# DEVELOPMENT OF A VISION BASED CROP YIELD ESTIMATION SYSTEM

**BY**

**CHIBUEZE GIDEON UWAOMA**

**GOU/U22/CSC/787**

A Project Submitted to the 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. CAMILLUS NJOKU**

**JUNE, 2026**

## APPROVAL PAGE

This research, titled "**DEVELOPMENT OF A VISION BASED CROP YIELD ESTIMATION SYSTEM**," has been assessed and approved by the Department of Computer Science Committees in the Faculty of Computing and Information Technology, Godfrey Okoye University, Enugu, Nigeria.

| | | |
|---|---|---|
| Engr Camillus Njoku | ................................. | ....................... |
| *Supervisor* | *Signature* | *Date* |
| External Examiner | ................................. | ....................... |
| *External Examiner* | *Signature* | *Date* |
| Dr. Frank Okebanama | ................................. | ....................... |
| *HOD, Computer Science* | *Signature* | *Date* |
| Prof. I. A. Ajah | ................................. | ....................... |
| *Dean, Faculty of Computing and IT* | *Signature* | *Date* |

## CERTIFICATION

In doing so, I confirm that I conducted this research myself and that most of the research was derived from my own observations and investigations. Therefore, in case it is proved that I cheated during this exercise, I accept the University's decision without question.

____________________________________

**CHIBUEZE GIDEON UWAOMA**

**GOU/U22/CSC/787**

## DEDICATION

This project is dedicated to my parents, and all who supported my academic journey through their prayers, sacrifice, and encouragement that made this academic journey possible.

## ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my project supervisor, ENGR CAMILLUS NJOKU, for the guidance, patience, and constructive feedback provided throughout the course of this research. Your direction and expertise were invaluable to the successful completion of this project.

I also acknowledge the Head of Department and all the lecturers in the Faculty of Computing and Information Technology at Godfrey Okoye University for their commitment to academic excellence and the knowledge imparted throughout my undergraduate studies.

I also want to acknowledge my friends with their endless support and consistency in our school journey together- Emmanuel Ezeabata, Daniel, , Samuel, Ballack, Ojorbo. Thanks a lot my friends.

My heartfelt thanks go to my parents and family for their unwavering support, prayers, and encouragement. Their belief in my ability has been a constant source of motivation throughout this project.

Above all, I thank God Almighty, the fountain of wisdom and knowledge, without which none of this could have happened.

## ABSTRACT

Food security and agriculture planning at the global level necessitate accurate prediction of crop yield estimates. The methods involved in predicting crop yield estimates usually involve manual farm surveys and simple statistical models which lack regional flexibility, do not capture the ecological diversity, and neglect the visual cues from the crops. With an aim to bridge these gaps, the current research seeks to propose a vision-based multi-modal system for estimating crop yield estimates in the Nigerian setting, emphasizing two of the most prevalent cash crops in the region, viz. cassava and maize. This research uses an OOD & Experimental Deep Learning-based approach..The new CNN architectures used in the model were invented on the basis of the EfficientNet-B0 architecture for analysis and identification of features from the visual images of the crops' leaves to categorize their health condition and diseases. In contrast to the XGBoost model, which receives as input the automatically acquired climate data from NASA (temperature, precipitation, and humidity), along with historical agricultural data from FAOSTAT, to calculate the amount of the output in metric tons per hectare, the innovative characteristic of the methodology presented here is that it does not explicitly measure the plant counts but rather infers the density of the crops by its CNNs. The architecture has been experimented and deployed using FastAPI – a fast and simple to use framework along with web app interface to integrate two explainable AI (XAI) pipelines: Grad-CAM and SHAP for explainable image classification and explainable feature attribution of the climate data correspondingly. The results of these experiments show that the test accuracies of maize diseases classifier is $81.17\%$ and cassava diseases classifier is $70.7\%$. From the two baselines of Linear Regression and Random Forest, the suggested yield regression head based on XGBoost algorithm became the most efficient one, with MAE=$0.00988$, RMSE=$0.02482$ and $R^2$ score=$0.99994$. Production in tons, along with other factors, such as crop growth periods and total area, turned out to be the most impactful. It can be seen that an application of computer vision and weather data may significantly contribute to the development of a very efficient, non-invasive and reliable system of precision agriculture with many possible applications. Keywords: Precision Agriculture, Computer Vision, Deep Learning, CNN, XGBoost, Explainable AI (XAI), Yield Prediction

# TABLE OF CONTENTS

# CHAPTER ONE

# INTRODUCTION

## Background of the Study

Global food security, economic growth and rural lives all depend on agricultural production. For both farmers and governments, accurate agricultural production forecasts will ensure effective resource allocation, policy planning and risk management (Lobell & Burke, 2021; Furbank & Tester, 2021). Crop yield forecasting is a critical aspect of addressing the urgent issue of global food security. By 2050, the world's population is projected to reach 9.3 billion, with a 60% increase in food demand, which requires urgent sustainable and innovative agricultural production methods (Wang et al., 2024).

Historically, estimates of agricultural output have been derived from expert evaluations or empirical models, which might be helpful, but most of the time they frequently fall short of capturing the complexity of environmental interactions influencing crop development (Basso and Liu, 2024). Large and varied datasets that impact agricultural outputs may now be obtained thanks to recent developments in data collecting technologies, such as satellite photography, remote sensing, Internet of Things (IoT) sensors and autonomous weather stations (Sharma et al., 2020; Wolfert et al., 2022). Machine learning (ML) techniques allow us to use complex, nonlinear relationships that would not otherwise be captured by traditional means of forecasting such as linear regression. Given the incredible amount of data that exists on the environment and soil types (LeCun et al., 2022), it makes perfect sense to use ML to more accurately in order to predict crop yield than what traditional methods can provide (Chlingaryan et al., 2023).

Without explicit programming, machine learning, a subfield of artificial intelligence, enables computers to identify patterns in data and gradually enhance their prediction capabilities (Jordan and Mitchell, 2022). For crop yield prediction tasks, methods like Random Forest, Gradient Boosting Machines, Support Vector Regression and Deep Learning models like Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks have been used more and more with encouraging outcomes (You et al., 2022; Khaki & Wang, 2024; Sun et al., 2024). Soil properties, such as pH, nutrient levels (NPK), organic matter and moisture content, also play a key role in soil fertility and plant health; however, weather factors, such as temperature, rainfall, humidity and solar radiation, have a strong effect on plant physiological processes (Pantazi et al., 2022; Zhai et al., 2022). The soil and weather data are added to the crop yield models to predict the entire spectrum of soil and weather dynamics that affect the yield performance of crops (Alaei & Bendechache, 2022).

Recent progress in Artificial Intelligence and Computer Vision has led to new ways of monitoring crops and estimating yields. Vision-based systems use images from cameras, drones, or satellites in order to track crop growth and estimate yields automatically. These systems use deep learning, especially Convolutional Neural Networks (CNNs) in order to find features in crop images and predict productivity. A study by Li et al. (2025) proposes a novel deep learning framework (CNNAtBiGRU) for maize yield estimation in Northeast China, which improves real-time prediction accuracy by integrating spatiotemporal data and anthropogenic factors such as mechanisation.

Vision-based methods will offer significant advantages compared to traditional approaches, such as enabling real-time monitoring, lowering labour costs, and improving the accuracy, unlike traditional approaches. Recent studies have shown that vision-based systems are capable of estimating crop yield through the use of standard camera images instead of having to source sensors and complex measurement equipment, which are very expensive to acquire. An example is seen in a study by Bellocchio et al. (2022), which introduced a vision-based yield estimation system that utilises monocular camera images in order to assess agricultural productivity, thereby decreasing dependence on costly platforms like LiDAR and GPS-based sensors.

Additionally, automated crop characteristic measurements, such as height, number of fruits, leaf area, and density, can all be automatically measured and reported using these computer vision systems. All these features are important measures of yield, which makes this capability significant in having the ability to improve the efficiency and scalability of yield estimating when combined with deep learning models. The use of intelligent technology in precision agriculture has recently developed in ways that can improve crop productivity and reduce waste. Vision systems have a critical role in smart farming, providing ongoing monitoring and early yield estimation in order to allow farmers to optimise their irrigation and fertiliser application schedules and harvest timing.

Despite these advancements, many existing vision-based crop yield estimation systems face challenges such as high computational requirements, limited datasets, and lack of user-friendly implementation. Therefore, this study aims to develop a vision-based crop yield estimation system using computer vision and machine learning techniques to provide an efficient and accessible solution for agricultural yield estimation.

## Statement of Problem

Due to the diversity and dynamic nature of agricultural ecosystems, farmers, politicians and agribusinesses have continued to face challenges in accurately predicting the actual crop yields from their farms.

i. Estimating crop yields using conventional methods does not capture the complexity among weather, soil characteristics and plant development processes that are critical to accurately estimate crop yield. Most estimates are based on historical data, manual surveys or simple statistical models, which cannot be used exclusively as a method of measuring yields.

ii. The difficulty in predicting crop yields is further exacerbated by the lack of consideration for the effects of climate change, soil degradation and the inconsistencies in data collection methods used in existing reports, resulting in inefficient use of resources, incorrect planning and increased financial risk throughout the agricultural supply chain.

iii. As the availability of extensive weather and soil data increases, the traditional crop yield estimation process must change its structure to allow an integration of more sophisticated prediction systems that can handle large and varied data sets and provide accurate predictions of crop yields.

iv. Most traditional machine learning systems for predicting crop yields do not have consistent environmental data, leading to lack of reliability.

This study, in order to support better decision-making and encourage sustainable agricultural practices, will create a reliable, data-driven deep learning framework, which will make use of image data to predict crop yields.

## Aim and Objectives

The study aims to develop a vision-based crop yield estimation system that uses computer vision and machine learning techniques to automatically estimate crop yield from images. The objectives are to:

a. Review related literature on the crop yield estimation approaches

b. Collect and preprocess the crop image dataset for training and testing the deep learning algorithm.
c. Implement a vision-based crop yield estimation system using deep learning models to accurately predict yield.
d. Design a user-friendly web application software interface for yield estimation
e. Evaluate the performance of the proposed model using various performance evaluation metrics

## Significance of the Study

To improve the agricultural production and ensure food security, it is essential to develop a vision-based crop yield estimation system. The precise yield predictions can then inform farming decisions, such as planting, irrigation, fertigation and harvest, optimising resource utilisation and reducing production costs. Accurate forecasts are essential for agricultural planning and policymakers to cope with the changing food system, market fluctuations and mitigate the impacts of climatic variations on food systems. Additionally, by showing how contemporary data-driven methods can solve long-standing problems with yield estimation and by incorporating a variety of environmental factors into a predictive framework, the model acts as a tool for advancing precision agriculture and sustainable farming practices.

This study advances technological innovation in agriculture. Additionally, it creates new research opportunities, promoting more investigation into the application of AI to actual agricultural issues and aiding in the development of resilient food systems in the face of environmental change on a global scale.

## Scope of the Study

The study will focus on using image datasets for predicting agricultural yields. The vision-based crop estimation specifically covers image-based crop monitoring through acquiring and analysing crop images, to enable the extraction of important visual features for yield prediction. It will employ a computer vision approach for effective crop detection and feature extractions, alongside machine learning models to estimate crop yield with improved accuracy. The scope is limited to selected crops which are frequently planted in southern Nigeria, such as cassava or maize, and does not extend to all crop types.

## 1.6 Limitations of the Study

The performance of the model is limited to static image input; the selected crops are indicated in the scope and the small-scale dataset. The system will not include weather forecasting or soil-based prediction, and there won't be a real-time drone deployment as it will be prototyped and tested with web application integration.

## Definition of Terms

**Crop Yield:** The amount of agricultural production harvested per unit of land area, typically measured in kilograms per hectare or tons per acre.

**Machine Learning:** A branch of artificial intelligence that involves building algorithms and models that enable computers to learn from and make predictions or decisions based on data without being explicitly programmed.

**Computer vision:** It is a field of artificial intelligence (AI) that trains computers and systems to process, analyse, and interpret important information from the visual inputs supplied to them, such as digital images and videos

**Deep Learning:** It is a subset of ML based on artificial neural networks with multiple layers that mimic the human brain to analyze complex data, and it enables computers to automatically learn patterns and features from vast amounts of unstructured data.

**Predictive Model:** A computational or mathematical framework that uses input data to predict future outcomes, in this case, the expected crop yield.

**Supervised Learning:** A machine learning approach where the model is trained on labelled data, meaning the input data is paired with the correct output, allowing the model to learn patterns and make predictions.

**Precision Agriculture:** A farming management concept that uses data analysis, remote sensing and technology to ensure crops and soil receive exactly what they need for optimal health and productivity.

**Model Robustness:** The ability of a predictive model to maintain accuracy and performance across various datasets and under different conditions.

# CHAPTER TWO

# LITERATURE REVIEW

## 2.0 Introduction

This chapter presents a detailed review of existing studies on crop yield prediction systems, including the theoretical background, a review of related literature, and a summary.

## 2.1 Theoretical background

### 2.1.1 The Concept of Smart Agriculture

According to Zhang et al. (2022), smart agriculture (also known as precision agriculture and sometimes even referred to as agriculture 4.0) is a modern approach of farming that, using state-of-the-art technologies, is supported by data-driven processes to improve productivity, optimise inputs, resource use and promote sustainability. Integrate information and communication technologies (ICT), IoT, Artificial Intelligence (AI), ML, satellite imagery, drones, robotics and big data analytics with farming operations to drive automation of the farming process towards real-time decision-making in automated agriculture driven by Industry 4.0 ( Kamilaris et al., 2022; Liakos et al., 2023). Smart agriculture fundamentally relies on sensors and remote sensing devices, which are used to collect data in real time. By applying sensors in the field, farmers measure each of these important variables (soil moisture, soil nutrients, temperature and humidity), allowing them to fine-tune irrigation/fertility/pesticide application practices well into the future (Wolfert et al., 2022). Such a data-led approach also minimises wastage of resources and helps maximize Operational efficiency. A case in point is precision irrigation technologies, which ensure that water is applied only when and where necessary, hence minimising the use of this resource and promoting sustainable agricultural practices (Gebbers & Adamchuk, 2021).

Another important component of smart agriculture is machine learning-powered predictive analytics. In order to estimate agricultural yields, anticipate disease outbreaks and suggest the best times for planting and harvesting depending on soil and climate conditions, machine learning algorithms examine vast datasets (Khaki & Wang, 2024). Farmers may proactively solve issues and increase output while lowering losses from unanticipated catastrophes like insect infestations or droughts by using predictive modelling. Precision agriculture's capabilities have increased with the introduction of unmanned aerial vehicles (UAVs), sometimes known as drones. High-resolution imaging from drones is useful for field variability assessment, disease early detection and crop health monitoring (Hunt et al., 2023). These initiatives are further supported by satellite images and Geographic Information Systems (GIS), which provide extensive insights into crop performance and climatic conditions, facilitating improved land management choices (Mulla, 2021).

Another essential component of smart agriculture is the automation of farming processes. In addition to lowering labour costs, autonomous tractors, robotic harvesters and automated weeding systems carry out agricultural tasks with high accuracy and consistency (Pedersen et al., 2020). These developments also aid in addressing the growing labour shortage that the agricultural sector faces globally. Furthermore, by encouraging the effective use of inputs and reducing environmental impact, smart agriculture supports sustainability objectives. According to Bongiovanni and Lowenberg-DeBoer (2004), precision farming technology help save soil, lower greenhouse gas emissions and protect water supplies. In light of climate change, land degradation and growing population demands, smart agriculture therefore aids international efforts to achieve food security

(FAO, 2022). Notwithstanding its many advantages, smart agriculture implementation is fraught with difficulties, including high upfront costs, smallholder farmers' limited access to technology and problems with data privacy and interoperability (Wolfert et al., 2022). To overcome these obstacles, supportive policies, rural infrastructure investments, capacity building and cooperation between the public and private sectors are necessary.

According to Liakos et al. (2023), smart agriculture represents a paradigm shift away from conventional farming methods and towards technology-driven, data-centric agricultural practices. It helps farmers make better decisions, increases the accuracy of yield predictions, improves resource efficiency and promotes sustainable farming practices. Smart agriculture is positioned to be crucial to maintaining a robust and sustainable global food system as AI, IoT and data analytics developments continue.

### 2.1.1 Importance of Smart Agriculture

An integrated approach that merges technologies such as IoT, AI coupled with big data and machine learning allows farmers to make better decisions in real-time and consequently improves sustainability and productivity, so smart agriculture (Wolfert et al., 2022). Smart agriculture is needed to solve the following existing global threats in food production: Climate change, population growth, resource scarcity and Environmental degradation (FAO, 2022). Smart farming provides accurate provision for optimal application of inputs (water, fertiliser and pesticides) that minimises waste, the cost of production and the environmental impact (Liakos et al., 2023).

The other potential of smart agriculture is to contribute towards food security and resilience. The automation of farming tasks using robotics and autonomous systems implies alleviating labour shortages and reducing the physical demands placed upon farmers, while advanced predictive models allow farmers to predict upcoming weather conditions, disease outbreaks and pest infestations proactively acting against them (Kamilaris et al., 2022) Smart agriculture also promotes sustainability by promoting practices that protect soil using methods to conserve water and mitigate greenhouse gas emissions (Bongiovanni & Lowenberg-DeBoer, 2004). As we observe the demand for food continuing to rise with an increasing population, smart agriculture provides us a means to produce food efficiently while preserving our ecosystems for generations to come.

### 2.1.2 Areas of Application of Smart Agriculture

Agricultural Precision Crop Farming (APCF) is a key facet of smart agriculture which uses technologies such as Global Positioning Systems (GPS), Unmanned Aerial Vehicles (UAVs), and Internet of Things devices to monitor crop health, soil conditions, and variability across fields. The goal of APCF is to improve efficiency, sustainability, and productivity in a wide range of agricultural applications (Mulla, 2021). By reviewing the data collected from this type of farming technology, farmers can dramatically reduce waste and minimize their environmental impact by applying nutrients, pesticides and water in more accurate and precise amounts (Liakos et al., 2023). Precision crop farming also provides opportunities for variable rate technology (VRT) to support real-time changes in input application rates based on specific site conditions.

#### a. Livestock Monitoring

As livestock monitoring continues to gain importance within plant and animal science, advances in wearable technology and smart sensor technologies enable farmers to monitor the health, behaviour, and activity levels of their animals in real-time on an ongoing basis (Wolfert et al., 2022). The benefit of continuous monitoring is that it allows for prompt identification of sick animals; however, it also enables farmers to better manage breeding programs and develop feeding regimens that result in improved production performance. Innovations such as controlledenvironment livestock facilities and automated dairy milking operations have had a positive effect on animal performance but have also provided additional benefits related to animal behaviour and welfare (Banhazi et al., 2021).

**a. Greenhouse Automation**

Greenhouse automation is a promising field in which smart technology is having a strong effect. Intelligent systems now control temperature, humidity, light, and soil moisture in greenhouses with great accuracy (Van Straten et al., 2021). By keeping conditions ideal all year, these systems will significantly help increase crop yields and improve produce quality. Many smart greenhouses use AI in order to adjust their settings automatically, which saves energy and reduces manual labor.

**b. Irrigation Management**

Smart farming technologies are helping farmers use water more efficiently for irrigation, through the effective use of IoT-based soil moisture sensors and automated irrigation systems, which make sure water is applied only when and where it is needed, which is especially important in areas with water shortages (Kim et al., 2020). This targeted method saves water and keeps crops healthier by preventing overwatering.

**c. Business and demand management**

Other areas where smart agriculture is proving to be effective include supply chain and logistics management, where technologies like blockchain and IoT are being used to monitor agricultural products from farm to market, thereby improving transparency and minimizing losses and ensuring food safety (Tian, 2022). Smart traceability systems also keep an eye on food handling during transport conditions, such as temperature and humidity, ensuring that the quality of the food remains excellent throughout the journey to consumers.

**d. Control of disease and pests.**

Smart agriculture is making a significant contribution to the management of diseases and pests in agriculture. Machine learning can analyse real-time images from drones and satellites to find early signs of problems in crops, such as disease or pest infestations (Kamilaris et al., 2022). By identifying these problems early, farmers can take prompt and corrective action, minimizing losses and the necessity for broad-spread pesticide application.

**e. Predictive Analytics**

In modern farming, weather and climate forecasting and climate smart agriculture are important. The use of predictive analytics and AI-based weather forecasting systems aids in determining optimal planting and harvesting schedules for farmers (Jones et al., 2022). This planning will be critical to dealing with unpredictable weather and climate extremes, to lessen risk and increase resilience.

## 2.3 Predictive Analytics in Smart Agriculture

The advent of predictive analysis is transforming smart agriculture. Historical and real-time data combined with machine learning, AI and statistical analysis can help farmers predict key outcomes and make better decisions (Kamilaris et al., 2022). The technology allows forecasting of weather, soil health, crop production, pest infestation, market needs, etc. and enables more proactive and efficient farming management (Wolfert et al., 2022).

One of the most significant applications of predictive analysis is in predicting crop yields. Machine learning models can use past information regarding soil quality, rainfall, temperature, types of crops and fertiliser application to predict the yield a farmer could produce in a season (Liakos et al., 2023). An understanding of this enables planning of storage, marketing, and selling. Another benefit of predictive tools is that they can identify signs of disease or pests early, based on drone or satellite imagery, allowing farmers to take timely action and prevent significant losses (Kamilaris & Prenafeta-Boldú, 2023).

Meteorological data input in predictive analytics can enable forecasting of extreme weather, rainfall and droughts, thus providing early warnings to farmers to adapt their planting or irrigation plans and preventing losses and boosting productivity (Jones et al., 2022). This approach can be used for soil health monitoring, which uses sensor data to notify farmers about the availability of nutrients or contamination and when and how to fertilise (Mulla, 2021).

Market price prediction is one of the emerging areas of application of AI, where, by understanding the dynamics of consumer demand, market behaviour, and supply chains, predictive tools can help farmers determine optimal harvest timings to maximise profits (Tian, 2022). They can also utilize these tools to operate the resources more effectively, including forecasting water demand according to the watering needs of different crop growth stages and weather, and optimizing irrigation to meet water demand more accurately and sustainably (Kim et al., 2020). Predictive analysis, when paired with the use of other technologies, such as IoT devices, drones, remote sensing, and big data platforms, enables increasingly accurate and reliable forecasts (Wolfert et al., 2022). This integration empowers farmers with the information they need to make informed decisions based on data. The advantages are huge, as it is not only about productivity and profits, it also plays a part in sustainable farming practices by minimizing waste, conserving resources and reducing the environmental footprint of agriculture.

### 2.2.1 Traditional Techniques for Predictive Analysis in Smart Agriculture

The techniques used in predictive analysis in agriculture can range from simple statistical analysis to more sophisticated machine learning methods. These approaches enable farmers to make more informed choices regarding their crop management, pest control, irrigation, resource allocation, and much more, resulting in increased productivity and sustainability. Let's take a look at some of the traditional methods utilized in smart agriculture:

**a) Regression Analysis**

In agriculture regression analysis has been a staple method of prediction for a long time. Linear regression, particularly, is commonly employed to estimate the yields of crops based on the soil fertility, temperature and rainfall (Mishra et al., 2022). These models can be used for predicting future trends and even demonstrate the impact that an input, such as irrigation or fertiliser use, may have on productivity based on the analysis of historical data.

**b) Projecting future values.**

Time series forecasting involves examining patterns in historical data to make forecasts about the future. They could, for instance, be used by farmers to predict the weather, temperature fluctuations, or even the occurrence of pests. The Auto-Regressive Integrated Moving Average (ARIMA) model is one popular model in this category, particularly for seasonal forecasting (Makridakis et al., 2023). This enables farmers to prepare for an alternate weather and climate situation.

**c) Expert Systems**

The knowledge and experience of agricultural professionals are the basis for the development of expert systems. They make predictions on various subjects such as pest activity, soil quality, or possible crop diseases using pre-defined rules. These types of systems can be very useful, but they tend to be logical based and rule-based, which can restrict their flexibility, particularly in the case of large, complex sets of data and unusual situations.

**d) Remote Sensing (Traditional Methods)**

Traditional remote sensing has been a major tool for agriculture monitoring over the past few decades. Aerial photography or satellite imagery can be used to determine land use, soil structure and crop condition. Plant vigour is monitored using a technique such as NDVI (Normalised Difference Vegetation Index) that analyses reflected light. Although these methods work, they can be costly and may not provide the immediate information that the modern-day farm requires (Mulla, 2021).

### 2.2.2 Vision-Based Crop Yield Estimation

Crop yield estimation using images and AI techniques is called vision-based crop yield estimation and involves capturing images of the crop using a camera, which can be mounted on a drone, satellite or ground-based platform, and then analyzing the resulting data to predict yield. They are based on digital images capturing meaningful features, including colour, texture, shape and spectral reflectance, that are directly related to plant health, biomass and productivity. Vision-based systems can estimate yield automatically and data-driven through real-time or near-real-time observations, as opposed to traditional methods based on manual field sampling or statistical models (Yu et al., 2024; Li et al., 2025). In real-world scenarios, image data collected from UAVs or image sensors on satellites are analyzed by computer vision algorithms to identify crop traits and estimate yield parameters, enhancing the accuracy and applicability of these estimates (Tatsumi & Usami, 2024; Wang et al., 2024).

One advantage of vision-based systems is that they can help in image-based crop monitoring, automate crop counting, and analyse crop growth. These systems can continuously capture images of the crop, identify various growth stages, identify stress indicators, and observe the changes in these vegetation indexes over time, which can then be used to make more precise predictions of yield (Cao et al., 2025; Yu et al., 2024).

Furthermore, computer vision methods like object detection and segmentation can be used to automatically count the number of plants or fruits, an essential process in yield estimation of crops, such as fruits and row crops, where the number of plants or fruits directly correlates with yield (Bellocchio et al., 2022; Zhao et al., 2023). Conventional methods are time-consuming, labour-intensive and often only work at a small spatial scale, whereas vision-based methods provide advantages including being non-destructive, potentially high levels of accuracy, reduced operational costs and the capacity for large-scale monitoring (Yu et al., 2024; Jacques et al., 2021). This has led to the adoption of vision-based yield estimation, as it has become a key part of modern precision agriculture, helping in intelligent decision-making and boosting farm productivity.

### 2.2.3 Modern method for Predictive analysis in Smart Agriculture

Predictive analysis in agriculture has now moved beyond mere descriptive statistics and simple predictions into a more sophisticated realm, making use of advanced technologies. In today's day and age, machine learning, deep learning and big data analytics are ushering in a new era of precision and efficiency, providing clear and accurate insights and predictions for farmers. The following are some of the modern techniques that are widely used in the field of smart agriculture:

**Machine Learning (ML) Algorithms**

**a) Random Forest**

Random Forest is a strong algorithm that uses several models to improve the accuracy of predictions made through tree structures. Using many different trees gives you more accurate results than using simply using one tree alone and reduces the risk of overfitting to either the training or test set. In agriculture, Random Forest can be used to predict the yield of crops, identify types of soils, and distinguish between healthy and unhealthy plants (Liakos et al., 2023). It has excellent features due to its ability to handle large datasets containing many variables, which is perfect for the multiple aspects of modern farming operations.

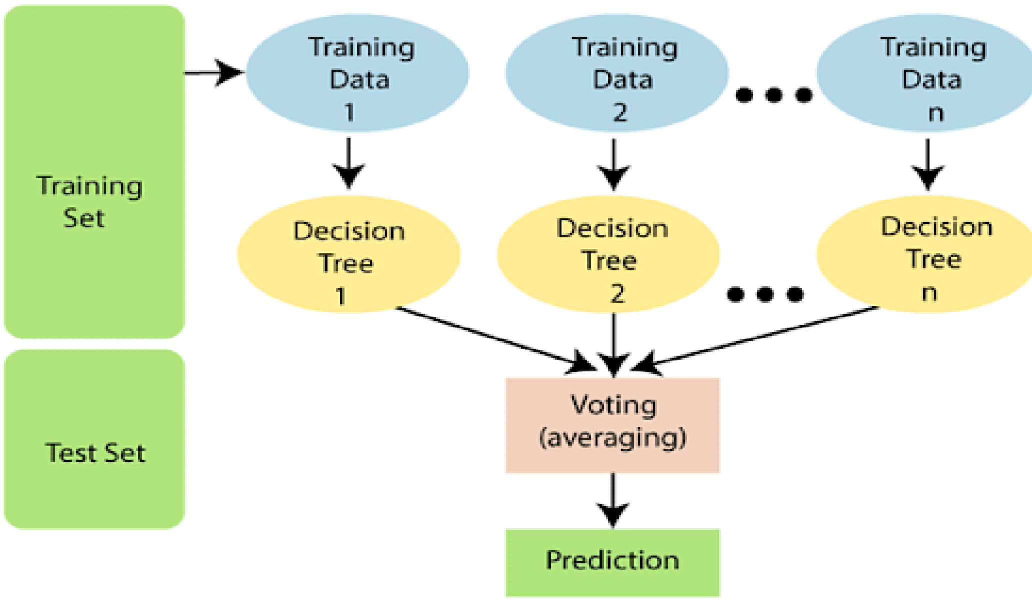



Figure 2.1: The Working Architecture of Random Forest (Simplilearn, 2024)

The Random Forest Algorithm in Figure 2.1 is explained in the following steps:

Step 1: Choose arbitrary samples from a training or data collection.

Step 2: For each training set of data, this algorithm will build a decision tree.

Step 3: The choice tree will be averaged for voting.

Step 4: Lastly, decide which forecast outcome received the most votes to be the final one.

**a) Support Vector Machine (SVM)**

Support Vector Machines (SVMs) are another powerful machine learning technique commonly used for classification and regression tasks. In agriculture, they're especially useful for identifying crop diseases, predicting soil properties and recognizing different stages of crop growth using sensor data. SVMs work well with small to medium-sized datasets and are particularly effective when the data is complex or nonlinear (Kamilaris & Prenafeta-Boldú, 2023). One of their strengths lies in creating clear boundaries between categories. As shown in Figure 2.2, we selected L2 as the hard margin for optimal class separation.

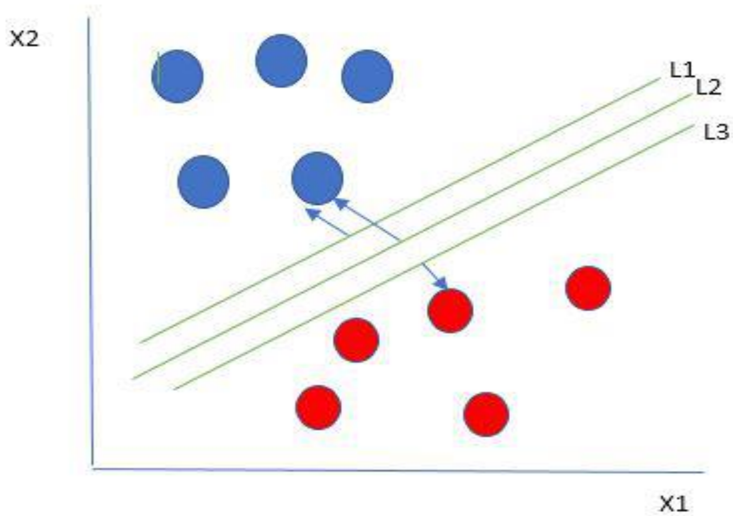



Figure 2.2: Multiple Hyperplanes Separate the data from two classes (Geeksforgeeks, 2025a)

Let's look at a binary classification problem where there are two classes, +1 and -1. The input feature vectors X and the class labels Y that correspond to them make up our training dataset. b) Artificial Neural Networks (ANNs) Artificial Intelligence (AI) is a field of technology that mimics the human brain's operations to pattern recognition and identify relationships in complex sets of data.

**b. Artificial Neural Networks (ANNs):** These are machines that are modeled after how the human brain works, and they are designed to identify patterns and relationships in complex data sets. They are used in agriculture for various prediction and forecasting tasks like yield prediction, soil moisture prediction, early detection of pests etc. ANNs have several impressive features, including the capacity to learn and improve continuously with the passage of time and with the increase of data. They are also very useful for climate prediction and disease detection in crops (Russell & Norvig, 2020).

**Deep Learning Models**

**a) Convolutional Neural Networks (CNNs):**

CNNs are a class of deep learning models which are very good at analyzing images. They are commonly deployed in smart farming for disease detection in crops, detecting plant stress, and biomass estimation using satellite or drone imagery. These models are capable of detecting indicators such as leaf discoloration or wilting, which can be signs of nutrient deficiencies or pest issues (Kamilaris & Prenafeta-Boldú, 2023). The CNNs are a crucial part of the shift towards accurate and technology-driven farming as they can handle huge amounts of visual data.

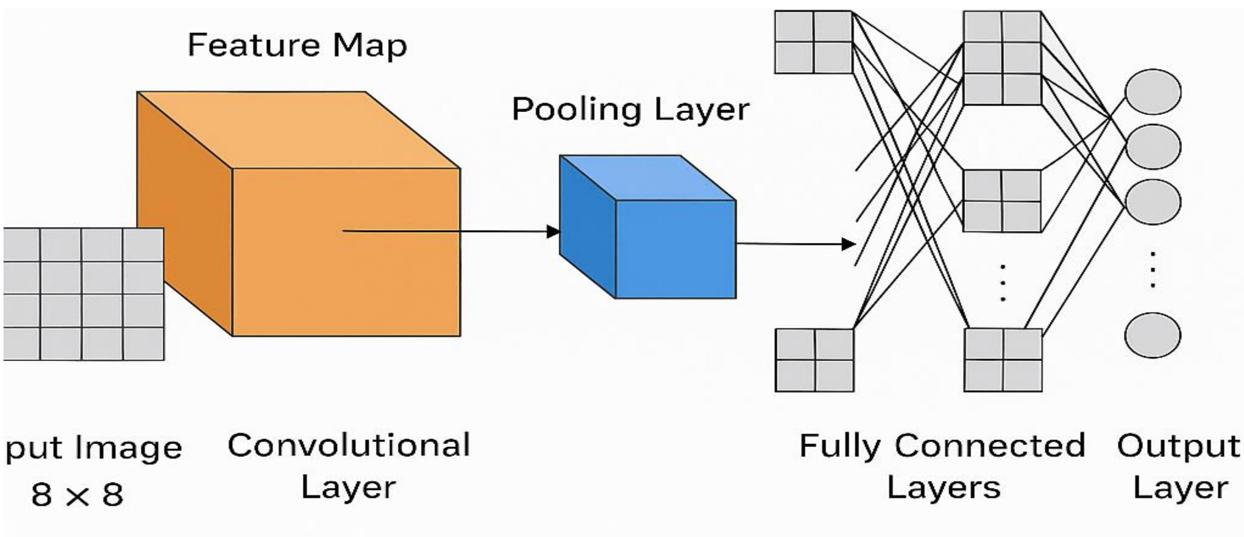



Figure 2.3: CNN Architecture

Figure 2.3 presents a 2D diagram illustrating the architecture of a Convolutional Neural Network (CNN), which in its architecture shows the flow of an 8x8 input image through various layers, including the Convolutional, Pooling, Fully Connected layers and the Output layer.

a) **Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTMs)**

The RNN and LSTM are very capable of analysing time-series data, particularly for detecting trends in previous time-series data as well as for making forecasts based on previous time-series data, like predicting future weather conditions, estimating crop yields that change with seasonality or monitoring soil moisture levels, for example. LSTMs, however, provide an advantage because they retain memory beyond time windows, which allows them to provide insights into highly complex climate behaviours or crop performance over an entire growing season (Zhang et al., 2020).

**b) Support vector machines (SVM)**: They are routinely used by machine learning practitioners for-classification or regression problems. The application of SVMs to the agriculture sector has included things such as identifying crops with diseases, predicting soil characteristics and identifying crop stages using sensors. SVMs are very suitable for small to medium-sized datasets and perform well when they are used with highly sophisticated datasets or datasets with non-linear characteristics (Kamilaris & Prenafeta-Boldú, 2023). SVM classifiers are able to define a separate boundary between the classes. As illustrated in Figure 2.2, we selected the L2 norm as the hard margin to achieve optimal class separation.

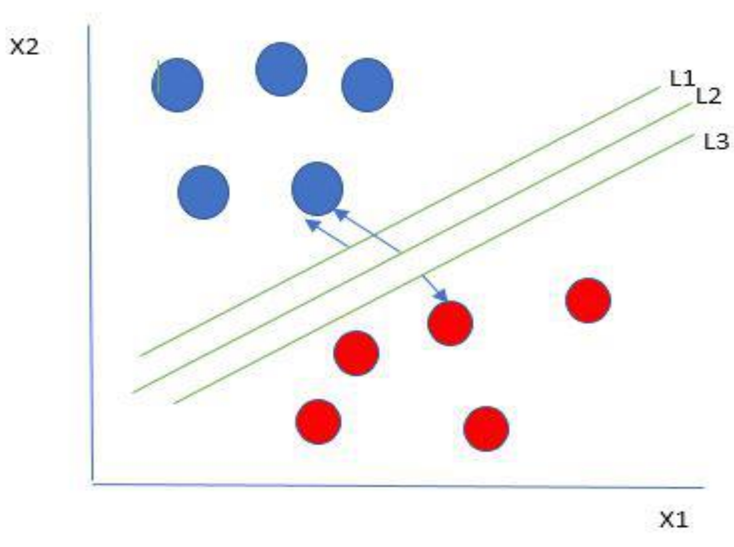



Figure 2.2: Multiple Hyperplanes Separate the data from two classes (Geeksforgeeks, 2025a)

Let's look at a binary classification problem where there are two classes, +1 and -1. The input feature vectors X and the class labels Y that correspond to them make up our training dataset.

## 2.3 Review of Related Works

Tatsumi and Usami (2024), developed a machine-learning plus feature-selection UAV-derived system trained on experimental field data collected by the authors at Tokyo University of Agriculture and Technology, using UAV RGB and multispectral imagery, to predict plant-level underground potato yield by identifying critical UAV-derived plant height and vegetation index variables. The study used five feature-selection algorithms to identify influential variables and refine top variable sets before ML prediction (Random Forest, Ridge Regression, Support Vector Machine). The results indicate that random forest outperforms the others, achieving $R^2 = 0.57$ (FTW), 0.45 (NMT), and 0.49 (FWT) with the best feature selection method.

Wang et al. (2024) conducted a review on deep learning methods for crop yield prediction. The study was motivated by the fact that traditional ML was described as less accurate, and prior literature had not sufficiently analysed the characteristics and limitations of different DL model types themselves, so the study compared model types, discussed crop-specific applications, and suggested future research directions. Findings from the study indicate challenges in data acquisition/quality, regional heterogeneity, the influence of natural conditions, costly training, difficulty in model selection, and poor interpretability of DL.

Li et al. (2025) proposed a novel deep learning framework (CNNAtBiGRU) for maize yield estimation, which aims to improve real-time prediction accuracy through the integration of spatiotemporal data and anthropogenic factors such as mechanisation. Furthermore, the work compared CNNAtBiGRU to CNNGRU, GRU, LightGBM, and LASSO, as well as with five other yield-estimation methods, to determine the best for the maize yield estimation. The result of the model evaluation shows that the CNNAtBiGRU achieved $R^2$ of 0.896 and RMSE of 908.33 kg/ha. Wei et al. (2022) applied environmental data, such as weather and soil data like temperature, rainfall and soil properties, in order to train a Random Forest (RF) regression that could help predict crop yield. Their results showed that Random Forest outperformed more traditional models like linear regression and SVM in terms of prediction accuracy, achieving a strong $R^2$ score of 0.85. However, since the study focused on a specific region and crop type, its findings may not serve as a basis for deployment in Sub-Saharan Africa, especially Nigeria.

Kumar et al. (2022) employed an ML technique, Support Vector Machines (SVM), in order to forecast wheat yield, in which the study factored variables such as temperature, rainfall and soil pH. The SVM model was effective at capturing complex, non-linear relationships between climate variables and yield, to the extent that it achieved an impressive accuracy of 89%. However, there is still a regional problem as the model was trained with wheat data from a single region, making its broader application uncertain.

In an attempt to estimate maize yield, Sharma et al. (2022) used Decision Tree Regression considering factors such as temperature, humidity and precipitation. The model had a good performance, with an RMSE value of 1.2 and an $R^2$ value of 0.81. The study they reported on underlined the importance of rainfall during the flowering stage of maize. When these data gaps were not corrected, the reliability of the model was hindered, however.

Wang et al. (2024) proposed a multi-model method combining machine learning models and remote sensing data in order to enhance the accuracy of yield prediction. They were able to predict the NDVI with an impressive accuracy of 92% using satellite indices and models such as the neural network and the random forests. However, they noted the significant computational needs and reliance on quality satellite data, which could pose barriers for implementation in lower-resource environments.

Liu et al. (2020) predicted rice yield using the artificial neural network (ANN) with both weather and soil data. The model was accurate with an $R^2$ of 0.87 and an RMSE of 1.1. ANNs proved to be effective, but their black-box nature resulted in difficulties in interpretation and the requirement of substantial volumes of data was a challenge in data-limited settings.

Zhang et al. (2020) proposed a hybrid model which combines the weather model and soil model for yield prediction. The model yielded good performance with $R^2$ of 0.93 and the RMSE was 0.9. This method has the advantage of using multiple data sources, but it may be difficult to scale due to the complexity and increased resource usage.

Sun et al. (2024) investigated the use of deep learning techniques such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for crop yield prediction. In terms of their accuracy, the models performed well, especially the CNNs, which achieved an accuracy of 94%. The RNNs also had a low mean squared error (MSE) of 0.07, which indicates their accuracy in capturing the time-related patterns. However, the study noted that deep learning requires a lot of computing resources and so is not easily accessible for those with less resources.

Gupta et al. (2023) focused on weather-based crop yield prediction, using data like temperature, rainfall and solar radiation. Their model gave a 84% accuracy and they mentioned the significance of rainfall during flowering stage. The simplicity of the model enabled easy access for smallholder farmers but the lack of soil data limited the model's accuracy in degraded soil areas.

Patel et al. (2020) developed a model combining climate and soil health data, with a sustainable agriculture perspective. Their model was accurate with an $R^2$ value of 0.91 and a MAPE value of

7%. Adding soil health greatly enhanced predictions, but limited variety of crops used in testing questions the model's applicability to other farming situations.

Singh et al. (2021) applied some machine learning models, including Random Forest, XGBoost and SVM, to compare them by using various weather parameters, satellite images and soil sensors. XGBoost came out on top with an impressive 95% accuracy. Multiple data types have helped the prediction, but it is a resource-intensive process to collect and integrate multiple data types.

Patil et al. (2020) used Deep Neural Networks (DNNs) for predicting crop yields based on past weather and soil data. The model performed well with an accuracy of 93% and the RMSE value was 1.0, which is better than those of the older statistical methods. Although DNNs are capable, they require significant amount of data and complexity and are not transparent enough to be used in small scale or resource-constrained farming operations.

Tan et al. (2022) investigated the enhancement of the crop yield prediction using the ensemble learning techniques bagging and BOOSTing. Their method showed high accuracy with an $R^2$ value of 0.90 and RMSE value of 0.8, by combining the strengths of multiple models. The outcomes showed that ensemble techniques are helpful in the reduction of overfitting and enhance the model's capacity to generalise among various datasets. The study also found that these improvements are not without their drawbacks, though, as ensemble models tend to be more complex and require more computational power, which can be a problem in low-resource agricultural environments.

Chen et al. (2021) adopted a different strategy that involved using Internet of Things (IoT) sensors and machine learning algorithms for providing real-time crop yield predictions. The system was able to monitor environmental conditions, such as soil moisture, humidity, and temperature, continuously and achieved a remarkable accuracy in predicting the results, of 92%. This enabled real-time adjustments to be made to provide predictions. Although the study showed good results, it admitted that there were a few flaws such as the deployment cost of IoT infrastructure and the difficulty of managing and processing the massive flow of data generated by sensors.

## Summary of Literatures

| Author(s) | Work Done | Technique Applied | Results | Research Gap |
|---|---|---|---|---|
| Tatsumi and Usami (2024) | Predict plant-level underground potato yield by identifying critical UAV-derived plant height and vegetation index variables. | Random Forest, Ridge Regression, Support Vector Machine | Random forest overall best with $R^2 = 0.57$ | Generalization problems and real time deployment or interface |
| Wang et al. (2024) | a review on deep learning methods for crop yield prediction | Literature Review | challenges in data acquisition/quality, regional heterogeneity | Survey paper and practical experimentation. |
| Li et al. (2025) | Maize yield estimation to improve real-time prediction | Hybrid: CNNAtBiGRU | $R^2 = 0.896$ | No real-time deployment; High computational cost |

| | | | | |
|---|---|---|---|---|
| | accuracy by integrating spatiotemporal data and anthropogenic factors. | | | |
| Wei et al. (2022) | Predicted crop yield using weather and soil data | Random Forest Regression | $R^2 = 0.85$ | Limited to a specific region and crop type |
| Kumar et al. (2022) | Predicted wheat yield based on weather and soil pH | Support Vector Machines | Accuracy = 89% | Only applicable to wheat; lacks generality across crops/regions |
| Sharma et al. (2022) | Predicted maize yield using climatic variables | Decision Tree Regression | $R^2 = 0.81$, RMSE = 1.2 | Sensitive to missing data, which impacts performance |
| Wang et al. (2024) | Combined ML with remote sensing for improved yield prediction | Multi-model: Neural Networks, Random Forests | Accuracy = 92% | High computational cost and satellite data dependency |

| | | | | |
|---|---|---|---|---|
| Liu et al. (2020) | Forecasted rice yield using weather and soil data | Artificial Neural Networks | $R^2 = 0.87$, RMSE = 1.1 | Lacks interpretability; requires large datasets |
| Zhang et al. (2020) | Proposed a hybrid model using weather and soil data | Hybrid ML Model | $R^2 = 0.93$, RMSE = 0.9 | Complexity and high cost of implementation |
| Sun et al. (2024) | Used CNN and RNN for crop yield forecasting | Deep Learning (CNN, RNN) | CNN Accuracy = 94%, RNN MSE = 0.07 | High computational resource requirement |
| Gupta et al. (2023) | Weather-based model focused on climatic data | Weather-based Regression Model | Accuracy = 84% | Minimal use of soil data reduces accuracy in poor soil conditions |
| Patel et al. (2020) | Combined climate and soil health data | Integrated Model with Sustainability Focus | $R^2 = 0.91$, MAPE = 7% | Limited crop diversity in testing |
| Singh et al. (2021) | Compared RF, XGBoost and SVM with | XGBoost outperformed others | Accuracy = 95% | Data integration is computationally |

| | | | | |
|---|---|---|---|---|
| | weather, imagery and soil sensor data | | | expensive and challenging |
| Patil et al. (2020) | Predicted crop yield using weather and soil history | Deep Neural Networks | Accuracy = 93%, RMSE = 1.0 | Lack of interpretability; needs large datasets |
| Tan et al. (2022) | Used bagging and boosting for ensemble learning in yield prediction | Ensemble Learning (Bagging, Boosting) | $R^2$ = 0.90, RMSE = 0.8 | Higher computational cost and model complexity |

## 2.4 Research Gap

Having reviewed different studies related to crop yield estimation, which employed different traditional machine learning and emerging AI approaches, a common and significant challenge identified across the reviewed literature is the issue of data availability and quality, model generalisation problems, and a lack of real-time deployment or interface.

Many researchers have pointed out that dealing with incomplete, noisy, or missing data can severely affect the accuracy and reliability of crop yield prediction models. These findings underline a broader issue in the field: the lack of high-quality, comprehensive datasets makes it difficult to develop models that are both robust and generalisable across different regions and crop types. Reliance on structured data on weather and soil factors alone cannot determine crop yield accurately. Therefore, the study proposes to address these challenges by developing a vision-based crop yield prediction system based on imagery data.

# CHAPTER THREE

# SYSTEM ANALYSIS AND DESIGN

## 3.1 Introduction

This chapter presents the methodology that was adopted for effectively developing the vision-based multimodal crop yield estimation system with a focus on cassava and maize cash crops. It explains the research design, data collection procedures, preprocessing techniques, model development process, system architecture, and implementation procedures which were used to carry out the study. The proposed system combines Convolutional Neural Network (CNN) techniques for crop disease classification along with XGBoost algorithms in order to predict crop yield using environmental and agricultural variables. The method chosen was designed to support accurate crop health assessment and predictive yield estimation within the Nigerian agricultural context.

## 3.2 Research Design Methodology

In this study, a hybrid Object-Oriented Design (OOD) and Experimental Deep Learning-based methodology is adopted for the development of a vision-based crop yield estimation system. The choice of OOD ensures that the system is modular, scalable, and easy to maintain, while deep learning enables the extraction of complex patterns from image data for accurate yield prediction. Unlike traditional systems that rely heavily on weather and soil parameters, this research focuses on image-driven intelligence, where crop images serve as the primary data source. The experimental deep learning-based approach enabled the training, testing, and validation of machine learning and deep learning models using real agricultural datasets. The study combined image-based analysis and numerical data modelling within a multimodal prediction framework, where the CNN architecture was employed in order to analyse crop leaf images and classify disease conditions in cassava and maize crops, while the XGBoost regression techniques were utilised in order to predict crop yield using agricultural production variables, which we obtained from FAOSTAT and climatic variables obtained from NASA databases. The study followed a step-by-step development process, which involves dataset acquisition, preprocessing, model training, model evaluation, system integration, interface implementation, and result analysis accordingly. According to the stipulated methodology and in the development environment, TensorFlow datasets were generated after preprocessing in order to effectively improve training efficiency and ensure compatibility with CNN architectures. Overall, the research design supported both predictive analysis and practical web-based deployment of the proposed system.

The system is designed using OOD principles and divided into key components such as: ImageCollector, ImagePreprocessor, CropDetector, FeatureExtractor (DL-based), YieldPredictor, and ResultVisualizer. The core of my system is a deep learning-based prediction module, in which we trained an advanced model and, in the same vein, optimised it in order to estimate crop yield from visual features effectively. The study evaluates the model performance across epochs using metrics such as accuracy, precision, recall, F1-score, RMSE, and MAE. Finally, this structured methodology will ensure that the system is not only accurate but also flexible enough in order to accommodate future improvements, such as the integration with IoT-based smart farming systems and the incorporation of XAI and GANs.

### 3.2.1 Study Area

For this study, Nigeria was chosen as the study area because agriculture remains one of the major contributors to food production, employment, and economic sustainability within our country and with the recent security situation happening, especially in the northern part, which has the most arable lands, it becomes paramount for us to optimise the agricultural process and also be able to predict farm produce in order to enable proper planning. Cassava and maize are among the most cultivated crops in Nigeria, and they are also widely planted and consumed across different regions of the country. Additionally, important features for yield estimation, including rainfall, temperature, humidity, and environmental conditions, vary across different regions in Nigeria and also significantly affect crop productivity and disease occurrence in Nigerian farmlands, therefore stressing the importance of its integration into the system. Furthermore, the increasing demand for intelligent agricultural systems in Nigeria motivated the development of the proposed prediction framework, which focuses on improving crop monitoring and yield estimation in order to effectively support Nigerian farmers, agricultural researchers, and policymakers.

## 3.3 System Analysis

This section evaluates existing crop yield estimation systems and highlights their limitations, followed by the design of a more efficient vision-based solution.

### 3.3.1 Analysis of Existing System

The current agricultural prediction systems are mostly based on manual observation systems, requiring physical visits to the farm, visual assessment and expert opinion, which are time-consuming, subjective and inconsistent. In addition, most of the traditional crop yield estimation systems are based on tabular data such as weather conditions, soil properties and past production history. For instance, in decision tree regression systems, the main variables used to predict yield are environmental variables. These systems offer moderate performance, but have drawbacks like relying heavily on structured data, which can be incomplete and unreliable, they cannot measure crop visual traits like plant density and crop condition, and manual feature engineering is needed. Upon studying this system of Sharma et al. (2022), they have implemented the Decision Tree Regression with weather and soil data to predict the maize yield. Some of the time it did okay, but there were some problems, like having a value missing or noisy data, that really confused their predictions. Another drawback to their model was that they could overfit, particularly if the data wasn't complete. Moreover, their system didn't employ advanced feature selection and preprocessing steps that might have impacted their accuracy.

However, recent studies have demonstrated that a significant increase in the accuracy of the predictions can be obtained by using image-based approaches that involve integrating the use of UAV and deep learning, indicating that the newer methods such as the application of deep learning or ensemble methods may yield better results, particularly when incorporating images (Tatsumi & Usami, 2024; Li et al., 2025).

### 3.3.2 Weaknesses of the Existing System

The major limitations of existing systems include:

1. **Poor Data Representation**: These traditional models ignore visual crop features.
2. **Lack of Integration**: The existing system didn’t integrate the crop disease classification and yield estimation processes
3. **Manual Feature Extraction**: Existing systems require human intervention
4. **Limited Accuracy**: The existing system cannot capture complex spatial patterns
5. **Overfitting Issues**: There are issues of overfitting, especially with small or noisy datasets

6. **Scalability Problems**: The existing systems are difficult to integrate with real-time imaging systems

### 3.3.3 Analysis of the Proposed System

The proposed system is a vision-based multimodal crop yield estimation framework which is designed to combine disease classification and crop yield prediction in a unified intelligent platform, which comprises CNN-based image classification along with XGBoost-based yield prediction in order to improve yield prediction for farmers and, in turn, improve agricultural decision-making. The working principle of the system starts with the crop leaf images being analysed using CNN algorithms in order to identify healthy and diseased crop conditions. This will improve existing approaches by shifting from traditional data-driven methods to a vision-based learning framework, and will eliminate the need for manual feature engineering by leveraging deep learning for automatic feature extraction. Additionally, the system introduces a density-aware regression mechanism, in which crop density information is learned implicitly from image features rather than explicitly computed, thereby reducing system complexity while maintaining predictive performance.

Through the use of modular design methods that incorporate Object-Oriented principles, every part of the system can be independently enhanced/replaced. This provides increased flexibility and accommodates further integration into future systems that provide real-time agricultural data. Figure 3.1 is a block diagram of the proposed system for crop yield determination using this vision-based technique. The process starts with obtaining high-resolution crop images from an agricultural site, which are pre-processed by resizing/normalising to reduce the impact of lighting differences across images.

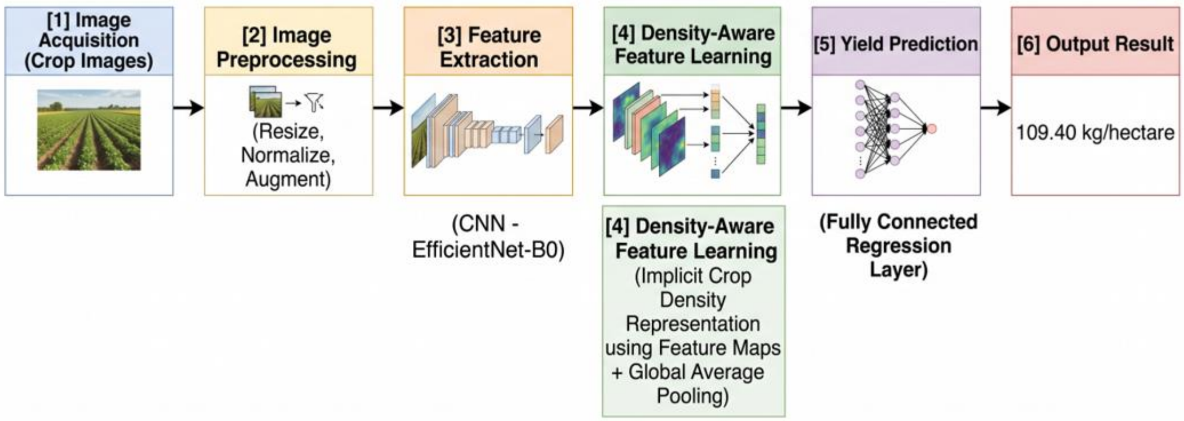



Figure 3.1: Block Diagram of the Proposed System

These images are then passed to a convolutional neural network, specifically EfficientNet-B0, which serves as the system's visual engine that is responsible for extracting key features such as leaf texture, colour variation, and canopy coverage in order to predict its health status. Furthermore, the model introduces a density-aware learning mechanism such that spatial feature maps are generated in order to capture the distribution and concentration of vegetation across the field. This information is summarised using global average pooling, which will represent overall crop density. Finally, the learned features are fed into a regression layer, specifically XGBoost, that estimates the crop yield, with techniques such as dropout applied to improve generalisation and prevent overfitting, resulting in a reliable prediction of yield values.

### 3.3.4 Advantages of the Proposed System

The new vision-based crop yield estimation system offers several significant advantages over traditional approaches, as it will leverage deep learning techniques in order to achieve improved prediction accuracy. It does this using its ability to learn complex visual patterns directly from crop images. The proposed system will support automated crop detection and analysis, which reduces the need for manual intervention and minimises human effort in the estimation process.

Furthermore, the model is capable of handling complex and nonlinear relationships inherent in agricultural data, which conventional methods often fail to capture effectively when they are predicting crop yield.

Another advantage of this proposed system is that it is designed using Object-Oriented principles, which makes it modular, scalable, and easy to maintain or extend for future improvements. Also, it is computationally efficient and has a structured workflow, which makes it suitable for real-time or near-real-time applications, especially when integrated with imaging devices such as drones or mobile platforms. Overall, the system aligns well with the goals of precision agriculture by enabling intelligent, data-driven decision-making and enhancing productivity in modern farming environments.

**3.3.5 Model Architecture**

The proposed CNN model architecture consists of convolutional layers, pooling layers, activation layers, flattening layers, and dense layers, which are used to carry out the feature extraction task. The convolutional layers were used to extract relevant features from crop leaf images, pooling layers reduced feature dimensionality and improved computational efficiency, while the dense layers performed final classification of crop disease categories. The TensorFlow and Keras libraries were used to implement the CNN architecture. On the other hand, the XGBoost algorithms were implemented for crop yield prediction using environmental and agricultural datasets. This final integrated architecture enables simultaneous crop disease classification and predictive yield estimation for the system, demonstrating how effective the combined model structure improved the system's ability to analyse crop conditions comprehensively.

Figure 3.2 presents the model architecture diagram of the vision-based crop yield estimation system.

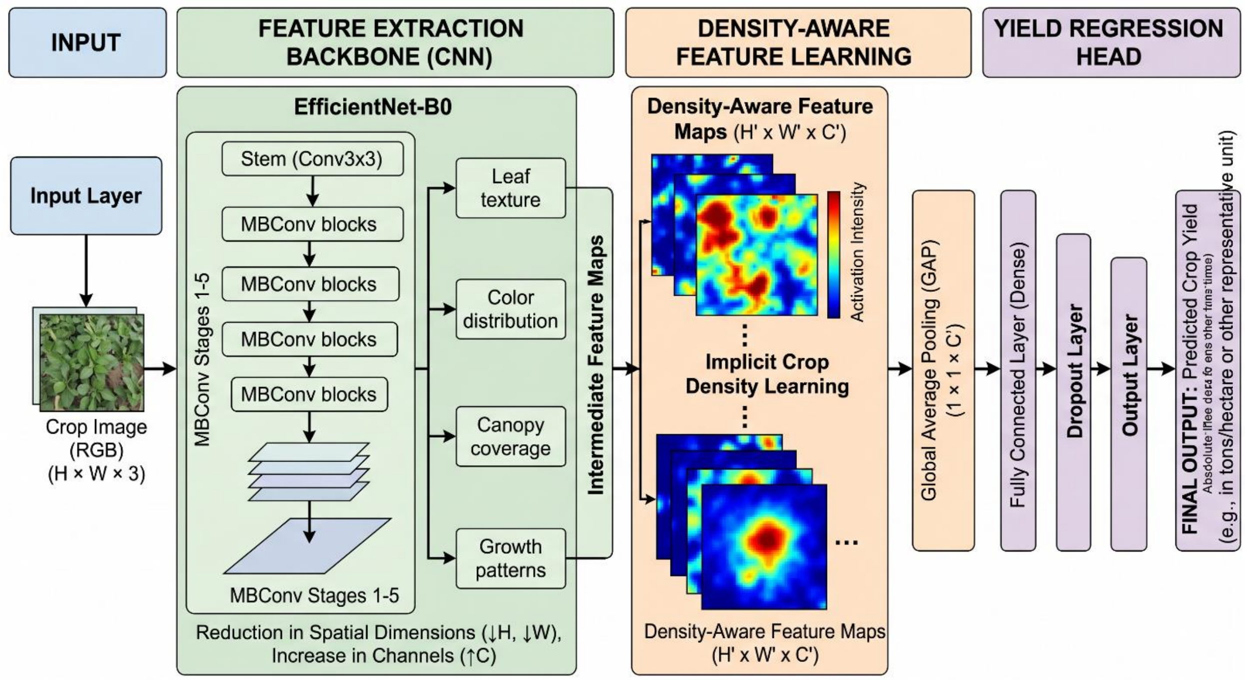



Figure 3.2: model architecture diagram

Rather than relying on explicit crop-counting methods, the model introduces a density-aware learning mechanism in which plant density is implicitly encoded in the spatial feature maps generated by the CNN. The model network will learn representations that reflect crop density and health conditions through the activation patterns and global average pooling features of the model architecture, which form the core novelty of the adopted approach for this crop yield estimation project. These extracted features are then passed to a regression head comprising fully connected layers with dropout regularisation to reduce overfitting, ultimately producing a continuous yield estimate.

### 3.3.6 Requirement Analysis

### A. Functional Requirements

i. The system should enable users to sign up for an account and be able to log in securely to the platform.

ii. The system should support the upload of cassava and maize leaf images for disease analysis. Before classification, the system should preprocess the images that have been uploaded.

iii. The system should use the CNN model to classify crop diseases and healthy leaf conditions.

iv. The system should be able to handle agricultural and climate parameters to predict yields.

v. The XGBoost model should make yield prediction using the integrated datasets from environmental factors.

vi. The system should be able to present its prediction results in an interactive user interface.

vii. The system shall be able to store information on prediction activities and previous analyses.

viii. The administrator should have control over datasets and be able to keep an eye on what is happening in the system.

ix. The system should produce prediction reports for the users.

**B. Non-Functional Requirements**

i. The system should provide a fast response time during prediction operations.

ii. The system should be able to maintain stable prediction performance during repeated usage.

iii. The web application should be user-friendly and will be easy to navigate.

iv. The system should ensure data confidentiality and secure user authentication.

v. The application should support scalability for future integration of additional crops.

vi. The system should maintain high reliability during continuous operation.

vii. The prediction framework should provide acceptable accuracy levels for agricultural analysis.

viii. The application should support compatibility across modern web browsers and devices.

## 3.4 Data Collection and Description

For this study, two major crop image datasets were collected from Kaggle for cassava and maize disease classification, and within the cassava dataset are five disease and healthy classes, which are suitable for CNN experimentation, while the maize dataset includes four disease and healthy classes used for disease classification training. The downloaded datasets were organised into structured project directories in order to simplify preprocessing and model training activities, with image samples being inspected in order to ensure suitability for deep learning experimentation and classification tasks.

The agricultural yield variables were obtained from FAOSTAT in order to support crop yield prediction processes, while the climatic and environmental variables were collected from NASA databases to improve environmental analysis within the proposed system prediction framework. The combined datasets formed the multimodal input structure used for our proposed system development and evaluation. The cassava dataset consisted of healthy and diseased cassava leaf images representing five different classification categories, while the maize dataset consisted of healthy and diseased maize leaf images representing four classification categories, and all were RGB images captured under different environmental and lighting conditions. The agricultural variables obtained from FAOSTAT include crop yield-related information used during predictive analysis, while the climatic variables obtained from NASA include rainfall, humidity, and temperature measurements associated with agricultural productivity. Overall, the collected datasets contained both categorical and numerical variables, which are very much suitable for multimodal learning processes. The data source table is present in Table 3.1, while Table 3.2 presents the input data description.

**Table 3.1: Data Properties and Description**

| Property | Source | Description |
|---|---|---|
| Cassava Leaf Images | Kaggle | Disease Classification |
| Maize Leaf Images | Kaggle | Disease Classification |
| Agricultural Yield Variables | FAOSTAT | Yield Prediction |
| Climate Variables | NASA | Environmental Analysis |

**Table 3.2 Description of Input Data Variables**

| Variable | Description |
|---|---|
| Rainfall | Measures precipitation level |
| Temperature | Measures atmospheric temperature |
| Humidity | Measures atmospheric moisture |
| Crop Type | Indicates cassava or maize |
| Disease Label | Indicates crop disease category |
| Yield Value | Represents agricultural yield output |

### 3.4.1 Data Preprocessing

Data preprocessing is an important stage in the development of the proposed vision-based crop yield estimation system because it will prepare our acquired crop image datasets so that they can achieve effective machine learning and deep learning operations. In this study, labelled cassava and maize leaf image datasets were collected from Kaggle before they were organised into structured class folders based on disease categories using Python programming codes. The preprocessing stage was implemented using Python within a Jupyter Notebook environment, while libraries such as OpenCV, NumPy, TensorFlow, and Matplotlib were installed in the development environment and then utilised for image processing and analysis.

During preprocessing, all images were loaded from their respective dataset directories and converted into a standard format which is suitable for model training. However, the original images, which we collected, possessed different dimensions and colour structures; we had to resize each image to a uniform resolution of 128 × 128 pixels in order to ensure consistency during convolutional neural network processing. The images were also converted from the default BGR colour representation used by OpenCV into RGB format in order to obtain an accurate visual interpretation and processing. Furthermore, pixel values were normalised from the range of 0–255 to 0–1 in order to improve computational efficiency, which in turn enhances model convergence during training. Class labels corresponding to the crop disease categories were equally generated and encoded to support supervised learning operations within the proposed vision-based crop yield estimation system. Table 3.3 presents a sample of the pre-processed dataset used as model input.

**Table 3.3 Sample of Preprocessed Dataset Used as Model Input**

| **Crop Type** | **Temperature** | **Rainfall** | **Humidity** | **Disease Label** | **Yield** |
|---|---|---|---|---|---|
| Cassava | 29.4 | 120.3 | 75 | Healthy | 4.8 |
| Maize | 27.1 | 98.6 | 68 | Diseased | 3.9 |

## 3.5 System Design Methodology

The study adopted the Object-Oriented Analysis and Design (OOAD) methodology for modelling and designing the proposed system, which was selected because it supports modularity, reusability, scalability, and efficient system representation. The methodology enabled the system requirements, processes, objects, and user interactions to be modelled effectively before implementation. And the Unified Modelling Language (UML) tools were utilised in order to effectively visually represent the structural and behavioural components of the proposed system.

### 3.5.1 System Modelling Using UML

The structure and the behaviour of the system are represented visually using Unified Modelling Language (UML) diagrams.

### 3.4.1 Use Case Diagram

The use case diagram shows how the interaction between the farmer and the vision-based crop yield estimation system works. Figure 3.3 presents the use case diagram of the vision-based crop yield prediction system.

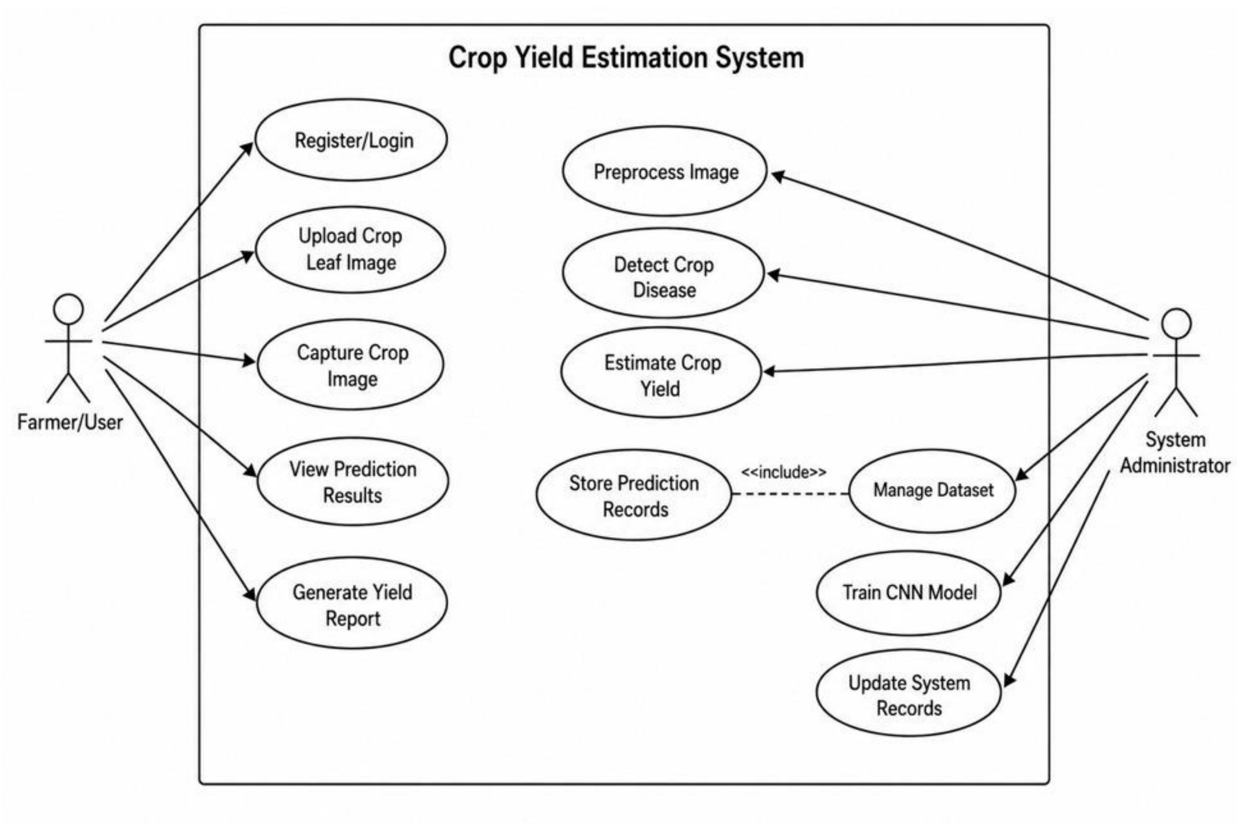



Figure 3.3 Use case diagram

Figure 3.3, which is the use case diagram, illustrates interactions between users, administrators, and the prediction system. The major use cases for the Vision-based crop yield estimation system are image upload, disease prediction, yield estimation, result viewing, and dataset management.

**B. Activity diagram**

Activity diagrams could be referred to as the best tools that could be employed for visualising the flow of activities or processes within a system. Figure 3.4 illustrates an activity diagram outlining the steps that are involved in implementing the vision-based crop yield estimation system :

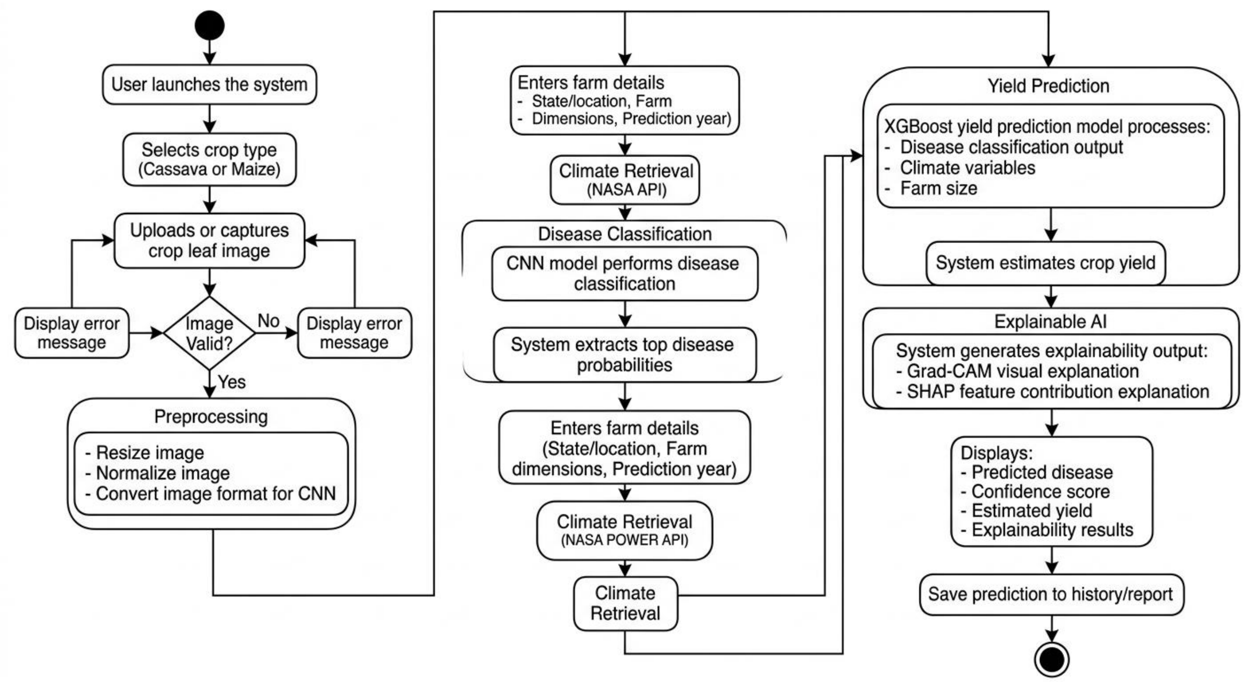



Figure 3.4: Activity diagram of the proposed system.

The activity diagram of the vision-based crop yield estimation system presented in Figure 3.4 represents the operational workflow of the system from image upload to prediction output. The workflow includes preprocessing, classification, prediction, and result display processes.

**C. Class Diagram of the Proposed System**

This section of this work of research holds the class diagram of the proposed system. The class diagram in Unified Modelling Language (UML) is one of the Unified Modelling Language diagrams that describes the architectural design of a software system by completely defining and laying out its classes, the attributes of the laid-out classes, the methods, and the relationships among these different objects that make up a system. Figure 3.8 illustrates the class diagram for the proposed system

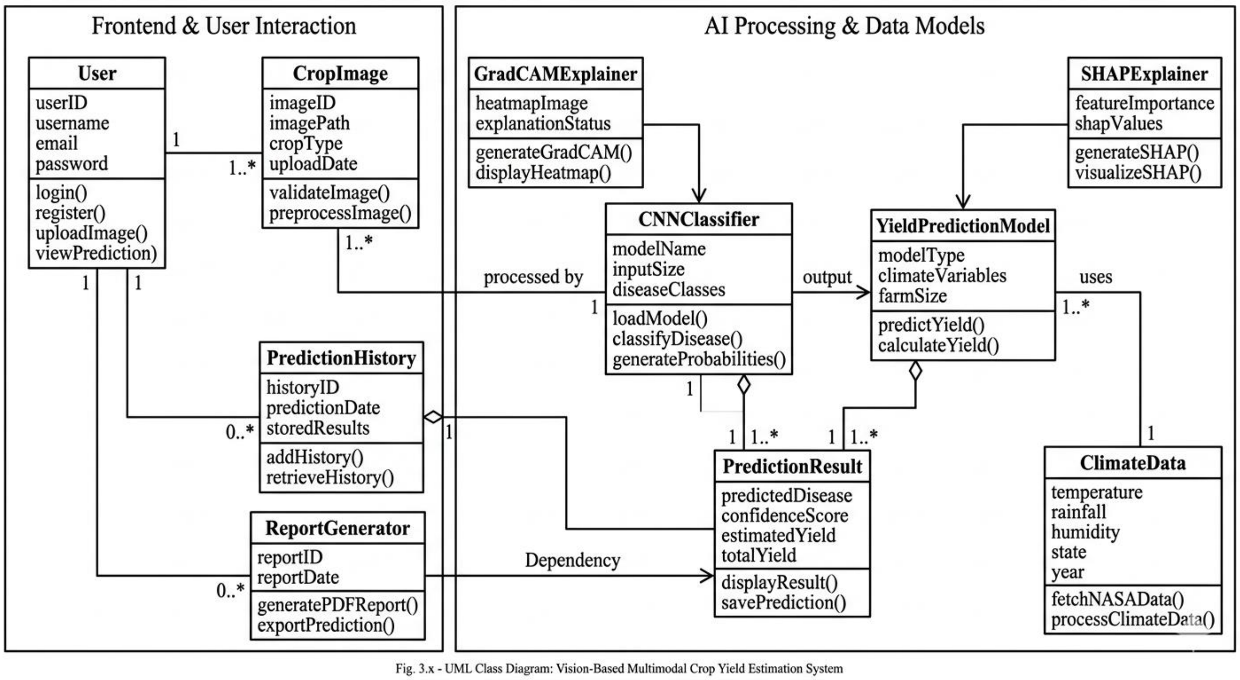



Figure 3.5 Class diagram of the proposed system

Figure 3.5 illustrates relationships between major system classes, such as the User, the CNN Classifier, the XGBoost Yield Prediction Model, the PredictionResult and the SHAP, along with the GRADCAM explainer. The class diagram shows the different classes that would be involved in the proposed system and their relationship with one another. It shows the way in which the system functions by outlining the specific attributes and methods that are present in the classes that are available in the system that is proposed in this work of research.

### 3.5.2 Database design

This design of a database is a crude depiction of how the database is going to appear when created. It shows the different tables that would be used for storing data within the database. However, the essence of the database design is to show a modelled view of the proposed system’s database. The proposed system’s database is a relational database designed and implemented using the MySQL query language. The database was designed to have entities and at least one attribute for each entity.

**Table 3.4 User Table for the Proposed Vision-Based Crop Yield Prediction System**

| **Attribute** | **Data Type** | **Description** |
|---|---|---|
| user_id | Int | The primary key uniquely identifies each user. |
| full_name | Varchar | Stores the user's full name. |
| username | Varchar | Stores the user’s login username. |
| email | Varchar | Stores the user’s email address. |
| password | Varchar | Stores the encrypted user password. |
| user_type | Varchar | Identifies user category such as farmer, researcher, or admin. |
| date_registered | Timestamp | Stores the user's registration date and time. |

**Table 3.5 Climate and Farm Data Table for the Proposed System**

| **Attribute** | **Data Type** | **Description** |
|---|---|---|
| climate_id | Int | Primary key uniquely identifying each climate record. |
| user_id | Int | Foreign key identifying the user. |
| state | Varchar | Stores the selected Nigerian state/location. |
| year | Int | Stores the prediction year. |
| temperature_c | Float | Stores average temperature value in degrees Celsius. |
| rainfall_mm | Float | Stores rainfall value in millimetres. |

| | | |
|---|---|---|
| humidity | Float | Stores average humidity percentage. |
| farm_length_m | Float | Stores farm length in metres. |
| farm_width_m | Float | Stores farm width in metres. |
| farm_area_hectares | Float | Stores calculated farm area in hectares. |
| climate_source | Varchar | Stores the climate data source, such as NASA POWER API. |
| date_recorded | Timestamp | Stores the date and time the climate data was retrieved. |

**Table 3.6 Model Parameter Table**

| **Attribute** | **Data Type** | **Description** |
|---|---|---|
| model_id | Int | Primary key uniquely identifying the AI model record. |
| model_name | Varchar | Stores the name of the AI model used. |
| cnn_model | Varchar | Stores CNN model information for disease classification. |
| xgboost_model | Varchar | Stores XGBoost model information for yield prediction. |
| gradcam_status | Varchar | Stores Grad-CAM explainability status. |
| shap_status | Varchar | Stores SHAP explainability status. |
| accuracy_score | Float | Stores model accuracy value. |
| r2_score | Float | Stores regression $R^2$ score value. |
| mae_score | Float | Stores Mean Absolute Error value. |
| rmse_score | Float | Stores Root Mean Squared Error value. |

| | | |
|---|---|---|
| created_at | Timestamp | Stores the date and time model information was recorded. |

**Table 3.7 Prediction History Table**

| **Attribute** | **Data Type** | **Description** |
|---|---|---|
| prediction_id | Int | Primary key uniquely identifying each prediction record. |
| user_id | Int | Foreign key identifying the user. |
| crop_type | Varchar | Stores the selected crop type such as cassava or maize. |
| predicted_disease | Varchar | Stores the predicted disease or crop condition. |
| confidence_score | Float | Stores the CNN prediction confidence score. |
| estimated_yield_tpha | Float | Stores estimated yield in tonnes per hectare. |
| total_estimated_yield | Float | Stores total estimated farm yield. |
| prediction_status | Varchar | Stores prediction status such as accepted or uncertain. |
| prediction_date | Timestamp | Stores the prediction date and time. |
| explainability_status | Varchar | Stores explainability module status. |
| report_path | Varchar | Stores generated prediction report location or reference. |

### 3.6 Model API Design

This section explains how the user input reaches the AI model and returns a prediction output using the Application Programming Interface. The proposed vision-based crop yield prediction system requires a middleware interface in order to connect the trained hybrid deep learning model with the web-based user interface. The model API will help in achieving real-time flood-risk prediction as it is designed to receive environmental input parameters, preprocess the data, execute inference using the trained CNN and XGBoost model, and return prediction outputs to the web application dashboard.

The API architecture is made up of components such as a web frontend, which serves as the user interaction interface, collecting input data and sending prediction requests to the API endpoint, which acts as the communication bridge between the interface and the backend system. Also, the validation and preprocessing modules of the API ensure that user inputs are accurate, secure, and transformed into the format required which is required by the trained prediction model, and such inputs include the rainfall intensity, rainfall duration, humidity, temperature, drainage condition, elevation, land use indicators, and water level. Subsequently, the input parameters the API receives are validated and processed, followed by model inference and flood probability, before warning classification and the JSON response.

**3.7 Web Application Design**

The web application was designed in order to provide a user-friendly and interactive interface for the crop yield estimation system, which users would operate seamlessly. The interface allows users to upload crop images and enter prediction-related information so that the inference engine (model) will preprocess the user's input so that the prediction results are displayed in an organised and understandable format. For this study, a responsive web design principle was implemented in order to improve accessibility across different devices and screen sizes. The application integrates backend machine learning models with frontend user interaction components.

# CHAPTER FOUR

# SYSTEM IMPLEMENTATION, RESULTS AND DISCUSSION

## 4.1 Introduction

System Implementation refers to the part where the software is brought to life. This chapter describes the implementation of the system, along with the user interface and tools used in the system. This focuses on showing how the implementation was done in the development of the system following the system analysis and design that was carried out.

## 4.2 Development Environment and Tools

This study was implemented within Jupyter Notebook and Visual Studio Code  VSCode as the primary deep learning experimentation environments, because of its support for multiple programming languages and rich extensions marketplace. Dataset preparation and preprocessing were done on VSCode using Python, while model development and training were accelerated using an NVIDIA A100 GPU in order to significantly reduce the computational time. Overall, the implementation environment utilised Python as the core programming language because it contains an extensive ecosystem for machine learning and scientific computing. 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.

### 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 Flood-Risk prediction and early warning guide system will employ a deep learning technique that 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 deep learning model training, Keras for simplifying neural network model development, Pandas for structured data handling, and the NumPy library for supporting numerical computations and array operations. In addition, Matplotlib and Seaborn were used for result visualisation and result plotting in accordance with the metrics considered.

### 4.2.2 Development Platform and Hardware Requirements

The system was implemented on a computer system with sufficient RAM and processing capability to support deep learning operations, and the GPU acceleration was utilised where available to improve CNN training speed. However, during the course of the training and implementation, adequate storage capacity was maintained for image dataset handling and model storage. 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 Jupyter, which was extensively used for 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

### 4.2 System Implementation

The proposed vision-based crop yield prediction system implementation process began with dataset preparation and preprocessing operations, in which the CNN models were trained using the processed cassava and maize image datasets. The TensorFlow datasets were generated in order to support efficient batch training operations, while the XGBoost models were trained using agricultural and climatic variables obtained from FAOSTAT and NASA. In conclusion, the trained models were integrated into a web-based application environment with the REST API, which connects the frontend and backend components of the software application system, such that the parameters entered by the user are stored temporarily before processing that prepares the input for model compatibility.

#### 4.2.1 Model Training Process

The study used multi-source data from multiple datasets, which were prepared and loaded into the deep learning training environment after the preprocessing step was completed in VSCode. The training dataset consists of Image data, climate data variables, and environmental observations, which were arranged into time-dependent sequences. The CNN training process involved image preprocessing, feature extraction, and disease classification stages, with the training and testing datasets being separated in order to evaluate model performance objectively. The cassava CNN model achieved approximately 70.7% test accuracy during disease classification experiments, while the maize fixed-split CNN model achieved approximately 81.17% test accuracy during evaluation. The CNN models provide a stable disease and healthy classification outputs during testing of the already trained model, while the XGBoost implementation produced strong regression performance during crop yield estimation, with results demonstrating values of MAE at 0.00988, RMSE at 0.024824, and $R^2$ Score at 0.999943. The low MAE and RMSE values generated by the XGBoost indicate minimal prediction error during yield estimation, while the high $R^2$ score demonstrates strong predictive capability between input variables and yield output.

**Table 4.1 Hyperparameter Configuration Table**

| **Hyperparameter** | **Configuration Used** | **Purpose** |
|---|---|---|
| Input image size | 128 × 128 × 3 | Standardised image dimensions for CNN processing |
| Model architecture | CNN–XGBoost Hybrid | Combines disease classification and yield prediction |
| Optimizer | Adam | Adaptive optimization during model training |
| Learning rate | 0.001 | Controls model weight updates |
| Loss function | Categorical Cross-Entropy | Measures CNN classification error |
| Batch size | 16 | Number of samples per training iteration |
| Epochs | 20–50 | Defines training duration |
| Dropout rate | 0.3 | Reduces model overfitting |
| Explainable AI | Grad-CAM and SHAP | Provides model interpretability |

| | | |
|---|---|---|
| Development framework | TensorFlow/Keras and XGBoost | Supports deep learning and regression implementation |

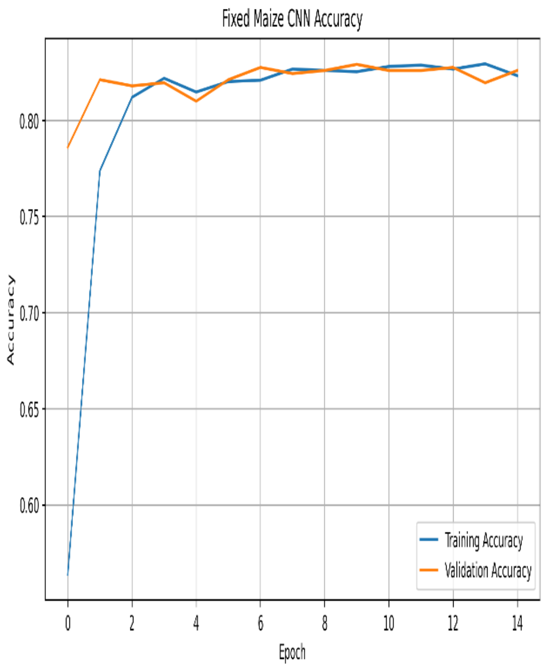
### 4.3 Results and Discussion

#### 4.3.1 Result of the CNN model for the Maize crop health status

The proposed model applies the convolutional feature extraction in order to effectively and efficiently classify the health status of the maize crop through its leaf images uploaded to the system. Table 4.2 presents the results of its initial performance after evaluation. Figures 4.1 and 4.2 show the accuracy and loss graph plot, while Figure 4.3 presents the confusion matrix.

**Table 4.2 Results of the hybrid model initial performance**

| **Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
|---|---|---|---|---|
| Blight | 0.61 | 0.96 | 0.74 | 173 |
| Common Rust | 0.96 | 0.92 | 0.94 | 197 |
| Gray Leaf Spot | 0.00 | 0.00 | 0.00 | 87 |
| Healthy | 0.97 | 0.95 | 0.96 | 175 |
| **Accuracy** | | | **0.81** | **632** |
| **Macro Average** | 0.64 | 0.71 | 0.66 | 632 |
| **Weighted Average** | 0.74 | 0.81 | 0.76 | 632 |



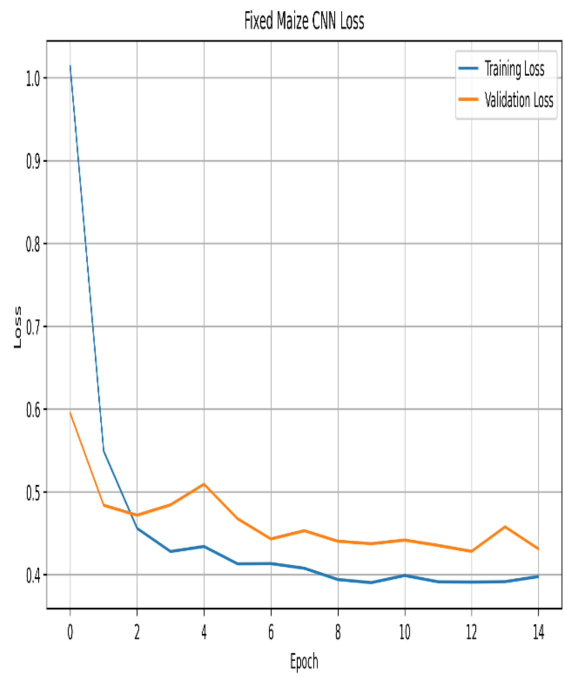




Figure 4.1: Plot of accuracy

Figure 4.2: Plot of Loss

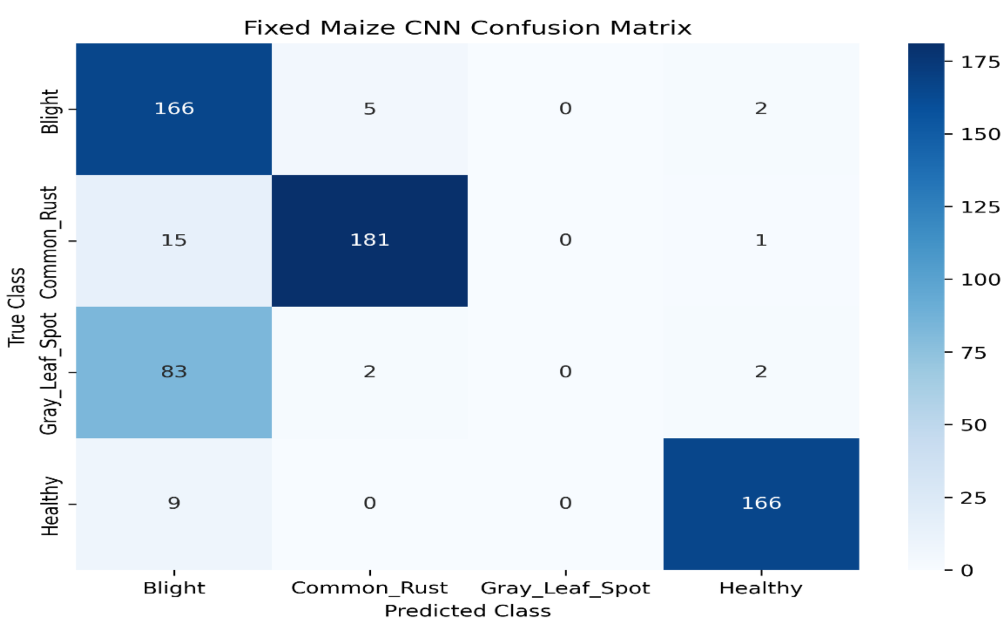



Figure 4.3: Confusion matrix

The results of Table 4.2 for the proposed model indicate its high performance in identifying Healthy maize leaves, which achieved 97% precision and 95% recall, meaning it has strong reliability in detecting healthy crops. Similarly, the Common Rust was also classified accurately, with a high F1-score of 0.94, showing that the model was highly effective in recognising this disease class, while in Blight detection, the model had a recall of 96%, meaning most blight-infected samples were successfully identified, although some false predictions still occurred. However, the performance of the model in the Gray Leaf Spot class was very poor, with precision, recall, and F1-score all close to 0.00, suggesting the model struggled to recognise this disease correctly.

On the other hand, Figures 4.1 and 4.2 show the model's accuracy and loss curves against the training epochs, respectively. From the graph, it can be seen that the training accuracy progressively increased throughout the training epochs as it approached the final training stages. The accuracy plot shows that the CNN model learned very quickly, with training accuracy increasing from about 56% to over 82% within a few epochs, which indicates effective feature learning from the maize leaf images.

Similarly, the validation accuracy remained very close to the training accuracy throughout the training process, suggesting that the model achieved good generalisation performance with minimal overfitting. While the loss plot demonstrates a steady reduction in both training and validation loss values, which means the model gradually improved its predictions and reduced classification errors as training progressed. Figure 4.3 shows the confusion matrix which reveals that the model performed strongly in detecting Blight, Common Rust, and Healthy maize leaves, as most samples from the dataset were correctly classified along the diagonal section of the matrix, but the model struggled significantly with Gray Leaf Spot, as most of its samples were incorrectly predicted as Blight, confirming the weak performance which was observed earlier in the classification report. Overall, the CNN model achieved stable learning behaviour, good convergence, and strong classification performance for most maize disease classes through its 81% classification accuracy, although additional optimisation is required in order to improve Gray Leaf Spot detection.

**4.3.1 Result of the CNN Model for Cassava Health**

The proposed CNN model for the Cassava disease/health status, in order to aid an effective crop yield prediction, is depicted in Figures 4.4, 4.5, and 4.6

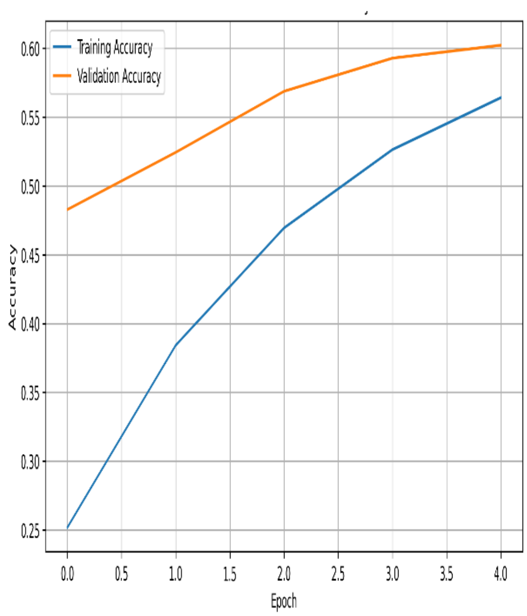



Figure 4.4: Accuracy plot of the tuned model

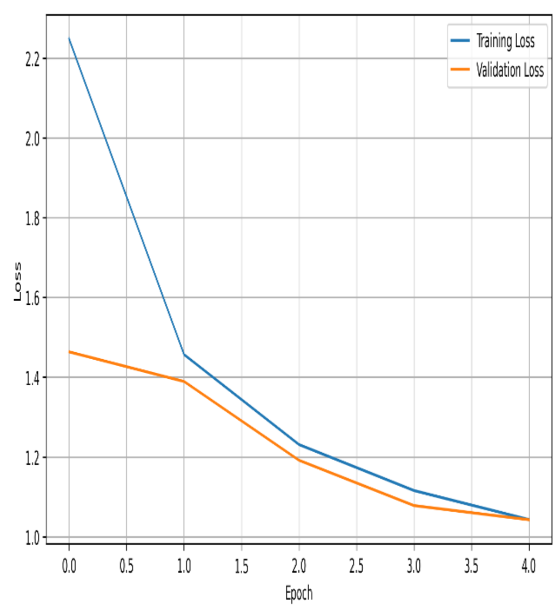



Figure 4.5: Loss plot of the tuned model

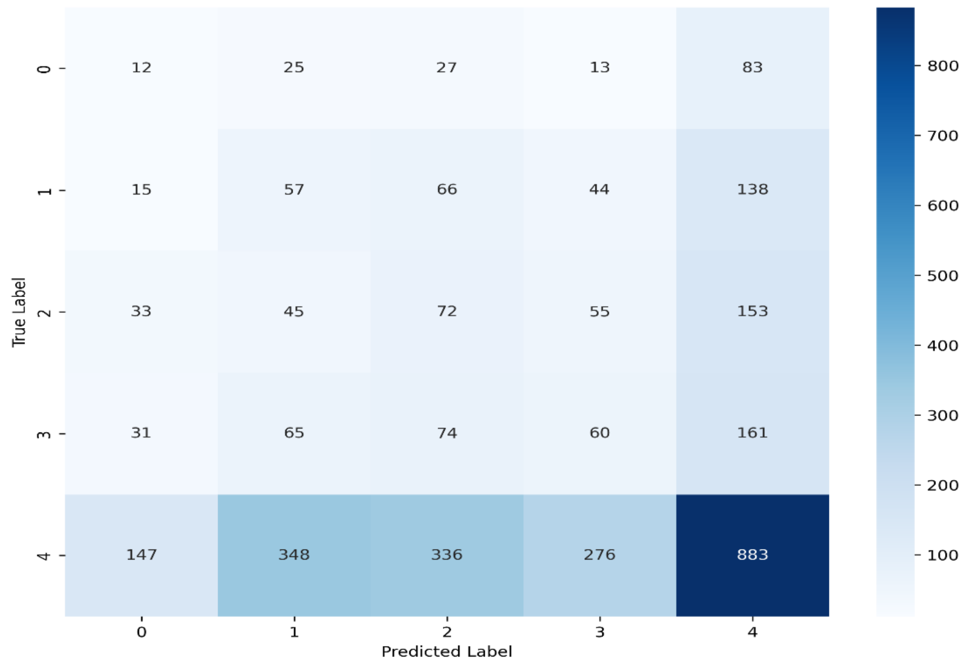



Figure 4.6: Confusion matrix of the cassava model

The accuracy plot in Figure 4.4 illustrates the accuracy of the CNN model across the epochs, displaying a steady increase in both the training and validation accuracy. This suggests that the CNN model was able to learn more meaningful patterns from the cassava leaf dataset as it progressed through the epochs. The validation accuracy was slightly higher than the training accuracy throughout the training process, indicating stable learning behaviour and indicating that there is not much overfitting during the model training process. Likewise, the loss plot in Figure 4.5 shows that both the training and the validation losses have been steadily decreasing over time, indicating that the model is gradually improving its prediction accuracy and the ability to classify the data over time. Lastly, the results presented in the confusion matrix illustrated in Figure 4.6 demonstrated that the model could generally well predict the class of cassava at grade 4, while some cassava samples in classes 0–3 were often misclassified, indicating that the model was not very successful in clearly and effectively classify some of the cassava disease categories.

### 4.3.2 Classification Model Result

The yield classification model adopted was the Extreme Gradient Boosting (XGBoost) after training and comparing it with three other regression models to determine the best for predicting the final crop yield in tonnes. Table 4.3 presents a comparative model performance for the regression models, while Figure 4.7 presents the XGBoost feature importance plot.

**Table 4.3 Comparative Model Performance**

| Model | MAE | RMSE | $R^2$ Score |
|---|---|---|---|
| Linear Regression | 0.469497 | 0.573890 | 0.969737 |
| Random Forest | 0.013074 | 0.027642 | 0.999930 |
| XGBoost | 0.009888 | 0.024824 | 0.999943 |

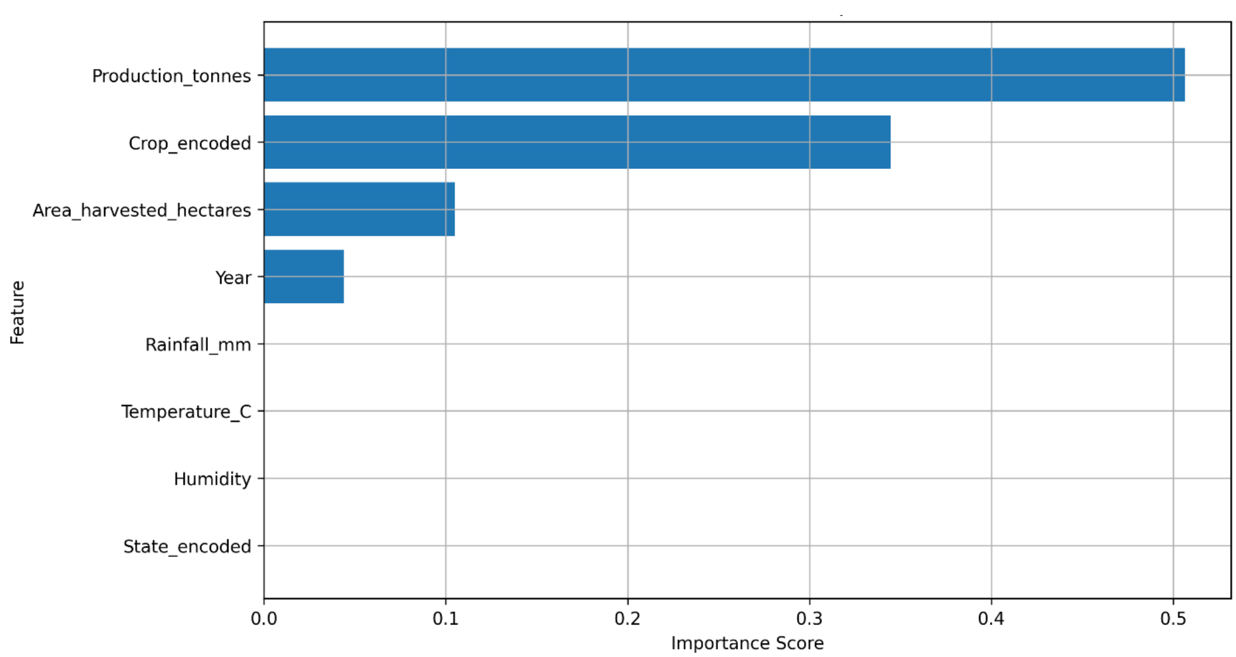



Figure 4.7: Feature Importance Plot

Table 4.3 indicates that the XGBoost model achieved the best overall performance, recording the lowest MAE and RMSE values alongside the highest $R^2$ score, which indicates extremely accurate yield prediction capability of this model. Although Linear Regression produced a strong $R^2$ score, but due to the presence of higher error values compared to Random Forest and XGBoost, it implies that advanced ensemble models handled the agricultural dataset more effectively. In Figure 4.7, the feature importance plot shows that Production_tonnes was the most influential variable in predicting crop yield, followed by Crop_encoded and Area_harvested_hectares. Environmental variables such as Rainfall, Temperature, and Humidity contributed very little to the XGBoost prediction process, which means that production-related variables had a stronger predictive influence in the dataset.

## 4.4 System Interface Description

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

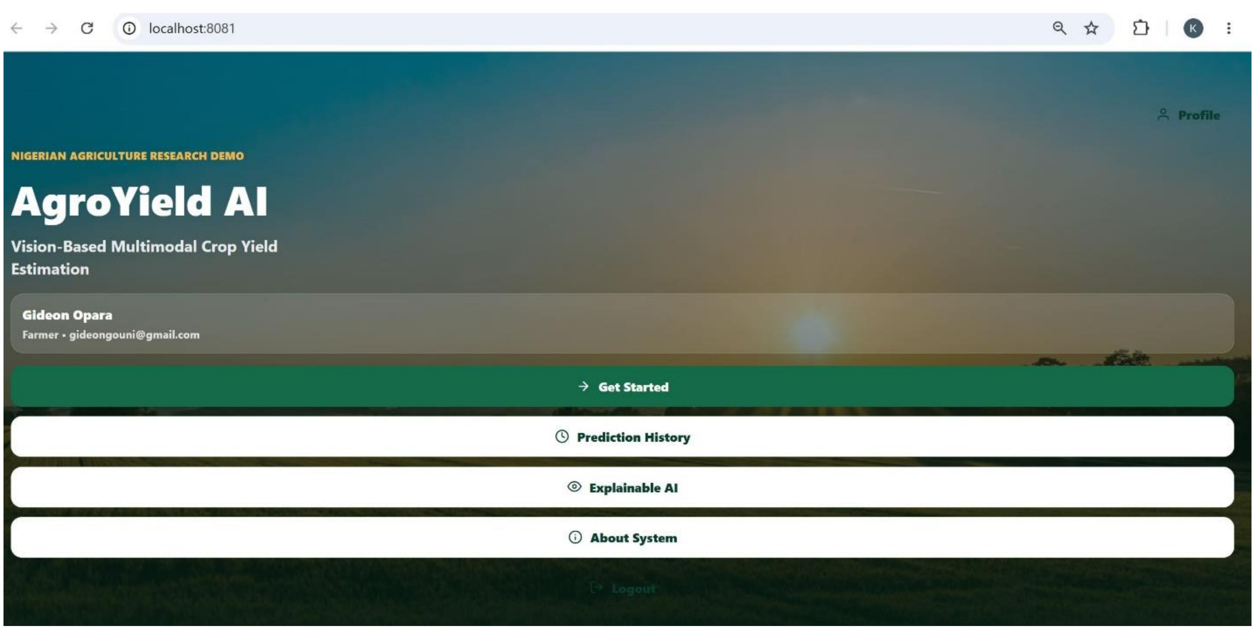



Figure 4.8: Homepage

**B. Parameter Input Page:** These two interfaces allow the farmer or user to upload a cassava or maize farm image and key in the farmland variables to supply to the input form for a prediction.

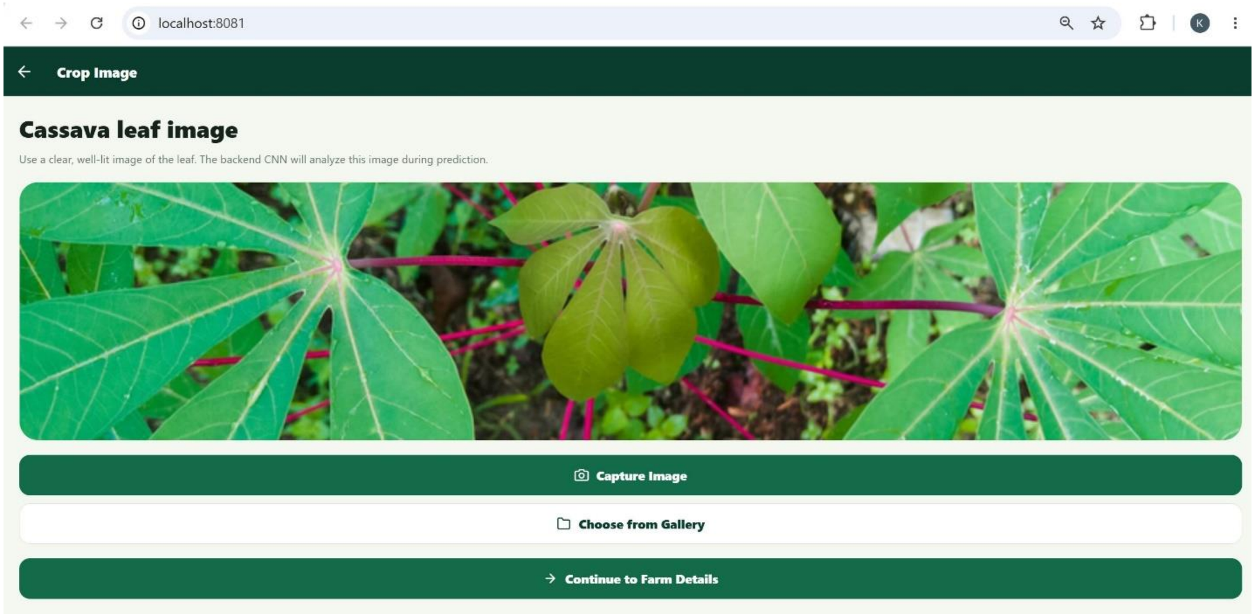



Figure 4.9: Image Input Page

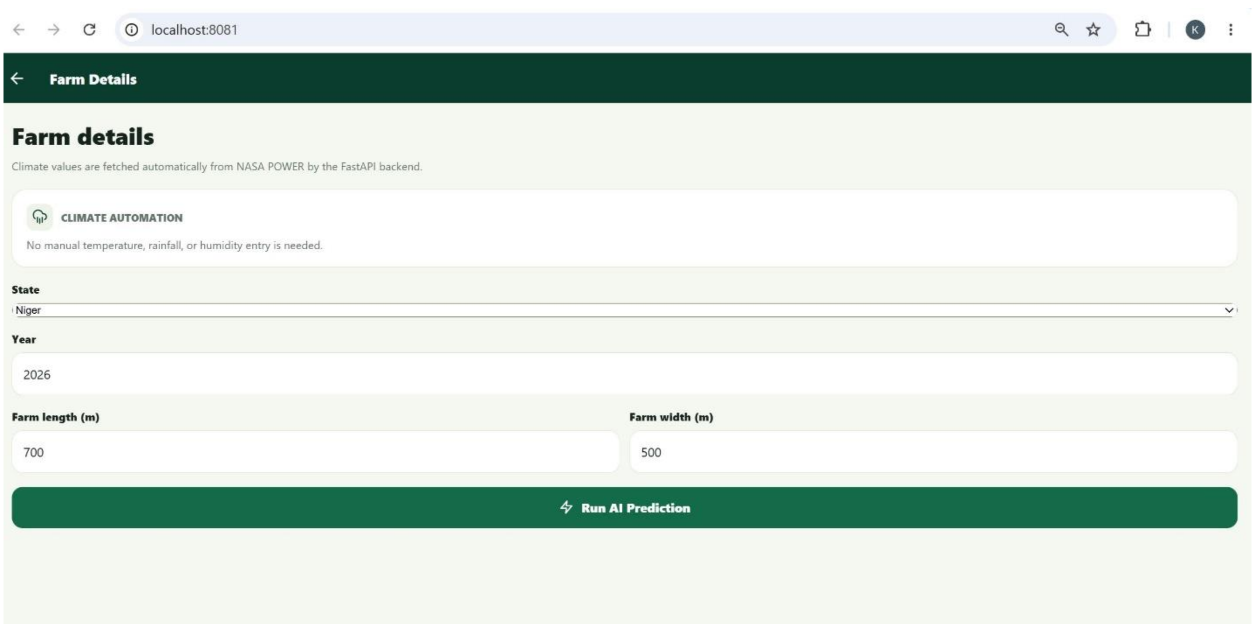



Figure 4.10: farm-details Input Page

**F. Result Output Page:** The output section displays the prediction made by the model after the user has clicked the submit prediction button.

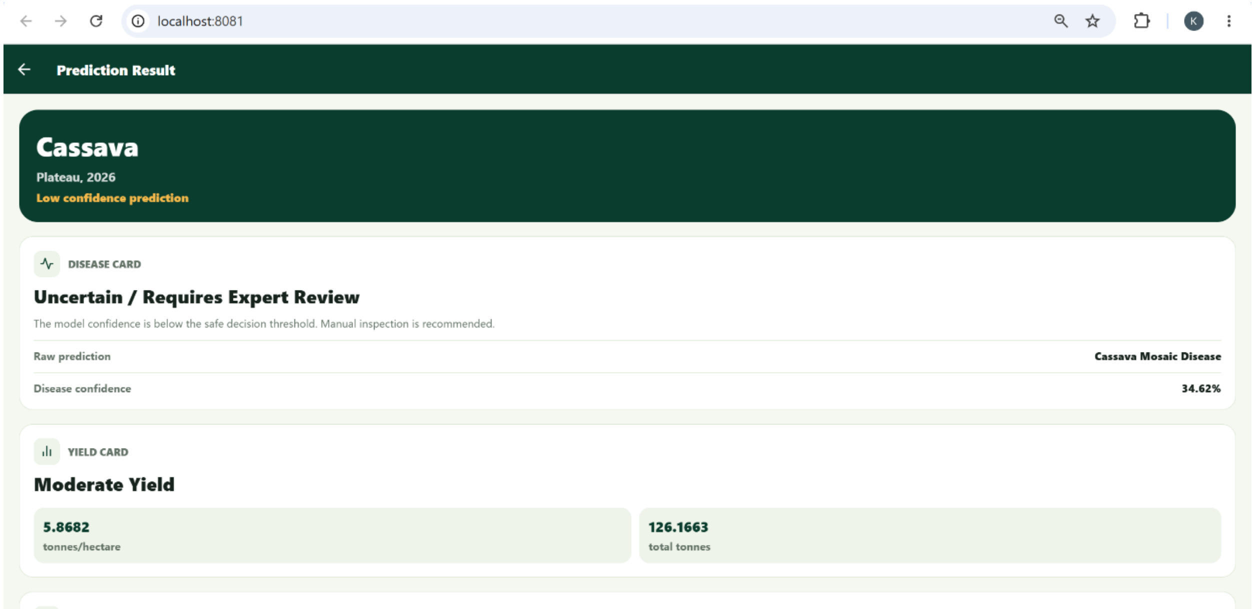



Figure 4.11: Result Output Page

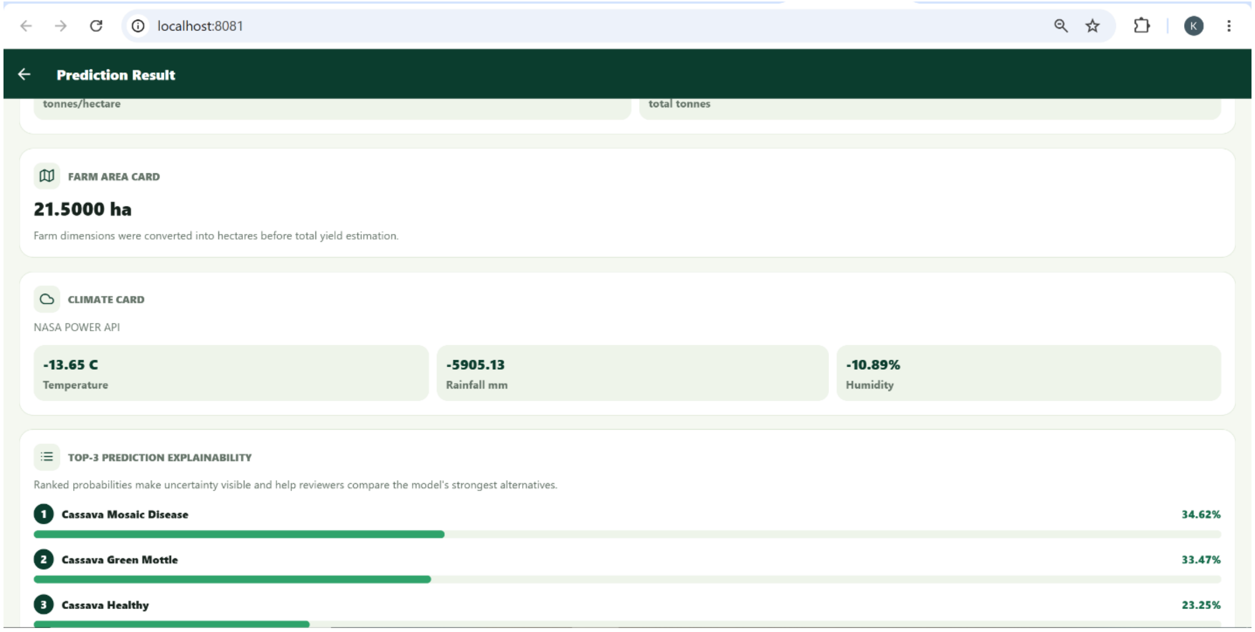



Figure 4.12: Result Output Page

**G. Prediction History Page:** The history section stores previous prediction records

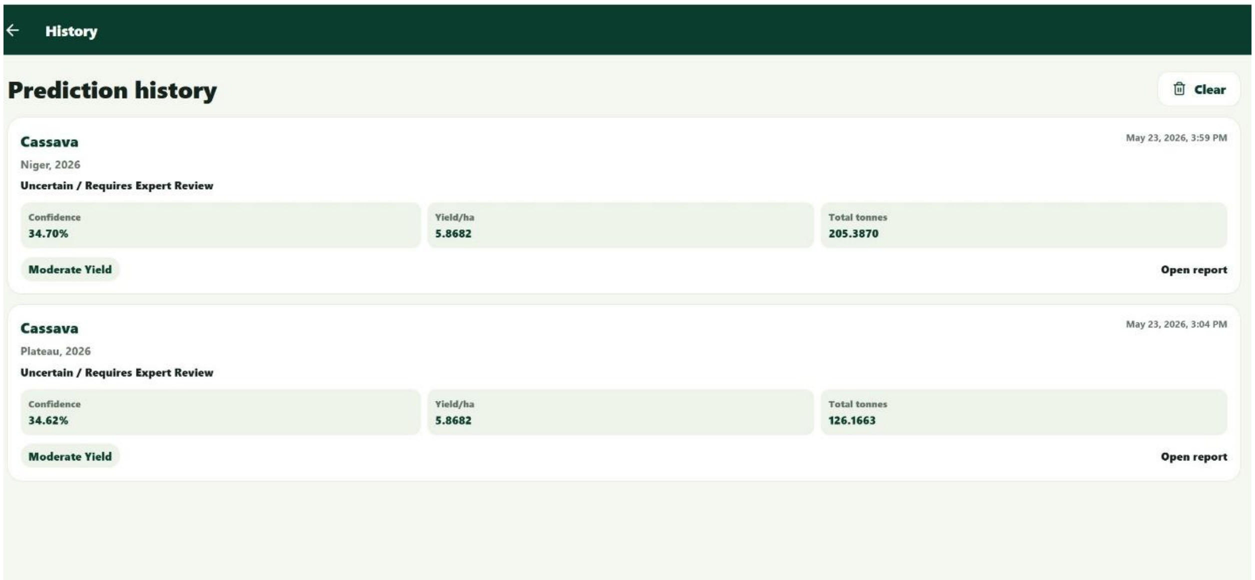



Figure 4.13: Predicteon History Page

# CHAPTER FIVE

# SUMMARY, CONCLUSION AND RECOMMENDATION

## 5.1 Summary

The study developed a vision-based multimodal crop yield estimation system integrating CNN-based disease classification and XGBoost-based yield prediction to effectively predict crop yield, especially for maize and cassava crops, which are the most planted crops in every region of Nigeria. The cassava and maize image datasets collected from Kaggle were utilised for disease classification experiments, while agricultural yield variables from FAOSTAT and climatic variables from NASA were integrated and combined with the crop image dataset in a pipeline that will support the yield prediction framework.

The implementation after dataset collection and preparation starts with image preprocessing operations, which include resizing, RGB conversion, normalisation, and label encoding procedures. TensorFlow datasets were generated to improve CNN training efficiency, and the cassava CNN model achieved approximately 70.7% test accuracy, while the maize model achieved approximately 81.17% test accuracy. Also the XGBoost implementation demonstrated strong predictive performance with low error values and a high $R^2$ score.

Finally, the developed system successfully integrated image analysis and environmental prediction within a unified agricultural framework, which demonstrates a successful improvement and enhancement on the existing AI-based yield prediction, which utilises only tabular data to forecast farm yield.

## 5.2 Conclusion

The study demonstrated the effectiveness of integrating CNN and XGBoost algorithms for intelligent agricultural prediction tasks, as we have accomplished. The developed system successfully classified cassava and maize diseases using crop leaf images in order to aid forecasting of the potential yield of the farmer, meaning a healthy crop will have positive effects on the future harvest. The integrated prediction framework effectively estimated crop yield using environmental and agricultural variables along with the disease classification model.

The multimodal architecture improved the ability of the system to combine visual crop analysis with numerical environmental prediction, thereby demonstrating the study's contribution to the application of artificial intelligence and machine learning within precision agriculture and smart farming systems. The developed framework can support agricultural monitoring, disease assessment, and crop productivity analysis within Nigeria.

## 5.3 Recommendations

i. The study recommends that future studies should utilise larger and more diverse agricultural datasets in order to improve model generalisation.

ii. Advanced CNN architectures may be explored to improve disease classification accuracy.

iii. Future systems may add more environmental variables to improve yield prediction performance, as well as the framework to support additional crop categories beyond cassava and maize.

# References


Alaei, M., & Bendechache, M. (2022). Machine learning techniques for crop yield prediction: A survey. *Journal of Agricultural Informatics, 13*(2), 1–15.

Banhazi, T. M., Lehr, H., Black, J. L., Crabtree, H., Schofield, P., Tscharke, M., & Berckmans, D. (2021). Precision livestock farming: An international review of scientific and commercial aspects. *International Journal of Agricultural and Biological Engineering, 5*(3), 1–9.

Basso, B., & Liu, L. (2024). Seasonal crop yield forecast: Methods, applications and accuracies. *Advances in Agronomy, 154*, 201–255.

Bellocchio, E., Crocetti, F., Costante, G., Fravolini, M. L., & Valigi, P. (2022). A novel vision-based weakly supervised framework for autonomous yield estimation in agricultural applications. *Engineering Applications of Artificial Intelligence, 109*, 104615. https://doi.org/10.1016/j.engappai.2021.104615

Bolton, D. K., & Friedl, M. A. (2021). Forecasting crop yield using remotely sensed vegetation indices and crop phenology metrics. *Agricultural and Forest Meteorology, 173*, 74–84.

Chen, F., & Zhang, P. (2021). Real-time crop yield prediction using IoT sensors and machine learning. *Journal of Agricultural Informatics, 12*(4), 51–62.

Chlingaryan, A., Sukkarieh, S., & Whelan, B. (2023). Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. *Computers and Electronics in Agriculture, 151*, 61–69.

Drummond, S. T., et al. (2021). Improving crop yield prediction through machine learning with satellite and ground-based data fusion. *Remote Sensing, 13*(22), 4561.

FAO. (2022). *The future of food and agriculture – Trends and challenges*. Food and Agriculture Organization of the United Nations.

Furbank, R. T., & Tester, M. (2021). Phenomics—Technologies to relieve the phenotyping bottleneck. *Trends in Plant Science, 16*(12), 635–644.

Gupta, R., & Sharma, P. (2023). Weather-based predictive model for crop yield forecasting. *Agricultural Climate Modeling, 25*, 100–110.

Hunt, E. R., Hively, W. D., Fujikawa, S. J., Linden, D. S., Daughtry, C. S., & McCarty, G. W. (2023). Acquisition of NIR-green-blue digital photographs from unmanned aircraft for crop monitoring. *Remote Sensing, 2*(1), 290–305.

Jones, J. W., Antle, J. M., Basso, B., Boote, K. J., Conant, R. T., Foster, I., ... & Rosenzweig, C. (2022). Toward a new generation of agricultural system models. *Agricultural Systems, 155*, 269–288.

Jordan, M. I., & Mitchell, T. M. (2022). Machine learning: Trends, perspectives and prospects. *Science, 349*(6245), 255–260.

Kamilaris, A., & Prenafeta-Boldú, F. X. (2023). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture, 147*, 70–90.

Kamilaris, A., Kartakoullis, A., & Prenafeta-Boldú, F. X. (2022). A review on the practice of big data analysis in agriculture. *Computers and Electronics in Agriculture, 143*, 23–37.

Khaki, S., & Wang, L. (2024). Crop yield prediction using deep neural networks. *Frontiers in Plant Science, 10*, 621.

Khosla, R., Pinter Jr, P. J., & Hatfield, J. L. (2021). Emerging opportunities in digital agriculture. *Agronomy Journal, 113*(2), 1141–1161.

Kim, Y., Evans, R. G., & Iversen, W. M. (2020). Remote sensing and control of an irrigation system. *IEEE Transactions on Instrumentation and Measurement, 57*(7), 1379–1387.

Kross, A., et al. (2022). Predicting maize biomass and yield. *Field Crops Research, 170*, 79–90.

Li, X., Lyu, Y., Zhu, B., Liu, L., & Song, K. (2025). Maize yield estimation using deep learning. *Scientific Reports, 15*, 12927. https://doi.org/10.1038/s41598-025-97563-6

Liu, Y., Zhang, Z., & Wang, L. (2020). Artificial neural networks for rice yield forecasting. *Agricultural Systems, 181*, 35–45.

Lobell, D. B., & Burke, M. B. (2021). Statistical models for crop yield prediction. *Agricultural and Forest Meteorology, 150*(11), 1443–1452.

Mulla, D. J. (2021). Remote sensing in precision agriculture. *Biosystems Engineering, 114*(4), 358–371.

Pantazi, X. E., et al. (2022). Wheat yield prediction using machine learning. *Computers and Electronics in Agriculture, 121*, 57–65.

Patel, A., & Jain, S. (2020). Climate and soil data integration. *Agricultural Systems, 152*, 88–97.

Patil, R., & Yadav, M. (2020). Deep neural networks for crop yield prediction. *Computers and Electronics in Agriculture, 177*, 105742.

Pedersen, S. M., Fountas, S., Blackmore, B. S., Gylling, M., & Pedersen, J. L. (2020). Precision agriculture adoption. *Computers and Electronics in Agriculture, 50*(2), 237–250.

Pôças, I., et al. (2020). Vegetation indices as crop yield predictors. *Remote Sensing, 12*(23), 4000.

Russell, S., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson.

Shahhosseini, M., Hu, G., & Archontoulis, S. V. (2020). Forecasting corn yield with machine learning. *Agronomy Journal, 112*(1), 613–626.

Sharma, N., et al. (2020). Applications of AI in agriculture. *Journal of Agriculture and Food Research, 2*, 100019.

Singh, S., Mehta, D., & Khan, R. (2021). Comparison of ML models for crop yield prediction. *Environmental Modelling & Software, 139*, 102512.

Sun, L., et al. (2024). Deep learning for plant stress phenotyping. *Trends in Plant Science, 24*(10), 883–898.

Sun, X., Zhang, X., & Lin, Q. (2024). Deep learning models for crop yield forecasting. *IEEE Transactions on Agricultural and Biological Engineering, 62*(8), 2345–2353.

Tan, L., Hu, Z., & Yu, H. (2022). Ensemble learning methods for crop yield prediction. *Applied Soft Computing, 56*, 13–22.

Tatsumi, K., & Usami, T. (2024). Plant-level prediction of potato yield. *Discover Applied Sciences, 6*, 649. https://doi.org/10.1007/s42452-024-06362-7

Van Straten, G., van Willigenburg, L. G., van Henten, E. J., & van Ooteghem, R. J. (2021). *Optimal control of greenhouse cultivation*. CRC Press.

Verdouw, C. N., Wolfert, J., Beulens, A. J. M., & Rialland, A. (2022). Virtualization of food supply chains. *Journal of Food Engineering, 176*, 128–136.

Wang, J., Li, Q., & Zhang, X. (2024). Multi-model crop yield prediction. *Agricultural Meteorology, 61*, 29–42.

Wang, Y., Zhang, Q., Yu, F., Zhang, N., Zhang, X., Li, Y., Wang, M., & Zhang, J. (2024). Progress in deep learning-based crop yield prediction. *Agronomy, 14*(10), 2264. https://doi.org/10.3390/agronomy14102264

Wei, Y., Chen, X., & Zhang, Q. (2022). Random forest regression for crop yield prediction. *Agricultural Systems, 152*, 72–80.

Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M. J. (2022). Big data in smart farming. *Agricultural Systems, 153*, 69–80.

You, J., Li, X., Low, M., Lobell, D., & Ermon, S. (2022). Deep Gaussian process for crop yield prediction. *AAAI Conference Proceedings, 31*(1).

Zhai, Z., Martín, M. P., & Duveiller, G. (2022). Predicting crop yields using ML methods. *Climate, 10*(2), 29.

Zhang, C., & Kovacs, J. M. (2021). UAV applications in precision agriculture. *Precision Agriculture, 13*(6), 693–712.

Zhang, L., Xu, X., & Yang, X. (2020). Hybrid model for crop yield prediction. *Journal of Environmental Management, 253*, 109–118.

Zhang, Y., Wang, S., & Wang, J. (2020). Time series forecasting using hybrid models. *Neurocomputing, 337*, 240–250.

# Appendix A: Source Code

```
from fastapi import FastAPI, File, UploadFile, Form, HTTPException
from fastapi.middleware.cors import CORSMiddleware
from datetime import date
import os

os.environ.setdefault("MPLCONFIGDIR", "/tmp/matplotlib")
os.makedirs(os.environ["MPLCONFIGDIR"], exist_ok=True)

from PIL import Image
import tensorflow as tf
import numpy as np
import io
import joblib
import requests
import pandas as pd
import shap
from gradcam_utils import generate_gradcam_base64

app = FastAPI(title="Vision-Based Multimodal Crop Yield Estimation API")

app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
    allow_credentials=True,
    allow_methods=["*"],
```

```
    allow_headers=["*"],
)

cassava_model = tf.keras.models.load_model("../models/cassava_baseline_cnn.keras")
maize_model = tf.keras.models.load_model("../models/final_fixed_maize_cnn.keras")

yield_model = joblib.load("../models/farmer_xgboost_yield_regressor.pkl")
scaler = joblib.load("../models/farmer_yield_feature_scaler.pkl")
state_encoder = joblib.load("../models/state_encoder.pkl")
crop_encoder = joblib.load("../models/crop_encoder.pkl")
feature_columns = joblib.load("../models/farmer_yield_feature_columns.pkl")
shap_explainer = shap.TreeExplainer(yield_model)

STATE_COORDINATES = {
    "Ondo": (7.25, 5.20),
    "Enugu": (6.45, 7.50),
    "Kogi": (7.80, 6.74),
    "Oyo": (7.38, 3.93),
    "Kaduna": (10.52, 7.44),
    "Plateau": (9.21, 9.52),
    "Niger": (9.93, 5.60),
    "Kano": (12.00, 8.52),
}

cassava_classes = [
    "Cassava Bacterial Blight",
    "Cassava Brown Streak Disease",
    "Cassava Green Mottle",
    "Cassava Healthy",
    "Cassava Mosaic Disease",
]

maize_classes = [
    "Blight",
    "Common Rust",
    "Gray Leaf Spot",
    "Healthy",
]

def preprocess_image(image_bytes):
```

```
    image = Image.open(io.BytesIO(image_bytes)).convert("RGB")
    image = image.resize((128, 128))
    image_array = np.array(image) / 255.0
    return np.expand_dims(image_array, axis=0)

def preprocess_image_with_original(image_bytes):
    image = Image.open(io.BytesIO(image_bytes)).convert("RGB")
    image = image.resize((128, 128))
    original_image_array = np.array(image).astype("uint8")
    image_array = original_image_array / 255.0
    return np.expand_dims(image_array, axis=0), original_image_array

def build_gradcam_response(model, image_tensor, original_image_array, predicted_index):
    """Generate live Grad-CAM while keeping prediction resilient if XAI fails."""
    try:
        gradcam_image_base64 = generate_gradcam_base64(
            model=model,
            image_tensor=image_tensor,
            original_image_array=original_image_array,
            class_index=predicted_index,
        )

        return {
            "gradcam_image_base64": gradcam_image_base64,
            "explainability_status": "active",
            "explainability_note": "Grad-CAM highlights image regions contributing most to CNN prediction.",
        }
    except Exception as error:
        return {
            "gradcam_image_base64": None,
            "explainability_status": "unavailable",
            "explainability_note": f"Grad-CAM generation failed, but prediction completed successfully: {str(error)}",
        }

def build_yield_feature_vector(crop, state, year, temperature_c, rainfall_mm, humidity):
    state_encoded = state_encoder.transform([state])[0]
    crop_encoded = crop_encoder.transform([crop])[0]
```

```
    feature_values = {
        "Year": year,
        "Temperature_C": temperature_c,
        "Rainfall_mm": rainfall_mm,
        "Humidity": humidity,
        "State_encoded": state_encoded,
        "Crop_encoded": crop_encoded,
    }

        features = pd.DataFrame([[feature_values[column] for column in feature_columns]],
columns=feature_columns)
    return features, feature_values

def generate_shap_explanation(crop, state, year, temperature_c, rainfall_mm, humidity):
    """SHAP explains the XGBoost yield regression side of the dual XAI pipeline."""
    try:
        features, feature_values = build_yield_feature_vector(
            crop,
            state,
            year,
            temperature_c,
            rainfall_mm,
            humidity,
        )
        features_scaled = scaler.transform(features)
        shap_values = shap_explainer.shap_values(features_scaled)

        if isinstance(shap_values, list):
            shap_values = shap_values[0]

        shap_values = np.array(shap_values)[0]
        base_value = shap_explainer.expected_value
        if isinstance(base_value, (list, np.ndarray)):
            base_value = np.array(base_value).flatten()[0]

        contributions = []
        for index, feature in enumerate(feature_columns):
            shap_value = float(shap_values[index])
            contributions.append({
                "feature": feature,
```

```
                "value": round(float(feature_values[feature]), 4),
                "shap_value": round(shap_value, 4),
                "impact": "positive" if shap_value >= 0 else "negative",
            })

        contributions = sorted(
            contributions,
            key=lambda item: abs(item["shap_value"]),
            reverse=True,
        )
        prediction_sum = float(base_value + np.sum(shap_values))
        top_factor = contributions[0]["feature"] if contributions else None

        return {
            "shap_explainability_status": "active",
            "shap_explainability_note": "SHAP explains how each tabular feature contributed to the
XGBoost yield prediction.",
            "shap_base_value": round(float(base_value), 4),
            "shap_prediction_sum": round(prediction_sum, 4),
            "shap_top_factor": top_factor,
            "shap_feature_contributions": contributions,
        }
    except Exception:
        return {
            "shap_explainability_status": "unavailable",
            "shap_explainability_note": "SHAP explanation could not be generated for this prediction.",
            "shap_base_value": None,
            "shap_prediction_sum": None,
            "shap_top_factor": None,
            "shap_feature_contributions": [],
        }

def fetch_climate_from_nasa(state, year):
    if state not in STATE_COORDINATES:
        raise HTTPException(
            status_code=400,
            detail=f"Unsupported state. Available states are: {list(STATE_COORDINATES.keys())}",
        )

    lat, lon = STATE_COORDINATES[state]
```

```
    current_year = date.today().year
    today = date.today()

    if year < current_year:
        start_date = f"{year}0101"
        end_date = f"{year}1231"
        climate_mode = "historical_full_year"
        climate_reference_year = year
    elif year == current_year:
        # Current-season inference uses only observations that exist so far.
        start_date = f"{year}0101"
        end_date = today.strftime("%Y%m%d")
        climate_mode = "current_year_to_date"
        climate_reference_year = year
    else:
        # Future daily climate is unavailable, so use the latest completed year as proxy input.
        proxy_year = current_year - 1
        start_date = f"{proxy_year}0101"
        end_date = f"{proxy_year}1231"
        climate_mode = "future_year_proxy_from_recent_completed_year"
        climate_reference_year = proxy_year

    url = "https://power.larc.nasa.gov/api/temporal/daily/point"

    params = {
        "parameters": "T2M,PRECTOTCORR,RH2M",
        "community": "AG",
        "longitude": lon,
        "latitude": lat,
        "start": start_date,
        "end": end_date,
        "format": "JSON",
    }

    response = requests.get(url, params=params, timeout=60)

    if response.status_code != 200:
        raise HTTPException(
            status_code=502,
            detail="Unable to fetch climate data from NASA POWER API.",
```

```
    )

    data = response.json()
    parameters = data["properties"]["parameter"]

    daily_df = pd.DataFrame({
        "Temperature_C": list(parameters["T2M"].values()),
        "Rainfall_mm": list(parameters["PRECTOTCORR"].values()),
        "Humidity": list(parameters["RH2M"].values()),
    })

    for column in ["Temperature_C", "Rainfall_mm", "Humidity"]:
        daily_df[column] = pd.to_numeric(daily_df[column], errors="coerce")

    daily_df = daily_df.replace([-999, -9999], np.nan).dropna()
    daily_df = daily_df[
        daily_df["Temperature_C"].between(0, 45)
        & (daily_df["Rainfall_mm"] >= 0)
        & daily_df["Humidity"].between(0, 100)
    ]

    if daily_df.empty:
        raise HTTPException(
            status_code=502,
            detail="Valid climate data could not be retrieved for the selected state/year.",
        )

    return {
        "temperature_c": round(float(daily_df["Temperature_C"].mean()), 4),
        "rainfall_mm": round(float(daily_df["Rainfall_mm"].sum()), 4),
        "humidity": round(float(daily_df["Humidity"].mean()), 4),
        "climate_mode": climate_mode,
        "climate_start_date": start_date,
        "climate_end_date": end_date,
        "requested_year": year,
        "climate_reference_year": climate_reference_year,
    }

def get_climate_mode_note(climate_mode):
    if climate_mode == "current_year_to_date":
```

```
        return "Current-year climate values are based on NASA POWER year-to-date observations."
    if climate_mode == "future_year_proxy_from_recent_completed_year":
        return "Future-year climate values are estimated using the most recent completed climate year as a proxy."
    return "Historical climate values are based on NASA POWER full-year observations."

def classify_yield(yield_value):
    if yield_value >= 10:
        return "High Yield"
    if yield_value >= 5:
        return "Moderate Yield"
    return "Low Yield"

def apply_confidence_safety(predicted_class, confidence, threshold=65):
    if confidence < threshold:
        return {
            "final_class": "Uncertain / Requires Expert Review",
            "status": "Low confidence prediction",
            "note": "The model confidence is below the safe decision threshold. Manual inspection is recommended.",
        }

    return {
        "final_class": predicted_class,
        "status": "Accepted prediction",
        "note": "The model confidence is above the safe decision threshold.",
    }

def get_top_predictions(prediction, class_names, top_n=3):
    prediction_values = prediction[0]
    top_indices = np.argsort(prediction_values)[::-1][:top_n]

    return [
        {
            "class": class_names[index],
            "confidence": round(float(prediction_values[index]) * 100, 2),
        }
        for index in top_indices
    ]
```

```
def predict_yield(crop, state, year, temperature_c, rainfall_mm, humidity):
    features, _ = build_yield_feature_vector(
        crop,
        state,
        year,
        temperature_c,
        rainfall_mm,
        humidity,
    )
    features_scaled = scaler.transform(features)
    predicted_yield = float(yield_model.predict(features_scaled)[0])

    return {
        "estimated_yield_tonnes_per_hectare": round(predicted_yield, 4),
        "yield_category": classify_yield(predicted_yield),
    }

@app.get("/")
def home():
    return {
        "message": "Vision-Based Multimodal Crop Yield Estimation API is running",
        "available_endpoints": ["/predict/cassava", "/predict/maize"],
        "climate_source": "NASA POWER API",
    }

@app.post("/predict/cassava")
async def predict_cassava(
    file: UploadFile = File(...),
    state: str = Form(...),
    year: int = Form(...),
    farm_length_m: float = Form(...),
    farm_width_m: float = Form(...),
):
    climate = fetch_climate_from_nasa(state, year)

    image_bytes = await file.read()
    image, original_image = preprocess_image_with_original(image_bytes)

    prediction = cassava_model.predict(image)
    predicted_index = int(np.argmax(prediction))
```

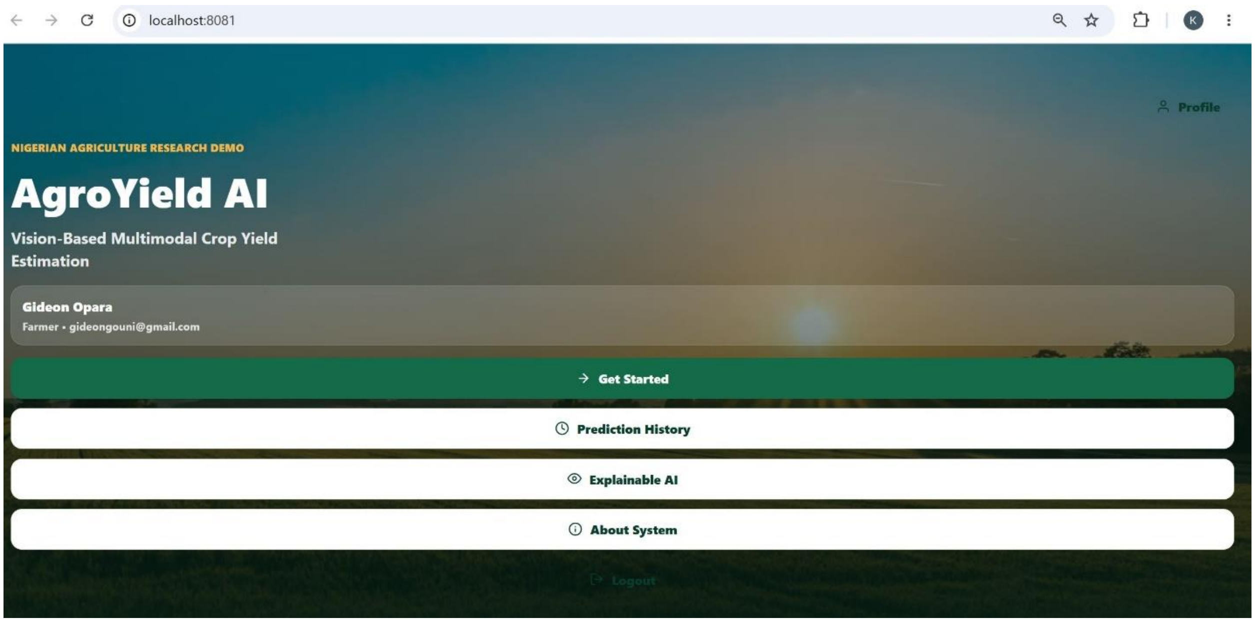
```
    confidence = round(float(np.max(prediction)) * 100, 2)
    raw_prediction = cassava_classes[predicted_index]

    safe_prediction = apply_confidence_safety(raw_prediction, confidence)
    top_predictions = get_top_predictions(prediction, cassava_classes)
    # Grad-CAM explains CNN visual diagnosis; SHAP explains XGBoost yield regression.
    explainability = build_gradcam_response(
        cassava_model,
        image,
        original_image,
        predicted_index,
    )

    yield_result = predict_yield(
        crop="Cassava",
        state=state,
        year=year,
        temperature_c=climate["temperature_c"],
        rainfall_mm=climate["rainfall_mm"],
        humidity=climate["humidity"],
    )
    shap_explainability = generate_shap_explanation(
        crop="Cassava",
        state=state,
        year=year,
        temperature_c=climate["temperature_c"],
        rainfall_mm=climate["rainfall_mm"],
        humidity=climate["humidity"],
    )

    farm_area_hectares = (farm_length_m * farm_width_m) / 10000
    total_estimated_yield = (
        yield_result["estimated_yield_tonnes_per_hectare"] * farm_area_hectares
    )

    return {
        "crop": "Cassava",
        "state": state,
        "year": year,
        "climate_source": "NASA POWER API",
```

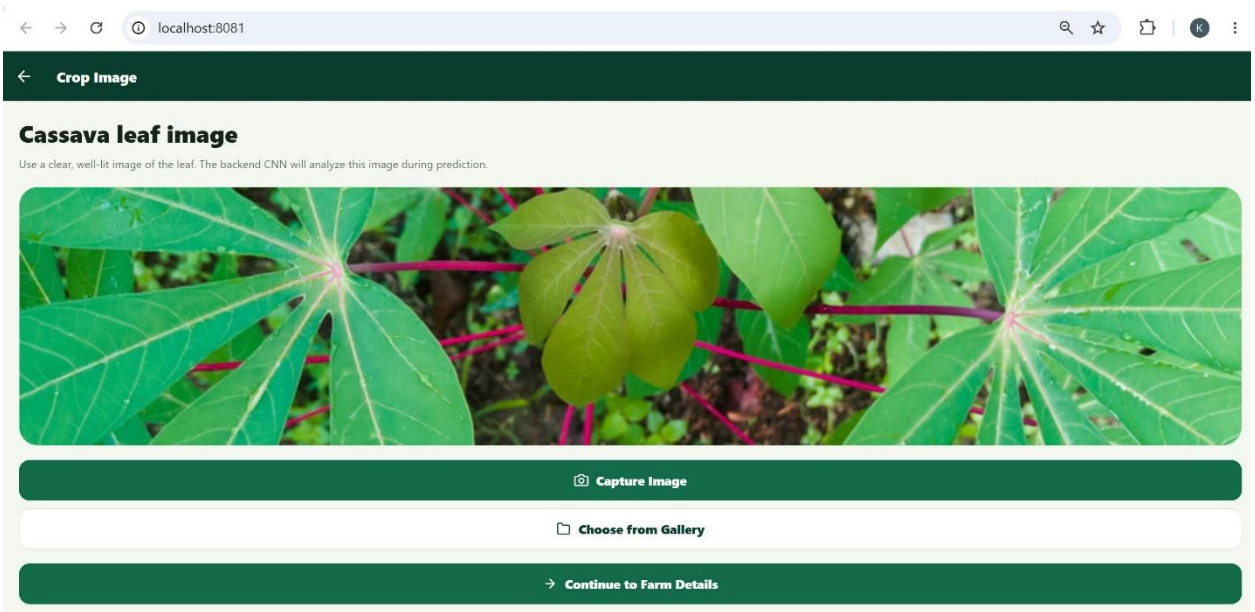
```
        "climate_variables": climate,
        "predicted_disease_or_condition": safe_prediction["final_class"],
        "raw_model_prediction": raw_prediction,
        "disease_confidence": confidence,
        "prediction_status": safe_prediction["status"],
        "prediction_note": safe_prediction["note"],
        "top_3_predictions": top_predictions,
        "estimated_yield_tonnes_per_hectare": yield_result["estimated_yield_tonnes_per_hectare"],
        "farm_area_hectares": round(farm_area_hectares, 4),
        "total_estimated_farm_yield_tonnes": round(total_estimated_yield, 4),
        "yield_category": yield_result["yield_category"],
        "note": f"Climate variables were automatically retrieved from NASA POWER. {get_climate_mode_note(climate['climate_mode'])} Yield prediction uses crop image analysis, climate variables, location, and farm size.",
        **explainability,
        **shap_explainability,
    }

@app.post("/predict/maize")
async def predict_maize(
    file: UploadFile = File(...),
    state: str = Form(...),
    year: int = Form(...),
    farm_length_m: float = Form(...),
    farm_width_m: float = Form(...),
):
    climate = fetch_climate_from_nasa(state, year)

    image_bytes = await file.read()
    image, original_image = preprocess_image_with_original(image_bytes)

    prediction = maize_model.predict(image)
    predicted_index = int(np.argmax(prediction))
    confidence = round(float(np.max(prediction)) * 100, 2)
    raw_prediction = maize_classes[predicted_index]

    safe_prediction = apply_confidence_safety(raw_prediction, confidence)
    top_predictions = get_top_predictions(prediction, maize_classes)
    # Together, Grad-CAM and SHAP provide dual explainable AI for image and yield models.
    explainability = build_gradcam_response(
```

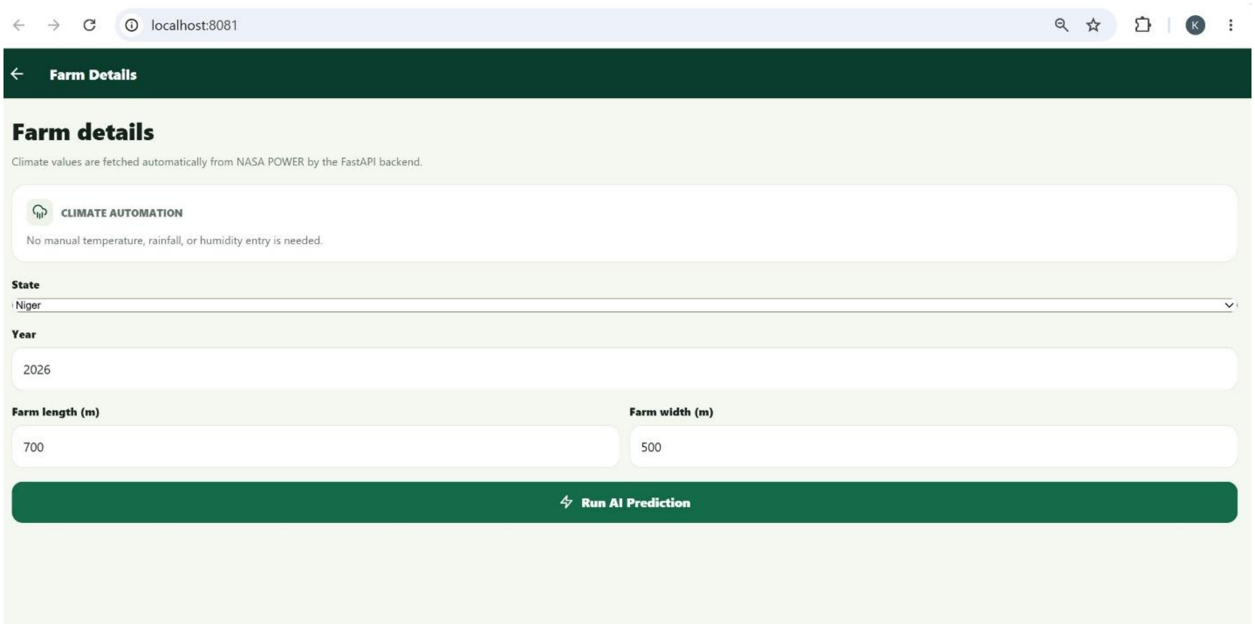
```
        maize_model,
        image,
        original_image,
        predicted_index,
    )

    yield_result = predict_yield(
        crop="Maize",
        state=state,
        year=year,
        temperature_c=climate["temperature_c"],
        rainfall_mm=climate["rainfall_mm"],
        humidity=climate["humidity"],
    )
    shap_explainability = generate_shap_explanation(
        crop="Maize",
        state=state,
        year=year,
        temperature_c=climate["temperature_c"],
        rainfall_mm=climate["rainfall_mm"],
        humidity=climate["humidity"],
    )

    farm_area_hectares = (farm_length_m * farm_width_m) / 10000
    total_estimated_yield = (
        yield_result["estimated_yield_tonnes_per_hectare"] * farm_area_hectares
    )

    return {
        "crop": "Maize",
        "state": state,
        "year": year,
        "climate_source": "NASA POWER API",
        "climate_variables": climate,
        "predicted_disease_or_condition": safe_prediction["final_class"],
        "raw_model_prediction": raw_prediction,
        "disease_confidence": confidence,
        "prediction_status": safe_prediction["status"],
        "prediction_note": safe_prediction["note"],
        "top_3_predictions": top_predictions,
```

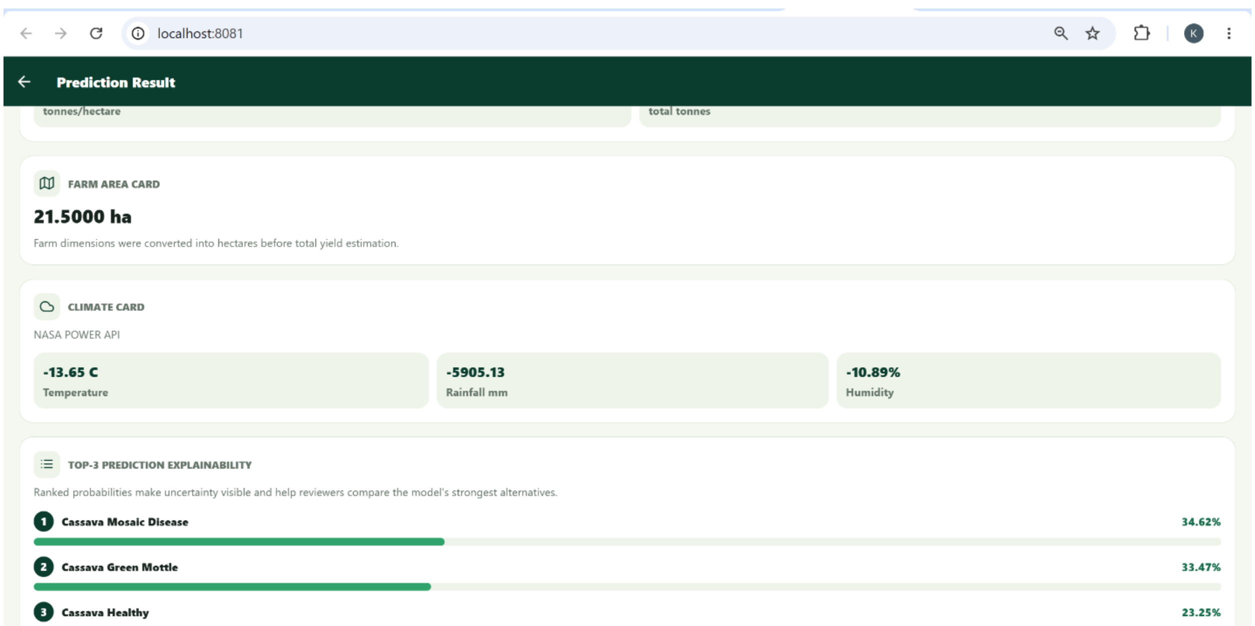
```
    "estimated_yield_tonnes_per_hectare": yield_result["estimated_yield_tonnes_per_hectare"],
    "farm_area_hectares": round(farm_area_hectares, 4),
    "total_estimated_farm_yield_tonnes": round(total_estimated_yield, 4),
    "yield_category": yield_result["yield_category"],
        "note": f"Climate variables were automatically retrieved from NASA POWER. {get_climate_mode_note(climate['climate_mode'])} Yield prediction uses crop image analysis, climate variables, location, and farm size.",
    **explainability,
    **shap_explainability,
  }
```

**Appendix B: User Interfaces**


localhost:8081
Profile
NIGERIAN AGRICULTURE RESEARCH DEMO
AgroYield AI
Vision-Based Multimodal Crop Yield Estimation
Gideon Opara
Farmer • gideongouni@gmail.com
Get Started
Prediction History
Explainable AI
About System
Logout


localhost:8081
Crop Image
Cassava leaf image
Use a clear, well-lit image of the leaf. The backend CNN will analyze this image during prediction.
Capture Image
Choose from Gallery
Continue to Farm Details


localhost:8081
Farm Details
Farm details
Climate values are fetched automatically from NASA POWER by the FastAPI backend.
CLIMATE AUTOMATION
No manual temperature, rainfall, or humidity entry is needed.
State
Niger
Year
2026
Farm length (m)
700
Farm width (m)
500
Run AI Prediction


localhost:8081
Prediction Result
tonnes/hectare
total tonnes
FARM AREA CARD
21.5000 ha
Farm dimensions were converted into hectares before total yield estimation.
CLIMATE CARD
NASA POWER API
-13.65 C
Temperature
-5905.13
Rainfall mm
-10.89%
Humidity
TOP-3 PREDICTION EXPLAINABILITY
Ranked probabilities make uncertainty visible and help reviewers compare the model's strongest alternatives.
1 Cassava Mosaic Disease
34.62%
2 Cassava Green Mottle
33.47%
3 Cassava Healthy
23.25%