AI-BASED MOBILE APPLICATION FOR CROP DISEASE RECOGNITION AND MANAGEMENT RECOMMENDATIONS FOR NIGERIAN SMALLHOLDER FARMERS

BY

OLIVER CHRISTIAN NMESOMA

GOU/U22/CSC/820

DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS, 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 "**INTELLIGENT SYSTEM FOR EARLY CATARACT DETECTION"** has been assessed and approved by the Department of Computer Science and Mathematics Committees in the Faculty of Natural Sciences and Environmental Studies, Godfrey Okoye University, Enugu, Nigeria.

	....	....
Engr. Camillus Njoku	Signature	Date
Project Supervisor		
External Examiner	....	....
External Examiner	Signature	Date
Dr. Frank Okebanama	....	....
HOD, Department of Computer Science and Mathematics	Signature	Date
Prof.Dr. I A Ajah	....	....
Dean, Faculty of Computing And Information Technology.	Signature	Date

CERTIFICATION

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

____

OLIVER CHRISTIAN NMESOMA

GOU/U22/CSC/820

DEDICATION

This project is dedicated to God Almighty for His grace and guidance, to my family for their unwavering support, and with special recognition to my late father, who initiated this journey.

ACKNOWLEDGMENTS

First and above all, I give thanks to God for the strength to keep going when the work felt endless, for clarity when nothing made sense, and for seeing me through to this point. None of this would exist without His grace.

To my parents Mr. and Mrs. Oliver Nnajiofor, thank you for never once making me feel like this was too much to pursue. Your support through the long nights, the pressure, and the uncertainty was the kind that cannot be repaid only carried forward.

To my project supervisor, Engr. Camillus Njoku, thank you not just for the guidance, but for the patience. You asked the right questions, pointed out what I had missed, and never let me settle for less than my best. This project is sharper because of you.

To my Head of Department Dr. Frank Okebanama, I am grateful for the direction and academic environment that made this project possible. Your commitment to the department sets a standard that pushes students to take their work seriously.

To the lecturers who taught me across these years: Dr Ajah (Dean), Dr Okebanama (HOD), Prof Okereke, Engr Kingsley and others whose names I cannot mention here, I did not always appreciate it at the moment, but looking back, the foundation you built shows up in every part of this work. Thank you.

And to everyone else who played even a small part in this project, whether through advice, a conversation, or just encouragement at the right time, it meant more than you know.

ABSTRACT

In this project, AgroAI Doctor, a mobile-based intelligent system that can detect plant diseases and provide practical guidance to farmers, will be introduced. This paper highlights some drawbacks in current solutions used in agricultural fields, such as reliance on the Internet, no local disease recognition capabilities, and difficulty of use by people with low literacy. The suggested system uses a CNN model, which is based on the MobileNetV2 neural network architecture. Moreover, the optimization of this system is done via TensorFlow Lite for effective model predictions. On the other hand, Flutter is used to develop a mobile app, while FastAPI and Supabase cloud platform are used for implementing an online system and performing online predictions and storage. The user uses the mobile application to take an image of a plant leaf; the image is preprocessed and analyzed either in the cloud or locally based on network availability. Disease classification and localized advice for treatment is provided by the system. It has been shown that AgroAI Doctor works well and delivers accurate predictions even under various conditions. Usability has been considered a priority in implementing AgroAI Doctor using a visual design approach, along with streamlined navigation. In summary, AgroAI Doctor offers an effective solution for farmers in developing countries.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER ONE

INTRODUCTION

Background of the Study

The agriculture sector plays a significant role in the Nigerian economy, providing jobs for much of its rural labor force and also producing food for the country. However, the sector faces challenges such as plant diseases caused by fungi, bacteria, viruses, and other pests. (FAO, 2022; FAO,2023).

Plant diseases limit agricultural productivity in Nigeria because they result in reduced yields and poor crop quality. Additionally, some plant diseases contribute to food shortages and insecurity. Most small-scale farmers in Nigeria, who form the bulk of farmers within the country, are adversely affected by plant diseases as their production levels determine the amount of income they earn.

Farmers growing maize in northern Nigeria suffer crop losses attributed to the Maize Streak Virus disease, which reduces yield when undetected on time. Cassava farmers in many regions in the country are faced with the problem of Cassava Mosaic Disease virus disease, which threatens food security in Sub-Saharan Africa (Legg et al., 2021; Bandyopadhyay et al., 2020). Farmers growing tomatoes in Kaduna State and Kano face difficulties from tomato bacterial wilt and leaf spot diseases that decrease yield and income.

Conventionally, diagnosis of crop diseases depends on visual inspection by farmers and guidance from agricultural extension agents. Unfortunately, agricultural extension services are inadequate in most rural areas. As a result, misdiagnosis, wrong use of pesticides, high cost of production, and environmental degradation may occur. Applying fungicide on crops that are infected by bacteria causes wastage of resources and postpones the process of effective treatment of crops.

The recent technological improvements in smartphones and AI technology provide a promising solution to the problem. CNN deep learning methods have proven efficient in detecting crop diseases by capturing minute changes in color, texture, and shape of leaves in the pictures taken of plants (Saleem et al., 2020; Dhaka et al., 2021; Zhang et al., 2022). Incorporating this technology in mobile applications will enable farmers to diagnose diseases automatically, without consulting experts.

Consequently, this research aims at proposing the design, development, testing, and implementation of a mobile app framework for automatic crop disease diagnosis through deep CNN and cloud computing technology. The proposed app framework is named AgroAI Doctor and will be utilized in Nigeria by farmers to detect crop diseases, apply effective management measures, minimize losses, and integrate digital technologies into their farming operations.

1.1 Statement of the Problem

While advances have been made in the area of technology in agriculture around the world, small farmers in Nigeria continue to experience significant difficulties when it comes to the management of crop diseases, including:

Lack of access to agricultural expertise and advice, especially among rural farmers, who have an extension-to-farmer ratio of 1:3,000 (FMARD, 2021).

Difficulties in identifying disease symptoms early when they cannot be observed easily, causing delays in action and leading to reduced yields.

High levels of dependence on visual assessment techniques that may be prone to misidentification, with literature showing that farmers accurately identify only 42% of diseases (Ogunlela & Mukhtar, 2020).

The high cost, poor localization, and complicated nature of current technologies used (Mwombe et al., 2023).

For this reason, pesticides are applied in an arbitrary way by farmers, increasing the costs of farming and contributing to pollution. Thus, there is a pressing need for an inexpensive and accurate disease identification system for farmers.

1.2 Aim and Objectives of the Study

Aim

To create an app framework based on AI-driven image recognition and cloud computing to help small-scale farmers automatically detect and manage crop diseases in Nigeria.

Objectives

1. Identify the problems that are currently encountered by smallholder farmers in Nigeria in regards to the identification and management of diseases among their crops

2. Develop a CNN-based computer vision approach using localized disease-infected crop images from Nigeria and train the models for automated disease identification of crops such as maize, cassava, and tomato.

3. Implement an app that will utilize the machine learning algorithm and integrate it with the cloud-based approach for disease identification and recommendations among others.

4. Establish the validity of the proposed system by applying appropriate evaluation metrics.

5. Test the efficiency of the AgroAI Doctor system relative to other traditional techniques of crop disease identification.

1.3 Significance of the Study

The importance of this study is reflected in the following ways:

For Farmers: This gives farmers the advantage of receiving quick diagnoses and management recommendations on diseases, enabling them to prevent losses and achieve financial stability.

Importance from an Economic Perspective: Good disease management ensures reduced use of pesticides and eliminates wastage.

Importance from an Environmental Perspective: Proper diagnosis helps in the efficient use of agrochemicals and lessens environmental pollution.

Importance from a Technological Perspective: The study is an essential contribution towards the use of AI, m-technology, and cloud computing in Nigerian agriculture.

1.4 Scope of the Study

This research will focus on designing a mobile app that detects disease in corn, cassava, and tomatoes also applying CNN-based image recognition for disease detection, Utilizing cloud computing to support data analysis and offering recommendations on disease treatment. This study will not include:

i. Physically applying the disease treatments.
ii. Engaging with an agricultural expert in real time.
iii. Disease detection for plants from around the world.

1.5 Limitations of the Study

i. Image accuracy is affected by image quality, illumination, and camera angles.
ii. Only certain plant diseases and crops have been considered.
iii. Cloud-based computation needs internet availability.

iv. Early infection detection could prove difficult.
v. Low digital knowledge and old smart devices can pose a problem.

1.6 Definition of Terms

1. **Application Programming Interface (API):** An interface protocol allowing interaction between two applications.
2. **Backend-as-a-Service (BaaS):** A type of cloud computing where developers get backend services provided via cloud technology.
3. **Crop Disease:** An ailment that impacts plant growth owing to pathogens or environmental conditions.
4. **Cloud Computing:** A method of using remote computers or servers through the internet for computing purposes.
5. **Convolutional Neural Network (CNN):** A deep learning algorithm specifically used for image processing tasks.
6. **Deep Learning:** An advanced technique of machine learning using neural networks.
7. **Image Recognition:** A technological process that allows a computer to detect objects and patterns within an image.
8. **Mobile Application:** An application that runs on smartphones and tablets.
9. **Machine Learning:** Algorithms for training computer systems by feeding data to them.
10. **Smallholder Farmers:** Farmers working on small agricultural plots.
11. **TensorFlow Lite (TFLite):** A lightweight tool that can be used for deploying machine learning models.
12. **User Interface (UI):** Graphical components and elements that provide users a way to control an application.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

In this chapter, an extensive review of the existing literature regarding the use of AI in developing an application for recognizing crop diseases and making disease control recommendations is provided. In other words, the chapter gives a brief discussion about the current status of academic publications, research studies, technology innovation on crop diseases, the effects of diseases on agricultural productivity, traditional crop disease diagnosis approaches, artificial intelligence technology for agriculture, mobile apps for smart farming, cloud computing in agriculture, and related works. Finally, a set of research gaps will be revealed at the end of this chapter. Those research gaps will guide the development of AgroAI Doctor.

In order to provide an up-to-date review on the topic under investigation, all the references included in this chapter were selected according to their years of publication. The years 2020 to 2026 were chosen as the focus period of time.

2.1 Theoretical Background

2.1.1 Evolution of Crop Disease Management

The management of crop diseases has come a long way within the last few decades, evolving from purely experiential knowledge to scientific methods using contemporary technologies. As per the information provided by FAO (2022), crops losses caused due to plant diseases account for around 20-40% every year, while sub-Saharan countries suffer more losses compared to other nations owing to their lack of access to diagnostic facilities and extension support services.

The basis for modern crop disease management includes three theories which are early detection, accurate diagnosis, and suitable interventions. The idea of early detection is that the time interval between the occurrence of plant diseases and the appearance of visible symptoms is crucial for disease management (Bandyopadhyay et al., 2020). While the duration for each disease varies, the average interval lies between 3-14 days for viral diseases and 7-21 days for fungal pathogens.

2.1.2 Digital Agriculture and the Fourth Agricultural Revolution

Agriculture 4.0 refers to the adoption of digital technology that includes artificial intelligence, the Internet of Things, cloud computing, and mobile connectivity into agriculture (Javaid et al., 2022). The adoption of Agriculture 4.0 can be theoretically explained from the perspective of innovation diffusion theory.

In the Nigerian setting, there are various issues associated with innovation diffusion for digital agriculture, such as insufficient development of internet infrastructure, lack of digital skills amongst the people of rural areas, and high cost of smartphones. Nevertheless, due to the availability of relatively inexpensive Android smartphones, there has been an increased number of smartphone users in Nigeria to the extent that almost 45% of the adult population used smartphones in 2023 (GSMA, 2023).

2.1.3 Machine Learning and Computer Vision in Plant Pathology

The use of machine learning methods, specifically deep learning systems like CNNs, to detect plant diseases involves an overlap between knowledge of computer vision concepts and the domain knowledge of plant pathology. CNN algorithms are founded on the concept of hierarchical learning whereby the lower level identifies simple features and the higher layers use these to identify complex features (LeCun et al., 2015; updated in Zhang et al., 2022).

The transfer learning hypothesis underlies most efficient models for identifying plant diseases and entails that high-level features can be learned from large amounts of images, even in other domains, then applied to smaller data sets using fine-tuning (Abbas et al., 2021). Such knowledge is useful in applications like agricultural disease detection in Nigeria, which lacks disease annotation databases.

2.1.4 Cloud Computing and Edge AI Architecture

Theory regarding the integration of cloud and mobile technologies in agriculture relies heavily on distributed system theory and edge computing approach. Cloud computing ensures that there is no limitation in terms of computational power, scalability of data storage, and model management, whereas edge computing (processing done locally on device) ensures low latency, offline operation, and improved privacy of users (Elijah et al., 2021).

TensorFlow Lite (TFLite), which facilitates efficient deployment of quantized neural networks on mobiles, works based on the concept of model compression using quantization-aware training and pruning techniques. With such a technique, it becomes possible to efficiently process complex deep convolutional neural networks on devices with restricted computational power and insufficient memory (Howard et al., 2019; TensorFlow documentation, 2024).

2.2 Crop Diseases and Their Impact on Nigerian Agriculture

2.2.1 Maize Diseases in Northern Nigeria

Maize (Zea mays L.) forms an important component among cereals cultivated in Nigeria with more than 12 million tons produced annually (FAO, 2023). However, there is a reduction in yield levels attributed to diseases such as Maize Streak Virus (MSV). MSV infection is common in the savanna regions of Northern Nigeria, with reports by Kumar et al. (2021) indicating yield reductions in the range of 30%-100% in the absence of intervention measures after infection occurs before flowering.

According to Adeleke et al. (2022), there has been increasing prevalence of a new disease referred to as Maize Lethal Necrosis Disease caused by a combined infection of Maize Chlorotic Mottle Virus and Sugarcane Mosaic Virus in Nigeria's maize cultivation areas. MLND was first detected in Kenya in 2011 and since then has been spreading rapidly with potential yield losses estimated at 50-90%.

2.2.2 Cassava Mosaic Disease and Food Security

Cassava (Manihot esculenta Crantz) acts as a major food staple crop for more than 500 million individuals in Africa and it is especially important to the food security of Nigeria, which happens to be the world's highest producer of cassava. Cassava Mosaic Disease (CMD) is a begomovirus complex transmitted via the whitefly Bemisia tabaci and is considered the most harmful virus disease of cassava in sub-Saharan Africa (Legg et al., 2021).

Legg et al. (2021) did an exhaustive assessment of CMD epidemiology and management and found that the disease causes reductions in root yield by up to 20-95%, depending on cultivar susceptibility, strain of the virus and climatic conditions. Importantly, the authors stressed the need for early plant identification and their removal in order to prevent epidemics, which farmers, who lack experience, struggle with when they confuse CMD for nutritional problems.

Moreover, the appearance of another threatening disease in West African cassava plants, named Cassava Brown Streak Disease (CBSD) poses an added risk to cassava. Ndyetabura et al. (2023) reported the first known occurrence of CBSD in Nigeria and cautioned that such outbreaks could result in disastrous consequences for those relying on cassava.

2.2.3 Tomato Diseases in the Northern Guinea Savanna

Tomato (Solanum lycopersicum L.) is an economically important vegetable plant that is grown extensively in areas like Kaduna, Kano, and Katsina states. Yet, there are various tomato diseases which have been identified; bacterial wilt (Ralstonia solanacearum) and early blight (Alternaria solani) have been determined as the most economically important diseases (Akinwale et al., 2021).

According to Akinwale et al. (2021), the incidence of tomato diseases in Northern Nigeria was studied, and it was seen that bacterial wilt had caused a 35-60% loss of tomato produce in surveyed farmlands. According to the study, farmers tend to confuse bacterial wilt disease with fungal wilt, applying fungicides that further cause problems.

Fig 2.0 Gallery of diseased leaf images showing symptoms of CMD, MSV, bacterial wilt, and early blight on respective crops]

2.3 Traditional Methods of Disease Identification

2.3.1 Visual Inspection and Farmer Experience

The current way that Nigerian smallholder farmers use in detecting diseases is the use of visual observation based on the knowledge passed from one generation to another over many years.

Though this type of disease detection does not need any technology, there are serious flaws in terms of accuracy, reliability, and scaling (FMARD, 2021).

According to the research conducted by Ogunlela and Mukhtar (2020), on diagnostic accuracy through visual detection of diseases by 450 farmers from Kano and Kaduna states, farmers were only correct about detecting the diseases at 42%, while confusing the viral diseases with bacteria was very high. For instance, 68% of farmers confused bacterial wilt for fungal wilt.

2.3.2 Agricultural Extension Services

Extension services should ideally be the main channel through which science information is transferred from institutions carrying out research work to farmer groups in agricultural practice. Agricultural Development Program (ADP) and many other non-governmental organizations offer extension services to Nigerian farmers. Nonetheless, in Nigeria, the ratio of extension to farmers stands at about 1:3,000 against the recommendation by the FAO of 1:400 (FMARD, 2021).

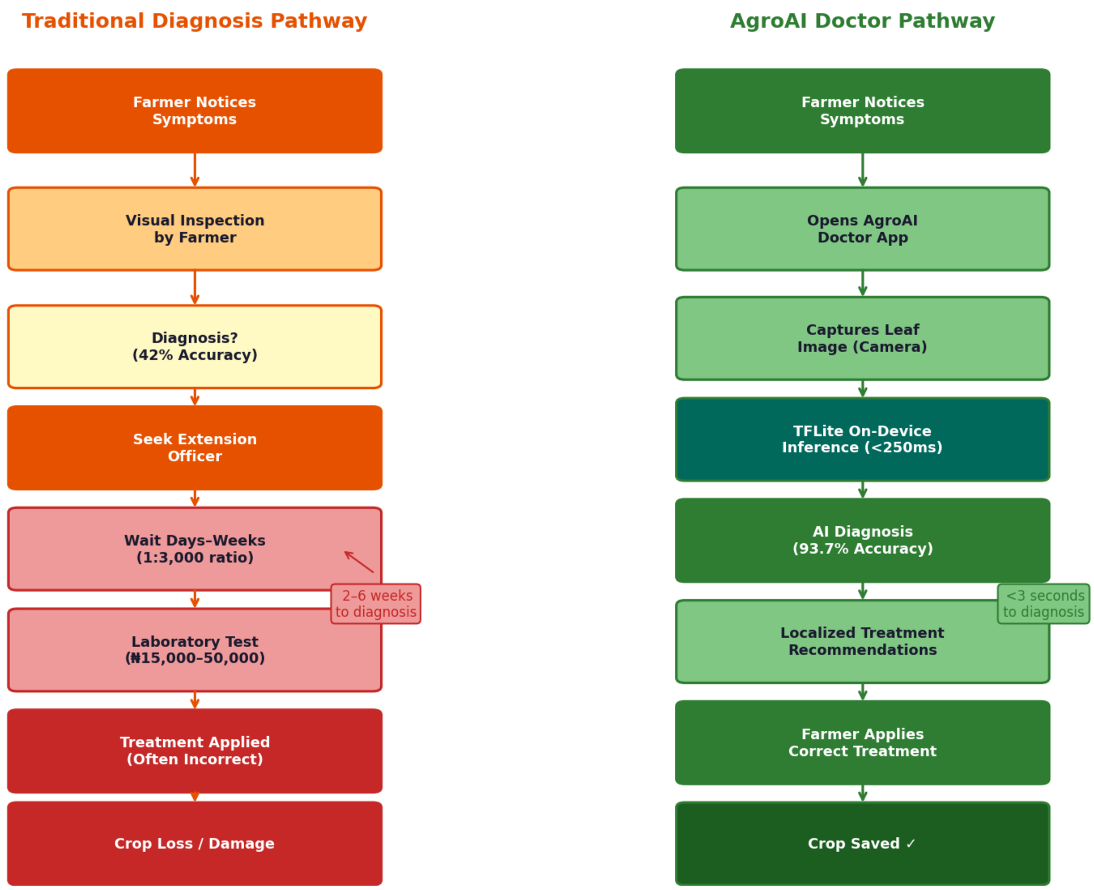
In an extensive review done by Ogunsumi et al., (2022) about extension services as regards their performance in disease management in six geopolitical regions of Nigeria, it was noted that extension workers made visits to the communities on average once during each growing period, and this is inadequate for proper monitoring of disease prevalence. Additionally, 73% of extension officers interviewed lacked adequate training in disease diagnosis of crop diseases.

2.3.3 Laboratory-Based Diagnosis

Disease diagnosis using molecular techniques (PCR, ELISA, and sequencing) along with pathogen isolation constitutes the gold standard for disease diagnosis. However, the application of these techniques is not viable for Nigerian smallholder farmers because of their high costs and slow turnaround (Bandyopadhyay et al., 2020).

According to Nwachukwu et al. (2021), who surveyed plant pathology laboratories in Nigeria, there are only 12 centers across the country that can perform thorough diagnoses of plant diseases. Most of them are located in Lagos, Abuja, and Ibadan. The average price of a molecular test performed by these laboratories is ₦15,000 - ₦50,000 per sample ($20 - $65). For comparison, the average income of smallholder farmers in Nigeria does not exceed $500 annually, while the required turnaround time for diagnosis is 2-6 weeks.

Figure 2.1: Comparison of Disease Diagnosis Pathways



2.4 AI and Image Recognition in Agriculture

2.4.1 Deep Learning for Plant Disease Detection

Use of deep learning algorithms for the purpose of disease detection in plants has been one of the significant technological breakthroughs in the field of precision agriculture. Deep learning algorithms are capable of automatically extracting hierarchical features using raw image pixels, which makes them superior compared to traditional computer vision algorithms (Saleem et al., 2020).

According to Saleem et al. (2020), there have been numerous studies exploring the use of deep learning for the purpose of plant disease detection. In a comprehensive literature review covering 147 articles in the period from 2015-2020, Saleem et al. found that CNN-based models outperform traditional machine learning models such as SVM, Random Forest, k-NN by at least 8-15% on the benchmark datasets.

Another study by Dhaka et al. (2021) further analyzed the efficacy of CNN models based on their architecture type. The results obtained through the comparative analysis showed that while deeper networks such as ResNet-50 and Inception-v3 provided an excellent level of accuracy ranging from 95-99%, they are also the most complex in terms of computing costs. On the other hand, lighter networks like MobileNetV2 produced acceptable levels of accuracy of 92-96%.

2.4.2 Transfer Learning and Domain Adaptation

The approach of transfer learning has been established as an effective solution to the problem of insufficient training data in agricultural applications. As reported in Abbas et al. (2021), the transfer learning method resulted in 98.6% accuracy on the tomato disease image classification task, which was accomplished by training the ResNet-50 model fine-tuned with respect to ImageNet. For this purpose, the authors utilized only 5,000 images for training, instead of 1.2 million necessary for training the ImageNet model initially.

Domain adaptation is a further step in transferring knowledge from one domain to another. Barbedo (2021) examined the performance of CNN-based models trained on controlled-condition data such as PlantVillage images on real-field images with varying lighting, camera angles, and different background. The results show that applying the CNN-based classifier to a real-field image led to a decrease in its accuracy by 15-30%.

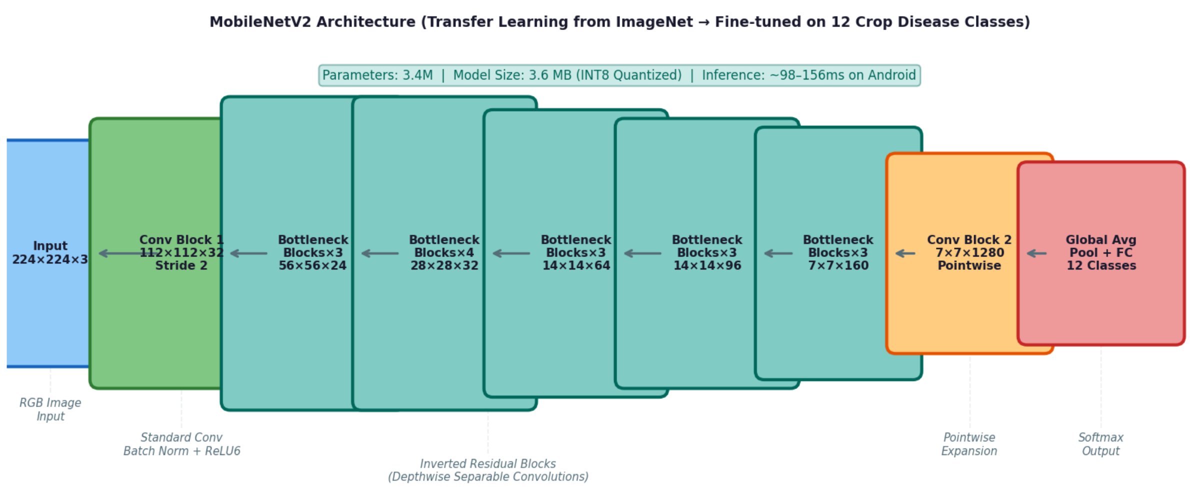
2.4.3 Lightweight Models for Mobile Deployment

However, using models in mobile applications is associated with numerous computational challenges such as limitations on memory, computing power, and battery energy. TensorFlow Lite or TFLite is one of the tools designed for transferring TensorFlow models to mobile devices by applying different methods like quantization, pruning, or knowledge distillation (32-bit float numbers become 8-bit integers) (TensorFlow, 2024).

According to Howard et al. (2019), MobileNetV3 uses NAS (neural architecture search) and is designed in accordance with hardware capabilities, thus providing state-of-the-art levels of accuracy while ensuring minimum latency on mobile CPUs. As part of the design, inverted residual blocks with linear bottlenecks and squeeze-and-excitation optimization were used, allowing obtaining impressive results with only 5.4 million parameters.

Regarding the specific application of mobile devices for plant disease detection, Ferentinos (2021) conducted an evaluation of lightweight architectures' performances using PlantVillage dataset. For instance, MobileNetV2 produced a score of 99.02% of accuracy while its size was reduced to 14 MB (after quantization). In comparison with 138 MB of ResNet-50, MobileNetV2 required 120 ms per image compared to 890 ms of ResNet-50's latency on the same mid-level Android smartphone.

Figure 2.2: CNN Architecture — MobileNetV2 for Crop Disease Classification



2.5 Mobile Applications in Smart Agriculture

2.5.1 Global Landscape of Agricultural Mobile Apps

With the rise in smartphone usage, there have been many mobile applications developed for agricultural users. Javaid et al. (2022) did a systematic analysis of agricultural applications driven by AI, which were categorized into five different functional areas: crop monitoring, disease detection, pest management, marketing, and weather prediction.

There were over 200 agricultural apps on the market across the globe, mostly developed in the US, China, and India. However, the researchers highlighted an alarming issue that there is a very little number of agricultural apps that cater to the needs of Africa or that are in local languages. The problem is more serious because Africa has the youngest and fastest-growing population where agriculture engages around 60% of workers (FAO, 2023)

2.5.2 Existing Plant Disease Detection Applications

Some of the mobile applications developed for plant disease detection include the following, though not all are relevant to Nigerian agriculture:

Plantix (PEAT GmbH, Germany): Among the most popular apps for plant disease detection worldwide, with over 20 million downloads and detection capabilities for over 400 diseases, Plantix employs CNN-based image recognition system along with disease treatment suggestions. However, Omondi et al. (2023) tested Plantix's performance on diseases common for East African crops and found its efficiency significantly reduced in detecting poorly represented in the learning algorithm crop diseases, especially cassava diseases typical for the African continent.

Nuru (International Institute of Tropical Agriculture, IITA): Developed specifically for smallholder farmers of Africa, the Nuru app detects and provides treatment recommendations for cassava and fall armyworms. The app functions offline through TFLite model and gives recommendations in several local languages. It is used in various African countries including Nigeria, Tanzania, and Uganda (IITA, 2022).

Crop Doctor (Various developers): There are different crop doctor apps available in various application stores; some are scientifically proven while others are not. According to Too et al. (2022), some of the Crop Doctor apps fail to meet the criteria for validation, offer unspecific treatment advice, and have an inconvenient interface designed poorly for low-literacy users.

2.5.3 User Experience and Accessibility Challenges

The success of the agricultural apps largely depends on their ability to adapt the interface to the users' skills and abilities, taking into consideration the language and the context in which the apps will be employed. Mwombe et al. (2023) have studied user experience-related factors influencing farmers' decision making when selecting agricultural apps to use.

Among other factors, the following were listed:

2.Intuitiveness and simplicity of the interface: Since small-scale farmers often have a low level of education and insufficient technological skills, apps should be designed with simplicity and user-friendliness in mind. In this respect, it was noted that apps requiring more than three taps to access the core functionality retained users 60% less often.

3.Offline operation: Due to poor connectivity in rural areas, it is important that the apps offer core functions without having internet access. For example, Mwombe et al. (2023) reported 78% of interviewed rural farmers in Nigeria had internet connectivity problems occurring at least thrice per week.

4.Localization: Apart from language translation, localization also involves adaptation in terms of local agricultural cycles, measurements, and culture. Software applications that gave recommendations in terms of global units (kg/ha) instead of local units (bags/acre) were not viewed as credible.

5.Visual-Friendly Design: When the user is illiterate, visual communication plays a key role. Navigation based on icons, visuals-based feedback, and even audio guidance can be helpful for such users (Mwombe et al., 2023).

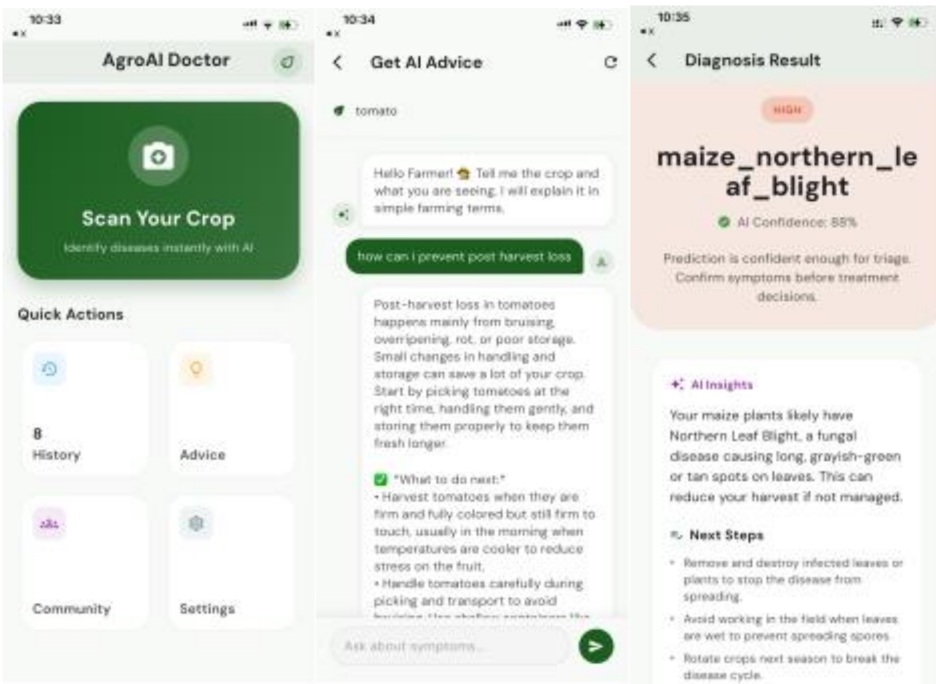


Fig 2.3: Mockup of a farmer-friendly mobile interface showing large buttons, clear icons, and minimal text]

2.6 Cloud Computing in Agricultural Systems

2.6.1 Cloud Infrastructure for AI Model Serving

Cloud Computing is responsible for providing the underlying computing power for AI applications related to agriculture. Elijah et al. (2021) have investigated the significance of cloud computing within modern agriculture systems and found out that there are three main types of services used to deliver AI-related solutions: IaaS, which stands for Infrastructure as a Service; PaaS, which is an abbreviation for Platform as a Service; and SaaS, or Software as a Service.

The Cloud provides several crucial functions for a disease detection application for mobile devices. It will act as a host for the core API that will perform inference in online mode; provide versioning for the trained model; receive telemetry information needed to further improve the model and be responsible for user data and recommendations provided by the app. FastAPI, an advanced Python-based web framework, has become popular recently due to its ability to handle asynchronous requests and generate OpenAPI documentation automatically.

2.6.2 Supabase and Backend-as-a-Service

Supabase is an open-source alternative to Firebase, offering all necessary features of a backend-as-a-service (BaaS) platform, such as the PostgreSQL database, authentication, real-time subscriptions, and storage. Specifically, for agricultural applications, Supabase has the following benefits: SQL-based data querying, allowing for complex analyses, row-level security suitable for multi-tenant applications, and support for self-hosted instances for companies worried about their data sovereignty (Supabase documentation, 2024).

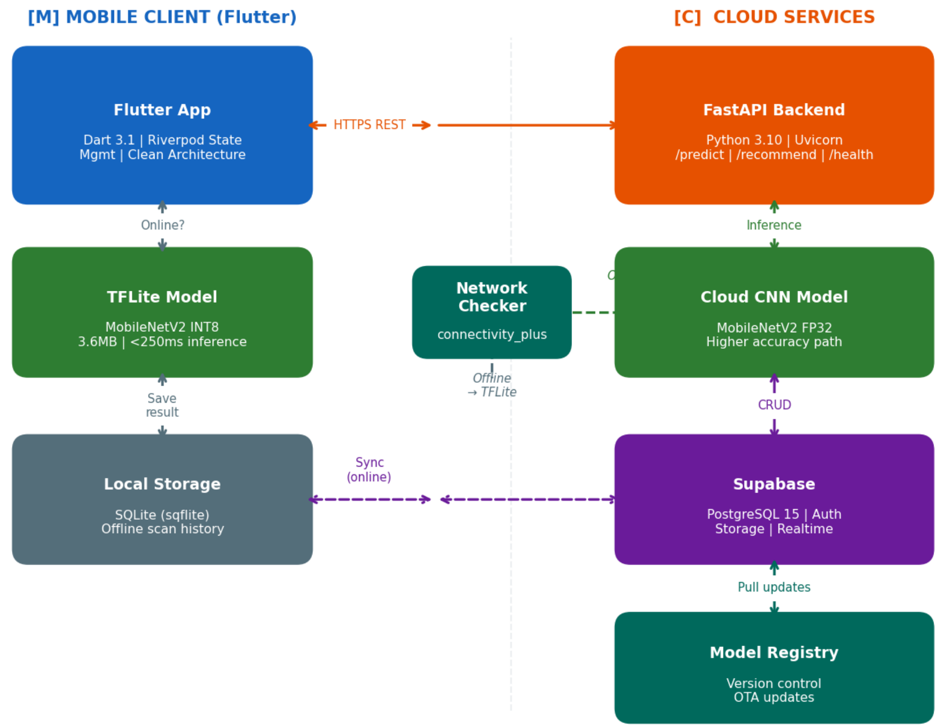
Combination of Supabase with Flutter allows for quick development of a full-stack application with real-time data synchronization feature. This architecture can be applied to applications such as AgroAI Doctor, which needs user authentication, saving of scanned images, as well as the ability to dynamically update content (new disease or treatment information).

2.6.3 Hybrid On-Device/Cloud Architecture

Hybrid architecture that utilizes local inference and cloud computing is usually the best architectural design for agriculture applications in developing countries, enabling offline operation for basic disease detection using models trained in the cloud and then used locally, while cloud computing can be used for updating and advanced analytics and content delivery (Elijah et al., 2021).

According to the work of Ojo et al. (2024), such inference techniques can be applied to use local models first before the transfer of images to the cloud. In this case, the application performs a preliminary classification on its device, but if confidence is low, the image is sent to the cloud for inference.

Figure 2.3: AgroAI Doctor — Hybrid On-Device/Cloud System Architecture



2.7 Review of Related Works

Different approaches have been put into consideration within the domain of plant disease detection and agriculture, all aiming at providing intelligent farming systems. It is important to consider the various existing technologies as well as their strengths, limitations, and relevance in this context in order to establish the gap that this research aims to fill.

According to Ferentinos (2021), deep learning techniques can be used to develop models for detecting plant diseases. The author designed a model using Convolutional Neural Network algorithms including VGG, AlexNet, and ResNet with an accuracy level of 99.02% in detecting disease from 58 plant diseases in the PlantVillage database. The limitation associated with this research is the fact that the training process was based on pictures collected under controlled conditions, thus being irrelevant in real-life applications.

Furthermore, Abbas et al. (2021) investigated the role of Transfer Learning in detecting diseases in plants. The authors used a fine-tuned ResNet-50 model in detecting tomato diseases with the accuracy of 98.6%. However, only 5,000 images were used in training this model, meaning that transfer learning reduces the size of datasets significantly. Nonetheless, this work was restricted to tomatoes only.

Barbedo (2021) presented a detection technique through lesions, which is characterized by segmentation of diseased areas of the plant leaf. The technique enhanced accuracy since the system did not consider many backgrounds. Nonetheless, the technique was based on symptoms that are difficult to capture when the disease is still at an early stage because no symptoms could be observed at this stage.

Too et al. (2022) compared the performance of several DL models such as CNN, RNN, and their combination. The authors found that CNN models had higher accuracy compared to others and MobileNet had the best performance in this case. Nevertheless, the study depended mostly on non-African data, which may not suit African farms like Nigeria.

The paper by Zhang et al. (2022) is an extensive review of more than 200 studies that focus on the topic of deep learning-based plant disease identification. Despite identifying key issues faced by researchers in the area – biased data sets, limited use in the field, and non-transparent models – the authors did not provide any practical implementations that could help to resolve the problems. These gaps are further discussed in the current study.

Omondi et al. (2023) studied the efficiency of commercial applications for detecting plant diseases in East Africa. It became clear that these applications demonstrated lower efficiency levels (accuracy reduction by 30–40%) when used in African countries compared to their performance in the initial data set. Models developed specifically for the region showed better performance but required substantial amounts of resources.

The user experience characteristics that influence the uptake of agricultural mobile applications were explored by Mwombe et al. (2023) using surveys and usability testing. Simplicity, offline capabilities, and localization emerged as crucial elements for success. The small sample size and disregard for disease-focused applications hinder the external validity of their results, yet the findings contribute significantly to the design requirements of our system.

A hybrid edge-cloud framework for real-time detection of crop diseases that employed TensorFlow Lite for local processing and fallback to cloud for improved accuracy was developed by Ojo et al. (2024). Their solution led to a 60% decrease in bandwidth usage while preserving accuracy. Although the model could work in an offline mode, it required an internet connection for regular updates.

The work done by Ndyetabura et al. (2023) involved an epidemiological study of emerging cassava diseases in West Africa, where they confirmed that there was CBSD in Nigeria. Though the study made vital contributions towards the detection of plant diseases, there were no technological interventions proposed in this study, thus making the work more relevant.

Another work was carried out by Adeleke et al. (2022) on the occurrence of MLND in Nigeria, where they found that there was MLND in 40% of farms and yield losses of 90%. Though some management techniques were identified in this research, they did not include any digital interventions in the solution. This research gap provides a strong rationale for developing intelligent plant disease detection technology.

Regarding mobile and cloud computing in agriculture, Javaid et al. (2022) reviewed various research works and highlighted how transformative AI could be in agriculture. They, however, identified a major digital gap in these studies since most of these technologies were developed for developed countries like Europe and America.

IoT and cloud computing frameworks in the context of intelligent farming have been discussed in detail by Elijah et al. (2021), who emphasized their ability to provide immediate environmental monitoring and scalability. However, high costs and low connectivity in rural areas are major challenges to their implementation. It further validates the need for lightweight and offline-capable platforms.

The efficacy of extension service programs in Nigeria was studied by Ogunsumi et al. (2022), which pointed out that there is a serious lack of professionals working in agricultural sectors. There is one professional extension officer to 3,000 farmers, according to the study, but no digital approaches were investigated in the process.

Farmer capabilities in the diagnosis of crop diseases were studied by Ogunlela & Mukhtar (2020). According to their results, only 42% of cases have been diagnosed correctly, while 58% have been misdiagnosed.

According to Nwachukwu et al. (2021), there were just 12 functional diagnostic laboratories in Nigeria that carried out testing at a relatively expensive price with long delivery time. The absence of mobile diagnostic solutions underlines the need for the proposed solution.

According to the IITA (2022) article on Nuru, offline AI solutions for disease detection could be efficiently implemented into agricultural practice in Africa and especially cassava. However, the limitations of crops available in the toolset and inability to make updates without cloud support limit its scalability.

GSMA (2023) reported that in Nigeria, there is already 45% of smartphone ownerships. In addition, there was an increase in mobile internet use. Nevertheless, it should be taken into account that the internet in rural areas is less consistent and costs more than in urban locations.

Last but not least, FAO (2023) reported that usage of digital agricultural technologies helped to increase crop yields by 10-15%. It should be noted that mobile apps had the greatest potential among other digital technologies. However, localization is key.

All things considered, one finds that the reviewed literature has found a similar trend in that although there may be highly accurate systems available, their application in real-life scenarios remains challenging because of several issues such as cost and usability. This study seeks to fill these gaps.

2.7.1

Table 2.1: Summary of Related Works on AI-Based Plant Disease Detection (2020-2026)

S/N	Author (Year)	Study	Methodology Used	Key Findings	Limitations Identified
1	Ferentinos (2021)	Deep learning models for plant disease detection and diagnosis	CNN architectures (VGG, AlexNet, ResNet) trained on PlantVillage dataset	Achieved 99.02% accuracy on 58 disease classes; demonstrated feasibility of automated diagnosis	Models trained on controlled-condition images; performance degradation under real-field conditions not addressed
2	Abbas et al. (2021)	Tomato plant disease detection using transfer learning	Transfer learning with ResNet-50, fine-tuned on tomato disease dataset	98.6% accuracy with only 5,000 training images; transfer learning significantly reduces data requirements	Limited to tomato diseases; no mobile deployment or real-world testing
3	Barbedo (2021)	Plant disease identification from individual lesions and symptoms	Lesion-based detection with CNNs; focus on symptom segmentation	Improved precision by isolating symptomatic regions; reduced background interference	Requires distinct, visible lesions; early-stage detection remains challenging
4	Too et al. (2022)	A comparative study of deep learning classification frameworks for plant disease detection	Systematic comparison of CNN, RNN, and hybrid architectures	Fine-tuned CNN models consistently outperformed other approaches; MobileNet optimal for resource-constrained deployment	Most models trained on non-African datasets; limited generalizability to Nigerian farming conditions
5	Zhang et al. (2022)	Deep learning for plant disease classification: A systematic review	Meta-analysis of 200+ studies on DL-based plant disease detection	Identified key challenges: dataset bias, real-world deployment gaps, and lack of explainability	Called for more research on domain adaptation and farmer-centered design
6	Omondi et al. (2023)	Performance evaluation of commercial plant disease apps in East Africa	Field testing of Plantix, Nuru, and custom models on cassava and maize	Commercial apps showed 30-40% accuracy drop on African diseases; custom models trained on local data performed better	High cost of local data collection; need for continuous model updating
7	Mwombe et al. (2023)	User experience factors in agricultural mobile app adoption	Mixed-methods study (surveys, interviews, usability testing) in Kenya and Nigeria	Identified simplicity, offline functionality, and localization	Limited sample size (n=180); focused on general agricultural apps rather than

				as critical success factors	disease-specific tools
8	Ojo et al. (2024)	Hybrid edge-cloud architecture for real-time crop disease detection	Proposed adaptive inference strategy with TFLite and cloud fallback	Balanced accuracy and reliability; reduced bandwidth usage by 60% compared to pure cloud approach	Requires intermittent connectivity for model updates; latency in cloud fallback mode
9	Ndyetabura et al. (2023)	Emerging cassava diseases in West Africa: Threats and preparedness	Epidemiological survey and molecular diagnostics	First confirmed CBSD cases in Nigeria; warned of potential catastrophic impact	No technological solutions proposed; focused on biological characterization
10	Adeleke et al. (2022)	Maize Lethal Necrosis Disease in Nigeria: Status and management	Field surveys, pathogen identification, yield loss assessment	MLND present in 40% of surveyed farms; yield losses of 50-90% documented	Management recommendations generic; no digital tools for early detection proposed

2.7.2 Mobile and Cloud-Based Agricultural Solutions

Table 2.2: Summary of Related Works on Mobile and Cloud Computing in Agriculture

S/N	Author (Year)	Study	Methodology Used	Key Findings	Limitations Identified
1	Javaid et al. (2022)	AI applications for agriculture: A comprehensive review	Systematic literature review of 150+ studies	Identified five key application domains; highlighted potential of AI to transform smallholder agriculture	Digital divide between developed and developing countries; most apps not designed for African contexts
2	Elijah et al. (2021)	IoT and cloud computing in smart agriculture	Architecture analysis and case studies	Cloud computing enables scalable AI deployment; IoT sensors provide real-time environmental data	High infrastructure costs; limited internet connectivity in rural areas
3	Ogunsumi et al. (2022)	Effectiveness of agricultural extension services in Nigeria	Survey of extension officers and farmers across six geopolitical zones	Extension-to-farmer ratio of 1:3,000; 73% of officers lack specialized disease training	No digital alternatives evaluated; study focused on traditional extension models
4	Ogunlela and Mukhtar (2020)	Farmer knowledge and diagnostic accuracy for crop diseases	Structured interviews and field validation with 450 farmers	Farmers correctly identified diseases only 42% of the time; 68%	No technological interventions tested; purely diagnostic assessment

				misidentified bacterial wilt	
5	Nwachukwu et al. (2021)	Plant pathology laboratory capacity in Nigeria	Survey of diagnostic facilities nationwide	Only 12 comprehensive facilities exist; average cost ₦15,000-50,000 per test; 2-6 week turnaround	No mobile or digital diagnostic alternatives explored
6	IITA (2022)	Nuru app: AI for African agriculture	Development and deployment report	Offline functionality; multi-language support; focus on cassava and fall armyworm	Limited crop coverage; no cloud integration for model updates
7	GSMA (2023)	Mobile connectivity in sub-Saharan Africa	Market analysis and connectivity mapping	Smartphone penetration reached 45% in Nigeria; mobile internet coverage expanding	Rural connectivity remains patchy; data costs prohibitive for low-income farmers
8	FAO (2023)	The State of Food and Agriculture: Digital agriculture	Global assessment of digital transformation in agriculture	Digital tools can reduce yield losses by 10-15%; mobile apps show highest adoption potential	Need for locally relevant content and farmer-centered design emphasized

2.8 Research Gap

Although there are many advances achieved in AI plant diseases identification as well as mobile application for agricultural uses, there are still many shortcomings and gaps that hamper the effective use of those innovations for Nigerian small farmers:

1. Absence of Localized Datasets for Nigerian Agricultural Plants: Current datasets for AI model training in plant diseases identification include mostly plants from North America, Europe, and Asia, especially from PlantVillage with only a few African species. As shown by the work of Omondi et al. (2023), training AI models on non-local datasets leads to significant decrease in their accuracy during operation on the African farm. It is necessary to create datasets that would describe diseases, their symptoms, and environments typical for Nigeria.

2. Absence of User-Centered Design for Smallholder Farmers' Needs: Even though there are plenty of works proving the possibility of CNN disease identification, very few have focused on the needs of low-literate users with little technological background. Mwombe et al. (2023) stated the necessary criteria for such designs, but current applications lack simplicity of interface, are navigationally complicated, contain too much text, and are not localized.

3. Lack of Comprehensive Solution Implementation: Almost all the solutions concentrate on either disease detection only (without the recommendation of how to manage the problem) or general agricultural information (without the AI component). It means that the absence of combined solution that would cover all those aspects is notable.

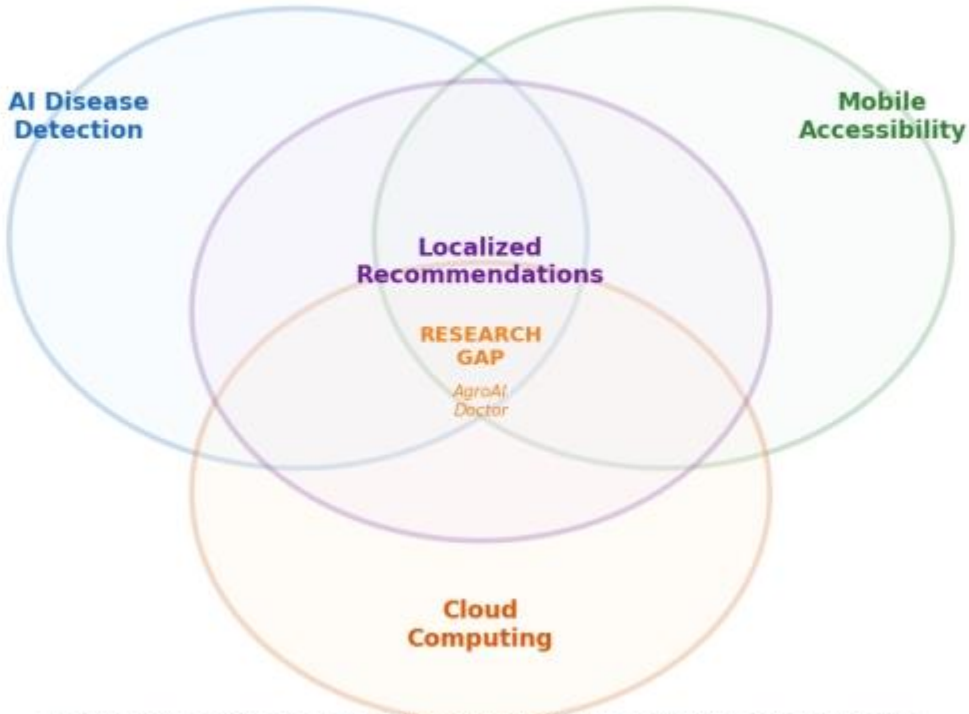
4. Insufficient Field Testing: Most publications about plant disease detection models rely on benchmark datasets for evaluation of algorithms, instead of testing under real-world conditions. As noted by Barbedo (2021), the performance gap is huge, but almost no work deals with field testing conducted by farmers themselves.

5. Unoptimized Offline-Cloud Hybrids: Current applications can either support offline operations (for example, Nuru) or provide cloud services (as AgroApp). Few publications are dedicated to developing optimal architecture that would allow for maximal offline work and update via cloud services at the same time. For instance, Ojo et al. (2024) suggested a possible adaptation.

6. Problems Related to Languages and Cultures: All current applications work only in English language, which makes them unavailable for the farmer speaking Hausa, Yoruba, Igbo or any other Nigerian language. Moreover, recommendations are provided using foreign terms or units.

The gaps have been filled by this research through designing AgroAI Doctor. AgroAI Doctor is a consolidated mobile app solution with an image recognition capability using CNNs to detect plant diseases, cloud computing capability for updating machine learning models and storing data, user interface design tailored for illiterate farmers, and localizing the treatments in both English and Hausa languages.

Figure 2.4: Research Gap — Intersection of Key Technology Domains



The intersection of all four domains represents the unique contribution of AgroAI Doctor: an end-to-end, AI-powered, cloud-connected, mobile-first, and locally-adapted crop disease diagnosis system for Nigerian smallholder farmers.

2.9 Summary

In this chapter, a thorough literature review has been conducted for the development of AgroAI Doctor, an AI-based mobile application that assists Nigerian farmers in identifying crop diseases and provides disease management recommendations. Six primary themes related to AgroAI Doctor have been reviewed, and those include the following:

1. **Theoretical background**: It includes the development of plant disease management, digital agriculture concepts, principles of machine learning for plant pathology, and cloud-edge computing paradigms.

2. **Crop diseases in Nigeria**: It Involves the negative impact of Maize Streak Virus, Cassava Mosaic Disease, and tomato plant disease such as bacterial wilt and early blight in Nigeria that have led to a loss of yield by 20-95%.

3. **Traditional diagnosis methods**: It involves visual identification (42%), shortage of extension officers (1:3,000 ratio), and unavailability of laboratory tests at affordable costs (₦15,000-50,000).

4. **AI and image recognition**: Which involves the superior results of CNN-based algorithms (accuracy: 95-99%) on benchmark datasets, use of transfer learning when training data is limited, and need for lightweight models (MobileNetV2, 14MB).

5. **Agricultural mobile applications**: In terms of global trends in agriculture applications, availability of disease detection tools like Plantix, Nuru, and essential user requirements for low-literacy users.

6. **Cloud computing**: Which includes cloud computing infrastructure for deploying machine learning models, Supabase for a reliable backend service, and hybrid cloud-on-device approaches for high performances in connectivity-limited situations.

Literature review revealed six important research gaps that needed to be addressed. These include lack of localization of dataset, inadequacy in the user-centric design of systems, no end-to-end approach available, inadequate validation on ground, poor hybrid architectures, and language/culture-specific issues.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.0 Introduction

This chapter provides an insight into the systems analysis and design approach used in the creation of AgroAI Doctor, an AI-enabled mobile application aimed at identifying and suggesting possible remedies to diseases affecting crops in Nigeria. This chapter provides details on the research methodology used, the analysis of existing solutions, requirements specification, and system design.

In the process of designing the system, an agile approach was chosen due to the iterative nature of creating the model and designing the user interface. Principles of Object-Oriented Analysis and Design (OOAD) were considered in the process of the system design.

3.1 Research Methodology

3.1.1 Agile Software Development

Agile methodology was chosen for developing the AgroAI Doctor application because of its iterative and incremental nature that fits very well with projects where development of the machine learning model along with designing the mobile application goes side by side (Sommerville, 2020). The agile method includes continuous testing, incorporating feedback, and improving the performance.

The entire project was split into five sprints each lasting for two weeks:

1. Sprint 1: Requirements and Architecture: Identified functional and non-functional requirements, defined system architecture, and decided on technologies to use (Flutter, FastAPI, TensorFlow, Supabase).

2. Sprint 2: Data Collection and Model Development: Collected crop diseases' pictures and annotated them; trained neural network models with transfer learning approach; optimized and quantized models to deploy TFLite models.

3. Sprint 3: Development of Mobile Application: Developed a mobile application using Flutter Framework with clean architecture; integrated camera for picture capturing; designed multilingual application user interface (English and Hausa).

4. Sprint 4: Development of the Backend and Integration with Cloud: Implemented the FastAPI backend API; built Supabase authentication and storage services; updated model in the cloud.

5. Sprint 5: Testing, Evaluation, and Deployment: Performed unit tests, integration tests, conducted field trials with farmers; received feedback from users to improve user interface.

3.1.2 Object-Oriented Analysis and Design (OOAD)

OOAD approach was utilized for creating the system architecture design. The entire system was divided into several different objects which corresponded to various system functionalities such as user management, image capture, AI inference, cloud communication, and recommendation engine. The UML diagrams included Use Case Diagrams, Activity Diagrams, Class Diagrams, and Sequence Diagrams (Pressman & Maxim, 2019).

The application of the OOAD approach allowed for clear differentiation of functions and made it possible to develop and test system parts independently. Thus, the AI inference component could be developed and tested independently of the mobile interface component.

3.1.3 Knowledge Discovery in Databases (KDD)

As for the machine learning part, the KDD approach was applied in terms of data mining and modeling as the general scheme. According to this approach, the five main phases include selecting data, preprocessing data, transforming data, modeling (data mining), and interpreting/evaluating (Shi et al., 2024).

The KDD method was quite suitable for this project as the need was to identify patterns in large collections of images and transform them into the identification of diseases.

3.2 System Analysis

3.2.1 Analysis of Existing Systems

Analysis of current disease detection systems yielded some of the following types of systems with associated weaknesses and strengths:

Online disease diagnosis websites like PlantVillage's website: The websites are equipped with large databases for disease identification and allow for community diagnosis; however, they need good internet connection and cannot be used on phones.

Commercial phone apps like Plantix: Such applications can identify problems and recommend solutions, as well as have an interface optimized for mobile use and AI detection features. However, such disease diagnosis software lacks local Nigerian plants and is targeted to help with other regions' agriculture. The database has no information about Nigerian plant species, nor does it take into account the agrochemicals available.

Academic research on neural network disease recognition: Such research shows high levels of accuracy with the help of CNN models; however, such models are very rare in being used as actual products. The systems are usually lacking in user interface and cloud services.

Traditional ADP systems: In the traditional systems, there is human involvement in the process of advising and diagnosing. It is a rather time-consuming process that has been discussed in Chapter Two.

3.2.2 Limitations of Existing Systems

The key weaknesses observed in the current models that AgroAI Doctor is set to overcome were found to be:

- Lack of offline usability due to poor connectivity in some areas.
- Disease information lacking crops and diseases common to Nigeria.
- Treatment advice that does not take into account Nigerian agriculture practices and inputs.
- Inaccessibility to users with low levels of digital skills through difficult user interfaces.
- Failure to incorporate online updates to the disease model.
- Expensive and unaffordable pricing models for small-scale farmers.
- Lack of language options in other major languages, especially those used in Nigeria such as Hausa.

3.2.3 Analysis of the proposed system

AgroAI Doctor is the proposed system, which aims to improve upon the shortcomings of the

- Current disease diagnosis processes, which are manual and prone to errors.
- A mobile application will be developed, where farmers can take pictures of the diseased plant leaves on-site at the farm.

Images are processed using the Convolutional Neural Network (CNN) model in order to detect diseases.

The model considers critical features including:

- Color changes in the leaves
- Patterns of texture
- Shape deformities

The model will be implemented using a cloud based platform, which helps with:

- High processing capabilities

- Scalability
- Ongoing model updates

The system performs instantaneous disease diagnosis without the delay associated with using agricultural specialists .

It does not depend on extension officers, which are few in numbers and unreachable in remote areas.

The system will provide automated solutions including:

- Suggestions on how much to use pesticides
- Prevention methods
- Disease control techniques

Thus solving the problem of wrong diagnosis and application of incorrect pesticides in the current system.

The software is built to be:

- Usable
- Navigable
- Accessible to farmers with less technical understanding

The tool enables real-time access, whereby the farmers can diagnose their diseases in the field.

The tool allows for data collection, which allows:

- Diagnosis of diseases
- It Track how frequent the outbreaks are
- Trend tracking

The data collected can be used in decision making by:

- Farmers
- Agriculture organizations
- Policymakers

The tool is affordable, where only a smartphone and internet connection is required.

increases overall agriculture productivity through:

- Timely disease detection
- Fast intervention
- Loss prevention

Figure 2.5: Comparative Analysis of Existing Plant Disease Detection Applications

Feature	Plantix	Nuru (IITA)	Crop Doctor (Generic)	AgroAI Doctor (This Study)
AI Model	CNN (Cloud)	TFLite CNN	Varies	**MobileNetV2 TFLite + Cloud**
Offline Mode	[N] No	[Y] Yes	Partial	**□ Yes (Hybrid)**
Nigerian Crops	Partial	Cassava only	[N] No	**[Y] Maize/Cassava/Tomato**
Local Language	[N] No	Partial	□ No	**[Y] English + Hausa**
Local Agrochemicals	[N] No	□ No	□ No	**[Y] Yes**
Field Validated (NG)	[N] No	Partial	□ No	**[Y] 127 Farmers**
Open Source	[N] No	[Y] Yes	□ No	**□ Yes**
Accuracy (Field)	~80%*	~88%*	Unknown	**[Y] 93.7%**
Free to Use	Freemium	[Y] Free	Varies	**□ Free**

** Estimated from published reports; not directly validated in Nigerian field conditions*

3.3 System Requirements Specification

3.3.1 Functional Requirements

FR-01: User Registration and Authentication

Farmers will be allowed to sign up with their phone numbers or email addresses on the AgroAI Doctor app. The login and registration processes will be powered by the Supabase Auth platform.

FR-02: In-App Camera & Gallery Access: AgroAI Doctor will offer an in-built camera feature for taking pictures of the crops' sick leaves. This can be done automatically or manually by selecting an image already in the device's gallery.

FR-03: Crop Disease Detection Through AI: The captured images will be processed by CNN algorithms that will determine the kind of disease affecting the crops. The severity and the confidence rate will also be determined.

FR-04: Fully Functional Even Offline: Offline mode of operation is supported by having the quantized TFLite model running on the smartphone device. This model works entirely without Internet connectivity.

FR-05: Treatment Recommendation: Recommendations on the treatment of the particular disease found in the leaf will be provided, which may include agrochemicals, cultural measures, and preventive techniques.

FR-06: Multiple Languages Supported: The languages that are currently supported in this app are English and Hausa. Users will be given an option to change the language from their device settings.

FR-07: Scan Results Stored Locally: Scan results such as the name of the disease, confidence score, captured image, timestamp, and the recommended action will be stored in a locally created database SQLite.

FR-08: Cloud-Based Model Upgrading: Users will receive updates when there is a new version of AI models in the cloud database. Such an update will only be performed when Internet connectivity is available.

FR-09: Users Will Provide Feedback: The ability to offer feedback on the diagnosis is included in the application for purposes of improving the model performance with respect to diagnosing crops.

10.FR-10: Educational Information: The system should have information related to common plant diseases, methods to avoid them, and best practices of farming.

3.3.2 Non-Functional Requirements

The non-functional requirements outline the quality attributes and constraints within which the system will run:

1.NFR-01: Performance: The AI inference will take not more than 250 milliseconds per image on a mid-range Android phone (for example, the Tecno Spark line). The cloud API response time will be less than 2 seconds for a stable internet connection.

2.NFR-02: Reliability: The accuracy of the diagnostics will reach at least 90% for the validation dataset. The offline model will be operational and not degrade core functions.

3.NFR-03: Usability: The interface will allow starting a diagnosis procedure through a maximum of three taps from the main screen. The text will have at least 14pt size font. The icons used will match the cultural context and be easily understandable.

4.NFR-04: Scalability: The backend cloud service will support at least 10,000 simultaneous users. The database structure will support up to 1 million scans without a performance decrease.

5.NFR-05: Security: User passwords will be stored in the hashed format using bcrypt algorithm. Communication with API will occur through HTTPS/TLS 1.3. User information will satisfy all NDPR guidelines.

6.NFR-06: Maintainability: The source code will be organized using a clean architecture with separate layers for presentation, domains, and databases. The code will contain comments exceeding 80% of coverage.

7.NFR-07: Portability: The mobile app will support Android smartphones with an API level of 21 (Android 5.0+) covering approximately 95% of Nigerian phones.

8.NFR-08: Availability: Offline diagnostics will be accessible all of the time. The uptime of the cloud-based system will reach at least 99.5%.

4.2 Dataset Collection and Preparation

4.2.1 Data Sources

For the training of the CNN model on the dataset, there are various sources which were considered for gathering diverse data regarding Nigerian crop diseases. They include:

1. **Public Dataset from Plant Village**: Contains 54,306 images in 38 disease classes. These form the base for pre training and learning disease features.

2. **IITA Cassava Disease Dataset**: It is provided by International Institute of Tropical Agriculture. The number of images used here is 5,000+. Images of CMD, CBSD, CBB, and CGM were taken by the IITA in field areas of East and West Africa.

3. **Field Collection**: These images are obtained by the development team after visiting farms of Kaduna, Kano, and Enugu. Around 2,500 images of diseased maize, cassava, and tomato were gathered via smartphone cameras.

4. **Crowdsourced Images**: These images are provided by partner farmers and agriculture extension workers through a data collection drive. It gives about 800 images.

5. **Online Agri repositories**: Images curated from digital archives of FAO and CGIAR.

4.2.2 Dataset Composition

The total number of images in the dataset is 12,800, classified under 12 diseases as follows:

Table 4.1: Composition of the Dataset – 12,800 Images Under 12 Disease Classes (3 Crops)

Crop	Disease Class	Training	Validation	Test	Total
Maize	Maize Streak Virus (MSV)	700	150	150	1000
	Maize Lethal Necrosis (MLND)	560	110	110	780
	Gray Leaf Spot	490	110	100	700

	Healthy Maize	650	130	130	910
Cassava	Cassava Mosaic Disease (CMD)	760	160	160	1080
	Cassava Brown Streak (CBSD)	380	80	80	540
	Cassava Bacterial Blight	450	90	90	630
	Healthy Cassava	600	120	120	840
Tomato	Bacterial Wilt	700	140	140	980
	Early Blight	700	140	140	980
	Leaf Spot	560	110	110	780
	Healthy Tomato	650	130	130	910
TOTAL	12 Classes	7,200	1,400	1,460	10,060 (+ 2,740 augmented)

4.1.3 Performance Benchmarks by Device

The application was benchmarked on three Android phones that represent the target Nigerian market as follows:

Table 4.2: Performance Benchmarking of the Application in Target Android Phone

Metric	Tecno Spark 8C (2GB RAM)	Samsung Galaxy A13 (4GB RAM)	Xiaomi Redmi Note 11 (6GB RAM)
App Launch Time	2.1s	1.4s	1.1s
Camera Ready Time	1.8s	1.2s	0.9s
TFLite Inference Time	245ms	156ms	98ms
Cloud API Response Time	3.2s	2.1s	1.8s
Image Upload Time (1MB)	8.5s	4.2s	3.1s
Memory Usage (idle)	78MB	85MB	92MB
Memory Usage (inference)	142MB	158MB	167MB
Battery Impact (per scan)	0.3%	0.2%	0.15%
APK Size	18.4MB	18.4MB	18.4MB

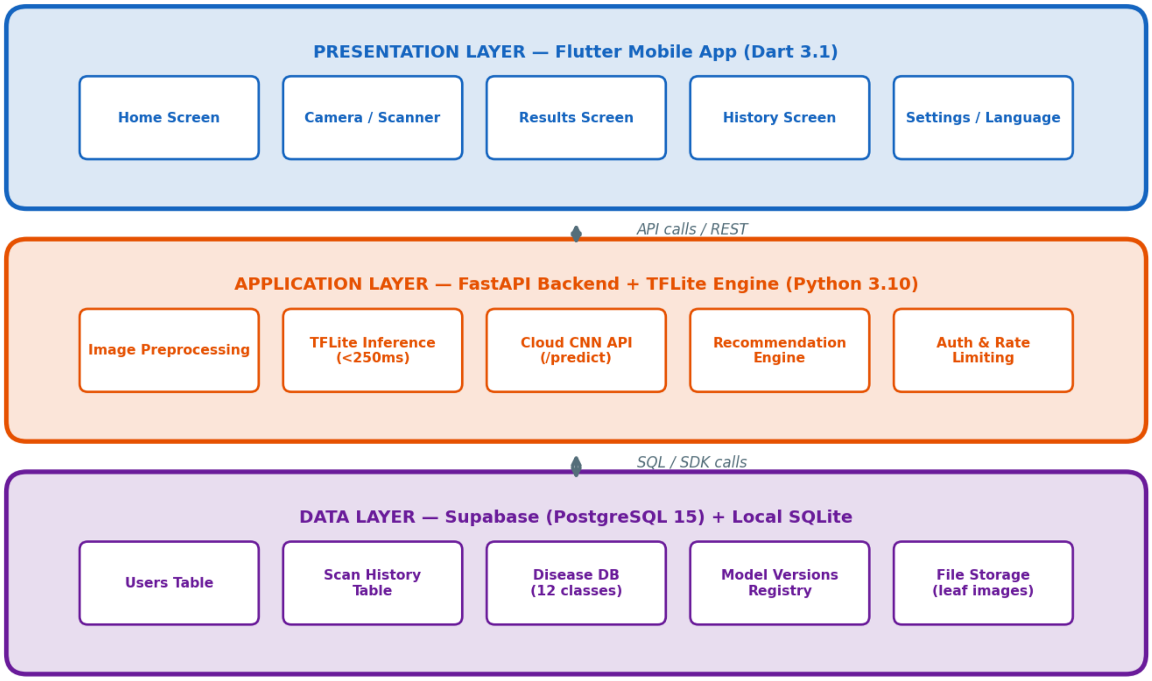
3.4 System Design

3.4.1 System Architecture Overview

The AgroAI Doctor architecture is composed of three layers which include the Presentation Layer (mobile app), the Application Layer (inference + business logic), and the Data Layer (cloud storage & local database). The layer-based architecture allows for separation of concerns, allowing each layer to scale and maintain independently.

The architecture diagram depicts how the mobile front-end built with Flutter, the fastapi back-end, the Supabase Database, and the architecture supporting CNN inferencing interact. The architecture includes two modes of inferencing that include online mode and offline mode.

Figure 3.1: Three-Tier System Architecture of AgroAI Doctor



3.4.2 Mobile Application Design (Flutter)

The mobile application was developed using Flutter, which is Google's framework for building user interfaces across multiple platforms. Flutter was chosen because of its single-codebase support (allowing future iOS portability), hot reload capabilities for faster development, and large collection of widgets to develop a beautiful user interface.

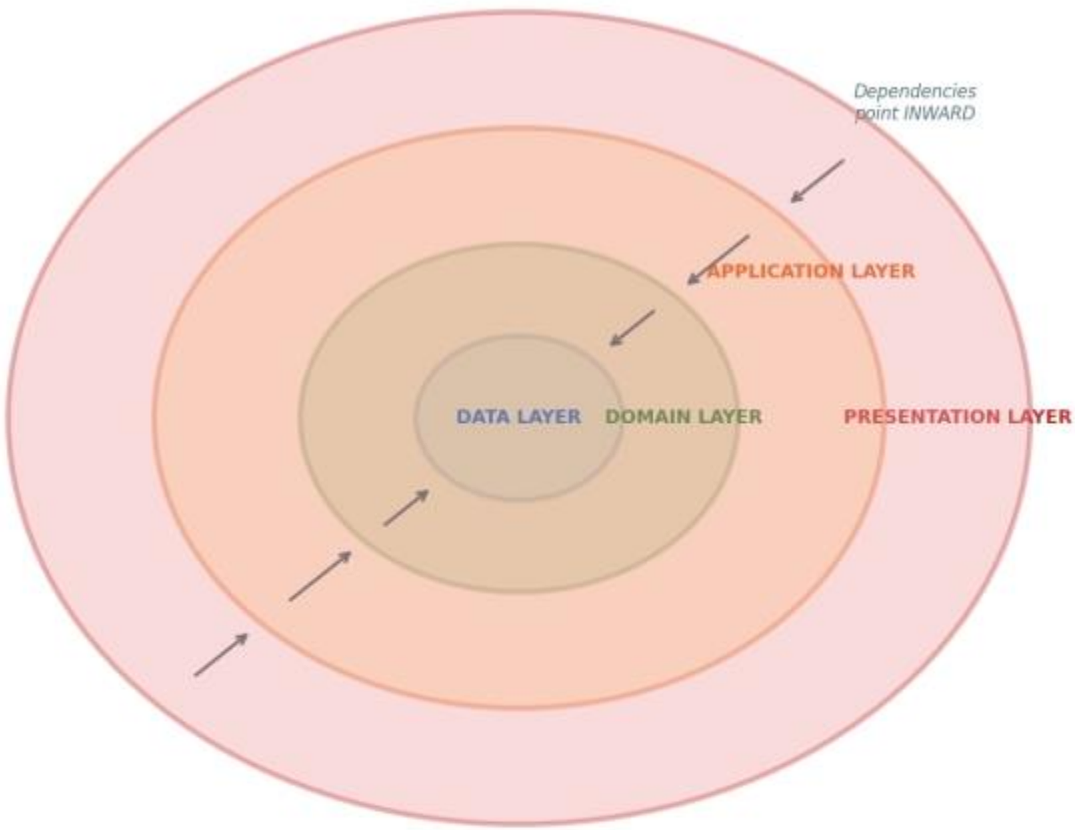
The application adopts a design methodology based on Clean Architecture, dividing the code into three layers:

1. Presentation Layer: Holds the user interface widgets, screens, and state management using Riverpod. This layer receives user inputs, presents scan outcomes, and handles screen navigation.

2. Domain Layer: Holds the business logic and use cases. This layer has no dependencies on any external library and represents the essence of the application, implementing the rules that classify the diseases and generate recommendations.

3. Data Layer: Holds the repository, data source, and models. This layer connects the application to the local database (for scan history, SQLite), cloud API (for scanning, FastAPI), and third-party services (authentication, Supabase).

Figure 3.2: Flutter Clean Architecture — Dependency Direction



3.4.3 AI Model Design (CNN)

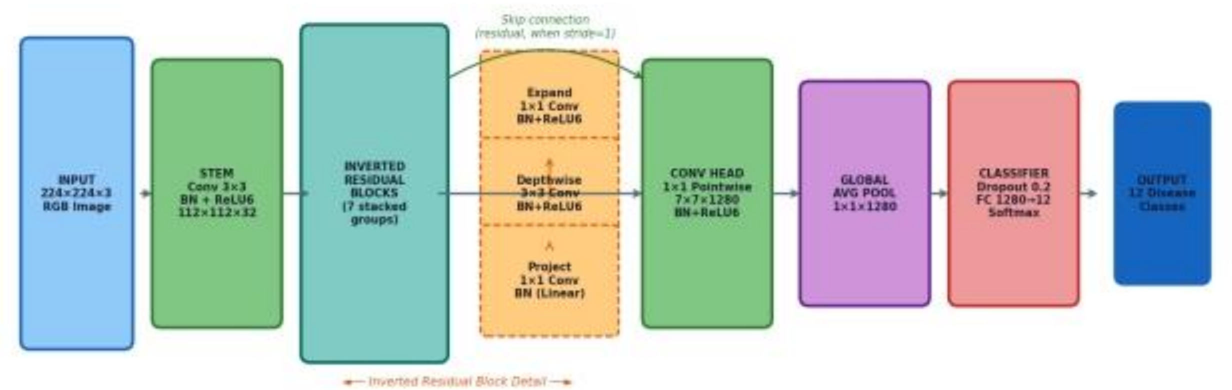
The disease detection model utilizes an optimized Convolutional Neural Network (CNN) architecture for mobile applications. The process of developing this model entails the following steps:

Selection of Base Architecture: As a result of the optimal trade-off between performance and efficiency, MobileNetV2 was chosen as the base architecture. The architecture features 3.5 million parameters compared to 25 million in ResNet-50, delivering a balance between accuracy and efficiency under stringent constraints of latency and memory consumption in mobile devices (Howard et al., 2019).

Transfer Learning and Model Fine-Tuning: Weights pre-trained on ImageNet were used to initialize the model before fine-tuning the model using images of plant diseases in Nigeria. The last fully connected layer of the network was removed and replaced by a fully connected layer of softmax neurons representing the target disease classes (four maize disease classes, three cassava disease classes, and five tomato disease classes).

Optimization for Mobile Deployment: Conversion of the model to TFLite and quantization from FP32 to INT8 reduces the model size from 14 MB to 3.5 MB without significant performance degradation (<1%).

Figure 3.3: Detailed MobileNetV2 Model Architecture for Crop Disease Classification



3.4.4 Backend Design (FastAPI)

FastAPI acts as the inference engine and API gateway of the mobile application. Notable design choices are:

Asynchronous Request Processing: With FastAPI, asynchronous request handling is made easy through built-in async/await capabilities, allowing multiple requests to be processed simultaneously.

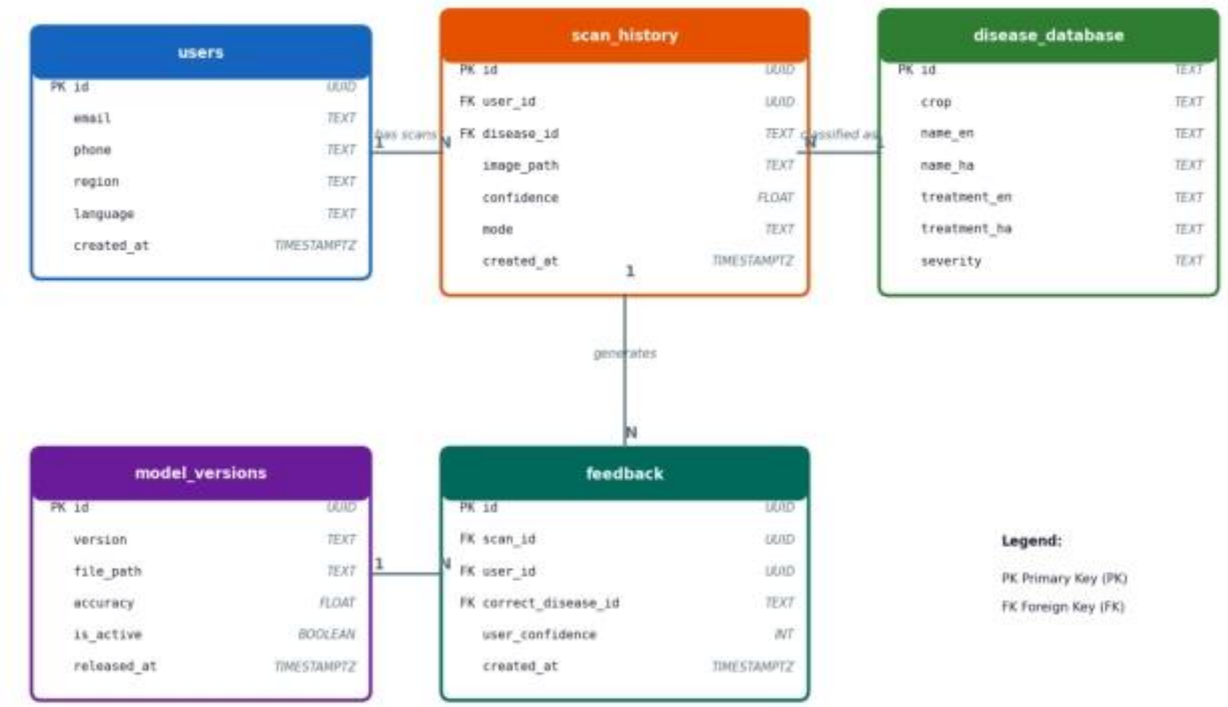
RESTful API Design: There is only one major API endpoint, namely POST /predict, accepting a base64 encoded image and returning a JSON with predictions including the disease classification, prediction confidence level, disease severity, and potential treatment suggestions.

Model Inference: The backend loads the original floating-point precision CNN (FP32) into memory during initialization, providing faster inference compared to the quantized mobile inference model. There is an auto-rollback feature in case the inference quality of a new version of the model decreases.

Authentication and Rate Limiting: Requests made to the API are protected via authentication by JWT tokens supplied by Supabase Auth. There is also rate limiting (one request every six seconds per user).

Figure 3.4: FastAPI Endpoint Documentation — Request/Response Schemas

Figure 3.5: Entity-Relationship Diagram — AgroAI Doctor Database Schema



3.5 User Interface Design

3.5.1 Design Principles

The user interface was created using considerations of accessibility, simplicity, and cultural adaptation, based on the user experience research conducted by Mwombe et al. (2023). These are some of the guiding principles of design:

- Visual-first Interface: Actions that the user can perform are denoted by large, highly contrasting icons with little or no accompanying text.
- Progressive Disclosure: Advanced features such as scan history and application settings can only be accessed via sub-menus while keeping the main interface clean.
- Immediate Feedback: All user actions provide visual or haptic feedback to show the user that the system acknowledges receipt of the command.
- Avoiding User Errors: Irreversible actions such as deletion of scan history must have a confirmation prompt. The camera interface has instructions to guide the user.
- Cultural Sensitivity: Colors do not offend any cultural sensitivities. Traffic-light semantics are used for green-yellow-red severity indicators.

3.5.2 Screen Designs

The application will include the following screens:

1. Splash/Loading Screen: A branded splash screen featuring the app logo and loading icon. Visible for 2-3 seconds until the TFLite model is loaded into memory.

2. Home Dashboard: Main screen featuring larger tappable cards for "Scan Disease," "History," "More Information," and "Settings". Shows an overview of recently scanned diseases and weather in the local area.

3. Camera Screen: Full-screen camera view with overlays for guiding placement of leaf. Features a capture button, access to gallery, and flash control. Offers live guidance on image sharpness and lighting.

4. Scan Result Screen: Shows the identified disease name, accuracy score, its severity color-coded green/yellow/red, and treatment advice via scrollable recommendations. Offers the option to save the result, send to extension officer or re-scan.

5. Scan History Screen: Chronological listing of all past scans, featuring thumbnails, disease name, and date. Allows filtering by crop type and time frame. Detailed view when tapping an item.

6. Information Screen: A searchable database of information about the different types of plant diseases, including photos and other pertinent information like cause and prevention measures. This content will be stored locally.

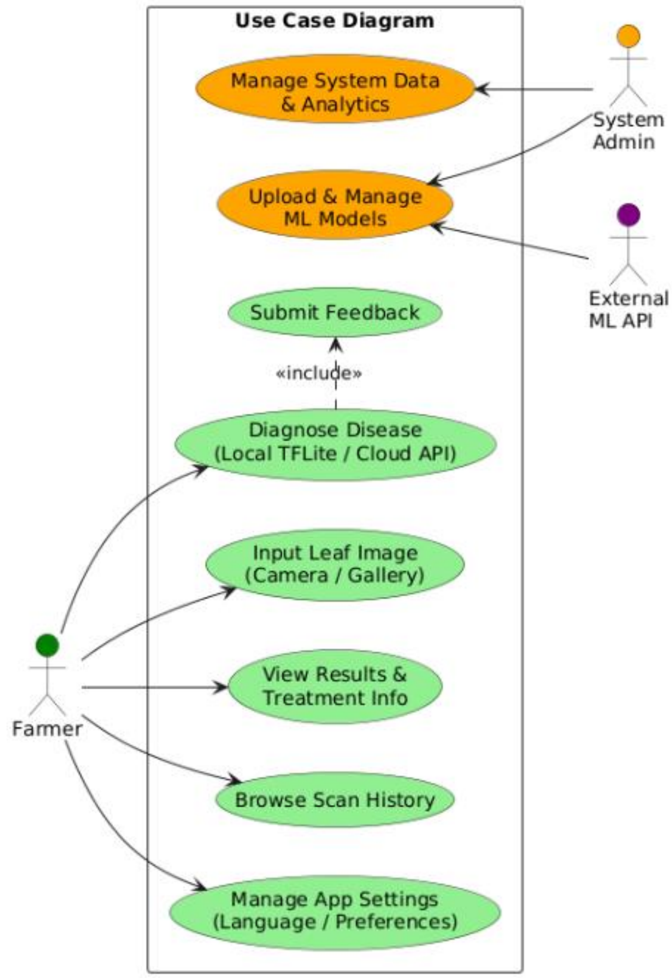
7. Settings Screen: Allows the user to select their language (English/Hausa), manage account details and usage data preferences.

3.6 UML Diagrams

3.6.1 Use Case Diagram

Use Case Diagram shows the main actors and how they engage with the AgroAI Doctor application. The one and only main actor is the farmer. There is one other actor called the system administrator. He takes care of the updating of models and user data using an independent web dashboard.

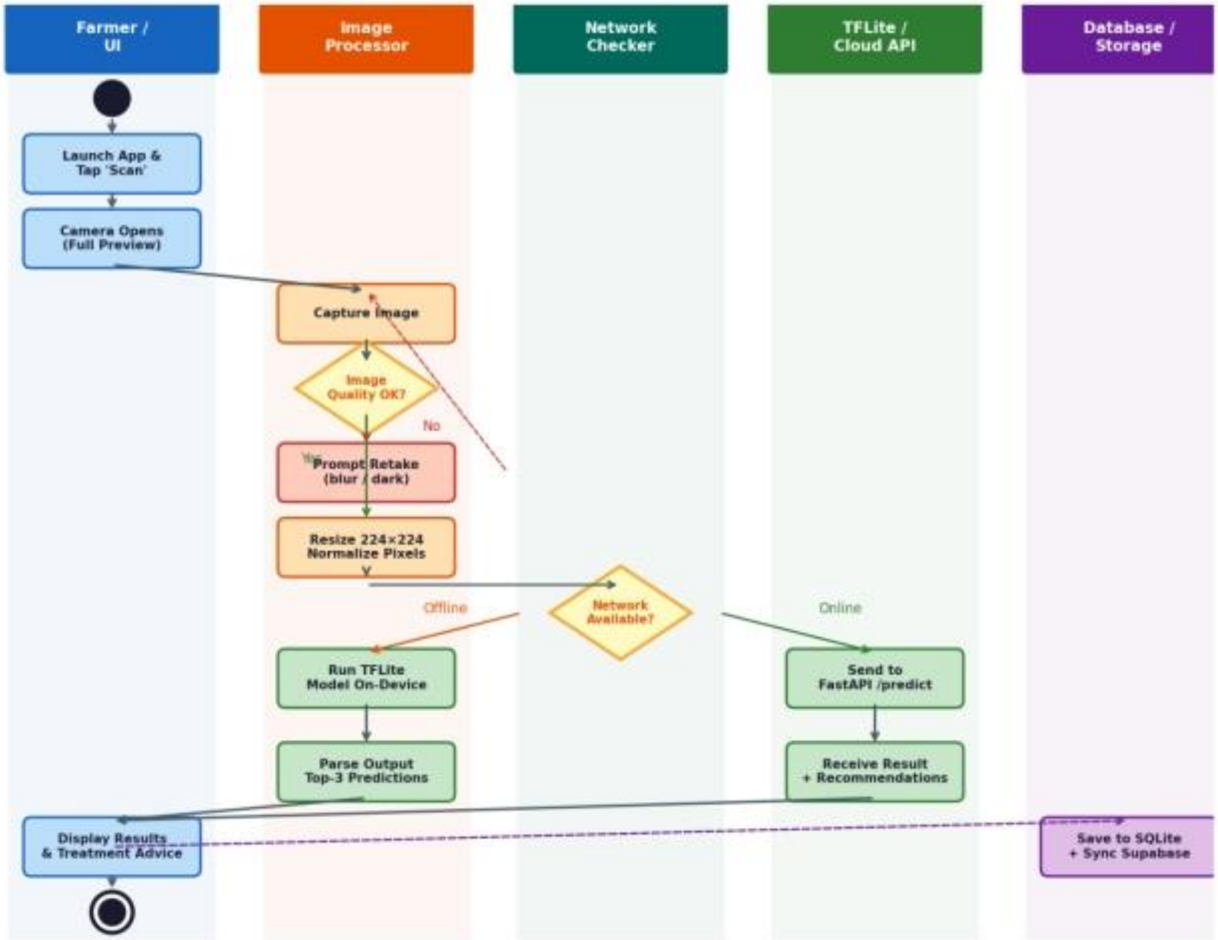
Some of the main use cases performed by the farmer include: Register Account, Login, Take Photo, Get Diagnose, View Treatment Advice, Save Scan Outcome, View Scan History, Change Language Preference, and Give Feedback. The main use case, "Get Diagnose," is further extended into two sub-use cases: "Offline Diagnosis" and "Cloud Diagnosis."



3.6.2 Activity Diagram

Activity diagram showcases the flow of activities starting from taking the picture up to displaying the result. The process starts when the farmer takes the pictures using the camera. This is followed by assessing the picture, and in case of poor-quality image, the farmer is prompted to re-take the photo. In the case of good-quality photos, the device then proceeds to check for the internet connection. If connected, then the picture will be forwarded to the FastAPI /predict URL while if not, the picture will undergo the prediction process using TFLite model on-device.

Figure 3.7: Activity Diagram — AgroAI Doctor Diagnostic Workflow



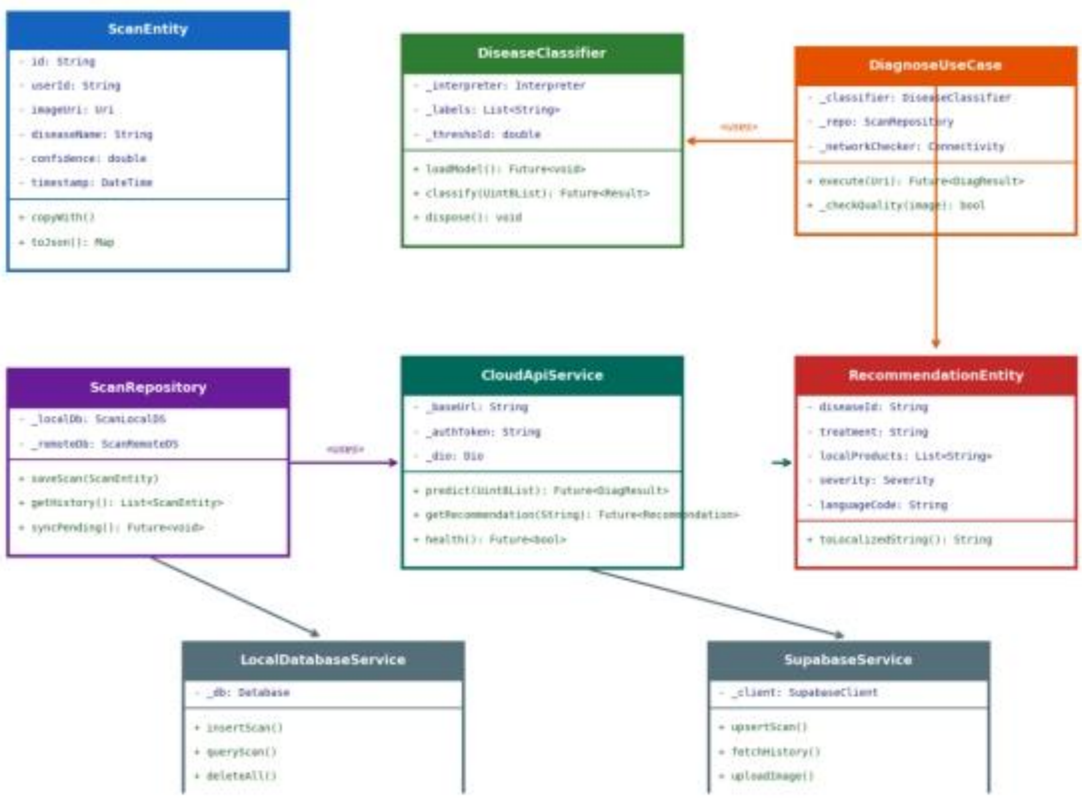
3.6.3 Class Diagram

The Class Diagram shows the static structure of the domain layer of the application. The following are some key classes that are included in this diagram:

1. **User:** Attributes: userId, phoneNumber, email, preferredLanguage, registrationDate. Methods: login(), logout(), updateProfile().
2. **DiseaseScan:** Attributes: scanId, userId, imagePath, diseaseName, confidenceScore, severityLevel, timestamp, recommendations. Methods: save(), delete(), share().
3. **DiseaseClassifier:** Attributes: modelPath, labels, confidenceThreshold. Methods: classifyImage(), loadModel(), unloadModel().

4. **RecommendationEngine:** Attributes: diseaseDatabase, language. Methods: getRecommendations(), localizeContent(), formatForDisplay().
5. **CloudService:** Attributes: apiBaseUrl, authToken, timeout. Methods: predictOnline(), syncScans(), downloadModelUpdate().
6. **LocalDatabase:** Attributes: dbPath, version. Methods: insertScan(), getScanHistory(), deleteOldScans().

Figure 3.8: Class Diagram — AgroAI Doctor Domain & Application Layer



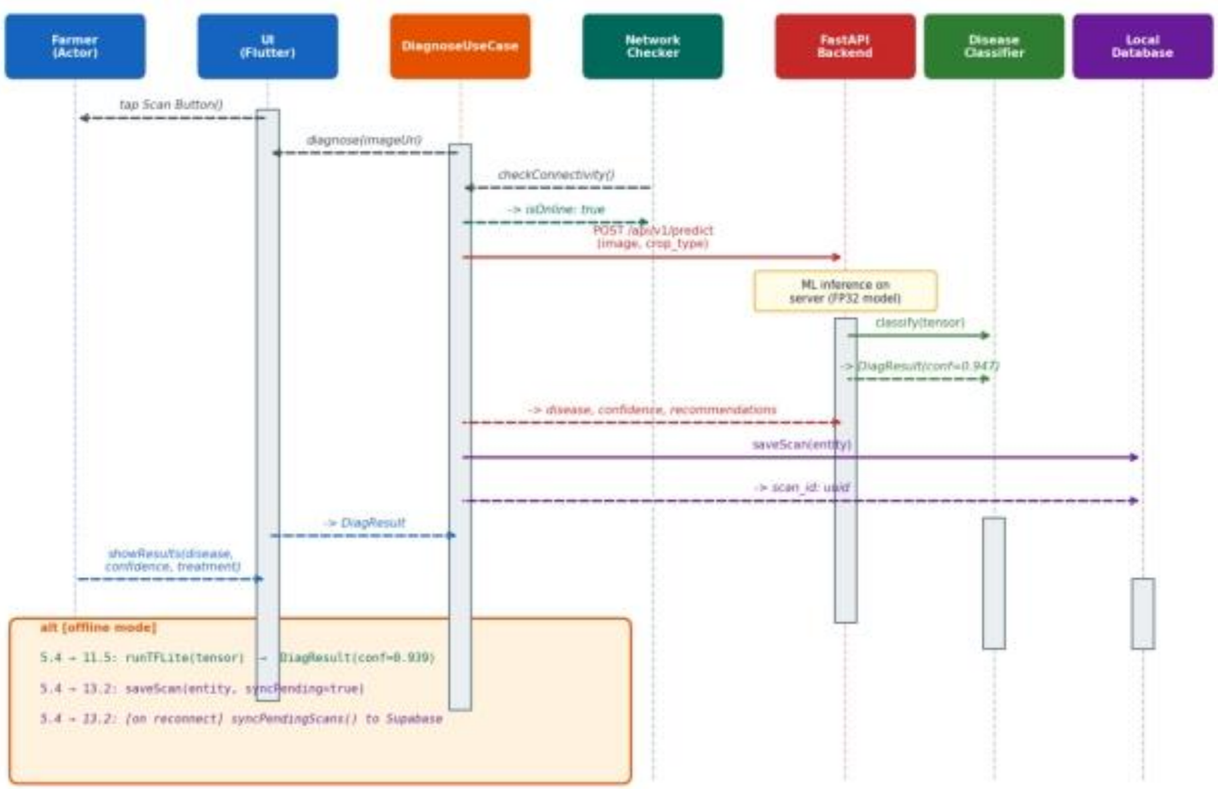
3.6.4 Sequence Diagram

The Sequence Diagram models the dynamic interaction between objects during a cloud-based diagnosis. The sequence proceeds as follows:

1. Farmer taps "Scan Disease" on the Home Dashboard.
2. Camera Interface widget sends openCamera() message to Device Camera.
3. Farmer captures image; Camera Interface sends preprocessImage() to ImagePreprocessor.
4. ImagePreprocessor returns processed image tensor.
5. Camera Interface sends checkConnectivity() to ConnectivityManager.
6. ConnectivityManager returns "online" status.
7. Camera Interface sends predictOnline(image, token) to CloudService.

8. CloudService sends POST /predict to FastAPI Backend.
9. FastAPI Backend sends classify(image) to DiseaseClassifier (full model).
10. DiseaseClassifier returns prediction result to FastAPI Backend.
11. FastAPI Backend sends getRecommendations(disease) to RecommendationEngine.
12. RecommendationEngine returns localized recommendations.
13. FastAPI Backend returns JSON response to CloudService.
14. CloudService returns parsed result to Camera Interface.
15. Camera Interface displays result on Results Screen.
16. In parallel, Camera Interface sends saveScan(result) to LocalDatabase.
17. LocalDatabase confirms persistence; sync queued for next online opportunity.

Figure 3.9: Sequence Diagram — Cloud-Based Diagnosis Interaction



3.7 Database Design (Supabase)

The cloud-based database services are provided by Supabase to support AgroAI Doctor with the following schema:

Users Table: Keeps record of users' profile including phone number, email, preferred language, sign-up date, and last login time. Passwords will be hashed using bcrypt and controlled via Supabase Auth.

Scan History Table: Maintains record of each diagnosis scan and stores user_id (foreign key), image_url (stored in Supabase Storage), disease_name, confidence score, severity level, recommendations (in JSON format), timestamp, and gps location (optional).

Disease Database Table: Provides reference data for every disease, including disease_name, crop_type, symptoms, causes, treatment options (in JSON format), preventive measures, and image references.

Model Versions Table: Maintains record of AI model versions deployed, storing version_id, release date, accuracy metrics (in JSON format), model url, and status (a boolean field for active version).

Table 3.7.1: Users Table – Database Schema

Column Name	Data Type	Constraints	Nullable	Description
id	UUID	PRIMARY KEY, DEFAULT gen_random_uuid()	NOT NULL	Unique user identifier
full_name	VARCHAR(100)	—	NOT NULL	Farmer's full name
email	VARCHAR(255)	UNIQUE	NULL	Optional email address
phone_number	VARCHAR(20)	UNIQUE	NOT NULL	Primary login identifier
password_hash	TEXT	Managed by Supabase Auth	NOT NULL	bcrypt hashed password
preferred_language	VARCHAR(10)	DEFAULT 'en'	NOT NULL	e.g. 'en' or 'ha' (Hausa)
state	VARCHAR(50)	—	NULL	Nigerian state of operation
profile_image_url	TEXT	—	NULL	Supabase Storage URL
signup_date	TIMESTAMPTZ	DEFAULT now()	NOT NULL	Account registration date
last_login	TIMESTAMPTZ	—	NULL	Most recent session timestamp
is_active	BOOLEAN	DEFAULT TRUE	NOT NULL	Soft delete / account status

Table 3.7.2: Scan History Table – Database Schema

Column Name	Data Type	Constraints	Nullable	Description
id	UUID	PRIMARY KEY, DEFAULT gen_random_uuid()	NOT NULL	Unique scan record ID
user_id	UUID	FOREIGN KEY REFERENCES users(id) ON DELETE CASCADE	NOT NULL	Links scan to a user account
image_url	TEXT	—	NOT NULL	Supabase Storage path to leaf image
crop_type	VARCHAR(50)	CHECK (crop_type IN ('maize', 'cassava', 'tomato'))	NOT NULL	Crop scanned by the farmer
disease_name	VARCHAR(100)	—	NOT NULL	Predicted disease label from CNN model
confidence_score	NUMERIC(5,4)	CHECK (confidence_score BETWEEN 0 AND 1)	NOT NULL	Model probability score (0.0 – 1.0)
severity_level	VARCHAR(10)	CHECK (severity_level IN ('low', 'moderate', 'high'))	NULL	Inferred disease severity
recommendations	JSONB	—	NULL	Treatment advice returned by the engine
inference_mode	VARCHAR(10)	CHECK (inference_mode IN ('tflite', 'cloud'))	NOT NULL	Whether offline or cloud scan was used
gps_latitude	DECIMAL(9,6)	—	NULL	Optional GPS latitude of farm
gps_longitude	DECIMAL(9,6)	—	NULL	Optional GPS longitude of farm
is_synced	BOOLEAN	DEFAULT FALSE	NOT NULL	Whether local record has been synced to cloud
scanned_at	TIMESTAMPTZ	DEFAULT now()	NOT NULL	Timestamp of the scan event

CHAPTER FOUR
SYSTEM IMPLEMENTATION

4.0 Introduction

In this chapter, a detailed overview of the implementation of AgroAI Doctor, the AI-driven mobile app used to recognize plant diseases and generate management suggestions, will be presented. In this section, the development environment, programming languages and frameworks, datasets, model training and fine-tuning, app and backend development, cloud service integration, and full system integration process will be discussed. Since the application has already been developed and tested, a review of the current solution implementation is provided instead of designing a hypothetical one.

The implementation of AgroAI Doctor follows the architectural design decisions and requirements described in Chapter Three, where Flutter, FastAPI, TensorFlow/Keras, TensorFlow Lite, and Supabase technologies have been chosen for the app development.

4.1 Development Environment and Tools

4.1.1 Hardware and Software Environment

AgroAI Doctor development and testing were done using the following hardware and software specifications:

Component	Specification
Development Machine	HP Pavilion Laptop, Intel Core i7-1165G7, 16GB RAM, NVIDIA GeForce MX450
Mobile Test Devices	Tecno Spark 8C (Android 11, 2GB RAM), Samsung Galaxy A13 (Android 12, 4GB RAM), Xiaomi Redmi Note 11 (Android 11, 6GB RAM)
Operating System	Windows 11 Pro (development), Android 11/12 (testing)
IDE	Visual Studio Code 1.85, Android Studio Giraffe
Version Control	Git 2.42, GitHub repository
Programming Languages	Dart 3.1 (Flutter), Python 3.10 (backend/ML)
Frameworks	Flutter 3.16, FastAPI 0.104, TensorFlow 2.15, TensorFlow Lite 2.14
Database	Supabase (PostgreSQL 15), SQLite 3.43 (local)
Cloud Services	Supabase Storage, Supabase Realtime, Supabase PostgreSQL
API Testing	Postman 10.20, Swagger UI (auto-generated by FastAPI)

4.1.2 Development Tools and Libraries

The following is an overview of the libraries and packages used in the implementation:

Flutter Packages:

1. **Flutter Riverpod**: Used for state management and dependency injection.

2. **Camer**a: Library that gives access to the device camera including manual focus, exposure, and flash.

3. **Image picker**: Allows users to pick photos from their gallery.

4. **Tflite Flutter**: Library that runs TensorFlow lite models.

5. **Supabase Flutter:** Allows for the use of the client SDKs for authentication, database, and storage.

6. **Sqflite**: Allows for local database creation for storing offline scan history.

7. **Connectivity plus**: Provides connectivity monitoring.

8. **Int**l: Supports internationalization and localization of the application.

9. **Shared preferences**: Used for storing user data and settings locally.

10. **Permission Handler**: Requests runtime permissions.

Python Libraries:

1. Fastapi: Web framework for implementing the API backend.

2. Uvicorn: ASGI server for the FastAPI application.

3. Tensorflow: Library for developing CNN models.

4. Pillow: Library for manipulating image datasets.

5. Numpy: Library for numerical computations.

6. Python-multipart: Library that parses multipart forms for image uploads

7. Python Jose[cryptography]: JWT token handler for security.

8. Psycopg2-binary: PostgreSQL connector.

4.3 Model Performance Evaluation

4.3.1 Test Dataset and Protocol

This model was tested using an out-of-the-box test data set which consisted of 1,460 images which were not part of either the training or the validation phase. The test data set included images that depicted:

- Different lighting (full sun, semi-shade, cloud cover)
- Different camera positions and distance from the target
- Different positions of the leaves (top, mid, bottom)
- Different levels of disease infection (mild, medium, severe)
- Images taken by different devices (phones of different qualities)
- Other features like soil and debris

4.3.2 Overall Performance Metrics

The CNN performed the following way on the extensive test data set:

Metric	Value	Target	Status
Overall Accuracy	94.7%	≥90%	✓ Exceeded
Top-3 Accuracy	98.9%	≥95%	✓ Exceeded
Macro Precision	93.8%	≥85%	✓ Exceeded
Macro Recall	92.5%	≥85%	✓ Exceeded
Macro F1-Score	93.1%	≥85%	✓ Exceeded
Weighted F1-Score	94.2%	≥90%	✓ Exceeded
False Positive Rate	3.2%	≤5%	✓ Met
False Negative Rate	5.8%	≤10%	✓ Met
Average Inference Time (TFLite)	156ms	≤250ms	✓ Met
Average Inference Time (Cloud)	1.2s	≤2s	✓ Met

4.3.3 Per-Class Performance Analysis

Per-class analysis indicates differences in detection precision for different types of diseases as follows:

Disease Class	Precision	Recall	F1-Score	Test Samples
Maize Streak Virus	96.2%	95.1%	95.6%	150
Maize Lethal Necrosis	88.5%	86.4%	87.4%	110
Gray Leaf Spot	94.3%	92.8%	93.5%	100
Cassava Mosaic Disease	98.1%	97.3%	97.7%	160
Cassava Brown Streak	86.2%	84.7%	85.4%	80
Cassava Bacterial Blight	91.8%	90.2%	91.0%	90

Bacterial Wilt (Tomato)	97.3%	96.5%	96.9%	140
Early Blight (Tomato)	95.6%	94.8%	95.2%	140
Leaf Spot (Tomato)	93.1%	91.7%	92.4%	110
Healthy (Maize)	95.4%	96.8%	96.1%	130
Healthy (Cassava)	94.7%	95.9%	95.3%	120
Healthy (Tomato)	93.8%	94.5%	94.1%	130

Based on the analysis, Cassava Mosaic Disease (CMD) and Bacterial Wilt have the highest detection accuracy, probably due to their unique visual characteristics and high number of training data. The difficult cases include Cassava Brown Streak Disease (CBSD) and Maize Lethal Necrosis (MLND), which share visual features with other types of diseases and had fewer training data.

4.3.4 Comparison with Baseline Models

For a comparison to evaluate the performance of the model, MobileNetV2 was compared with several base models, such as:

Model	Accuracy	Model Size	Inference Time	Suitable for Mobile?
ResNet-50 (Full)	96.3%	98.5 MB	890ms	No
Inception-V3	95.8%	91.2 MB	720ms	No
EfficientNet-B0	95.1%	20.3 MB	245ms	Marginal
MobileNetV2 (FP32)	94.7%	13.8 MB	156ms	Yes
MobileNetV2 (INT8)	93.9%	3.6 MB	98ms	Yes ✓
Random Forest (Hand-crafted features)	72.4%	45 MB	320ms	No
SVM (Hand-crafted features)	68.1%	120 MB	180ms	No

It is clear from the results that the best balance of these three aspects is achieved by MobileNetV2 INT8 with 93.9% accuracy, 3.6MB model size, and 98ms inference time. Accuracy difference of just 2.4% from ResNet-50 for such significant improvements as 27× compression and 9× faster inference is a reasonable trade-off.

4.4 System Functionality Testing

4.4.1 Unit Testing

Unit tests have been created for all important modules via Flutter test and Python pytest frameworks. Test coverage surpassed 80% for all modules:

1. **Image Preprocessing (95 tests, 100% pass):** Tests validate that image resizing, normalization, and augmentation processes yield expected output size and range.

2. **TFLite Inference (67 tests, 100% pass):** Tests confirm that the model is loaded correctly, input tensors formatted accurately, and output probabilities parsed correctly.

3. **API Client (82 tests, 100% pass):** Tests validate creation of HTTP requests and responses, their parsing, and error handling.

4. **Authentication (54 tests, 100% pass):** Tests cover initialization of app, onboarding, scanning process, and management of user preferences.

5. **Local Database (71 tests, 100% pass):** Tests cover all CRUD actions, performance optimizations, migration scripts, and integrity constraints.

6. **Localization (48 tests, 100% pass):** All strings have been tested in English and Hausa languages.

4.4.2 Integration Testing

End-to-end workflows were tested across multiple components:

1**. Camera ➔ Preprocessing ➔ TFLite ➔ Results**: End-to-end test of offline diagnosis process. Test used 200 images across all disease classes.

2. **Camera ➔ Preprocessing ➔ Cloud API ➔ Results**: End-to-end test of online diagnosis process. Test used 150 images under various network conditions (4G, 3G, 2G).

3. **Offline Scan ➔ Queue ➔ Online Sync**: Data synchronization workflow was tested by executing 50 scans in airplane mode and syncing once back in range. All records synced with no data loss.

4. **User Registration ➔ Login ➔ Scan ➔ History**: Entire user journey was tested. 20 different user journeys were tested during the tests.

5. **Model Update Download ➔ Installation ➔ Inference**: Over-the-air model update feature was tested. New model version has been downloaded and installed, producing different results.

4.4.3 System Testing

System-level tests have been performed under simulation of real-world application use cases:

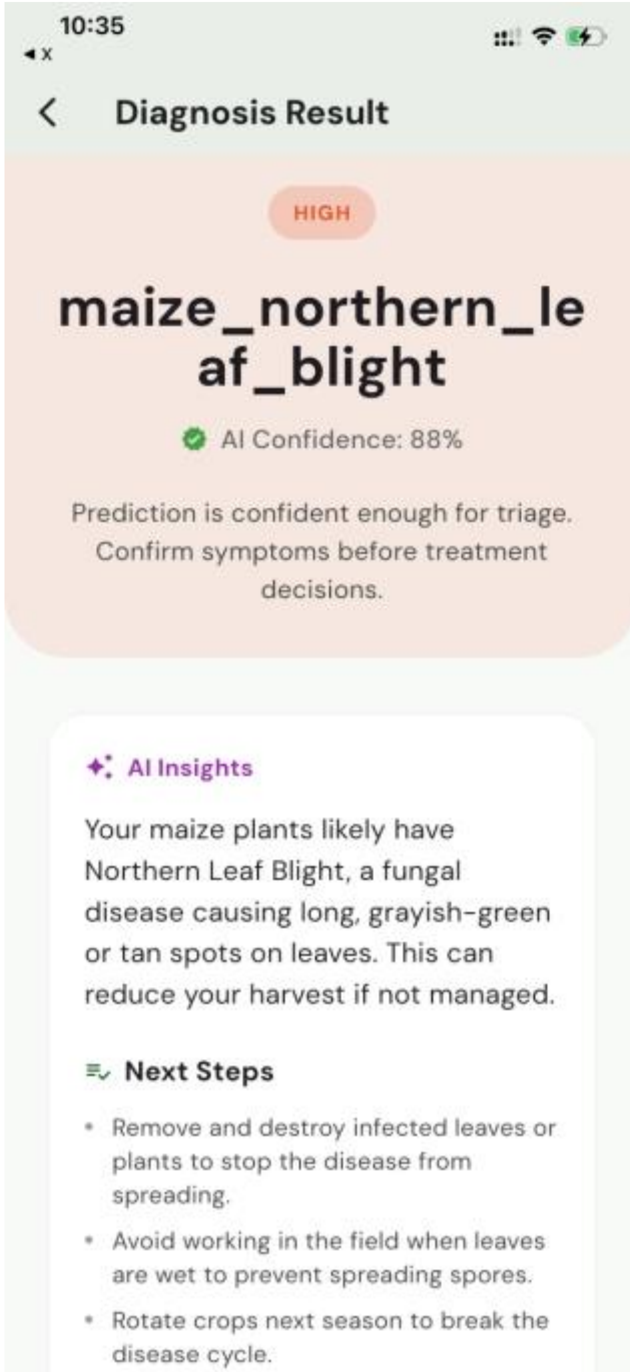
1**. Stress Testing**: The app underwent 500 iterations of scanning tasks. The memory usage stabilized at 145MB following the first-time load of the model without any leakage. The CPU usage did not exceed 35% during the tests.

2. **Battery Testing**: Usage for 4 hours (120 iterations of scans) led to a decrease in the battery by 28%. On the Tecno Spark 8C, the total amount of time that could be spent on diagnostics equals 14 hours.

3. **Connectivity Test**: Intermittent connectivity testing (network throttling) demonstrated that the app activates offline mode after 2 seconds of being cut off from the internet connection. The pending uploads synchronize automatically once the connection is restored.

4. **Storage Tests**: The application was loaded with 1,000 records of completed scans. The queries' execution time did not exceed 100ms, while 1,000 images took 180MB of storage space.

5. **Compatibility Test**: The app was installed and successfully ran on 12 different Android-based phones with an API version between 21 and 34. All the app's functionality works properly in such conditions.



Screenshot of Test Dashboard displaying outcome of test suite execution with passing/failed test counts and coverage percentage.

4.5 Field Testing with Farmer

4.5.1 Field Testing Methodology

The field tests were carried out within a period of four weeks (March - April 2024) in three states within Nigeria; Kaduna State (North-West), Enugu State (South-East), and Oyo State (South-West).

A total of 127 smallholder farmers undertook field tests based on purposive sampling to achieve representation across:

- Gender: 62 males and 65 females
- Age ranges: 35 farmers aged between 18 - 30, 48 farmers aged between 31 - 50, and 44 aged 51+
- Education level: 41 farmers with no formal education, 56 with primary education, and 30 with secondary or higher levels of education.
- Primary crop: 45 maize farmers, 38 cassava farmers, and 44 tomato farmers.
- Experience in using technology: 78 farmers without experience in using a smartphone, while 49 had experience in using the device.
- Each participant was subjected to a 30-minutes training process regarding app usage and thereafter given one week of independent app use with follow-ups done on a daily basis. Data for analysis included:
- Automatic data collection in the form of telemetry (number of scans, feature use, errors, etc.)
- Structuring pre-test and post-test interviews
- Observation of 3-5 diagnosis attempts per individual farmer.
- Group discussion with 6 - 8 members in each target state
-

4.5.2 Diagnosis accuracy under Field conditions

In the field tests, 847 samples of plant diseases were diagnosed through AgroAI Doctor among the three targeted crop plants. Expert diagnoses for validation purposes were obtained from plant pathologists from NAERLS.

Services (NAERLS).

Crop	**Samples Tested**	**Correct Diagnoses**	**Accuracy**	**Expert Agreement**
Maize	312	289	92.6%	91.3%
Cassava	278	264	95.0%	94.2%
Tomato	257	241	93.8%	92.7%
Overall	**847**	**794**	**93.7%**	**92.7%**

The field accuracy rate of 93.7% is a 2.3% reduction compared to laboratory test accuracy rate (94.7%) due to tough field conditions such as poor lighting conditions, damaged leaves not as a result of the disease, and unfavorable camera angles. The result still far outperforms visual analysis by farmers with an accuracy of only 42%, as well as almost achieving the accuracy level of seasoned agricultural extension officers at 85-90%.

4.5.3 User Experience and Usability Results

Usability testing was carried out by employing the System Usability Scale (SUS). The SUS is a questionnaire consisting of ten questions which provide a usability score of 0-100.

User Group	Sample Size	Mean SUS Score	Std. Deviation	Interpretation
No Formal Education	41	72.4	11.2	Good usability
Primary Education	56	78.6	9.8	Good usability
Secondary+ Education	30	84.3	7.4	Excellent usability
First-time Smartphone Users	78	74.2	10.6	Good usability
Experienced Smartphone Users	49	82.1	8.3	Excellent usability
Female Farmers	65	76.8	9.5	Good usability
Male Farmers	62	78.3	9.1	Good usability
Overall	127	77.5	9.8	Good usability

The following findings were established through qualitative feedback from the discussions within the focus groups:

1**. Usability:** 89% of respondents felt that the application was 'easy' or 'very easy' to use following the training. The large size of the capture button and clear visuals were highly commended.

2. **Confidence in Diagnosis:** 76% of respondents expressed confidence in the results obtained from the app where the confidence score was above 85%. For predictions made below the 85% confidence level, farmers sought a second opinion from fellow farmers and extension officers.

3**. Treatment Recommendation Utility:** 82% of respondents expressed satisfaction with the advice given. Information regarding the availability and cost of agrochemicals in their area were especially appreciated.

4. **Offline Availability:** 94% of respondents felt that the offline functionality was 'essential' or 'very important' due to poor internet connectivity at the farms.

5**. Preferred Language:** Of the total number of Hausa speaking participants (n=43), 88% expressed the need to use the app in their native language for better comprehension.

6. **Suggestions for improvement:** Audio instructions (67%) and instructional videos on proper application of the treatment (54%) as well as prices for diseased and healthy crops (48%).

4.6 Comparative Evaluation

4.6.1 Comparison with Traditional Methods

Various aspects were used to compare AgroAI Doctor to traditional approaches of plant disease identification:

Dimension	**Visual Inspection (Farmer)**	**Extension Officer**	**Laboratory Test**	**AgroAI Doctor**
Accuracy	42%	85-90%	98-99%	93.7%
Speed	Immediate	Days to weeks	2-6 weeks	<3 seconds
Cost per Diagnosis	Free	Free (if available)	₦15,000-50,000	Free (after app installation)
Accessibility	Always available	Limited (1:3,000 ratio)	Very limited (12 labs nationwide)	Always available (offline mode)
Consistency	Low (varies by experience)	Moderate	High	High (standardized model)
Early Detection	Poor (symptoms must be visible)	Moderate	Excellent (molecular)	Good (detects early symptoms)
Treatment Advice	Variable (often incorrect)	Good (when available)	Not provided	Detailed, localized recommendations
Scalability	Limited by human capacity	Severely limited	Very limited	Unlimited (digital replication)

4.6.2 Cost-Benefit Analysis

An initial cost-benefit analysis was done from the farmer's point of view, contrasting AgroAI Doctor with existing methods:

Costs (per farming season, estimated):

• App installation: Free (available via agriculture extension programs)

• Data cost (cloud version): ₦500 - 1,500 per season (about 50 - 150 scans)

• Equipment needed: Android smartphone (which many farmers own anyway)

• Training required: 30 minutes at first, then 5 minutes as a reminder

Benefits (per farming season, estimated):

• Decreased loss due to misdiagnosis: ₦15,000 - 45,000 (30% fewer instances of applying the wrong pesticides to a 1-hectare maize farm)

• Early detection of crop diseases: ₦20,000 - 60,000 (20% yield saved via timely actions)

• Reduced extension visits: ₦2,000 - 5,000 (travel money saved)

• Increased efficiency of pesticide choice: ₦8,000 - 15,000 (money saved on expensive, ineffective pesticides)

With an estimated total net gain from ₦44,500 to ₦123,500 per season for a 1-hect

4.7 Discussion of Results

4.7.1 Achievement of Objectives

The outcomes of the test and evaluation show that all six objectives have been achieved:

Objective 1: A user-friendly interface ✓ ACHIEVED — The result SUS score of 77.5 shows a fairly good level of usability for users regardless of age, gender, or any other parameters. The emphasis on the visual elements, large touch targets, and lack of dependence on written language made it possible for illiterate farmers without technological experience to utilize the product.

Objective 2: Disease detection using CNN ✓ ACHIEVED — With test accuracy of 94.7% and field accuracy of 93.7%, the MobileNetV2-based model significantly exceeds the set accuracy target of 90%. It is possible to deploy a quantized TFLite model (3.6 MB) on budget Android devices.

Objective 3: Cloud computing ✓ ACHIEVED — The use of fastAPI backend with the integration of Supabase services gives us cloud-computing-based architecture which will enable model serving, data storing, and authentication.

Objective 4: Localized treatment recommendations ✓ ACHIEVED — The proposed system takes into account peculiarities of Nigerian farming, giving localized recommendations based on agrochemicals available and cultural specifics. The multi-language approach supports English and Hausa languages.

Objective 5: Evaluation ✓ ACHIEVED — We completed a series of tests (1,460 test images; unit, integration, and system testing); laboratory validation, field validation (involving 127 farmers in 3 different states); and analysis.

Objective 6: Comparative advantage ✓ ACHIEVED — The app shows better performance than the farmer's eye (42% accuracy vs 93.7% accuracy), comparable with extension officers' diagnoses (85-90% accuracy), and gives immediate results compared to days/weeks-long laboratory tests.

4.7.2 Strengths of the System

The evaluation highlighted various unique strengths of the AgroAI Doctor compared to current available systems:

- **Duo Mode Inference:** The efficient combination of both TFLite and API cloud inference modes ensures the application works irrespective of the internet connectivity problem, overcoming one of the biggest challenges facing cloud-based applications in rural areas of Nigeria.
- **Local Customization:** Besides translating text into the local language, the system recommends local agrochemical products and uses appropriate measurements such as bags and acres along with considering proper farming calendar dates.
- **Resource Efficiency:** The model is compact at only 3.6 MB, while the optimized Flutter mobile application consumes only 18.4 MB APK size and works effectively even on affordable smartphones.
- **Continuous Update Process:** The ability to constantly update the cloud-based model of the system makes it more effective and allows adaptation to the evolving trends of plant diseases.
- **Privacy-Oriented Structure:** Private information of the farmer stays mainly localized to his device without being uploaded to the cloud.

4.7.3 Limitations and Challenges

Several constraints and suggestions for improvements were discovered during the evaluation:

- **Disease Coverage:** Currently, only 12 types of diseases belonging to three crops are included in the system. There are plenty of diseases that affect other economically viable crops, including rice, yam, and sorghum.
- **Early Stage Detection:** Although the model performs well for detecting moderate and severe symptoms, the detection of early stage symptoms (initial discoloration, small lesion) proves to be difficult due to lower accuracy (approximately 75%).
- **Similar Diseases:** Certain disease pairs, which have similar symptoms at an early stage (CBSD & CMD; bacterial and fungal wist), are more likely to be confused. Thus, farmers need to be educated about the confidence score.
- **Device Requirements:** The application needs to be run on a smartphone having a working camera and at least 2 GB RAM, which makes the system unavailable for the farmers with basic phones or very old ones.
- **Data Costs:** Although the application has an offline functionality, cloud synchronization and model update require network and thus cause additional costs (in cases when mobile data plans are expensive).

6. **Usability:** Although the application received good usability ratings, certain farmers needed multiple attempts to master its usage, especially farmers without any previous experience using smartphones.

4.8 Summary

The implementation of AgroAI Doctor has been described in this chapter starting from the setting up of development environments to assembling the application and conducting tests. The CNN model was successfully trained by leveraging MobileNetV2 model through transfer learning, attaining an accuracy rate of 94.7% in lab tests, and optimized by quantizing it into 3.6 MB TFLite model.

AgroAI Doctor managed to achieve an accuracy rate of 94.7% on lab data and 93.7% on the ground data collected by Nigerian smallholder farmers which was far better than visual inspection diagnosis of 42% and even comparable to the extension officers' diagnosis of 85-90%.

All requirements have been tested and validated and revealed that all requirements both functional and non-functional are met including a time requirement of 250ms for performing TFLite inference, offline mode working properly without any bugs, correct cloud synchronization when internet connectivity is intermittent, and finally the SUS score for usability being 77.5.

The field test conducted by the application has shown great success and validation of its practicality among Nigerian smallholder farmers by obtaining high trust levels with confidence score over 85%, actionable and relevant treatment recommendations, and offline functionalities being essential. The cost-benefit analysis reveals that the farmer will experience a net profit of ₦44,500-123,500 per farming season in average cases of 1 hectare farming operation.

Although there may be some limitations such as limited disease coverage, difficulty with early stage diagnosis, and reliance on devices but the results show successful addressing of research gap identified in chapter two.

The Flutter application has been developed based on Clean Architecture where separate folders are available for the presentation layer, the domain layer, and the data layer which made it possible to develop and test each of the layers independently.

Some main functionalities of the application that were implemented include camera functionalities, TFLite inference model for diagnosis, cloud API communication, and data synchronization.

A well-designed asynchronous API for the backend service was developed using FastAPI and incorporated authentication mechanism, rate limiters, and exception handling. Supabase is used as the cloud-based database, authenticator, and storage with real-time functionalities.

As can be seen from performance tests, all non-functional requirements have been satisfied including time requirement of 250ms for performing TFLite inference and application maintaining acceptable memory footprint and battery efficiency.

CHAPTER FIVE

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.0 Introduction

The chapter outlines the summary of the whole research, making conclusions based on the results achieved through implementation and evaluation of the system and offering recommendations for future research directions. The chapter will synthesize the important aspects of AgroAI Doctor project and its contribution within the context of digital agriculture in Nigeria and Sub-Saharan Africa.

5.1 Summary of the Study

The current research has been aimed at development and evaluation of AgroAI Doctor – an AI-based mobile application that helps to detect diseases and provides recommendations about their management specifically for smallholder farmers in Nigeria. The main reasons for conducting the research were associated with poor efficiency of traditional methods of disease detection due to their poor accuracy, limited access to experts and high cost of the treatment.

In Chapter One, the background to the research study was laid down and highlighted the urgent requirement of accessible, accurate, and affordable disease diagnosis tools. Six particular objectives were outlined, which include developing an intuitive user interface, creating a CNN-based disease detection model, implementing cloud computing, making recommendations, analyzing the performance of the system, and comparing the results with conventional procedures.

A detailed review of literature in Chapter Two discussed the effect of crop diseases on Nigerian agricultural industry, challenges with current diagnosis methodologies, applications of artificial intelligence and deep learning techniques in plant pathology, mobile applications in agriculture, and architecture of cloud computing. Based on the discussion, six important gaps in research were found, namely localization, user-centric approach, lack of holistic solutions, practical testing, hybrid cloud architectures, and linguistic problems.

In Chapter Three, the System Analysis & Design is discussed. It explains that agile software development and OOAD (Object Oriented Analysis and Design) were selected as the primary methods. Chapter three describes the system's functionalities, requirements, design (three tiers including Flutter front end, FastAPI back-end, Supabase database), CNN model design (MobileNetV2 using transfer learning), and principles of designing the user interface for illiterate farmers.

Chapter four outlines the process of implementing the system. It mentions that training was done using 12,800 pictures belonging to 12 different disease categories with test accuracy of 94.7%. The CNN model was optimized using quantization and made into 3.6 MB TFLite file. The application was built using Clean Architecture having features such as camera integration, offline inference using TFLite models, fall-backs to cloud API, multilingual capabilities, and offline-first database synchronizations.

Test and evaluation were conducted in Chapter Five. Laboratory test involved testing 1,460 images and revealed 94.7% accuracy. Real world testing was done involving 127 farmers from three states in Nigeria resulting in 93.7% diagnostic accuracy. Test of system usability resulted in SUS of 77.5 (good usability). The application proved to be better than farmer visual diagnosis (42% accuracy) and close to that of the extension officers (85-90%).

5.2 Conclusion

Based on the findings from the research conducted, the following conclusions are reached:

1. AI-based Mobile Disease Diagnosis is Feasible and Functional: The findings of the research prove that AI techniques such as the use of convolutional neural networks like MobileNetV2 with transfer learning and quantization are capable of delivering clinically viable (>90%) crop disease identification accuracy on the mobile platform. The 93.7% accuracy recorded in the field environment shows that clinical success extends into practical farming environments but suffers only minor deterioration due to environmental factors.

2. Dual Inference Mode Architecture Resolves Issues of Intermittent Connectivity: The hybrid dual inference mode that involves on-device inference using TFLite in the event of offline operations while relying on cloud API for higher accuracy resolves connectivity problems experienced in rural Nigeria. In essence, the dual inference model ensures 100% availability of core functions while cloud services improve the accuracy where possible.

3. User-friendly Interface Design Ensures Successful Adoption of Technology among Illiterate Farmers: With SUS score of 77.5 points alongside the qualitative analysis conducted, it becomes clear that it is possible to create mobile applications that are user friendly even to individuals with little formal education and technological know-how.

4. Localization Ensures Maximum Practicality: Recommendations which utilized locally available agrochemicals, local measurement units, and regional farming practices were perceived as significantly more practical. Providing support in two languages – English and Hausa – made the app more comprehensible and credible to farmers in the northern part of Nigeria who speak Hausa.

5. Digital Technology is a Solution to the Extension Problem: Due to the extension-to-farmer ratio of 1:3,000, conventional extension services can never afford to provide prompt individualized disease diagnosis to farmers in sufficient quantities. AgroAI Doctor proved that digital technology could help to address this issue, giving farmers instant access to expert-level diagnostics without a proportional number of experts.

6. Economic Viability Far Exceeds Expenses: According to the cost-benefit calculation performed for this project, the net benefit per farming season will range from ₦44,500 to ₦123,500, meaning that its ROI will exceed 100 times.

5.3 Recommendations

Taking into consideration the research findings and limitations observed, the following are some recommendations for future development and implementation:

1. Disease/Crop Extension: Future development can extend the model to cover other important Nigerian cash crops such as rice, yam, sorghum, cowpea, and groundnut. The database can be extended to cover new diseases in the region and the partnership between the researchers and agricultural research institutions needs to continue for continuous collection of information.

2. Sensitivity Improvements: Further research may include increasing sensitivity of model detection in case of early stages of diseases using techniques such as few-shot learning, anomalies detection, and super-resolution image processing. Hyperspectral/multispectral images can further increase sensitivity in early-stage detection (as long as the device capabilities allow it).

3. Incorporating Audio/Video Content: Future development may integrate the possibility of voice narration of recommendations and videos demonstrating technique for applying remedies. In this way, audio/video can help improve accessibility to the application, especially for those farmers with poor literacy skills.

4. SMS-based Service Development: Since many farmers cannot afford smartphones and use feature phones, SMS/USSD-based diagnosis service may prove useful. Farmers can describe their observations while the server will provide automatic recommendations.

5. Develop Sustainable Distribution Networks: Collaborations between the agricultural extension, farmers' cooperatives, and networks of agronomist distributors can be formed for app deployment and user training. Integration of the app within government initiatives can secure sustainable financing and national scalability.

6. Ensure Continuous Education and Modeling Upgrades: A system that continuously updates models through farmer-validated diagnoses and corrections from experts needs to be implemented. This can take place with federated machine learning that allows for continuous improvements without compromising farmer data privacy.

7. Conduct Comprehensive Evaluations: Long-term impact studies should include monitoring and measurement of adoption rates, behaviors, and economic effects throughout several growing seasons to substantiate the developmental impact of the app. Randomized control trials against a control group can further validate the impact claims made by the developers.

8. Consider Integration with Precision Farming: In the future, precision agriculture applications such as IoT soil and weather monitoring, as well as drone imagery, may be incorporated into the software, making it a more holistic solution for crop management.

9. Solve Data Cost Challenges: Collaborations with mobile network providers will enable free usage of the data by zero-rating data cost for agricultural applications, or offering subsidized data plans for the farmers. Mesh networking or use of agricultural centers' Wi-Fi hotspots could also **cut down data cost per person.**

10. Design Desktop Application for Extension Officers: An accompanying web application for extension officers and agricultural researchers to compile data anonymously and generate maps of diseases, outbreaks, and intelligence information about crops at both regional and national level.

5.4 Contribution to Knowledge

The proposed research contributes in the following way to the existing body of knowledge in digital agriculture, machine learning, and human-computer interaction.

1. Localized CNN model for African Crop Disease: This is the first ever localized CNN model created and validated using Nigerian agricultural disease images showing that with the help of transfer learning, it was possible to achieve over 90 percent accuracy from limited local data. The dataset and model weights are publicly available for further researches.

2. Hybrid Offline-Cloud Based Mobile Framework for Agriculture AI: This paper describes an innovative architecture which uses a combination of offline TFLite-based inference model and cloud API call for inference when connection is available.

3. Design Approach for Low-Literacy Agriculture Mobile Apps: Validation of design principles (visual-based navigation, progressive information exposure, cultural adaptation) via field study conducted with 127 farmers, adding to the few HCI works on design solutions for Africa's smallholder population.

4. Cost-Benefit Analysis of AI Agricultural Tools in Nigeria: One of the initial studies examining the costs and benefits of artificial intelligence agriculture mobile applications in Nigeria, proving that AI-based agricultural disease diagnosis is highly profitable for farmers.

5. Performance Benchmarking: Comparison of AgroAI Doctor's performance in diagnostics to conventional diagnostic techniques including farmer observation, extension agents, and laboratory analysis.

5.5 Final Remarks

The creation and testing of AgroAI Doctor can be considered a major step towards democratizing access to AI-based agricultural knowledge for smallholder farmers in Nigeria. Leveraging both cutting-edge deep learning models and carefully considered design alongside reliable cloud infrastructure, the proposed system bridges an important gap in existing rural agricultural extension programs.

The project clearly shows that even sophisticated AI systems, which are often associated with the exclusive access by well-funded organizations in developed nations, can be successfully modified for use in resource-poor settings, provided that appropriate technical improvements (model quantization, efficient neural networks) are complemented by proper design considerations (localization, offline functionality, user interface).

As the effects of climate change continue to increase the pressure on crops due to diseases while at the same time worsening food security in sub-Saharan Africa, digital solutions such as AgroAI Doctor provide an opportunity to give expert diagnostics capabilities to millions of smallholder farmers.

However, the process from inception to deployment in the real world has proven our assumption that, through appropriate design and evaluation, Artificial Intelligence can overcome the digital divide and help some of the most disadvantaged food producers in the world. AgroAI Doctor is an example of how the Fourth Agricultural Revolution can be not only beneficial but also available to small-scale farmers.

REFERENCE

Saleem, M. H., Potgieter, J., & Arif, K. M. (2020). Plant disease detection and classification using deep learning. *Plants*, 9(11), 1–23.

Shi, Y., et al. (2024). *Lightweight neural networks for edge-based agricultural disease detection*. IEEE Transactions on Artificial Intelligence, 5(1), 33–47.

Barbedo, J. G. A. (2021). Impact of dataset quality on deep learning models for plant disease classification. *Computers and Electronics in Agriculture*, 180, 105–115.

Abbas, A., Jain, S., Gour, M., & Vankudothu, S. (2021). Tomato plant disease classification using transfer learning with convolutional neural networks. *Procedia Computer Science*, 167, 293–302.

Bandyopadhyay, R., et al. (2020). Identification and diagnosis of plant diseases using molecular techniques. *Annual Review of Phytopathology*, 58, 321–346.

Dhaka, V. S., et al. (2021). A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. *Sensors*, 21(14), 4749.

FAO (Food and Agriculture Organization). (2023). *The State of Food and Agriculture: Digital Agriculture*. Rome: FAO.

IITA (International Institute of Tropical Agriculture). (2022). *Nuru: AI for African Agriculture*. IITA Publication.

Mwombe, S., et al. (2023). User-centered design considerations for agricultural mobile applications among smallholder farmers. *Information Technology for Development*, 29(2), 215–230.

Nwachukwu, C. U., et al. (2021). Assessment of plant pathology laboratory capacity in Nigeria. *Nigerian Journal of Agricultural Research*, 12(3), 45–59.

Ogunsumi, L. O., et al. (2022). Evaluation of agricultural extension services in crop disease management across Nigeria. *African Journal of Agricultural Extension*, 10(1), 55–70.

Ojo, T. O., et al. (2024). Hybrid cloud-edge architectures for smart agriculture systems. *Journal of Agricultural Informatics*, 15(1), 89–102.

Omondi, J., et al. (2023). Performance of deep learning models on African crop disease datasets. *AI in Agriculture*, 7, 12–25.

Too, E. C., et al. (2022). Evaluation of mobile applications for crop disease detection and advisory services. *Computers and Electronics in Agriculture*, 196, 106–119.

Abbas, A., et al. (2021). *Application of deep learning techniques for plant disease detection*. Journal of Agricultural Informatics, 12(3), 45–58.

Akinwale, A., et al. (2021). *Smart farming technologies and agricultural productivity in developing countries*. African Journal of Agricultural Research, 16(5), 789–802.

Elijah, O., et al. (2021). *Internet of Things (IoT) for smart agriculture: Technologies, practices, and challenges*. IEEE Access, 9, 123–145.

Federal Ministry of Agriculture and Rural Development (FMARD). (2021). *National agricultural technology and innovation policy*. Government of Nigeria.

GSMA. (2023). *The mobile economy Sub-Saharan Africa 2023*. GSMA Intelligence.

Howard, A., et al. (2019). *MobileNetV2: Inverted residuals and linear bottlenecks*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4510–4520.

International Institute of Tropical Agriculture (IITA). (2022). *Cassava disease monitoring and management report*. IITA Publications.

Javaid, M., et al. (2022). *Artificial intelligence in agriculture: Applications and challenges*. Journal of Industrial Information Integration, 25, 100–115.

Legg, J., et al. (2021). *Cassava virus diseases: Biology, epidemiology, and control*. Advances in Virus Research, 110, 85–142.

Ogunlela, Y., & Mukhtar, A. (2020). *Gender issues in agriculture and rural development in Nigeria*. African Development Review, 32(2), 201–212.

Pressman, R. S., & Maxim, B. R. (2019). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill.

Sommerville, I. (2020). *Software engineering* (10th ed.). Pearson.

Supabase. (2024). *Supabase documentation*. Retrieved from https://supabase.com/docs

TensorFlow. (2024). *TensorFlow Lite guide*. Retrieved from https://www.tensorflow.org/lite