AI-BASED CASSAVA LEAF DISEASE DETECTION AND CLASSIFICATION USING DEEP LEARNING

BY

UZOCHUKWU TOCHUKWU BRIGHT
GOU/U22/CSC/803

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY, GODFREY OKOYE UNIVERSITY, ENUGU

JULY, 2026

AI-BASED CASSAVA LEAF DISEASE DETECTION AND CLASSIFICATION USING DEEP LEARNING

BY

UZOCHUKWU TOCHUKWU BRIGHT
GOU/U22/CSC/803

SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE,

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY, GODFREY OKOYE UNIVERSITY, ENUGU

IN PARTIAL FULFILMENT OF THE AWARD OF BACHELOR OF SCIENCE DEGREE

SUPERVISED BY ENGR DR. CALISTUS UGORJI

JULY, 2026

CERTIFICATION

This is to certify that this research project titled "**AI-BASED CASSAVA LEAF DISEASE DETECTION AND CLASSIFICATION USING DEEP LEARNING**" was carried out by Uzochukwu Tochukwu Bright with the Matriculation Number GOU/U22/CSC/803 under my supervision.

____

Uzochukwu Tochukwu Bright
GOU/U22/CSC/803

APPROVAL PAGE

This is to validate that this research work titled **AI-BASED CASSAVA LEAF DISEASE DETECTION AND CLASSIFICATION USING DEEP LEARNING**” was carried out by **UZOCHUKWU TOCHUKWU BRIGHT** with the matriculation number **GOU/U22/CSC/803** in partial fulfilment of the requirements for the award of Bachelor of Science (B.Sc.) in Computer Science,

Faculty of Computing and Information Technology, Godfrey Okoye University, Enugu.

____________________________	____________________________
Engr. Dr. Callistus Ugorji (Supervisor)	Date
____________________________	____________________________
Dr. Franklin Uchenna Okebanama Head of Department	Date
____________________________	____________________________
External Examiner	Date

DEDICATION

This research work is dedicated to God Almighty and also to my parents, Mr and Mrs Uzochukwu. Their unwavering support, sacrifice, and belief in my potential have been the foundation of everything I have achieved to this day. I pray that the good lord continues to guide, protect, and bless them abundantly for all the days of their lives.

ACKNOWLEDGEMENTS

First of all, I would like to appreciate the almighty God for his abundant mercies, inspiration and grace throughout this project. In a special way, I want to also acknowledge and appreciate my supervisors, Engr Dr. Callistus Ugorji for the time he brought out to make sure that this project is a success and also for guiding me throughout this project.

I would also like to appreciate my lecturers in the person of Dr. U.F. Okebanama, Dr. I.A Ajah, Engr. Kingsley Chibueze, Dr. Echezona and Prof. G.E. Okereke

I would also like to appreciate my parents, Mr. & Mrs. Uzochukwu for their love, support and prayers for me throughout this project, my brother also, Engr. ifeanyichukwu for his support, and I pray that the good Lord will continue to bless them all.

ABSTRACT

The reliance on manual, subjective visual inspections for the identification of cassava crop pathologies remains a critical bottleneck in sub-Saharan agricultural productivity, often leading to delayed interventions and significant yield losses. This research presents the design and implementation of a robust, AI-powered predictive system engineered to automate the detection and classification of diseases affecting cassava leaves. Leveraging the **YOLOv11** deep learning architecture, the developed system facilitates high-precision, real-time diagnostic capabilities by processing foliar imagery to identify and classify specific pathogens while simultaneously highlighting localized areas of infection. The model was trained and validated using a hybrid dataset—integrating open-source repositories with localized field samples from Nigeria—to ensure superior generalizability across diverse environmental and phenotypic conditions. Adhering to object-oriented design principles, the platform features an intuitive web-based interface that empowers farmers and agricultural extension officers to make data-driven management decisions. Empirical evaluation of the system yielded a **Top-1 accuracy of 87.95%** and a **Top-5 accuracy of 100%**, effectively demonstrating the model's reliability in distinguishing between complex disease patterns. Ultimately, this study provides a scalable, sustainable technological solution to enhance food security and optimize crop health monitoring within regional agricultural workflows.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 **BACKGROUND TO THE STUDY**

A disease can be conceptualized fundamentally as a physiological condition that is anomalous which compromises the functional mechanism of an organisms and is therefore distinguished by a discrete manifestation of clinical signs or indicators that are symptomatic (Al Husaini et al., 2025). However, within the contexts of agronomics, the structural degradation of plants is primarily instigated by agents that are pathogenic which include parasitic vegetation, fungal colonies, bacterial organisms, viruses and microscopic nematodes (Eliwa, & Abd El-Hafeez, 2025).

Consequently, these widespread infections severely disrupt the availability of food globally and also the output of agricultural operations, therefore frequently culminating in depletion of the yield of crops catastrophically. Thus, when dating back in time, agriculturalists have ever since relied on the direct inspection and the direct observation of farms by humans to identify and manage this outbreak of viruses. Moreover, notwithstanding the fact that they are used widely out of necessity, these traditional paradigms have ever since then proven to be exceedingly inefficient, labour-demanding, and highly susceptible to diagnostic inaccuracies (Egere et al., 2025).

However, the macro-variations in climates and also the epidemics of crops that are evolving have chronically posed threats that are severe to the global agricultural production and the sustainability of macroeconomics as one a poignant historical precedent is the great Irish famine of the 1840's which was precipitated by the rapid proliferation of the phytophthora infestans (late blight) pathogen that triggered a great mass of demographic displacement and catastrophic starvation. However, in contemporary history, emergencies of virus such as the wheat rust epidemics that was localizes across east Africa and the banana wilt crisis In the central Africa have caused shocks to the regional food networks in a way that is devastating and also to the stability of microeconomics. (Kalezhi & Shumba, 2025). On this note, these remedies that are traditional have manifested frequently as unviable economically while still introducing server damages that are ecological, most notably the widespread of toxicity in the environment (Alimon 2025).

Conversely, the breakthroughs in technology that are recent within the domain of computational data analytics have yielded predictive models that are sophisticated and also algorithmic frameworks that are capable of projecting environmental threats that may occur in the future and also the anomalies in the biological make up of plants. Therefore, within the domain of modern agronomics, a primary application of there systems that are computational involves the forecasting of the propagation of pathogens by systematically evaluating the history of disease matrices and variables that are multi-spectral (Zhong et al., 2025). Furthermore, these prognostic platforms that are intelligent optimize agrarian management of workflows that are related to agriculture by reducing the dependency on agrochemicals that are toxic and also enabling responses that are swift and targeted to localized crop infections (Peng et al., 2025)

Moreover, in the recent times, the precision and reliability of the plant pathology diagnostics have changed significantly due to the integration of the frameworks of machine learning and deep neural architectures into software that are predictive. Thus, some of the methodologies of machine learning that are standard have streamlined the identification of phenotypic patterns that are intricate in a great way by processing historical datasets of documented cassava diseases that are extensive (Kalezhi & Shumba, 2025). However, this whole process therefore enables high-precision of anomalies in an automated manner and also multi-class labelling of pathogens thereby reducing the severity and widespread of crop failure drastically (Li et al., 2020).

In summary, the engineering and successful deployment of architectures that are predictive which merges the structures of machine learning together with algorithms of deep convolutional networks in other to monitor and forecast the vectors of cassava disease therefore represent a solution that is vital to current vulnerabilities in agriculture. However, this methodology sterns forward to equip modern farmers with the digital tools that are necessary for the rapid, automated diagnosis and also crop treatment that is informed, therefore marking the shift toward cultivating an agricultural framework that is both structurally resilient and sustainable environmentally.

1.2 STATEMENT OF PROBLEM

The traditional method of identifying the pathology of crops and making decisions that are informed on how to counter these diseases has brought about a lot of challenges, which include but are not restricted to the following:

1. The incapacity of agricultural practitioners to correctly identify and diagnose cassava diseases of crops using conventional detection and diagnosis techniques.
2. Inefficiency of the currently existing traditional methods of collecting data, leading to the gathering of inaccurate data.
3. The available data is inaccurate, resulting in an irregular use of insecticides, leading to both decreased yield and an increase in the overall cost of crop management.
4. Farmers currently waste a lot of time trying to keep an eye on their farms, which in general, lowers the efficiency of labour.

1.3 AIM AND OBJECTIVES OF THE STUDY

The primary aim of this project work is to develop a predictive model for detecting and classifying the cassava crop leaf disease by making use of the algorithms of machine learning**.** However, this project work intends to be able to succeed in this by following the objectives that are listed below, which are to:

1. Investigate and characterize the limitations that are structural and inherent in the manual traditional approaches of diagnosing cassava crop pathologies.
2. Compile, clean, and then normalize a dataset of cassava folia images that is robust and also optimize them for training and validating deep learning models.
3. Engineer and optimize a detection architecture that is automated by leveraging on the framework of YOLOv11 for the localization and classification of cassava diseases at the early stage.
4. Construct a user-friendly web interface that is intuitive and dedicated to deploying, testing, and evaluating the pipeline that is predictive in the scenarios of the real world.
5. Critically assess the performance of the platform with regards to diagnostic accuracy and the reduction in the application of agrochemicals in a way that is indiscriminate.

1.4 SIGNIFICANCE OF THE STUDY

This research work is significant to the following groups of people and institutions**:**

1. Cassava farmers (small-scale and large-scale)
2. Agricultural extension officers
3. Agricultural researchers (Academic community)
4. Government and policy-making bodies in the agricultural sector and the general society.

However, as cassava farmers continue to suffer from serious losses of crops as a result of an inaccurate or delayed identification of leaf diseases, this turns out to reduce the yield and crop quality of their crops. Therefore, this project is significant to the above-mentioned people in the following ways:

1. To the cassava farmers, it provides them with a practical tool that they can use for early detection of diseases, thereby allowing them to intervene in time before the disease spreads extensively across all their crops, thereby reducing the use of pesticides and minimizing the loss of crops.
2. This project is significant to the agricultural extension officers as it offers them a reliable decision-support system that helps them enhance the advisory services that they provide to farmers.
3. This project contributes heavily to the field of academics by demonstrating the application of advanced deep learning models, particularly YOLO-based architectures, in the detection of diseases in agriculture.
4. This project is also of importance to the government and the policy makers, as it provides a technological solution that greatly supports the security of food and also promotes a method of agriculture that is sustainable.

1.5 SCOPE OF THE STUDY

This project work of research focuses mainly on the development of a model that is predictive which would be used for the identification of cassava crop diseases at the early stages using machine learning algorithms. Geographically, this study is limited to Nigeria, focusing on the particular agricultural conditions and farming practices, and cassava varieties common to the Nigerian landscape.

This project emphasizes on addressing the difficulties that are encountered by farmers when they are trying to make decisions regarding the identification of diseases that affect their crop plants in a way that is well-informed. However, the scope of this project extends beyond the landscape of the farmer to include people and organizations that engage in the production of cassava for both personal and commercial reasons.

1.6 LIMITATION OF THE STUDY

The study was limited by inadequate advanced technical resources. Some of these limitations are as follows:

1. **Technical Expertise:** The researcher encountered issues that are associated with the lack of professional capabilities in the areas of software development, which were necessary for the successful deployment of the system.
2. **Inefficiency in Data Precision:** The underlying gaps, noise, and inaccuracies that exist within the environment, soil, or crop yields that are available within the datasets for training the model will directly compromise the reliability of the operation of the system and also its confidence in protection.
3. **Applicability Constraints:** This project work focuses mainly on the technical workflows that are computational and therefore does not account for socio-economic variables that are external. Therefore, factors such as volatile market dynamics, asymmetric distribution of resources, and cultural traditions of farming, amongst others, fall outside the scope of this project and therefore limit its application in the real world.

1.6 OPERATIONAL DEFINITION OF TERMS

There is a lot of terminology and terms that are associated with this project work, but for the purpose of this project, we will just look at a handful, which are mentioned below.

1. **Crop Disease:** An anomalous physiological or pathological state in the vegetation of crops..
2. **Pathogen:** Any biological entity that is capable of inciting disease.
3. **Predictive Model:** A computational or statistical framework that is engineered to forecast prospective events.
4. **Deep Learning:** This is a specialized discipline within the machine learning domain that uses neural architectures that are multi-layered in order to autonomously extract features that are highly dimensional from complex inputs.
5. **Algorithm:** This is a step-by-step mathematical or computational procedure that is designed to resolve a specific problem or execute a defined task.
6. **Neural Network:** A biomimetic computational architecture that functions as a primary framework for machine learning tasks.
7. **Convolutional Neural Network (CNN):** A specialized class of deep neural networks optimized explicitly for extracting spatial features.
8. **Image Analysis:** It is defined as the algorithmic or manual processing of visual data to extract meaningful structural information.
9. **Machine Learning:** a subfield of artificial intelligence that is dedicated to developing software platforms that execute decisions directly from empirical datasets.
10. **Artificial Intelligence (AI):** the act of computers performing tasks that normally require human intelligence.

11. **Data Set:** This is A structured compilation of homogeneous or heterogeneous data points,

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter of this project reviews the different literatures that are relevant to this project as it helps both the researcher and the readers to fully understand and also be aware of the developments, background and the different limitations of the studies which are in one way or the other related to this project work. At this, it provides justification for this project by pointing out what has been achieved by previous researchers and the shortcomings/gaps of the studies that needs to be filled. However, this part of this project not only exposes the gaps in knowledge that is already existing, but also assesses the pros and cons of the solutions that are available through extensively analysing various academic journals, research papers and other previous project reports as well.

2.1.1 DESCRIPTION OF THE DIFFERENT CATEGORIES OF CASSAVA CROP DISEASE PREDICTION SYSTEM

The software that are predictive which are used in monitoring the health of crops can be seen to generally fall into four structural categories that are distinct which are based on the configuration of data. These categories include the following.

1. **Image-Based Vision Systems:**

   This term refers to the framework for predicting the diseases of crops which apply computer vision and convolutional neural networks (CNNs) directly foliar imagery for the purpose of image recognition that is non-invasive and high-speed spatial. One good example of this type of system is the TomatoDet model, which analyses the matrices of leaves in other to rapidly diagnose early and late blights and at this establishes a paradigm that is efficient for crop analysis that is non-contact (Sakkarvarthi et al., 2022).

2. **Environmental Telemetry Models:**

This topology of the cassava disease prediction system bypasses the analysis of leaf images completely by using streams from microclimatic and localized IoT sensors (monitoring relative humidity, surface temperature, and soil moisture metrics) that is continuous to calculate the incubation and outbreak probabilities of pathogens mathematically (Lee and Yun, 2023).

3. Hybrid Multi-Modal Topologies:

This category of cassava disease prediction system work by fusing input data that are heterogeneous dynamically. It does this by joining merging visual imagery, atmospheric telemetry, and the registries of historical outbreak with the use of early or late decision-level fusion of machine learning. However, this approach that is multi-layered balances out the inherent blind spots of data streams that are isolated (Sanida et al., 2023).

4. Genomic Data-Based Models:

This is a branch of cassava prediction systems that is emerging in the direction of research which evaluates the genetic sequences and molecular profiles of planta in other to map alleles that are specific and linked to hereditary disease immunity or vulnerability. However, the field implementation that is practical remains bounded by the cost of infrastructure and the researches that is active is focused on integrating makers that are genomic into predictive analytics (Thangaraj et al., 2022).

2.2 THEORETICAL BACKGROUND

2.2.1 Overview of cassava leaf disease detection

The cassava crop stands as one of the crops that are very important in the tropical region, most especially in Africa, where it stands to play a very important role as both a source of food and income for large part of the population. One of the characteristics of the crop is that it can withstand both drought and poor soil but still suffers heavily from different leaf diseases that are usually responsible for the crop's poor growth and low yield (Jafar et al., 2024). However, various studies in the past have shown that the diseases that are associated with the cassava plants are mainly as a result of microorganisms such as viruses, bacteria, and fungi that infest the leaves, stems, or roots of the plant to cause biological disorders. Studies have shown that the Cassava Mosaic Disease (CMD), Cassava Brown Streak Disease (CBSD), and Cassava Bacterial Blight (CBB) are the three most severe cassava leaf diseases that cause the highest damage as the pathogens that cause these diseases wound the leaves, thus interfering with the process of photosynthesis, and as a consequence, the plants grow slower, the roots get less developed, and even the worst case of total crop failure may happen (Legg et al., 2019).

However, observing from the theoretical aspects of plant pathology it is clear that the first signs of disease usually come with prominent signs that are very visible like discoloration of leaves, distortion of the veins, or forming of spots on the plants (Delfani et al., 2024). These signs are of high importance as detecting them early could lead to reducing the areas infected in a way that is considerable and also reduce the risk that is associated with loss of harvest. Nevertheless, such symptoms are too often unable to be recognized with certainty by farmers lacking expert knowledge, especially in cases where several diseases are presenting visual patterns that are similar. Therefore, it is a firm argument in favour of the implementation of smart systems that would be able to learn, identify, and categorize the symptoms of diseases from the cassava leaf images with great accuracy (Ramcharan et al., 2017).

2.2.2 Traditional Methods of detecting cassava leaf diseases

Traditional ways of identifying cassava diseases have for long now been heavily dependent on the physical examination of farmers, agricultural extension officers, or plant pathologists. This method involves expert knowledge that these farmers and officers have acquired over the years through experience and observation in the field (Basavant et al., 2024). However, the subjectivity of professionals, fatigue and also the lack of trained experts in rural agriculture communities are some of the drawbacks of these conventional methods, although there are still somewhat effective in in some ways. In many instances, these cassava diseases gets to be diagonized wrongly or detected too late, which in turn leads to wrong treatments and more extensive crop loss (FAO, 2020).

On the other hand, laboratory-based methods of diagnosis like polymerase chain reaction (PCR) and serological tests are very accurate but, at the same time, are not suitable for everyday use on farms because they are costly and take a lot of time to be done (Wang & Liu, 2024). Moreover, these techniques of testing demand the use of advanced equipment and skilled labour, which is usually not available in some of the countries which are still developing. Therefore, the inefficiency of manual and laboratory-centric methods, in large-scale or real-time agricultural monitoring contexts, is made clear from a systems theory viewpoint by these limitations.

2.2.3 The Application of Machine Learning in the detection and classification of cassava leave disease

Machine learning on its own is a part of Artificial Intelligence (AI) that allows computers to recognize the trends in the data that is given to it and then use it to either predict or decide on their own what to do next, without the need for explicit commands (Prottasha et al., 2024). The foundation of machine learning therefore is in statistical learning theory that gives the rules of how algorithms can use the knowledge that is acquired from training data and then apply it to the cases that were not included in the training. However, in terms of detecting diseases in cassava, its proper to apply machine learning as the disease symptoms do show up in a way that can be recognized and learned through the process of working with huge sets of leaf pictures that are labelled (Mohanty et al, 2016).

Deep learning on the converse side, is a sub-sector of machine learning, helps to improve this approach by using Neural Networks alongside a lot of layers to extract features that are meaningful from raw images. Convolutional Neural Networks (CNNs), on its own are particular suitable and well adapted to the treatment of images since they imitate the way the human visual system processes shapes, textures, and spatial relationships (Basavant et al., 2024). YOLO models on their own tends to take this one step further by merging together the feature of object detection and classification into one framework that is unified, thereby allowing the system to not only find the areas of infected plants but also tell the type of disease at the same time. However, newer versions like YOLOv11 are based on this principle of enhancing accuracy, feature extraction, , and speed, making them suitable for application in real-time in farming (Bochkovskiy et al., 2020; Redmon et al., 2016).

Forging ahead, the use of machine learning in predicting the diseases of the cassava crop has bridged the gap between the field agricultural science and computer science. These systems diminish the need for an expert by acquiring their knowledge from pictures that were taken in the fields and at that releases the detection of diseases early, thereby allowing for the making of decisions to be guided by data (Wang & Liu, 2024). This theoretical underpinning therefore warrants the employment of the models of deep learning such as YOLOv11 in this project because it corresponds not only to the biological attributes of cassava diseases but also to the computing principles that are necessary for precise and fast prediction.

2.3 REVIEW OF EXISTING RELATED WORKS

The development of diagnostics systems that are automated in the domain of agriculture have continuously built upon a foundation of macine learning and computer vison research that is very rich. However, this section of this project work of research stands to provide an appraisal of the architectures that are existing in a critical manner, in other to highlight the technical milestones, operational performance and development constraints that these exiting architectures have achieved. Some of these existing systems include the following:

Prottasha et al., (2021) all put their resources together to develop a CNN system that is lightweight for the purpose of identifying the pathologies of 12 rice plant in Bangladesh. The system was able to achieve a 95.4% validation accuracy and at that the real-world generalizability of the system remained constrained as a result of limited size of testing dataset.

Shoaib et al., (2022) embarked on a study where they tested the inceptionNet model against the U-Net variants on 18,000 images of tomato leaves and at that, they U-Net algorithm that they modified was able to achieve a segmentation accuracy of 98.66%, while on the other hand, the InceptionNet produced 99.955 when it was tested for binary classification and also 99.12% when it was tested for six-class segmentation.

Gottemukala et al., (2024) joined forces to formulate a diagnostic feature pipeline that is early-stage for the purpose of detecting the diseases of the cassava plant. In their work,They combined the Principal Component Analysis (PCA) and also the Ant Colony Optimization (ACO) in other to minimise the dimensionality of features and also the complexity of computations without having to sacrifice the reliability of diagnosis.

Khan et al., (2024) proposed a deep learning and machine learning model that is hybrid and stacking in other to classify 10 categories of tomato leaves. They trained the model with the dataset from PlantVillage and at the end of their research, the system that they developed was able to reach an accuracy of 98.27% and a 0.174-second speed of inference, even though that it lacked a user interface that is intuitive.

Liu and Wang (2020) in their study compared the extraction of features that are crafted by hand with architectures that are end to end. The system that was developed as a result of their study made use of the YOLOv3 engine with image pyramids that are multi-scale and at this demonstrated that the object detection models of deep learning yield a localization accuracy that is superior under field conditions that are volatile and natural.

Manchanda et al., (2024), worked on creating a CNN classifier that is fine-tuned for the identification of cassava diseases in agriculture in real-time especially in zones like india. However, their system was able to automate the process of diagnostics to a labour that is reduced, but then it lacked a validation that is rigorous against datasets that are external.

Kaushik et al., (2023), engineered the TomFusioNet system. the system was a mobile framework for the analysis of the tomato crop diseases using the NSGA-II algorithm and also a subtraction background that is based on HSV. However, the system demonstrated an accuracy that is high i.e. 99.93% DeepRec, 98.32% DeepPred, but then it was limited ecause it was never tested in environmental conditions that are volatile.

Aishwarya et al., (2023), built a Custom Convolutional Neural Network (CCNN) with three Convolutional layers in other to classify 10 classes of tomato leaves. Their study resulted in the development of a network that is streamlined helped in reducing the latency of processing when compared to the deep architectures that are standard, even though that the edge deployment of the system in the real-world required exploration that is deeper.

Yildiz et al., (2024), evaluated the machine learning models that are classical i.e. RF, KNN, LR, ANN, SVM using the orange software on a dataset that was gotten from Kaggle. However, their study resulted in the development of a Python-implemented classifier that is custom which was able to achieve a highest classification performance of 96% thus outperforming the baseline of models that standard.

Jha et al., (2024) formulated an ensemble transfer learning model that was fine-tuned on 10,403 images of different crops. The system combined the MobileNet algorithm and the backbones of Inception and the result of the system when it was tested was able to reach a diagnostic accuracy of 98.95% and at this outperformed the architectures that are standalone.

Tan et al.(2021)., worked on a research where they compared a feature descriptor that was crafted by hand i.e. LBP, GLCM and colour histograms against the pipelines of deep neural network on a dataset that was gotten from PlantVillage. However, while the matrix of COLOR+GLCM was able to optimize machine learning that are classical, A model of ResNet34 was responsible for setting the performance ceiling with the use of a 99.7% score that was uniform across all metrics

Badiger and Mathew (2023) joined forces to design the DbneAlexNet which was a deep learning model that is optimized using a framework of the Gradient Jaya-Golden Search Optimization (GJ-GSO). The system they developed was able to achieve a classification accuracy of 92.4% but it failed to address the accessibility of the software to smallholders that are local.

Ullah et al., (2023) work on developing DTomatoDNet which is a neural architecture that is lightweight and which was trained from scratch on 10,000 images across 10 classes. The system was able to achieve an accuracy of 99.345 for pathologies that were targeted, but then it lacked a repository that is open and public for the deployment of the system.

Zhou et al., (2021), worked on reconstructing a Residual Dense Network (RDN) for the identification of tomato diseases. The system merged the residual connections algorithm with the features of a dense blocks features reuse that were enhanced and the efficiency parameters however, the system yielded an average top-1 accuracy of 95% that is suitable for edge computing.

Verma et al., (2019), built a cloud-hosted CNN classifier that was trained using 3,000 images of tomato leaves. the model that was developed as a result of their study was able to achieve a classification accuracy of 98.49% by optimizing the hyperparameters of the network. However, the system was limited because the its Google Collab infrastructure restricted the use of the model users who are only tech literate.

Tarek et al., (2022), put their resources together to benchmark a convolutional model that is pre-trained in other to reveal that the Mobile NetV3 architectures outperformed the standard networks. However the MobileNetV3 Large was able to hit an accuracy score of 99.81%, therefore recording an edge latency 348ms that is low when it was tested on a Raspsberry Pi 4.

Alzahrani and Alsaade (2023), worked together to evaluate the DenseNet169, ResNet50V2 and the Vision Transformers (ViT) on foliar images of the tomato plant. However, the DenseNet121 was able to achieve the highest performance with a 99.00% testing accuracy, which demonstrated a training and inference speed that is very fast.

Agarwal et al., (2020) all put in equal efforts to engineer a nine-layered CNN model that is lightweight in other to minimize the overhead of computation. However the custom network that they developed was able to achieve an accuracy of 98.4% on the data of PlantVillage, therefore outperforming the standard VGG16 model which produced an accuracy of 93.5%) while still showing a cross-dataset generalizability of 98.7%t that is strong.

Singh et al., (2024) pulled their resources together to benchmark the classifies of deep learning against the models that are classical on the different pathologies of tomatoes. The result of their study shoed that the InceptionV3 achieved the baseline that is highest in terms of performance with an accurary of 97%, followed closely by the DenseNet (94%), while on the other hand the classical Random Forests lagged at a performance floor of 68%.

Jasim (2021) single handedly explored the convolutional network that is deep for the purpose of diagnosing the health of plants in a way that is non-invasive using a dataset of 18,000 images of multiple species that are available publicly. However the CNN model that was developed was able to reach a peak accuracy score of 97% thereby validating the potential of deep learning for a crop management approach that is scalable.

2.2 SUMMARY OF REVIEW OF RELATED WORKS

S/N	AUTHOR(S)	WORK DONE	TECHNIQUE	limitation
1	Prottasha et al., (2021	Developed a CNN architecture that is lightweight for detecting diseases that affect rice in Bangladesh	CNN architecture	Not enough data for testing
2	Shoaib et al., (2022).	Developed a system that identified the disease of tomato plant using leaf images.	U-Net framework	Was never deployed
3	Gottemukkala et al., (2024).	Was able to reduce the complexity of computation.	PCA (Principal Component Analysis)	The system wasn't specific.
4	Khan et al., (2024).	Identified images that are blurry in 0.174 seconds	Convolutional Neural Network (CNN)	The system was not easily accessible to users.
5	Liu, J., & Wang, X. (2020).	Improved the feature extraction of traditional methods.	YoloV3	The study made use of an outdated model.
6	Manchanda et al., (2024).	Automated the detection of diseases and Enhanced the applicability precision and generalization.	Convolutional Neural Network (CNN)	The system's proficiency wasn't determined.
7	Kaushik et al (2023)	Tackled the challenges of disease detection in real-time.	NSGA-II algorithm	The diverse environmental conditions were never tested.
8	Aishwarya et al., (2023).	Detected and classified the tomato plant diseases.	Convolutional Neural Network (CNN)	The system was never deployed to a publicly accessible repository.
9	YILDIZ et al., (2024)	Enabled the management and diagnostics of cassava diseases.	Hybrid algorithm	Was not tested for applicability in the real world.
10	Jha, P et al., (2024)	Addressed the food security challenges that are evident in India.	Convolutional Neural Network (CNN)	The scope was limited to India.
11	Tan et al., (2021)	Showed that the Deep learning models are superior for tasks that are related to classification.	Hybrid algorithm	The study was just a review and didn't develop any model.
12	Badiger, M., & Mathew, J. A. (2023).	Demonstrated how deep learning and sophisticated optimization methods could be combined for precision agriculture.	Gradient Jaya-Golden Search Optimization (GJ-GSO) algorithm	Did not put the acceptability of the model by the local farmers into consideration.
13	Ullah et al., (2023)	Created a lightweight design that is suitable for mobile application.	Deep learning algorithm.	Was never made available for the public.
14	Zhou et al., (2021).	Created a system that achieved a superior performance compared to	deep residual dense networks	The combination of different features

		other methods while still requiring fewer computational resources.		came with a lot of complications.
15	Verma et al., (2018).	Helped to improve the yield of crops through the detection of diseases early.	Residual Dense Network (RDN)	The system was only available to users that are technologically inclined.

2.4 CONTRIBUTION TO KNOWLEDGE (KNOWLEDGE GAP)

The agricultural sector of the sub-Sahara has for long now relied heavily on the manual visual auditing to diagnose the diseases of crops and while the systems that are automated are exiting in literature, they still suffer from geographic data bias that were critical because they are trained almost exclusively on the datasets that are gathered from macroclimatic zones that are entirely different. Meanwhile, due to the variations in weather, illumination and cultivar phenotypes in different regions, these models that are currently exiting have failed to generalize and experience performance degradation that is severe when deployed in sub-Saharan fields that are native.

However, this project work of research resolves these bottlenecks of validation directly by deploying a dataset strategy that is hybrid by fusing repositories that are open source with field samples that are localized and captured under environmental conditions that are native in order to establish a geographical contextualized model for regional precision agronomy that is very accurate.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.1 OVERVIEW

Crop diseases have for long now remained one of the factors that is most significant which reduce the productivity of agriculture across the world, as farmers often experience losses that are severe when the diseases that affect their crops are not detected early enough. However, cassava which serves as a major staple food crop in many tropical regions is vulnerable to a lot of diseases and the early identification of the diseases plays a critical role in minimizing the loss of crops. This chapter of this project work, however, focuses mainly on the analysis and design of the system that has been proposed by this study using Object-Oriented Analysis and Design (OOAD) methodology and Unified Modelling Language (UML) diagrams. It also explains the internal structure of the system, evaluates the solutions which are already existing, identifies their limitations and most importantly presents the design models that guides the implementation of the proposed system.

3.2 SYSTEM ANALYSIS

System analysis in this project is a term used to describe all the activities that are involved in the examination of the structure of the proposed system in other to understand its operations and how its components interact with each other. However, in this project work the system analysis focuses on the understanding of the challenges that are associated with the detection of cassava leaf disease and evaluating the different approaches that are already existing which have been used in previous studies. It also examines the weaknesses of those systems and presents an analysis of the proposed predictive model that is based on YOLOv11 in a detailed manner.

3.2.1 ANALYSIS OF EXISTING SYSTEMS

The conventional paradigm of identifying and detecting the diseases of plants have required the agricultural workers to manually navigate through the fields of their farm to conduct plant-by-plant phenotypic visual inspections. Based on these, these approach is highly labour-demanding and inflicts operational strains that are severe, therefore making it incompatible fundamentally with commercial farming with high-throughput.

However, in other to automate the process of diagnostics, Prottasha et al. put their resources together and engineered a CNN model that is streamlined and optimized to identify 12 different diseases of the rice plant in Bangladesh. Similarly, the Deep Tomato Detection Network (DTomatoDnet) made use of an architecture that has 19 layers paired with 1 to 1 convolutional kernels in other to suppress the parameter density and therefore execute edge diagnostics that makes use of resources that are low using a dataset of 10,000 images.

Moreover, Verma et al., developed a CNN Framework that is specialized and designed to protect the yields of crops through the detection of early-stage anomaly as a result of being trained on 3,000 foliar images. However, their system processes inputs through a structured pipeline encompassing the preprocessing of images, segmentation of a region that is targeted and the extraction of a feature that is deep and focused on colour, texture and edge metrics.

3.2.2 WEAKNESSES OF EXISTING SYSTEMS

Despite the different systems that are existing currently in this domain of study, these different frameworks that are currently exiting fall short in some domains that are critical which affects farmers ultimately. As a result of this, some of these prominent shortfalls include the reality that these traditional detection methods are tedious, labour-demanding and prone to error, which in most cases results in prognoses and crop diagnoses that is biased and subjective. Furthermore, a lot of the existing computational models that were developed using data parameters that are incomplete as they are sourced entirely from the regions that are not in the sub-Sahara, therefore rendering them not suitable for deploying farmers that are within the environments of the sub-Saharan.

However, beyond all these limitations, the structural weakness that are specific stems from other existing systems which includes the following:

1. **Susceptibility to Overfitting**: Due to their simplicity in structure, many models that are existing currently are highly prone to overfitting when processing datasets that are highly dimensional or noisy.
2. **Lack of Adaptability to Imbalanced Data**: The dataset of plant pathology exhibit class imbalances that are severe frequently, therefore simple algorithms routinely struggle to handle this data distribution that is skewed effectively, leading to a model prediction that are biased and significantly reduced the sensitivity toward the minority classes that are critical .
3. **Vulnerability to Missing Data**: most of the systems that are exiting lack data streams that are robust, internal mechanisms to navigate missing or incomplete, which are common challenges in real-world agricultural environments. Therefore, this handling of datasets that are incomplete in an inadequate manner introduces statistical bias therefore reduces the reliability of the predictions of these models severely.

3.2.3 DATA COLLECTION

The Empirical collection of data represents a critical phase that is foundational in engineering or a roust model that is predictive for the purpose of diagnosing the pathology of crops, as the input of data quality dictates the performance of models ultimately. This project work of research therefore establishes a data ingestion pipeline that is heterogeneous by harvesting imagery from agricultural institutions that are in the sub-Saharan regions alongside the database of PlantVillage.

However, the pipeline of collecting data involves a direct collaboration with smallholders in regional areas and other agronomic specialists in other to capture foliar images of both healthy and diseased cassava crops that are of high resolution under different microclimatic and illumination conditions, thereby ensuring that the dataset that is used in training the model absorbs variations of phenotypic disease manifestations that are localized. Therefore, each of the visual asset is strictly paired with a metadata that is structured such as the cultivar type and the indices of disease severity in other to provide the context that is essential for the training of models using samples of the harvested datasets that is representative as expressed below:

Figure 3.1 Some of the image data that was gathered represents the normal leaf, mosaic disease, brown streak disease, bacterial blight to name a few.

3.2.4 ANALYSIS OF THE PROPOSED SYSTEM

The diagnostic system that is proposed in this project work of research addresses the loopholes and gaps in empirical knowledge that is systemic and inherent to the agricultural methods of making decisions which are traditional by using the eastern Nigeria as a geographic case study. This computational architecture therefore integrates a macroclimatic dataset that has high fidelity from online repositories that are authoritative, together with the algorithms of deep learning with are optimized explicitly to predict and classify folia diseases of the cassava crops. However, unlike the single-variable approaches that are conventional, this framework elevates precision that are semantic by dynamically weighing a geographical data that is nuanced, soil variations and other atmospheric conditions that are unique to the ecosystem of eastern Nigeria, thereby yielding agronomic recommendations that are both highly accurate and at the same time localized.

Moreover, the proposed framework which is holistic in nature therefore pushes forward to help farmers make decision that are driven by data by combining physical signs in realtime with analytics that are predictive, in other to allow them to anticipate the impact of the environment on the health of their crops. Therefore, by fusing climate telemetry that are advanced with machine learning, this proposed system offers a solution that is comprehensive in other to revolutionize regional agricultural making of decisions and also foster agrarian practices that are sustainable. Methodologically, the proposed system objectives is achieved through the use of a Late Fusion (Decision-level) paradigm of integration, which executes a YOLO object detection model independently for the analysis of visual foliar and a climatic model that is parallel, thereby subsequently combining their outputs together to produce a prediction that are highly resilient and at the same time multi-modal in nature.

3.2.5 SYSTEM REQUIREMENTS

System requirements constitute the technical specifications that are essential or parameters that define the operational function or capabilities a platform must exhibit explicitly. These architectural requirements however, are bifurcated formally into two primary dimensions which include the functional requirements and the non-functional requirements.

3.2.5.1 Functional requirements

1. **Feature Extraction and Selection**: The system must be able to execute the extraction of features in other to be able to identify the biomarkers of diseases that are suffered by crops from the input data by making use of feature selection in other to reduce the dimensionality and therefore maximize interpretability without compromising the accuracy of the system.
2. **Machine Learning Model Development**: The proposed framework must facilitate the configuration and training of algorithms that are diverse right from classifiers that are classical (Random Forests, Logistic Regression, SVMs) to the architectures of deep learning.

3. **Real-time Inference and Prediction**: The system must be able to support inference that has low latency, by delivering crop disease predictions that are instantaneous or risk scores directly from inputs in real time.

3.2.5.2 Non-Functional requirements

1. **Performance**: the proposes system must deliver a computational efficiency that is very high and also a fast processing time for handling data, training model as well as inference in other to enable field interventions that is rapid.

2. **Scalability**: The architecture of the system must be able to scale efficiently in other to manage the continuously expanding volume of data and the user loads that is continuously increasing without sacrificing the speed of operation or the reliability of the system.

3. **Reliability**: The system must be able to maintain an operational uptime that is very high and a service disruption that is minimal in other to ensure an availability that is continuous for farmers and professional researchers.

4. **Accuracy**: The model of the system must be able to maintain a very high precision in diagnosis, in other to minimize the rate of both false positives and false negatives so as to eliminate the risk of misdiagnosis .

5. **Robustness**: The system pipeline must always remain resilient against variations in data in other to effectively mitigate noise, outliers and values that are missing across farm environments that are heterogeneous.

3.2.6 PROPOSED METHODOLOGY: OBJECT-ORIENTED ANALYSIS AND DESIGN

Object-Oriented Analysis and Design (OOAD) works as a framework for software engineering that is foundational which leverages principal that are modular in nature to structure and execute systems that are complex. The process of object-oriented analysis and design therefore begins with the engineering of the system requirements in a rigorous manner, then progressing directly to the delineation of the classes of the system and their structural interdependencies. However, by integrating the unified modelling language (UML) schematics with use cases that are practical, the OOAD bridges the architectural design that is abstract and also a programmatic implementation that is concrete and at this visually maps the interactions of the different components of the system in other to guid e the whole system development process.

3.2.6.1 UNIFIED MODELLING LANGUAGE

The unified Modelling Language (UML) diagrams is a schematics diagram that provide a vital and standardized framework for the purpose of mapping the intricacies of a systems architecture visually. These diagrams are graphical abstractions which enable architects, developers, and stakeholders to be able to conceptualize and communicate structural designs collaboratively. However, among these diagrams, the use case diagram is arguably the most critical as it delineates boundaries between external actors and internal operations explicitly and with this, the use case diagram establishes the behavioural blueprint for deciphering functional system requirements in a very essential manner.

3.2.6.2 USE CASE DIAGRAM OF PROPOSED SYSTEM

From the point of view that is structural, the use case diagrams is seen to comprise of four core elements which include the use cases who’s function is to outline the system’s discrete functionalities, the actors that denotes the external entities that play different roles in the system, the associations which functions to map the mutual interactions that happen between the actors, and the system boundaries which established the architectural scope.

However, in the context of this project work's proposed framework, the operation of the system that is proposed in this study are driven by two main primary actors which are the user (specifically the farmer) and the system model. The user can initiate the start of the work flow by using the system to capture images or upload and already captured image to the system. on the other hand, the backend system will concurrently ingest the visual data, execute the processing of images digitally, diagnose the specific phytopathology of the leaf image via the features that it extracted from the image and then finally generate mitigation strategies that are actionable.

Use case diagram:

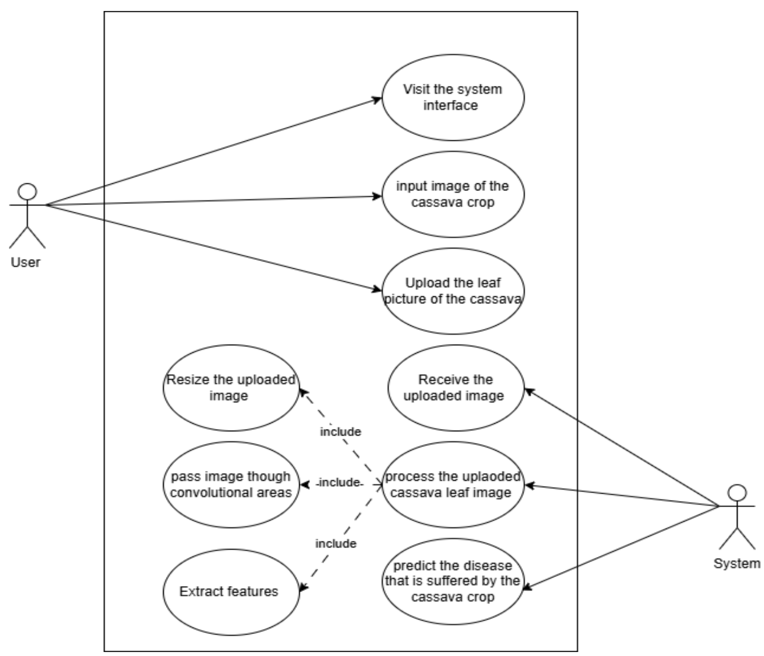


Figure 3.2 Use case diagram of the proposed system

The illustration that is presented above gives a clear depiction of the use case activity that occurs in the system that have been suggested in this project work the suggested system consists of two primary entities, and the model of the system and the user who can engage in a wide range of activities in the system which includes, visiting the system and uploading crop leaf images, while on the contrary, the system must receive the image that the user provided, evaluate it, identify the specific disease based on the features it found in the uploaded image, and then provide guidance on how to address and treat the plant disease, among other things.

3.2.6.3 ACTIVITY DIAGRAM OF THE PROPOSED SYSTEM

The activity diagram is one of the UML diagrams that serves as a vital representation of the behaviours that are vital in the Unified Modelling Language (UML) framework which complements the static structural models and as well maps the dynamic trajectories of a software system in an operational manner. The activity diagram therefore functions essential as an architectural flowchart that is very advanced as it delineates the operations of a software system workflows. However, the illustration that is attached below shows the execution of the proposed system in a behavioural manner, therefore tracing the operational pipeline from the initial time that the user who makes use of the systems accesses the system and uploads a leaf image which is ingested into the system model to the final stage of algorithmic classification of disease and the production of a diagnostic output.

Activity diagram of the proposed system:

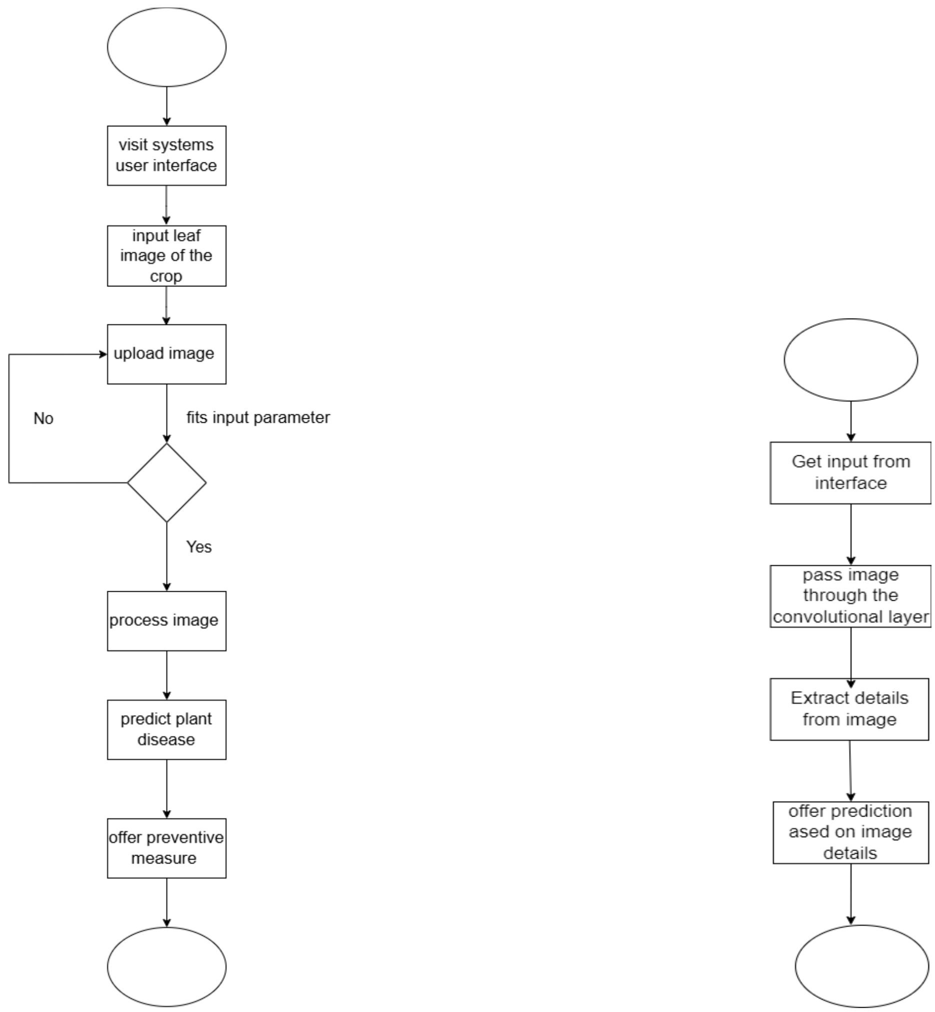


Figure 3.3 Activity diagram of the system and the analysis process of the system

The schematic illustration above shows the diagram of activity of the system that is proposed in this project which provides a diagrammatic representation of the operational workflow and control logic of the proposed system. the illustration shows the modelling of how the system functions as it maps the pipeline of execution of the system in a sequential manner by tracing the different processes that takes place in the system right from the initial time that the user (farmer) interacts with the system platform and uploads a leaf image to the time the uploaded leaf image is ingested through the backend algorithmic evaluation down to the final stage of classifying the pathology of the plant.

3.2.6.4 CLASS DIAGRAM OF THE PROPOSED SYSTEM

The class diagram stands to represent the system design of the platform that is proposed in this project in a structural manner by identifying the different classes that make up the system. This section of this project therefore shows how the attributes and methods, contained in each of the classes that are present in the proposed system work together hand in hand to define the functionality of the system. However, each of the classes interact with each other to ensure that images are uploaded, processed, analysed and stored properly within the system.

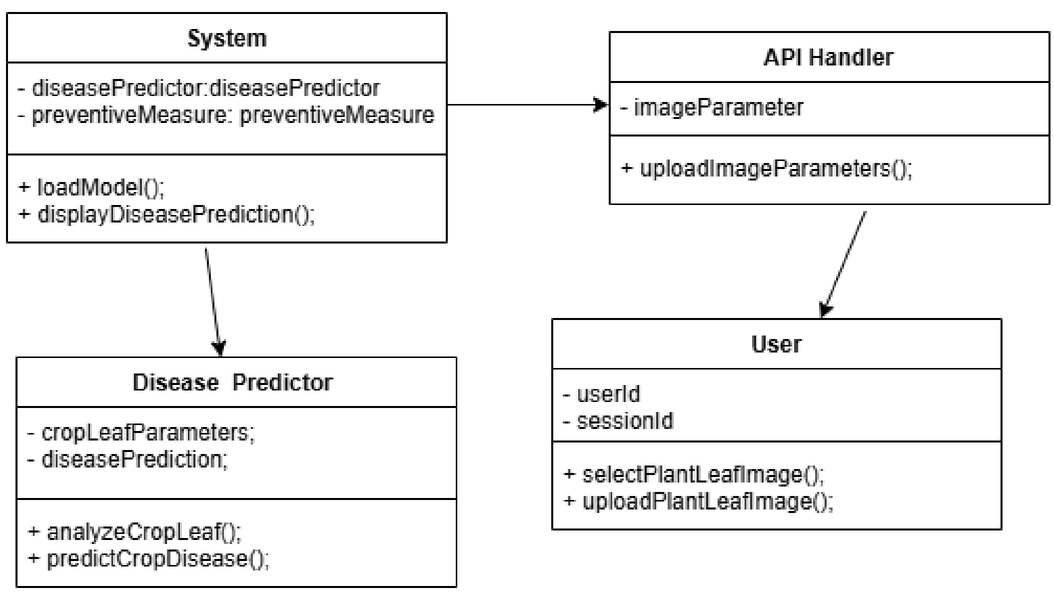


Figure 3.4Class diagram of the proposed system

The diagrammatical illustration that is attached above shows the proposed system's class diagram of which is attached in this project work for the sole purpose of illustrating the different proceedings that are taking place in the system that have been proposed in the earlier section of this project work. The illustration therefore shows the different classes that are existing within the system in other to enable it to function properly.

3.2.6.5 UML SEQUENCE DIAGRAM

The unified Modelling Language (UML) sequence diagram is an illustration that offers a temporal representation of a systems behaviours and which functions to map the discrete interactions of objects in a chronological other within the system. However, this architecture that is depicted by the sequence diagram is made up of lifelines that denote the entities that are participating in the system execution, the messages that represent the communication pathways between them, and the vertical activation bars which functions to quantify the exact duration of the object's lifecycle during the process of execution. Moreover, this sequence diagram captures the chronological dimension of the system and at this, helps the developers and stakeholders that are non-technical to gain an overview of both the sequence and precise timing of the systems operations in a way that is very transparent.

In the proposed system, the sequence of actions stars hen the user accesses the system interface and uploads an image of a cassava leaf. Once the image is uploaded, the uploaded image is processed by the preprocessing module and passed to the YOLOv11 deep learning model which detects the disease patterns that are present on the cassava leaf. After that, the disease is now predicted using the details of the patterns that have been observed from the leaf image, and the result of the prediction is now displayed on the system interface for the user to view.

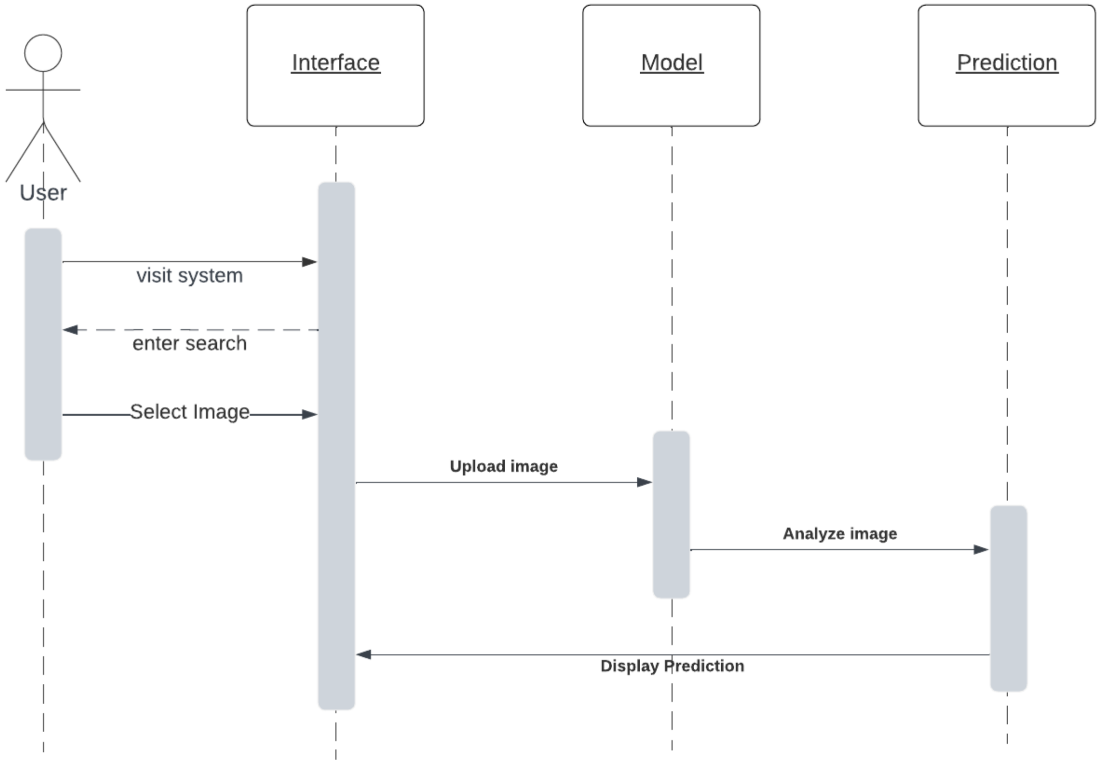


Figure 3.5 Sequence diagram of the proposed system

The illustration that is shown above shows the chronological order of executions that this happening in the proposed system starting from the time that the user visited the user interface of the system to the point which the system gives a prediction of the pathology of the plant using the image of leaf which have been provided by the user. The illustration therefore stands as a visual representation of the system that is proposed in this project work, thereby making it possible for researchers in the future to be able to see the processes that are internal in the system.

3.3 SYSTEM DESIGN

The system design section of this project encompasses the comprehensive process that is involved in defining and engineering the architecture, components that are modular, and the interactions that are internal within the framework of software that are complex. This section therefore integrates the interfaces of both the input and output sections as well as the architecture of the database, so as to ensure that the exchange and synchronization of data between the user and the platform is seamless. However, the objective of the system design is to construct a framework that will be able to satisfy the specifications of the user while still guaranteeing the efficiency of operations, scalability, reliability and a maintainability that is long-term.

3.3.1 INPUT INTERFACE DESIGNS OF THE PROPOSED SYSTEM:

This section of this project goes on to translate the system requirements that have been established previously into a software architecture that is tangible by presenting the structural design of the system's input interface. The design however shows the graphical layout that is explicitly engineered to optimize the accessibility of the system and therefore ensure that the core of the system's core analytical features remains very intuitive for the end users.

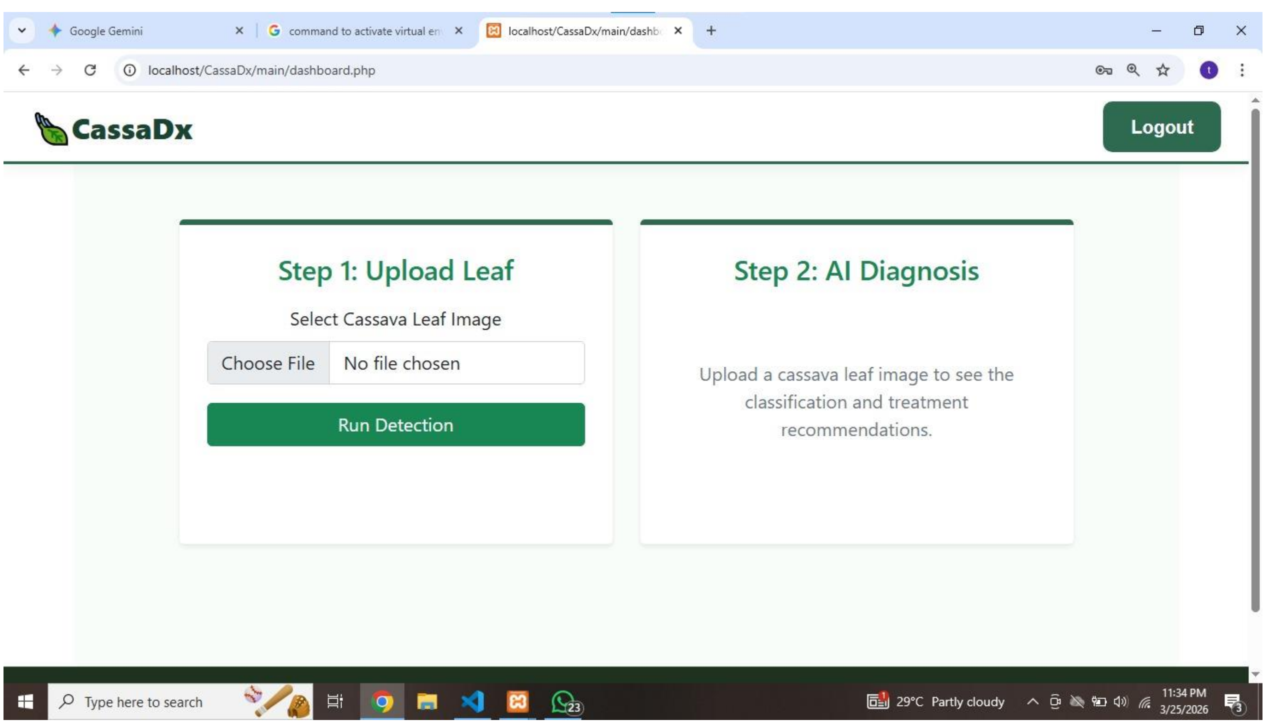


Figure 3.6 Input designs for the proposed system

The user interface diagram that is illustrated above delinates the primary portal for data ingestion in the proposes system in other to enable the operator to select and upload a localized crop leaf sample directly from their device onto the analytical platform.

3.3.2 OUTPUT INTERFACE DESIGN OF THE PROPOSED SYSTEM:

This section of this project work presents the architecture of the proposed system output interface as it demonstrates the modality of the system for rendering the data of diagnosis of plants post inference. The layout therefore illustrates how the result of classification of diseases and insights that are actionable are displayed following the algorithmic evaluation of the leaf image that is submitted by the user.

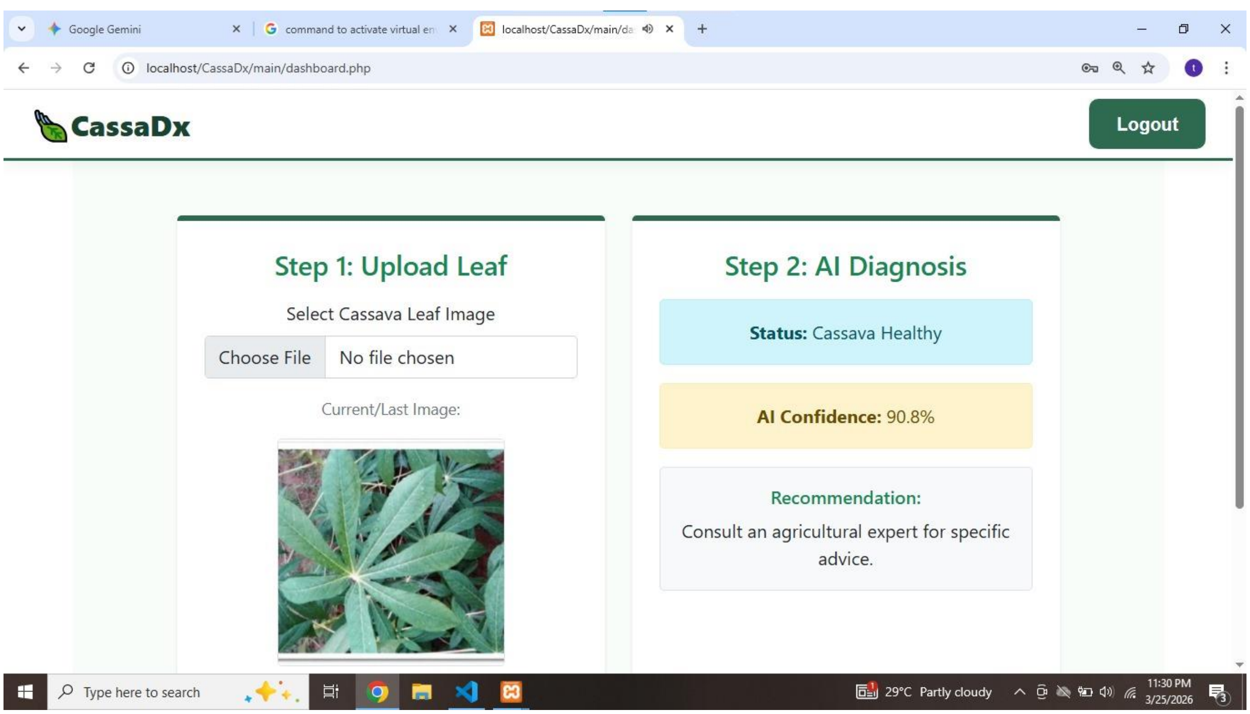


Figure 3.7 Output designs for the proposed system

The illustration above represents the output screen of the envisioned system. however the picture depicts how the output will the organized and shown on the screen to the user following the forecast of the plant pathology. Moreover, generally speaking, it is an illustration of the interface that the suggested system uses to output data.

3.3.3 DATABASE DESIGN

This section of this project discusses the architectural schema which provides a blueprint of the target database in a way that is abstract by detailing the structural tables that are designated for a continuous storage of data. However, this database design maps out the necessary system entities alongside their corresponding attributes to establish the schema of the proposed system application that is robust by being implemented as model that is relational, leveraging the querying framework of MySQL.

Database schema for the proposed system

Table 4.1 User table for the proposed system

Attribute	**Data Type**	**Description**
Id	Int (11)	primary key
Username	Varchar (50)	The username of the user
Email	Varchar (100)	The email of the user
Password	Varchar (255)	The password of the user
Date_joined	timestamp	The date a user signed up to the platform.

Table 4.2 Prediction table of the proposed system

Attribute	Data Type	Description
Id (pk)	Int (11)	Primary key
Prediction	Varchar (100)	The predicted disease
User	Varchar (50)	The user that made use of the system.
Session	timestamp	The session of the user.

Table 4.3 Suggestion table of the proposed system

Attribute	Data Type	Description
Id (pk)	Int (11)	Primary key
Prediction	Varchar (100)	The predicted disease
Suggestion	Varchar (255)	The systems suggestion to the farmer
Session	timestamp	The session of the prediction.

CHAPTER FOUR

SYSTEM IMPLEMENTATION

4.1 introduction

This chapter of this research work explains how the Cassava leaf disease detection and classification system that was proposed at the beginning section of this research was implemented in detail with respect to all the designs mock-up that have been outlined in the analysis and design section of this the project. This chapter however, pays keen attention on how the system moved from the point of conception and problem identification to a fully functional system. In addition to the above, this chapter also outlined all the technologies, tools and processes that were employed in the development of the system.

4.2 IMPLEMENTATION TOOLS (PROPOSED SYSTEM'S TECH STACK)

The process of implementing and bringing the system to life was done using tools and frameworks that are modern and are suitable for applications of computer vison that are based on the web and which function in real-time these tools that were employed during the implementation of this system are outlined below in detail.

4.2.3 Frontend

i. **Python:** This is one of the most versatile programming languages and it was used in this project work for backend development, AI model training, and also in building simple test graphical user interfaces.

ii. **JavaScript**: This is the most popular scripting languages and was adopted in this project work to add interactivity to the frontend of the system.

iii. **HTML:** This is a simple markup language that was used in the system development process to build the elemental structure of the system's user interface.

iv. **CSS3**: This is a cascading stylesheet that was used in this project work to structure and add styles to the interface of the system.

4.2.4 Backend

i. **Python:** This is one of the most versatile programming languages and it was used in this project work for backend development, AI model training, and also in building simple test graphical user interfaces.

ii. **PHP:** This is one of the major languages that is used for backend and server-side programming because of its lightweight nature and ease of use. It was employed in this research when writing the server-side logic and for file handling.

4.2.5 Frameworks and Libraries

i. **YOLOv11 (Ultralytics)**: This is a high-level library of Python and a computer vision platform that is designed for developing, training, and deploying AI models in real time.

ii. **OpenCV**: This is a computer vision library that is open source and which contains over 2,500 algorithms that are optimized for real-time computer vision, image processing, and machine learning. It was used in the development process of this system for video processing and frame extraction.

iii. **PyTorch**: This is a machine learning library that is open source and was built based on the Python programming language. It is used widely for building deep learning applications as it offers tensor computing with strong CPU acceleration that is flexible.

iv. **cURL**: This is a very powerful command-line library (libcurl) that is used for transferring data across networks, as it supports protocols like HTTP, HTTPS, FTP, and SCP. It was employed in the development of this system for the purpose of bridging the PHP backend to the AI script of python.

4.3 DEVELOPMENT PALTFORM AND HARDWARE REQUIREMENTS

The development process of the system was done using Visual Studio Code, which is a very extensive but easy-to-use coding IDE that is suitable and compatible with an array of programming languages and frameworks. However, the model was initially trained using Kaggle notebook, which is a simplified but powerful environment that is used for training machine learning models. Moreover, the hardware specifications of the personal computer that was made use of while developing the proposed system includes the following:

i. **Processor:** intel Core i7 or equivalent AMD Ryzen 7 processor
ii. **RAM:** 16GB
iii. **Storage:** At least 20 GB available disk space
iv. **GPU;** NVIDIA RTX 3060 (minimum of 6GB VRAM) for local training acceleration

After the training of the model, the model was deployed using the Linux-based ubuntu server with at least 4GB RAM to be able to power the interface.

4.4 DATASET DESCRIPTION AND PREPROCESSING

4.4.1 Dataset Description

The system was trained using a dataset that was made up of data gotten from the Kaggle Cassava Leaf Disease dataset, with a focus on Sub-Saharan Africa. The researcher also employed the use of live data that was gotten from farms in the Nigeria (a country in the Sub-Sahara region of Africa). The dataset was composed of leaf images of cassava crop indicating normal leaf, mosaic disease, brown streak disease, bacterial blight to name a few.

4.4.2 Data Processing

Before training the model, the dataset was cleaned and transformed so as to make it palatable for machine learning. These different steps include the following:

1. **Resizing**: All the pictures form the dataset was resized to a fixed dimension that is compatible with YOLOv11 (e.g., 640×640 pixels)
2. **Normalization**: The values of the images' pixels were normalized by scaling them to [0,1]
3. **Data Augmentation**: The researcher applied horizontal and vertical flips, adjusted the brightness of some of the images to simulate the harsh African field conditions, and also augmented the mosaic appearance of the leaf images.

4.4.3 Training and Validation Split

After the preprocessing of the dataset, the dataset was divided into 3 main subsets: the training dataset, the validation dataset and the testing dataset. A 70:20:10 split ratio was used where 70% of the dataset was used to train the machine learning model, 20% of the dataset was used to validating the model's performance and 10% of the dataset was used in testing the effectiveness of the system. However, this split ensured that the model generalizes well to unseen data.

4.5 MODEL ARCHITECTURE AND TRAINING

The system made use of YOLOv11 (You Only Look Once) models that have been pretrained based on Convolutional Neural Networks (CNNs) for the detection of objects. It makes use of a C3k2 backbone and a C2PSA (Cross Stage Partial with Pointwise Self-Attention(system in other to be able to perform at a high speed, high accuracy while extracting features from leaf images.

4.5.1 Training Parameters

Different parameters were adopted during the process of training the models of the system. These parameters include the following:

i. **Data:** "Kaggle/Workings/Cassava_data".

ii. **Epochs:** 50

iii. **Batch Size**: 16

iv. **Image_size**: 640

v. **Project:**"Kaggle/Working/runs".

vi. **Name:** "cassava_cls".

vii. **Device:** 0

4.6 SYSTEM IMPLEMENTATION AND MODULES

The system was implemented using a modular architecture and PHP acts as the bridge:

1. **Upload Module (HTML/JS/PHP):** This module handles the selection and secure transfer of the leaf image to the server.

2. **AI inference Engine (Python) Module:** This is a standalone script which is triggered by PHP. It loads the .pt model and processes the upload image action.

3. **Communication Bridge:** This is the way PHP uses the exec() or an API call to send the path of an image to Python and also receive a JSON response that contains the name of the disease and the confidence score.

4. **Result Display Module:** This module of the system is responsible for passing the JSON and then dynamically updates the User Interface of the system to show the diagnosis..

However, notwithstanding that the system is made up of different modules, all the modules in the system still work together synchronously to enable the system to function effectively and smoothly.

4.7 TESTING, EVALUATIONAND RESULTS

4.7.1. Evaluation metrics of the proposed system

In other to evaluate the performance and effectiveness of the YOLOv11 model in solving the problems that were outlined at the earlier section of this research work, the researcher made use of classification metrics that are standard. This classification metrics is discussed in detail below:

1. **Top-1 Accuracy:** This represents the percentage of the images that was used in testing the model, where the model's highest probability prediction was the correct label for the disease. However, based on the training logs, the model achieved a Top-1 accuracy of 87.95%

2. **Top-5 Accuracy:** This group shows the amount of the test images in percentage that where the correct label was among the model's top 5 predictions. However, the model achieved a 100% accuracy, thereby proving that even when the first guess was wrong the correct disease was always identified as a secondary possibility.
3. **Cross Entropy Loss:** During the training process of the model for the system, the error rate was monitored and at the end of the training process, the validation loss settled at 0.4877, thereby indicating that the model is converging well without an overfitting level that is significant.
4. **Confusion Matrix:** The confusion matrix is a model evaluation tool that measures how accurately a model identifies the different cassava leaf disease classes. Below is the image of the confusion matrix:

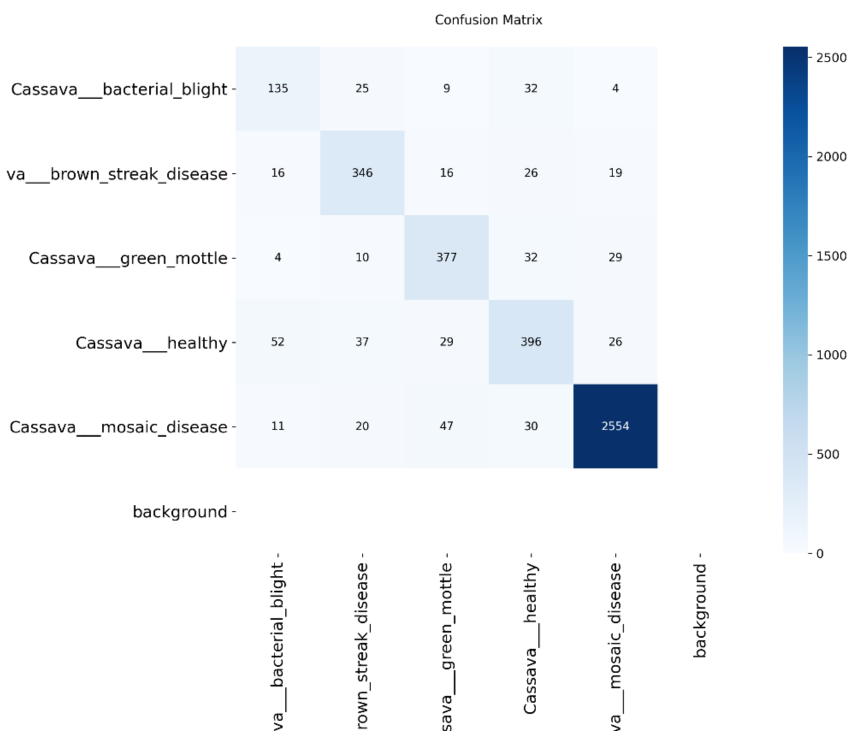


FIG 4.1: Diagram of the confusion matrix

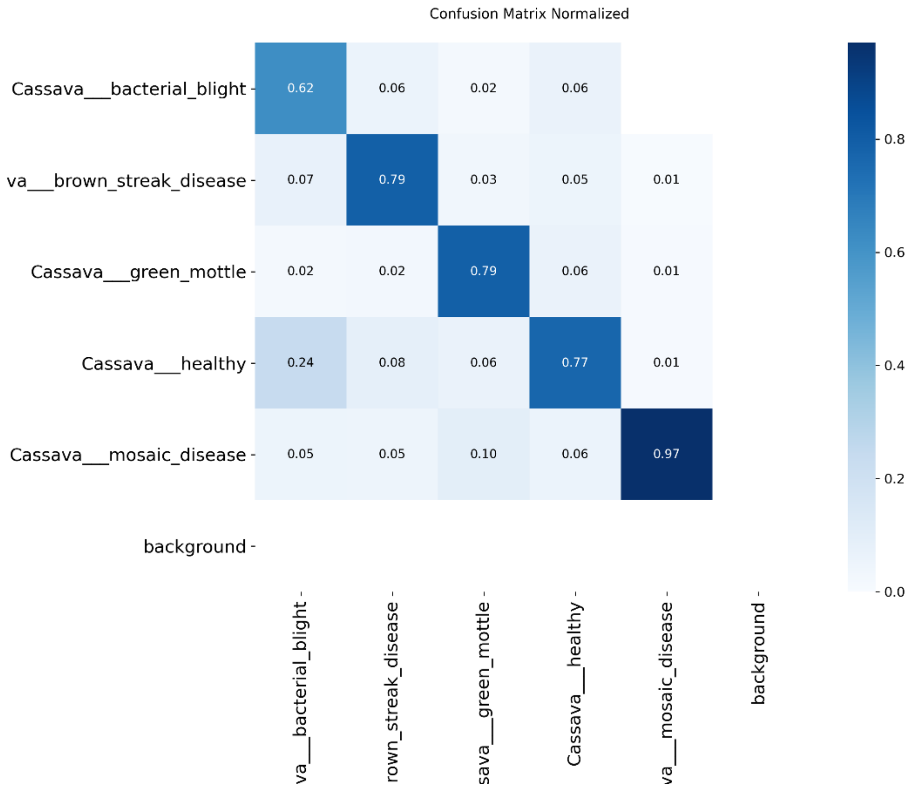


Fig 4.2: Diagram of the confusion matrix normalized

1. **Macro Precision:** Precision measures the accuracy of the positive predictions (out of all leaves the model claimed had a disease, how many actually did?)

   It is calculated with the formula:

   Precision = True Positives/True Positives + False Positives

   Precision = 0.620+0.790+0.790+0.770+0.970/5 = 0.788 or 78.8%

2. **Macro Recall:** Recall measures how perfectly the model finds the actual diseased leaves.

   It is calculated with the formula: Recall = True positives/True Positives + False Negatives

   Recall = 0.816+0.832+0.878+0.644+0.789/5 = 0.796 or 79.6%

3. **Macro F1 Score:** The F1 score is the harmonic mean of precision and recall.

   It is calculated with the formula:

   F1-score = 2 * Precision * Recall/Precision + Recall

F1-score = 2* (0.788* 0.796)/(0.788 + 0.796) = 0.792 or 79.2%

However, during the evaluation process, 2 graphs were plotted, the loss curve which showed the decrease in error as the model learns to identify unique necrotic patterns and mosaic discoloration on the leaves, and the Accuracy curve which illustrates ability of the model to classify diseases correctly as the training progresses. The images of the 2 graphs are attached below:

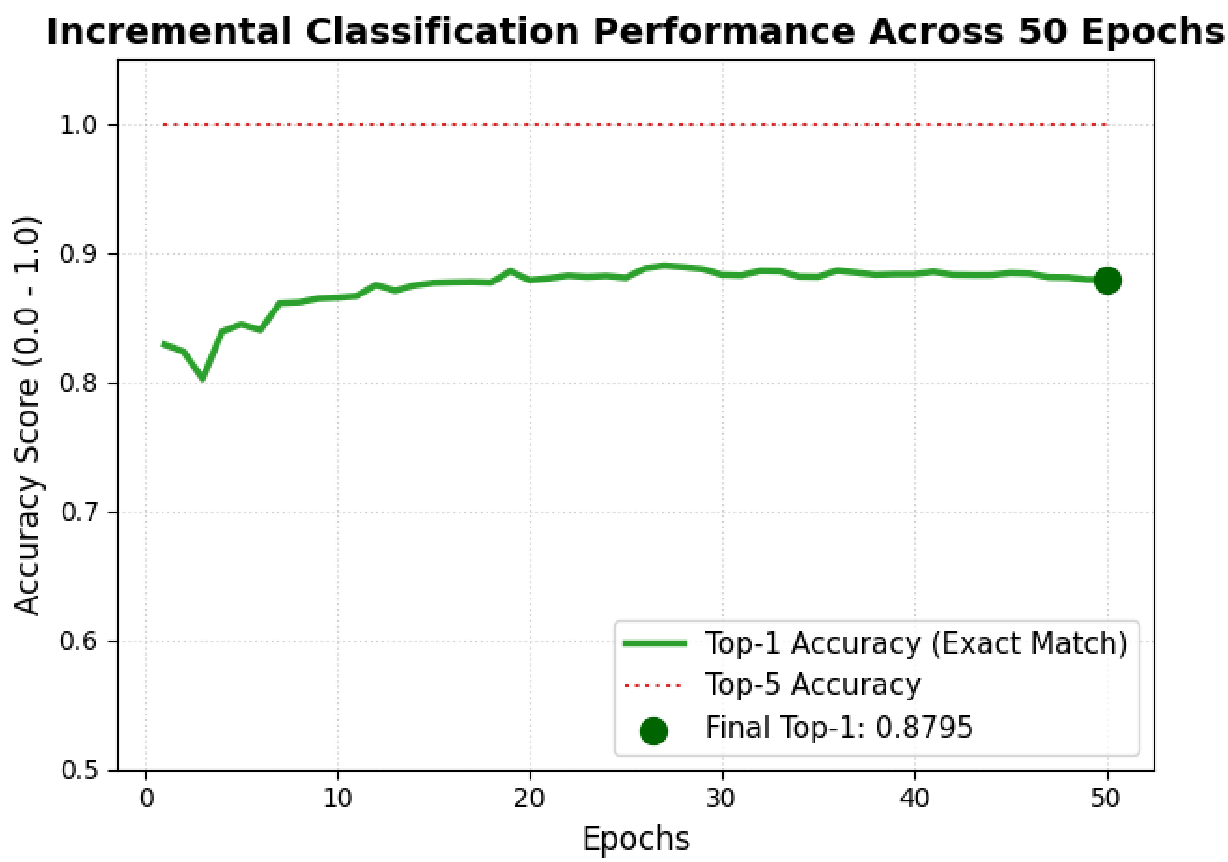


FIgure 4.3: Graph of the incremental classification performance across 50 epoch

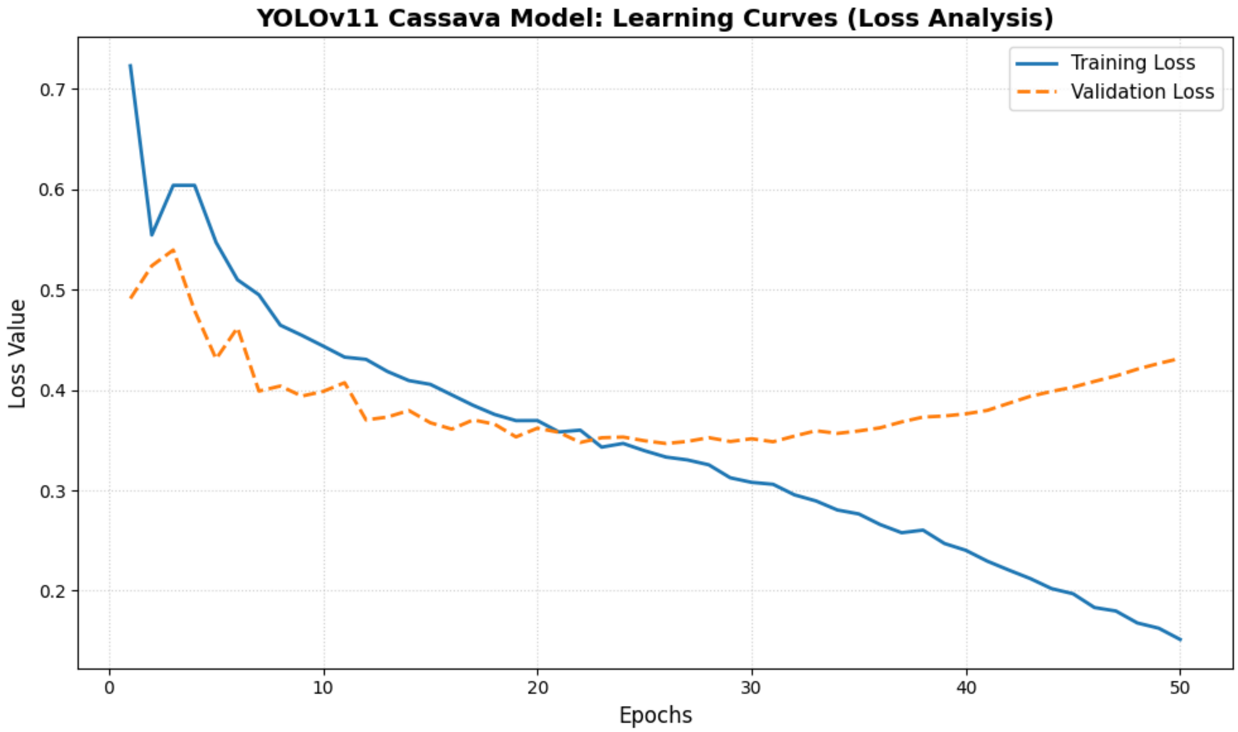
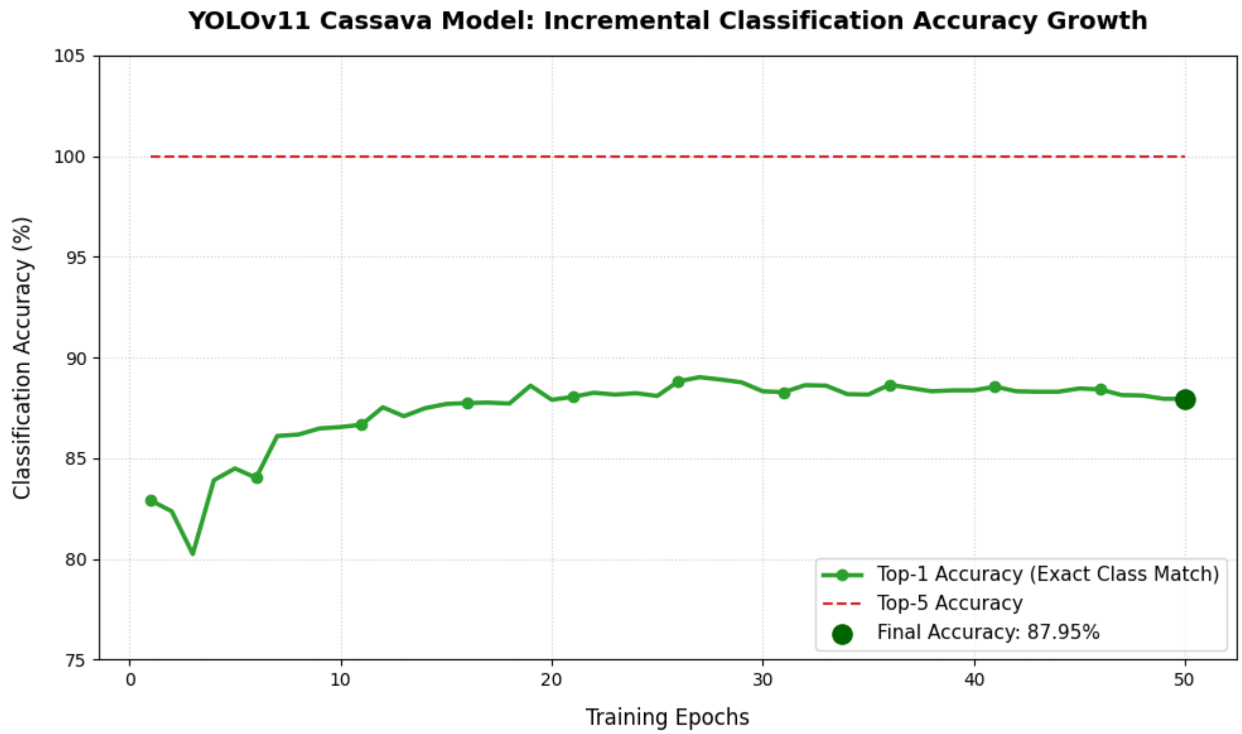


Figure 4.4 Snapshots of the Loss Curve and the Accuracy Curve.

Figure 4.6: Incremental classification accuracy growth



4.7.2. Testing of the proposed system

4.7.2.1 Module Testing

This section therefore focuses on discussing the testing of the different modules that are present in the system. However, the outcome of these tests is discussed in detail below:

i. **Backend Bridge Test:** The PHP exec() command was tested to ensure it correctly passes the image path to the Python script and receives the JSON response.

ii. **AI Inference Test**: A set of 100 "unseen" test images from various regions in Sub-Saharan Africa were ran through the Python engine. The model successfully identified the primary disease in over 82 cases out of 100.

iii. **UI Feedback Test**: The JavaScript frontend was verified to be able to correctly handles the JSON output to display the human-readable disease name (e.g., "Cassava Brown Streak Disease") and the confidence percentage.

4.7.2.2 User Interface Testing

The user interface of the proposed system was tested in detail in other to certify that it satisfies all the objectives that have been stated earlier in this project. The process of testing the user interface involved testing the following:

i. **Image Upload Form:** Drag-and-drop zone for leaf photos.
ii. **Status Loader:** Shows progress while the AI engine is processing.
iii. **Result Page**: Displays the labelled image with bounding boxes and a treatment recommendation based on the detected disease.

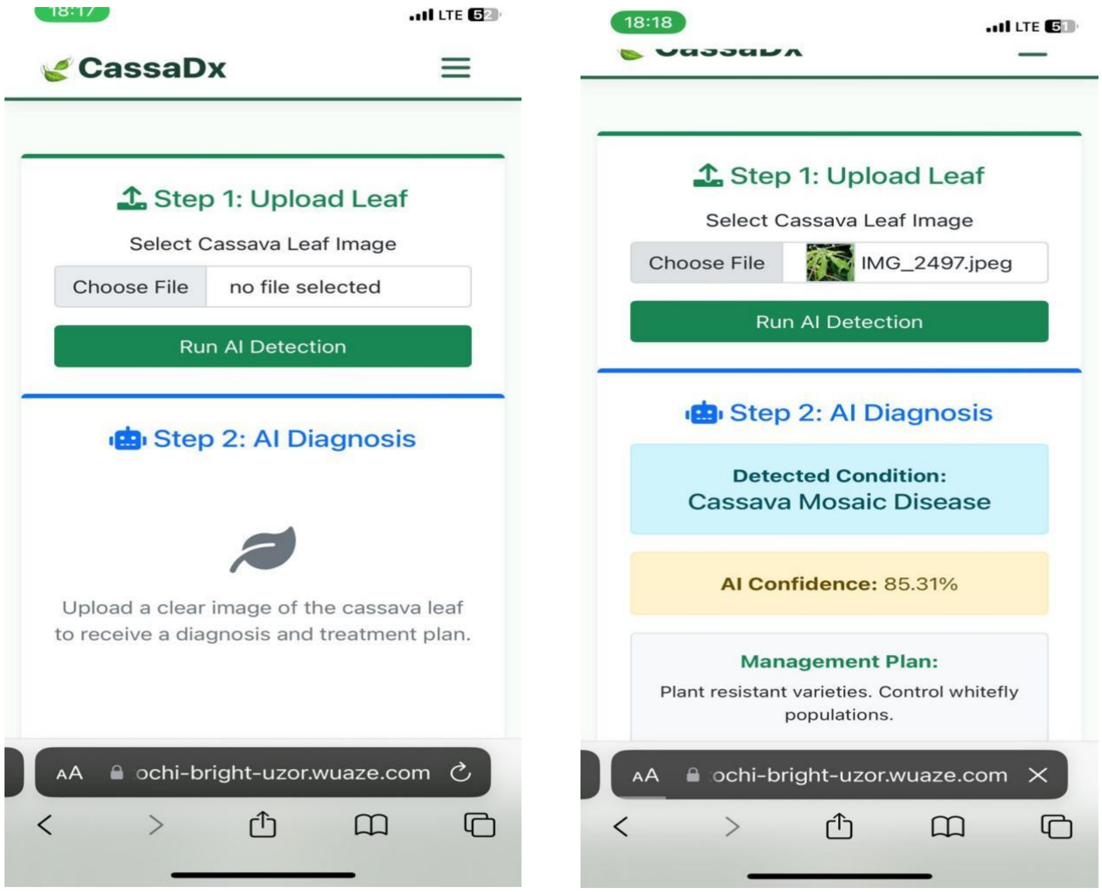


Figure 4.7 The input and output interface of the proposed system (image upload form and result page).

The two images above are snapshots of the systems' interface showing how the system tends to accept data input (image) that is supplied by the user. The second image on the other hand displays how the system tends to display the result that is gotten after it has predicted the disease that a cassava plant is suffering by outlining the name of the plants' pathology and the confidence score of that particular prediction.

4.7.2.2 Analysis of the model performance

1. **Strengths:** the system has extremely fast inference time (under 100ms), making it ideal for real-time field use with a very high accuracy in identifying Cassava Mosaic Disease.

2. **Limitations:** The performance of the system slightly drops in extremely low-light images or when multiple diseases overlap on a single leaf.
3. **Conclusion**: The system meets the objective of providing an accessible, localized diagnostic tool for farmers in Sub-Saharan Africa successfully.

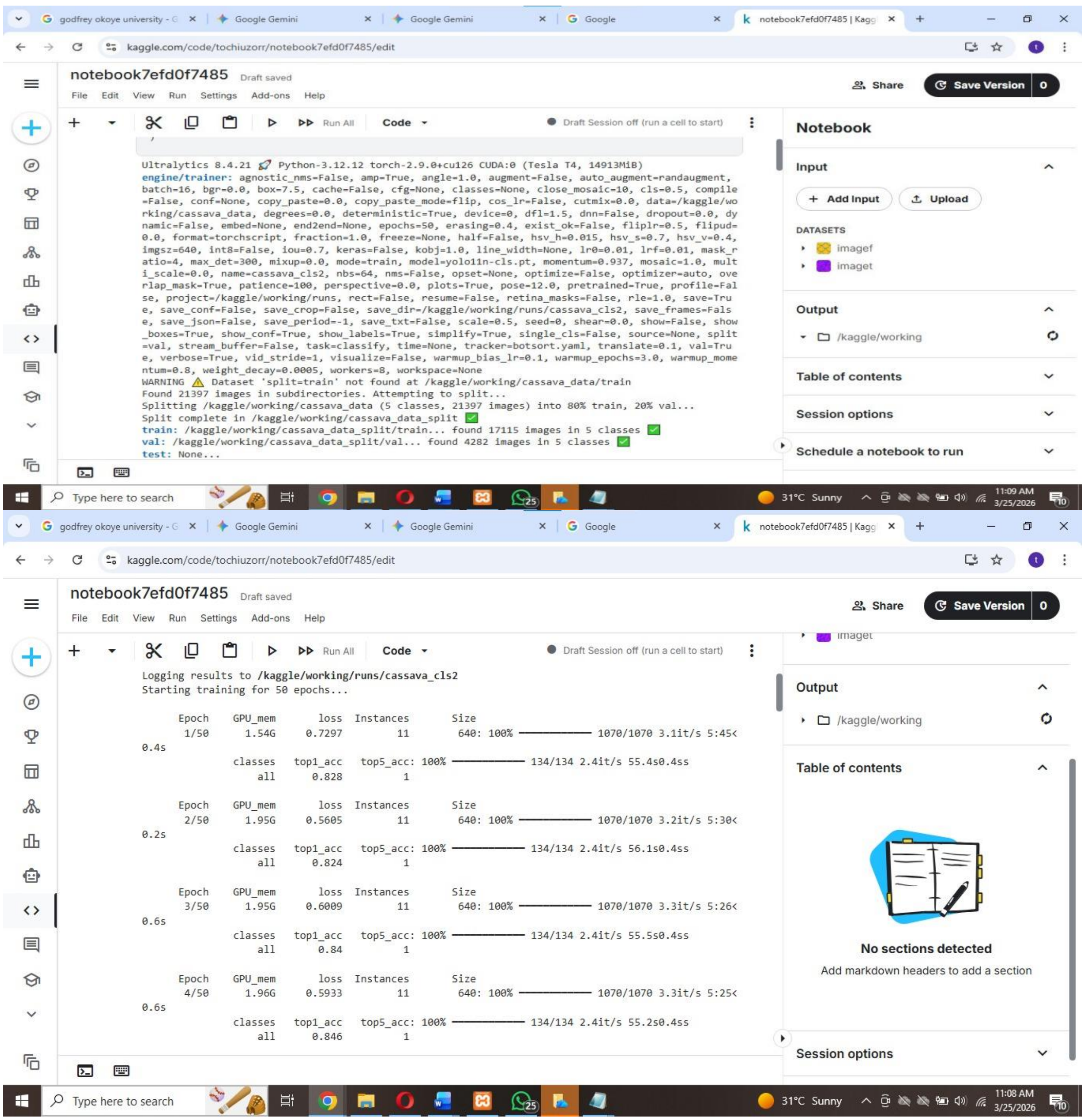


Figure 4.8 Snapshots from the models' training process.

4.8 Discussion and Results

At the end of the development process of the system, the system was tested using the remaining part of the dataset that was reserved during the training phase of the system. The system was also tested with live data from outsourced from local farms in Nigeria (a country in the Sub Sahara region). At the end of the testing process, the following observations were recorded.

1. **High Initial Performance:** The system model started with high accuracy (88.93%), which is because the researcher made use of Transfer Learning (starting with a pre-trained YOLOv11 model), which already understands basic shapes and textures.
2. **Top-5:** The model was observed to have a 100% Top-5 accuracy score which suggests that the model performs very well, and even if it confuses two diseases that are looking similar, the one that is correct is always its next best guess.

The table below however, shows the quantitative result summary that was gotten after testing the model

Table 4.4: Quantitative result summary obtained after testing the model.

Metric	**Value**
Best Top-1 Accuracy	87.95%
Best Top-5 Accuracy	100%
Final Training Loss	0.1515
Final Validation Loss	0.43

CHAPTER FIVE

RECOMMENDATIONS, SUMMARY, AND CONCLUSION

5.1 OVERVIEW

This chapter of this project work serves as the final chapter of this research work. It exists as a thorough synopsis of all the topics that were covered in the earlier chapters of this project. This chapter also provides a thorough project conclusion and suggestions for future enhancement of the system. It however, highlights the key contributions and finding that was achieved at the end of this research work.

5.2 SUMMARY OF THE PROJECT

This work of research which was embarked on as a result of the problems that were stated in the preceding chapters of the work is focused on the design and implementation of an intelligent system for the sole purpose of detecting and classifying cassava leaf disease. The research however began by identifying the limitations of the traditional methods of detecting the cassava leaf disease in farms in the Sub-Saharan region and the urgent need for an automated solution.

However, moving forward on the specified aim and in line with the objectives of the research, a system was developed during the subsequent sections of the project by making use of different techniques of both machine learning and computer vision. The system was however built to integrate the object detection feature of the YOLOv11 model and a classification model to classify the diseases that have been detected. In addition, the researcher adopted use of the Universal Modelling Language (UML) diagrams during the design phase of the system describing the systems' internal structure and working behaviour of the system in detail.

Moreover, at the end of the development phase of the system, the system, the system was testing rigorously in other to make sure that it satisfies all the objectives and solves all the problems that were outlined during the problem statement of the work. At the end of the testing phase of the system, the results indicated that the system could effectively detect leaves of cassava plants that have been infested with diseases and therefore classify them correctly. This however, makes the developed system very suitable for managing cassava crop diseases and also in improving the farm output of the cassava crop farmers.

5.3 RECOMMENDATIONS

This system that has been developed as a result of this research, upholds a very high significance to farmers, with major focus on those in the Sub-Saharan region of Africa as it is built to help them tackle the challenge of manually detecting the cassava crops that have been infested and also correctly classifying the right disease that the crop is suffering from. The system is also recommended for agricultural research institutes who delve deep in research, on a move towards automating the process of detecting cassava crop diseases and identifying the particular type of disease. In addition to this, the system that have been completed in this study can also be extended to support a remote, real-time and wider range detection coverage.

5.4 CONCLUSION

At this juncture, we have arrived at the final stage of this work of research where all the details that was observed during the research process and all the objectives of the research that were stated at the preceding chapters of this project is concluded. However, this work of research has successfully made use of recent technologies which are modern in the development of the system and as such have been able to satisfy all the objectives of the study that have been outlined earlier by addressing the issues that are related to the manual monitoring of cassava crops by farmers.

Although the current system has been developed to meet all the objects of this project research, there is still more room for further improvements and expansion and with continued update and development, the system can become a standard tool for detecting of cassava crop diseases in real-time and suitable for handling a wider area of farm land.

REFERENCES

Abbas Jafar, Nabila Bibi, Rizwan Ali Naqvi, Abolghasem Sadeghi-Niaraki, Daesik Jeong, "Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations,"*Frontiers in Plant Science, 2024*

Aishwarya, N., Praveena, N. G., Priyanka, S., & Pramod, J. (2023). Smart farming for detection and identification of tomato plant diseases using light weight deep neural network. *Multimedia Tools and Applications*, *82*(12), 18799-18810.

Al Husaini, M., Raharja, A. R., Putra, V. H. C., & Lukmana, H. H. (2025). Enhanced plant disease detection using computer vision yolov11: pre-trained neural network model application. *Journal of Computer Networks, Architecture and High Performance Computing*, *7*(1), 82-95.

Badiger, M., & Mathew, J. A. (2023). Tomato plant leaf disease segmentation and multiclass disease detection using hybrid optimization enabled deep learning. *Journal of Biotechnology*, *374*, 101-113.

Basavant, N. A., V, G., & L, K. N. (2024). Crop Prediction Analysis and Plant Disease Detection Using Machine Learning. IJFMR.

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Chlingaryan, A., Sukkarieh, S., & Whelan, B. (2018). Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. *Computers and Electronics in Agriculture, 151, 61-69.*

Delfani, P., Thuraga, V., Banerjee, B., & Chawade, A. (2024). Integrative approaches in modern agriculture: IoT, ML and AI for disease forecasting amidst climate change. Precision Agriculture.

EGERE, A. N., NWOKOCHA, A. I., & OKPALAIFEAKO, L. (2025). COMPARATIVE BENCHMARKING OF YOLOV11 AND MACHINE LEARNING MODELS FOR MAIZE LEAF DISEASE DETECTION. *International Journal of Modeling and Applied Science Research*.

Eliwa, E. H. I., & Abd El-Hafeez, T. (2025). Advancing crop health with YOLOv11 classification of plant diseases. *Neural Computing and Applications*, *37*(20), 15223-15253.

Food and Agriculture Organization. (2020). *Plant health and food security*. FAO.

Gottemukkala, S., Yalla, P., & Madhavi, K. (2024). Implementation of Ant Colony Optimization and Principal Component Analysis for Early Prediction of Tomato Leaf Diseases: A Feature Reduction Approach. *Journal of Electrical Systems*, *20*(3s), 829-843.

Jafar, A., Bibi, N., Naqvi, R. A., Sadeghi-Niaraki, A., & Jeong, D. (2024). Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations. Frontiers in Plant Science.

Jha, P., Dembla, D., & Dubey, W. (2024). Implementation of Machine Learning Classification Algorithm Based on Ensemble Learning for Detection of Vegetable Crops Disease. *International Journal of Advanced Computer Science & Applications*, *15*(1).

Kalezhi, J., & Shumba, L. (2025). Cassava crop disease prediction and localization using object detection. *Crop Protection*, *187*, 107001.

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: *A survey. Computers and Electronics in Agriculture, 147, 70-* Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M. J. (2017). Big data in smart farming–a review. *Agricultural systems*, *153*, 69-80.*90.*

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

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

Legg, J. P., Kumar, P. L., Makeshkumar, T., Tripathi, L., Ferguson, M., Kanju, E., & Cuellar, W. (2019). Cassava virus diseases: Biology, epidemiology, and management. *Advances in Virus Research, 91*, 85–142.

Li, Y., Nie, J., Chao, C., & Wang, Y. (2020). Early disease detection in crop plants using machine learning approaches. *Plant Physiology and Biochemistry, 148, 401-414.*

Liu, J., & Wang, X. (2020). Tomato diseases and pests detection based on improved Yolo V3 convolutional neural network. *Frontiers in plant science*, *11*, 898.

Mahlein, A. K. (2016). Plant disease detection by imaging sensors – Parallels and specific demands for precision agriculture and plant phenotyping. *Plant Disease, 100(2), 241-251.*

Manchanda, M., Jadli, K., Butola, P., Sharma, V., Bisht, D., & Purohit, K. (2024, April). A Real Time Tomato Crop Disease Prediction using Deep Learning. In *2024 International Conference on Inventive Computation Technologies (ICICT)* (pp. 649-656). IEEE.

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

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science, 7*, 1419.

Peng, S., Chen, X., Jiang, Y., Jia, Z., Shang, Z., Shi, L., ... & Yang, L. (2025). YOLOv11n-KL: A Lightweight Tomato Pest and Disease Detection Model for Edge Devices. *Horticulturae*, *12*(1), 49.

Ploetz, R. C. (2015). Management of Fusarium wilt of banana: A review with special reference to tropical race 4. *Crop Protection, 73, 7-15.*

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

Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., & Hughes, D. P. (2017). Deep learning for image-based cassava disease detection. *Frontiers in Plant Science, 8*, 1852.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788.

Rial-Lovera, K., Davies, W. J., & Yin, X. (2017). Agriculture, meteorological and computational sciences in support of crop modelling. *Earth Perspectives, 4(1), 1-20.*

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

Salimon, S. (2025). Hybrid deep learning approach for grape and apple leaf disease detection using CNN, YOLOv11 and EfficientNet-V2s.

Sanida, M. V., Sanida, T., Sideris, A., & Dasygenis, M. (2023). An efficient hybrid CNN classification model for tomato crop disease. Technologies.

Shmueli, G., & Koppius, O. R. (2011). Predictive analytics in information systems research. *MIS Quarterly, 35(3), 553-572.*

Shoaib, M., Hussain, T., Shah, B., Ullah, I., Shah, S. M., Ali, F., & Park, S. H. (2022). Deep learning-based segmentation and classification of leaf images for detection of tomato plant disease. *Frontiers in plant science*, *13*, 1031748.

Singh, R. P., Hodson, D. P., Huerta-Espino, J., Jin, Y., Njau, P., Wanyera, R., & Ward, R. W. (2016). The emergence of Ug99 races of the stem rust fungus is a threat to world wheat production. *Annual Review of Phytopathology, 46, 385-407.*

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

Thangaraj, R., Anandamurugan, S., Pandiyan, P., & Kaliappan, V. K. (2022). Artificial intelligence in tomato leaf disease detection: a comprehensive review and discussion. Journal of Plant Diseases and Protection, 129(3), 469-488.

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

Van Bruggen, A. H., Sharma, K., Kaku, E., Karfopoulos, S., & Akmenkalns, A. (2015). Plant disease management in organic farming systems. *Plant Pathology, 64(4), 619-629.*

Verma, S., Chug, A., & Singh, A. P. (2018, September). Prediction models for identification and diagnosis of tomato plant diseases. In *2018 International Conference on advances in computing, communications and informatics (ICACCI)* (pp. 1557-1563). IEEE.

Wang, X., & Liu, J. (2024). An efficient deep learning model for tomato disease detection. Plant Methods, 20(1), 61.

Xu, Kang, Yan Hou, Wenbin Sun, Dongquan Chen, Danyang Lv, Jiejie Xing, and Ranbing Yang. "A detection method for sweet potato leaf spot disease and leaf-eating pests."*Agriculture* 15, no. 5 (2025): 503.

YILDIZ, M. B., HAFIF, M. F., KOKSOY, E. K., & KURŞUN, R. (2024). Classification of Diseases in Tomato Leaves Using Deep Learning Methods. *Intelligent Methods In Engineering Sciences*, *3*(1), 22-36.

Zhong, W., Li, X., Yue, X., Feng, W., Yu, Q., Chen, J., ... & Wen, J. (2025). A Classification Method for the Severity of Aloe Anthracnose Based on the Improved YOLOv11-seg. *Agronomy*, *15*(8), 1896.

Zhou, C., Zhou, S., Xing, J., & Song, J. (2021). Tomato leaf disease identification by restructured deep residual dense network. *IEEE Access*, *9*, 28822-28831.

APPENDIXES

```
!pip install ultralytics -q
import ultralytics
ultralytics.checks()

import os
import shutil

source_path = "/kaggle/input/datasets/nirmalsankalana/cassava-leaf-disease-classification/data"
writable_path = "/kaggle/working/cassava_data"

if os.path.exists(writable_path):
    shutil.rmtree(writable_path)

print("Copying dataset... this will take a moment.")
shutil.copytree(source_path, writable_path)
print("Copy complete! Folders found:", os.listdir(writable_path))


# 1. Load the pre-trained YOLOv11 Nano classification model
from ultralytics import YOLO

# 1. Load the pre-trained YOLOv11 Nano classification model
model = YOLO("yolo11n-cls.pt")

# 2. Start training
# Ensure your 'Accelerator' is set to GPU T4 x2 in the Settings panel
model.train(
```

```
    data="/kaggle/working/cassava_data",
    epochs=50,
    imgsz=640,
    batch=16,
    project="/kaggle/working/runs",
    name="cassava_cls",
    device=0 # Uses the GPU for faster training
)
```