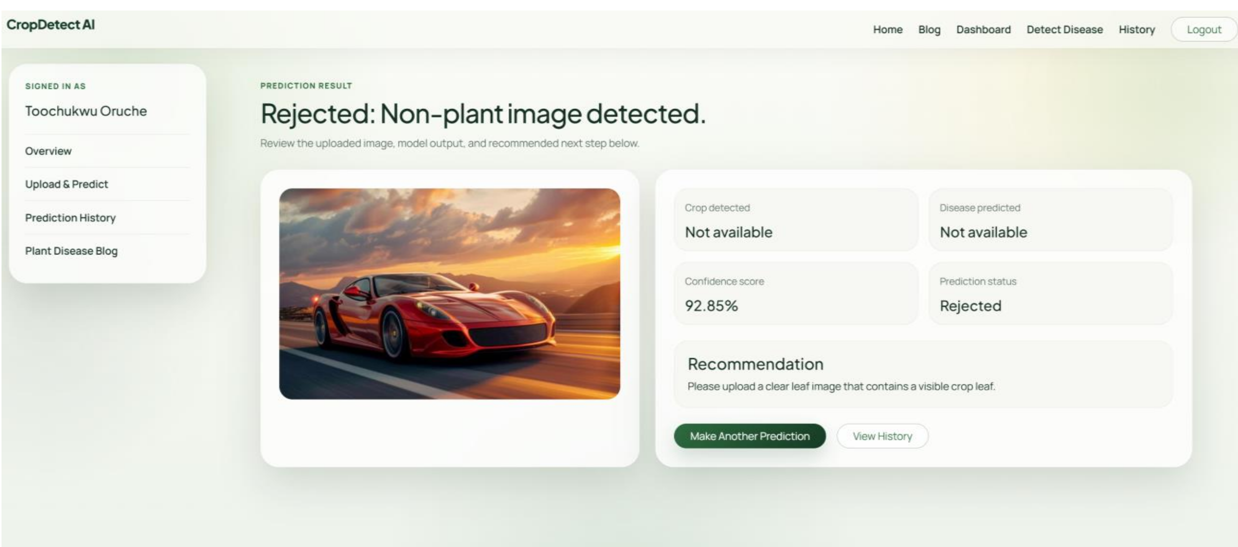


Figure 4.15 Rejected non-plant image by the new system

E. History Page: The history section stores previous prediction records.

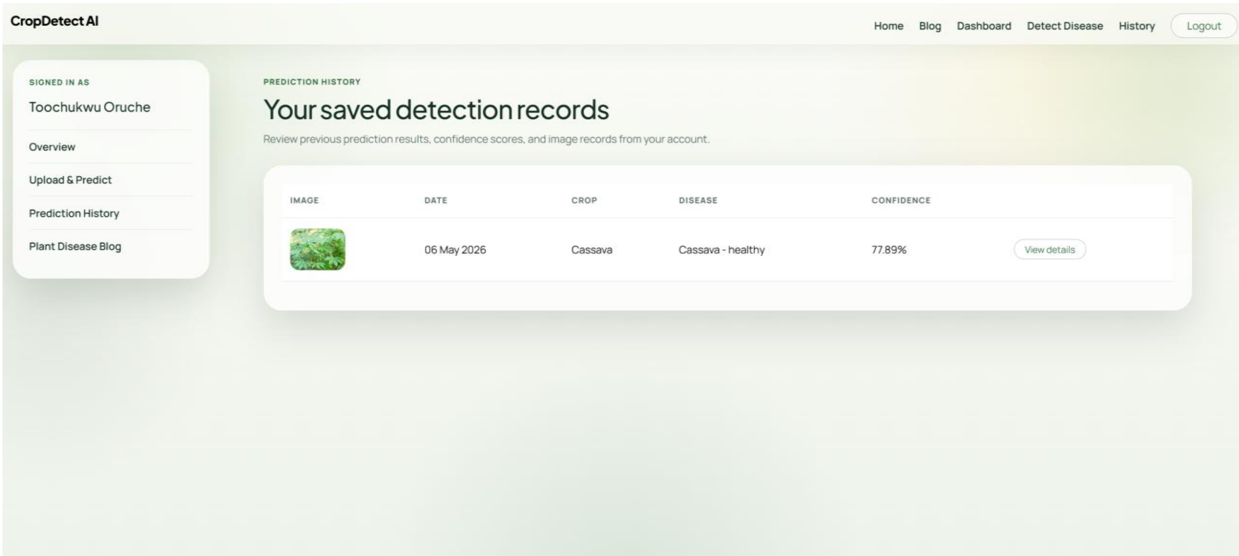


Figure 4.16 History Page

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 Summary

The research came up with a plant disease identification solution based on artificial intelligence (AI). This solution was developed in light of a systematic multiple stages classification model to deal with the problems faced by the local farmers as well as the large scale farmers. The aim was to minimize the chances of losing crops and increasing the crop quantity and quality by providing an effective model which could work on the prediction of diseases. The system accepts a crop image as input, which is then passed to the input processing layer. The gate model first examines the image after the preprocessing steps, which involve image resizing and normalisation, to improve image consistency and learning stability. EfficientNetB0 is used for deep feature extraction in the gate model to perform binary classification, rejecting non-plant images immediately and advancing plant images to the next stages, consisting of the crop classifier and disease diagnostic model to reduce unnecessary disease classification errors.

Models were built using supervised approach by dividing datasets into training, validation, and testing folders (70:15:15). Data augmentation was used for further improvements on the AI model performance. The system was evaluated considering accuracy, precision, recall, and F1-score, showing that the plant gate had remarkable accuracy, the crop classifier had a good accuracy also, the cassava model achieved excellent performance results, and the maize model showed strong but slightly lower performance. In all, the system reduced misclassification through staged processing, while the pipeline helped improve the accuracy and prediction reliability of the system. Finally, the system was implemented using the FastAPI framework for the web application interface for a user-friendly experience for the users, thereby demonstrating suitability for real-world agricultural use.

5.2 Conclusion

The study has successfully developed an intelligent solution for plant disease prediction and classification using deep learning for image-based diagnosis. It employed a multi-stage pipeline to improve classification accuracy, in which a gate model filters out non-plant images to reduce irrelevant data processing, while the crop routing by the crop gate classifier will then identify the plant type after the gate model filtration to enhance disease prediction precision.

The system was evaluated considering appropriate metrics; the cassava model achieved high diagnostic performance, showing strong classification capability, while the maize model performed well across most classes. However, visual similarity affected some disease prediction. Finally, the system achieved high overall performance across metrics, with the model demonstrating strong generalisation ability and the architecture supporting scalable system expansion. The integration improves overall system reliability, emphasising that this approach is suitable for smart farming applications as the system aids early disease detection decisions and provides an efficient alternative to manual diagnosis.

5.3 Recommendations

To further improve the effectiveness and usability of the proposed system, I will recommend several practical steps:

1. First, the system should be expanded to support more crop types so that it can serve a wider range of agricultural needs.

2. The dataset should be improved by including more real farm images to better reflect field conditions.

3. Increasing the number of training samples for classes with lower performance and applying advanced data augmentation techniques will help reduce misclassification.

4. Integration of an explainable AI method, such as Grad-CAM, in order to make the model's predictions more transparent and easier to understand.

References

Abbas, J., Bibi, N., Naqvi, R. A., Sadeghi-Niaraki, A., & Jeong, D. (2024). Revolutionizing agriculture with artificial intelligence: Plant disease detection methods, applications and their limitations. *Frontiers in Plant Science, 15*. https://doi.org/10.3389/fpls.2024.1234567

Agarwal, M., Gupta, S. K., & Biswas, K. K. (2020). Development of efficient CNN model for tomato crop disease identification. *Sustainable Computing: Informatics and Systems, 28*, 100407. https://doi.org/10.1016/j.suscom.2020.100407

Aishwarya, N., Praveena, N. G., Priyanka, S., & Pramod, J. (2023). Smart farming for detection and identification of tomato plant diseases using lightweight deep neural network. *Multimedia Tools and Applications, 82*(12), 18799–18810. https://doi.org/10.1007/s11042-023-15243-7

Alzahrani, M. S., & Alsaade, F. W. (2023). Transform and deep learning algorithms for the early detection and recognition of tomato leaf disease. *Agronomy, 13*(5), 1184. https://doi.org/10.3390/agronomy13051184

Badiger, M., & Mathew, J. A. (2023). Tomato plant leaf disease segmentation and multiclass disease detection using hybrid optimization enabled deep learning. *Journal of Biotechnology, 374*, 101–113. https://doi.org/10.1016/j.jbiotec.2023.02.001

Basavant, N. A., V, G., & L, K. N. (2024). Crop prediction analysis and plant disease detection using machine learning. *International Journal for Research in Applied Science and Engineering Technology, 12*(3), 1234–1245.

Chlingaryan, A., Sukkarieh, S., & Whelan, B. (2018). Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. *Computers and Electronics in Agriculture, 151*, 61–69. https://doi.org/10.1016/j.compag.2018.05.012

Delfani, P., Thuraga, V., Banerjee, B., & Chawade, A. (2024). Integrative approaches in modern agriculture: IoT, ML and AI for disease forecasting amidst climate change. *Precision Agriculture, 25*(2), 456–470. https://doi.org/10.1007/s11119-023-10067-z

Gottemukkala, S., Yalla, P., & Madhavi, K. (2024). Implementation of ant colony optimization and principal component analysis for early prediction of tomato leaf diseases: A feature reduction approach. *Journal of Electrical Systems, 20*(3s), 829–843.

Jasim, Y. A. (2021). High-performance deep learning to detection and tracking tomato plant leaf predict disease and expert systems. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, 10*(2), 45–58.

Jha, P., Dembla, D., & Dubey, W. (2024). Implementation of machine learning classification algorithm based on ensemble learning for detection of vegetable crops disease. *International Journal of Advanced Computer Science and Applications, 15*(1), 123–130.

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture, 147*, 70–90. https://doi.org/10.1016/j.compag.2018.02.016

Kaushik, H., Khanna, A., Singh, D., Kaur, M., & Lee, H. N. (2023). TomFusioNet: A tomato crop analysis framework for mobile applications using the multi-objective optimization based late fusion of deep models and background elimination. *Applied Soft Computing, 133*, 109898. https://doi.org/10.1016/j.asoc.2023.109898

Khan, B., Das, S., Fahim, N. S., Banerjee, S., Khan, S., Al-Sadoon, M. K., ... & Islam, A. R. M. T. (2024). Bayesian optimized multimodal deep hybrid learning approach for tomato leaf disease classification. *Scientific Reports, 14*(1), 21525. https://doi.org/10.1038/s41598-024-52177-2

Li, Y., Nie, J., Chao, X., & Wang, Y. (2020). Early disease detection in crop plants using machine learning approaches. *Plant Physiology and Biochemistry, 148*, 401–414. https://doi.org/10.1016/j.plaphy.2020.01.027

Liu, J., & Wang, X. (2020). Tomato diseases and pests detection based on an improved Yolo V3 convolutional neural network. *Frontiers in Plant Science, 11*, 898. https://doi.org/10.3389/fpls.2020.00898

Lee, S., & Yun, C. M. (2023). A deep learning model for predicting risks of crop pests and diseases from sequential environmental data. *Plant Methods, 19*, Article 145. https://doi.org/10.1186/s13007-023-01122-x

Manchanda, M., Jadli, K., Butola, P., Sharma, V., Bisht, D., & Purohit, K. (2024, April). A real-time tomato crop disease prediction using deep learning. In *2024 International Conference on Inventive Computation Technologies (ICICT)* (pp. 649–656). IEEE. https://doi.org/10.1109/ICICT60155.2024.10544718

Nourali, Z., & Shortridge, J. E. (2024, November). *Predicting the impact of extreme weather on agricultural losses in the Delmarva Peninsula using multi-step machine learning and financial crop loss data* [Preprint]. Research Square. https://doi.org/10.21203/rs.3.rs-5398230/v1

Prottasha, M. S. I., Tasnim, Z., Reza, S. S., & Hossain, D. A. (2021, February). A lightweight CNN architecture to identify various rice plant diseases in Bangladesh. In *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)* (pp. 316-320). IEEE. https://doi.org/10.1109/ICICT4SD50815.2021.9396898

Puls, M. (2021). *Predictive agriculture: The art of understanding uncertainty*. The University of Queensland. https://stories.uq.edu.au/research/2021/predicting-the-future-of-agriculture/index.html

Sakkarvarthi, G., Sathianesan, G. W., Murugan, V. S., Reddy, A. J., Jayagopal, P., & Elsisi, M. (2022). Detection and classification of tomato crop disease using convolutional neural network. *Electronics, 11*(21), 3618. https://doi.org/10.3390/electronics11213618

Sanida, M. V., Sanida, T., Sideris, A., & Dasygenis, M. (2023). An efficient hybrid CNN classification model for tomato crop disease. *Technologies, 11*(2), 45. https://doi.org/10.3390/technologies11020045

Shoaib, M., Hussain, T., Shah, B., Ullah, I., Shah, S. M., Ali, F., & Park, S. H. (2022). Deep learning-based segmentation and classification of leaf images for detection of tomato plant disease. *Frontiers in Plant Science, 13*, 1031748. https://doi.org/10.3389/fpls.2022.1031748

Singh, A., Kumar, S., & Choudhury, D. (2024, January). Tomato leaf disease prediction based on deep learning techniques. In *International Conference on Computation of Artificial Intelligence & Machine Learning* (pp. 357-375). Springer Nature. https://doi.org/10.1007/978-3-031-45645-9_20

Tan, L., Lu, J., & Jiang, H. (2021). Tomato leaf diseases classification based on leaf images: A comparison between classical machine learning and deep learning methods. *AgriEngineering, 3*(3), 542-558. https://doi.org/10.3390/agriengineering3030035

Tarek, H., Aly, H., Eisa, S., & Abul-Soud, M. (2022). Optimized deep learning algorithms for tomato leaf disease detection with hardware deployment. *Electronics, 11*(1), 140. https://doi.org/10.3390/electronics11010140

Thangaraj, R., Anandamurugan, S., Pandiyan, P., & Kaliappan, V. K. (2022). Artificial intelligence in tomato leaf disease detection: A comprehensive review and discussion. *Journal of Plant Diseases and Protection, 129*(3), 469-488. https://doi.org/10.1007/s41348-021-00542-y

Ullah, N., Khan, J. A., Almakdi, S., Alshehri, M. S., Al Qathrady, M., Aldakheel, E. A., & Khafaga, D. S. (2023). A lightweight deep learning-based model for tomato leaf disease classification. *Computers, Materials & Continua, 77*(3), 3969-3992. https://doi.org/10.32604/cmc.2023.046210

Verma, S., Chug, A., & Singh, A. P. (2018, September). Prediction models for identification and diagnosis of tomato plant diseases. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1557-1563). IEEE. https://doi.org/10.1109/ICACCI.2018.8554655

Wang, X., & Liu, J. (2024). An efficient deep learning model for tomato disease detection. *Plant Methods, 20*(1), 61. https://doi.org/10.1186/s13007-024-01185-4

World Health Organization. (2020). *Disease definition*. https://www.who.int

Yildiz, M. B., Hafif, M. F., Koksoy, E. K., & Kurşun, R. (2024). Classification of diseases in tomato leaves using deep learning methods. *Intelligent Methods in Engineering Sciences, 3*(1), 22–36. https://doi.org/10.61300/IMES.2024.3.1.3

Zhou, C., Zhou, S., Xing, J., & Song, J. (2021). Tomato leaf disease identification by restructured deep residual dense network. *IEEE Access, 9*, 28822-28831. https://doi.org/10.1109/ACCESS.2021.3058947

Ashurov, A. Y., Al-Gaashani, M. S. A. M., Samee, N. A., Alkanhel, R., Atteia, G., Abdallah, H. A., & Muthanna, M. S. A. (2025). Enhancing plant disease detection through deep learning:

A depthwise CNN with squeeze and excitation integration and residual skip connections. *Frontiers in Plant Science, 15,* 1505857. https://doi.org/10.3389/fpls.2024.1505857

Chen, J., Zhang, D., & Nanehkaran, Y. A. (2023). A deep learning-based framework for plant disease detection using large-scale field images collected by mobile devices. *Computers and Electronics in Agriculture, 212,* 108072. https://doi.org/10.1016/j.compag.2023.108072

Jha, S., & Kumar, A. (2025). Intelligent plant disease classification using convolutional neural networks and edge-based smart sensors. *Artificial Intelligence in Agriculture, 9,* 45–58. https://doi.org/10.1016/j.aiia.2025.02.004

Jothilakshmi, R., & Sharanesh, R. (2020). Automated plant disease detection using deep learning architectures with autonomous rover. *International Journal of Recent Technology and Engineering, 9*(2), 248–254. https://doi.org/10.35940/ijrte.B3469.079220

Kumar, A., Kumar, P., & Suman, K. (2023). Automated plant disease diagnosis using deep learning model. *International Journal on Recent and Innovation Trends in Computing and Communication, 11*(11).

Li, Y., Zhang, J., & Wang, C. (2024). A lightweight CNN model for real-time plant disease identification using transfer learning on PlantVillage dataset. *Computers and Electronics in Agriculture, 220,* 108865. https://doi.org/10.1016/j.compag.2024.108865

Pai, P., Amutha, S., Patil, S., Shobha, T., Basthikodi, M., Shafeeq, B. M. A., & Gurpur, A. P. (2025). Deep learning-based automatic diagnosis of rice leaf diseases using ensemble CNN models. *Scientific Reports, 15,* 27690. https://doi.org/10.1038/s41598-025-13079-z

Sharma, P., Tripathi, S., & Verma, R. (2024). Deep learning-based computer vision systems for automated plant disease detection: A systematic review. *Computers and Electronics in Agriculture, 217,* 108622. https://doi.org/10.1016/j.compag.2024.108622

Kaur, P. P., & Singh, S. (2021). Classification of herbal plant and comparative analysis of SVM and KNN classifier models on the leaf features using machine learning. *Soft Computing for Intelligent Systems*. https://api.semanticscholar.org/CorpusID:238015920

Iqbal, A., Jaffar, M. A., & Jahangir, R. (2024). Enhancing brain tumour multi-classification using EfficientNet-B0-based intelligent diagnosis for Internet of Medical Things (IoMT) applications. *Information*, 15(8), 489. https://doi.org/10.3390/info15080489

Appendix A

Models.py

```
from datetime import datetime
from sqlalchemy import DateTime, Float, ForeignKey, Integer, String, Text
from sqlalchemy.orm import Mapped, mapped_column, relationship
from app.database import Base
class User(Base):
    __tablename__ = "users"
    id: Mapped[int] = mapped_column(Integer, primary_key=True, index=True)
    full_name: Mapped[str] = mapped_column(String(120), nullable=False)
    email: Mapped[str] = mapped_column(String(255), unique=True, index=True, nullable=False)
    password_hash: Mapped[str] = mapped_column(String(255), nullable=False)
    created_at: Mapped[datetime] = mapped_column(DateTime, default=datetime.utcnow,
nullable=False)

    predictions: Mapped[list["PredictionHistory"]] = relationship(
        back_populates="user",
        cascade="all, delete-orphan",
    )
class PredictionHistory(Base):
    __tablename__ = "prediction_history"
    id: Mapped[int] = mapped_column(Integer, primary_key=True, index=True)
    user_id: Mapped[int] = mapped_column(ForeignKey("users.id"), nullable=False, index=True)
```

```
    image_path: Mapped[str] = mapped_column(String(255), nullable=False)
    crop_name: Mapped[str | None] = mapped_column(String(120), nullable=True)
    disease_name: Mapped[str | None] = mapped_column(String(180), nullable=True)
    confidence_score: Mapped[float] = mapped_column(Float, default=0.0, nullable=False)
    result_status: Mapped[str] = mapped_column(String(120), nullable=False)
    recommendation: Mapped[str | None] = mapped_column(Text, nullable=True)
    created_at: Mapped[datetime] = mapped_column(DateTime, default=datetime.utcnow,
nullable=False)
    user: Mapped[User] = relationship(back_populates="predictions")
```

Dashboard_routes.py

```
from fastapi import APIRouter, Depends, Request
from fastapi.responses import RedirectResponse
from fastapi.templating import Jinja2Templates
from sqlalchemy.orm import Session
from app.auth import get_current_user, pop_flash_messages
from app.content_assets import assign_unique_static_images, resolve_static_image_path
from app.database import get_db
from app.models import PredictionHistory
from app.prediction import format_label
router = APIRouter()
templates = Jinja2Templates(directory="app/templates")
```

```
CAROUSEL_SLIDES = [
    {
        "preferred_image": "carouselfarm.jpg",
        "badge": "AI Crop Intelligence",
        "title": "Smart Plant Disease Predictor",
        "subtitle": "AI-powered diagnosis for cassava and maize leaves.",
        "primary_label": "Detect Disease",
        "primary_href": "/detect",
        "secondary_label": "Learn More",
        "secondary_href": "/blog",
    },
    {
        "preferred_image": "carousel2.jpj.jpg",
        "badge": "Faster Decisions",
        "title": "Early Disease Detection",
        "subtitle": "Detect crop diseases before serious yield loss occurs.",
        "primary_label": "Detect Disease",
        "primary_href": "/detect",
        "secondary_label": "View Dashboard",
        "secondary_href": "/dashboard",
    },
    {
        "preferred_image": "carousel3.jpg",
```

```
        "badge": "Focused Models",
        "title": "Crop-Specific Diagnosis",
        "subtitle": "Cassava and maize models provide focused predictions.",
        "primary_label": "Learn More",
        "primary_href": "/blog",
        "secondary_label": "View Dashboard",
        "secondary_href": "/dashboard",
    },
    {
        "preferred_image": "carousel4.jpg",
        "badge": "Actionable Insights",
        "title": "Farmer-Friendly Results",
        "subtitle": "Upload a leaf image and receive instant guidance.",
        "primary_label": "Detect Disease",
        "primary_href": "/detect",
        "secondary_label": "View Dashboard",
        "secondary_href": "/dashboard",
    },
]

FEATURED_POSTS = [
    {
        "title": "Cassava Mosaic Disease",
```

```
        "excerpt": "Learn how leaf distortion and mosaic patterns can affect cassava yield.",
        "preferred_image": "cassavamossiac.jpg",
    },
    {
        "title": "Maize Leaf Blight",
        "excerpt": "Understand early signs of leaf blight and how farmers can respond quickly.",
        "preferred_image": "maizeblight.jpg",
    },
    {
        "title": "Food Security and Plant Health",
        "excerpt": "See why rapid disease detection matters for harvest quality and farm income.",
        "preferred_image": "cassavaspot.jpg",
    },
]


def get_homepage_slides(current_user) -> list[dict]:
    assigned_images = assign_unique_static_images(
        "carousel",
        [slide.get("preferred_image") for slide in CAROUSEL_SLIDES],
        fallback_name="carouselfarm.jpg",
    )
    fallback_image = resolve_static_image_path("carousel", "carouselfarm.jpg")
    slides: list[dict] = []
```

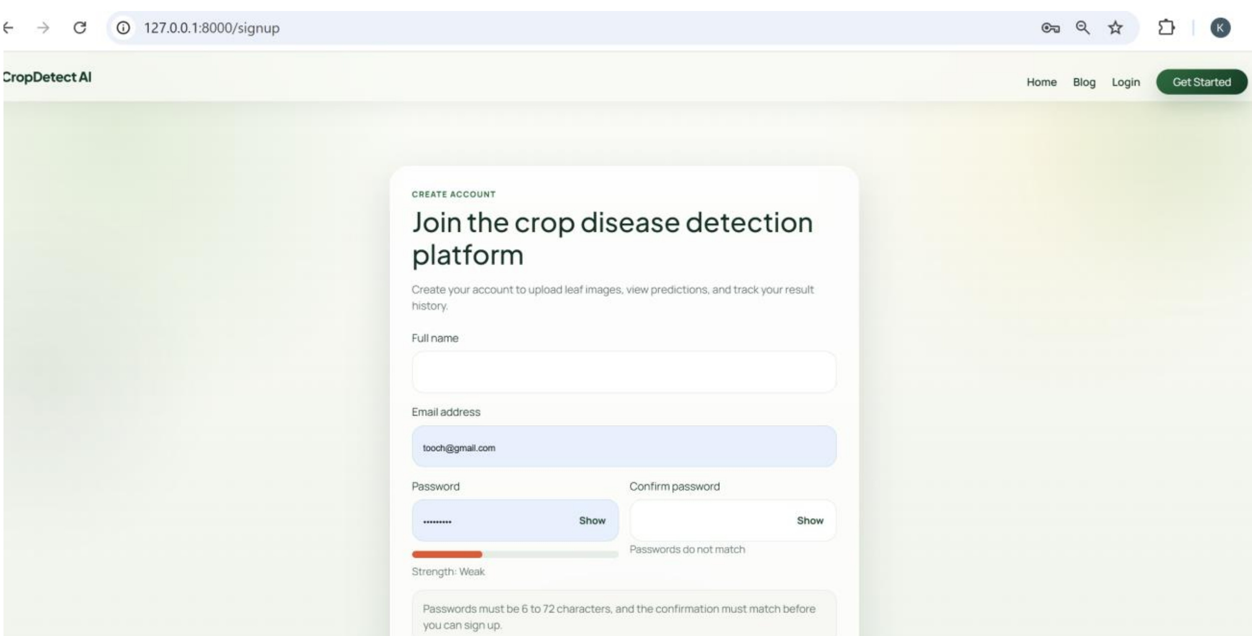
```
    for slide, image_path in zip(CAROUSEL_SLIDES, assigned_images):
        secondary_href = slide["secondary_href"]
        secondary_label = slide["secondary_label"]
        if secondary_href == "/dashboard" and current_user is None:
            secondary_href = "/signup"
            secondary_label = "Get Started"

        slides.append(
            {
                **slide,
                "image_path": image_path,
                "fallback_image_path": fallback_image,
                "secondary_href": secondary_href,
                "secondary_label": secondary_label,
            }
        )

    return slides
def get_featured_posts() -> list[dict]:
    assigned_images = assign_unique_static_images(
        "blog",
        [post.get("preferred_image") for post in FEATURED_POSTS],
```

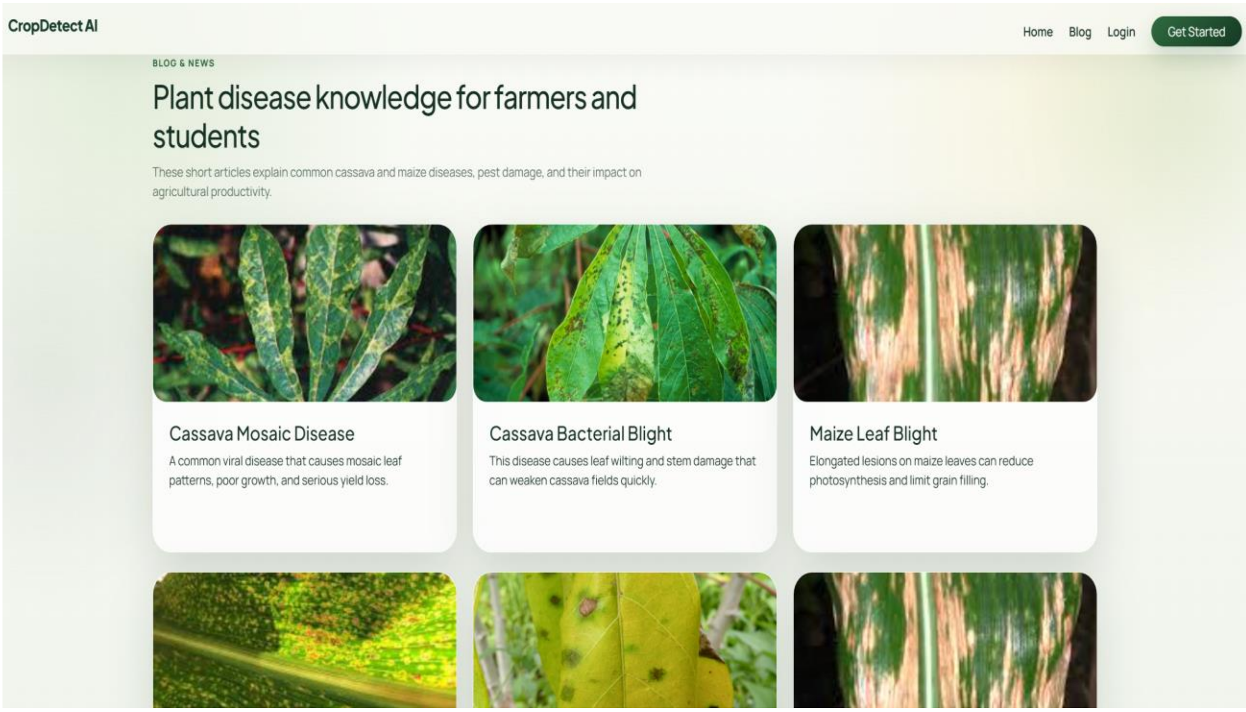
```
        fallback_name="cassavablight.jpg",
    )
    fallback_image = resolve_static_image_path("blog", "cassavablight.jpg")
    posts: list[dict] = []
    for post, image_path in zip(FEATURED_POSTS, assigned_images):
        posts.append(
            {
                **post,
                "image_path": image_path,
                "fallback_image_path": fallback_image,
            }
        )
    return posts
@router.get("/")
def home(request: Request, db: Session = Depends(get_db)):
    current_user = get_current_user(request, db)
    return templates.TemplateResponse(
        request=request,
        name="index.html",
        context={
            "request": request,
            "current_user": current_user,
            "page_title": "Crop Disease Detection System",
```

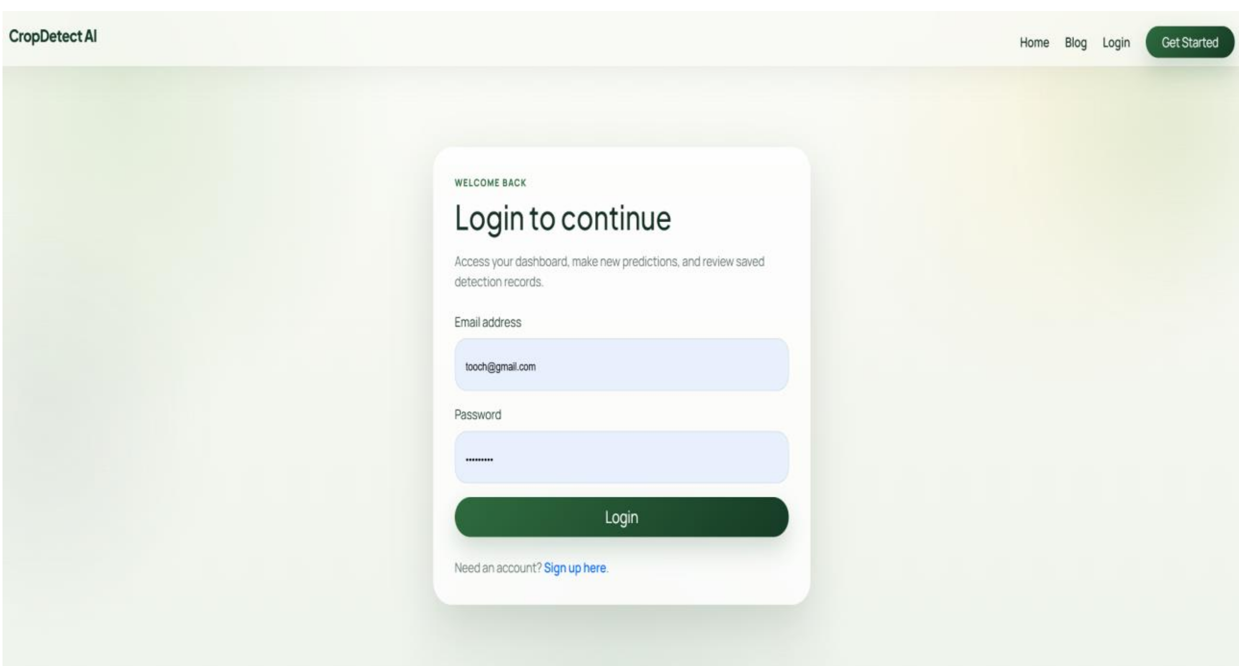
```
            "flashes": pop_flash_messages(request),
            "carousel_slides": get_homepage_slides(current_user),
            "featured_posts": get_featured_posts(),
        },
    )
@router.get("/dashboard")
def dashboard(request: Request, db: Session = Depends(get_db)):
    current_user = get_current_user(request, db)
    if current_user is None:
        return RedirectResponse(url="/login", status_code=303)
    history = (
        db.query(PredictionHistory)
        .filter(PredictionHistory.user_id == current_user.id)
        .order_by(PredictionHistory.created_at.desc())
        .all()
    )
    recent = history[0] if history else None
    return templates.TemplateResponse(
        request=request,
        name="dashboard.html",
        context={
            "request": request,
            "current_user": current_user,
```

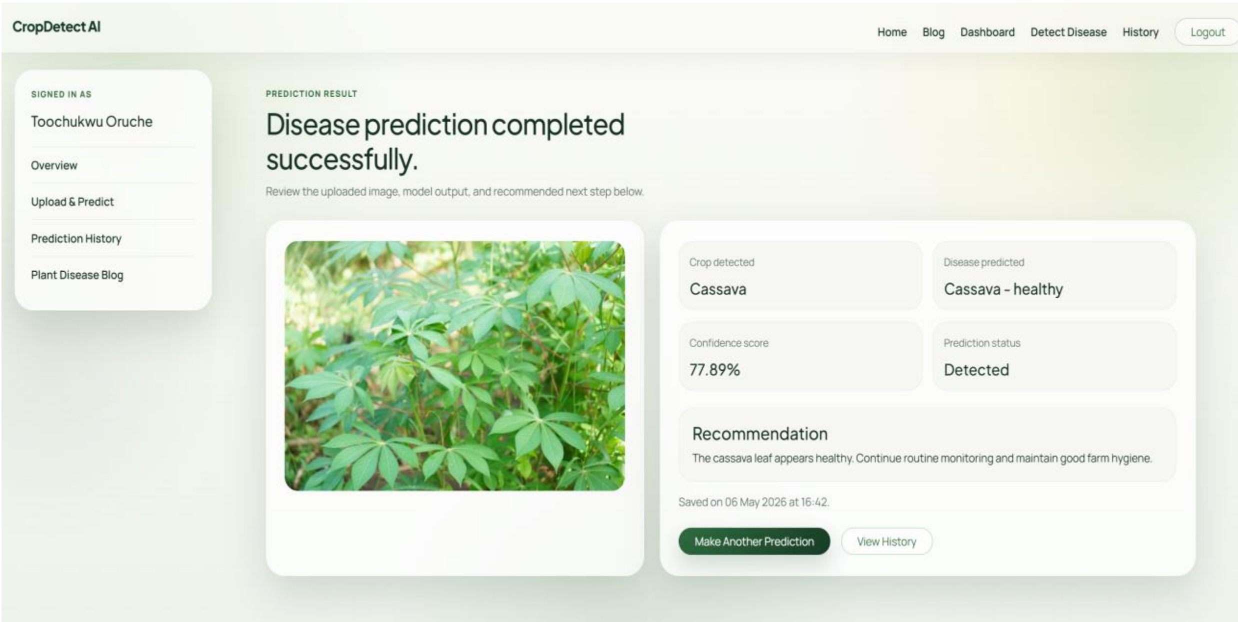
```
            "page_title": "Dashboard",
            "flashes": pop_flash_messages(request),
            "total_predictions": len(history),
            "supported_crops": "Cassava and Maize",
            "recent_result": format_label(recent.disease_name) if recent else "No predictions yet",
            "system_accuracy": "Gate: 96.3% | Crop: 84.85%",
            "recent_history": history[:5],
            "format_label": format_label,
        },
    )
```

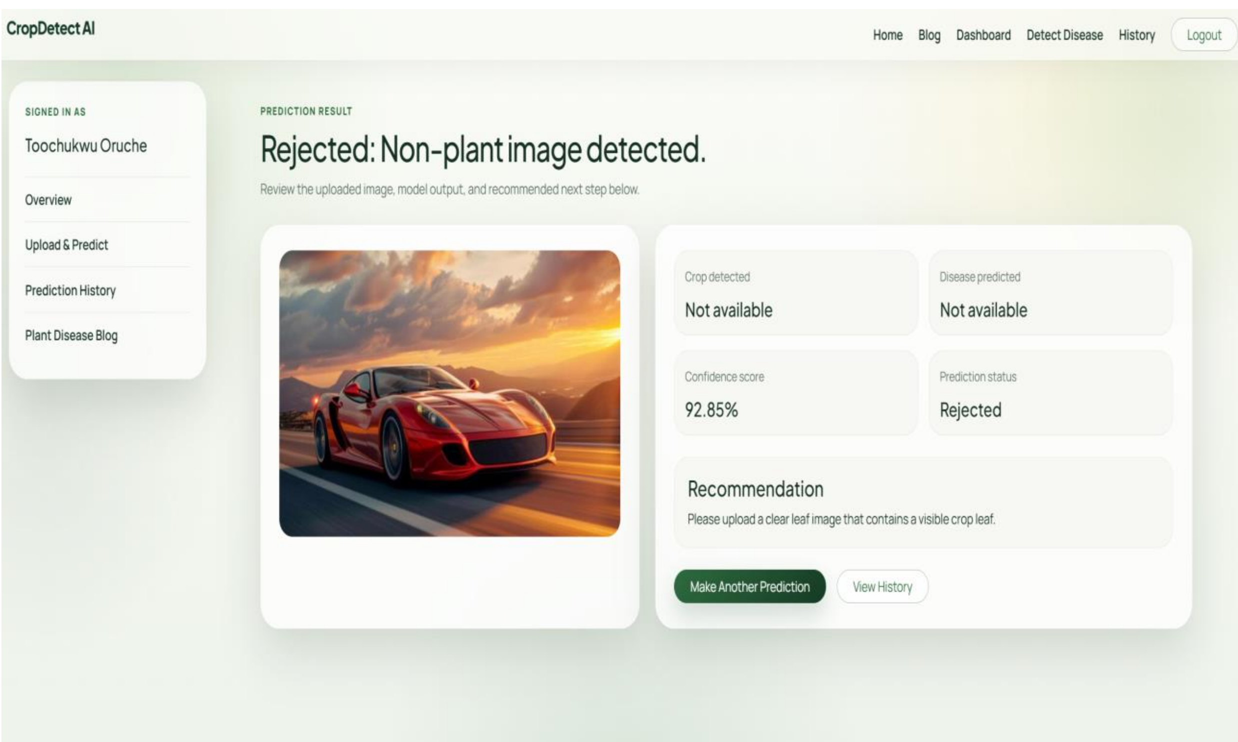
Appendix B Output

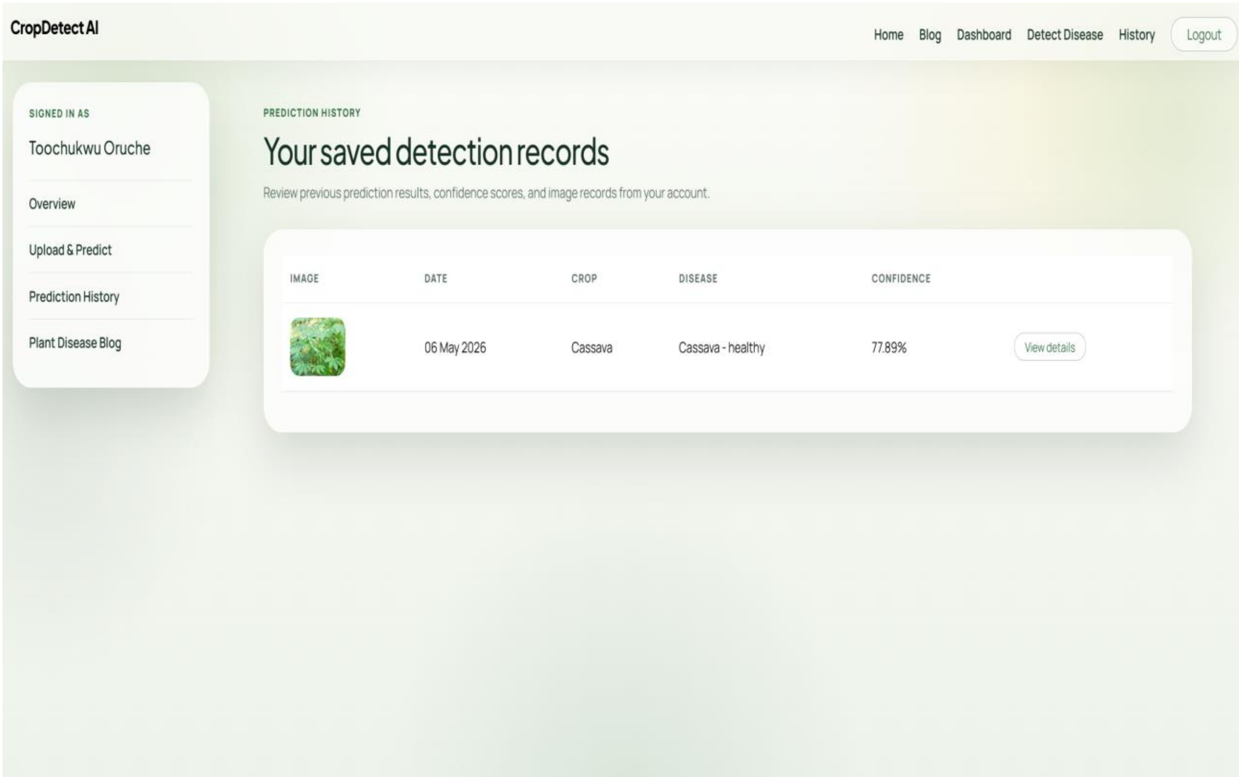



CropDetect AI
Home
Blog
Login
Get Started
BLOG & NEWS
Plant disease knowledge for farmers and students
These short articles explain common cassava and maize diseases, pest damage, and their impact on agricultural productivity.
Cassava Mosaic Disease
A common viral disease that causes mosaic leaf patterns, poor growth, and serious yield loss.
Cassava Bacterial Blight
This disease causes leaf wilting and stem damage that can weaken cassava fields quickly.
Maize Leaf Blight
Elongated lesions on maize leaves can reduce photosynthesis and limit grain filling.


CropDetect AI
Home
Blog
Login
Get Started
WELCOME BACK
Login to continue
Access your dashboard, make new predictions, and review saved detection records.
Email address
tooch@gmail.com
Password
Login
Need an account? Sign up here.


CropDetect AI
Home
Blog
Dashboard
Detect Disease
History
Logout
SIGNED IN AS
Toochukwu Oruche
Overview
Upload & Predict
Prediction History
Plant Disease Blog
PREDICTION RESULT
Disease prediction completed successfully.
Review the uploaded image, model output, and recommended next step below.
Crop detected
Cassava
Disease predicted
Cassava - healthy
Confidence score
77.89%
Prediction status
Detected
Recommendation
The cassava leaf appears healthy. Continue routine monitoring and maintain good farm hygiene.
Saved on 06 May 2026 at 16:42.
Make Another Prediction
View History


CropDetect AI
Home
Blog
Dashboard
Detect Disease
History
Logout
SIGNED IN AS
Toochukwu Oruche
Overview
Upload & Predict
Prediction History
Plant Disease Blog
PREDICTION RESULT
Rejected: Non-plant image detected.
Review the uploaded image, model output, and recommended next step below.
Crop detected
Not available
Disease predicted
Not available
Confidence score
92.85%
Prediction status
Rejected
Recommendation
Please upload a clear leaf image that contains a visible crop leaf.
Make Another Prediction
View History


CropDetect AI
Home
Blog
Dashboard
Detect Disease
History
Logout
SIGNED IN AS
Toochukwu Oruche
Overview
Upload & Predict
Prediction History
Plant Disease Blog
PREDICTION HISTORY
Your saved detection records
Review previous prediction results, confidence scores, and image records from your account.
IMAGE
DATE
CROP
DISEASE
CONFIDENCE
06 May 2026
Cassava
Cassava - healthy
77.89%
View details