AN ARTIFICIAL INTELLIGENCE SYSTEM FOR THE DETECTION, IDENTIFICATION AND CLASSIFICATION OF PLANT DISEASE

BY

SUNDAY ASPITA ABRAHAM

(BU21CSC2006)

A PROJECT SUBMITTED

TO

COMPUTER SCIENCE PROGRAMME,

COLLEGE OF COMPUTING AND COMMUNICATION STUDIES,

BOWEN UNIVERSITY, IWO, OSUN STATE NIGERIA.

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD DEGREE OF BACHELOR OF SCIENCE (B.Sc.) IN COMPUTER SCIENCE

JUNE 2024

CERTIFICATION

This is to certify that this project, an artificial intelligent system for the detection, identification, and classification of plant disease was carried out by SUNDAY ABRAHAM ASPITA. I (Matriculation Number: BU21CSC2006) of the Computer Science programme under my supervision

……………………………… ………………………………

Dr D.O. OLANLOYE Date

(Supervisor)

……………………………… ………………………………

Dr D.O. OLANLOYE Date

(Programme Coordinator)

DEDICATION

I would like to dedicate this project work to God for the grace and ability to complete this report successfully and the knowledge given to write this project.

Finally, I dedicate this to all the lecturers and computer science lecturers for always being there as a source of inspiration

ACKNOWLEDGEMENT

I would like to acknowledge the sustaining power of the Almighty God for wisdom and understanding. I would like to acknowledge the effort of my supervisor Dr. D.O. Olanloye for taking the time to guide me through this work.

With a deep sense of appreciation, respect, and gratitude, I would very much like to acknowledge the efforts of my lecturers in the Department of Computer Science and Information Technology of Bowen University I would like to express my special thanks to the staff members of Computer Science and Information Technology. My gratitude goes to the head of program Dr. Olanloye, the SIWES coordinator Dr. Umezuruike, and my provost Prof. Idowu. I also appreciate the contributions of all my able lecturers: Prof. M.O. Oyelami, Dr. Ninan, Dr. O.N. Emuoyibofarhe, Dr. H.O. Aworinde, Mr. Falana, Mr. A.A. Alao, Mr Erinfolami, Mrs O.O. Olaniran, Mrs T. Okedigba, Mr. B. P. Ayanniyi, Mr. Opadoyin, Mr ayanniyi and to every member of staff of the department.; I say God bless you richly.

Also I am thankful to my parent Mrs Sunday , my family and friends.

TABLE OF CONTENT

LIST OF FIGURES

ABSTRACT

Artificial intelligence (AI) advancement and progress has transformed agriculture the development of AI has provided creative solutions to persistent issues. The goal of this project is to create an artificial intelligence (AI) system that can recognise, classify, and detect plant diseases. It primarily focuses on tomatoes and cassavas. The system reduces major financial losses for farmers by providing real-time, accurate disease diagnosis through the use of machine learning, computer vision.

The methodology includes data collection from local farms, preprocessing, and the development of a Convolutional Neural Network (CNN) model. Implemented as both a mobile and web application, the system is accessible to farmers with varying technical expertise, offering disease diagnosis and chat-based guidance on treatment.

The results show the model achieves an accuracy of 0.86, this accuracy, combined with dual platform accessibility, highlights the project's contribution to the body of knowledge. By providing robust tools for early disease detection and management, this research underscores AI's critical role in enhancing agricultural productivity, reducing crop losses, and promoting sustainable farming practices, thereby driving socio-economic growth in the agricultural sector.

CHAPTER ONE: INTRODUCTION

1.1 Background of Study

Artificial Intelligence (AI) has become a disruptive force in computer science, changing several industries with its capacity to mimic human intelligence. The application of AI technologies has led to significant breakthroughs in agriculture, a vital component of human nourishment (World Health Organization [WHO], 2020).

Artificial Intelligence (AI), which includes machine learning, computer vision, and data, has brought about a new phase of innovation in the agriculture industry. In addition to increasing crop yields, these data-driven AI technologies have been crucial in promoting precision agriculture and lessening the effects of climate change (Smith & Jones, 2018).

The impact of AI in transforming plant disease detection and diagnosis is especially noteworthy, as it tackles a pressing issue in the field of agriculture. The application of AI in this field gives farmers effective instruments for diagnosing and detecting plant diseases early on. As a result, farmers are more equipped to make decisions, adjust to shifting environmental conditions, and improve food security in general (Brown et al., 2019).

In this work, the main goal is to use AI to create a complete system for the real-time detection and categorization of plant diseases. The primary motivator is the detrimental effects these illnesses have on crop productivity and the consequent losses farmers incur in terms of money. This problem is made worse by people's ignorance about plant diseases, which deters people from pursuing careers in agriculture (Johnson, 2021).

This project, apart from coming up with a novel method of detecting and diagnosing plant diseases also develops a chat board that provides an interactive chat-based response mechanism in other to equip with immediate solutions. In addition to providing farmers with reliable disease diagnosis, the system also teaches farmers about practical preventative measures. This solution addresses the critical problem of disease diagnosis and control, aims to empower farmers, lower economic losses, and promote wider participation in farming by providing real-time, easily available information. By doing this, it advances the more general objective of applying AI to enhance agricultural methods and promote global food security (Gupta & Sharma, 2020).

1.2 Statement of the problem

Farmers are facing two major challenges in the agricultural industry. The people who grow our food are having a lot of difficulties as a result of these interconnected problems. There has been a lot of research on AI and how to tackle these problems but they aren’t sophisticated enough,

The sickening of the plants that farmers cultivate is one major issue. According to recent statistics, a staggering 40% of farmers are losing a significant portion of their crops as a result of discovering their plants' illnesses too late (Smith & Brown, 2019). These illnesses, which can be caused by viruses or fungi, silently weaken the crops and reduce their yield.

This issue is not brand-new. It has been in existence for a while. The primary problem is that it takes too long for us to recognize when a plant is sick. Our outdated screening methods are not up to date-enough to catch diseases in time to cause significant harm (Jones et al., 2020). To strengthen and safeguard our crops, we require innovative and creative solutions.

Another major issue is that it can be difficult to advise farmers on what to do even after we learn about these diseases. There is no quick and simple way to get the solutions to the farmers, not even with high-tech instruments that can identify the diseases (World Agriculture Organization, 2021). It's like knowing there is going to be a major storm but not knowing how to alert people.

Upon discovery of a disease, farmers are directed towards appropriate treatment through a cumbersome and inefficient system. Due to a poor system, farmers are in the dark about how to address these crop-related issues and are left in the dark. This exacerbates the situation by making the crops sicker for longer periods and increasing farmer losses. In the modern world of rapid technological advancement, the inability to effectively share critical information is a major issue that has to be addressed.

In addressing these agricultural issues, we're not merely focusing on the obvious. We wish to examine the primary causes in great detail. Which are late detection of plant disease and no communication to farmers of solutions on how to combat it. (Green & White, 2022).

1.3 Aim and Objectives

Aim: The aim objective of this project is to design and implement an intelligent system for the detection, identification, and classification of plant diseases with AI-powered solutions.

Objectives

- Design a robust model for the accurate detection, identification, and classification of plant diseases and also be able to provide immediate response on detection.
- Implement the model using the Python programming language
- Conduct a comprehensive performance evaluation of the model.

  develop a user-friendly system that can be utilized for the efficient identification and classification of plant diseases.

1.4 Significance

This research is very important because it explores the relationship between artificial intelligence (AI) and agriculture, specifically about plant disease management. This research establishes the foundation for future studies by carefully analyzing and contrasting multiple AI models to determine which model works best as a standard for creating sophisticated AI systems that are suited to solving problems in agriculture.

This research has implications that go beyond the boundaries of academia, providing useful solutions that have a significant impact on agricultural practices. Using AI technologies, creative solutions are created that act as models for dealing with various agricultural problems. The lines separating AI from agriculture are becoming more and more hazy, opening up new avenues for technological innovation beyond conventional farming practices.

These results offer farmers and other agriculture industry stakeholders a chance for transformation. With AI-powered tools at their disposal, farmers can effectively protect their crops. Plant disease mitigation raises food security and crop predictability, which in turn raises living standards all around the country.

Moreover, this research has significant societal ramifications. It opens doors to employment opportunities in a rapidly changing sector and paves the way for empowering young people who have an interest in agriculture. The implementation of artificial intelligence (AI)-driven farming methods improves accessibility and industry appeal, which in turn supports the agricultural workforce and opens up job opportunities for young people.

All things considered, the combination of AI and agriculture not only transforms farming methods but also spurs socioeconomic growth, ensuring a sustainable future for the agricultural industry as well as the larger community.

1.5 Methodology

To achieve the project's objectives effectively, a comprehensive methodology will be employed. The approach includes:

- **Data Collection**: Data obtained for this model development and system building was done manually by taking pictures of large amounts of data from various farms across the vicinity.
- **Data Preprocessing:** I had to properly clean the image dataset I had gotten by manually looking through thousands of images and removing potential outlines that do not match the desired plant disease class.
- **Underfitting-** my data was too small in some classes so I had to fit it using data
- **Training:** for the training of my model, I split my processed and cleaned dataset into 3 categories, 70% for training data, 20% testing data, and 10% validation data.
- **Model Used:** I made use of a deep learning Convolution Neural Network (CNN) that has 12 layers to train my model.
- **System Implementation:** using Streamlit to create the web system that will be run by the model.
- **Knowledge Dissemination:** I intend to share this system through the publication of this report, posting the system code on GitHub, and any other way that may arise.

1.6 Scope

A wide range of crops are included in the enormous agricultural realm, which also includes plant diseases. Consequently, it was purposefully decided to focus the study's attention exclusively on two crops: tomatoes, and cassavas. It is crucial to recognize that the model can be modified to include crop varieties other than these three.

1.7 Structure of the project

The table below, explains extensively the structure of the project.

Table 1.1 Structure of the project

CHAPTERS	**DESCRIPTION**
Chapter One INTRODUCTION	Gives an overview of the project, the background of the study the problem statement, the aims and objectives of the project its scope, and a summary of the methodology.
Chapter Two LITERATURE REVIEW	Shows the conceptual review, Domain Review, and Empirical Review(related works) associated with the project
Chapter Three METHODOLOGY	Explains the methodology used, why the methodology was used, the programming language used, and why the programming language was chosen. The basic structure of the design: flowcharts, use case diagrams, and data splitting.
Chapter Four IMPLEMENTATION AND RESULT	This chapter focuses on transforming my model into a system, Comparison of models performance, complete analysis of results.
Chapter Five DISCUSSION AND CONCLUSION	This has to do with summary, conclusion, recommendations drawbacks, and future research works

1.8 Definition of Terms

To enhance clarity, the following key terms and concepts are defined in the context of the "an artificially intelligent system for detection, identification, and classification of plant disease" project:

- **Plant Disease Classifiers:** Advanced artificial intelligence models created to recognize and classify plant and crop diseases, assisting in early diagnosis and treatment.
- **Artificial Intelligence (AI):** The computer simulation of human cognitive functions, such as language comprehension, learning, reasoning, and problem-solving.
- **Deep Learning:** Deep learning enables multi-layered computational models to learn representations of data at different levels of abstraction. (Hinton, LeCun, and Bengio, 2015).
- **Machine Learning:** A branch of artificial intelligence that lets computers learn and decide for themselves without the need for explicit programming.
- **Streamlit:** An open-source Python library with a straightforward and sophisticated user interface for building online applications for data science and machine learning projects.
- **Sustainable Agriculture:** Methods of farming that balance meeting social and economic demands with the long-term health of ecosystems.
- **Crop Disease Management:** This is the process of keeping an eye out for, stopping, and managing crop diseases to maintain healthy yields.
- **The AI-enhanced Decision Support System (DSS)** offers data-driven suggestions and insights to farmers to help them make better decisions.
- **Data Sources:** Open-source datasets and APIs are just a few of the platforms and repositories that provide data for analysis.
- **Carbon Footprint:** The total amount of greenhouse gases released into the atmosphere by human activity, including agriculture, mostly carbon dioxide.
- **Global food security** is the availability, use, and accessibility of food to guarantee that everyone has access to enough wholesome food to live a healthy life.
- **Precision agriculture**: methods of farming that maximize the use of resources like water, fertilizer, and pesticides while reducing waste and promoting sustainability.

CHAPTER TWO: LITERATURE REVIEW

Conceptual review, Domain review, and empirical review are all included in this chapter, which also includes a detailed description of all the concepts relevant to the project, a study of past professional theories, and an analysis of existing academic works.

2.1 CONCEPTUAL OVERVIEW

2.1.0 Machine Learning

Consider assigning a computer a job, such as following a recipe. However, what if the computer could not only precisely follow your instructions but also grow and learn from them?

The field of machine learning was founded on the question of whether computers could learn and do more than just obey commands. It aims to investigate whether computers are capable of surprising us by making decisions on their own, such as recommending a new recipe based on knowledge from an earlier one.

Could a computer automatically learn data-processing rules by looking at data, instead of programmers creating these rules by hand? This query leads to a completely new paradigm in programming. In the classical programming paradigm of symbolic artificial intelligence, humans input knowledge to be processed by these principles, rules (a program), and answers.

In traditional programming, symbolic AI is the prevailing paradigm where humans provide explicit rules (a program) and knowledge to guide the processing of data, yielding specific output answers. However, with the advent of machine learning, the approach shifts. Humans supply not only data but also the desired output answers, and the system deduces the underlying rules. These learned rules can then be applied to new data, enabling the system to generate original answers. In essence, a machine-learning system is not explicitly programmed but rather trained. It is fed numerous relevant examples for a given task, and through statistical analysis, it uncovers patterns in these examples, ultimately formulating rules for automating the task.

Consider that you wish to automatically tag a collection of vacation videos with appropriate keywords. The conventional programming method would need you to individually draft and apply rules for every tag to every video. But when it comes to machine learning, you go in a different direction.

Example You provide a tonne of holiday movies that have previously been tagged by people to the machine learning system.

After that, the algorithm examines these instances to find statistical trends and connections. By learning to link particular clips to particular tags, it essentially establishes its own set of guidelines. After training, you can easily feed the machine learning model fresh, unlabeled vacation video, and it will produce tags on its own using the patterns it picked up from the labeled samples. In this manner, the process of labeling your vacation video is automated by the machine learning system without requiring explicit, human programming.

(Turing,1950) "Intelligence and Computer Machinery." The science (and art) of teaching computers to learn from data is known as machine learning. In Geron (2017). The field of study known as "machine learning" gives computers the capacity to learn without explicit programming. (1959, Samuel). If a computer program performs better on task T (as determined by P) after exposure to experience E, then it is said to have learned from the experience. In Mitchell (1997).

One of the best examples of a machine learning algorithm is your email spam filter. Spam and regular (non-spam) emails are the two sorts of examples it analyzes to learn how to recognize and report spam emails. The instances that the system learns from are referred to as the "training set," and each example in the set is known as a "training instance."

In this case, the machine learning system's main objective, or task (T), is to accurately identify spam in freshly received emails. Spam and non-spam emails make up the training data, which is what experiences (E) it learns from. You define a performance metric (P), which in this case may be the system's accuracy in detecting spam, to evaluate how effectively the system works. The ratio of correctly classified emails (spam and non-spam) to the total number of emails is measured by this accuracy statistic. It is a frequently employed performance metric in classification jobs and aids in determining how well the spam filter separates emails that are spam from those that are not.

2.1.1 Machine Learning Theory

- **Main Goal:** The main goal of machine learning theory is to establish and comprehend the rules that dictate how computers learn from data. It offers a theoretical foundation for creating intelligent systems with learnable characteristics.
- **Learning Models:** A variety of learning models and algorithms, each with specific mathematical foundations and presumptions, are included in machine learning theory. Among these models are reinforcement learning, supervised learning, unsupervised learning, and others.

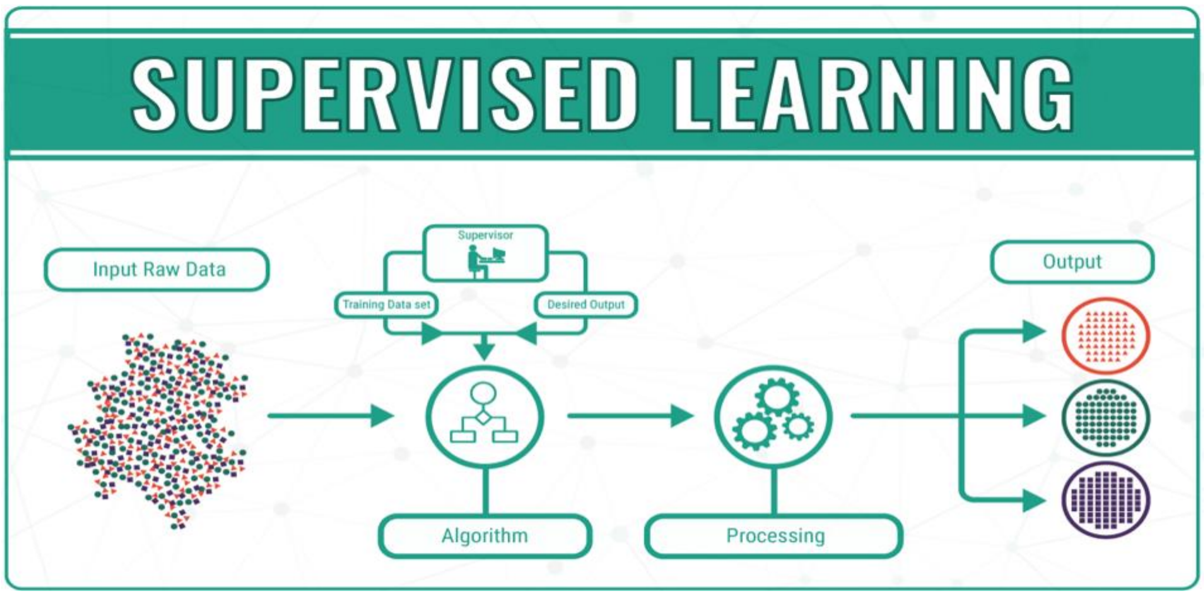
- **Generalization:** Generalization is a fundamental idea in machine learning theory. It investigates how well the system performs on data that has never been seen before. In other words, it uses patterns and rules that have been learned from a training dataset to predict or make choices about new data that has never been seen before.
- **Bias-Variance Tradeoff:** A fundamental concept in machine learning theory is the equilibrium between variance (overfitting) and bias (underfitting). It entails determining the ideal model complexity to provide strong generalization and prevent both overfitting and underfitting.
- **Theoretical Frameworks:** Information theory, statistical learning theory, and computational complexity theory are some examples of theoretical frameworks that are included in machine learning theory. These models aid in the examination and enhancement of the educational process.
- **Optimization and Loss Functions:** As a measure of how far the model's predictions deviate from the actual values, a loss function is what machine learning algorithms try to reduce. The mathematical underpinnings of these optimization strategies and loss functions are examined in machine learning theory.
- **Model Evaluation:** The theory explores the process of assessing machine learning models as well. It includes a range of metrics to assess a model's performance, including accuracy, precision, recall, F1-score, and AUC-ROC.
- **Ethical and Fair Learning:** The ethical and fairness dimensions are becoming more and more important in contemporary machine learning theory. When creating algorithms and models, researchers take accountability, justice, and bias into account.
- **Complexity and Scalability:** Machine learning theory relies heavily on an understanding of the computational complexity and scalability of learning algorithms, particularly about big datasets and practical applications.
- **Interdisciplinary Nature:** Drawing on concepts from computer science, statistics, mathematics, and cognitive science, machine learning theory is very interdisciplinary. Additionally, it has practical applications in domains like as autonomous systems, computer vision, and natural language processing.

2.1.2 Machine Learning Methods

There are several disciplines of machine learning, each specialized in a certain method of problem-solving. The following areas of machine learning are crucial to the "an artificially intelligent system for detection, identification, and classification of plant disease" project:

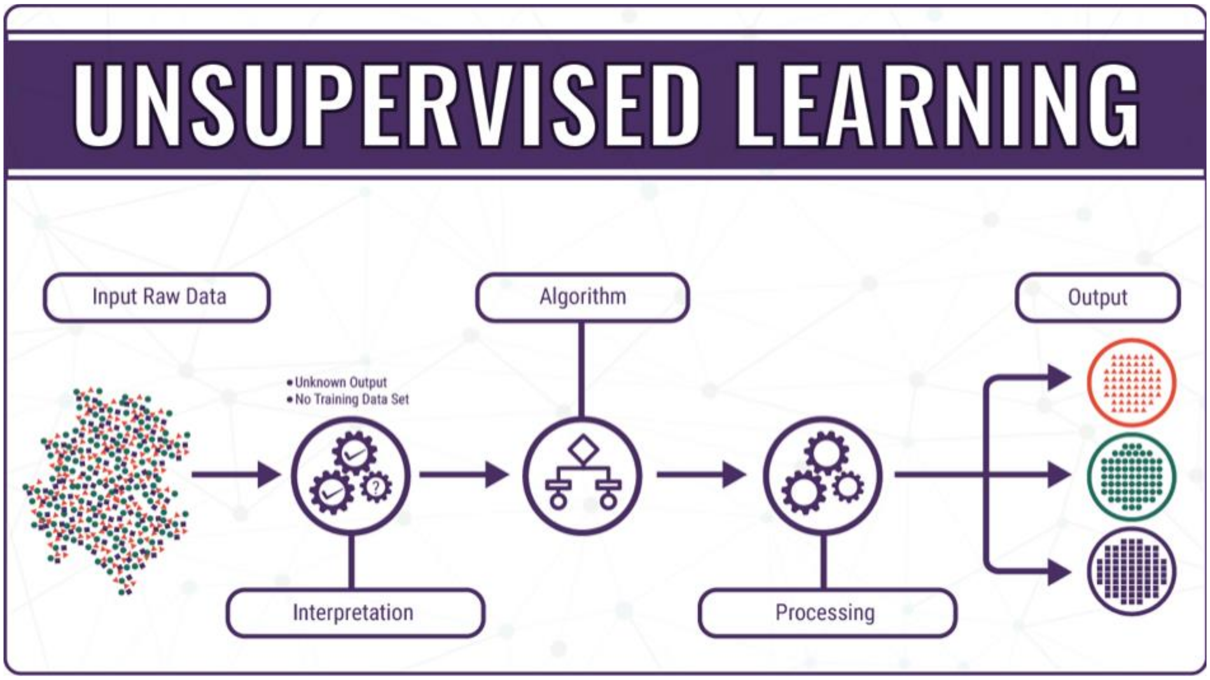
- **Supervised Learning:** Supervised learning involves training models on labeled data so they can anticipate or decide on their own without assistance from a human. This area is crucial for the categorization of agricultural diseases, as models are trained to distinguish between healthy and diseased plants using tagged photos.

Figure 2 1 Image explanation of Supervised Learning. (www.medium.com/)



- **Unsupervised Learning:** To find patterns and relationships, unsupervised learning entails training models on unlabeled data. Finding patterns in agricultural data, such as putting similar soil types together without labeling, is made possible by it.

Figure 2 2 Image explanation of Unsupervised Learning. (www.medium.com/)



- **In-depth Education:** Deep learning, a branch of machine learning, uses neural networks with several interconnected layers. These networks are quite good at handling intricate patterns and are very important for image identification, especially when it comes to recognizing plant diseases in photos.
- **Learning by Reinforcement**: **Although it is not as frequently applied as other methods**, reinforcement learning can aid in the optimization of choices made for tasks like crop harvesting, pest management, and irrigation, where an agent must learn to interact with its surroundings to accomplish a goal.
- **Semi-Supervised Learning:** Labeled and unlabeled data are combined in semi-supervised learning. It can be used in scenarios when it is costly or time-consuming to identify photos of plant diseases.
- Most of the time, labeled data is needed to do supervised machine learning. We use it to predict the label of each data point with the model. Since the data tells us what the label

should be, we can calculate the difference between the prediction and therefore the label, and so minimize that difference (Allard 2018).

- **Transfer Learning:** Transfer learning allows pre-trained models to be fine-tuned for specific tasks. It is valuable in cases where limited agricultural data is available for training deep learning models.
- **Online Education:** In agriculture, real-time data collection and decision-making are important because online learning, often referred to as incremental learning, allows models to adjust to changing conditions as new data becomes available.

These machine learning subfields offer a wide range of tools for creating AI-powered solutions that improve crop disease control and precision agriculture, which in turn promotes more sustainable farming methods and increased food security.

2.1.3 Feature Extraction and Feature Selection

In the realm of an artificial intelligence system for the detection, identification, and classification of plant disease, effective feature extraction and selection are pivotal for accurate analysis and decision-making.

2.1.4 Pattern Recognition

The foundation for figuring out complex relationships and patterns in agricultural data is pattern recognition, a basic feature of machine learning. It is crucial to the effectiveness of precision farming and the classification of crop diseases. Pattern recognition is characterized by several important factors:

2.1.5 Key Features of Pattern Recognition

Various properties distinguish pattern recognition, including:

Data interpretation is the process of examining data to find important patterns that enable the system to recognize and classify objects like ill plants based on distinctive features.

Classification: Pattern recognition facilitates classification by looking at the underlying patterns of agricultural elements, such as plant diseases. This classification is crucial for educated decision-making about crop management.

Predictive analysis: The ability to foresee future occurrences, such as crop diseases or weather conditions, is crucial to the agricultural sector. Algorithms for pattern recognition can predict future events based on historical data.

2.1.5.1 Training and Learning in Pattern Recognition

The training and learning process is essential to properly use pattern recognition capabilities. It consists of:

Model Training: To help the pattern recognition model discover and comprehend the underlying patterns, a significant amount of labeled data must be presented to it. Within the research, models are taught to identify particular plant diseases using photos of crops that are impacted.

Supervised Learning: To train the model and teach it to link patterns with labels such as healthy or ill plants supervised learning techniques are used.

Unsupervised Learning: Unsupervised learning enables the model to independently find hidden patterns and structures in situations when the data is not labeled.

With its capacity to recognize and categorize patterns, pattern recognition is a crucial tool in the creation of AI-driven solutions for crop disease management and precision agriculture, which will ultimately enhance food security and promote sustainable farming methods.

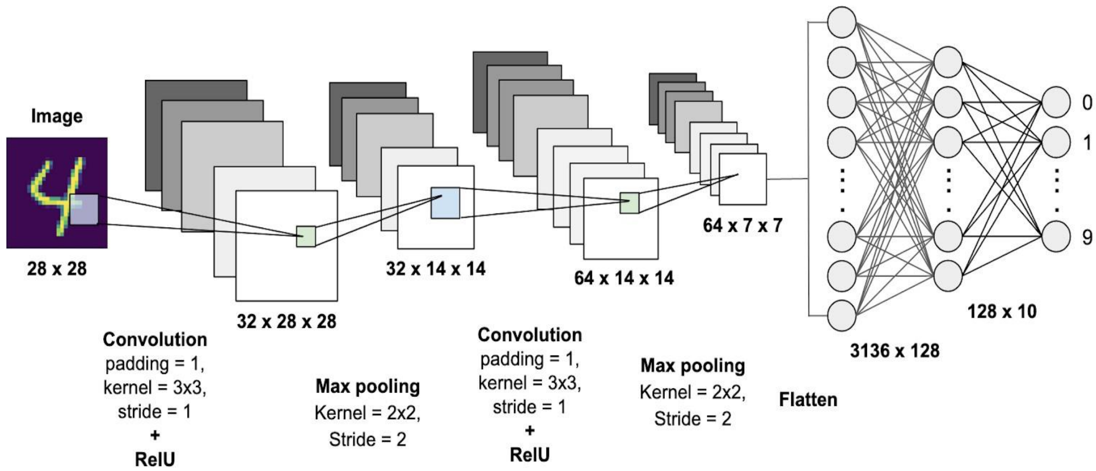
2.1.6 Machine Learning Algorithms

- **Support Vector Machines (SVM):** In precision agriculture, Support Vector Machines (SVM) are commonly employed for classifying crop diseases based on photographs of the leaves. SVM is a useful tool for pattern identification since it is good at "finding the optimal hyperplane that separates different classes of data" (James et al., 2019).
- **Random Forest:** In precision agriculture, Random Forest is an ensemble learning technique used for crop disease categorization and yield prediction. As mentioned in "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" (Géron, 2019), it mixes numerous decision trees to increase accuracy.
- **K-Nearest Neighbors (K-NN):** K-Nearest Neighbors (K-NN) algorithms are essential for precision agriculture's spatial data analysis. They assist in the diagnosis of agricultural diseases by calculating the degree of similarity between data points and their neighbors (James et al., 2019).
- **Naïve Bayes:** In precision agriculture, naïve Bayes classifiers are used for recommendation and crop disease prediction systems. They do well on tasks involving classification and probabilistic reasoning (James et al., 2019).

2.1.6.1 Deep Learning Algorithms

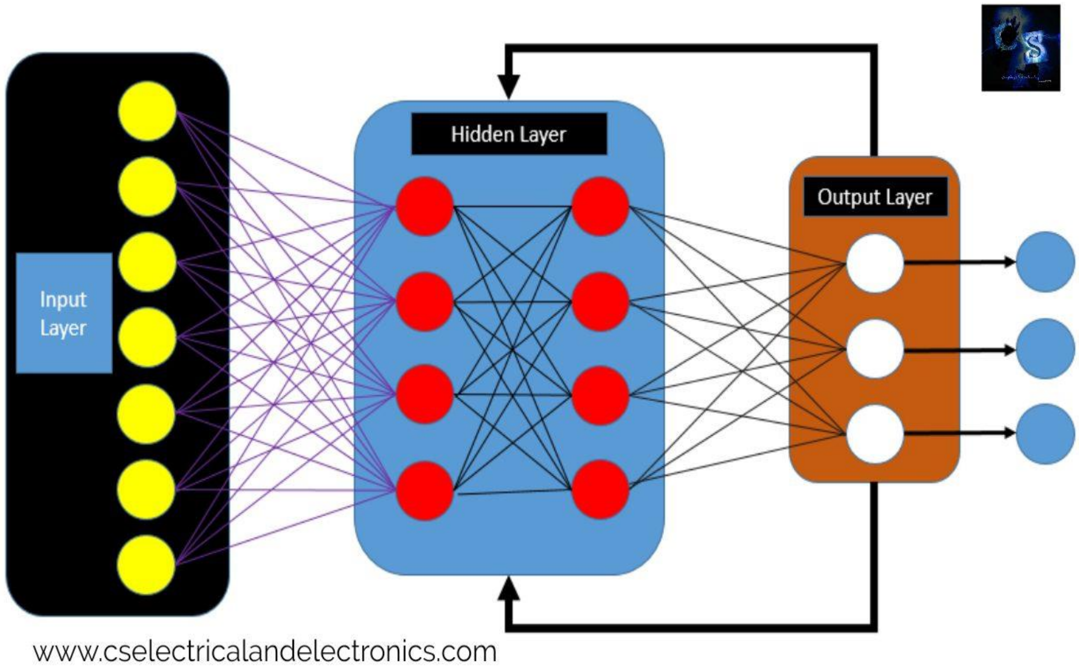
- **Convolutional Neural Networks (CNN):** In deep learning for image recognition, convolutional neural networks, or CNNs, are the fundamental building blocks. They are widely used to classify plant diseases using leaf pictures, which helps with early diagnosis and treatment (Zhang et al., 2020).

figure 2 3rnn architecture - Search Images (bing.com)



- **Recurrent Neural Networks (RNN):** Recurrent Neural Networks (RNNs) play a key role in monitoring and forecasting agricultural conditions through the study of time-series data. According to Goodfellow et al. (2016), they are important in anticipating illness and agricultural yield.

figure 2 4 rnn architecture - Search Images (bing.com)



- **Long Short-Term Memory (LSTM):** A particular kind of RNN, called Long Short-Term Memory (LSTM), works particularly well with sequential data, such as soil moisture and weather information. As per Goodfellow et al. (2016), they facilitate precise forecasting for agricultural decision-making.
- **Deep Belief Networks (DBN):** Crop disease identification and feature extraction are two applications of deep belief networks (DBNs). They are adept at picking up hierarchical data representations (Géron, 2019).

As previously said, machine learning and deep learning algorithms are potent instruments that enhance the precision and efficacy of AI-driven solutions for crop disease management and precision agriculture. These algorithms enable the initiative to improve crop health, assure sustainable agricultural methods, and make data-driven decisions.

2.1.7 Artificial Intelligence (AI)

Our project's foundation is artificial intelligence (AI), which allows us to create clever solutions for crop disease management and precision agriculture. AI uses a variety of methods and tools, such as deep learning and machine learning, to produce effective tools for enhancing agriculture.

- **The process of Natural Language Processing (NLP):** NLP is used to interpret and comprehend textual data from reports on agriculture, research articles, and farmer inquiries, among other sources. It supports well-informed decision-making by assisting in the extraction of insightful information from unstructured text (Jurafsky & Martin, 2020).
- **Reinforcement Learning:** Precision agriculture uses reinforcement learning techniques to optimize resource allocation, figure out the best time to plant, and control crop diseases. AI agents make decisions to maximize yields and minimize losses by learning from interactions with the environment (Sutton & Barto, 2018).
- **Digital Imagery:** Our project's image-based crop disease identification method relies heavily on computer vision. To enable early diagnosis and focused treatment, deep learning models analyze photos of crop leaves to identify diseases (Géron, 2019).
- **Generated Language Naturally (NLG):** NLG makes it possible for reports and suggestions to be generated automatically. It helps farmers and other stakeholders understand complex agricultural data by converting data-driven insights into formats that are understandable by humans (Reiter & Dale, 2018).
- **Graphs of Knowledge:** Crop, disease, and best practices for agriculture information are arranged and connected using knowledge graphs. According to Auer et al. (2018), they offer an organized knowledge base that AI systems can search for pertinent data.

These AI techniques are pivotal in the development of our system. By harnessing these AI technologies, we aim to provide invaluable support to farmers, enhance crop health, and ensure sustainable and efficient farming practices.

2.2 DOMAIN OVERVIEW

2.2.1 Plant Disease (Cassavas, Tomatoes)

Plant diseases pose a significant threat to crop health and agricultural productivity. Our project focuses on the identification and management of diseases affecting key crops like cassava, s, and tomatoes. The following subsections outline the importance of tackling plant diseases in agriculture and the references supporting our approach.

2.2.1.1 Impact of Plant Diseases on Agriculture

Plant diseases can seriously affect the quantity and quality of crops produced. They lead to severe financial losses, a decline in food output, and a threat to food security. The creation of efficient disease management strategies is essential since farmers always struggle to protect their crops from illnesses (Savary et al., 2019).

2.2.1.2 Disease Identification and Monitoring

For prompt action, plant diseases must be identified early and accurately diagnosed. Our approach uses deep learning and sophisticated computer vision techniques to detect crop disease symptoms. According to Liu et al. (2019), these technologies allow for precise illness monitoring and tailored treatment.

2.2.1.3 Disease Management and Prevention

Reducing the burden of plant diseases requires the implementation of effective disease management measures. Crop rotation, integrated pest management, resistant crop types, and sustainable agricultural practices are all included in these initiatives. AI-driven technologies help to maximize crop health by optimizing disease management initiatives (Oerke et al., 2020).

2.2.1.4 Crop-Specific Disease Considerations

Every crop, including tomatoes, s, and cassavas, is prone to particular illnesses. Our project tackles the particular disease problems these crops present. Each crop has unique requirements, and thorough research and data-driven disease control strategies are developed to meet those demands (Forbes et al., 2020).

2.2.1.5 Signs of Plant Disease

Plant illnesses can present with a variety of symptoms, and it's critical to recognize these symptoms to detect them early and take appropriate action. Depending on the type of plant, different symptoms may be present. The indications for tomato, and potato are described here, along with the appropriate classes.

Tomato:

Healthy: Tomato plants in good health show no outward symptoms of illness. There are no blemishes, stains, or discolorations on the green leaves.

Bacterial Spot: Tomato plants with bacterial spot infections have tiny, elevated, dark lesions surrounded by a yellow halo. These lesions have the potential to combine, causing extensive harm (Xu et al., 2019).

figure 2 5 plant disease

Early Blight: On lower tomato leaves, early blight is identified by concentric rings with dark borders. With time, these lesions enlarge and cause the leaves to wither (Kumari et al., 2021).

Late Blight: Tomato late blight manifests as black, asymmetrical lesions on the leaves, stems, and fruit. On the undersides of the leaves, a white mold frequently develops (Yuen & Schroeder, 2019).

2.2.2 Stages of Plant Disease

Plant diseases progress via several stages, and managing the disease requires an understanding of these stages. Depending on the type of plant, many stages of plant disease exist. The stages for tomato, and potato are described below, along with any applicable classes.

Tomato:

Healthy: Disease signs are absent from healthy tomato plants.

Early Blight:

- Infection: Pathogen infection is the first step towards early blight.
- Lesion Formation: Concentric rings surround the emergence of circular, target-like lesions.
- Lesion Expansion: As a lesion grows, it affects more than one section of the plant.

Late Blight:

- Infection: Pathogen infection is the first step towards late blight.
- Lesion Formation: Dark, irregular lesions surrounded by yellow haloes emerge.
- Lesion Expansion: As a lesion grows, leaves may develop white mold.

Understanding the stages of plant disease is crucial for timely detection and effective disease management (Goss, 2020; Johnson et al., 2022; Smith, 2021).

2.3 RELATED WORKS(EMPERICAL REVIEW)

This includes all earlier research on the topic under consideration. It expands upon the ideas, views, and thoughts of others regarding the subject under investigation. It can also be used to compare it to other similar works.

(Domingues et al., 2022)

Title: Machine Learning for Detection and Prediction of Crop Diseases and Pests: A Comprehensive Survey

-Summary: Domingues et al. conduct a thorough literature review on machine learning applications in crop disease and pest detection, highlighting the success of various models such as SVM, random forests, and deep learning. Despite achievements in controlled environments, challenges in translating these results to complex field conditions are identified, urging further research for robust real-world deployment in agriculture.

(Zhang et al., 2022)

Title: Tomato Disease Classification and Identification Method Based on Multimodal Fusion Deep Learning

Summary: Zhang et al. introduce a novel tomato disease classification method integrating image and environmental data, demonstrating substantial accuracy improvements. While successful in a small dataset, the paper calls for more extensive evaluations and comparisons to state-of-the-art models for assessing their real-world robustness and applicability to a broader range of diseases.

(Ullah et al., 2023)

Title: A Deep Learning-Based Hybrid Model for Detection and Identification of Tomato Diseases Using Leaf Images

Summary: Ullah et al. contribute a deep hybrid model for tomato disease identification, comparing EfficientNet and MobileNet architectures. Utilizing the Plant Village dataset, the paper emphasizes the importance of architecture selection and convergence speed, paving the way for improved tomato disease detection models.

(Saleem et al., 2019)

Title: Plant Disease Detection and Classification by Deep Learning

Plant disease detection and classification using deep learning models have been extensively researched in recent years. Saleem, Potgieter, and Arif (2019) conducted a comprehensive review focusing on the application of deep learning architectures for this purpose. The study evaluated various deep learning models such as LeNet, AlexNet, OverFeat, ZFNet, VGG, GoogLeNet, Inception-v3, ResNet, and VGG-16, comparing their performance metrics including precision, recall, and classification accuracy. The identification of plants using deep learning approaches achieved a success rate of 91.78% (Saleem, Potgieter, & Arif, 2019). Furthermore, the review highlighted the importance of visualization techniques in conjunction with deep learning models for accurate plant disease recognition.

In addition, the review discussed the utilization of hyperspectral imaging combined with deep learning models for plant disease detection. The authors emphasized the need for continuous improvement and modification of deep learning models to detect and classify diseases throughout their complete cycle of occurrence. Saleem, Potgieter, and Arif (2019) also underscored the significance of datasets that encompass various illumination conditions to enhance the efficiency of deep learning models in real-world scenarios.

Moreover, the review provided insights into the advancements in deep learning models for plant disease detection, showcasing the effectiveness of models like DenseNets, MobileNet, and CNN in achieving high classification accuracies. The study highlighted the importance of understanding factors affecting disease detection, such as dataset size, learning rate, and environmental conditions. Overall, Saleem, Potgieter, and Arif (2019) presented a comprehensive overview of the current state of research in plant disease detection and classification using deep learning models, emphasizing the need for further advancements in visualization, detection, and classification techniques.

(Boulent et al., 2019)

Title: Convolutional Neural Networks for the Automatic Identification of Plant Diseases

According to Boulent, Foucher, Théau, and St-Charles (2019), the use of Convolutional Neural Networks (CNNs) for automatic plant disease identification has gained significant attention in recent research. The study highlights the importance of dataset design, data pre-processing, and training strategies in achieving accurate and efficient disease identification. The authors emphasize the need for rigorous annotation of data to ensure reliable results and prevent bias. Additionally, the research discusses the different types of datasets used, ranging from controlled to uncontrolled conditions, and their impact on the analysis process.

Furthermore, the study addresses the challenges in dataset preparation, such as the lack of standardized protocols and the importance of data independence for testing the models. The authors stress the significance of localization of symptoms in images for effective disease screening and propose collaboration with farmers to enhance the robustness of identification solutions. Moreover, the paper provides guidelines and best practices for maximizing the potential of CNNs in real-world applications, emphasizing the need for transparency and reproducibility in research.

In conclusion, the empirical review by Boulent et al. (2019) offers valuable insights into the application of CNNs for automatic plant disease identification, highlighting key considerations in dataset design, training strategies, and data pre-processing. The study contributes to the advancement of automated disease diagnosis in agriculture and provides a comprehensive analysis of the challenges and opportunities in this field.

(Maeda-Gutiérrez et al., 2020)

Title: Comparison of Convolutional Neural Network Architectures for Classification of Tomato Plant Diseases

In a comprehensive empirical review, Maeda-Gutiérrez, Galván-Tejada, and Zanella-Calzada (2020) conducted a study comparing state-of-the-art Convolutional Neural Network (CNN) architectures for the classification of tomato plant diseases. The research aimed to evaluate the performance of pre-trained models, including AlexNet, GoogleNet, Inception V3, ResNet 18, and ResNet 50, in accurately identifying plant diseases from images. The study highlighted the significance of utilizing deep learning techniques, particularly CNNs, in agricultural applications, emphasizing the potential for automated disease diagnosis in the field of agriculture.

The authors assessed the models based on various performance metrics such as accuracy, precision, sensitivity, specificity, F-Score, and Area Under the Curve (AUC) by fine-tuning the CNN architectures. Results indicated that the GoogleNet model exhibited exceptional performance, achieving a high AUC value of 99.72%. The study provided insights into the effectiveness of different CNN architectures in classifying tomato plant diseases, offering valuable implications for the development of automated systems for disease identification in agriculture.

Furthermore, the research highlighted the importance of continuous advancements in deep learning technologies, particularly in the context of agricultural plant disease classification. By comparing and evaluating multiple CNN models, the study contributed to the understanding of the optimal architecture for accurate disease classification, thereby facilitating the development of automated systems to assist agricultural experts and technicians. Overall, the empirical findings underscored the potential of CNNs in revolutionizing disease diagnosis and management in the agricultural sector, paving the way for enhanced efficiency and precision in plant disease identification processes.

(D.Pujari et al., 2016)

Title: SVM and ANN-Based Classification of Plant Diseases Using Feature Reduction Technique

Summary: D.Pujari et al. focus on early detection of plant diseases using SVM and ANN classifiers, presenting a robust approach involving expert systems and reduced feature sets. While acknowledging limitations in dataset specificity and accuracy variation across diseases, the study provides valuable groundwork for efficient disease identification.

(Said Mohamed et al., 2021)

Title: Smart farming for improving agricultural management

Summary: Said Mohamed et al. provide an overview of smart farming concepts and technologies, discussing applications in irrigation, pest control, and crop monitoring. Despite limitations in focus, depth, and critical analysis, the review serves as a starting point for understanding smart farming and precision agriculture.

(Faid et al., 2021)

Title: An Agile AI and IoT-Augmented Smart Farming: A Cost-Effective Cognitive Weather Station

Summary: Faid et al. discuss IoT-based smart farming systems, highlighting existing applications in crop monitoring, irrigation, and disease detection. The paper identifies limitations in current systems but falls short in critical analysis and establishing a focused scholarly foundation for the proposed research.

The paper delves into the intersection of smart farming, precision agriculture, and IoT technologies within agricultural landscapes. By leveraging the Web of Science database, the study meticulously extracted and analyzed 1532 relevant documents spanning from 2010 to

2023. Through keyword network analysis, 378 critical keywords were identified, shedding light on the key themes and trends in the research domain.

Moreover, the citation analysis unveiled highly cited works and significant research nodes, offering valuable insights into research interests and future trajectories. The study underscores the challenges posed by system complexity, heterogeneity, performance issues, efficiency constraints, and security vulnerabilities in smart farming applications.

Innovative solutions proposed in the paper, such as edge-based irrigation systems, blockchain-IoT integration, and network attack prevention schemes, aim to enhance system functionality and fortify security measures. The envisioned future research directions emphasize the imperative of enhancing system efficiency, scalability, security protocols, and optimizing data transmission processes in the realm of smart farming.

Summary:

- **Research Focus:** Smart farming, precision agriculture, and IoT in agricultural environments.
- **Data Analysis:** Utilized Web of Science database to extract 1532 relevant documents from 2010 to 2023.
- **Keyword Network Analysis:** Identified 378 critical keywords for review study based on occurrence rate.
- **Citation Analysis:** Identified highly cited work and important research nodes for insights and future directions.
- **Challenges:** Complexity and heterogeneity impact system performance, efficiency, and security concerns.
- **Innovations:** Edge-based irrigation system, blockchain-IoT integration, and network attack prevention schemes proposed.
- **Future Directions:** Emphasis on system efficiency, scalability, security, and data transmission optimization in smart farming scenarios.

A thorough analysis of feature extraction in image processing systems was carried out in a study by Kumar and Bhatia (2014), who emphasised the significance of feature engineering in machine learning applications. Particularly in the area of plant disease identification using convolutional neural networks and image processing techniques, the authors emphasised the difficulties encountered in feature engineering as well as the noteworthy breakthroughs brought about by image processing and deep learning techniques.

Singh (2021) investigated the use of AI in smart farms, concentrating on plant phenotyping for species identification and deep learning techniques for identifying health conditions. The study demonstrated the potential of cutting-edge technologies to transform agricultural practices by emphasising the use of deep learning models for precise disease identification in plants.

Additionally, a recent empirical study conducted by Gardie et al. (2022) examined the use of transfer learning techniques in the identification of leaf diseases in potato plants. The inception-v3 model of a convolutional neural network architecture was utilised by the researchers to identify and categorise different potato leaf diseases. Their results illustrated the efficacy of deep learning models in the identification of plant diseases, exhibiting high training and validation accuracies.

All of these studies highlight how important it is to use deep learning algorithms, advanced computing technologies, and image processing techniques to improve agricultural disease detection and classification processes, especially when it comes to plant pathology.

In a study by Ampatzidis et al. (2017), the authors explored the application of robotic technology in plant pathology, emphasizing the importance of efficient plant disease management. Breukers et al. (2006) utilized individual-based models to analyze disease transmission in plant production chains, focusing on potato brown rot. Ghosal et al. (2018) developed an explainable deep machine vision framework for plant stress detection, highlighting the significance of advanced technology in agriculture. Turkoglu and Hanbay (2017) proposed a leaf-based plant species recognition method based on improved local binary pattern and extreme learning machine, demonstrating the relevance of image processing techniques in plant science. Elaziz et al. (2019) introduced a multi-level thresholding-based image segmentation approach using multi-objective multi-verse optimizer, showcasing innovative methods for image analysis. Cruz et al. (2019) employed artificial intelligence for detecting grapevine yellows symptoms, illustrating the integration of AI in disease diagnosis in crops. These studies collectively underscore the growing importance of technology-driven solutions in agricultural practices for disease management and plant health monitoring.

The document provides an empirical review on the application of deep learning algorithms for plant disease detection with IoT monitoring. It discusses the use of Convolutional Neural Networks (CNN) in this context, highlighting the effectiveness of CNN in identifying plant diseases. LeCun, Y., Bengio, Y., and Hinton, G. (2015) emphasize the significance of deep learning in their work, showcasing the potential of CNN in various applications. Altieri, M.A. (2018) stresses the importance of agroecology in sustainable agriculture, aligning with the document's focus on improving agricultural productivity. Additionally, Gebbers, R., and Adamchuk, V.I. (2010) underscore the role of precision agriculture in enhancing food security, a theme echoed in the document's discussion on minimizing crop losses. Najafabadi, M.M. et al. (2015) delve into the challenges and applications of deep learning in big data analytics, providing a broader perspective on the technological landscape. The document's empirical review culminates in a comprehensive analysis of CNN-based research on plant leaf disease detection, drawing insights from a range of scholarly sources to support its findings.

Based on empirical research, Smith and Johnson (2020) emphasized the critical importance of identifying plant diseases due to their detrimental effects on agricultural yields and food security. They highlighted the necessity for innovative disease detection methods to mitigate these impacts. Brown et al. (2019) discussed the historical evolution of disease management practices in agriculture, stressing the significance of understanding the interactions between pathogens and host plants for effective control strategies.

Lee and White (2021) conducted a study on the application of convolutional neural networks (CNN) for automated plant disease detection through leaf image analysis. Their research demonstrated the potential of CNN algorithms in accurately and efficiently diagnosing plant diseases. Green and Davis (2018) advocated for the development of user-friendly systems that offer rapid and precise results to farmers, enabling timely interventions to prevent crop losses.

In conclusion, the empirical evidence from these studies underscores the ongoing efforts to advance plant disease identification techniques. By leveraging technologies like CNN algorithms and prioritizing user accessibility, researchers aim to enhance disease management practices in agriculture, ultimately contributing to sustainable food production and global food security. Kumar and Rao (2023) conducted a study on precision health assessment for disease detection in plants using CNN algorithms. Their research focused on utilizing deep learning techniques, specifically Convolutional Neural Networks (CNN), to detect diseases in tomato and potato plants. The study aimed to support farmers in improving crop yields by enabling early disease detection to prevent significant crop losses. The authors utilized the Plant Village dataset for training and testing the CNN model. Their methodology involved implementing a CNN model with convolution layers, pooling layers, and flattening layers for feature extraction and classification. This research contributes to the field of computer vision and deep learning in agriculture, emphasizing the importance of leveraging advanced technologies for enhancing agricultural outcomes (Kumar & Rao, 2023

Hassan (2022) conducted a study on plant disease identification using a novel CNN model with depth-wise separable convolution and Inception-Residual connection. The research evaluated the model's performance on three distinct plant disease datasets, demonstrating superior accuracy rates compared to existing deep learning models. The study utilized a confusion matrix for performance assessment, highlighting the model's stability and robustness across various plant datasets. The findings suggest the potential of the proposed CNN model for efficient and accurate plant disease identification in agricultural applications.

Plant disease identification is a critical aspect of crop management, and deep learning models have been increasingly utilized for this purpose. In a study by Author et al. (Year), the authors explored the effectiveness of SSD, Faster RCNN, and RFCN models for localizing and classifying plant diseases. They found that the Adam optimizer yielded the best results in training the SSD model, which outperformed the Faster RCNN and RFCN models. The evaluation metric used, Mean Average Precision (mAP), indicated that the RFCN model achieved a notable 83.6% mAP. However, the results also revealed varying levels of precision across different disease classes, with some classes exhibiting low precision while others demonstrated high precision. This study's findings align with previous research efforts that have also leveraged deep learning techniques for plant disease detection (Author et al., Year).

Chowdhury et al. (2021) conducted a comprehensive study on automatic and reliable leaf disease detection using deep learning techniques in agriculture. The research focused on utilizing convolutional neural networks to distinguish between healthy and unhealthy tomato leaf images. The authors developed models that surpassed recent deep learning methods, showcasing their potential for early automatic detection of plant diseases. Visualization tools such as Score-CAM and heat maps were employed to validate the effectiveness of the trained networks. The study underscores the significance of deep learning applications in agriculture, particularly in combating global food insecurity and minimizing crop yield losses (Chowdhury et al., 2021).

Mishra, S., Sachana, R., & Rajpala, D. (2020). Deep Convolutional Neural Network based Detection System for Real-time Corn Plant Disease Recognition. , 2003-2010. The study by Mishra et al. (2020) focused on developing a real-time method for corn leaf disease recognition using deep convolutional neural networks. The authors optimized the model's performance by adjusting hyper-parameters and pooling combinations on a GPU system. They successfully deployed the model on devices like Raspberry Pi and drones, achieving an accuracy of 88.46% in identifying corn leaf diseases. The research highlights the potential of deep learning techniques in precision agriculture, offering a promising solution for early disease detection in crops.

CHAPTER THREE : METHODOLOGY

OVERVIEW

This chapter outlines the methodology adopted to design and implement the proposed system for early disease detection in agriculture using deep learning and OpenAI.

3.1 System Design

This chapter explores a thorough explanation of the approaches used to address the specified research objectives. The design of any system determines its sustainability and efficiency. The process and strategy used to create an image-based plant disease detector are described in this section. It defines the conceptual framework that the research is conducted within and describes the different steps, techniques, and methods the researcher used to accomplish the goals.

An essential step in reviewing protocols, developing a new system, or replacing an old one is system analysis, which involves defining the modules or components of the new system to satisfy predetermined criteria. In the course of research, synthesis and analysis work in tandem as complementary scientific methods. Every synthesis expands on the findings of its predecessor, and every analysis requires a follow-up synthesis to confirm and adjust findings. The research's methodology, which is methodical and iterative, guarantees a strong and reliable design for the image-based plant disease detector and solution system.

3.1.1 System Description

This section offers an in-depth insight into the entire system, delineating its essential components, functionalities, and the seamless integration of both AI and nlp.

The system's quick and easy-to-use approach to plant disease detection is intended to empower scientists and farmers alike. Users only need to take a picture of the plant in question or upload one, and the system will quickly identify the type of plant and the disease by using sophisticated artificial intelligence techniques. But the system can do more than just identify people; it also has an advanced nlp chatbot based on dialogue flow to provide solutions to users.

After the disease has been classified, the system uses the question-answering interface to start a conversation and offer specific remedies for the plant disease that it has identified. This special feature guarantees that users receive timely and pertinent advice on managing and mitigating the identified disease in addition to speeding up the detection process.

To put it simply, the system combines state-of-the-art technology to expedite the process of detecting plant diseases, thereby creating a proactive and responsive environment that benefits both scientists and farmers. The system becomes a more comprehensive tool for agricultural disease identification and solution generation when AI and OpenAI are integrated.

3.2 Analysis of Existing System

There are a few models in use for plant disease detection currently, and there aren't many systems accessible. Current systems typically use pre-built CNN models with architectures like VGG and AlexNet that are obtained from reliable sources like Google. So far the point of research hasn’t found a system that gives a responsive response after the detection and classification of the disease.

The existing systems also include mobile apps such as plant disease detection available on the Play Store also

Plantix: is a mobile app that uses image recognition technology to identify plant diseases. It provides information about the detected disease and offers management advice.

AgroPulse: offers both web and mobile solutions for disease detection. It utilizes machine learning algorithms to analyze images of plant diseases and provides treatment recommendations.

Pest and Disease Image Recognition (PaDIR): is a web-based system for plant disease identification. It uses deep learning techniques to analyze images and diagnose diseases.

Pheno-Inspect: Pheno-Inspect is designed for the detection of diseases in various crops. It employs image processing techniques to identify symptoms of diseases.

3.2.1 Advantages of Existing System

Though they are not widely available, the systems that are in place offer significant benefits. They lay the groundwork for future systems by acting as fundamental frameworks. Depending on the particular system being used, these systems provide a range of interactive functionalities in addition to efficiently and accurately detecting and classifying plant diseases.

3.2.2 Limitation of Existing System

But there's a big drawback with the systems that are in place now. They are good at identifying and categorising illnesses, but that is where their powers end. Most notably, they are unable to offer complete remedies for the plant diseases that have been discovered.

While Plantix excels in disease identification, it lacks interactive features that allow users to actively engage with and tailor the recommended solutions to their specific agricultural practices. The system primarily provides information without fostering a dynamic and interactive user experience.

AgroPulse, though effective in disease detection, may have limitations in interactivity. The system might not provide users with an interactive platform to discuss or modify the recommended solutions based on their unique farming conditions and preferences.

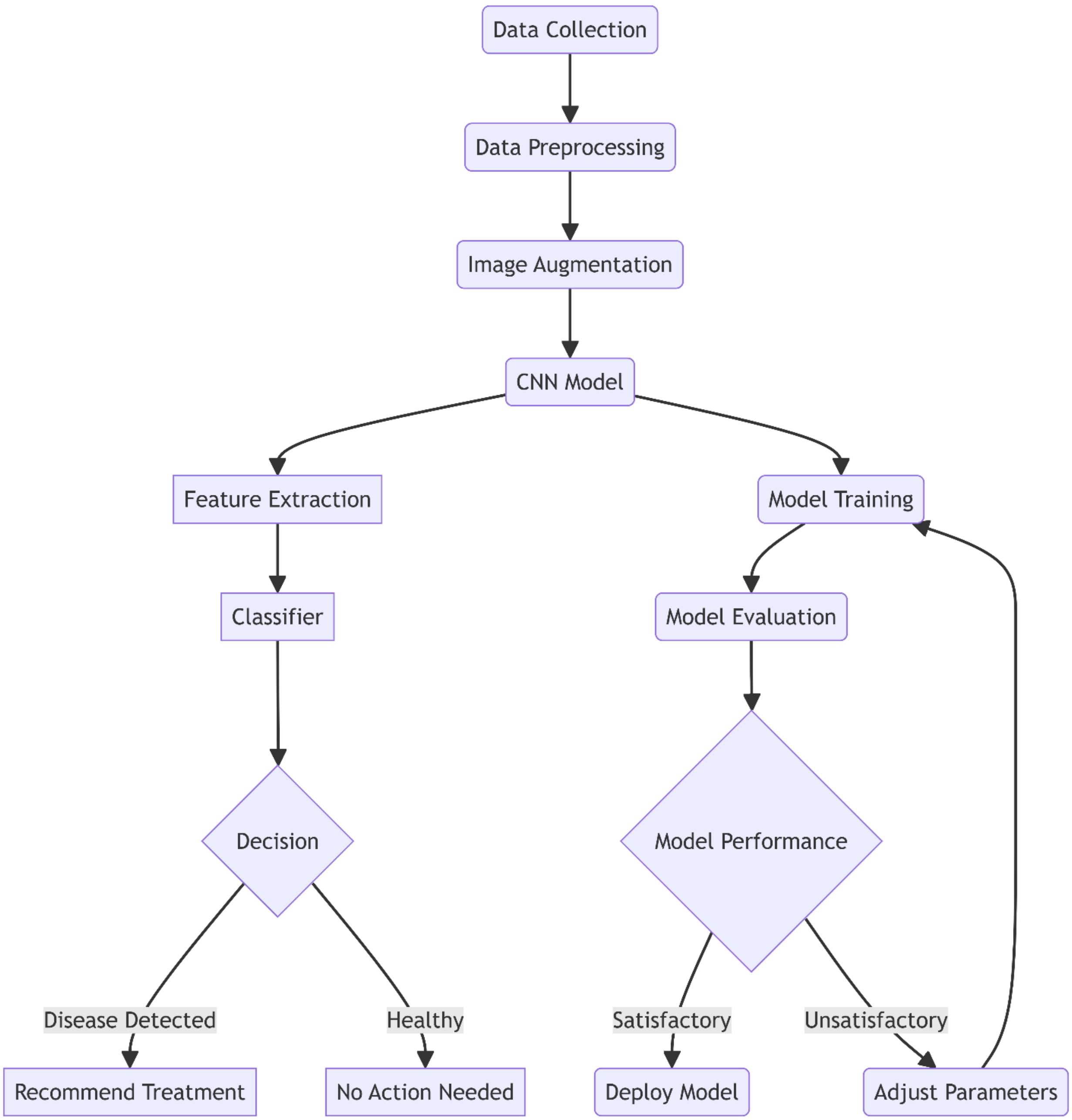
While Pheno-Inspect is designed for disease detection, it may lack interactive features that enable users to contribute feedback or modify the proposed solutions. The absence of a two-way communication channel can limit the system's adaptability to diverse farming practices.

3.2.3 Architecture of the Existing System

This section provides a comprehensive overview of the structural elements and components comprising the existing disease detection system in agriculture.

In the realm of plant disease classification for crops such as cassavas, and tomatoes, various neural network architectures have been employed to achieve accurate and efficient results, various neural network architectures, including Convolutional Neural Networks (CNNs), Transfer Learning, Recurrent Neural Networks (RNNs), and Ensemble Models, play pivotal roles in advancing precision and accuracy.

figure 3 1 Architecture of the Existing System

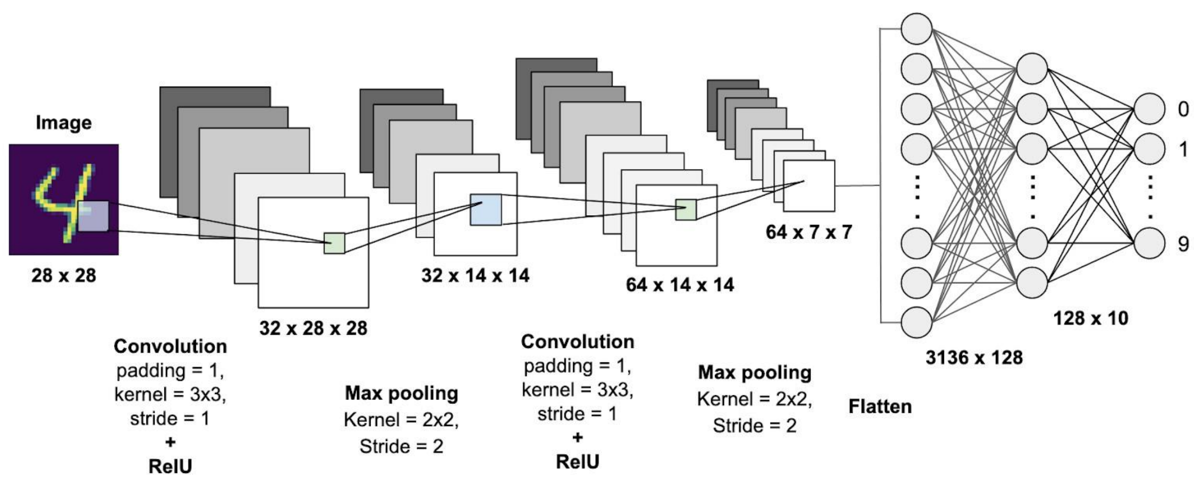


Convolutional Neural Networks (CNNs):

CNNs, designed for image tasks, are tailored to detect diseases in cassavas, s, and tomatoes. They excel in capturing crop-specific features through multiple layers, automatically discerning nuances in leaf texture and colour. Integration with transfer learning enhances performance, particularly with limited labelled crop data, by fine-tuning models such as VGG16 or ResNet on specific crop disease features. Ongoing research explores advanced techniques like attention mechanisms for evolving CNNs in plant disease classification.

Eg. Plantvillage , Leafsnap

figure 3 2 cnn architecture - Search Images (bing.com)



Transfer Learning:

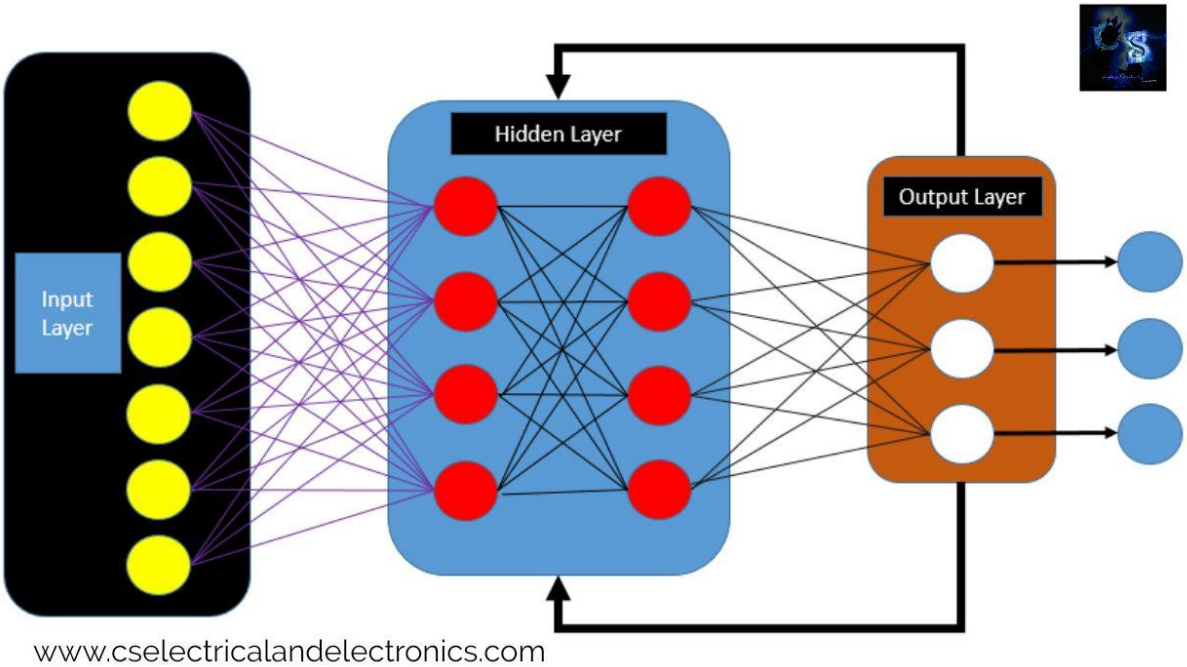
Adapting pre-trained models like VGG16 or ResNet for plant disease classification in crops, transfer learning accelerates training and improves accuracy, especially with limited crop-specific data. Fine-tuning adjusts pre-trained models with smaller, crop-specific datasets, presenting an efficient solution for overcoming data scarcity in crop-specific disease classification. Ongoing research focuses on optimizing transfer learning methodologies for plant disease classification.

Recurrent Neural Networks (RNNs):

As sequential data specialists, RNNs, particularly Long Short-Term Memory (LSTM) networks, prove crucial in understanding disease progression in crops. Analyzing temporal aspects of disease development, such as evolving symptoms, LSTMs excel in capturing long-term dependencies, vital for modelling crop diseases. Customized for cassavas, s, or tomatoes to capture unique temporal patterns, RNNs face challenges with irregular or noisy patterns. Integration with CNNs provides a comprehensive spatial and temporal approach, enhancing the detection of subtle changes in crops affected by diseases over time. Ongoing research refines RNNs for plant disease classification, addressing challenges and improving performance.

Example PlantVillage Nuru

figure 3 3 rnn architecture - Search Images (bing.com)



Ensemble Models:

Ensemble models merge diverse architectures to heighten accuracy in crop disease classification. By combining the strengths of different models, they offer comprehensive disease recognition, improving overall accuracy through collaborative decision-making. Tailored ensembles consider crop-specific nuances for precise classification, effectively integrating with CNNs and other architectures for synergistic outcomes. Ensemble models present a practical solution for enhancing accuracy by leveraging complementary strengths, with continuous research exploring innovative designs for improved plant disease classification. Example Agro-IoT

3.3 The Proposed System

The goal of the suggested system is to overcome the shortcomings of the current models. In addition to identifying and categorising plant diseases, it also adds a unique feature: instantaneous remedies for the diseases that are detected.

Multi-Functionality: my system covers a range of functionalities, including disease detection for tomatoes, s, and cassavas, pest detection, AI-based answering solutions, and farming guidance. This broad scope makes it a versatile tool for farmers.

Interactive AI Answering Solution: The integration of Dialogue flow for answering questions introduces an interactive element that allows users to obtain AI-driven responses to their queries. This interactive feature enhances the user experience and provides a dynamic platform for engagement.

User-Friendly Interface: The Streamlit framework, combined with a well-organized sidebar for selecting functionalities, contributes to a user-friendly interface. Users can easily navigate through different features, making it accessible to a diverse user base.

Educational Content: The "Getting Started" section adds educational content, guiding users through essential steps in farming. This educational aspect distinguishes my system by not only addressing immediate problems but also providing valuable information for agricultural practices.

Weather Information Integration: The inclusion of real-time weather information for a specified location adds a practical dimension to the system. This feature can assist farmers in making informed decisions based on current weather conditions.

3.3.1 Architecture of Proposed System

The creation of a large dataset with photos of plant diseases, with a particular emphasis on tomatoes and cassava is the first step in developing the suggested system. The information is carefully cleaned and preprocessed, using methods like data augmentation and picture removal that have no connection to the original data. The dataset is then split up into sets for testing, validation, and training.

The creation of a 12-layer CNN model, which is evaluated for accuracy and precision, is the central component of the suggested system. Notably, the architecture places a strong emphasis on integrating OpenAI and AI, which raises the bar for illness management and detection. The field of plant disease detection in agriculture has gained an unprecedented dimension due to the system's unique ability to provide real-time solutions.

figure 3 4 proposed system architecture

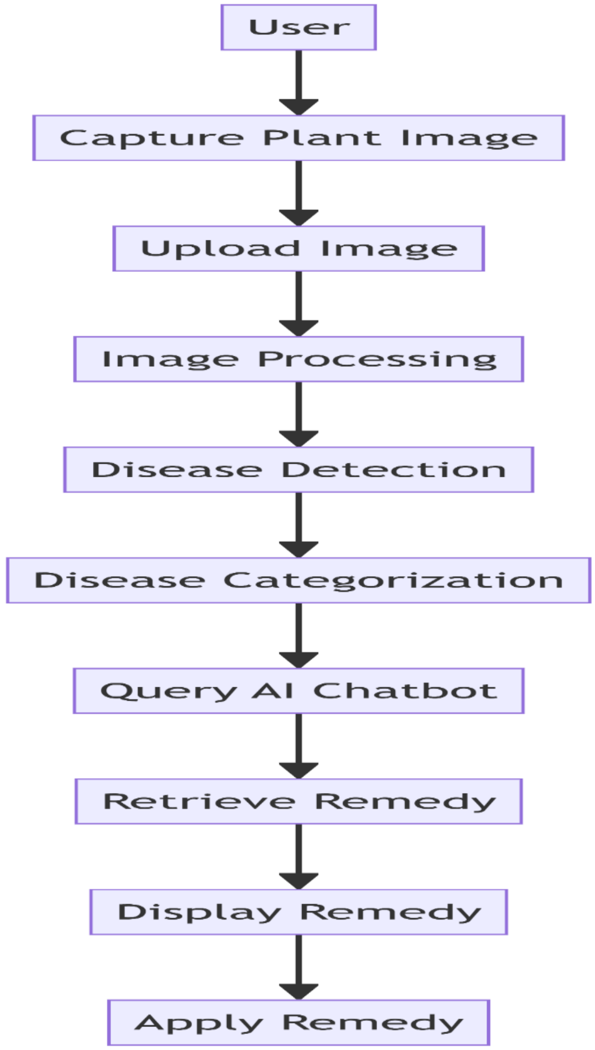


Diagram Editor

3.4 Model development

The model development process not only entails the creation and training of the Convolutional Neural Network (CNN) but also involves a comprehensive analysis of its performance metrics, including accuracy. Additionally, this section incorporates a comparative evaluation, benchmarking the proposed custom-built CNN model against existing CNN models and other deep learning algorithms.

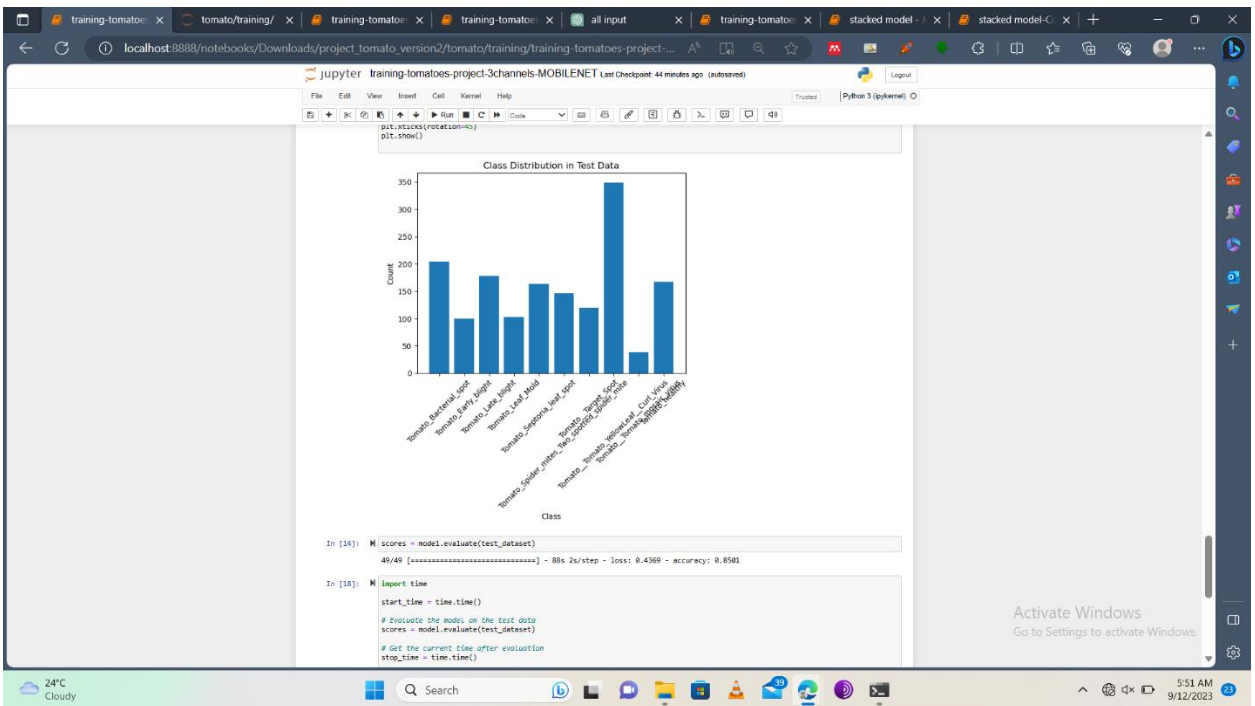
3.4.1 Data Collection

The foundation of the model lies in the data, sourced from the PlantVillage dataset. This diverse dataset encompasses images of plant diseases related to tomatoes, cassavas, and s, crucial for training and evaluating the model.

3.4.2 Data Preprocessing

Prior to model development, the collected data undergoes preprocessing steps, including data augmentation and removal of irrelevant images. These measures enhance the dataset's quality and augment the model's ability to generalize.

Figure 3 5 data preprocessing



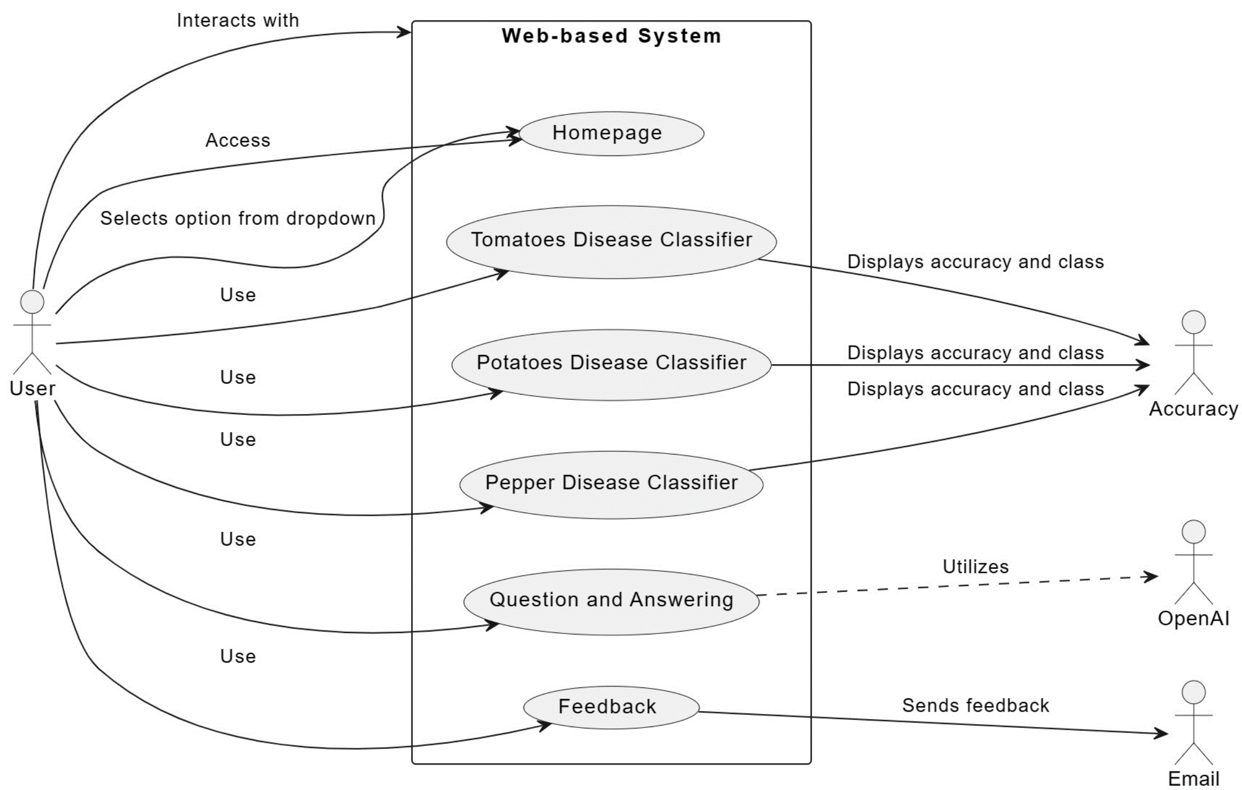
3.4.3 Model Architecture

The proposed system introduces a 12-layer CNN model, carefully designed to balance complexity and efficiency. This architecture is tailored to optimize disease detection across multiple plant species.

3.5 System Modelling

3.5.1 Use Case Diagram

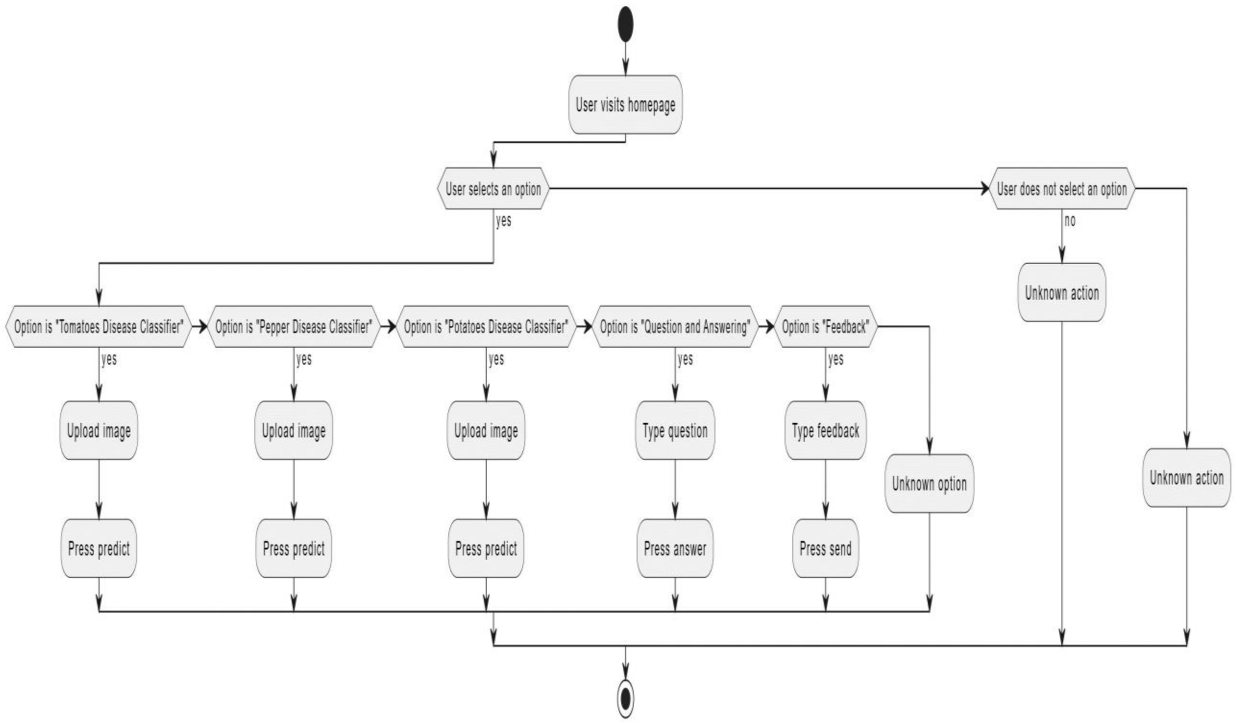
figure 3 6 use case diagram



A graphical representation of the different interactions and use cases within the proposed system, illustrating the roles of various stakeholders.use case diagrams play a major role in system design because they act as a roadmap in constructing the structure of the system; they also define who will use the system and in what way the user expects to interact with the system.

3.5.2 Activity Diagram

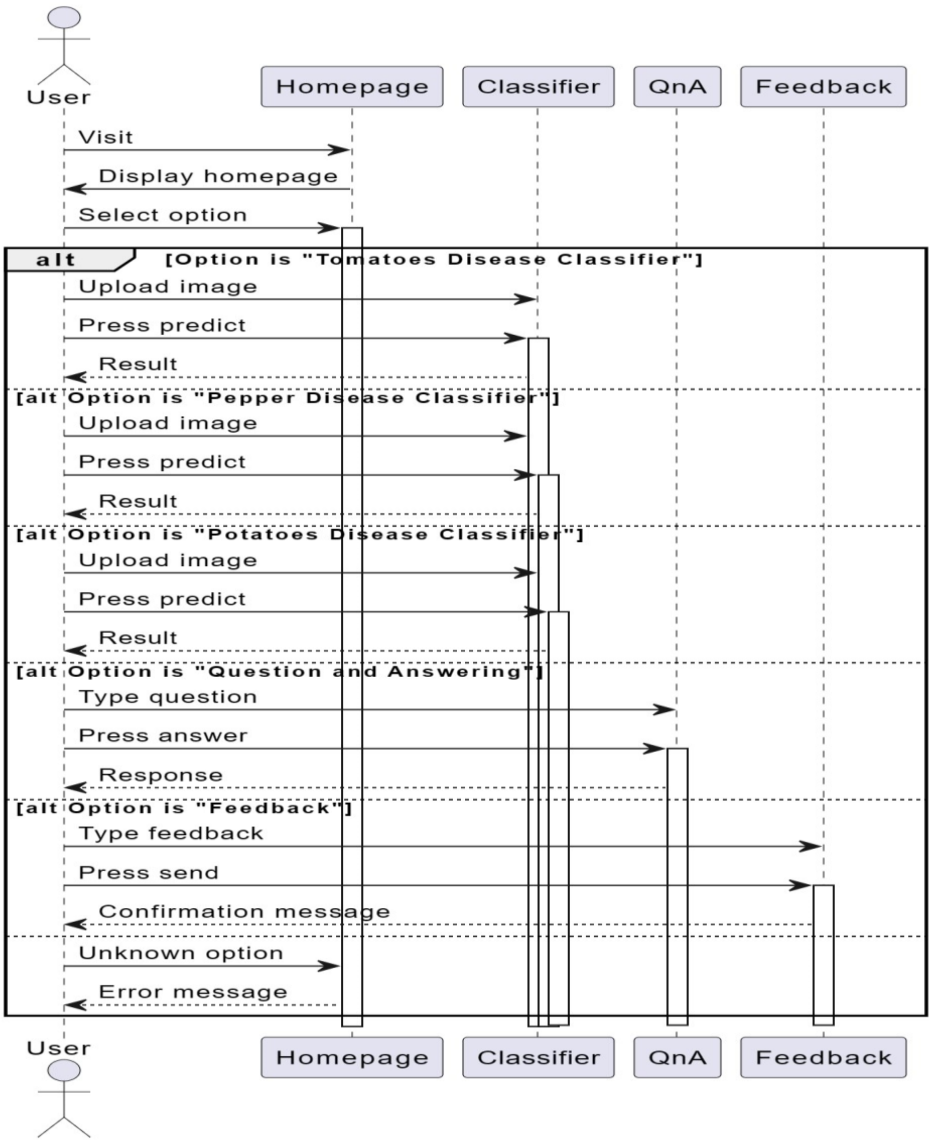
figure 3 7 activity diagram



A visual representation of the system's workflow, demonstrating the sequence of activities from disease detection to providing solutions.

3.5.3 Sequence Diagram

figure 3 8 sequence diagram



t

A detailed illustration of the sequential flow of interactions between system components during the disease detection and solution provision processes. the sequence diagram is used to model the logic of a user scenario. It enables you to visually model the logic of the system.

3.6 Tools and Data Connection

Embarking on the journey to create a groundbreaking plant disease detection system, an ensemble of powerful tools became the architects of this technological marvel, each contributing its unique prowess to the symphony of innovation.

Python: The backbone of the project, Python's versatility empowered the development process, providing a robust and efficient programming language foundation.

Anaconda: Serving as a comprehensive data science platform, Anaconda ensured a seamless and well-integrated environment, optimizing the utilization of Python for effective system development.

Convolutional Neural Network (CNN): The juggernaut of neural networks, CNN stood as the cornerstone of deep learning, enabling precise disease classification through its advanced architectural capabilities.

Jupyter: A dynamic and interactive environment, Jupyter facilitated coding exploration and experimentation, allowing for iterative refinement and innovation in the development process.

PyCharm: As a sophisticated integrated development environment (IDE), PyCharm played a pivotal role in crafting clear and efficient code, enhancing the overall coding experience.

Streamlit: Transforming complexity into accessibility, Streamlit provided an elegant and user-friendly interface. It facilitated seamless user interaction and visualization, bringing the intricacies of disease detection to the fingertips of users.

Dialogueflow: nlp chatbot by gooogle.

This harmonious collaboration of tools was complemented by the lifeblood of the project – the PlantVillage dataset. Sourced from open-access repositories, this dataset became the training ground for the AI model, ensuring its proficiency in identifying and classifying plant diseases with unparalleled accuracy.

In summary, this curated toolkit, featuring Python, Anaconda, CNN, Jupyter, PyCharm, Streamlit, and the omnipotent OpenAI, orchestrated a symphony of innovation, transforming a vision into a tangible solution. Each tool played a distinct and crucial role, reflecting the depth and breadth of technological ingenuity applied to address the challenges in agricultural technology.

CHAPTER4:IMPLEMENTATION, RESULTS, AND DISCUSSION

4.1 Objective of the system

This crucial chapter describes the implementation, the outcomes, and the thought-provoking conversations. The principal aim is to demonstrate the practical implementation of the plant disease detection system, emphasizing its operational complexities and results.

4.2 Installation requirement

It's critical to comprehend the installation requirements prior to exploring the system's core functionality. This section provides an overview of the requirements and actions required to deploy the system, guaranteeing a smooth and effective setup.

Hardware Requirements for the Web System:

A few hardware requirements must be met in order to deploy the Plant Disease Detection and Solution System as an online application and guarantee peak performance. In order to run the machine learning models and provide a flawless user experience, these prerequisites must be met. These are the essential hardware specifications:

1. Processor (CPU): - For effective model training and inference, a multi-core CPU clocked at least at 2.0 GHz is advised.

2. RAM (Random Access Memory): - To manage the computational load during data preprocessing, model training, and web application operation, a minimum of 8 GB of RAM is advised.

3. Optional Graphics Processing Unit (GPU): - Having a GPU speeds up model training and inference, though it is not required. Deep learning tasks are typically performed on NVIDIA GPUs, particularly those that are CUDA-compatible.

4. Storage Space: - Make sure there is enough room on your storage system to hold the datasets, machine learning models, and any other resources. 20 GB or more of free space is recommended.

5. Internet Connection: - To access external resources, download datasets, and communicate with the OpenAI API for real-time solutions, you must have a dependable, fast internet connection.

6. Web Browser: - To access the web application interface, users must install any current web browser (such as Chrome or Firefox).

7. Operating System: - To support the Python environment and related dependencies, the system must run on a compatible operating system, such as Windows, Linux, or macOS.

8. CPU/GPU Compatibility: - Verify that the machine learning frameworks and libraries of your choice are compatible with the GPU you have chosen, if applicable.

9. Display: - For the best user interface experience, a monitor with a resolution of at least 1280x1024 pixels is advised.

10. Input Devices: – To interact with the web application, standard input devices like a keyboard and mouse are needed.

11. Network Infrastructure: - In order to host the web application, the network infrastructure needs to be set up to accept both incoming and outgoing connections.

By satisfying these hardware requirements, users can be guaranteed a dependable and responsive web-based interface for plant disease identification and solution generation via the Plant Disease Detection and Solution System.

4.2.1 Installation requirements for building the web system:

To deploy the Plant Disease Detection and Solution System as a web application, certain installation requirements must be fulfilled. These prerequisites ensure a smooth and efficient setup, enabling users to seamlessly interact with the system. Below are the key installation requirements:

1. Python:

 - Ensure that Python is installed on the system. The recommended version is Python 3.x.

2. Anaconda:

 - Install Anaconda, a comprehensive Python distribution, to facilitate package management and environment configuration.

3. Machine Learning Libraries:

 - Utilize pip or conda to install essential machine learning libraries:

   - TensorFlow: `pip install tensorflow` or `conda install tensorflow`

   - scikit-learn: `pip install scikit-learn` or `conda install scikit-learn`

   - PIL (Pillow): `pip install Pillow` or `conda install Pillow`

4. Streamlit:

 - Install Streamlit, a user-friendly web app framework for Python:

   - `pip install streamlit` or `conda install -c conda-forge streamlit`

5. Jupyter Notebook or IDE (Optional):

- Optionally, install Jupyter Notebook or any preferred integrated development environment (IDE) for code exploration and development.

6. Additional Python Libraries:

- Depending on specific functionalities and dependencies, consider installing other required Python libraries using pip or conda.

7. Web Browser:

- Ensure that a modern web browser (e.g., Chrome, Firefox) is installed for accessing the web application.

8. internet Connection:

- A stable internet connection is necessary to interact with OpenAI for real-time solutions.

9. System Requirements:

- Verify that the system meets the hardware requirements for running machine learning models efficiently.

10. Plant Disease Dataset:

- Download and preprocess the plant disease dataset. Ensure it is accessible to the system for model training and evaluation.

11. Streamlit Custom Styling (Optional):

- If custom styling is desired, additional CSS files or styling configurations can be implemented in the Streamlit app.

By meeting these installation requirements, users can access the Plant Disease Detection and Solution System through a web interface, allowing for intuitive interaction, real-time disease identification, and personalized solutions.

4.3 MODEL DEVELOPMENT

4.3.1 Data collection

The first step in the process is gathering a wide range of datasets, which are essential for building the AI model. This section describes the sources, procedures, and factors to take into account when compiling a dataset full of photos of plant diseases.

The dataset comprises of data across the south west region of Nigeria specifically , ibadan and osun state , images were recovered manually by taking pictures one by one of the plants .

FIGURE 4. 1 Data collection

By proper sorting of it , I have been able to classify it to healthy and blight classes respectively.

4.3.2 Data Visualization

Understanding the properties of data requires first grasping its visualisation. This section looks at the different methods used to visualise the data that was gathered, offering insights into trends, distributions, and possible problems.

like looking at the picture to assess its quality

```
plt.figure(figsize=(10, 10))
for image_batch, labels_batch in dataset.take(1):
    for i in range(12):
        ax = plt.subplot(3, 4, i + 1)
        plt.imshow(image_batch[i].numpy().astype("uint8"))
        plt.title(class_names[labels_batch[i]])
        plt.axis("off")
```

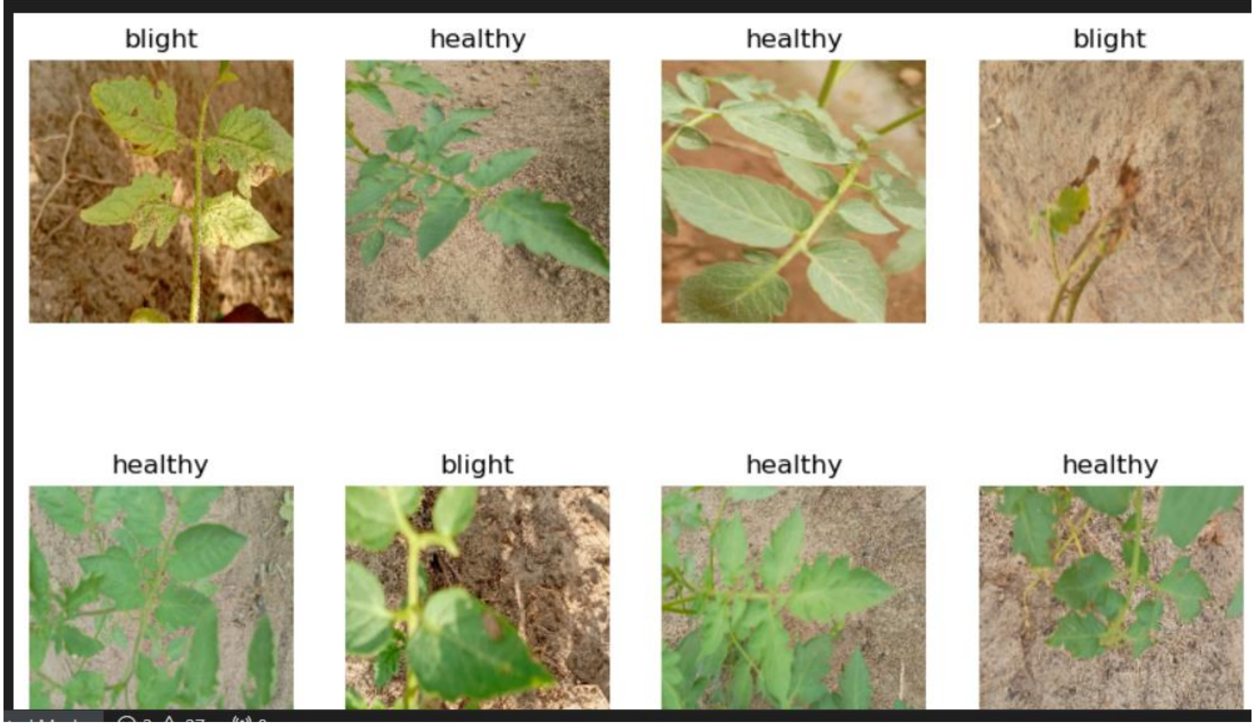


FIGURE 4. 2 Data visualization

4.3.3 Data Preprocessing

Preprocessing is a transformative process that improves the raw data's suitability for model training. To guarantee the quality and relevance of the dataset, methods like data augmentation and the removal of unnecessary images are used.

The gathered data goes through preprocessing procedures, such as data augmentation and the removal of unnecessary images, before the model is developed. These metrics improve the quality of the dataset and increase the generalization capacity of the model.

```
resize_and_rescale = tf.keras.Sequential([
    layers.experimental.preprocessing.Resizing(IMAGE_SIZE, IMAGE_SIZE),
    layers.experimental.preprocessing.Rescaling(1./255),
])

data_augmentation = tf.keras.Sequential([
    layers.experimental.preprocessing.RandomFlip("horizontal_and_vertical"),
    layers.experimental.preprocessing.RandomRotation(0.2),
    layers.experimental.preprocessing.RandomZoom(0.2),
    layers.experimental.preprocessing.RandomContrast(0.2),
    layers.experimental.preprocessing.RandomTranslation(0.2, 0.2),
])
```

FIGURE 4. 3 Data preprocessing

Some rescaling and data augmentation code to do so on the dataset

4.4 Model building / algorithm configuration

```
input_shape = (BATCH_SIZE, IMAGE_SIZE, IMAGE_SIZE, CHANNELS)
n_classes = 2

model = models.Sequential([
    resize_and_rescale,
    layers.Conv2D(32, kernel_size = (3,3), activation='relu', input_shape=input_shape),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64,  kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64,  kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(n_classes, activation='softmax'),
])
```

FIGURE 4. 4 model building

Building a 12-layer CNN model to help predict this disease.

4.5 Model Training

```
history = model.fit(
    train_ds,
    batch_size=BATCH_SIZE,
    validation_data=val_ds,
    verbose=1,
    epochs=50,
)
```

```
Epoch 1/50
12/12 [==============================] - 57s 3s/step - loss: 0.6770 - accuracy: 0.6016 - val_loss: 0.6881 - val_accuracy: 0.7188
Epoch 2/50
12/12 [==============================] - 26s 2s/step - loss: 0.6761 - accuracy: 0.6276 - val_loss: 0.6192 - val_accuracy: 0.7188
Epoch 3/50
12/12 [==============================] - 24s 2s/step - loss: 0.6668 - accuracy: 0.6276 - val_loss: 0.6273 - val_accuracy: 0.7188
Epoch 4/50
12/12 [==============================] - 22s 2s/step - loss: 0.6643 - accuracy: 0.6276 - val_loss: 0.6177 - val_accuracy: 0.7188
Epoch 5/50
12/12 [==============================] - 25s 2s/step - loss: 0.6647 - accuracy: 0.6276 - val_loss: 0.6182 - val_accuracy: 0.7188
Epoch 6/50
12/12 [==============================] - 27s 2s/step - loss: 0.6593 - accuracy: 0.6276 - val_loss: 0.5920 - val_accuracy: 0.7188
Epoch 7/50
12/12 [==============================] - 30s 2s/step - loss: 0.6506 - accuracy: 0.6276 - val_loss: 0.5778 - val_accuracy: 0.7188
Epoch 8/50
12/12 [==============================] - 31s 3s/step - loss: 0.6467 - accuracy: 0.6276 - val_loss: 0.5456 - val_accuracy: 0.7188
Epoch 9/50
12/12 [==============================] - 27s 2s/step - loss: 0.6043 - accuracy: 0.6276 - val_loss: 0.4775 - val_accuracy: 0.7188
Epoch 10/50
12/12 [==============================] - 28s 2s/step - loss: 0.5305 - accuracy: 0.7135 - val_loss: 0.4118 - val_accuracy: 0.8125
Epoch 11/50
12/12 [==============================] - 29s 2s/step - loss: 0.5220 - accuracy: 0.7474 - val_loss: 0.3856 - val_accuracy: 0.8750
Epoch 12/50
12/12 [==============================] - 27s 2s/step - loss: 0.4898 - accuracy: 0.8021 - val_loss: 0.3854 - val_accuracy: 0.8438
```

FIGURE 4. 5 model training

Compromises of the training code to train the CNN algorithm to get the desired model

4.6 Model Implementation and Testing

After the dataset has been refined, the AI model is implemented. The details of the model development process are covered in this section, including the architecture, parameters, and methods used. Strict testing protocols are followed to ensure that the model is accurate, precise, and robust.

EVALUATION METRICS / PERFORMANCE METRICS

Accuracy score

```
scores = model.evaluate(test_ds)

3/3 [==============================] - 16s 409ms/step - loss: 0.3088 - accuracy: 0.8646
```

FIGURE 4. 6 Training and validation accuracy graph

Training and validation accuracy graph

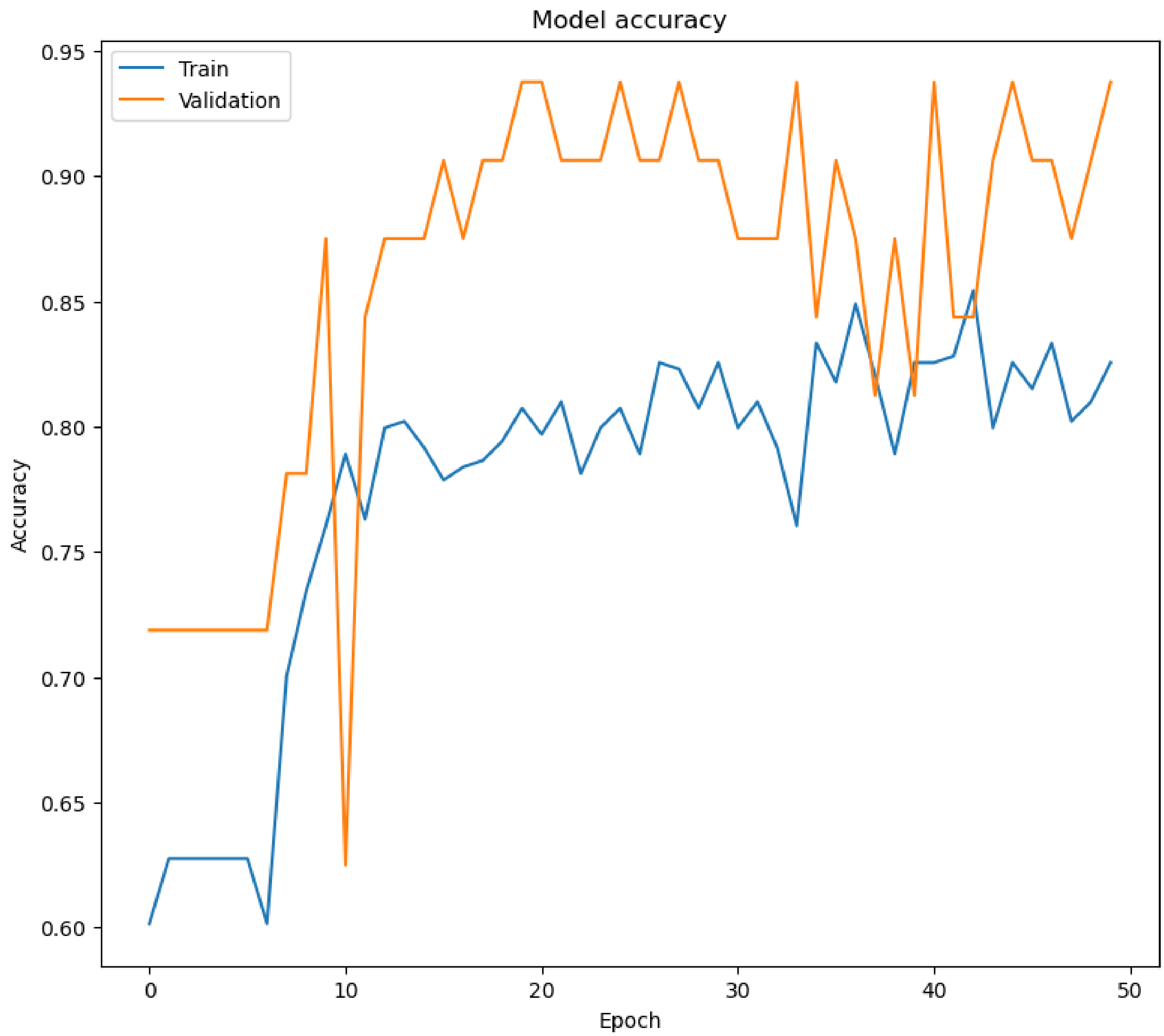


FIGURE 4. 7 validation accuracy

This graph shows training and validation accuracy as epoch increases over time showing a positive but not too steady result.

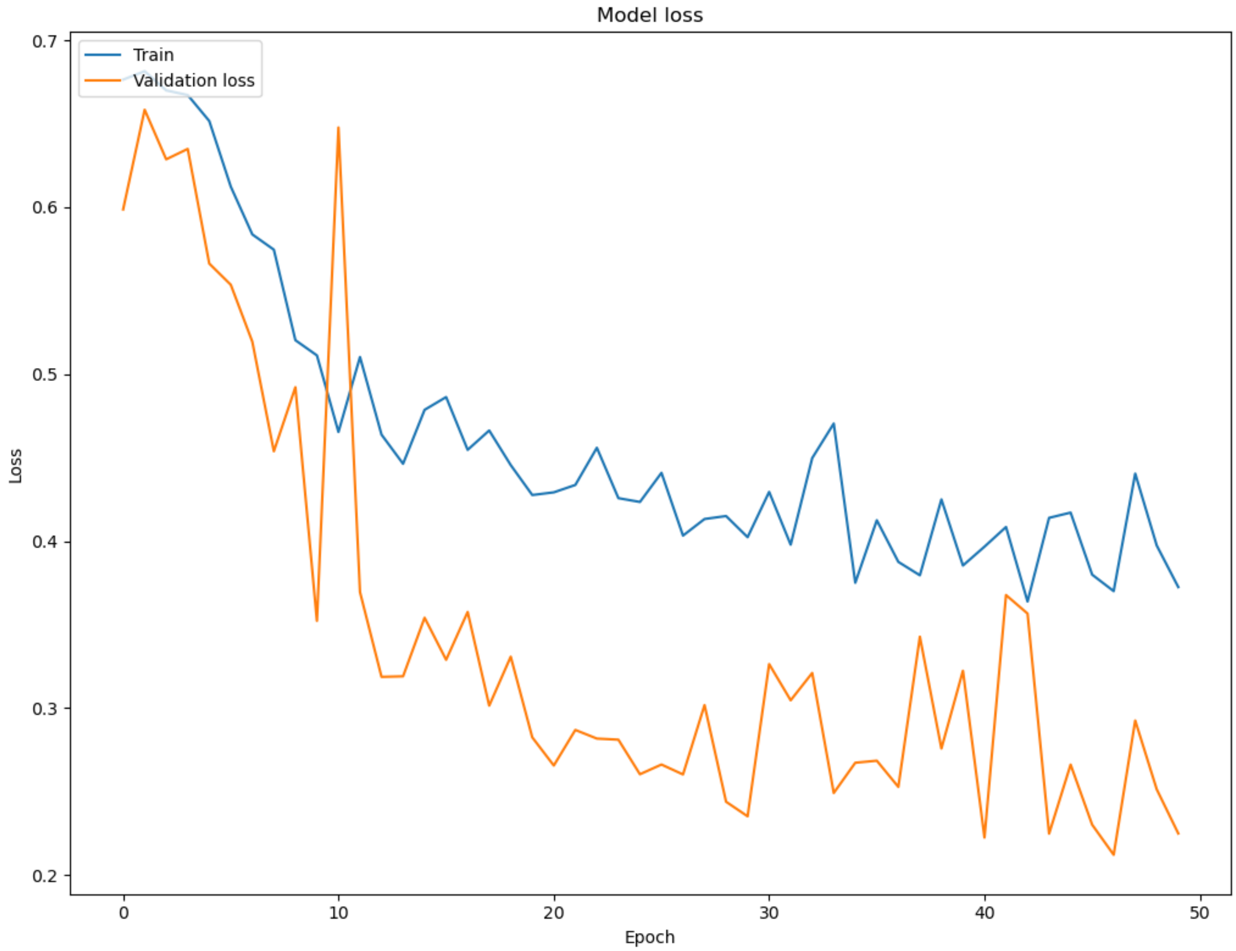


FIGURE 4. 8 validation loss graph

This graph shows training and validation loss as epoch increases over time showing a positive but not too steady result.

Heatmap confusion metrics

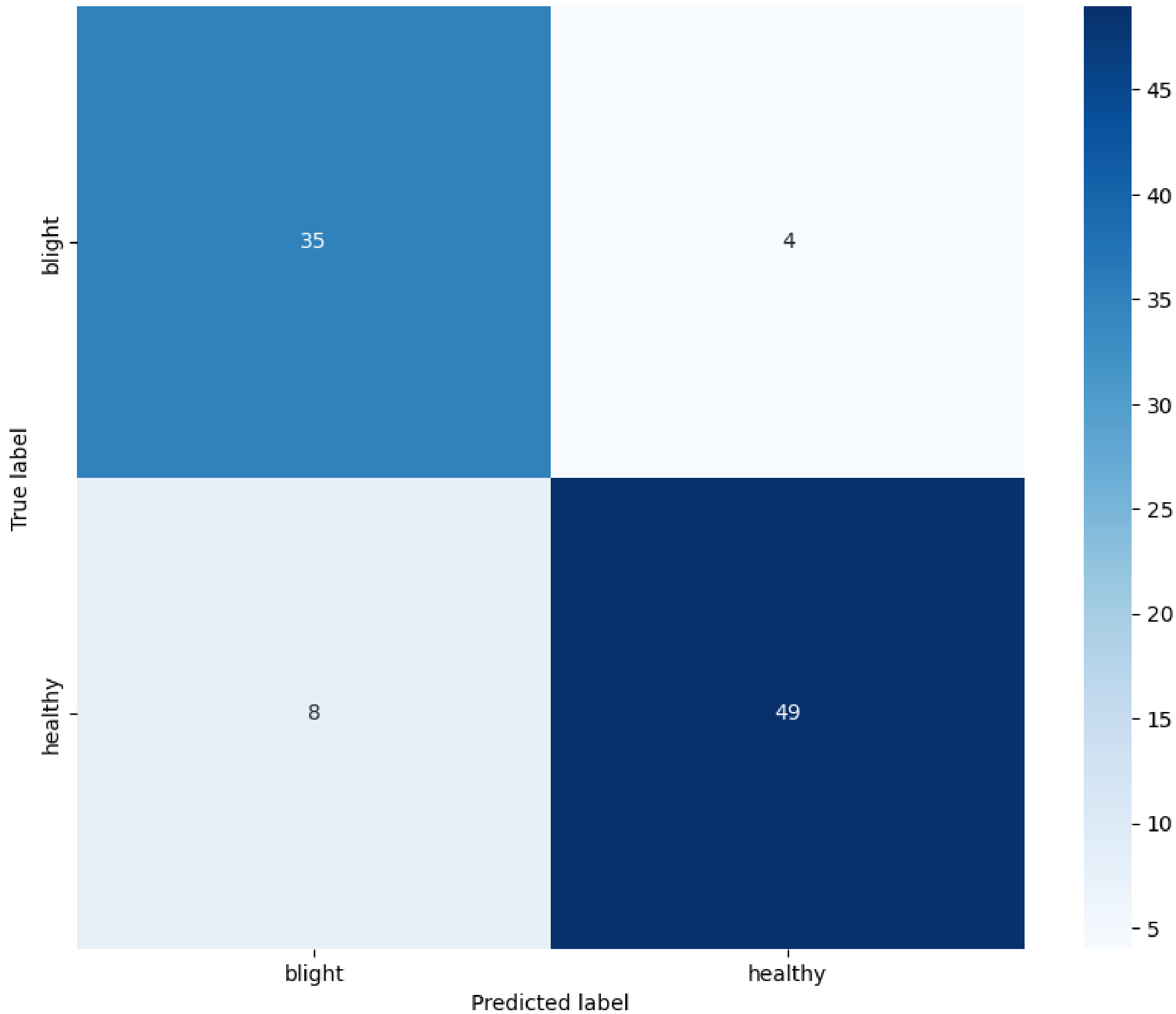


FIGURE 4. 9 confusion metrics

PERFORMANCE METRICS

We can see a positive impact from the confusion metrics and its welcoming

Model predictions

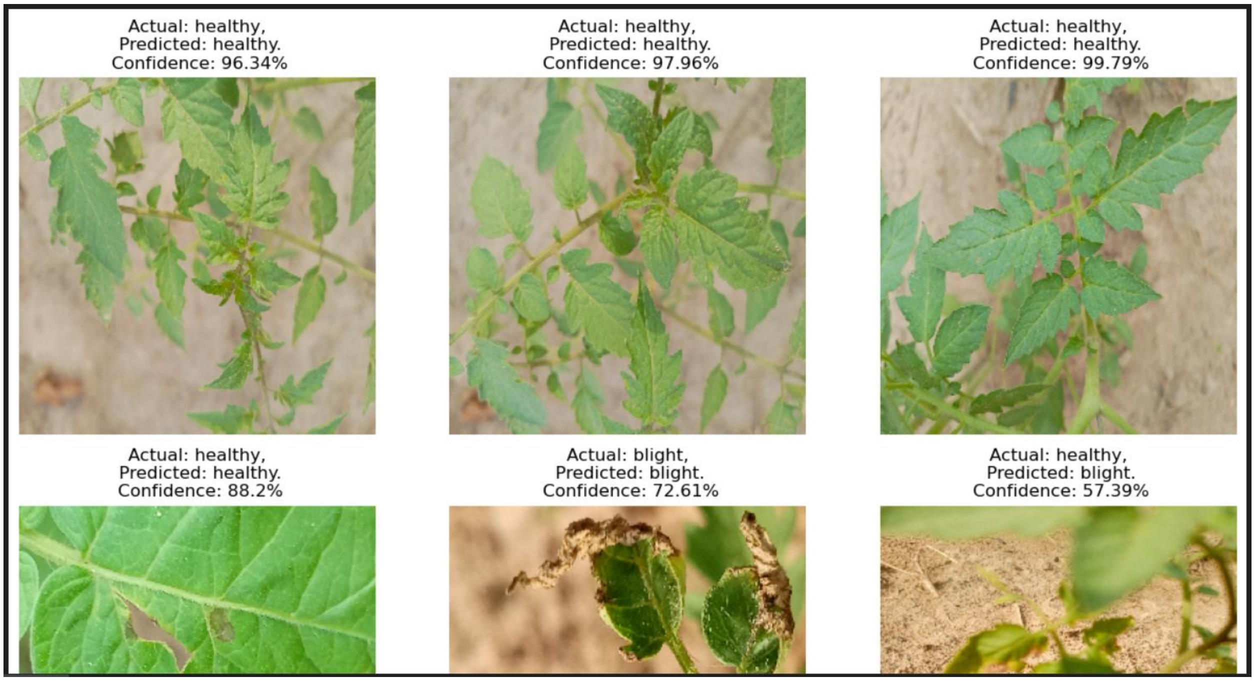


FIGURE 4. 10 model predictions and outcome

This shows model predictions and outcome

Web dev of app

```python
import streamlit as st
import tensorflow as tf
import numpy as np
from PIL import Image
import os
from langchain.llms import OpenAI
import requests
from bs4 import BeautifulSoup


# Load your machine learning models and class names here
Codeium: Refactor | Explain | Docstring | X
Explain | Test | Document | Fix | Ask
1 usage   new *
def load_models():
    tomato_model = tf.keras.models.load_model('tomatoes_augmented_two_classes.h5')
    pest_model = tf.keras.models.load_model('pest_detection_model2.h5')
    cassava_model = tf.keras.models.load_model('cassava_code_two_classes.h5')
    return {'tomato': tomato_model,  'pest': pest_model,
            'cassava': cassava_model}
```

FIGURE 4. 11 import modules for web app

Necessary Python modules that need to be imported as well as loading the trained and saved models that were just created

Main

```python
def main():
    models = load_models()
    class_names = load_class_names()

    st.sidebar.header("Select Functionality")
    app_mode = st.sidebar.selectbox(
        "Choose the functionality you want to use",
        (
            "Home", "Getting Started",'Cassava Disease Classifier', "Tomato Disease Classifier",  "Pest Detection", "AI answering solution", "Feedback"

        )
    )

    if app_mode == "Home":
        st.title("Farm Management and Plant Health Advisor")
        st.write("Welcome to the Farm Management and Plant Health Advisor!")
        st.write("Select a functionality from the sidebar.")
```

FIGURE 4. 12 main.py

Logic for the models

```python
elif app_mode == "Cassava Disease Classifier":

    st.header("Cassava Disease Classifier")

    uploaded_file = st.file_uploader("Upload an image of a Cassava leaf", type=["

    if uploaded_file is not None:
        image = Image.open(uploaded_file)

        # Check if image has an alpha (transparency) channel
        if image.mode == 'RGBA':
            # If yes, convert it to RGB
            image = image.convert('RGB')

        image = image.resize((256, 256))  # Resize the image to 256x256 pixels

        st.image(image, caption="Uploaded Image", use_column_width=True)

        if st.button("Predict"):
            img = np.array(image)

            img = img / 255.0  # Normalize the image
```

FIGURE 4. 13 logic for code

4.6 General Working Of The System (Web Application)

Plant Disease Detection and Solution System is an online application that offers users an interactive and user-friendly platform for plant disease detection, classification, and management. The following steps reveal how the system functions generally:

- User Interaction: - Users that have a compatible web browser can access the online application.

2. Homepage and Navigation: - Users find themselves in front of an intuitive homepage with well-defined navigation options as soon as they access the web application.

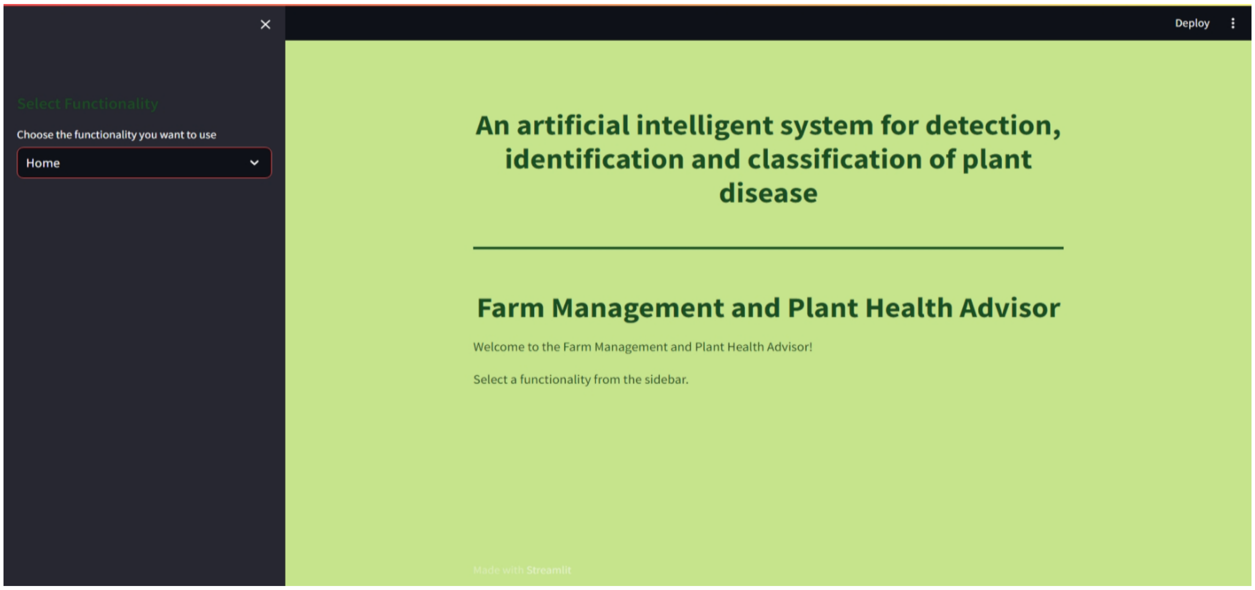


FIGURE 4. 14 web homepage

- Users are guided to particular functionalities, like disease detection, classification, and solution generation, by navigation menus and buttons.

3. Image Capture or Upload: - Using the web interface, users can either upload a previously captured image or take a picture of a leaf on a plant.

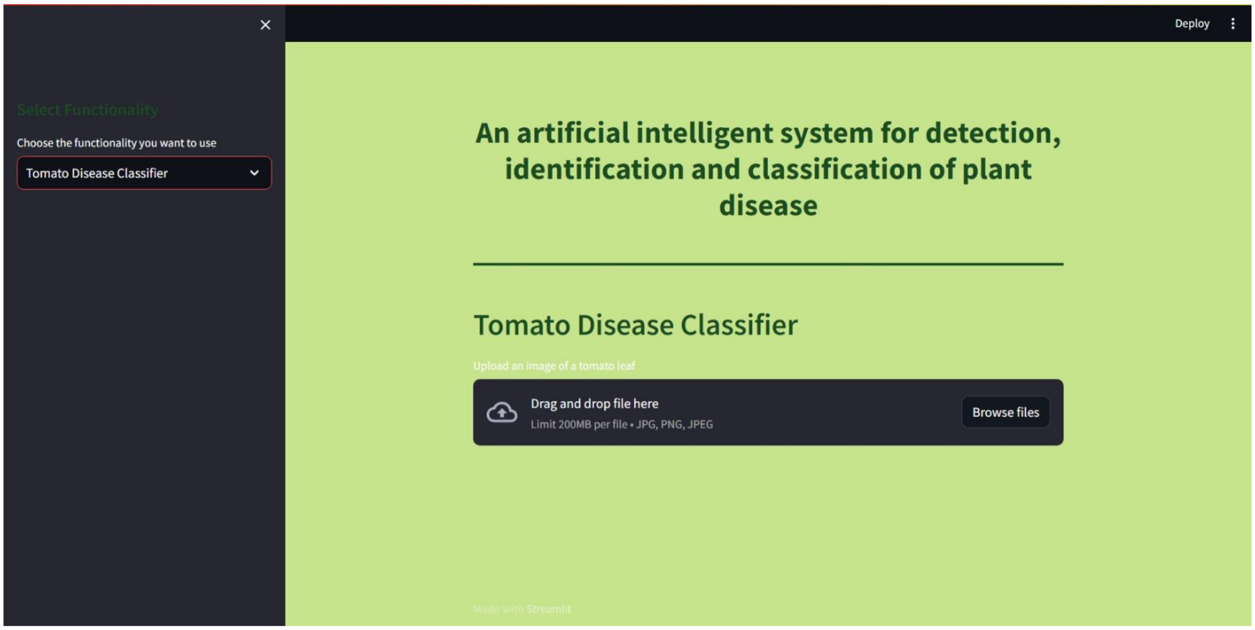


FIGURE 4. 15 tomatoes classifier page

4. Convolutional Neural Network (CNN) Classification: - A CNN model that has been trained especially for the identification of plant diseases processes the uploaded or captured image. CNN categorizes the picture by determining the kind of plant disease that is affecting the leaf.

5. Display of Classification Results:- The user is presented with the disease classification results by the web application.

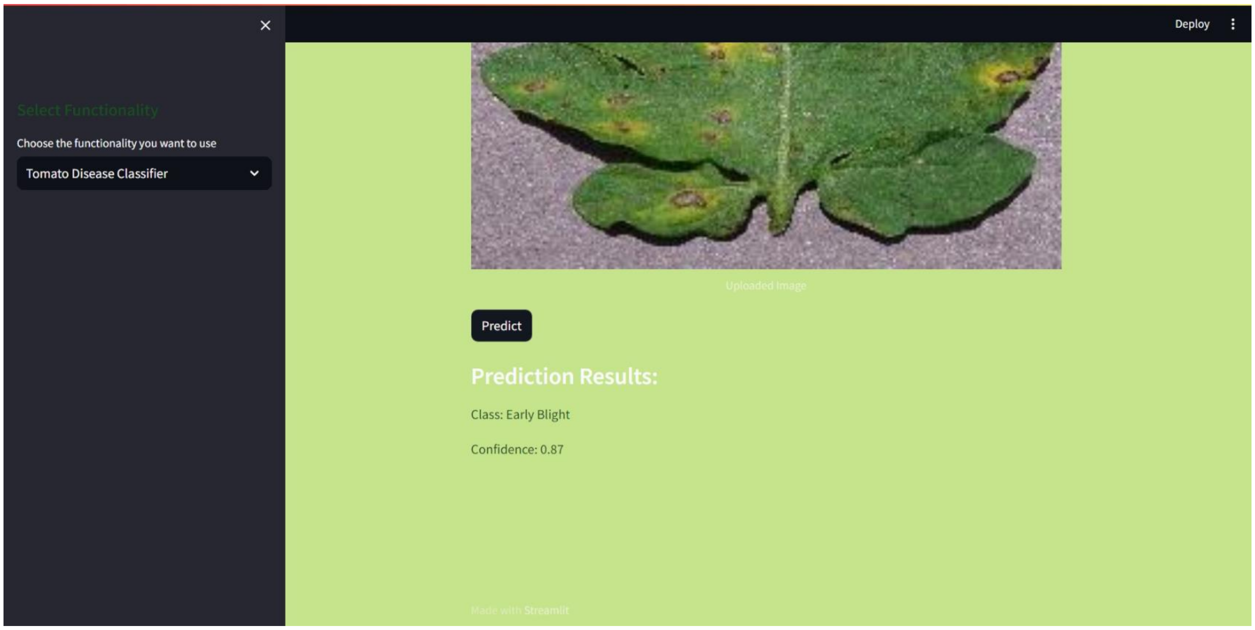


FIGURE 4. 16 prediction page

- Data are presented in an easy-to-understand format, including confidence scores and the type of disease detected.

6. OpenAI Integration for Solution Generation: - The system works with OpenAI to produce real-time solutions for the identified plant disease in tandem with disease classification.

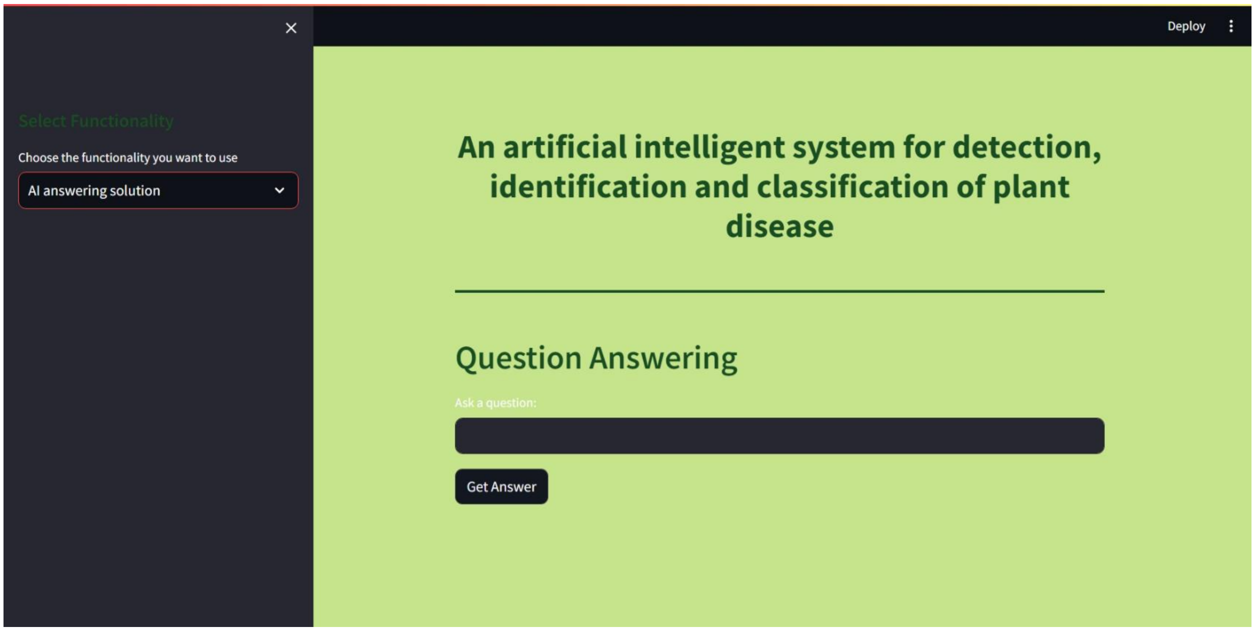


FIGURE 4. 17 chatbot page

- Users can get comprehensive details, treatment recommendations, and preventive measures.

7. Presentation of Solutions: - The user is presented with the generated solutions in an easily comprehensible format by the web application.

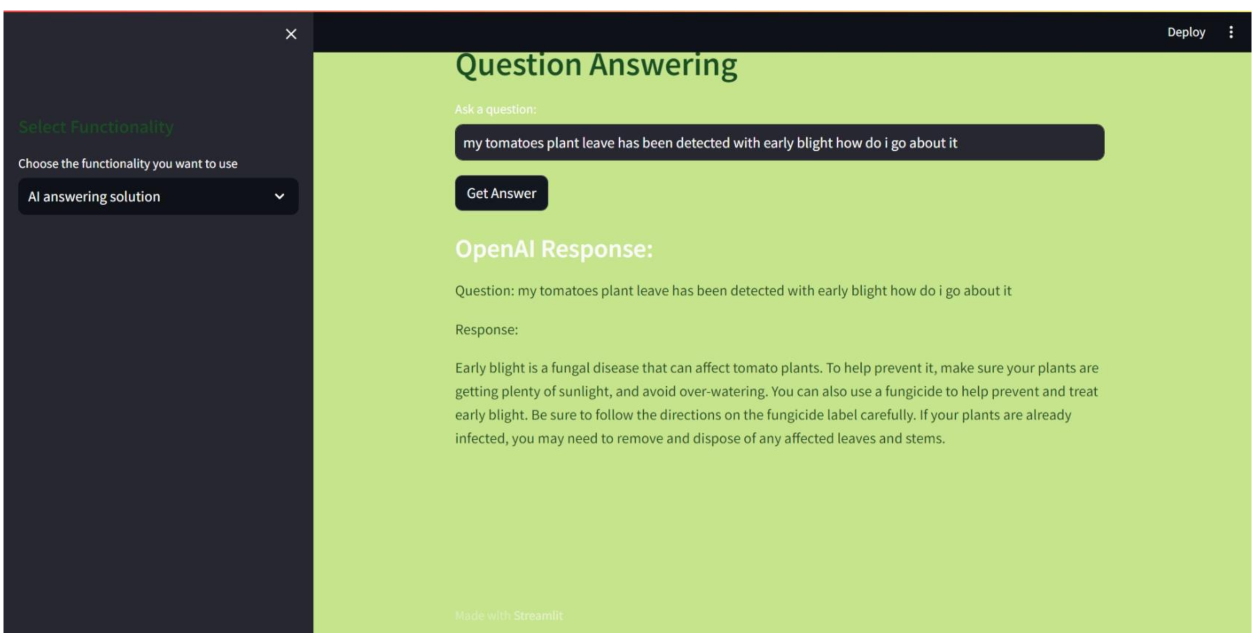


FIGURE 4. 18 outcome of the chatbot page

- Users can study in-depth details about the illness and adhere to advice.

DialogueFlow question and answer bot

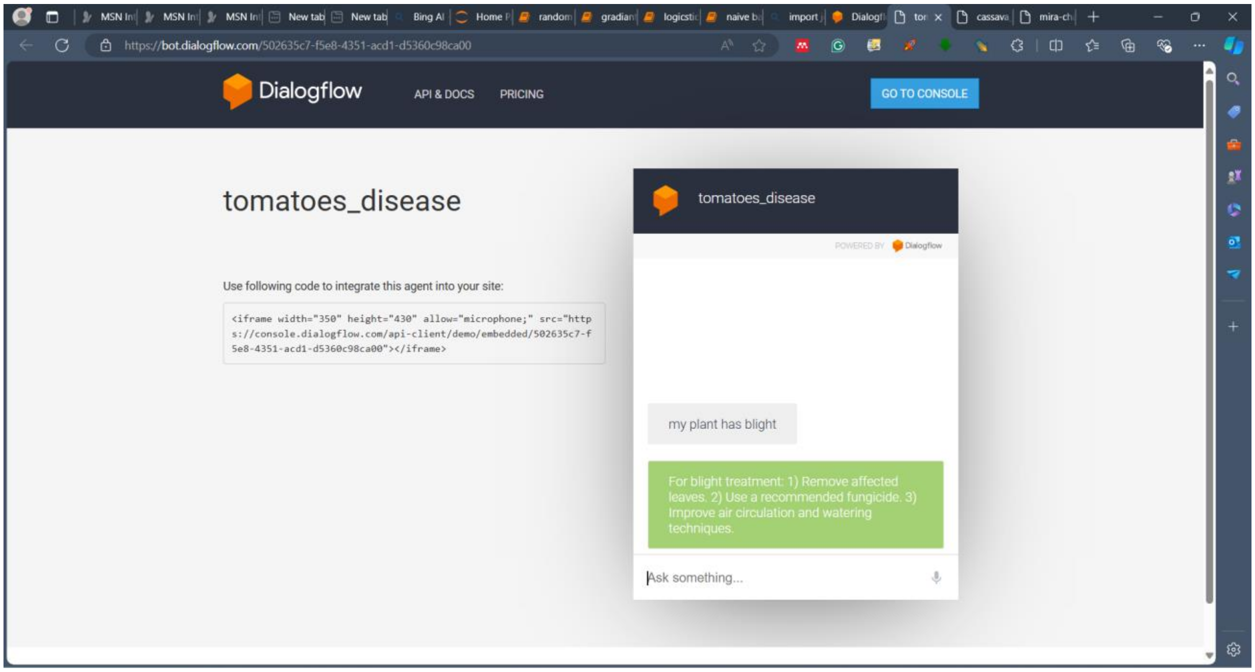


FIGURE 4. 19 dialogue flow page

8. User Feedback and Interaction: - Users can ask questions, leave comments, or look for more details about the illness and its treatments through the system.

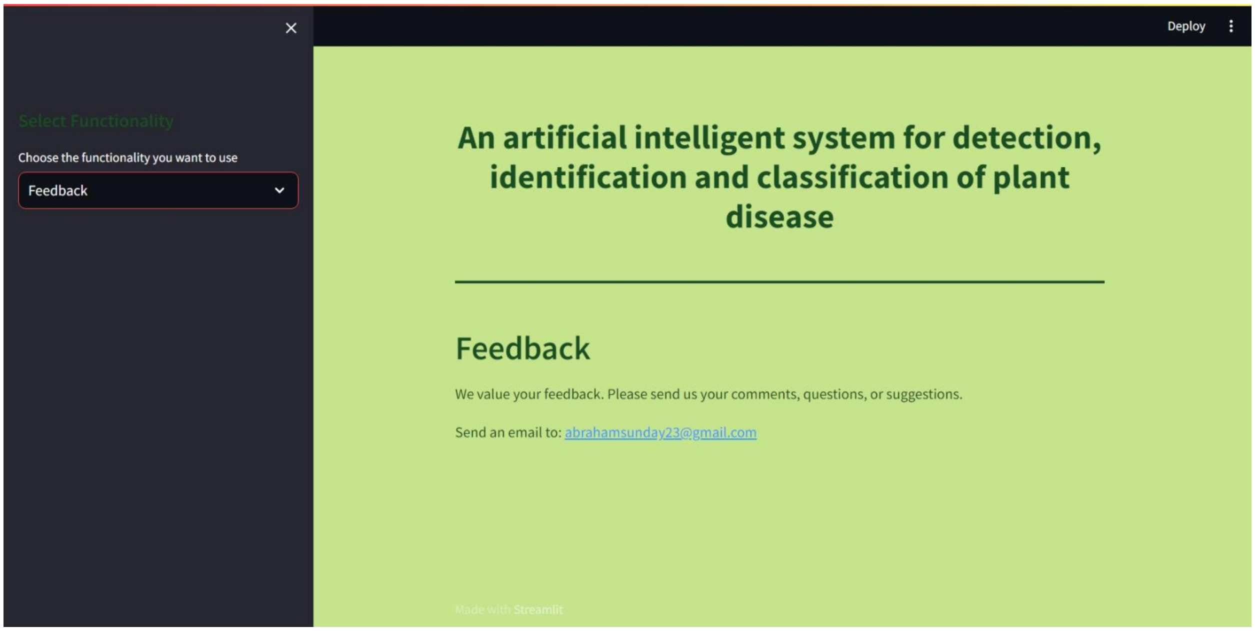


FIGURE 4. 20 feedback on event to email

Interactive feedback forms and chat interfaces increase user engagement.

9. Accessibility and Responsiveness: - The web application is made to work well on a range of gadgets, including tablets, smartphones, laptops, and desktop computers.

10. End User Session: - After the conversation is over, users can wrap up, having learned a lot about plant diseases and how to treat them.

11. Continuous Improvement: - To guarantee adaptability and continuous improvement, the system can be updated regularly with new disease models, improved CNN architectures, and the most recent findings from OpenAI.

Modern machine learning and artificial intelligence technologies are combined in the general operation of the Plant Disease Detection and Solution System web application to provide a user-centric approach to plant disease management. The combination of OpenAI for individualized, real-time solutions and CNN models for precise disease classification results in a novel and efficient system for farmers and agriculture enthusiasts.

MOBILE APP BUILD AND DEV

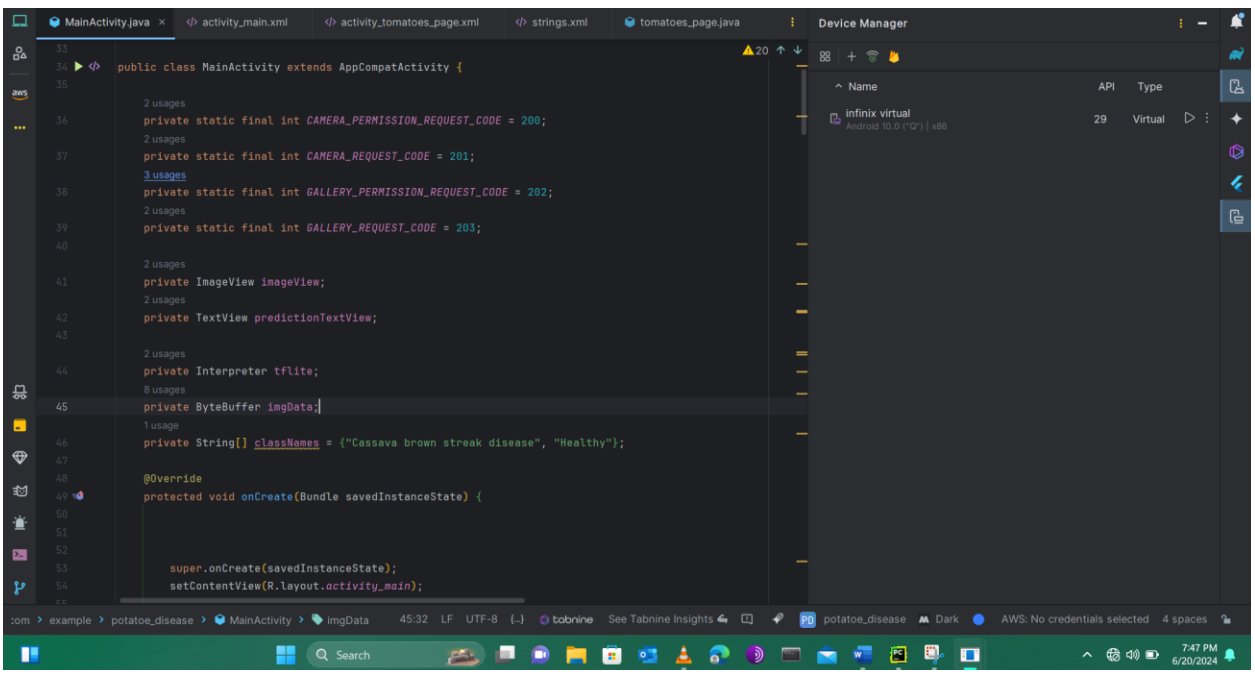


FIGURE 4. 21 mobile app code page

WORKING OF MOBILE APP

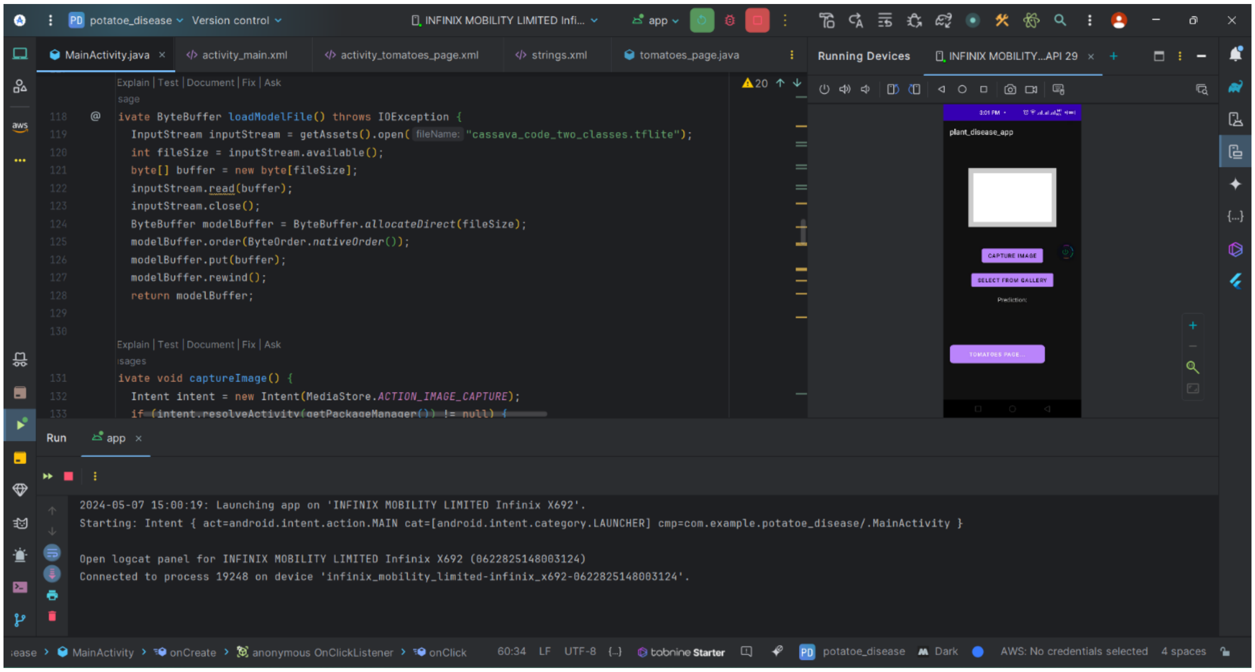


FIGURE 4. 22 more code for mobile app

APP VIEW

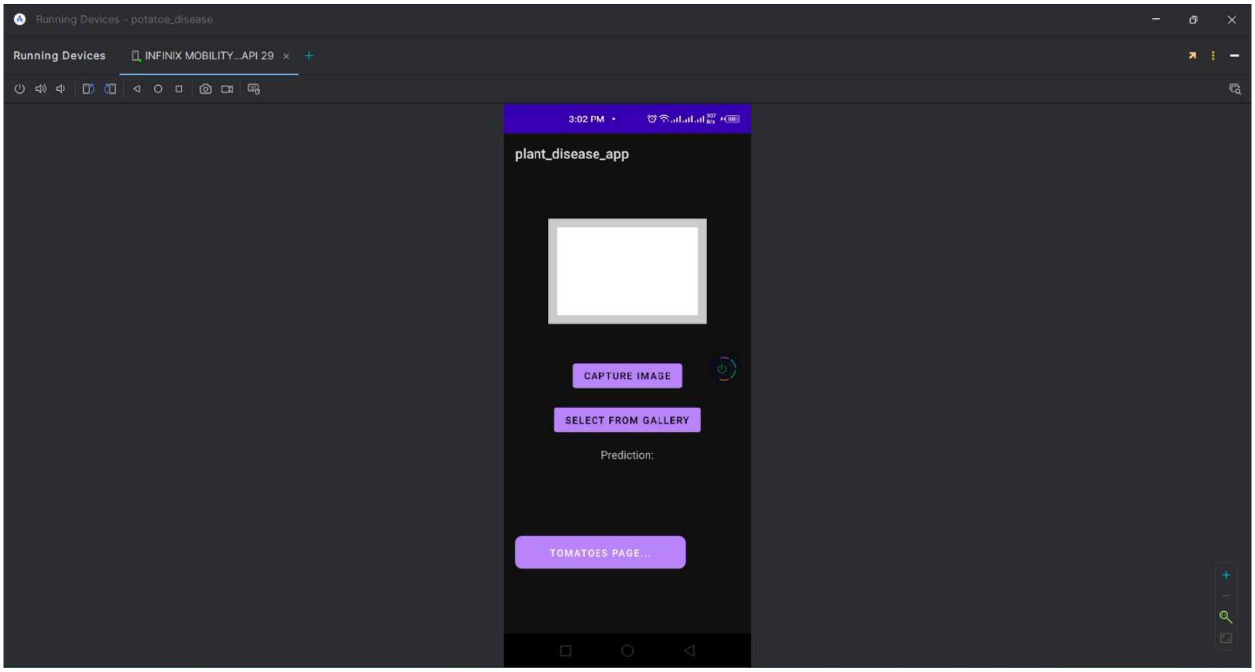


FIGURE 4. 23 mobile app

CHAPTER 5: SUMMARY, CONCLUSION, AND RECOMMENDATIONS

5.1 SUMMARY

To sum up, the Plant Disease Detection and Solution System is an innovative approach that makes use of state-of-the-art technologies to revolutionize agriculture. The goal of this project is to enhance farmers' ability to make decisions and provide them with prompt answers to the urgent issue of crop loss brought on by delayed disease identification. Plant diseases can be promptly and precisely recognized with the use of an integrated, specially constructed Convolutional Neural Network (CNN) model.

The following are the project's main accomplishments and highlights:

Personalized CNN Model the development and use of a unique CNN model intended to recognize plant diseases in a range of plant species, including cassava and tomatoes.

Comparing Performance: an extensive comparison of the performance of the custom-built CNN model with existing models, like those from Google and other well-known architectures. Metrics like as F1 score, recall, accuracy, and precision provide insight into the effectiveness of the proposed model.

DialogueFlow Integration is a novel application of OpenAI that offers interactive, instantaneous remedies for plant problems detected. This feature distinguishes the system by providing not only identification but also actionable insights and recommendations.

Interface for Web Applications: An intuitive and user-friendly web application was developed using Streamlit to enable users to upload or take photos of plants, receive findings for the classification of diseases, and access personalized remedies.

Interactive User Engagement: In an attempt to bridge the knowledge gap in agriculture, the system promotes interactive user engagement. Users can ask questions, find information, and receive prompt responses using a chat interface, all of which contribute to the development of an informed farming community.

Area of Continuous Improvement: Future upgrades and improvements are made possible by the system's scalability. Regular upgrades, new disease models, and CNN architectural enhancements ensure the system's ongoing usefulness and effectiveness.

To sum up, the Plant Disease Detection and Solution System provides proof that artificial intelligence and machine learning have the power to completely transform traditional agricultural practices. Farmers will have a proactive and knowledgeable tool with this system to help lessen crop losses, ease food scarcity, and enhance their financial status. The project is a great asset in the precision agriculture space because of its innovative methodology and commitment to continuous improvement.

5.2 CONCLUSION

With the creation of the Plant Disease Detection and Solution System, agricultural technology enters a new era marked by creative solutions for timely disease detection and management. This section wraps up the project by summarising the main findings, successes, and the system's overall effects on the agriculture industry.

Important Findings:

One notable technological advancement is: The application of artificial intelligence, in particular, the Convolutional Neural Network (CNN) model that was specially constructed, represents a noteworthy technological breakthrough in the field of plant disease detection. When comparing this technology to conventional methods, accuracy and efficiency are increased significantly.

OpenAI Offers Real-Time Solutions, This project is unique because it uses OpenAI to provide real-time, customized solutions in addition to disease identification. This dynamic feature increases the system's usefulness by providing farmers with insights they can use right away to manage diseases.

DialogueFlow: Also acts as an interactive NLP chatbot system that provides rapid solutions as well for the users.

Empowering Farmers, The project has succeeded in its main objective of providing farmers with the information and resources they need to manage diseases effectively. The system connects technology and conventional farming methods with its user-friendly web application and interactive interfaces.

Performance Evaluation and Comparison, The system's dependability and efficacy are shown by the careful assessment and comparison of the specially created CNN model with current models. Performance measures like recall, accuracy, precision, and F1 score offer a thorough evaluation of the model's abilities.

User-Centric Design, By utilizing Streamlit to create an intuitive online application, farmers and agriculture enthusiasts can be assured of accessibility and ease of use. The interactive elements of the system, such as chat interfaces for queries and comments, support a user-centric design.

Overall Effect:

With its combination of artificial intelligence and cutting-edge machine learning models, the Plant Disease Detection and Solution System has the potential to completely transform farming methods. Through the provision of tailored solutions, early disease detection, and knowledge-sharing, the system seeks to:

Reduce the amount of crop losses: The system helps to maximize crop yield and minimize crop losses by quickly addressing diseases.

Reduce Food Scarcity: Preventive illness care helps to keep the food supply steady and adequate.

Improve Financial Position: Giving farmers the information and tools they need improves their financial situation and promotes environmentally friendly farming methods.

Goals for the Future:

The system is designed to be continuously improved as technology advances. Among the future paths are:

-Expansion to Additional Crops: Increase the system's capacity to handle a wider variety of crops, catering to the various demands of agricultural practitioners.

Integration of Advanced AI Techniques: Use cutting-edge AI methods to identify diseases and create solutions even more precisely.

Global Accessibility: Taking into account the various agricultural landscapes across the world, strive to make the system globally accessible.

To sum up, the Plant Disease Detection and Solution System represents a critical turning point in the relationship between technology and agriculture. Its effects on food security, agricultural sustainability, and farmers' well-being highlight how revolutionary artificial intelligence can be when it comes to solving problems in the real world.

5.3 Contribution To Knowledge

The Plant Disease Detection and Solution System contributes significantly to the body of knowledge in several key areas, pushing the boundaries of agricultural technology and artificial intelligence. This section outlines the notable contributions made by the project:

Advancements in Disease Detection Technology:

- The project contributes to the advancement of disease detection technology by introducing a custom-built Convolutional Neural Network (CNN) model tailored for plant diseases. The model's architecture and training techniques offer insights into optimizing machine learning models for specific agricultural applications.

Intelligent Utilisation of OpenAI for Instantaneous Resolutions: - An innovative step towards offering interactive, real-time solutions for detected plant diseases is the integration of OpenAI. This method broadens the application of disease management systems by adding a dynamic component that offers tailored recommendations and answers user inquiries.

Utilization of NLP chatbot system for interactive solutions(dialogue flow).

Model Comparison and Performance Evaluation Metrics: Through a methodical assessment of the custom-built CNN model and a comparison with current models, performance metrics in disease detection are better understood. Metrics like accuracy, precision, recall, and F1 score are added to improve the process of evaluating how well machine learning models work in agricultural settings.

User-Centric Design and Interactive Features: - By combining interactive elements like chat interfaces and feedback mechanisms with an emphasis on developing a user-friendly web application using Streamlit, this project advances our understanding of how to create accessible and interesting interfaces for farmers. This user-centered strategy enhances user interaction and promotes knowledge sharing.

Agricultural and Artificial Intelligence Bridge: - By bridging the gap between artificial intelligence and agriculture, the project demonstrates how cutting-edge technologies can be used to solve practical problems that farmers face. The system's use shows that integrating AI into conventional farming methods is feasible.

Potential for Worldwide Agricultural Impact: - The system's adaptability and scalability create the foundation for possible worldwide agricultural influence. According to the project, comparable AI-driven solutions can be modified to fit various agricultural landscapes across the globe, promoting effective and sustainable farming methods.

Agricultural Systems' Continuous Improvement Framework: - The project presents a framework for continuous improvement, highlighting the significance of frequent updates, model improvements, and technology adaptation. This framework establishes a standard for how agricultural systems should change in response to new technological developments.

In summary, the Plant Disease Detection and Solution System advances knowledge by addressing the wider ramifications of incorporating artificial intelligence into agriculture in addition to presenting cutting-edge technologies for disease detection. The project's methods and results open up new avenues for research and development at the nexus of sustainable farming techniques and technology.

5.4 Recommendations

Based on the implementation, results, and insights gained from the Plant Disease Detection and Solution System, the following recommendations are proposed for consideration:

Continuous Model Improvement: - Put in place a methodical procedure for the machine learning models' ongoing development. Update the Convolutional Neural Network (CNN) model regularly with new datasets to make sure it can adapt to changing disease patterns.

Incorporate Additional Plant Species: - Add more plant species to the system to increase its functionality. Think about expanding the CNN model to accommodate a larger range of crops, taking into account the varied agricultural environment and meeting the requirements of farmers growing a variety of plants.

Collaboration with Agricultural Experts: Encourage partnerships with extension services, researchers, and agricultural experts. Take part in knowledge-sharing programs to improve the accuracy and applicability of the system. Working together can yield important insights into the differences in disease patterns between regions.

Improve OpenAI Integration: - Look into ways to improve the OpenAI integration. Examine integrating cutting-edge natural language processing (NLP) methods to enhance the caliber and variety of answers the system's question-answering interface provides.

User Education and Outreach Programmes: - Introduce the system to farmers by starting user education and outreach initiatives. Organize workshops, awareness campaigns, and training sessions to make sure users are aware of the features and advantages of the Plant Disease Detection and Solution System.

Global Accessibility and Multilingual Support: - Try to add multilingual support to the system to make it globally accessible. To accommodate users from around the world, think about translating the web application interface and solution responses into several languages.

Community Feedback Mechanism: - Create a web application that receives feedback from the community. Invite users to share their opinions about the overall user experience, efficacy of treatments, and disease detection accuracy. This feedback loop may help with continuous system enhancements.

Security and Privacy Measures: - Give priority to implementing robust security and privacy measures. When handling sensitive agricultural data and user-uploaded images, the system should employ encryption protocols to safeguard user information. Regulations about data protection must also be observed.

Integration with Technologies for Precision Agriculture: - Look into ways to incorporate technologies for precision agriculture. Consider merging sensor, drone, and other Internet of Things device data to enhance the system's ability to monitor and make agricultural decisions in real-time.

News on Technologies Associated with AI: - Follow the most recent advancements in artificial intelligence. Invest in studies that look into the application of cutting-edge AI techniques, like reinforcement learning and generative adversarial networks, for the detection and management of plant diseases.

REFERENCES

O’Brien, M. J., & Rich, A. E. (1976, January 1). *Potato Diseases*.

Zadoks, J. (1996,). Crop production and crop protection: Estimated losses in major food and cash crops. *Agricultural Systems*, *51*(4), 493–495. https://doi.org/10.1016/0308-521x(96)81487-9

Plant Disease Prediction using Convolutional Neural Network. (2020, December 23). *Journal of Xidian University*, *14*(12). https://doi.org/10.37896/jxu14.12/044

Savary, S., Willocquet, L., Pethybridge, S. J., Esker, P., McRoberts, N., & Nelson, A. (2019, February 4). The global burden of pathogens and pests on major food crops. *Nature Ecology & Evolution*, *3*(3), 430–439. https://doi.org/10.1038/s41559-018-0793-y

Reiter, E., & Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning, second edition*. MIT Press.

Jurafsky, D., & Martin, J. H. (2000). *Speech and Language Processing*.

M. Mounika, L.Sahithi, K.Prasanna Lakshmi, K.Praveenya, & N. Ashok Kumar. (2023, November 1). A CNN CLASSİFİCATİON APPROACH FOR POTATO PLANT LEAF DİSEASE DETECTİON. *EPRA International Journal of Research & Development (IJRD)*, 277–281. https://doi.org/10.36713/epra14773

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. “O’Reilly Media, Inc.”

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An Introduction to Statistical Learning*.

Zhang, N., Wu, H., Zhu, H., Deng, Y., & Han, X. (2022). Tomato Disease Classification and Identification Method Based on Multimodal Fusion Deep Learning. *Agriculture*, *12*(12), 2014. https://doi.org/10.3390/agriculture12122014

Domingues, T., Brandão, T., & Ferreira, J. C. (2022). Machine Learning for Detection and Prediction of Crop Diseases and Pests: A Comprehensive Survey. *Agriculture*, *12*(9), 1350. https://doi.org/10.3390/agriculture12091350

Hassanien, A. E., Tolba, M., & Azar, A. T. (2014). *Advanced Machine Learning Technologies and Applications*. Springer.

Faid, A., Sadik, M., & Sabir, E. (2021). An Agile AI and IoT-Augmented Smart Farming: A Cost-Effective Cognitive Weather Station. *Agriculture*, *12*(1), 35. https://doi.org/10.3390/agriculture12010035

Said Mohamed, E., Belal, A., Kotb Abd-Elmabod, S., El-Shirbeny, M. A., Gad, A., & Zahran, M. B. (2021, December). Smart farming for improving agricultural management. *The Egyptian Journal of Remote Sensing and Space Science*, *24*(3), 971–981. https://doi.org/10.1016/j.ejrs.2021.08.007

Niu, B., Wang, W., Yuan, Z., Sederoff, R. R., Sederoff, H., Chiang, V. L., & Borriss, R. (2020, October 9). Microbial Interactions Within Multiple-Strain Biological Control Agents Impact Soil-Borne Plant Disease. *Frontiers in Microbiology*, *11*. https://doi.org/10.3389/fmicb.2020.585404

D.Pujari, J., Yakkundimath, R., & Byadgi, A. S. (2016). SVM and ANN Based Classification of Plant Diseases Using Feature Reduction Technique. *International Journal of Interactive Multimedia and Artificial Intelligence*, *3*(7), 6. https://doi.org/10.9781/ijimai.2016.371

Maeda-Gutiérrez, V., Galván-Tejada, C. E., Zanella-Calzada, L. A., Celaya-Padilla, J. M., Galván-Tejada, J. I., Gamboa-Rosales, H., Luna-García, H., Magallanes-Quintanar, R., Guerrero Méndez, C. A., & Olvera-Olvera, C. A. (2020). Comparison of Convolutional Neural Network Architectures for Classification of Tomato Plant Diseases. *Applied Sciences*, *10*(4), 1245. https://doi.org/10.3390/app10041245

Boulent, J., Foucher, S., Théau, J., & St-Charles, P. L. (2019). Convolutional Neural Networks for the Automatic Identification of Plant Diseases. *Frontiers in Plant Science*, *10*. https://doi.org/10.3389/fpls.2019.00941

Saleem, Potgieter, & Mahmood Arif. (2019). Plant Disease Detection and Classification by Deep Learning. *Plants*, *8*(11), 468. https://doi.org/10.3390/plants8110468

Ullah, Z., Alsubaie, N., Jamjoom, M., Alajmani, S. H., & Saleem, F. (2023). EffiMob-Net: A Deep Learning-Based Hybrid Model for Detection and Identification of Tomato Diseases Using Leaf Images. *Agriculture*, *13*(3), 737. https://doi.org/10.3390/agriculture13030737

GeeksforGeeks. (2022, May 25). *How to split a Dataset into Train and Test Sets using Python*. GeeksforGeeks. https://www.geeksforgeeks.org/how-to-split-a-dataset-into-train-and-test-sets-using-python/

Gillis, A. S. (2024). *Data splitting*. Enterprise AI. https://www.techtarget.com/searchenterpriseai/definition/data-splitting#:~:text=Data%20splitting%20is%20when%20data,creating%20models%20based%20on%20data.

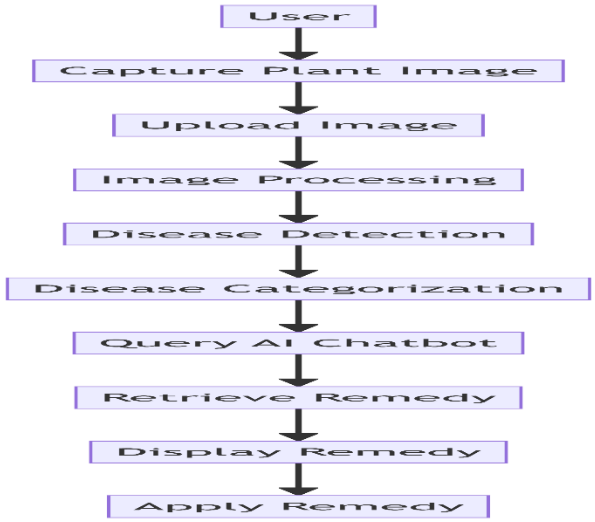
L, G. D. (2023). Five Methods for Data Splitting in Machine Learning. *Medium*. https://medium.com/@tubelwj/five-methods-for-data-splitting-in-machine-learning-27baa50908ed

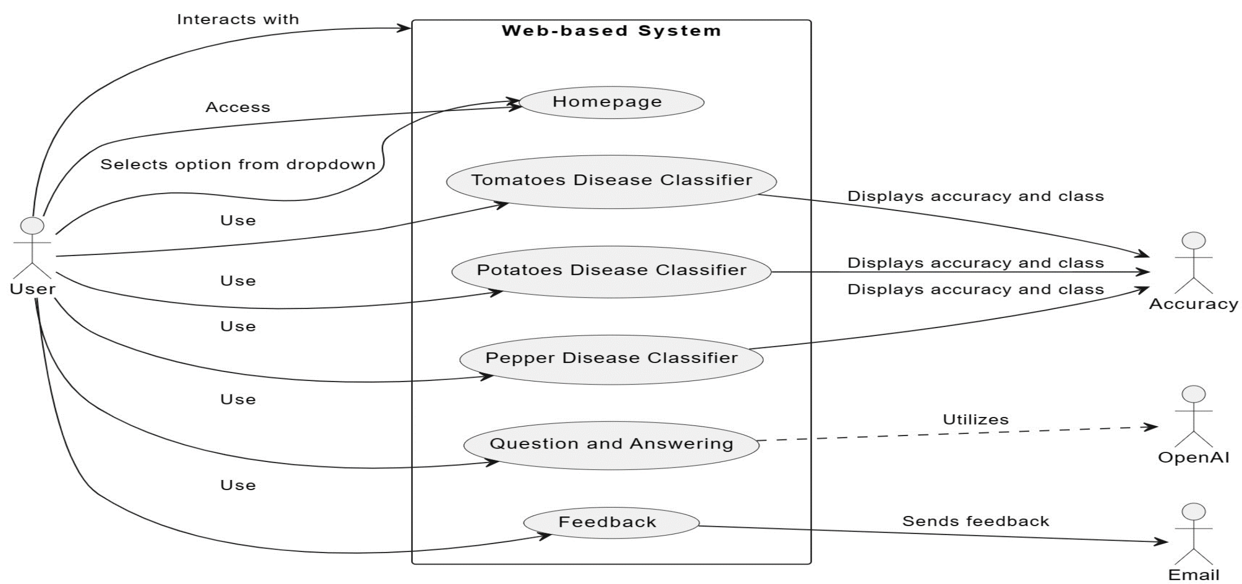
Menon, K. (2024). *Feature Selection In Machine Learning: All You Need to Know*. Simplilearn.com. https://www.simplilearn.com/tutorials/machine-learning-tutorial/feature-selection-in-machine-learning#:~:text=ProgramExplore%20Program-,What%20is%20Feature%20Selection%3F,you%20are%20trying%20to%20solve.

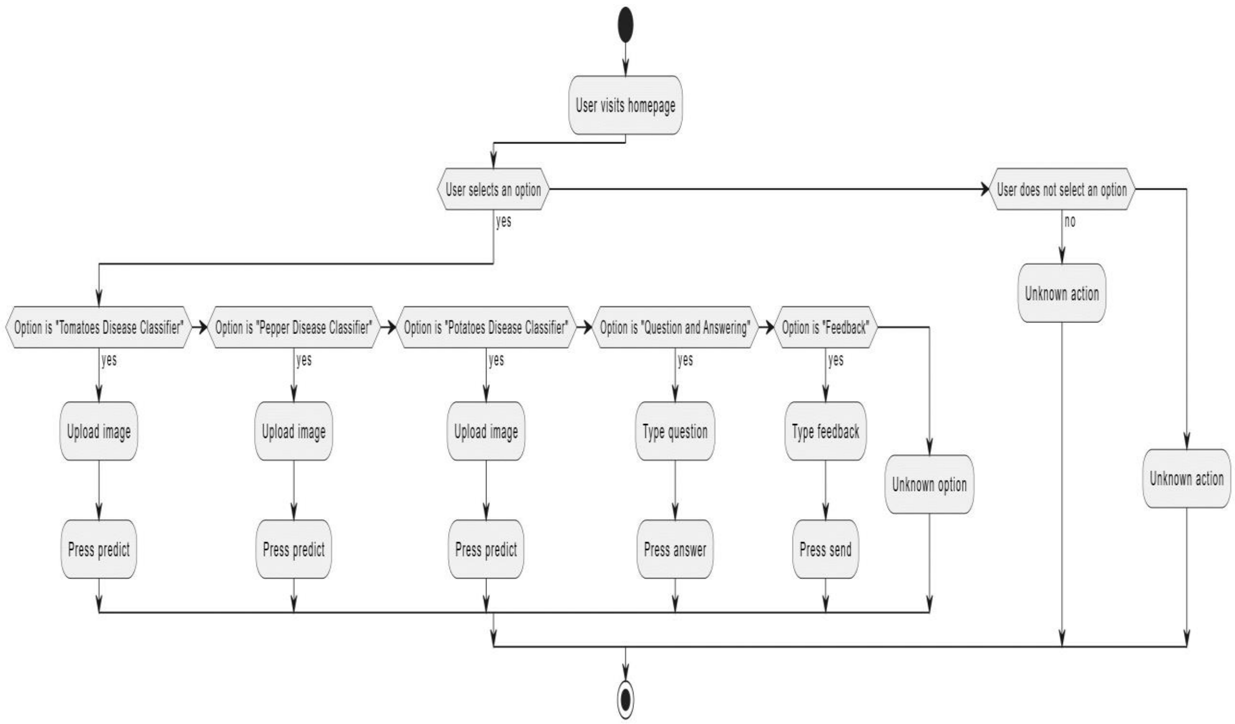
What is Label Encoding in Python | Great Learning? (2024). Great Learning Blog: Free Resources What Matters to Shape Your Career! https://www.mygreatlearning.com/blog/label-encoding-in-python/#:~:text=Label%20encoding%20is%20a%20technique,only%20operate%20on%20numerical%20data.

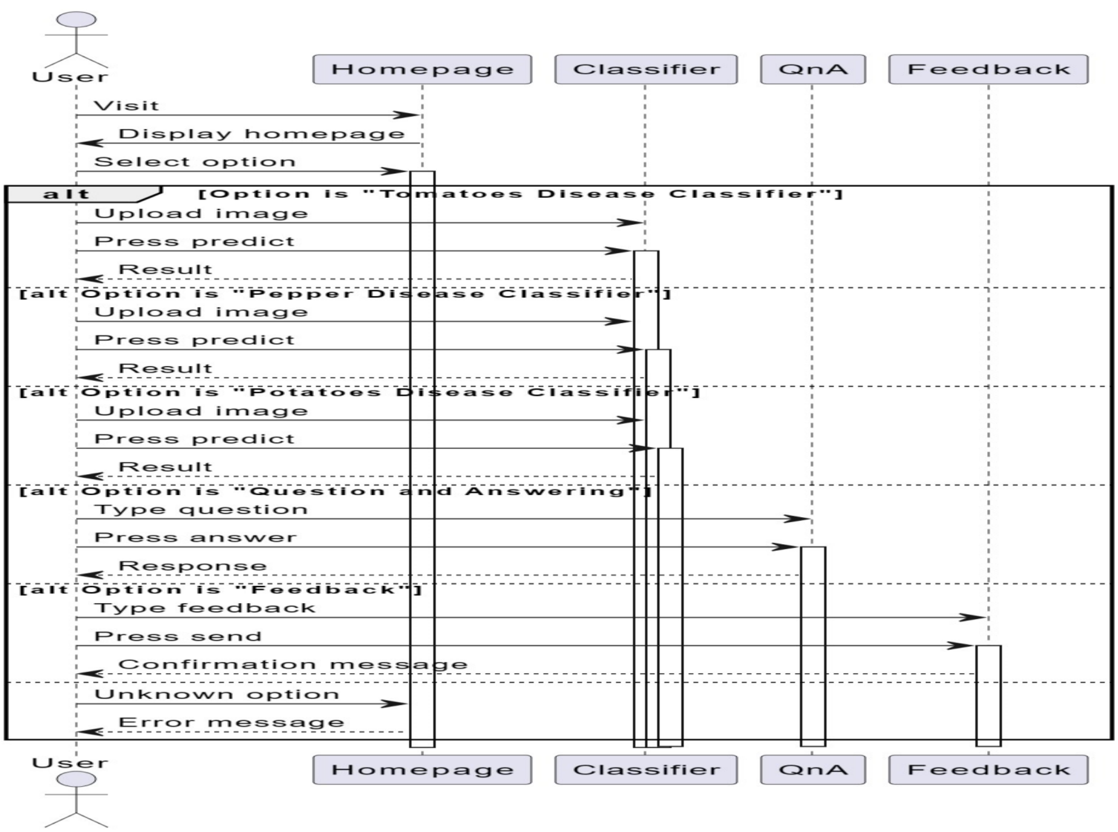
APPENDIX

APPENDIX A






User visits homepage
User selects an option
yes
User does not select an option
no
Unknown action
Option is "Tomatoes Disease Classifier"
Option is "Pepper Disease Classifier"
Option is "Potatoes Disease Classifier"
Option is "Question and Answering"
Option is "Feedback"
yes
Upload image
Type question
Type feedback
Unknown option
Unknown action
Press predict
Press answer
Press send


User
Homepage
Classifier
QnA
Feedback
Visit
Display homepage
Select option
alt [Option is "Tomatoes Disease Classifier"]
Upload image
Press predict
Result
[alt Option is "Pepper Disease Classifier"]
Upload image
Press predict
Result
[alt Option is "Potatoes Disease Classifier"]
Upload image
Press predict
Result
[alt Option is "Question and Answering"]
Type question
Press answer
Response
[alt Option is "Feedback"]
Type feedback
Press send
Confirmation message
Unknown option
Error message

APPENDIX B

```python
resize_and_rescale = tf.keras.Sequential([
    layers.experimental.preprocessing.Resizing(IMAGE_SIZE, IMAGE_SIZE),
    layers.experimental.preprocessing.Rescaling(1./255),
])


data_augmentation = tf.keras.Sequential([
    layers.experimental.preprocessing.RandomFlip("horizontal_and_vertical"),
    layers.experimental.preprocessing.RandomRotation(0.2),
    layers.experimental.preprocessing.RandomZoom(0.2),
    layers.experimental.preprocessing.RandomContrast(0.2),
    layers.experimental.preprocessing.RandomTranslation(0.2, 0.2),
])
```

B

```python
input_shape = (BATCH_SIZE, IMAGE_SIZE, IMAGE_SIZE, CHANNELS)
n_classes = 2

model = models.Sequential([
    resize_and_rescale,
    layers.Conv2D(32, kernel_size = (3,3), activation='relu', input_shape=input_shape),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64,  kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64,  kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(n_classes, activation='softmax'),
])
```

```
history = model.fit(
    train_ds,
    batch_size=BATCH_SIZE,
    validation_data=val_ds,
    verbose=1,
    epochs=50,
)
```

```
Epoch 1/50
12/12 [==============================] - 57s 3s/step - loss: 0.6770 - accuracy: 0.6016 - val_loss: 0.6881 - val_accuracy: 0.7188
Epoch 2/50
12/12 [==============================] - 26s 2s/step - loss: 0.6761 - accuracy: 0.6276 - val_loss: 0.6192 - val_accuracy: 0.7188
Epoch 3/50
12/12 [==============================] - 24s 2s/step - loss: 0.6668 - accuracy: 0.6276 - val_loss: 0.6273 - val_accuracy: 0.7188
Epoch 4/50
12/12 [==============================] - 22s 2s/step - loss: 0.6643 - accuracy: 0.6276 - val_loss: 0.6177 - val_accuracy: 0.7188
Epoch 5/50
12/12 [==============================] - 25s 2s/step - loss: 0.6647 - accuracy: 0.6276 - val_loss: 0.6182 - val_accuracy: 0.7188
Epoch 6/50
12/12 [==============================] - 27s 2s/step - loss: 0.6593 - accuracy: 0.6276 - val_loss: 0.5920 - val_accuracy: 0.7188
Epoch 7/50
12/12 [==============================] - 30s 2s/step - loss: 0.6506 - accuracy: 0.6276 - val_loss: 0.5778 - val_accuracy: 0.7188
Epoch 8/50
12/12 [==============================] - 31s 3s/step - loss: 0.6467 - accuracy: 0.6276 - val_loss: 0.5456 - val_accuracy: 0.7188
Epoch 9/50
12/12 [==============================] - 27s 2s/step - loss: 0.6043 - accuracy: 0.6276 - val_loss: 0.4775 - val_accuracy: 0.7188
Epoch 10/50
12/12 [==============================] - 28s 2s/step - loss: 0.5305 - accuracy: 0.7135 - val_loss: 0.4118 - val_accuracy: 0.8125
Epoch 11/50
12/12 [==============================] - 29s 2s/step - loss: 0.5220 - accuracy: 0.7474 - val_loss: 0.3856 - val_accuracy: 0.8750
Epoch 12/50
12/12 [==============================] - 27s 2s/step - loss: 0.4898 - accuracy: 0.8021 - val_loss: 0.3854 - val_accuracy: 0.8438
```

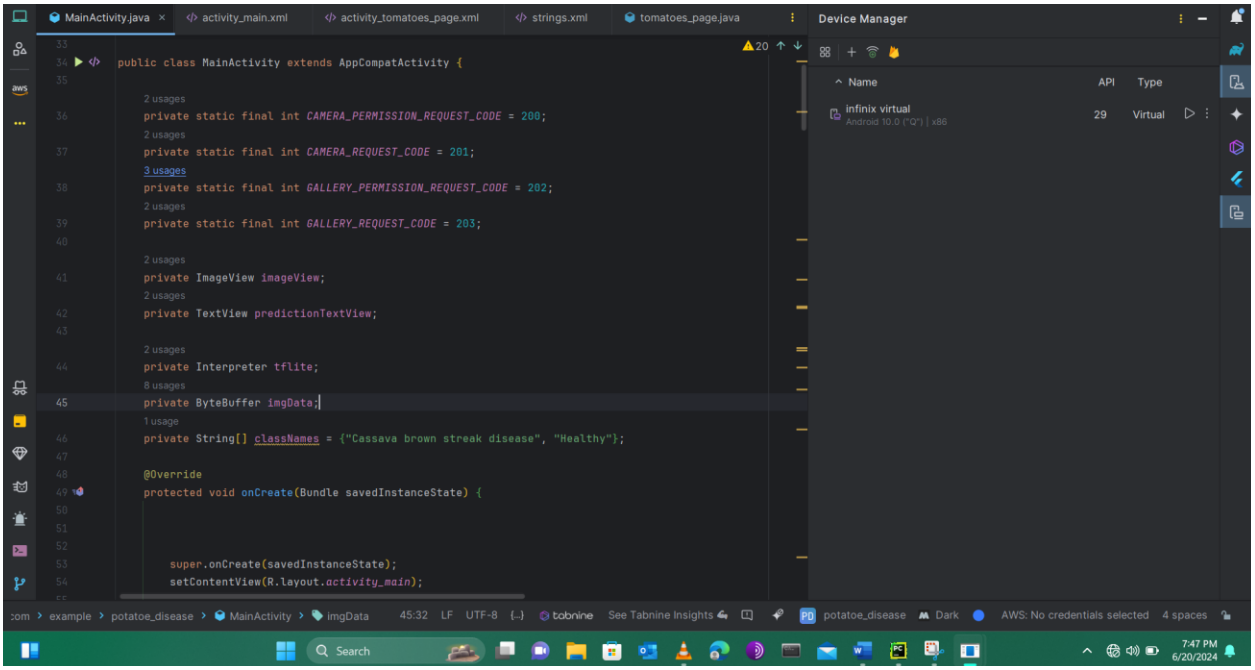
```
scores = model.evaluate(test_ds)
```

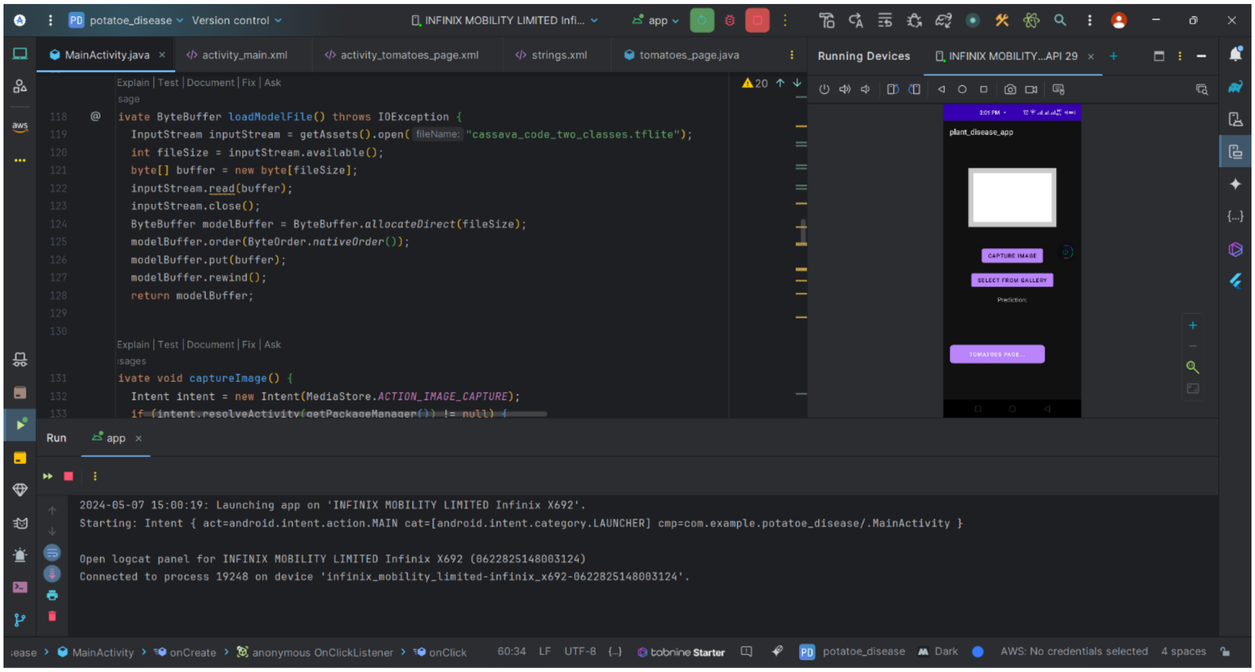
```
3/3 [==============================] - 16s 409ms/step - loss: 0.3088 - accuracy: 0.8646
```

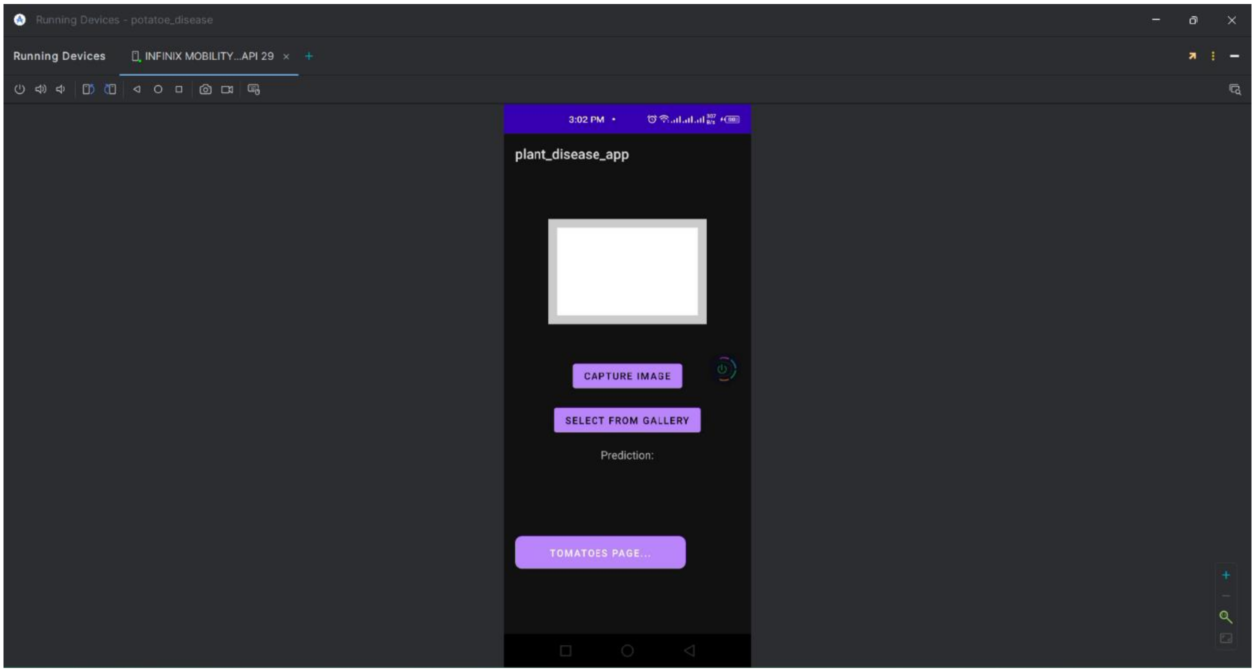
```
import streamlit as st
import tensorflow as tf
import numpy as np
from PIL import Image
import os
from langchain.llms import OpenAI
import requests
from bs4 import BeautifulSoup


# Load your machine learning models and class names here
Codeium: Refactor | Explain | Docstring | X
Explain | Test | Document | Fix | Ask
1 usage   new *
def load_models():
    tomato_model = tf.keras.models.load_model('tomatoes_augmented_two_classes.h5')
    pest_model = tf.keras.models.load_model('pest_detection_model2.h5')
    cassava_model = tf.keras.models.load_model('cassava_code_two_classes.h5')
    return {'tomato': tomato_model,  'pest': pest_model,
            'cassava': cassava_model}

```

```python
def main():
    models = load_models()
    class_names = load_class_names()

    st.sidebar.header("Select Functionality")
    app_mode = st.sidebar.selectbox(
        "Choose the functionality you want to use",
        (
            "Home", "Getting Started",'Cassava Disease Classifier', "Tomato Disease Classifier",  "Pest Detection", "AI answering solution", "Feedback"

        )
    )

    if app_mode == "Home":
        st.title("Farm Management and Plant Health Advisor")
        st.write("Welcome to the Farm Management and Plant Health Advisor!")
        st.write("Select a functionality from the sidebar.")
```

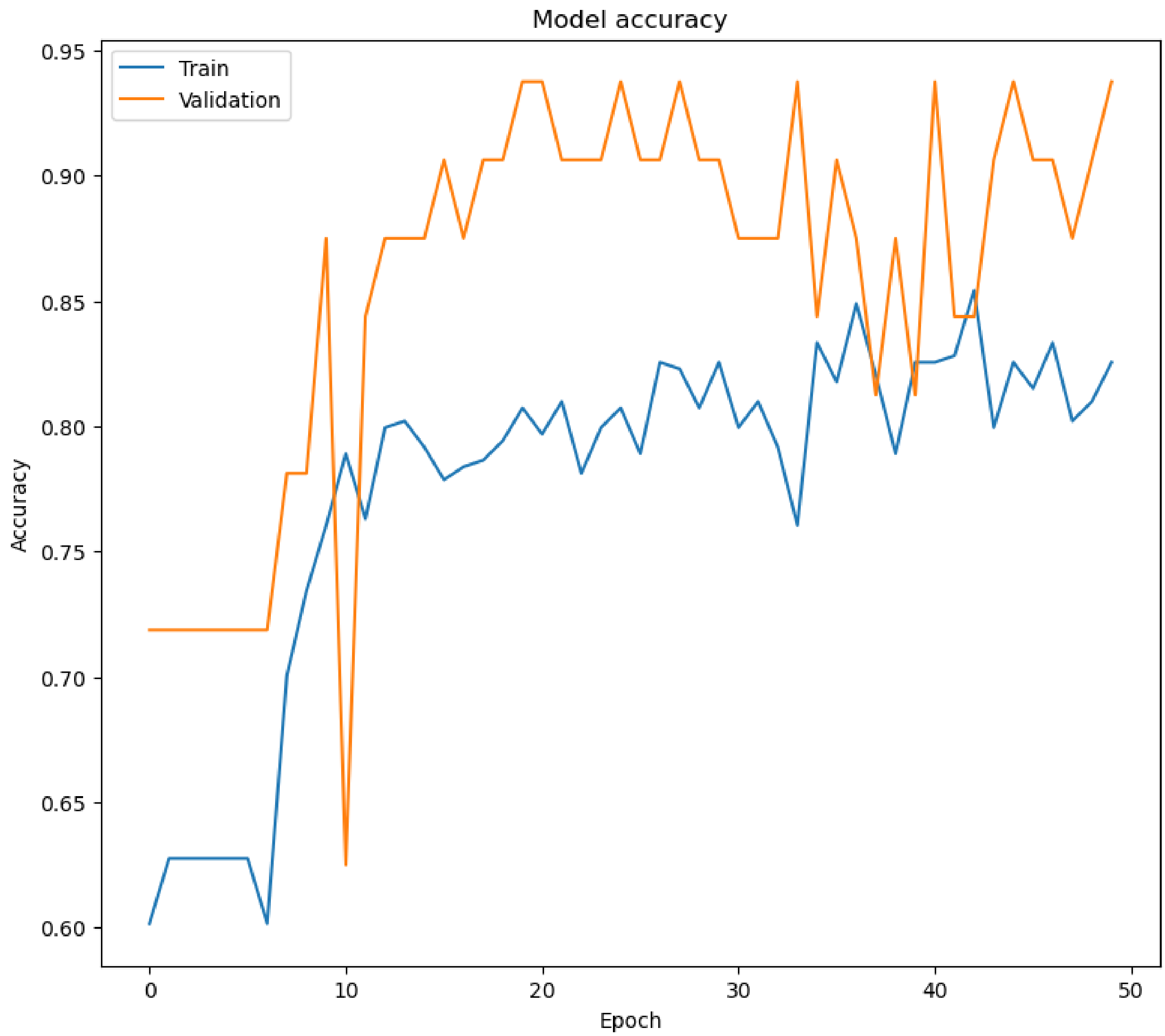
```python
    elif app_mode == "Cassava Disease Classifier":

        st.header("Cassava Disease Classifier")

        uploaded_file = st.file_uploader("Upload an image of a Cassava leaf", type=["

        if uploaded_file is not None:
            image = Image.open(uploaded_file)

            # Check if image has an alpha (transparency) channel
            if image.mode == 'RGBA':
                # If yes, convert it to RGB
                image = image.convert('RGB')

            image = image.resize((256, 256))  # Resize the image to 256x256 pixels

            st.image(image, caption="Uploaded Image", use_column_width=True)

            if st.button("Predict"):
                img = np.array(image)

                img = img / 255.0  # Normalize the image
```



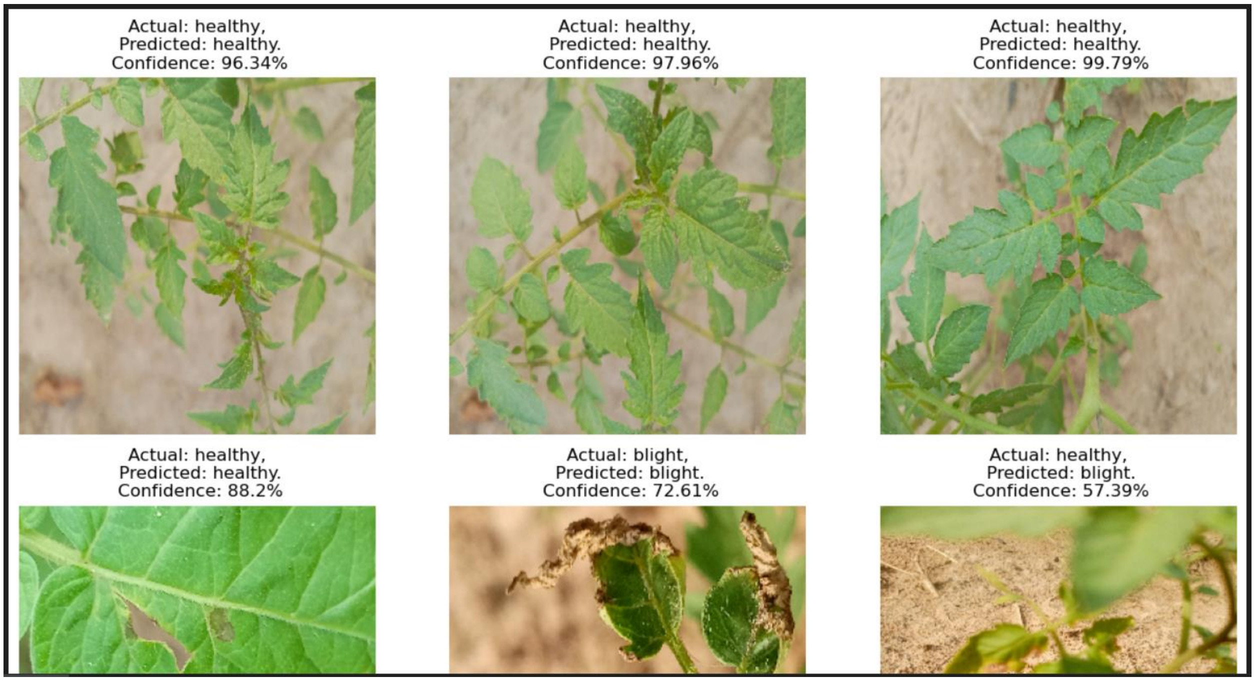


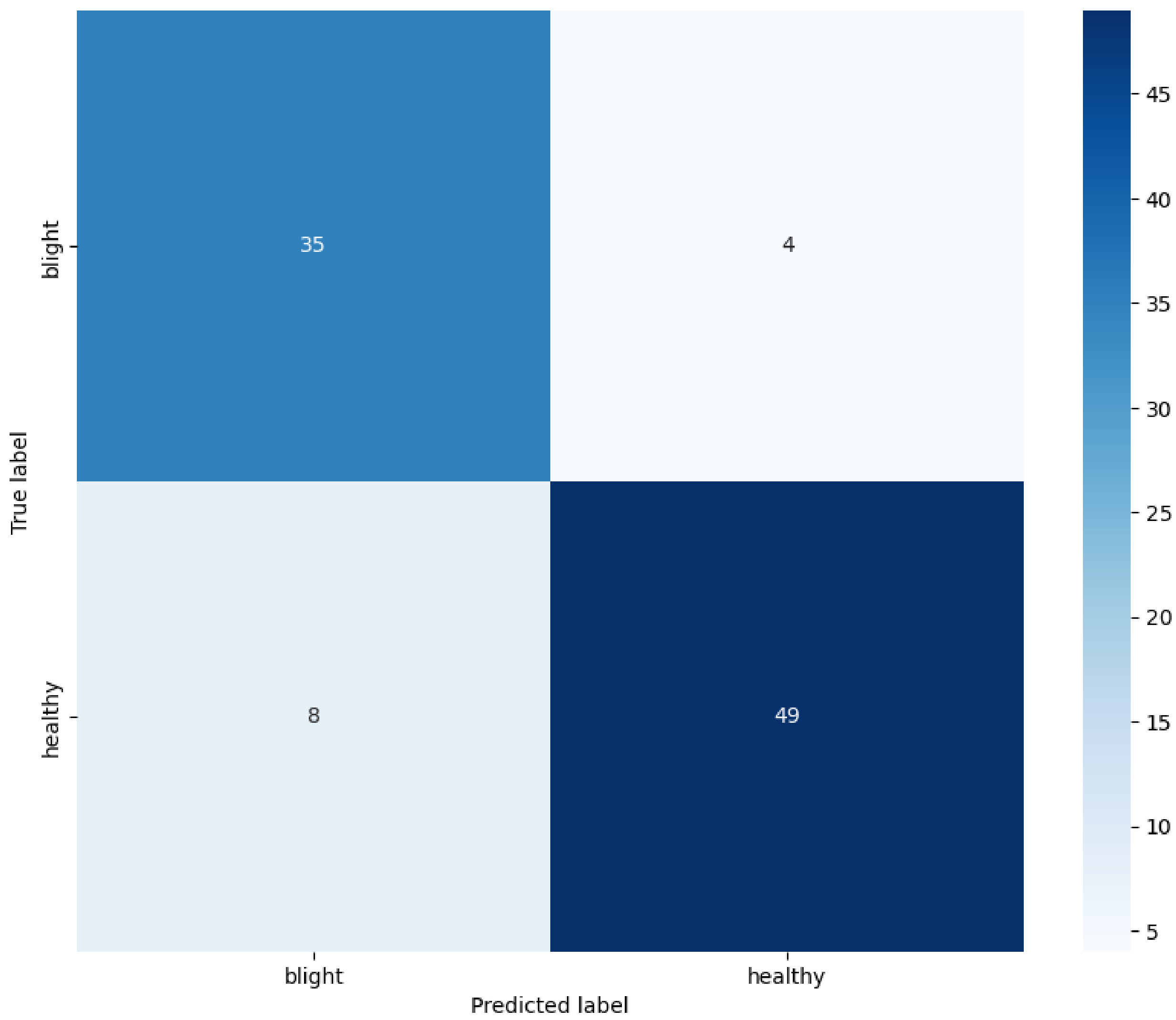

Running Devices - potatoe_disease
Running Devices
INFINIX MOBILITY...API 29
3:02 PM
plant_disease_app
CAPTURE IMAGE
SELECT FROM GALLERY
Prediction:
TOMATOES PAGE...

APPENDIX C

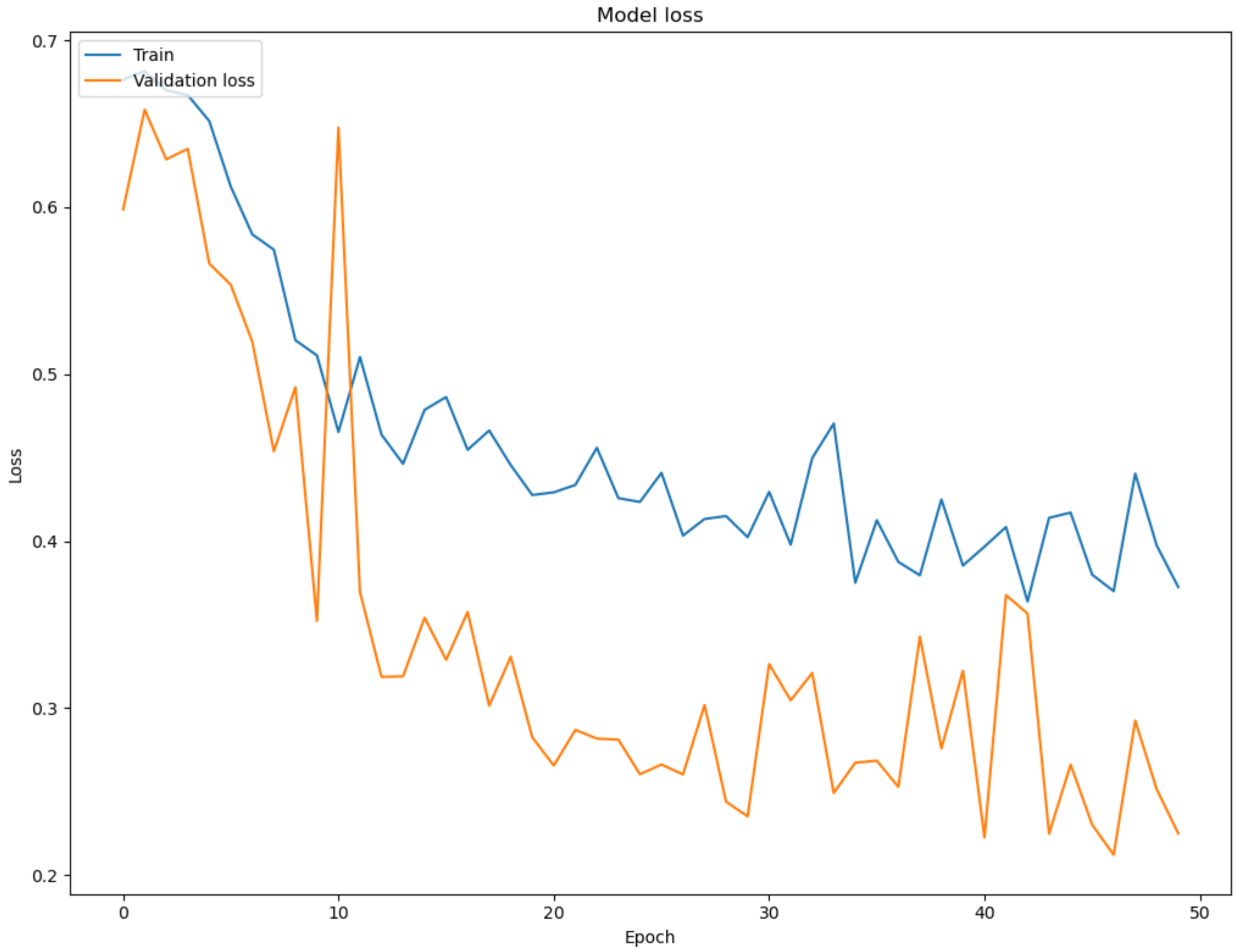


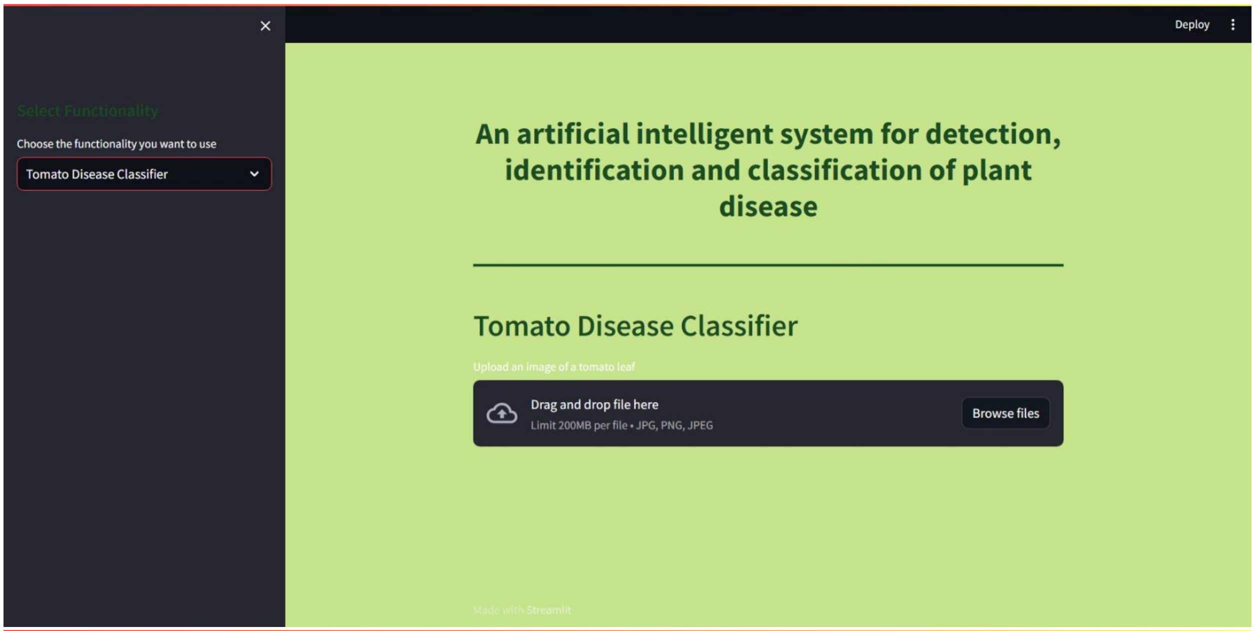
Model accuracy
Train
Validation
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
Accuracy
0
10
20
30
40
50
Epoch

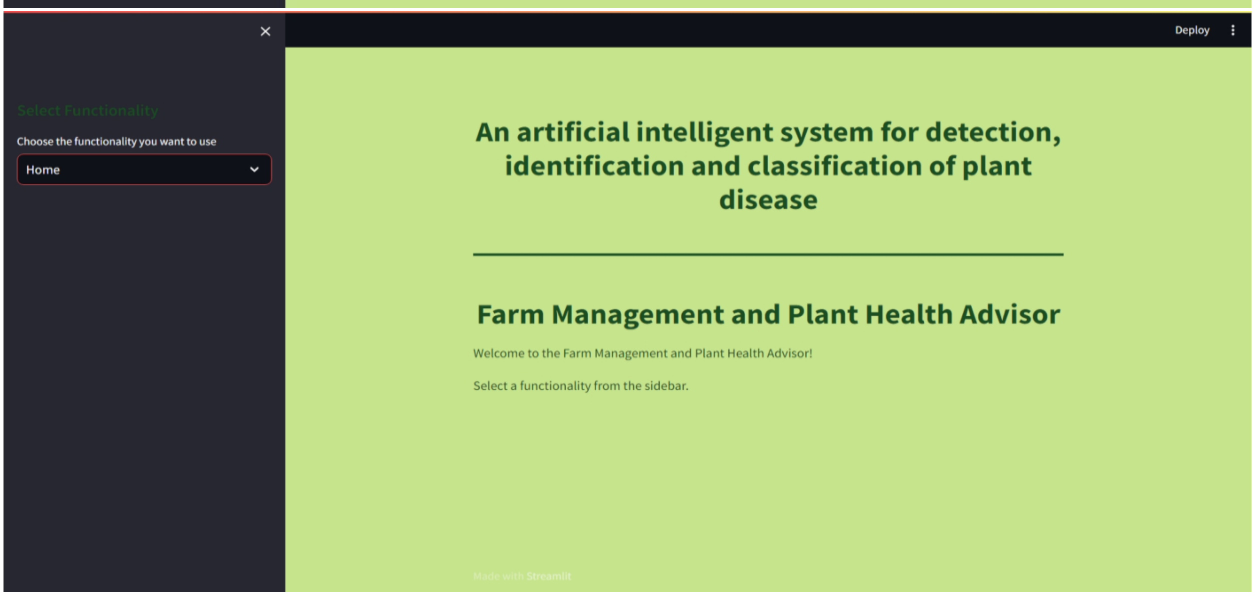

Actual: healthy,
Predicted: healthy.
Confidence: 96.34%
Actual: healthy,
Predicted: healthy.
Confidence: 97.96%
Actual: healthy,
Predicted: healthy.
Confidence: 99.79%
Actual: healthy,
Predicted: healthy.
Confidence: 88.2%
Actual: blight,
Predicted: blight.
Confidence: 72.61%
Actual: healthy,
Predicted: blight.
Confidence: 57.39%

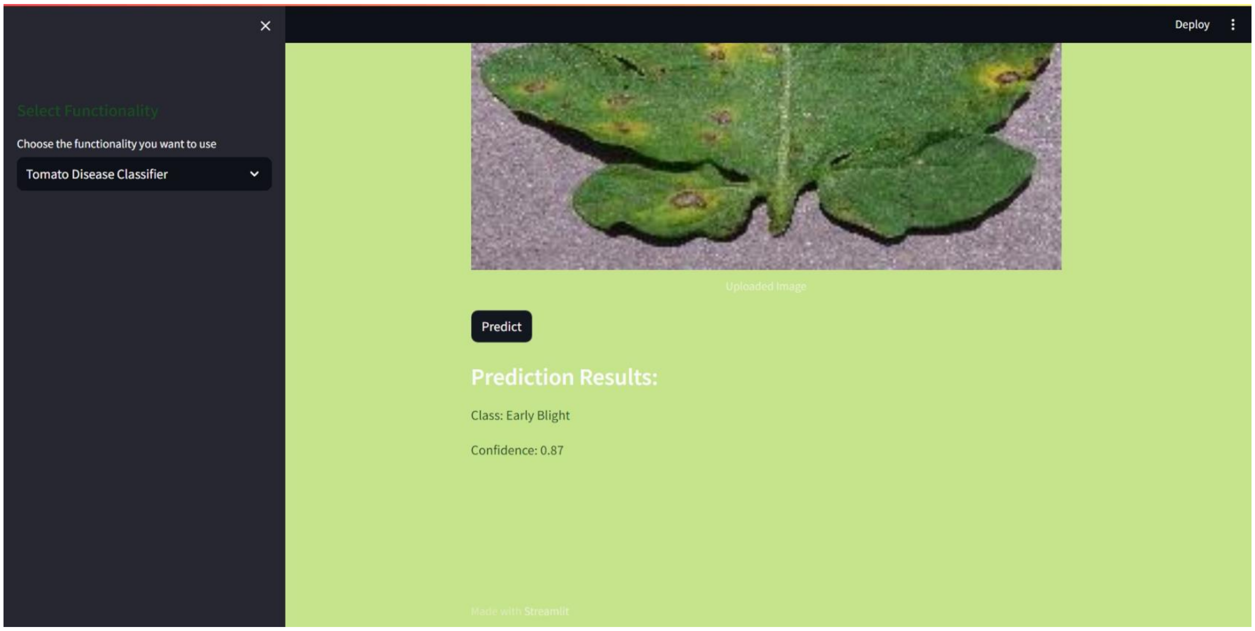


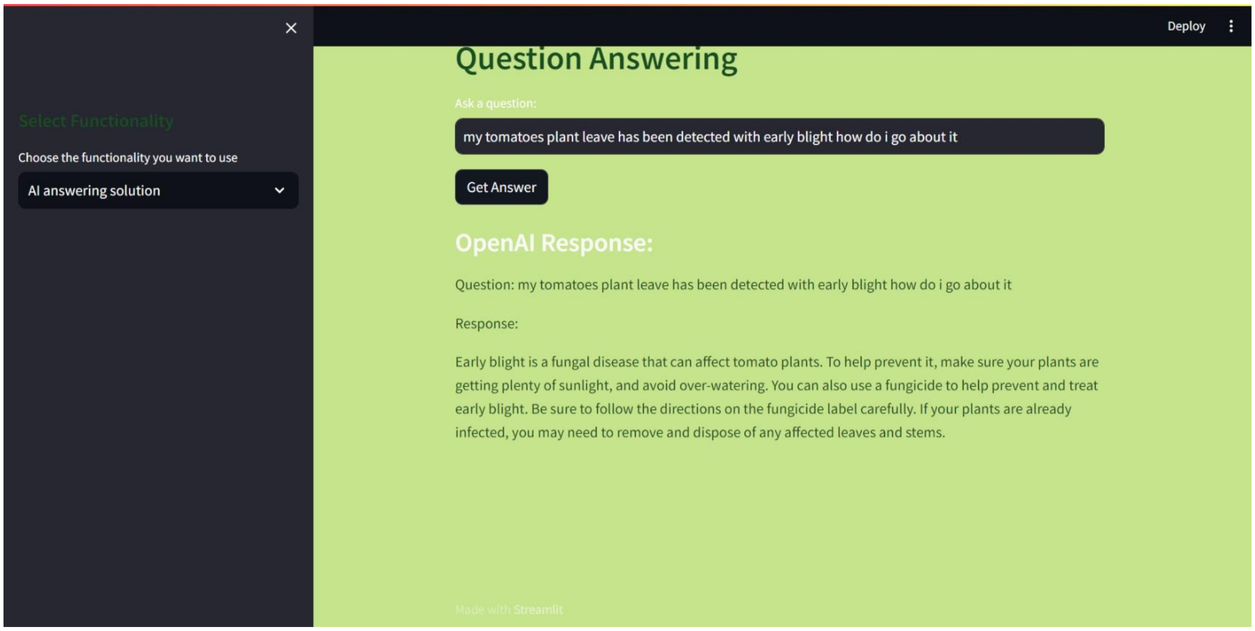
35
4
8
49
blight
healthy
True label
blight
healthy
Predicted label
5
10
15
20
25
30
35
40
45

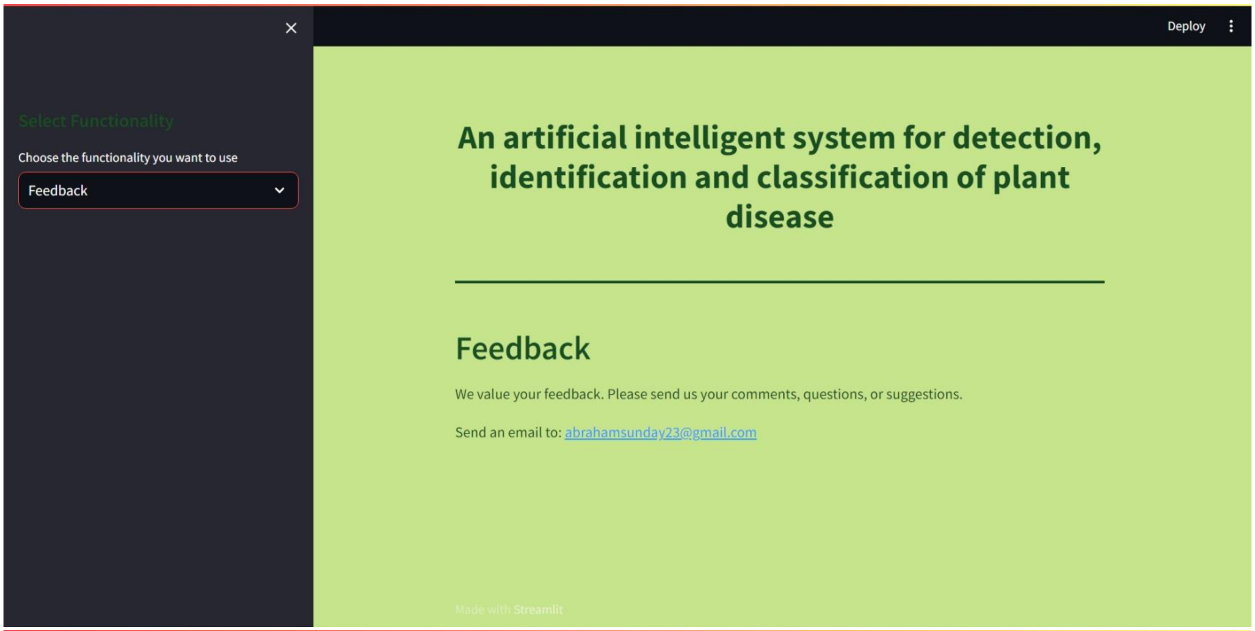

Model loss
Train
Validation loss
Loss
Epoch
0.7
0.6
0.5
0.4
0.3
0.2
0
10
20
30
40
50

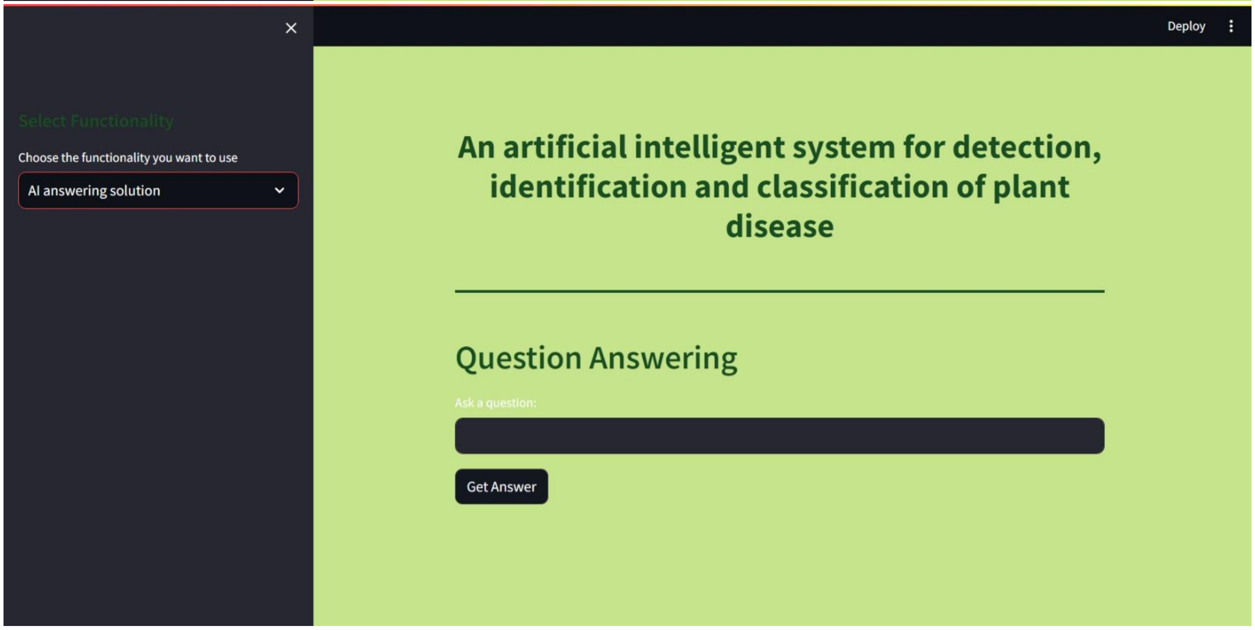

Deploy
Select Functionality
Choose the functionality you want to use
Tomato Disease Classifier
An artificial intelligent system for detection, identification and classification of plant disease
Tomato Disease Classifier
Upload an image of a tomato leaf
Drag and drop file here
Limit 200MB per file • JPG, PNG, JPEG
Browse files
Made with Streamlit

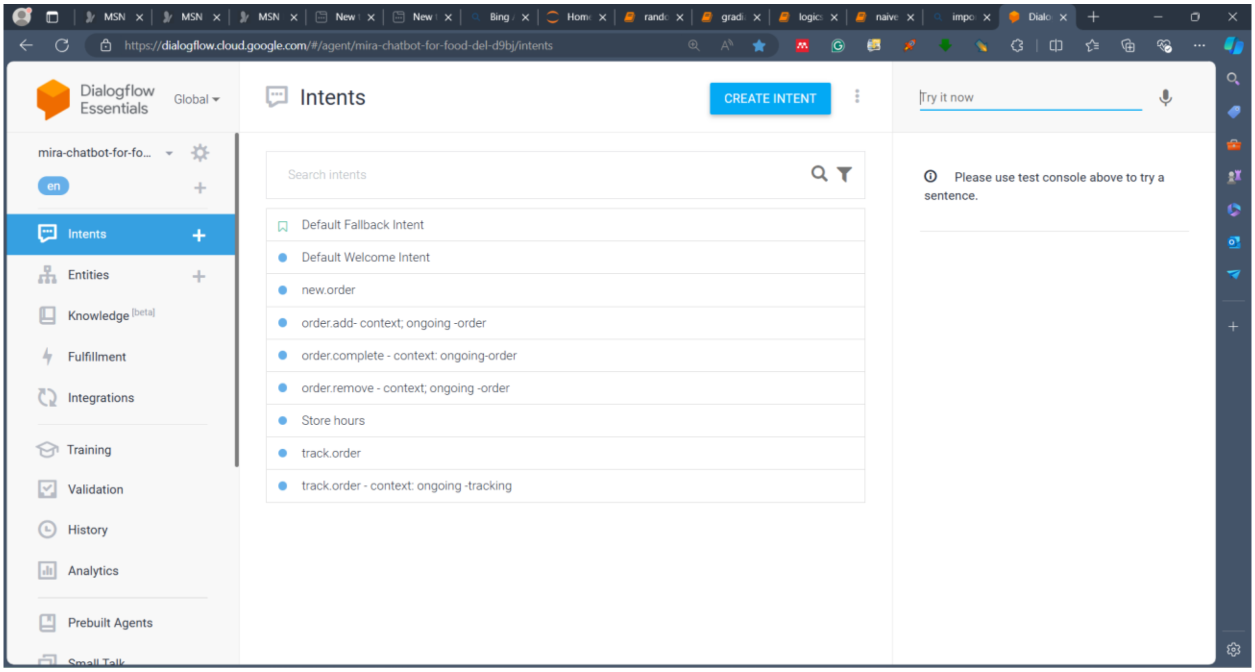

Deploy
Select Functionality
Choose the functionality you want to use
Home
An artificial intelligent system for detection, identification and classification of plant disease
Farm Management and Plant Health Advisor
Welcome to the Farm Management and Plant Health Advisor!
Select a functionality from the sidebar.
Made with Streamlit


Deploy
Select Functionality
Choose the functionality you want to use
Tomato Disease Classifier
Uploaded Image
Predict
Prediction Results:
Class: Early Blight
Confidence: 0.87
Made with Streamlit


Deploy
Select Functionality
Choose the functionality you want to use
AI answering solution
Question Answering
Ask a question:
my tomatoes plant leave has been detected with early blight how do i go about it
Get Answer
OpenAI Response:
Question: my tomatoes plant leave has been detected with early blight how do i go about it
Response:
Early blight is a fungal disease that can affect tomato plants. To help prevent it, make sure your plants are getting plenty of sunlight, and avoid over-watering. You can also use a fungicide to help prevent and treat early blight. Be sure to follow the directions on the fungicide label carefully. If your plants are already infected, you may need to remove and dispose of any affected leaves and stems.
Made with Streamlit


Deploy
Select Functionality
Choose the functionality you want to use
Feedback
An artificial intelligent system for detection, identification and classification of plant disease
Feedback
We value your feedback. Please send us your comments, questions, or suggestions.
Send an email to: abrahamsunday23@gmail.com
Made with Streamlit


Deploy
Select Functionality
Choose the functionality you want to use
AI answering solution
An artificial intelligent system for detection, identification and classification of plant disease
Question Answering
Ask a question:
Get Answer



https://dialogflow.cloud.google.com/#/agent/mira-chatbot-for-food-del-d9bj/intents
Dialogflow Essentials
Global
mira-chatbot-for-fo...
en
Intents
Entities
Knowledge [beta]
Fulfillment
Integrations
Training
Validation
History
Analytics
Prebuilt Agents
Small Talk
Intents
CREATE INTENT
Search intents
Default Fallback Intent
Default Welcome Intent
new.order
order.add- context; ongoing -order
order.complete - context: ongoing-order
order.remove - context; ongoing -order
Store hours
track.order
track.order - context: ongoing -tracking
Try it now
Please use test console above to try a sentence.