FERTILIZERS RECOMMENDATION SYSTEM FOR DISEASE PREDICTION

PROJECT REPORT

SUBMITTED BY

TEAM ID : PNT2022TMID33287

S. AKALYA	**922519104004**
K. DIVYA	**922519104037**
S. SOWNDARYA	**922519104155**
M. SWETHA	**922519104170**

In partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

In

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

V. S. B. ENGINEERING COLLEGE, KARUR

V.S.B. ENGINEERING COLLEGE, KARUR

(Approved by AICTE & Affiliated by Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report titled **"Fertilizers Recommendation System for Disease Prediction"** is the bonafide record work by **S.AKALYA (922519104004), K.DIVYA (922519104037), S.SOWNDARYA (922519104155) and M.SWETHA (922519104170)** for **IBM-NALAIYATHIRAN** in **VII** semester of **B.E.,** degree course **in Computer Science and Engineering** branch during the academic year of 2022-2023

Staff-In Charge
Nandhini Devi S

Evaluator
Gunasekaran P

Head of the Department
Mr. Anbumani P

ACKNOWLEDGEMENT

First and foremost, we express my thanks to our parents for providing us a very nice environment for doing this project. We wish to express our sincere thanks to our founder and Chairman **Shri. V. S. BALSAMY** for his endeavour in educating us in this premier institution.

We wish to express our appreciation and gratefulness to our principal, **Dr. V. NIRMAL KANNAN** and vice principal **Mr. T .S.KIRUBASANKAR** for

their encouragement and sincere guidance.

We are grateful to our head of the department **Mr. P. ANBUMANI** and our Nalaiyathiran project coordinator **Mr. P. GUNASEKARAN** Department of Computer Science and Engineering for their valuable support.

We express our indebtedness to the supervisor of our Nalaiyathiran project, **Mr. P. GUNASEKARAN** Assistant Professor, Department of Computer Science and Engineering, for guidance throughout the course of our project.

Our sincere thanks to all the teaching staff of V.S.B Engineering College and our friends for their help in the successful completion of this IBM Nalaiyathiran project work. Finally, we bow before God, the almighty who always had a better plan for us. We give our praise and glory to Almighty God for successful completion of this IBM Nalaiyathiran.

Table of Contents

ABSTRACT:

Agriculture is one of the major key for the economical growth of the world. Without agriculture, the existence of human beings is not possible as it is the main source of our food supply to sustain on the earth and it also helps to grow our economy across the world. In today's world, farmers are facing lots of challenges and troubles to successfully harvest the crops. One of the major obstacle that they face in day-day life is the various disease caused in the plants. The early diagnosis and prediction of the diseased plants can save lots of yield in most cases. Most of the farmers are unaware of the appropriate knowledge about those disease and the diagnosis techniques. These problems can be resolved by publishing a model that can able to diagnosis the disease of the plant and also be able to recommend a proper remedial through fertilizers for the plants. This will help in greatly enhancing both the quantity and quality of the crops. The model should be easily accessible web based application and should be easily understandable to the farmers. It should be user-friendly to the farmers. The model should able to identify the symptoms of the leaf through the image that is uploaded by the farmer and should be able to predict the disease and recommend the suitable fertilizers for the treatment. This can be achieved through Deep Learning Techniques. The image classification can be performed by the CNN deep learning algorithm and using proper deep learning strategies, the disease predicted can be categorized and then the fertilizer can be recommended from the database.

An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

1. INTRODUCTION

The Detection and recognition of plant diseases using machine learning are very efficient in providing symptoms of identifying diseases at its earliest. Plant pathologists can analyze the digital images using digital image processing for diagnosis of plant diseases. Application of computer vision and image processing strategies simply assist farmers in all of the regions of agriculture. Generally, the plant diseases are caused by the abnormal physiological functionalities of plants. Therefore, the characteristic symptoms are generated based on the differentiation between normal physiological functionalities and abnormal physiological functionalities of the plants. Mostly, the plant leaf diseases are caused by Pathogens which are positioned on the stems of the plants. These different symptoms and diseases of leaves are predicted by different methods in image processing. These different methods include different fundamental processes like segmentation, feature extraction and classification and so on. Mostly, the prediction and diagnosis of leaf diseases are depending on the segmentation such as segmenting the healthy tissues from diseased tissues of leaves.

Agriculture is the most important sector in today's life. Most plants are affected by a wide variety of bacterial and fungal diseases. Diseases on plants placed a major constraint on the production and a major threat to food security. Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality. In recent years, the number of diseases on plants and the degree of harm caused has increased due to the variation in pathogen varieties, changes in cultivation methods, and inadequate plant protection techniques.

An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

1.1 PROJECT OVERVIEW

In this project, two datasets name fruit dataset and vegetable dataset are collected. The collected datasets are trained and tested with deep learning neural network named Convolutional Neural Networks (CNN). First, the fruit dataset is trained and then tested with CNN. It has 6 classes and all the classes are trained and tested. Second, the vegetable dataset is trained and tested. The software used for training and testing of datasets is Python. All the Python codes are first written in Jupyter notebook supplied along with Anaconda Python and then the codes are tested in IBM cloud. Finally, a web-based framework is designed with help Flask a Python library. There are 2 html files are created in templates folder along with their associated files in static folder. The Python program 'app.py' used to interface with these two webpages is written in Spyder-Anaconda python and tested.

- Pre-process the images.
- Applying the CNN algorithm to the dataset.
- How deep neural networks detect the disease.

- You will be able to know how to find the accuracy of the model.
- You will be able to build web applications using the Flask framework.

Create a system for predicting crops according to soil details, predicting fertilizers according to soil and crop details, and detecting diseases in the plant. The objective of our system is to help farmers because it's difficult to grow interventions. Each control approach is then accompanied with the aid of using danger nodes representing unsure occasions (i.e., 'disorder free' or 'dead', with a view to having possibilities connected to them. Finally, endpoints of DTs are represented with the aid of using a terminal node (triangle) on the proper of the tree. The final results measures (e.g., software value) are usually connected to those endpoints. Costs, however, are connected to occasions in the tree, in addition to endpoints. The anticipated values (expenses and effectiveness) related to every department are expected with the aid of using 'averaging out' and 'folding back' the tree from proper to left

1.2 PURPOSE

The main purpose of this automated system is to identify the plant disease using its image and recommending fertilizers to treat the plant for a good quality and quality of the yield. This is used to test the fruits and vegetables samples and identify the different diseases. Also, this project recommends fertilizers for predicted diseases. In day-to-day life, Agriculture is the most important sector. Most plants are affected by a wide variety of bacterial and fungal diseases. Farmers face several challenges when growing crops like uncertain irrigation, poor soil quality, etc. Especially in India, a major fraction of farmers does not have the knowledge to select appropriate crops and fertilizers. Mostly, the plant leaf diseases are caused by Pathogens which are positioned on the stems of the plants. These different symptoms and diseases of leaves are predicted by different methods in image processing. These different methods include different fundamental processes like segmentation, feature extraction and classification and so on. Mostly, the prediction and diagnosis of leaf diseases are depending on the segmentation such as segmenting the healthy tissues from diseased tissues of leaves Moreover, crop failure due to disease causes a significant loss to the farmers, as well as the consumers. While there have been recent developments in the automated detection of these diseases using Machine Learning techniques, the utilization of Deep Learning has not been fully explored. Additionally, such models are not easy to use because of the high-quality data used in their training, lack of computational power, and poor generalizability of the models. To this end, we create an open- source easy-to-use web application to address some of these issues which may help improve crop production. In particular, we support crop recommendation, fertilizer recommendation, plant disease prediction, and an interactive news-feed. In addition, we also use interpretability techniques in an attempt to explain the prediction made by our disease detection model.

2. LITERATURE SURVEY

Numerous articles have been reviewed and their conclusions are summarized in this section. This section presents documents that were studied before and during project development. The documents provided a better understanding of existing solutions, how algorithms could be optimized and how selection could be facilitated algorithms on the basis of their performance.

2.1 EXISTING SYSTEM

The author says that in India, the largest source of subsistence is agriculture and its federated sectors. In rural regions, there are about 82% of small and marginal farmers, and 70% of rural households depend primarily on agriculture only. The proposed system recommends the suitable crops for the lands with varied soil nutrients. The appropriate fertilizers that are suitable for specific soil nutrient and crop sown are also recommended. Plant physiology can be damaged due to fungal, viral or bacterial diseases. Plants affected from the above pathogens are detected. Random forest classifier gives an accuracy of 98% for recommendation system, and PyTorch neural network gives an accuracy of 99.2% for disease prediction [1]. The Author states that Agriculture is the main aspect for the economic development of a country. Agriculture is the heart and life of most Indians. The soil type, fertilizer recommendation, diseases in plants and leaves. Plant disease, especially on leaves, is one of the major

factors that reduce the yield in both quality and quantity of the food crops. Finding the leaf disease is an important role to preserve agriculture. Smart analysis and Comprehensive prediction model in agriculture helps the farmer to yield right crop at the right time. The main benefits of the proposed system are as follows: Yield right crop at the right time, Balancing the crop production, control plant disease, Economic growth, and planning to reduce the crop scarcity. Hence to Detect and recognize the plant diseases and to recommend fertilizer it is necessary to provide symptoms in identifying the disease at its earliest [2]. The Author claims that Agriculture is the mainstay of a rising economy in India. Traditionally farmers followed ancestral farming patterns and norms. However, a single farmer cannot be expected to take into account all innumerable factors that contribute to crop growth. A single misguided or imprudent decision by the farmer can have undesirable ramifications. With the advancements in various domains, intelligent agricultural system is needed for upliftment of Indian economy. The collaboration of recommender system with machine learning will lead to Intelligent Agriculture System that helps the farmer community in their decision making of farm management and agribusiness activities such as

a. Predicting agriculture commodity market price before cultivation,
b. Determining best cultivars to plant
c. Determine optimum cultivation date
d. Evaluate demand and supply risk
e. Investment Prioritizing. It also helps farmer to perform the activities like crop management including applications on yield prediction, disease detection, weed detection, crop quality, and growth prediction etc. This chapter describes the case study on “Crop Disease Detection and Yield prediction”. The study includes identification of crop condition, disease detection, prediction about specific crop and recommendation using machine learning algorithms. It gives an idea about how recommender system is used in agriculture for disease detection and prediction [3].

2.2 REFERENCE

[1] SuriyaKrishnaan, K., Kumar, L. C., & Vignesh, R. (2022). Recommendation System for Agriculture Using Machine Learning and Deep Learning. In Inventive Systems and Control (pp. 625-635). Springer, Singapore.

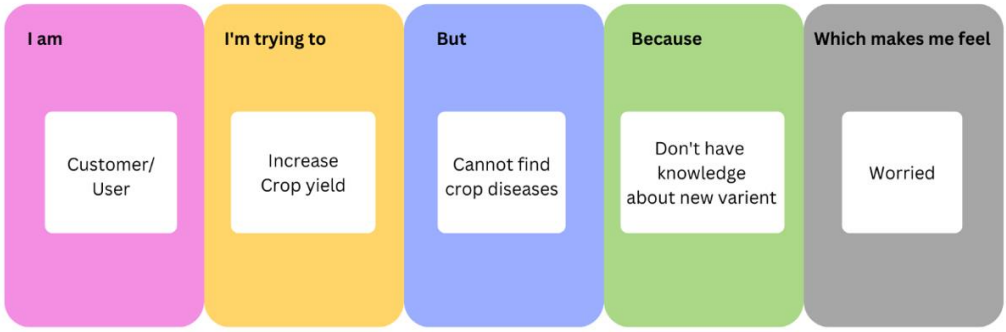
[2] Selvi, P. P., & Poornima, P. Soil Based Fertilizer Recommendation System for Crop Disease Prediction System.

[3] Akulwar, P. (2020). A recommended system for crop disease detection and yield prediction using machine learning approach. Recommender System with Machine Learning and Artificial Intelligence: Practical Tools and Applications in Medical, Agricultural and Other Industries, 141-163.

2.3 PROBLEM STATEMENT DEFINITION

Farmer's customary strategies for farming development are inadequate. It does not make proper use of every accessible asset. Farmers cannot identify crop diseases because of an absence of information and old practices, which frequently bring about soil supplement weakening and depletion. Thus, crop failure happens. Developing just specific yields drains the dirt, what's more, assuming that the yields are hurt by diseases. Farmers are ignorant of how to recuperate such crops. Food needs cannot be met until and except if productive asset the executives and use is carried out

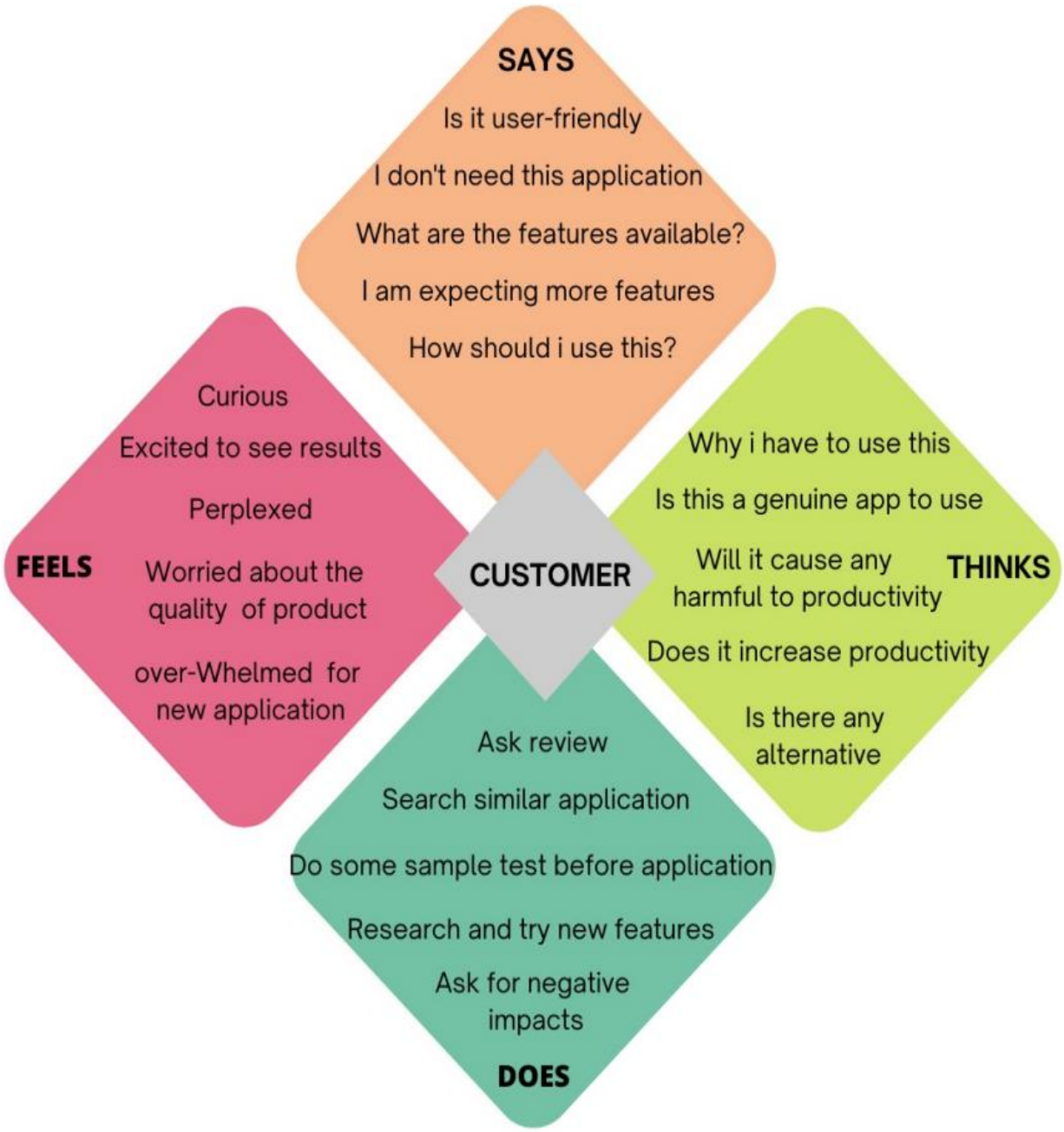
I am	Ram (customer) and I am a farmer.
I'm trying to	Get more yield with good quality.
But,	I cannot find the diseases on plants by using onlythe old practices, which leads to poor quality andlow quantity of the crop.
Because	I don't have much knowledge about the new variant of the crop diseases.
Which makes me feel	If I have any source for the identification of thecrop disease, it will be helpful to increase the crop yield.



Problem Statement (PS)	**I am**	**I am trying to**	**But**	**Because**	**Which makes me feel**
PS-1	Ram(customer)	Increase my crop yield.	I could not identify the crop disease.	I am not aware of the new variant of the disease.	Unhappy
PS-2	Sam(Customer)	Get good quality crop yield.	I do not know how to use appropriate fertilizers for the crop disease.	I do not know thecorrect proportion of the fertilizers forthe particular disease.	Worried about the yield. If I had a sourceto identify the disease, it will be more helpful to increase the crop yield.

3. IDEATION & PROPOSED SOLUTION

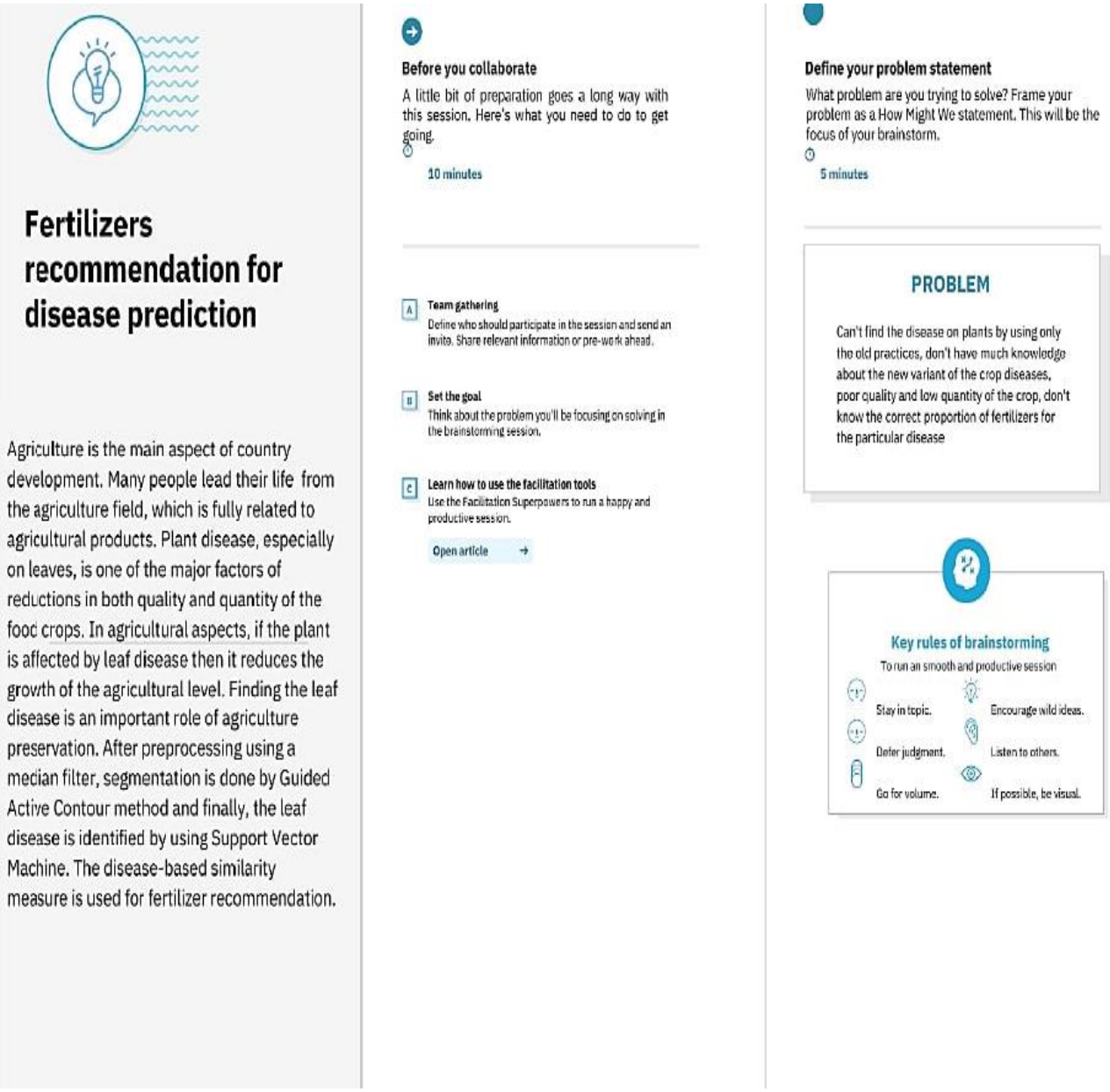
3.1 EMPATHY MAP CANVAS



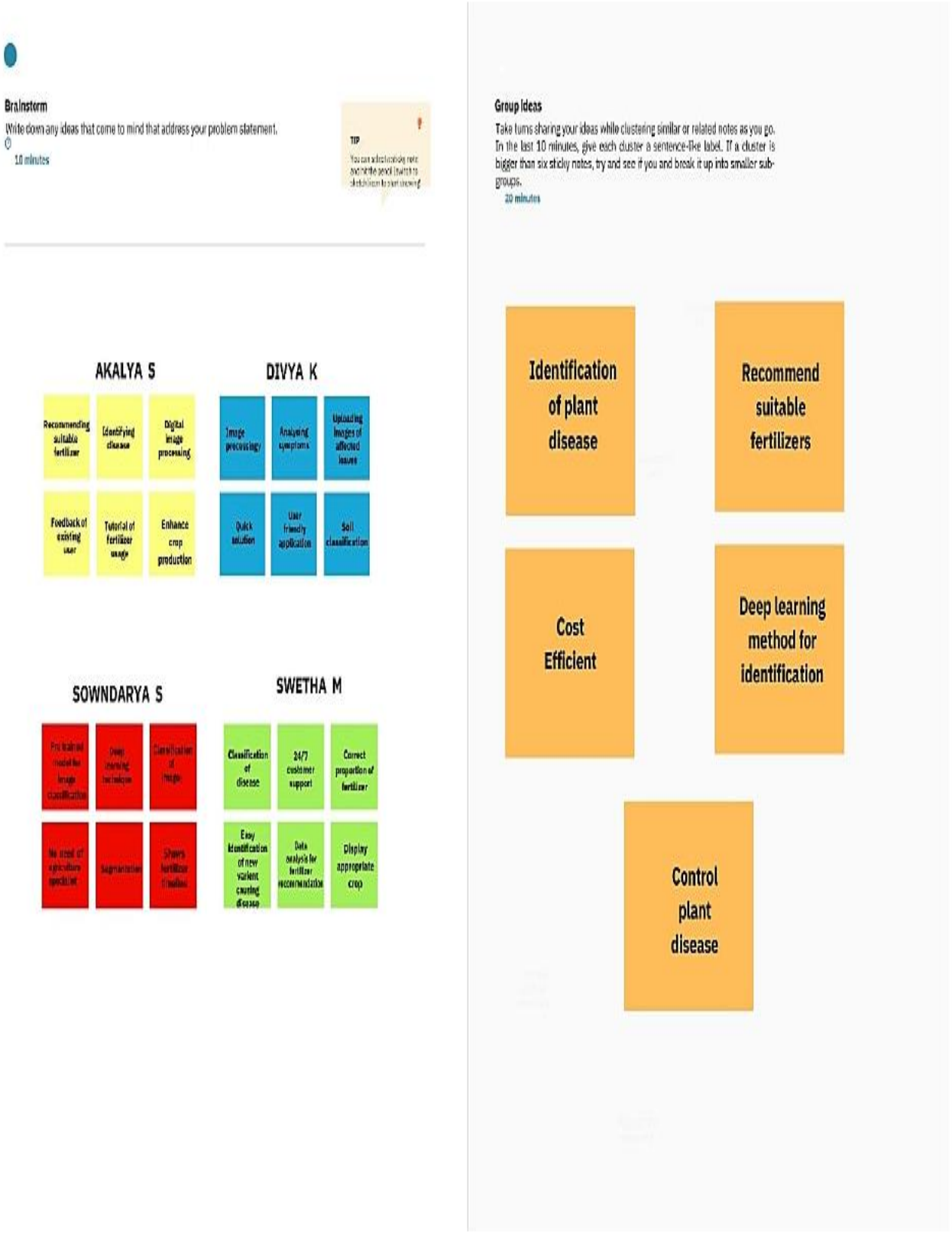
3.2 IDEATION AND BRAINSTORMING

Conceptualizing gives a free and open climate that supports everybody inside a group to partake in the imaginative reasoning cycle that prompts critical thinking. Focusing on volume over esteem, out-of-the-case thoughts are gladly received and based upon, and all members areurged to team up, helping each other foster a rich measure of clever fixes

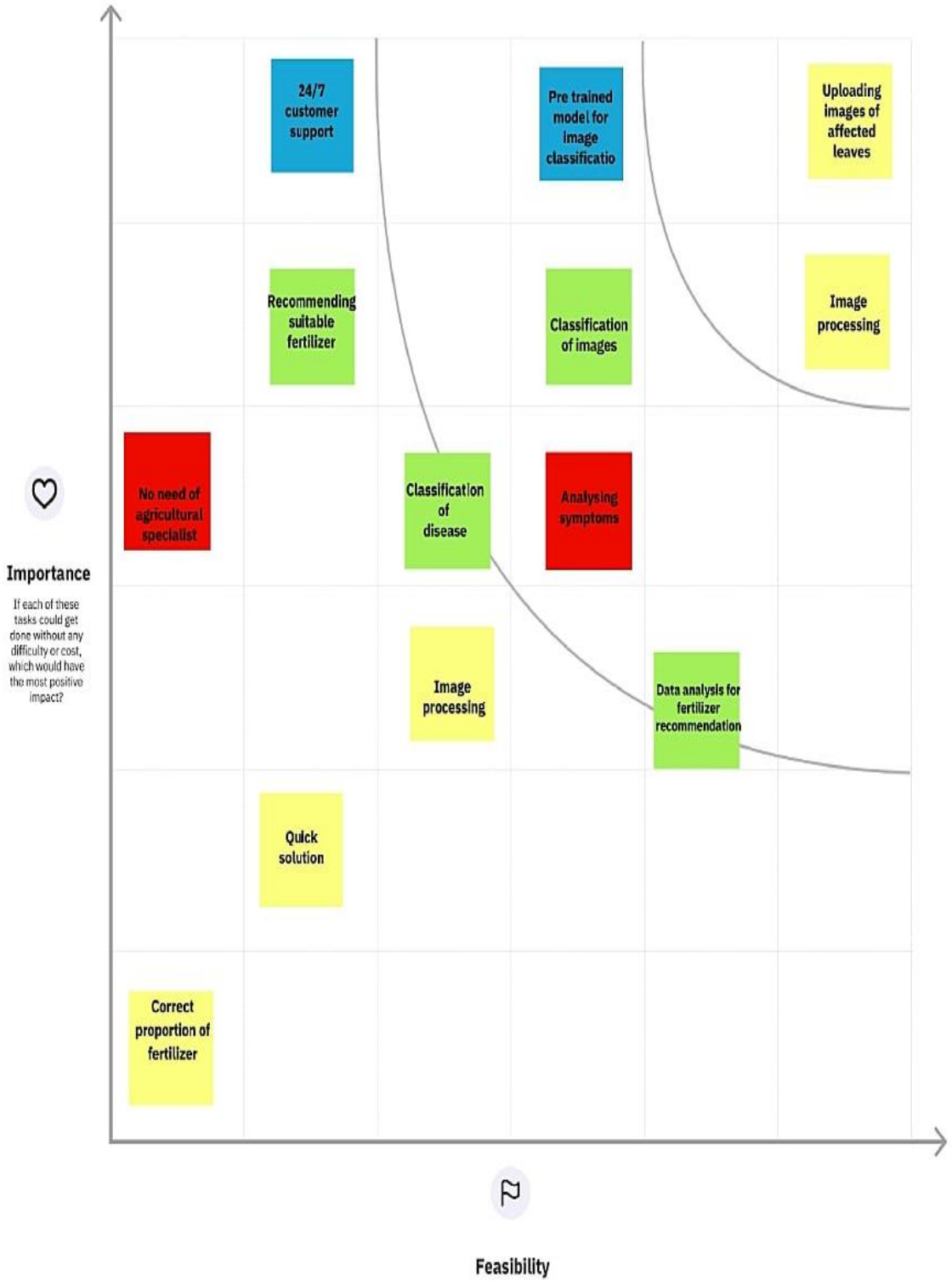
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



3.3 PROPOSED SOLUTION

In this project work, a deep learning based neural network is used to train the collected datasets and test the same. The deep learning based neural network is CNN which gives more than 90% classification accuracies. By increasing Image Pre-processing Image dataset collection Image dataset training Build & Save Mode Predict the test dataset the more number of dense layers and by modifying hyperparameters such as number of epchs, batch size, the accuracy rate can be increased to 95% to 98%.

S.No	Parameter	Description
1.	Problem Statement (Problem to besolved)	Farmers' customary strategies for farming development are inadequate. It doesn't make proper use of every accessible asset. Farmers can't identify crop diseases because of an absence of information and old practices, which frequently bring about soil supplement weakening and depletion. Thus, crop failure happens. Developing just specific yields drain the dirt, what's more, assuming that the yields are hurt by diseases. Farmers are ignorant of how to recuperate such crops. Food needs can't be met until and except if productive assets the executives and use is carried out.

2.	Idea / Solution description	Now-a-days, farmers are struggling to identify the diseases on plants by using only the old practices and techniques. So, an AI based automated software is introduced to identify the types of disease by analysing the symptoms that shown on the leaves of the plant. After the identification of the disease, a suitable fertilizer needs to be recommended for treating the plant diseases. Though fertilizers have some disadvantages, proper and limited usage of the fertilizers can cure the diseaseas well as give more production to the farmers. An AI-based technology helps to improve efficiency in all sectors and manages The challenges facing various industries including various sectors in the agricultural sector such as crop harvesting, irrigation, soil content sensitivity, crop monitoring, weed, harvest and establishment.
3.	Novelty / Uniqueness	Artificial intelligence and sensor technology play a vital role in the farming field. The field of agriculture is in a great threat this includes the diseases that attack the plant leaves. Our system finds the area of the leaf that has been affected and also the disease that attacked the leaves. A system that automatically detects leaf disease with the help of image processing is being developed. This system does few image pre-processing techniques like image acquisition, image segmentation, feature extraction and classification. Modern agricultural practices assure great development of cultivation. We have many smart agriculture developing models to monitor the temperature, humidity, moisture content and spots in leaves that do work automatically but there are few systems that detects problems and provides suggestions to the problem. One such automatic disease detection system is developed for the identification of the disease and recommend appropriate fertilizer.

4.	Social Impact / Customer Satisfaction	Conventional method of preparation can be characterized as a methodology of treatment by which little, ignorant, rental land farmers, and farmers' more youthful age utilize the high measure of fertilizers to acquire high harvest yield and cash, without the thought of hardships. The fact that excessive and unpredictable use makes plants dry and weak and bug sprays are genuinely unsafe to some climates across the world. But the fertilizers in the correct proportion is what is needed for healthy crops and human health which is suggested accurately by our automated software.
5.	Business Model (Revenue Model)	Fertilizers can aid in making profitable changes in farming by increasing crop yield. Farmers can reduce costs per unit of production and increase the margin of return over total cost by increasing rates of application of fertilizer on principal cash and feed crops. This not just offers significant advantages for farmer livelihoods and food security, yet additionally delivers ecological advantages by decreasing our requests for farmland.

6.	Scalability of the Solution	The fundamental rationale of these applications and accessible strategies of computerized reasoning to settle the issues of farmers in getting the expected yield. The different literature in artificial knowledge, which reflects different approaches to recognize the illnesses in crops. From this, it is presumed that counterfeit knowledge is an incredible instrument for a country's agronomics. Subsequently, future specialists ought to coordinate a legitimate dataset covering all fields of farming and upgrade the accessible advances to build the efficiency of essential areas.

3.4 PROBLEM SOLUTION FIT

4. REQUIREMENT ANALYSIS

There are two types of requirement analysis, namely

1. Functional Requirement
2. Non-functional Requirements

4.1 FUNCTIONAL REQUIREMENTS

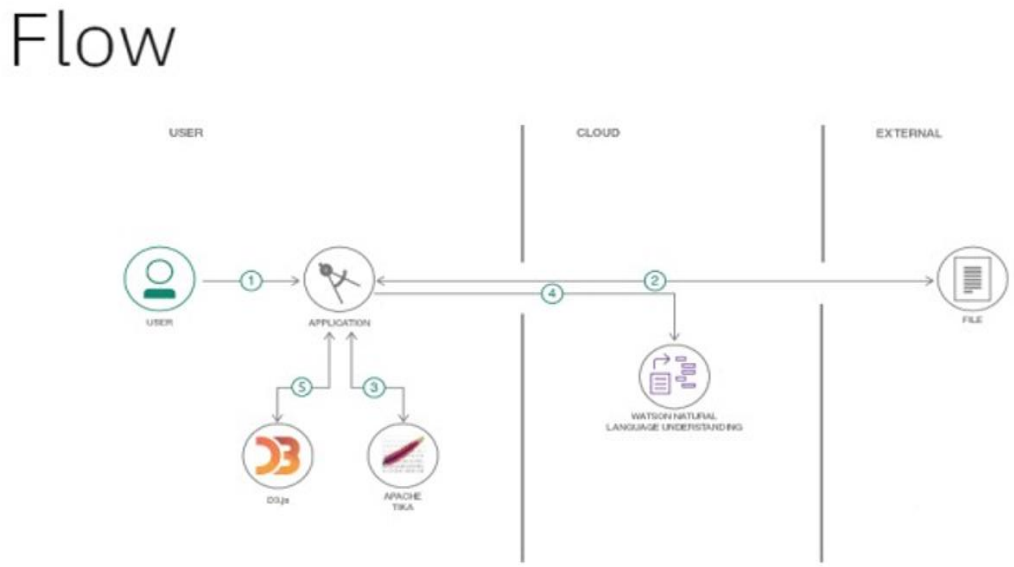
FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	User Registration	Registration through Mobile number Registration through Email
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Materials and Methods	A camera or similar devices are used to capture different images of the affected leaves and then it helps to identify the disease of the leaves.
FR-4	Crop Details	User can provide details like images of the leaves, and also provide the details of the affected plant like its name, variety and so on
FR-5	Prediction	The system will predict the disease by the symptoms identified on the uploaded image of the affected leaves and trained data.
FR-6	Fertilizer Recommendation	The system will recommend suitable fertilizer based on the type of disease identified.

4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system will be user-friendly and allows the user to perform the task easily and efficiently
NFR-2	**Security**	Information gathered from the user is highlysecured.
NFR-3	**Reliability**	The prediction will be accurate and the recommended fertilizer will cure the plant disease
NFR-4	**Performance**	The performance is based on the quality of the imagefor disease prediction.
NFR-5	**Availability**	It is available to predict the disease in plants/leavesfor all user.
NFR-6	**Scalability**	Increase the precise of disease prediction in leaf.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



1. User configures credentials for the Watson Natural Language Understanding service and starts the app.
2. User selects data file to process and load.
3. Apache Tika extracts text from the data file.
4. Extracted text is passed to Watson NLU for enrichment.
5. Enriched data is visualized in the UI using the D3.js library.

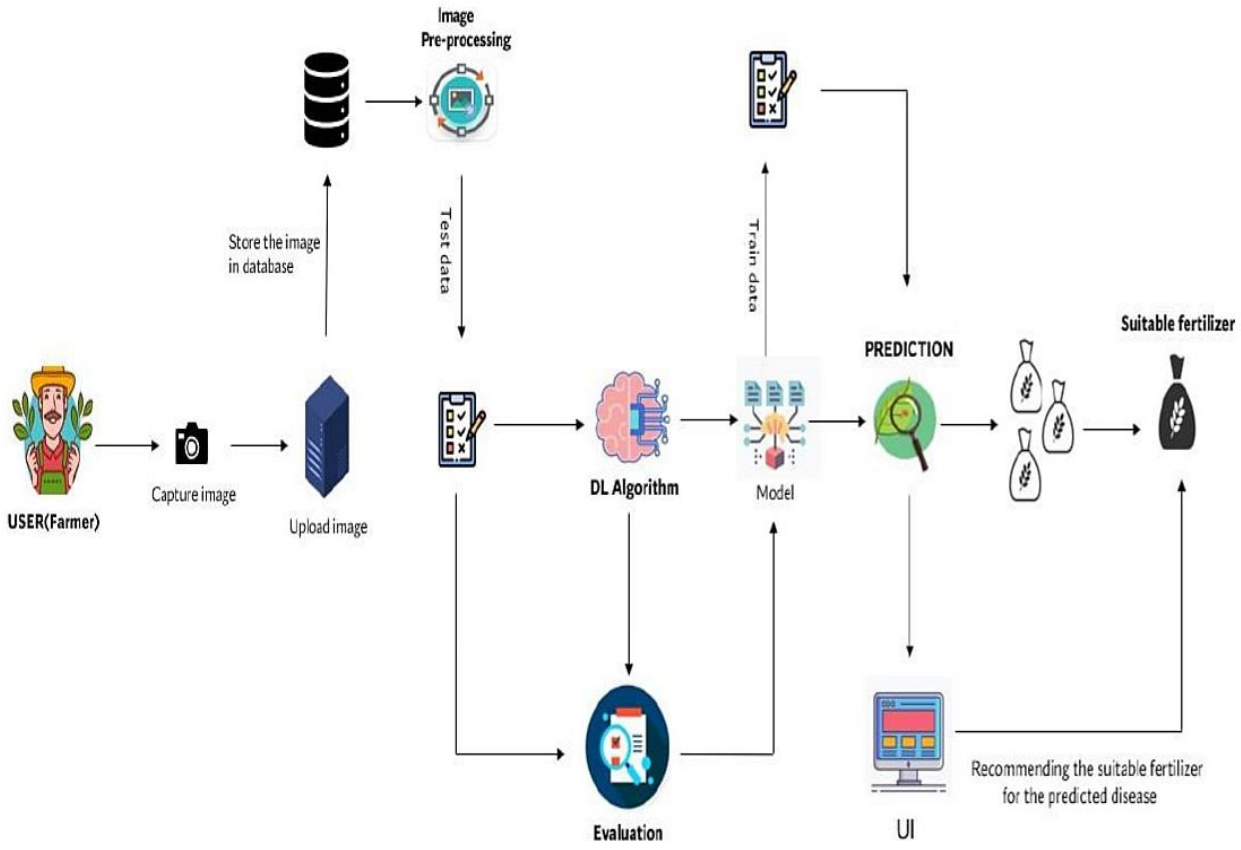


Image
Pre-processing
Store the image
in database
Test data
Train data
PREDICTION
Suitable fertilizer
USER(Farmer)
Capture image
Upload image
DL Algorithm
Model
Evaluation
UI
Recommending the suitable fertilizer
for the predicted disease

User Type	Functional Requirement (Epic)	User Story Number	User Story/ Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming mypassword.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I willreceive confirmation emailonce I have registered for the application	I can receive confirmation email & clickconfirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register &access the dashboard with FacebookLogin	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register & access thedashboard with Gmail Login	Medium	Sprint-1
	Login	USN-5	As a user, I can log intothe application by entering email & password	I can login & accessthe dashboard with Facebook Login	High	Sprint-1
Customer (Web user)	Login	USN-1	As a registered user,I can log in to the web application	I can access my account usingmy login credentials through web application	High	Sprint-1
	Logout	USN-2	As a user, I can log out in to the web application	I can exit from my web application	High	Sprint-1
	Reset my password	USN-3	As a logged in user,If I forget my password I can resetmy password	I receive reset passwordlink through my email	Medium	Sprint-1

	Upload image	USN-4	As a user, I will uploadthe image of affectedleaves	I will receive the result(predicted disease) of the image	High	Sprint-1
	Comment	USN-5	As a logged user, I can posta comment aboutan application	I access the comment section option through my web app	Low	Sprint-2
Administrator	Predicting the disease and recommend fertilizer for it	USN-1	Predicting the disease and will get the recommended fertilizer for the diseased plant	Push the notifications to thecustomer about the results	High	Sprint-1

5.2 SOLUTION ARCHITECTURE

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions.

Farmers lack the knowledge of disease and hence they produce less production. Due to the improvement and development in technology our system are smart enough to recognize and detect plant diseases and recommend the suitable fertilizer.

The major steps involved are as follows:

✓ Image acquisition

✓ Pre-processing

✓ Segmentation

✓ Disease Prediction

✓ Recommend the suitable fertilizer.

The images of the diseased plants are obtained and store the images in the database. It is Pre-processed against the dataset of diseased plants.

Deep Learning Algorithm is used to process the images and then it is evaluated for the prediction.

Then a model is built on the evaluations, it is then trained and prediction of disease is done. The system will automatically recommend the suitable fertilizer for the predicted disease.

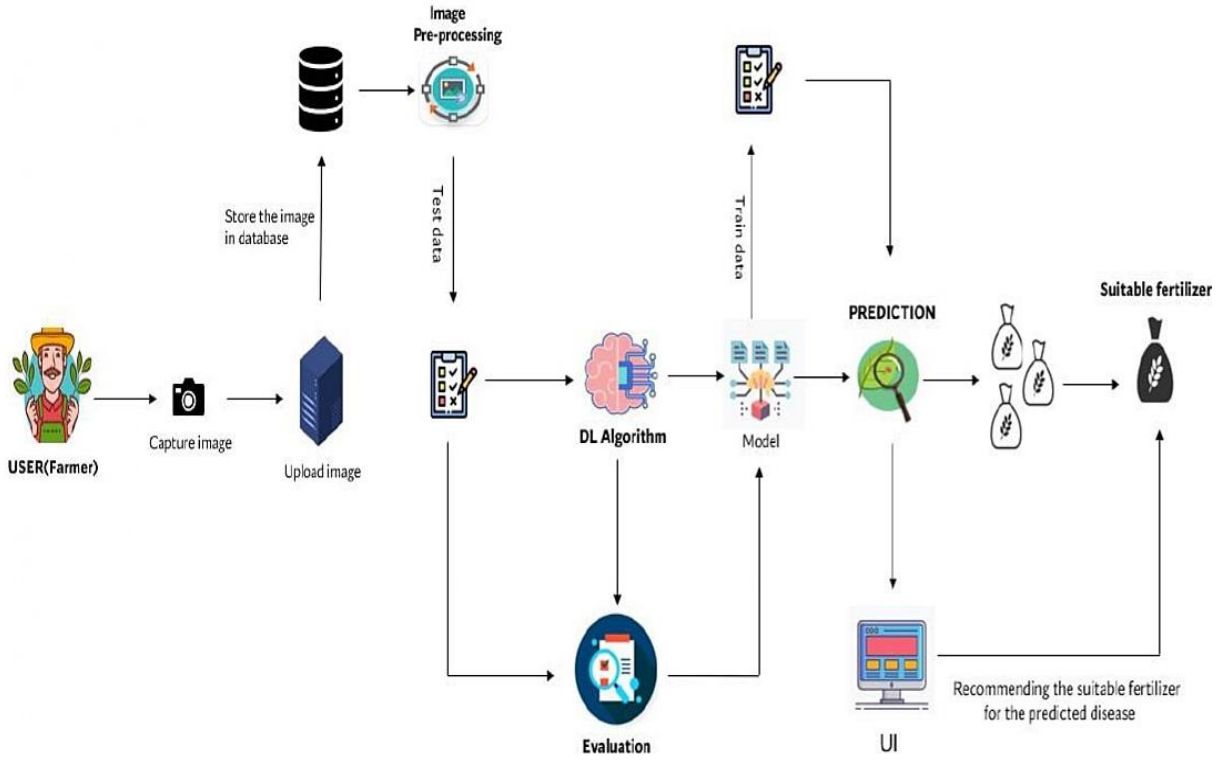


Image
Pre-processing
Store the image
in database
Test data
Train data
PREDICTION
Suitable fertilizer
USER(Farmer)
Capture image
Upload image
DL Algorithm
Model
Evaluation
UI
Recommending the suitable fertilizer
for the predicted disease

TECHNICAL ARCHITECTURE

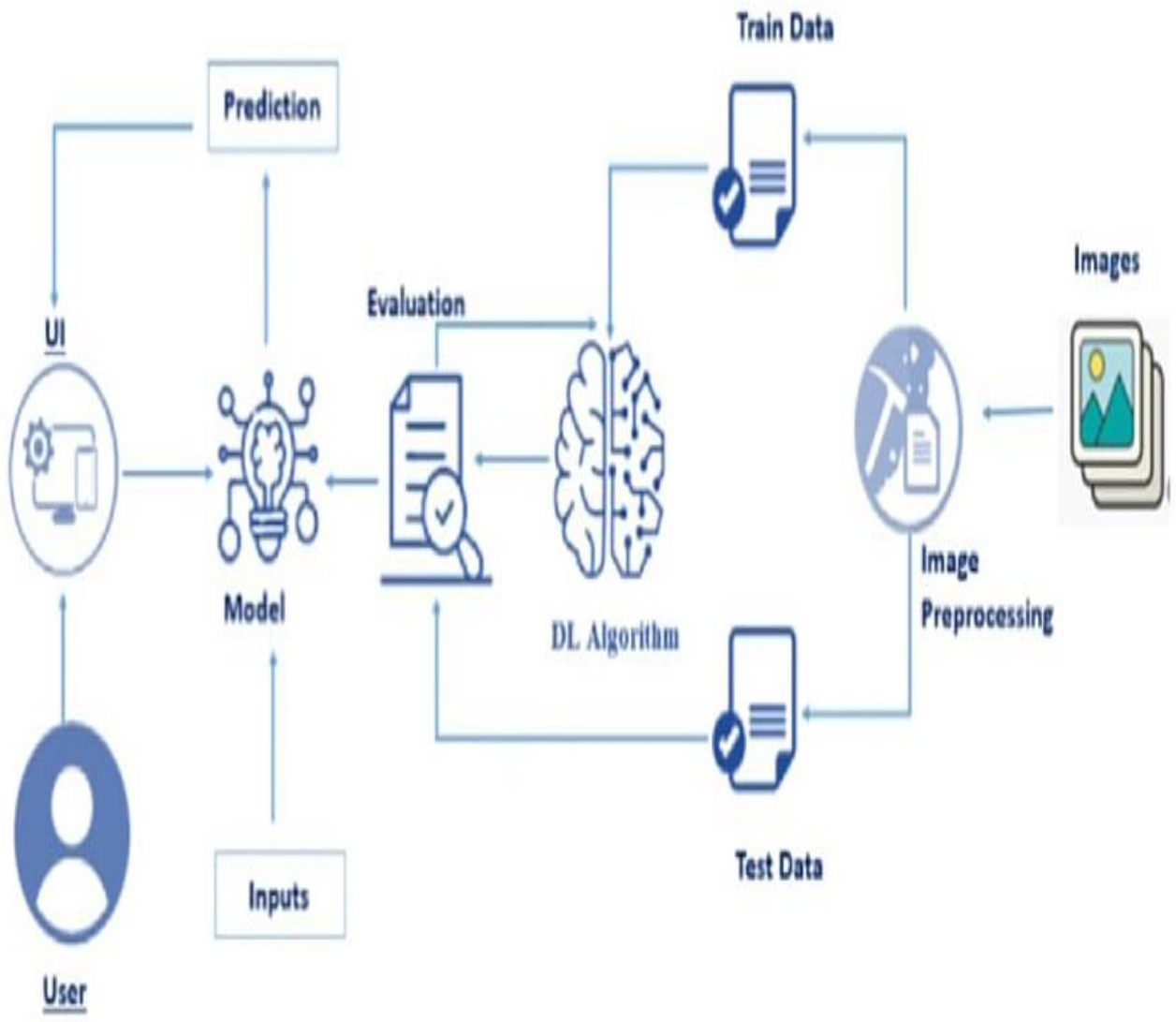


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	These techniques used by recommender systems.	Collaborative based filtering, content based technique and hybridalgorithm.
2.	Application Logic-1	Itis used for developing the devices.	Python.
3.	Application Logic-2	It provides multiple services, including frequent cropremainders, weatherreports.	Farmer app.
4.	Application Logic-3	It providing symptoms of identifying diseases at itsearliest.	Machine learning.
5.	Database	Itanalyzing the national soil database.	SRDI web-based software.
6.	Cloud Database	Helps to farmers make better decisions about managing their crops.	Temperature and moisture sensors,satellite images ,weather station.
7.	File Storage	It used for image recognition and tasks that involvetheprocessing of pixel data.	Convolutional neural networks model.
8.	External API-1	It enables programmatic access for integration, withmethods for managing products, product variants, images.	Open foodnetwork API.
9.	External API-2	It provides accessto data fromdevices, user profiledata and more.	Auto grow REST API.
10.	Machine Learning Model	Focusing on each component it is important tominimize the overall lossesin production.	Pre –harvesting machine learning.
11.	Infrastructure (Server/ Cloud)	Hybrid clouds allow data and apps to move betweenthetwo environment.	Hybrid cloudcomputing.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Leading to higher yields and improved quality of thefood and fiber thatwas grown.	Irrigation and air seedingtechnology.
2.	Security Implementations	It allows farmers to accurately navigate to specificlocation in the field, year after year, to collect soil samples or monitor cropconditions.	GPS technology.
3.	Scalable Architecture	By using thesetechnologies you canbuild a scalableweb application.	Microservices, cloud storage and caching.
4.	Availability	Water-soluble fertilizer is often useful as a quick boost for vegetables, liquids mixed with water areapplied as frequently as once a week.	Collaborative basedfiltering techniqueandhybrid algorithm.
5.	Performance	Theproposed method uses SVM to classify tree leaves, Identify the disease and suggest thefertilizer.	Support vector machine.

5.3 USER STORIES

A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming mypassword.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I willreceive confirmation email once I have registered for the application	I can receive confirmation email & clickconfirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register &access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register & access thedashboard with Gmail Login	Medium	Sprint-1
	Login	USN-5	As a user, I can log intothe application by entering email & password	I can login & accessthedashboard with Facebook Login	High	Sprint-1
Customer (Web user)	Login	USN-1	As a registered user,I can log in to the web application	I can access my account usingmy login credentials through web application	High	Sprint-1
	Logout	USN-2	As a user, I can log out in to the web application	I can exit frommy web application	High	Sprint-1
	Reset my password	USN-3	As a logged in user,If I forget my password Ican resetmy password	I receive reset passwordlink through my email	Medium	Sprint-1

	Upload image	USN-4	As a user, I will uploadthe image of affectedleaves	I will receive the result(predicted disease) of the image	High	Sprint-1
	Comment	USN-5	As a logged user, I can posta comment aboutan application	I access the comment section option through my web app	Low	Sprint-2
Administrator	Predicting the disease and recommend fertilizer for it	USN-1	Predicting the disease and will get the recommended fertilizer for the diseased plant	Push the notifications to the customer about the results	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Milestone Name	Milestone Number	Description	Mandatory	Optional
Project Objectives	M-001	We will be able to learn to prepare dataset, image processing, working with CNN layers, read images using Open CV and CNN for computer vision AI	Yes	-
Project Flow	M-002	A project management process flowchart is a graphical aid, designedto visualize the sequence of steps to be followed throughout the project management process	Yes	
Pre-Requisites	M-003	To complete this project we should have known following project such as Keras, Tensor flow, Python, Anaconda, Open CV, Flask, Scikit-learn etc...	Yes	
Prior Knowledge	M-004	One should have knowledge on the Supervised Learning ,CNN and Regression Classification and Clustering, ANN	Yes	
Data collection	M-005	We can collect dataset from different open sources like kaggle.com, UCI machine learning etc	Yes	
Image Preprocessing	M-006	Importing the Image Data Generator libraries, Define Parameters/Arguments for Image Data Generator class, Applying Image Data Generator Functionality to train set and test set.	Yes	
Model Building	M-007	Importing the model building libraries, Initializing the model, Adding CNN layers, Adding Dense layers, Configuring the learning Process, Train the model, Save the model, Predictions.	Yes	
Video Analysis	M-008	Open cv for video processing, creating an account in twilio service andsending alert message	Yes	
Train CNN model	M-009	Register for IBM Cloud and train Image Classification Model	Yes	
Ideation Phase	M-010	Prepare Literature Survey on the selected Project and Information Gathering, empathy map and ideation	Yes	
Project Design Phase-I	M-011	Prepare Proposed solution , problem-solution fit and Solution Architecture	Yes	
Project Design Phase-II	M-012	Prepare Customer journey ,functional requirements, Data flow diagram andTechnology Architecture	Yes	
Project Planning Phase	M-013	Prepare Milestone list , Activity list and Sprint Delivery Plan	Yes	
Project Development Phase	M-014	Project Development delivery of Sprint 1, Sprint 2, Sprint 3, Sprint 4	Yes	

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Functional Requirement (Epic)	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint-4	Home Page	USN-1	As a user, I can view the homepage which hasa description of the plant disease as well as options to sign up or log in.	1	Low	Akalya Divya Sowndarya
Sprint-4	Sign Up Page	USN-2	As a user, I can register for the application by entering my name, phone number, email, password, and confirming my password.	1	High	Akalya Divya Sowndarya Swetha
Sprint-4	Authorization	USN-3	As a user, I will receiveconfirmation email onceI have registered for the application.	1	High	Akalya Divya Sowndarya Swetha
Sprint-4	Login	USN-4	As a user, I can log intothe application by entering email & password.	1	High	Akalya Divya Sowndarya Swetha
Sprint-4	Dashboard	USN-5	As a user, I can upload images of the affectedleaves to the website in order to receive the plant disease.	1	High	Akalya Divya Sowndarya Swetha

Sprint	Functional Requirement (Epic)	User Story Number	UserStory / Task	StoryPoints	Priority	Team Members
Sprint-4	Results	USN-6	As a user, I can receivea disease in addition to recommendations on what I should do now.	1	High	Akalya Divya Sowndarya Swetha
Sprint-1	Data Collection	USN-7	I need to collect data (images of leaf and with disease stages).	5	High	Akalya Divya Sowndarya Swetha
Sprint-1	Data Pre-Processing	USN-8	I need to cleanmy data and prepare it for modelbuilding by doing pre-processing activities such as resizing etc.	7	High	Akalya Divya Sowndarya Swetha
Sprint-2	Model Building	USN-9	I need to buildthe model usingConvolution Neural Network (CNN).	12	High	Akalya Divya Sowndarya Swetha
Sprint-3	Model Deployment	USN-10	I need to deploythe Machine Learning modelthat was built.	12	Medium	Akalya Divya Sowndarya Swetha
Sprint-4	Application Building	USN-11	I need to buildthe website forthe applicationusing HTML, CSS etc.	6	High	Akalya Divya Sowndarya Swetha

ProjectTracker, Velocity & BurndownChart

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	12	6 Days	24 Oct2022	29 Oct 2022	12	30 Oct 2022
Sprint-2	12	6 Days	31 Oct2022	05 Nov 2022	12	06 Nov 2022
Sprint-3	12	6 Days	07 Nov 2022	12 Nov 2022	12	12 Nov 2022
Sprint-4	12	6 Days	14 Nov 2022	19 Nov 2022	12	19 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Average Velocity(A---V) = 12/6 = 2

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

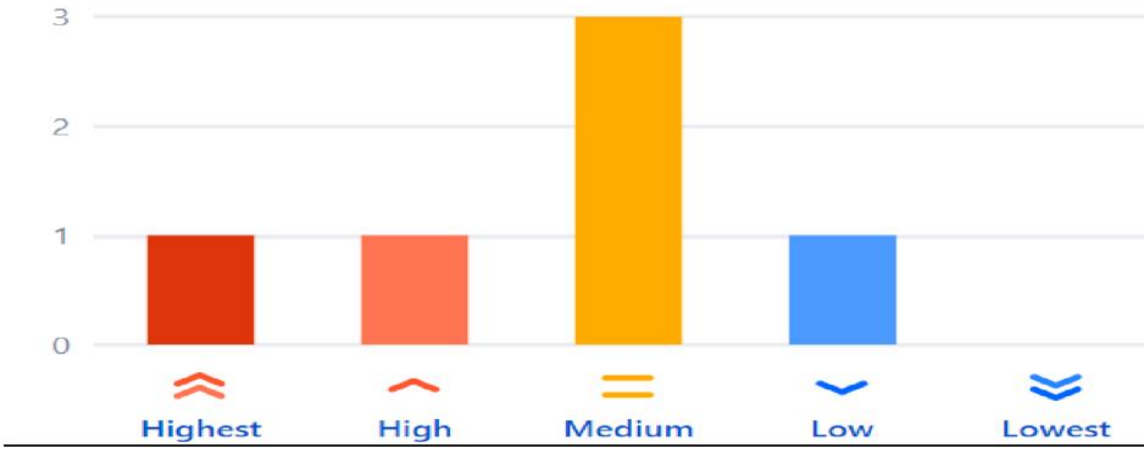
Status overview

View the progress of your project based on the status of each item. For more details, go to the board view.

Open	0
To Do	0
In Progress	0
In Review	0
Cancelled	0
Done	10
Rejected	0
Total	**10**

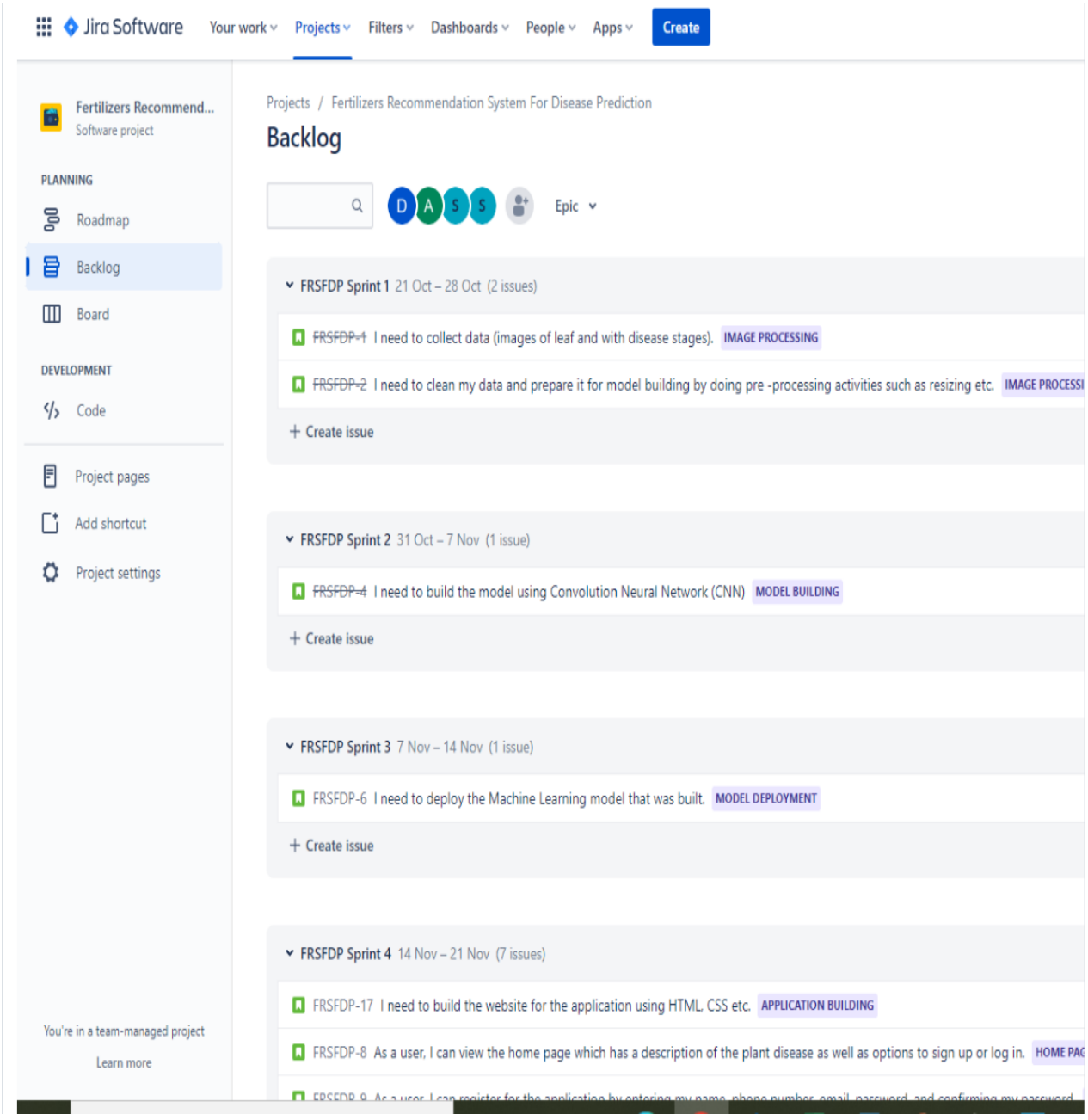
Priority breakdown

Get a holistic view of how work is being prioritized within your project. To check if the team's focusing on the right work, go to the list view.

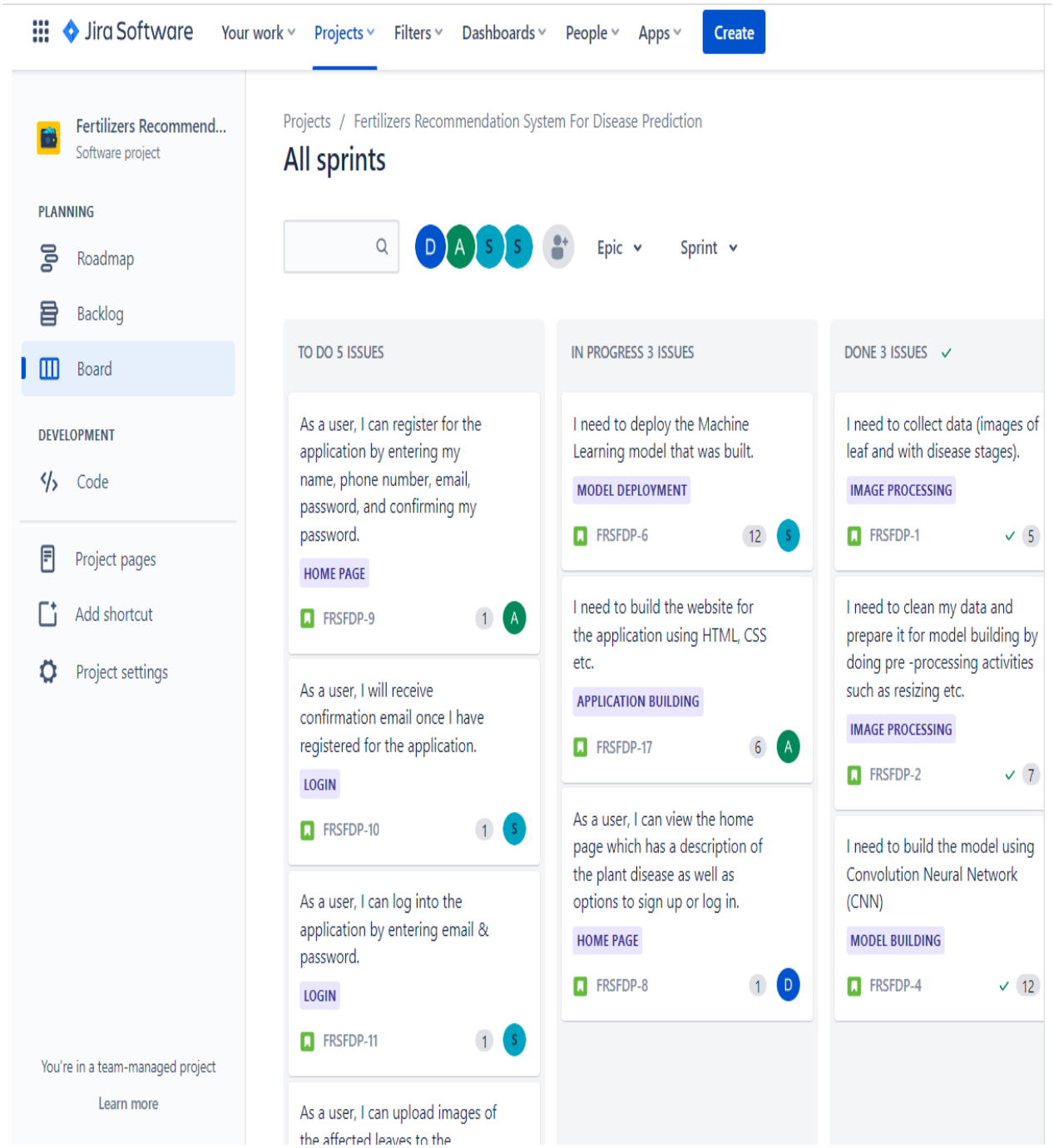


6.2 REPORTS FROM JIRA

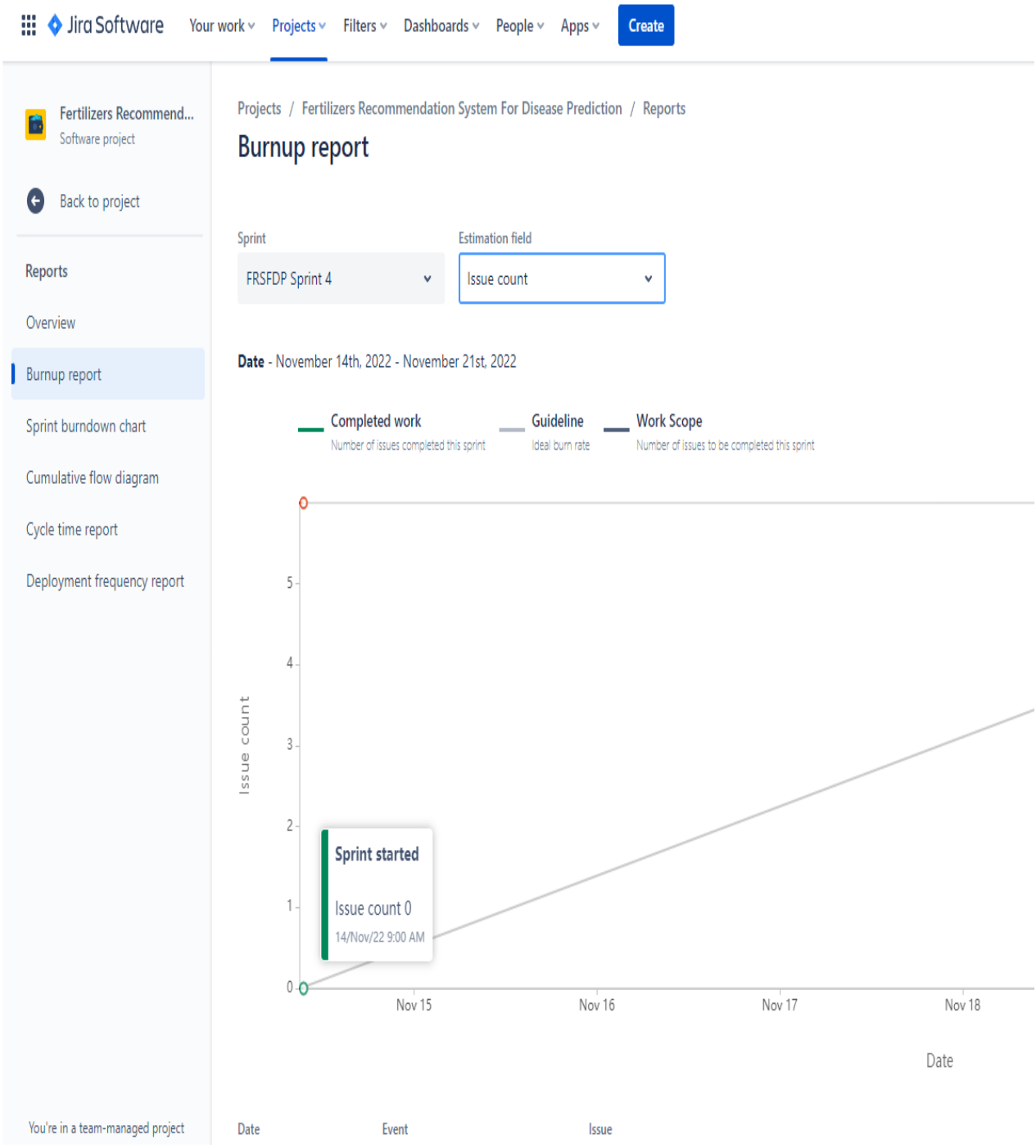
BACKLOG



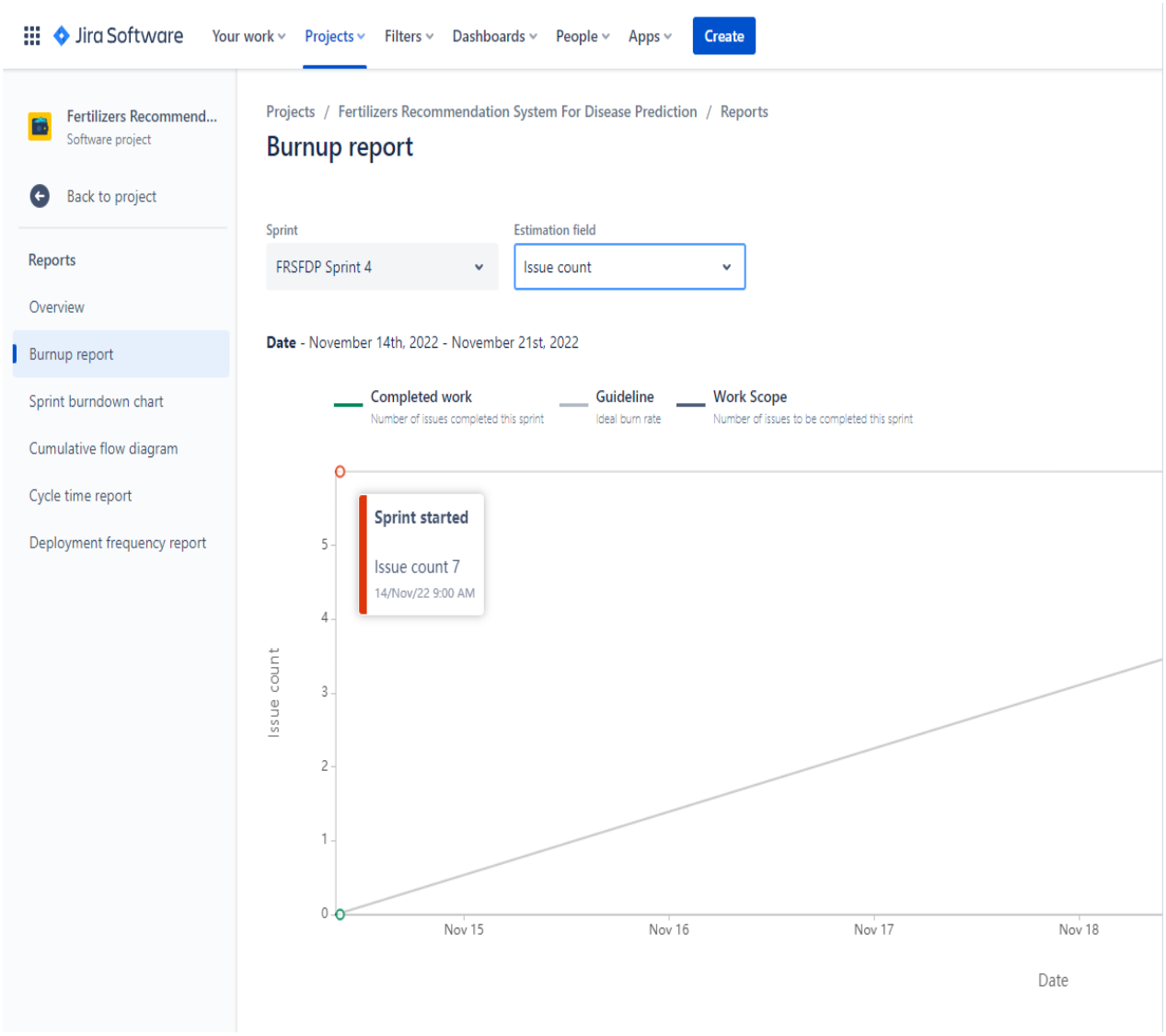
BOARD


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Fertilizers Recommend...
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / Fertilizers Recommendation System For Disease Prediction
All sprints
Epic
Sprint
TO DO 5 ISSUES
As a user, I can register for the application by entering my name, phone number, email, password, and confirming my password.
HOME PAGE
FRSFDP-9
As a user, I will receive confirmation email once I have registered for the application.
LOGIN
FRSFDP-10
As a user, I can log into the application by entering email & password.
LOGIN
FRSFDP-11
As a user, I can upload images of the affected leaves to the
IN PROGRESS 3 ISSUES
I need to deploy the Machine Learning model that was built.
MODEL DEPLOYMENT
FRSFDP-6
I need to build the website for the application using HTML, CSS etc.
APPLICATION BUILDING
FRSFDP-17
As a user, I can view the home page which has a description of the plant disease as well as options to sign up or log in.
HOME PAGE
FRSFDP-8
DONE 3 ISSUES
I need to collect data (images of leaf and with disease stages).
IMAGE PROCESSING
FRSFDP-1
I need to clean my data and prepare it for model building by doing pre -processing activities such as resizing etc.
IMAGE PROCESSING
FRSFDP-2
I need to build the model using Convolution Neural Network (CNN)
MODEL BUILDING
FRSFDP-4

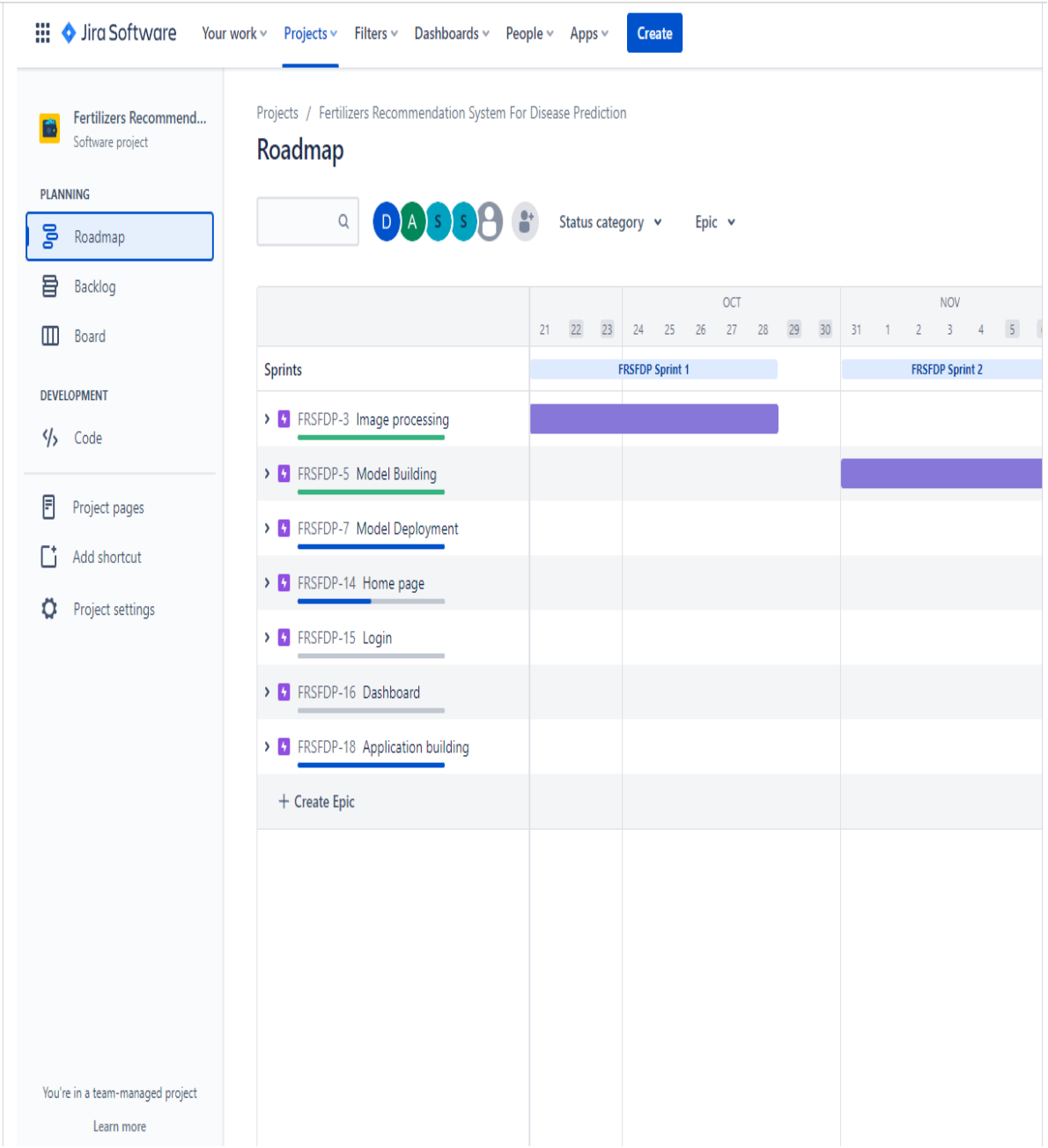
BURNUP CHART - 1



BURNUP CHART - 2



ROAD MAP


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Fertilizers Recommend...
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / Fertilizers Recommendation System For Disease Prediction
Roadmap
Status category
Epic
OCT
NOV
21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5
Sprints
FRSFDP Sprint 1
FRSFDP Sprint 2
FRSFDP-3 Image processing
FRSFDP-5 Model Building
FRSFDP-7 Model Deployment
FRSFDP-14 Home page
FRSFDP-15 Login
FRSFDP-16 Dashboard
FRSFDP-18 Application building
Create Epic

7. CODING AND SOLUTIONING:

7.1 FEATURE 1:

The leaf is uploaded in the prediction page and the page will ask seperately for the fruit and the vegetabe. According to the leaf category, the concerned leaf is uploaded in the page. After the leaf image is uploaded, the result is predicted for each of the leaf image. If the laef has no disease, it will indicate that the laef has no problem and the yield will be good. Suppose, if the leaf contains diseases, it will predict the disease. Also it will recommed the suitable fertilizer to solve the disease.

7.2 FEATURE 2:

The trained machine learning model can predict the output from an image that is uploaded, and the nutrition facts are also displayed on the same page. The model's accuracy was determined to be 95%, and when it was trained on the IBM cloud, it reached 100%. An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

7.3 DATABASE SCHEMA:

For designing the webpage, we have used HTML. We have developed the HTML page to structure the web page and used CSS(Cascading Style Sheets)

1_Histogram.ipynb

```
import cv2
import matplotlib.pyplot as plt
import numpy as np
img = cv2.imread('/content/drive/MyDrive/0a285c8b-1c31-48d4-89f2-
af8b9edc36f6___RS_HL 5759.JPG', 0)
plt.imshow(img, cmap='gray'), plt.grid(False)
plt.xticks([]), plt.yticks([])
hist = cv2.calcHist([img],[0],None,[50],[0,256])

# different methods for displaying a histogram
plt.bar(range(50), hist.ravel())
plt.title('Histogram of the airplane image')
plt.xlabel('Gray values')
plt.ylabel('Frequency')
# Another method
hist,bins = np.histogram(img.ravel(),256,[0,256])
plt.plot(hist)
```

```
# Let's read two other images
high = cv2.imread('/content/drive/MyDrive/aa04db6d-645f-4e8a-88dc-
c9f9396f0800___RS_HL 1975.jpg')
low = cv2.imread('/content/drive/MyDrive/0a285c8b-1c31-48d4-89f2-
af8b9edc36f6___RS_HL 5759.JPG')
# show images
plt.subplot(121), plt.imshow(high)
plt.grid(False), plt.xticks([]), plt.yticks([])

plt.subplot(122), plt.imshow(low)
plt.grid(False), plt.xticks([]), plt.yticks([])
plt.show()
# Calculate histogram of both images for the last channel.
# Channels can differ from 0 to 2.
hist_high = cv2.calcHist([high],[2],None,[256],[0,256])
hist_low = cv2.calcHist([low],[2],None,[256],[0,256])

# Plot histograms
plt.subplot(121)
plt.plot(hist_high)

plt.subplot(122)
plt.plot(hist_low)

plt.show()
cdf_low = hist_low.cumsum()
cdf_high = hist_high.cumsum()

# plot cumulative histograms
plt.subplot(221), plt.plot(cdf_high), plt. title('cdf of bright image')
plt.subplot(222), plt.plot(hist_high, 'k'), plt. title('pdf of bright image')

plt.subplot(223), plt.plot(cdf_low), plt. title('cdf of dark image')
plt.subplot(224), plt.plot(hist_low, 'k'), plt. title('pdf of dark image')

# adjust the placement of subplots
plt.subplots_adjust(bottom=2, right=0.8, top=3)

plt.show()
low_gray = cv2.cvtColor(low, cv2.COLOR_BGR2GRAY)
high_gray = cv2.cvtColor(high, cv2.COLOR_BGR2GRAY)
# show images and their histograms
plt.subplot(221), plt.imshow(high_gray, cmap='gray')
plt.grid(False), plt.xticks([]), plt.yticks([])
plt.subplot(223), plt.plot(cv2.calcHist([high_gray],[0],None,[256],[0,256]))

plt.subplot(222), plt.imshow(low_gray, cmap='gray')
```

```
plt.grid(False), plt.xticks([]), plt.yticks([])
plt.subplot(224), plt.plot(cv2.calcHist([low_gray],[0],None,[256],[0,256]))

plt.show()
# Define a function to easily handle manipulation.
def manip_image(image, alpha, beta):

  new_image = np.zeros(image.shape, image.dtype)

  for y in range(image.shape[0]):
    for x in range(image.shape[1]):
      new_image[y,x] = np.clip(alpha*image[y,x] + beta, 0, 255)

  return new_image
# Test on the image
bright = manip_image(img, 1, 30)
dark = manip_image(img, 1, -30)

# Compare the results
plt.figure()
plt.subplot(231), plt.imshow(dark, cmap='gray')
plt.grid(False), plt.xticks([]), plt.yticks([])

plt.subplot(232), plt.imshow(img, cmap='gray')
plt.grid(False), plt.xticks([]), plt.yticks([])


plt.subplot(233),plt.imshow(bright, cmap='gray')
plt.grid(False), plt.xticks([]), plt.yticks([])

plt.subplot(234)
plt.plot(cv2.calcHist([dark],[0],None,[256],[0,256])), plt.ylim((0, 1750))
plt.subplot(235)
plt.plot(cv2.calcHist([img],[0],None,[256],[0,256]))

plt.subplot(236)
plt.plot(cv2.calcHist([bright],[0],None,[256],[0,256]))
# Test on the dark image
l_bright = manip_image(low_gray, 1, 150)
l_dark = manip_image(low_gray, 1, -25)

# Compare the results
plt.figure()
plt.subplot(231), plt.imshow(l_dark, cmap='gray')
plt.grid(False), plt.xticks([]), plt.yticks([])

plt.subplot(232), plt.imshow(low_gray, cmap='gray')
```

```
plt.grid(False), plt.xticks([]), plt.yticks([])


plt.subplot(233),plt.imshow(l_bright, cmap='gray')
plt.grid(False), plt.xticks([]), plt.yticks([])

plt.subplot(234)
plt.plot(cv2.calcHist([l_dark],[0],None,[256],[0,256])), plt.ylim((0, 1100))

plt.subplot(235)
plt.plot(cv2.calcHist([low_gray],[0],None,[256],[0,256])), plt.ylim((0, 1100))

plt.subplot(236)
plt.plot(cv2.calcHist([l_bright],[0],None,[256],[0,256])), plt.ylim((0, 1100))
# Test on the image
increase_contrast = manip_image(img, 1.35, 0)
decrease_contrast = manip_image(img, 0.35, 0)

# Compare the results
plt.figure()
plt.subplot(231), plt.imshow(decrease_contrast, cmap='gray')
plt.grid(False), plt.xticks([]), plt.yticks([])

plt.subplot(232), plt.imshow(img, cmap='gray')
plt.grid(False), plt.xticks([]), plt.yticks([])


plt.subplot(233),plt.imshow(increase_contrast, cmap='gray')
plt.grid(False), plt.xticks([]), plt.yticks([])

plt.subplot(234)
plt.bar(range(256),
    cv2.calcHist([decrease_contrast],[0],None,[256],[0,256]).ravel())
plt.ylim((0, 1750))

plt.subplot(235)
plt.bar(range(256),
    cv2.calcHist([img],[0],None,[256],[0,256]).ravel())
plt.ylim((0, 1750))

plt.subplot(236)
plt.bar(range(256),
    cv2.calcHist([increase_contrast],[0],None,[256],[0,256]).ravel())
plt.ylim((0, 1750))
img_eq = cv2.equalizeHist(img)

grid = plt.GridSpec(3, 4, wspace=0.4, hspace=0.3)
```

```
plt.subplot(grid[:2, :2])
plt.imshow(img, cmap='gray')
plt.grid(False), plt.xticks([]), plt.yticks([])

plt.subplot(grid[:2, 2:])
plt.imshow(img_eq, cmap='gray')
plt.grid(False), plt.xticks([]), plt.yticks([])

plt.subplot(grid[2, :2])
plt.bar(range(256),
      cv2.calcHist([img],[0],None,[256],[0,256]).ravel())

plt.subplot(grid[2, 2:])
plt.bar(range(256),
      cv2.calcHist([img_eq],[0],None,[256],[0,256]).ravel())
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(16, 16))
img_cl = clahe.apply(img)

grid = plt.GridSpec(3, 4, wspace=0.4, hspace=0.3)

plt.subplot(grid[:2, :2])
plt.imshow(img, cmap='gray')
plt.grid(False), plt.xticks([]), plt.yticks([])

plt.subplot(grid[:2, 2:])
plt.imshow(img_cl, cmap='gray')
plt.grid(False), plt.xticks([]), plt.yticks([])

plt.subplot(grid[2, :2])
plt.bar(range(256),
      cv2.calcHist([img],[0],None,[256],[0,256]).ravel())

plt.subplot(grid[2, 2:])
plt.bar(range(256),
      cv2.calcHist([img_cl],[0],None,[256],[0,256]).ravel())
```

2_Spatial_Filtering.ipynb

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
ref = cv2.imread('/content/drive/MyDrive/0a285c8b-1c31-48d4-89f2-
af8b9edc36f6___RS_HL 5759.JPG')
plt.imshow(ref), plt.grid(False)

# while learning how to perform spatial filtering,
```

```
# you can also note how to apply different python commands.
plt.title('The original image')
plt.xticks([])
plt.yticks([])
plt.show()
# Defining a kernel using numpy.
kernel_5 = np.ones((5,5),np.float32)/25
kernel_3 = np.ones((3,3),np.float32)/9

# Convolves an image with the kernel.
# -1 means that the center of the kernel is located on the center pixel.
# compare two kernel sizes.
filtered_5 = cv2.filter2D(ref,-1,kernel_5)
filtered_3 = cv2.filter2D(ref,-1,kernel_3)

# plot the results in two subplots.
fig=plt.figure(figsize=(14, 14), dpi= 80, facecolor='w', edgecolor='k')

plt.subplot(121), plt.imshow(filtered_3), plt.title('3-by-3 filter')
plt.grid(False)
plt.xticks([])
plt.yticks([])

plt.subplot(122), plt.imshow(filtered_5), plt.title('5-by-5 filter')
plt.grid(False)
plt.xticks([])
plt.yticks([])

plt.show()
# you can check the docs for further information.
blurred = cv2.blur(ref, (5, 5), -1)

plt.imshow(blurred), plt.grid(False), plt.xticks([]), plt.yticks([]), plt.show()
top = 10; bottom = 5; left = 20; right = 5
const = 100
img2 = cv2.copyMakeBorder(ref, top, bottom, left, right,
                cv2.BORDER_WRAP)
img3 = cv2.copyMakeBorder(ref, top, bottom, left, right,
                cv2.BORDER_REFLECT)
img4 = cv2.copyMakeBorder(ref, top, bottom, left, right,
                cv2.BORDER_REPLICATE)
img5 = cv2.copyMakeBorder(ref, top, bottom, left, right,
                cv2.BORDER_CONSTANT, const)

# Display the images
fig=plt.figure(figsize=(14, 14), dpi= 80, facecolor='w', edgecolor='k')
```

```
plt.subplot(221), plt.imshow(img2), plt.grid(False)
plt.xticks([]), plt.yticks([]), plt.title('wrap')

plt.subplot(222), plt.imshow(img3), plt.grid(False)
plt.xticks([]), plt.yticks([]), plt.title('reflect')

plt.subplot(223), plt.imshow(img4), plt.grid(False)
plt.xticks([]), plt.yticks([]), plt.title('replicate')

plt.subplot(224), plt.imshow(img5), plt.grid(False)
plt.xticks([]), plt.yticks([]), plt.title('constant')

plt.show()
# vertical gradient kernel
# define a random kernel
vertical_gd = np.array([[1, 0, -1], [1, 0, -1], [1, 0, -1]])

# apply it.
filter_v = cv2.filter2D(ref[:, :, 2], -1, vertical_gd)

# show in a different colormap.
plt.imshow(filter_v, cmap='gray'), plt.grid(False)
plt.xticks([]), plt.yticks([])
s_and_p = np.random.rand(ref.shape[0], ref.shape[1])

# if we consider 5% salt and pepper noise, we'd like to have
# 2.5% salt and 2.5% pepper. thus:
salt = s_and_p > .975
pepper = s_and_p < .025

# in order to add some noise, we should turn off black (pepper) locations and
# turn on white (white) locations.
channel_2 = np.atleast_1d(ref[:, :, 1])
noisy = np.zeros_like(channel_2)

for i in range(channel_2.shape[0]*channel_2.shape[1]):
  if salt.ravel()[i] == 1:
    noisy.ravel()[i] = 255
  elif pepper.ravel()[i] == 1:
    noisy.ravel()[i] = 0
  else:
    noisy.ravel()[i] = channel_2.ravel()[i]

# apply median filter with size 3
Med = cv2.medianBlur(noisy, 3)

# Display the results
```

```
fig=plt.figure(figsize=(14, 14), dpi= 80, facecolor='w', edgecolor='k')
plt.subplot(121), plt.xticks([]), plt.yticks([])
plt.imshow(noisy, cmap='gray'), plt.grid(False)
plt.subplot(122), plt.xticks([]), plt.yticks([])
plt.imshow(Med, cmap='gray'), plt.grid(False)
plt.show()
# Creating random normal (gaussian) noise with pre-defined mean and std.
# The noisy image should be the size of the reference image.
mean = 0
sigma = 20.0
gauss_noise = np.random.normal(mean, sigma, (ref.shape[0], ref.shape[1]))

# Convert RGB image to Grayscale image using cvtColor()
gray = cv2.cvtColor(ref, cv2.COLOR_BGR2GRAY)

# Add gaussian noise to the image
g_noisy = gray + gauss_noise # Gaussian noisy image

# Showing gray image, noise image, and noisy image
fig=plt.figure(figsize=(14, 14), dpi= 80, facecolor='w', edgecolor='k')
plt.subplot(131), plt.xticks([]), plt.yticks([])
plt.imshow(gray, cmap='gray'), plt.grid(False)
plt.subplot(132), plt.xticks([]), plt.yticks([])
plt.imshow(gauss_noise, cmap='gray'), plt.grid(False)
plt.subplot(133), plt.xticks([]), plt.yticks([])
plt.imshow(g_noisy, cmap='gray'), plt.grid(False)
g_filtered = cv2.GaussianBlur(g_noisy, (3, 3), 20, 20)

# Display the result
plt.imshow(g_filtered, cmap='gray'), plt.grid(False)
plt.xticks([]), plt.yticks([])
# Create a single gaussian kernel
g_kernel = cv2.getGaussianKernel(3, 20)
print(g_kernel)

# Apply two separate kernels over the image.
g_filtered_2 = cv2.sepFilter2D(g_noisy, -1, g_kernel, g_kernel)

# Displaying the results.
fig=plt.figure(figsize=(14, 14), dpi= 80, facecolor='w', edgecolor='k')
plt.subplot(121), plt.xticks([]), plt.yticks([]), plt.title('first method')
plt.imshow(g_filtered, cmap='gray'), plt.grid(False)
plt.subplot(122), plt.xticks([]), plt.yticks([]), plt.title('second method')
plt.imshow(g_filtered_2, cmap='gray'), plt.grid(False)
```

3_Intensity_Transformations.ipynb

```
from skimage.io import imread
import matplotlib.pyplot as plt
import numpy as np
live = imread('/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple___Black_rot/0b37761a-de32-47ee-a3a4-e138b97ef542___JR_FrgE.S 2908.JPG')
mask = imread('/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple___Black_rot/00e909aa-e3ae-4558-9961-336bb0f35db3___JR_FrgE.S 8593.JPG')

plt.figure(figsize=(10, 10))
plt.subplot(121), plt.imshow(live, cmap='gray')
plt.subplot(122), plt.imshow(mask, cmap='gray')
plt.show()
plt.figure(figsize=(10, 10))
plt.subplot(121), plt.imshow(live, cmap='gray')
plt.subplot(122), plt.imshow(live - 20, cmap='gray')
plt.show()
plt.figure(figsize=(10, 10))
plt.subplot(131), plt.imshow(mask - live, cmap='gray')
plt.subplot(132), plt.imshow(-(mask - live + 128), cmap='gray')
plt.subplot(133), plt.imshow(mask - live + 128, cmap='gray')
plt.show()
shaded = imread('/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple___Black_rot/0b37761a-de32-47ee-a3a4-e138b97ef542___JR_FrgE.S 2908.JPG')
shading = imread('/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple___Black_rot/00e909aa-e3ae-4558-9961-336bb0f35db3___JR_FrgE.S 8593.JPG')

plt.figure(figsize=(10, 10))
plt.subplot(121), plt.imshow(shaded, cmap='gray')
plt.subplot(122), plt.imshow(shading, cmap='gray')
plt.show()
plt.figure(figsize=(10, 10))
plt.subplot(121), plt.imshow(np.multiply(shaded, 1/shading), cmap='gray')
plt.subplot(122), plt.imshow(shaded, cmap='gray')
plt.show()
# Test on the X-ray dental image
xray = imread('/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple___Black_rot/00e909aa-e3ae-4558-9961-336bb0f35db3___JR_FrgE.S 8593.JPG')
mask_xray = imread('/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Peach___Bacterial_spot/00ddc106-692e-4c67-b2e8-569c924caf49___Rutg._Bact.S 1228.JPG')
plt.figure(figsize=(10, 10))
```

```
plt.subplot(121), plt.imshow(xray, cmap='gray')
plt.subplot(122), plt.imshow(mask_xray, cmap='gray')
plt.show()
plt.figure()
plt.imshow(np.multiply(xray, mask_xray/255), cmap='gray')
plt.show()
# Test on another image
scan = imread('/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Corn_(maize)___Northern_Leaf_Blight/0a62fe5a-22db-42e2-bca0-53a8dcfd8129___RS_NLB 0810.JPG')
print(scan.shape)
# Showing the body scan image
plt.figure(figsize=(7, 7))
plt.imshow(scan, cmap='gray')
plt.show()
plt.figure(figsize=(10, 10))
plt.subplot(211), plt.imshow(xray, cmap='gray')
plt.subplot(212), plt.plot(np.histogram(xray, bins=256)[0])
plt.show()
```

Image Preprocessing.ipynb

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator (rescale = 1./255, shear_range= 0.2,zoom_range= 0.2, horizontal_flip = True)
test_datagen =ImageDataGenerator (rescale = 1)
x_train = train_datagen.flow_from_directory(r'/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/fruit-dataset/test',target_size = (128,128), batch_size = 32, class_mode = 'categorical')
x_test = test_datagen.flow_from_directory(r'/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/fruit-dataset/train',target_size = (128,128), batch_size = 32, class_mode = 'categorical')
x_train = train_datagen.flow_from_directory(r'/content/drive/MyDrive/DataSet/Dataset Plant Disease/Veg-dataset/Veg-dataset/test_set',target_size = (128,128), batch_size = 32, class_mode = 'categorical')
x_test = test_datagen.flow_from_directory(r'/content/drive/MyDrive/DataSet/Dataset Plant Disease/Veg-dataset/Veg-dataset/test_set',target_size = (128,128), batch_size = 32, class_mode = 'categorical')
```

TRAINING THE MODELS

FruitData.ipynb

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten
```

```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,
vertical_flip=False)
test_datagen=ImageDataGenerator(rescale=1./255)
x_train=train_datagen.flow_from_directory(r"/content/drive/MyDrive/DataSet/Dataset
Plant Disease/fruit-dataset/fruit-dataset/train",target_size=(128,128),
                                          class_mode='categorical',batch_size=24)
x_test=test_datagen.flow_from_directory(r"/content/drive/MyDrive/DataSet/Dataset Plant
Disease/fruit-dataset/fruit-dataset/test",target_size=(128,128),
                                          class_mode='categorical',batch_size=24)
model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.summary()
32*(3*3*3+1)
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(6,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
len(x_train)
model.fit(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=le
n(x_test),epochs=10)
model.save('fruitdata.h5')
```

Vegetable Data.ipynb

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,
vertical_flip=False)
test_datagen=ImageDataGenerator(rescale=1./255)
x_train=train_datagen.flow_from_directory(r"/content/drive/MyDrive/DataSet/Dataset
Plant Disease/Veg-dataset/Veg-dataset/train_set",target_size=(128,128),
                                          class_mode='categorical',batch_size=24)
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/DataSet/Dataset Plant
Disease/Veg-dataset/Veg-dataset/test_set',target_size=(128,128),
                                          class_mode='categorical',batch_size=24)
model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
```

```
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.summary()
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(9,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
model.fit(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=le
n(x_test),epochs=10)
model.save('vegetabledata.h5')
```

TESTING THE MODELS

Tested_fruitdat.ipynb

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
test_dir=r'/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/fruit-
dataset/test'
model = tf.keras.models.load_model(r'/content/drive/MyDrive/DataSet/Dataset Plant
Disease/fruit-dataset/fruitdata.h5')
test_datagen_1=ImageDataGenerator(rescale=1)
test_generator_1=test_datagen_1.flow_from_directory(
   test_dir,
   target_size=(128,128),
   batch_size=20,
   class_mode='categorical'
)
img=image.load_img(r"/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-
dataset/fruit-dataset/test/Apple___healthy/00fca0da-2db3-481b-b98a-
9b67bb7b105c___RS_HL 7708.JPG",target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
index=['Apple___Black_rot', 'Apple___healthy', 'Corn_(maize)___healthy',
'Corn_(maize)___Northern_Leaf_Blight', 'Peach___Bacterial_spot', 'Peach___healthy']
index[y[0]]
model.evaluate(test_generator_1,steps=50)
```

Tested_vegetabledata.ipynb

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
test_dir=r'/content/drive/MyDrive/DataSet/Dataset Plant Disease/Veg-dataset/Veg-
dataset/test_set'
model = tf.keras.models.load_model(r'/content/drive/MyDrive/DataSet/Dataset Plant
Disease/Veg-dataset/vegetabledata.h5')
test_datagen_1=ImageDataGenerator(rescale=1)
test_generator_1=test_datagen_1.flow_from_directory(
   test_dir,
   target_size=(128,128),
   batch_size=20,
   class_mode='categorical'
)
img=image.load_img(r"/content/drive/MyDrive/DataSet/Dataset Plant Disease/Veg-
dataset/Veg-dataset/test_set/Pepper,_bell___healthy/b303761b-5357-4d82-9e78-
1b26c2804196___JR_HL 7879.JPG",target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
index=['Pepper,_bell___Bacterial_spot', 'Pepper,_bell___healthy', 'Potato___Early_blight',
'Potato___healthy', 'Potato___Late_blight',
'Tomato___Bacterial_spot','Tomato___Late_blight','Tomato___Leaf_Mold','Tomato___Se
ptoria_leaf_spot']
y[0]
index[y[0]]
```

APPLICATION BUILDING

app.py

```
#Import necessary libraries
from flask import Flask, render_template, request

import numpy as np
import os

from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
```

```
filepath =
'C:/Users/Anandh/AppData/Local/Programs/Python/Python38/Tomato_Leaf_Disease_Pred
iction/fruitdata.h5'
model = load_model(filepath)
print(model)

print("Model Loaded Successfully")

def pred_tomato_dieas(plant):
  test_image = load_img(plant, target_size = (128, 128)) # load image
  print("@@ Got Image for prediction")

  test_image = img_to_array(test_image)/255 # convert image to np array and normalize
  test_image = np.expand_dims(test_image, axis = 0) # change dimention 3D to 4D

  result = model.predict(test_image) # predict diseased palnt or not
  print('@@ Raw result = ', result)

  pred = np.argmax(result, axis=1)
  print(pred)
  if plant=="fruit":
    if pred==0:
       return "Apple__Black_rot", 'Apple__Black_rot.html'

    elif pred==1:
       return "Apple__healthy", 'Apple__healthy.html'

    elif pred==2:
       return "Corn_(maize)__healthy", 'Corn_(maize)__healthy.html'

    elif pred==3:
       return "Corn_(maize)__Northern_Leaf_Blight",
'Corn_(maize)__Northern_Leaf_Blight.html'

    elif pred==4:
       return "Peach__Bacterial_spot", 'Peach__Bacterial_spot.html'

    elif pred==5:
       return "Peach__healthy", 'Peach__healthy.html'
  elif plant=="Vegetable":
    if pred==0:
       return "Pepper,_bell__Bacterial_spot", 'Pepper,_bell__Bacterial_spot.html'

    elif pred==1:
       return "Pepper,_bell__healthy", 'Pepper,_bell__healthy.html'

    elif pred==2:
```

```
        return "Potato__Early_blight", 'Potato__Early_blight.html'

    elif pred==3:
        return "Potato__healthy", 'Potato__healthy.html'

    elif pred==4:
        return "Potato__Late_blight", 'Potato__Late_blight.html'

    elif pred==5:
        return "Tomato__Bacterial_spot", 'Tomato__Bacterial_spot.html'

    elif pred==6:
        return "Tomato__Late_blight" , 'Tomato__Late_blight.html'

    elif pred==7:
        return "Tomato__Leaf_Mold" , 'Tomato__Leaf_Mold.html'

    elif pred==8:
        return "Tomato__Septoria_leaf_spot" , 'Tomato__Septoria_leaf_spot.html'




# Create flask instance
app = Flask(__name__)

# render index.html page
@app.route("/", methods=['GET', 'POST'])
def home():
        return render_template('index.html')


# get input image from client then predict class and render respective .html page for
solution
@app.route("/predict", methods = ['GET','POST'])
def predict():
    if request.method == 'POST':
        file = request.files['image'] # fet input
        filename = file.filename
        print("@@ Input posted = ", filename)

        file_path =
os.path.join('C:/Users/Madhuri/AppData/Local/Programs/Python/Python38/Tomato_Leaf
_Disease_Prediction/static/upload/', filename)
        file.save(file_path)

        print("@@ Predicting class.....")
```

```
    pred, output_page = pred_tomato_dieas(tomato_plant=file_path)

    return render_template(output_page, pred_output = pred, user_image = file_path)

# For local system & cloud
if __name__ == "__main__":
   app.run(threaded=False,port=8080)
```

HTML AND CSS CODE

index.html

```
<html>
<head>
<style>

* {
        margin: 0px;
        padding: 0px;
        box-sizing: border-box;


    }



    form {
        display: flex;
        height: 85vh;
        justify-content: center;
        align-items: center;
        margin-top: -150px;
        width: 60%;
        text-align: center;
                        margin-left:300px;

    }


    .details h2 {

        position: relative;
        top: 100px;
        margin: auto;
        color: rgb(18, 231, 231);
        font-size: 3rem;
    }
```

```
label:hover {
   transform: scale(1.03);
}

.details h2 {
   /* margin-bottom: 300px; */
   position: relative;
   top: 100px;
   margin: auto;
   color: rgb(18, 231, 231);
   font-size: 3rem;

}




.details h1 {
   color: white;
   padding: 20px;
   border-radius: 15px;
   background-color: rgb(8, 8, 8);
}


.upload {

   font-size: 20px;
   background-color: rgb(255, 252, 252);
   border-radius: 20px;
   outline: none;
   width: 500px;
   color: rgb(0, 0, 0);
   border: 3px solid rgb(45, 47, 49);
}

.lload {
   position: relative;
   top: 40px; left: 390px;
   font-size: 20px;
   background-color: rgb(255, 252, 252);
   border-radius: 20px;
   outline: none;
   width: 500px;
   color: rgb(0, 0, 0);
   border: 3px solid rgb(45, 47, 49);
```

```
    }

    .txt {
       position: relative;
       top: 40px; left: 390px;
    }

    ::-webkit-file-upload-button {
       color: rgb(255, 252, 252);
       padding: 20px;
       border: 2px solid rgb(201, 6, 6);
       background-color: rgb(201, 6, 6);
       border-radius: 15px;

    }

    ::-webkit-file-upload-button:hover {
       border-radius: 20px;
       border: 2px solid rgb(177, 174, 174);


    }



    input[type="submit"] {
       position: absolute;
       margin-top: 200px;
       padding: 15px 35px;
       background-color: rgb(31, 185, 190);
       border-radius: 15px;
       color: black;
       font-size: 1.5rem;
       border: 4px solid rgb(31, 185, 190);
    }
</style>
</head>
<body style="background-image: url('Background.jpeg');">
        <h1 style="text-align:center;font-size:4rem;">Predict Plant Disease & Get
Cure</h1>
<section>


  <label for="cars" class = "txt" >Choose your plant:</label>
  <select name="cars" id="cars" class="lload">
    <option value="Fruit">Fruit plant</option>
```

```
      <option value="Vegetable">vegetable plant</option>
   </select>
         <form action="/predict" method="post" enctype="multipart/form-data"
onsubmit="showloading()">

            <br>
            <input type="file" name="image" class="upload">
                              <br>
                              <br>
            <input type="submit" value="Predict">

         </form>
      </div>
   </section>

</body>
</html>

Pepper_bell_bacterial_spot.html
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">

   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
      integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
crossorigin="anonymous">

   <title>PLANT DISEASE PREDICTION</title>

   <style>
      * {
         margin: 0px;
         padding: 0px;
         box-sizing: border-box;

      }

      .border img {
         border-radius: 15px;
         border: 2px solid black;
      }
```

```
   </style>
</head>

<body style="background-image: url('https://media.istockphoto.com/photos/tractor-spraying-pesticides-on-soybean-field-with-sprayer-at-spring-picture-id966855552?b=1&k=20&m=966855552&s=170667a&w=0&h=xtZdO1-cST47qsG2rGkhCpyC7poG3kGcTrYVC3djIG0=');background-size: cover;background-attachment: fixed;">

   <div>
      <h1 style="text-align:left;font-size: 50px;padding: 20px;color: white;">Predict Disease And Get Cure</h1>
   </div>

  <br>
  <br>


   <div class="container my-2">
      <div class="row mb-5">

         <div class="col-sm" style="margin-bottom: 23px;">
            <span class="border border-primary">
               <img src="{{url_for('static',filename = 'images/Tomato___Leaf_Mold.JPG' )}}" alt="Image Not Found" style="width:500px;height:500px;">


            </span>
         </div>

         <div class="col-sm">

            <div>
               <h1 style="padding: 15px; background-color:rgb(95, 231, 95); color: white;"
                  class="text-center mb-5 content-h1 rounded">
                  {{pred_output}} </h1>
            </div>
            <div class="details">
                                       <div class="box" style="background-color: aliceblue;width: 200px;border-radius: 10px;position: relative;left: 200px;top: 72px;">
                  <p style="text-align: center;font-size: 30px;color: black"><b>Treatment</b></p>
               </div>

               <div class="box-sol" style="background-color: white;height: 200px;width: 600px;border-radius: 10px;position: relative;top: 50px;">
```

```
                <div class="mypara" style="height: 150;width: 550px; position: relative;left:
20px;top: 30px;">
                    <p style="text-align: center;position: relative;top: 0px;font-size: 15.5px;">
                     Control of bacterial spot on greenhouse transplants is an essential step for
preventing the spread of the leaf spot bacteria in the field. Transplants should be inspected
regularly to identify symptomatic seedlings. Transplants with symptoms may be removed
and destroyed or treated with streptomycin, if detected at the very early stage of disease
development. It should be noted that strains of leaf spot bacteria resistant to streptomycin
may arise with multiple applications of streptomycin

                </p>
                </div>

              </div>
            </div>
          </div>
        </div>


    </div>
</body>

</html>
```

pepper_bell_healthy.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
      integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
crossorigin="anonymous">

    <title>PLANT DISEASE PREDICTION</title>

    <style>
        * {
            margin: 0px;
            padding: 0px;
            box-sizing: border-box;
```

```
    }

    .border img {
       border-radius: 15px;
       border: 2px solid black;
    }

  </style>
</head>

<body style="background-image: url('https://media.istockphoto.com/photos/tractor-
spraying-pesticides-on-soybean-field-with-sprayer-at-spring-picture-
id966855552?b=1&k=20&m=966855552&s=170667a&w=0&h=xtZdO1-
cST47qsG2rGkhCpyC7poG3kGcTrYVC3djIG0=');background-size: cover;background-
attachment: fixed;">

  <div>
    <h1 style="text-align:left;font-size: 50px;padding: 20px;color: white;">Predict
Disease And Get Cure</h1>
  </div>

 <br>
 <br>


  <div class="container my-2">
    <div class="row mb-5">

      <div class="col-sm" style="margin-bottom: 23px;">
        <span class="border border-primary">
          <img src="{{url_for('static',filename = 'images/Tomato___Leaf_Mold.JPG'
)}}" alt="Image Not Found" style="width:500px;height:500px;">


        </span>
      </div>

      <div class="col-sm">

        <div>
          <h1 style="padding: 15px; background-color:rgb(95, 231, 95); color: white;"
            class="text-center mb-5 content-h1 rounded">
            {{pred_output}} </h1>
        </div>
        <div class="details">
                                    <div class="box" style="background-color:
aliceblue;width: 200px;border-radius: 10px;position: relative;left: 200px;top: 72px;">
```

```
                <p style="text-align: center;font-size: 30px;color:
black"><b>Treatment</b></p>
            </div>

            <div class="box-sol" style="background-color: white;height: 200px;width:
600px;border-radius: 10px;position: relative;top: 50px;">
                <div class="mypara" style="height: 150;width: 550px; position: relative;left:
20px;top: 30px;">
                  <p style="text-align: center;position: relative;top: 50px;font-size: 22px;">
                    This leaf is <span style="color: green;"><b>healthy</b></span>.
                </p>
                </div>

            </div>
          </div>
        </div>
      </div>


   </div>
</body>

</html>
```

Potato_early_blight.html

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">

   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
      integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
crossorigin="anonymous">

   <title>PLANT DISEASE PREDICTION</title>

   <style>
      * {
         margin: 0px;
         padding: 0px;
         box-sizing: border-box;
```

```
    }

    .border img {
      border-radius: 15px;
      border: 2px solid black;
    }

  </style>
</head>

<body style="background-image: url('https://media.istockphoto.com/photos/tractor-spraying-pesticides-on-soybean-field-with-sprayer-at-spring-picture-id966855552?b=1&k=20&m=966855552&s=170667a&w=0&h=xtZdO1-cST47qsG2rGkhCpyC7poG3kGcTrYVC3djIG0=');background-size: cover;background-attachment: fixed;">

  <div>
    <h1 style="text-align:left;font-size: 50px;padding: 20px;color: white;">Predict Disease And Get Cure</h1>
  </div>

 <br>
 <br>


  <div class="container my-2">
    <div class="row mb-5">

      <div class="col-sm" style="margin-bottom: 23px;">
        <span class="border border-primary">
          <img src="{{url_for('static',filename = 'images/Tomato___Leaf_Mold.JPG' )}}" alt="Image Not Found" style="width:500px;height:500px;">


        </span>
      </div>

      <div class="col-sm">

        <div>
          <h1 style="padding: 15px; background-color:rgb(95, 231, 95); color: white;"
            class="text-center mb-5 content-h1 rounded">
            {{pred_output}} </h1>
        </div>
        <div class="details">
                                    <div class="box" style="background-color: aliceblue;width: 200px;border-radius: 10px;position: relative;left: 200px;top: 72px;">
```

```
                <p style="text-align: center;font-size: 30px;color:
black"><b>Treatment</b></p>
            </div>

            <div class="box-sol" style="background-color: white;height: 200px;width:
600px;border-radius: 10px;position: relative;top: 50px;">
                <div class="mypara" style="height: 150;width: 550px; position: relative;left:
20px;top: 30px;">
                  <p style="text-align: center; font-size: 15.5px;">
                   Early blight can be minimized by maintaining optimum growing
conditions, including proper fertilization, irrigation, and management of other pests. Grow
later maturing, longer season varieties. Fungicide application is justified only when the
disease is initiated early enough to cause economic loss.
                </p>
                </div>

            </div>
          </div>
        </div>
      </div>


   </div>
</body>

</html>
```

Potato_healthy.html

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">

   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
      integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
crossorigin="anonymous">

   <title>PLANT DISEASE PREDICTION</title>

   <style>
      * {
         margin: 0px;
         padding: 0px;
```

```
        box-sizing: border-box;

      }

      .border img {
        border-radius: 15px;
        border: 2px solid black;
      }

   </style>
</head>

<body style="background-image: url('https://media.istockphoto.com/photos/tractor-
spraying-pesticides-on-soybean-field-with-sprayer-at-spring-picture-
id966855552?b=1&k=20&m=966855552&s=170667a&w=0&h=xtZdO1-
cST47qsG2rGkhCpyC7poG3kGcTrYVC3djIG0=');background-size: cover;background-
attachment: fixed;">

   <div>
      <h1 style="text-align:left;font-size: 50px;padding: 20px;color: white;">Predict
Disease And Get Cure</h1>
   </div>

  <br>
  <br>


   <div class="container my-2">
      <div class="row mb-5">

         <div class="col-sm" style="margin-bottom: 23px;">
            <span class="border border-primary">
               <img src="{{url_for('static',filename = 'images/Tomato___Leaf_Mold.JPG'
)}}" alt="Image Not Found" style="width:500px;height:500px;">


            </span>
         </div>

         <div class="col-sm">

            <div>
               <h1 style="padding: 15px; background-color:rgb(95, 231, 95); color: white;"
                  class="text-center mb-5 content-h1 rounded">
                  {{pred_output}} </h1>
            </div>
            <div class="details">
```

```
                                    <div class="box" style="background-color:
aliceblue;width: 200px;border-radius: 10px;position: relative;left: 200px;top: 72px;">
                <p style="text-align: center;font-size: 30px;color:
black"><b>Treatment</b></p>
              </div>

              <div class="box-sol" style="background-color: white;height: 200px;width:
600px;border-radius: 10px;position: relative;top: 50px;">
                <div class="mypara" style="height: 150;width: 550px; position: relative;left:
20px;top: 30px;">
                  <p style="text-align: center;position: relative;top: 50px;font-size: 22px;">
                    This leaf is <span style="color: green;"><b>healthy</b></span>.
                </p>
                </div>

              </div>
            </div>
          </div>
        </div>


    </div>
</body>

</html>
```

Potato_late_blight.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
      integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
crossorigin="anonymous">

    <title>PLANT DISEASE PREDICTION</title>

    <style>
        * {
            margin: 0px;
            padding: 0px;
            box-sizing: border-box;
```

```
    }

    .border img {
      border-radius: 15px;
      border: 2px solid black;
    }

  </style>
</head>

<body style="background-image: url('https://media.istockphoto.com/photos/tractor-spraying-pesticides-on-soybean-field-with-sprayer-at-spring-picture-id966855552?b=1&k=20&m=966855552&s=170667a&w=0&h=xtZdO1-cST47qsG2rGkhCpyC7poG3kGcTrYVC3djIG0=');background-size: cover;background-attachment: fixed;">

  <div>
    <h1 style="text-align:left;font-size: 50px;padding: 20px;color: white;">Predict Disease And Get Cure</h1>
  </div>

 <br>
 <br>


  <div class="container my-2">
    <div class="row mb-5">

      <div class="col-sm" style="margin-bottom: 23px;">
        <span class="border border-primary">
          <img src="{{url_for('static',filename = 'images/Tomato___Leaf_Mold.JPG' )}}" alt="Image Not Found" style="width:500px;height:500px;">


        </span>
      </div>

      <div class="col-sm">

        <div>
          <h1 style="padding: 15px; background-color:rgb(95, 231, 95); color: white;"
            class="text-center mb-5 content-h1 rounded">
            {{pred_output}} </h1>
        </div>
        <div class="details">
                                    <div class="box" style="background-color: aliceblue;width: 200px;border-radius: 10px;position: relative;left: 200px;top: 72px;">
```

```
                <p style="text-align: center;font-size: 30px;color:
black"><b>Treatment</b></p>
            </div>

            <div class="box-sol" style="background-color: white;height: 200px;width:
600px;border-radius: 10px;position: relative;top: 50px;">
                <div class="mypara" style="height: 150;width: 550px; position: relative;left:
20px;top: 30px;">
                 <p style="text-align: center; font-size: 15.5px;">
                  Late blight is controlled by eliminating cull piles and volunteer potatoes,
using proper harvesting and storage practices, and applying fungicides when necessary.
Air drainage to facilitate the drying of foliage each day is important. Under marginal
conditions, overhead sprinkler irrigation can favor late blight; in Tule Lake under solid set
sprinklers, conditions conducive to late blight development are enhanced by day time
irrigation but not night time irrigation.
                </p>
                </div>

            </div>
          </div>
        </div>
      </div>


   </div>
</body>

</html>

Tomato_bacterial_spot.html

<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">

   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
      integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
crossorigin="anonymous">

   <title>PLANT DISEASE PREDICTION</title>

   <style>
```

```
      * {
         margin: 0px;
         padding: 0px;
         box-sizing: border-box;

      }

      .border img {
         border-radius: 15px;
         border: 2px solid black;
      }

   </style>
</head>

<body style="background-image: url('https://media.istockphoto.com/photos/tractor-
spraying-pesticides-on-soybean-field-with-sprayer-at-spring-picture-
id966855552?b=1&k=20&m=966855552&s=170667a&w=0&h=xtZdO1-
cST47qsG2rGkhCpyC7poG3kGcTrYVC3djIG0=');background-size: cover;background-
attachment: fixed;">

   <div>
      <h1 style="text-align:left;font-size: 50px;padding: 20px;color: white;">Predict
Disease And Get Cure</h1>
   </div>

  <br>
  <br>


   <div class="container my-2">
      <div class="row mb-5">

         <div class="col-sm" style="margin-bottom: 23px;">
            <span class="border border-primary">
               <img src="{{url_for('static',filename = 'images/Tomato___Leaf_Mold.JPG'
)}}" alt="Image Not Found" style="width:500px;height:500px;">


            </span>
         </div>

         <div class="col-sm">

            <div>
               <h1 style="padding: 15px; background-color:rgb(95, 231, 95); color: white;"
                  class="text-center mb-5 content-h1 rounded">
                  {{pred_output}} </h1>
```

```
            </div>
            <div class="details">
                                        <div class="box" style="background-color:
aliceblue;width: 200px;border-radius: 10px;position: relative;left: 200px;top: 72px;">
                    <p style="text-align: center;font-size: 30px;color:
black"><b>Treatment</b></p>
                </div>

                <div class="box-sol" style="background-color: white;height: 200px;width:
600px;border-radius: 10px;position: relative;top: 50px;">
                    <div class="mypara" style="height: 150;width: 550px; position: relative;left:
20px;top: 30px;">
                      <p style="text-align: center; font-size: 15.5px;">
                        Plant pathogen-free seed or transplants to prevent the introduction of
bacterial spot pathogens on contaminated seed or seedlings.  If a clean seed source is not
available or you suspect that your seed is contaminated, soak seeds in water at 122°F for 25
min. to kill the pathogens.
                    </p>
                    </div>

                </div>
            </div>
          </div>
        </div>


    </div>
</body>

</html>
```

Tomato_late_blight.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
        integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
crossorigin="anonymous">

    <title>PLANT DISEASE PREDICTION</title>
```

```
    <style>
        * {
            margin: 0px;
            padding: 0px;
            box-sizing: border-box;

        }

        .border img {
            border-radius: 15px;
            border: 2px solid black;
        }

    </style>
</head>

<body style="background-image: url('https://media.istockphoto.com/photos/tractor-
spraying-pesticides-on-soybean-field-with-sprayer-at-spring-picture-
id966855552?b=1&k=20&m=966855552&s=170667a&w=0&h=xtZdO1-
cST47qsG2rGkhCpyC7poG3kGcTrYVC3djIG0=');background-size: cover;background-
attachment: fixed;">

    <div>
        <h1 style="text-align:left;font-size: 50px;padding: 20px;color: white;">Predict
Disease And Get Cure</h1>
    </div>

    <br>
    <br>


    <div class="container my-2">
        <div class="row mb-5">

            <div class="col-sm" style="margin-bottom: 23px;">
                <span class="border border-primary">
                    <img src="{{url_for('static',filename = 'images/Tomato___Leaf_Mold.JPG'
)}}" alt="Image Not Found" style="width:500px;height:500px;">


                </span>
            </div>

            <div class="col-sm">

                <div>
                    <h1 style="padding: 15px; background-color:rgb(95, 231, 95); color: white;"
                        class="text-center mb-5 content-h1 rounded">
```

```
                {{pred_output}} </h1>
            </div>
            <div class="details">
                                        <div class="box" style="background-color:
aliceblue;width: 200px;border-radius: 10px;position: relative;left: 200px;top: 72px;">
                <p style="text-align: center;font-size: 30px;color:
black"><b>Treatment</b></p>
              </div>

              <div class="box-sol" style="background-color: white;height: 200px;width:
600px;border-radius: 10px;position: relative;top: 50px;">
                <div class="mypara" style="height: 150;width: 550px; position: relative;left:
20px;top: 30px;">
                  <p style="text-align: center; font-size: 15.5px;">
                    Continue weekly spray applications to protect plants from further
infection. Severely infected plants can be rogued and either buried or burned. Avoid
composting diseased plants
                </p>
                </div>

              </div>
            </div>
          </div>
        </div>


    </div>
</body>

</html>
```

Tomato_leaf_mold.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
        integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
crossorigin="anonymous">

    <title>PLANT DISEASE PREDICTION</title>
```

```
    <style>
        * {
            margin: 0px;
            padding: 0px;
            box-sizing: border-box;

        }

        .border img {
            border-radius: 15px;
            border: 2px solid black;
        }

    </style>
</head>

<body style="background-image: url('https://media.istockphoto.com/photos/tractor-spraying-pesticides-on-soybean-field-with-sprayer-at-spring-picture-id966855552?b=1&k=20&m=966855552&s=170667a&w=0&h=xtZdO1-cST47qsG2rGkhCpyC7poG3kGcTrYVC3djIG0=');background-size: cover;background-attachment: fixed;">

    <div>
        <h1 style="text-align:left;font-size: 50px;padding: 20px;color: white;">Predict Disease And Get Cure</h1>
    </div>

    <br>
    <br>


    <div class="container my-2">
        <div class="row mb-5">

            <div class="col-sm" style="margin-bottom: 23px;">
                <span class="border border-primary">
                    <img src="{{url_for('static',filename = 'images/Tomato___Leaf_Mold.JPG' )}}" alt="Image Not Found" style="width:500px;height:500px;">


                </span>
            </div>

            <div class="col-sm">

                <div>
                    <h1 style="padding: 15px; background-color:rgb(95, 231, 95); color: white;"
```

```
                class="text-center mb-5 content-h1 rounded">
                {{pred_output}} </h1>
            </div>
            <div class="details">
                                        <div class="box" style="background-color:
aliceblue;width: 200px;border-radius: 10px;position: relative;left: 200px;top: 72px;">
                <p style="text-align: center;font-size: 30px;color:
black"><b>Treatment</b></p>
              </div>

              <div class="box-sol" style="background-color: white;height: 200px;width:
600px;border-radius: 10px;position: relative;top: 50px;">
                <div class="mypara" style="height: 150;width: 550px; position: relative;left:
20px;top: 30px;">
                  <p style="text-align: center; font-size: 15.5px;position:relative;top: 50px;">
                    Maintain night temps higher than outside temperatures.

                </p>
                </div>

              </div>
            </div>
          </div>
        </div>


    </div>
</body>

</html>
```

Tomato_septoria_leaf_spot.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
      integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
crossorigin="anonymous">

    <title>PLANT DISEASE PREDICTION</title>
```

```
    <style>
        * {
            margin: 0px;
            padding: 0px;
            box-sizing: border-box;

        }

        .border img {
            border-radius: 15px;
            border: 2px solid black;
        }

    </style>
</head>

<body style="background-image: url('https://media.istockphoto.com/photos/tractor-
spraying-pesticides-on-soybean-field-with-sprayer-at-spring-picture-
id966855552?b=1&k=20&m=966855552&s=170667a&w=0&h=xtZdO1-
cST47qsG2rGkhCpyC7poG3kGcTrYVC3djIG0=');background-size: cover;background-
attachment: fixed;">

    <div>
        <h1 style="text-align:left;font-size: 50px;padding: 20px;color: white;">Predict
Disease And Get Cure</h1>
    </div>

    <br>
    <br>


    <div class="container my-2">
        <div class="row mb-5">

            <div class="col-sm" style="margin-bottom: 23px;">
                <span class="border border-primary">
                    <img src="{{url_for('static',filename = 'images/Tomato___Leaf_Mold.JPG'
)}}" alt="Image Not Found" style="width:500px;height:500px;">


                </span>
            </div>

            <div class="col-sm">

                <div>
                    <h1 style="padding: 15px; background-color:rgb(95, 231, 95); color: white;"
                        class="text-center mb-5 content-h1 rounded">
```

```
                {{pred_output}} </h1>
            </div>
            <div class="details">
                                        <div class="box" style="background-color:
aliceblue;width: 200px;border-radius: 10px;position: relative;left: 200px;top: 72px;">
                <p style="text-align: center;font-size: 30px;color:
black"><b>Treatment</b></p>
              </div>

              <div class="box-sol" style="background-color: white;height: 200px;width:
600px;border-radius: 10px;position: relative;top: 50px;">
                <div class="mypara" style="height: 150;width: 550px; position: relative;left:
20px;top: 30px;">
                  <p style="text-align: center; font-size: 15.5px;">
                    Improve air circulation around the plants. If the plants can still be handled
without breaking them, stake or cage the plants to raise them off the ground and promote
faster drying of the foliage.
                </p>
                </div>

              </div>
            </div>
          </div>
        </div>


    </div>
</body>

</html>
```

Apple_Black_Rot.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
        integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
crossorigin="anonymous">

    <title>PLANT DISEASE PREDICTION</title>
```

```
    <style>
        * {
            margin: 0px;
            padding: 0px;
            box-sizing: border-box;

        }

        .border img {
            border-radius: 15px;
            border: 2px solid black;
        }

    </style>
</head>

<body style="background-image: url('https://media.istockphoto.com/photos/tractor-
spraying-pesticides-on-soybean-field-with-sprayer-at-spring-picture-
id966855552?b=1&k=20&m=966855552&s=170667a&w=0&h=xtZdO1-
cST47qsG2rGkhCpyC7poG3kGcTrYVC3djIG0=');background-size: cover;background-
attachment: fixed;">

    <div>
        <h1 style="text-align:left;font-size: 50px;padding: 20px;color: white;">Predict
Disease And Get Cure</h1>
    </div>

    <br>
    <br>


    <div class="container my-2">
        <div class="row mb-5">

            <div class="col-sm" style="margin-bottom: 23px;">
                <span class="border border-primary">
                    <img src="{{url_for('static',filename = 'images/Tomato___Leaf_Mold.JPG'
)}}" alt="Image Not Found" style="width:500px;height:500px;">


                </span>
            </div>

            <div class="col-sm">

                <div>
                    <h1 style="padding: 15px; background-color:rgb(95, 231, 95); color: white;"
```

```
                class="text-center mb-5 content-h1 rounded">
                {{pred_output}} </h1>
            </div>
            <div class="details">
                                        <div class="box" style="background-color:
aliceblue;width: 200px;border-radius: 10px;position: relative;left: 200px;top: 72px;">
                <p style="text-align: center;font-size: 30px;color:
black"><b>Treatment</b></p>
              </div>

              <div class="box-sol" style="background-color: white;height: 200px;width:
600px;border-radius: 10px;position: relative;top: 50px;">
                <div class="mypara" style="height: 150;width: 550px; position: relative;left:
20px;top: 30px;">
                  <p style="text-align: center;">
                    Carefully prune and dispose of dead wood. This should be an important
component of both current-season and long-range management.
                    Prune and remove cankers; properly dispose of prunings by burial or
burning.
                    Remove all mummified fruit.
                    Control fire blight by pruning out infected wood or controlling insect
vectors
                </p>
                </div>

              </div>
            </div>
          </div>
        </div>


    </div>
</body>

</html>
```

Apple_Healthy.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
```

```
    integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
crossorigin="anonymous">

  <title>PLANT DISEASE PREDICTION</title>

  <style>
    * {
      margin: 0px;
      padding: 0px;
      box-sizing: border-box;

    }

    .border img {
      border-radius: 15px;
      border: 2px solid black;
    }

  </style>
</head>

<body style="background-image: url('https://media.istockphoto.com/photos/tractor-
spraying-pesticides-on-soybean-field-with-sprayer-at-spring-picture-
id966855552?b=1&k=20&m=966855552&s=170667a&w=0&h=xtZdO1-
cST47qsG2rGkhCpyC7poG3kGcTrYVC3djIG0=');background-size: cover;background-
attachment: fixed;">

  <div>
    <h1 style="text-align:left;font-size: 50px;padding: 20px;color: white;">Predict
Disease And Get Cure</h1>
  </div>

 <br>
 <br>


  <div class="container my-2">
    <div class="row mb-5">

      <div class="col-sm" style="margin-bottom: 23px;">
        <span class="border border-primary">
          <img src="{{url_for('static',filename = 'images/Tomato___Leaf_Mold.JPG'
)}}" alt="Image Not Found" style="width:500px;height:500px;">


        </span>
      </div>
```

```
        <div class="col-sm">

            <div>
              <h1 style="padding: 15px; background-color:rgb(95, 231, 95); color: white;"
                class="text-center mb-5 content-h1 rounded">
                {{pred_output}} </h1>
            </div>
            <div class="details">
                                        <div class="box" style="background-color:
aliceblue;width: 200px;border-radius: 10px;position: relative;left: 200px;top: 72px;">
                <p style="text-align: center;font-size: 30px;color:
black"><b>Treatment</b></p>
              </div>

              <div class="box-sol" style="background-color: white;height: 200px;width:
600px;border-radius: 10px;position: relative;top: 50px;">
                <div class="mypara" style="height: 150;width: 550px; position: relative;left:
20px;top: 30px;">
                  <p style="text-align: center;position: relative;top: 50px;font-size: 22px;">
                    This leaf is <span style="color: green;"><b>healthy</b></span>.
                </p>
                </div>

              </div>
            </div>
          </div>
        </div>


    </div>
</body>

</html>
```

Corn(maize)_Healthy.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
```

```
    integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
crossorigin="anonymous">

  <title>PLANT DISEASE PREDICTION</title>

  <style>
    * {
      margin: 0px;
      padding: 0px;
      box-sizing: border-box;

    }

    .border img {
      border-radius: 15px;
      border: 2px solid black;
    }

  </style>
</head>

<body style="background-image: url('https://media.istockphoto.com/photos/tractor-
spraying-pesticides-on-soybean-field-with-sprayer-at-spring-picture-
id966855552?b=1&k=20&m=966855552&s=170667a&w=0&h=xtZdO1-
cST47qsG2rGkhCpyC7poG3kGcTrYVC3djIG0=');background-size: cover;background-
attachment: fixed;">

  <div>
    <h1 style="text-align:left;font-size: 50px;padding: 20px;color: white;">Predict
Disease And Get Cure</h1>
  </div>

 <br>
 <br>


  <div class="container my-2">
    <div class="row mb-5">

      <div class="col-sm" style="margin-bottom: 23px;">
        <span class="border border-primary">
          <img src="{{url_for('static',filename = 'images/Tomato___Leaf_Mold.JPG'
)}}" alt="Image Not Found" style="width:500px;height:500px;">


        </span>
      </div>
```

```
        <div class="col-sm">

            <div>
              <h1 style="padding: 15px; background-color:rgb(95, 231, 95); color: white;"
                class="text-center mb-5 content-h1 rounded">
                {{pred_output}} </h1>
            </div>
            <div class="details">
                                        <div class="box" style="background-color:
aliceblue;width: 200px;border-radius: 10px;position: relative;left: 200px;top: 72px;">
                <p style="text-align: center;font-size: 30px;color:
black"><b>Treatment</b></p>
              </div>

              <div class="box-sol" style="background-color: white;height: 200px;width:
600px;border-radius: 10px;position: relative;top: 50px;">
                <div class="mypara" style="height: 150;width: 550px; position: relative;left:
20px;top: 30px;">
                  <p style="text-align: center;position: relative;top: 50px;font-size: 22px;">
                    This leaf is <span style="color: green;"><b>healthy</b></span>.
                </p>
                </div>

              </div>
            </div>
          </div>
        </div>

    </div>
</body>

</html>

Corn(maize)_northern_leaf_blight.html

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
```

```
    integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
crossorigin="anonymous">

  <title>PLANT DISEASE PREDICTION</title>

  <style>
    * {
      margin: 0px;
      padding: 0px;
      box-sizing: border-box;

    }

    .border img {
      border-radius: 15px;
      border: 2px solid black;
    }

  </style>
</head>

<body style="background-image: url('https://media.istockphoto.com/photos/tractor-
spraying-pesticides-on-soybean-field-with-sprayer-at-spring-picture-
id966855552?b=1&k=20&m=966855552&s=170667a&w=0&h=xtZdO1-
cST47qsG2rGkhCpyC7poG3kGcTrYVC3djIG0=');background-size: cover;background-
attachment: fixed;">

  <div>
    <h1 style="text-align:left;font-size: 50px;padding: 20px;color: white;">Predict
Disease And Get Cure</h1>
  </div>

 <br>
 <br>


  <div class="container my-2">
    <div class="row mb-5">

      <div class="col-sm" style="margin-bottom: 23px;">
        <span class="border border-primary">
          <img src="{{url_for('static',filename = 'images/Tomato___Leaf_Mold.JPG'
)}}" alt="Image Not Found" style="width:500px;height:500px;">


        </span>
      </div>
```

```
        <div class="col-sm">

            <div>
              <h1 style="padding: 15px; background-color:rgb(95, 231, 95); color: white;"
                class="text-center mb-5 content-h1 rounded">
                {{pred_output}} </h1>
            </div>
            <div class="details">
                                        <div class="box" style="background-color:
aliceblue;width: 200px;border-radius: 10px;position: relative;left: 200px;top: 72px;">
                <p style="text-align: center;font-size: 30px;color:
black"><b>Treatment</b></p>
              </div>

              <div class="box-sol" style="background-color: white;height: 200px;width:
600px;border-radius: 10px;position: relative;top: 50px;">
                <div class="mypara" style="height: 150;width: 550px; position: relative;left:
20px;top: 30px;">
                  <p style="text-align: center;position: relative;top: 0px;font-size: 18px;">
                    First, choose corn varieties or hybrids that are resistant or at least have
moderate resistance to northern corn leaf blight. When you grow corn, make sure it does
not stay wet for long periods of time. The fungus that causes this infection needs between
six and 18 hours of leaf wetness to develop. Plant corn with enough space for airflow and
water in the morning so leaves can dry throughout the day.

                </p>
                </div>

              </div>
            </div>
          </div>
        </div>


    </div>
</body>

</html>
```

Peach_bacterial_spot.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
        integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
crossorigin="anonymous">

    <title>PLANT DISEASE PREDICTION</title>

    <style>
        * {
            margin: 0px;
            padding: 0px;
            box-sizing: border-box;

        }

        .border img {
            border-radius: 15px;
            border: 2px solid black;
        }

    </style>
</head>

<body style="background-image: url('https://media.istockphoto.com/photos/tractor-
spraying-pesticides-on-soybean-field-with-sprayer-at-spring-picture-
id966855552?b=1&k=20&m=966855552&s=170667a&w=0&h=xtZdO1-
cST47qsG2rGkhCpyC7poG3kGcTrYVC3djIG0=');background-size: cover;background-
attachment: fixed;">

    <div>
        <h1 style="text-align:left;font-size: 50px;padding: 20px;color: white;">Predict
Disease And Get Cure</h1>
    </div>

   <br>
   <br>


    <div class="container my-2">
        <div class="row mb-5">

            <div class="col-sm" style="margin-bottom: 23px;">
                <span class="border border-primary">
```

```
            <img src="{{url_for('static',filename = 'images/Tomato___Leaf_Mold.JPG'
)}}" alt="Image Not Found" style="width:500px;height:500px;">


          </span>
        </div>

        <div class="col-sm">

          <div>
            <h1 style="padding: 15px; background-color:rgb(95, 231, 95); color: white;"
              class="text-center mb-5 content-h1 rounded">
              {{pred_output}} </h1>
          </div>
          <div class="details">
                                        <div class="box" style="background-color:
aliceblue;width: 200px;border-radius: 10px;position: relative;left: 200px;top: 72px;">
              <p style="text-align: center;font-size: 30px;color:
black"><b>Treatment</b></p>
              </div>

              <div class="box-sol" style="background-color: white;height: 200px;width:
600px;border-radius: 10px;position: relative;top: 50px;">
                <div class="mypara" style="height: 150;width: 550px; position: relative;left:
20px;top: 30px;">
                  <p style="text-align: center; font-size: 15.5px;">
                    The best strategy is to use cultivars with better bacterial spot resistance.
This is especially true for orchards in sandy sites prone to the disease. Varieties developed
by breeding programs in wet, sandy regions will tend to have better bacterial spot
resistance than those developed for dry regions such as California. A variety with moderate
resistance to bacterial spot may do well in a site sheltered from wind but have unacceptable
symptoms in a bacterial spot-prone site or if planted with highly susceptible varieties.
                </p>
                </div>

              </div>
            </div>
          </div>
        </div>


    </div>
</body>

</html>
```

Peach_healthy.html

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">

   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
      integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
crossorigin="anonymous">

   <title>PLANT DISEASE PREDICTION</title>

   <style>
      * {
         margin: 0px;
         padding: 0px;
         box-sizing: border-box;

      }

      .border img {
         border-radius: 15px;
         border: 2px solid black;
      }

   </style>
</head>

<body style="background-image: url('https://media.istockphoto.com/photos/tractor-
spraying-pesticides-on-soybean-field-with-sprayer-at-spring-picture-
id966855552?b=1&k=20&m=966855552&s=170667a&w=0&h=xtZdO1-
cST47qsG2rGkhCpyC7poG3kGcTrYVC3djIG0=');background-size: cover;background-
attachment: fixed;">

   <div>
      <h1 style="text-align:left;font-size: 50px;padding: 20px;color: white;">Predict
Disease And Get Cure</h1>
   </div>

  <br>
  <br>


   <div class="container my-2">
```

```
      <div class="row mb-5">

         <div class="col-sm" style="margin-bottom: 23px;">
            <span class="border border-primary">
               <img src="{{url_for('static',filename = 'images/Tomato___Leaf_Mold.JPG'
)}}" alt="Image Not Found" style="width:500px;height:500px;">


            </span>
         </div>

         <div class="col-sm">

            <div>
               <h1 style="padding: 15px; background-color:rgb(95, 231, 95); color: white;"
                  class="text-center mb-5 content-h1 rounded">
                  {{pred_output}} </h1>
            </div>
            <div class="details">
                                       <div class="box" style="background-color:
aliceblue;width: 200px;border-radius: 10px;position: relative;left: 200px;top: 72px;">
                  <p style="text-align: center;font-size: 30px;color:
black"><b>Treatment</b></p>
               </div>

               <div class="box-sol" style="background-color: white;height: 200px;width:
600px;border-radius: 10px;position: relative;top: 50px;">
                  <div class="mypara" style="height: 150;width: 550px; position: relative;left:
20px;top: 30px;">
                    <p style="text-align: center;position: relative;top: 50px;font-size: 22px;">
                     This leaf is <span style="color: green;"><b>healthy</b></span>.
                  </p>
                  </div>

               </div>
            </div>
         </div>
      </div>


   </div>
</body>

</html>
```

8. TESTING

8.1 TEST CASES

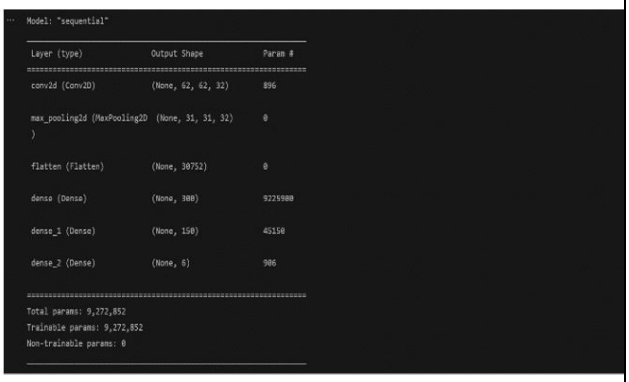
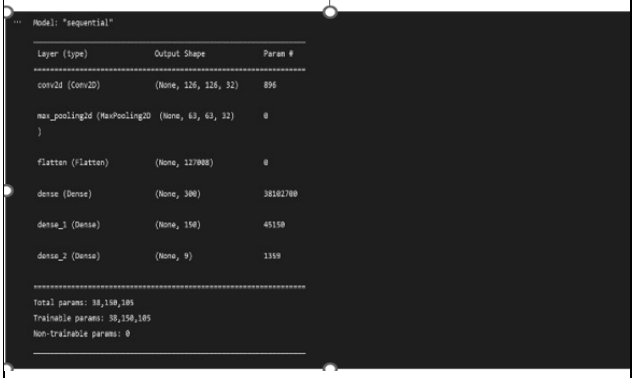
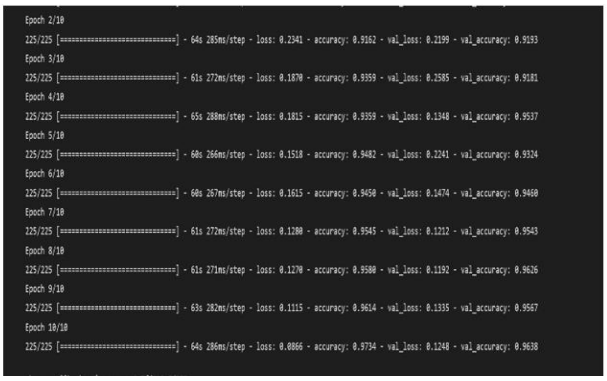
Test Case	Feature	Components	Test Scenario	Expected Result	Actual Result	Status	Comments	Bug	Executed by
Home Page	Functional	Home Page	Verify User Can See the Home Page	Yes, The Option Is Available	Option is Available	Pass	Success	No	Akalya S
Prediction For Fruit	Functional	Home Page	Verify User Can See the Home Page	Can Visible	Yes Visible	Pass	Success	No	Divya K
Prediction For Vegetable	Functional	Home Page	Verify User Can See the Home Page	Can Visible	Yes Visible	Pass	Success	No	Sowndarya S
Backend Process	Functional	Python Coding	Coding is done to implement the application	Working	Working	Pass	Success	No	Swetha M

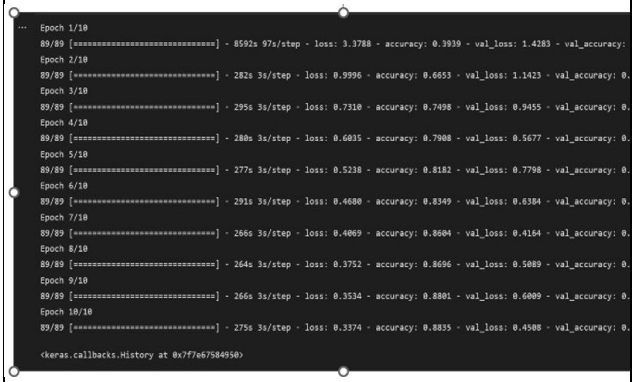
Results:

Fertilizer Recommendation System using Artificial Intelligence is developed and executed at the level of completed progress.

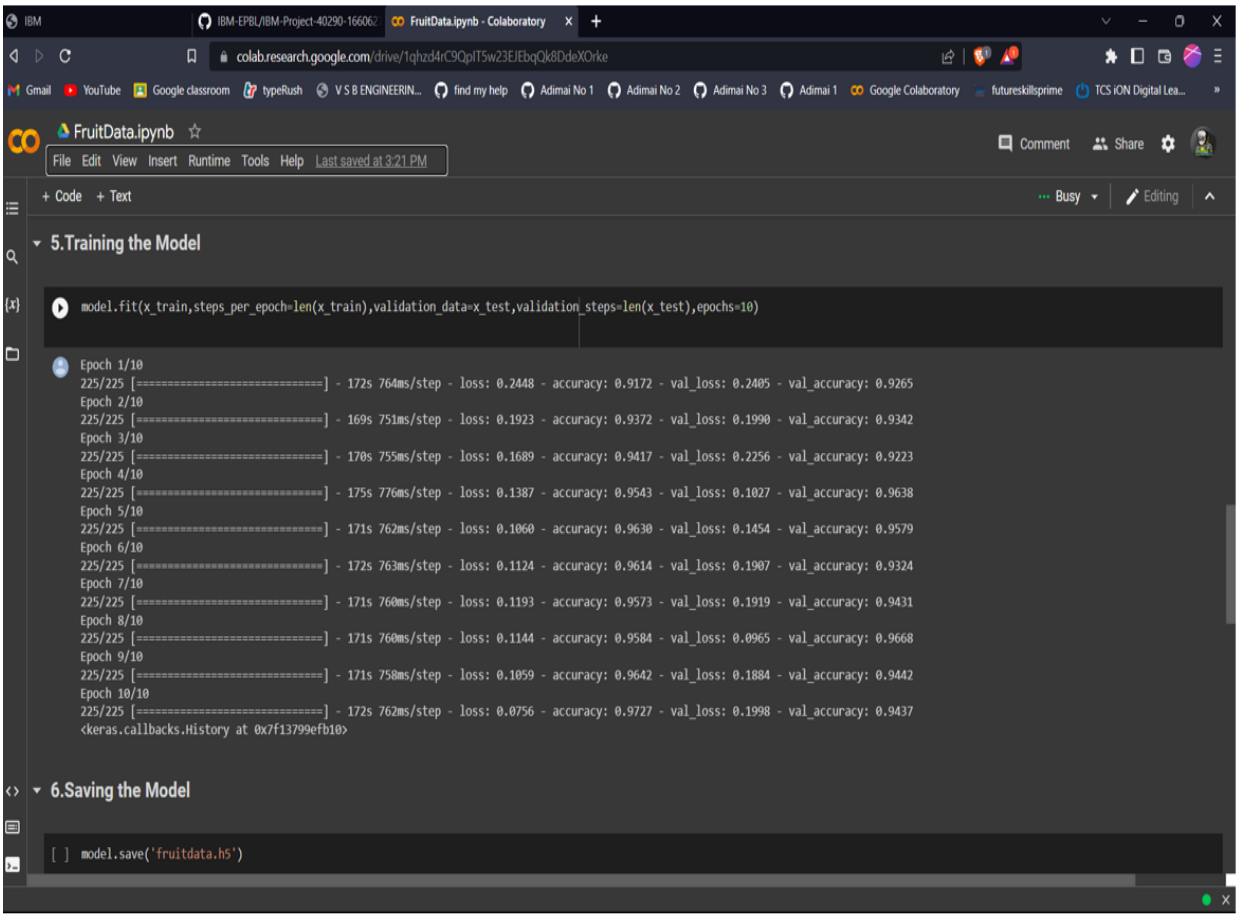
9. RESULTS

9.1 PERFORMANCE METRICS

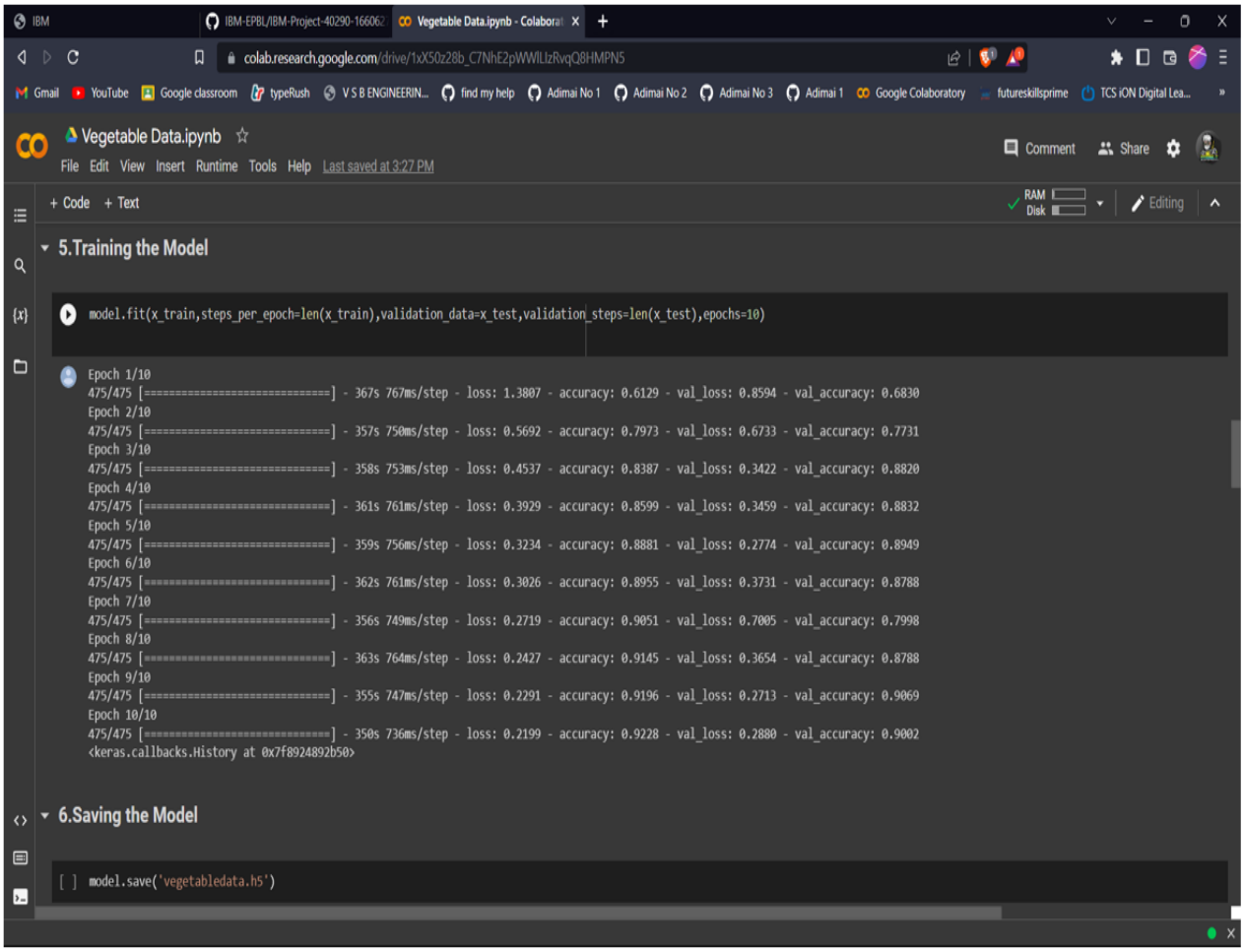
S. No.	Parameter	Values	Screenshot
1.	Model Summary of Fruit	Training the dataset of Vegetable images by usingthe CNN models to predictthe disease of the given leaves.	
2.	Model Summary for Vegetable	Training the dataset of Vegetable images by usingthe CNN models to predictthe disease of the given leaves.	
3.	Accuracy forFruit	Training Accuracy - 0.9734 Validation Accuracy - 0.9638	

4.	Accuracy for Vegetable	Training Accuracy - 0.8835 Validation Accuracy - 0.8448	

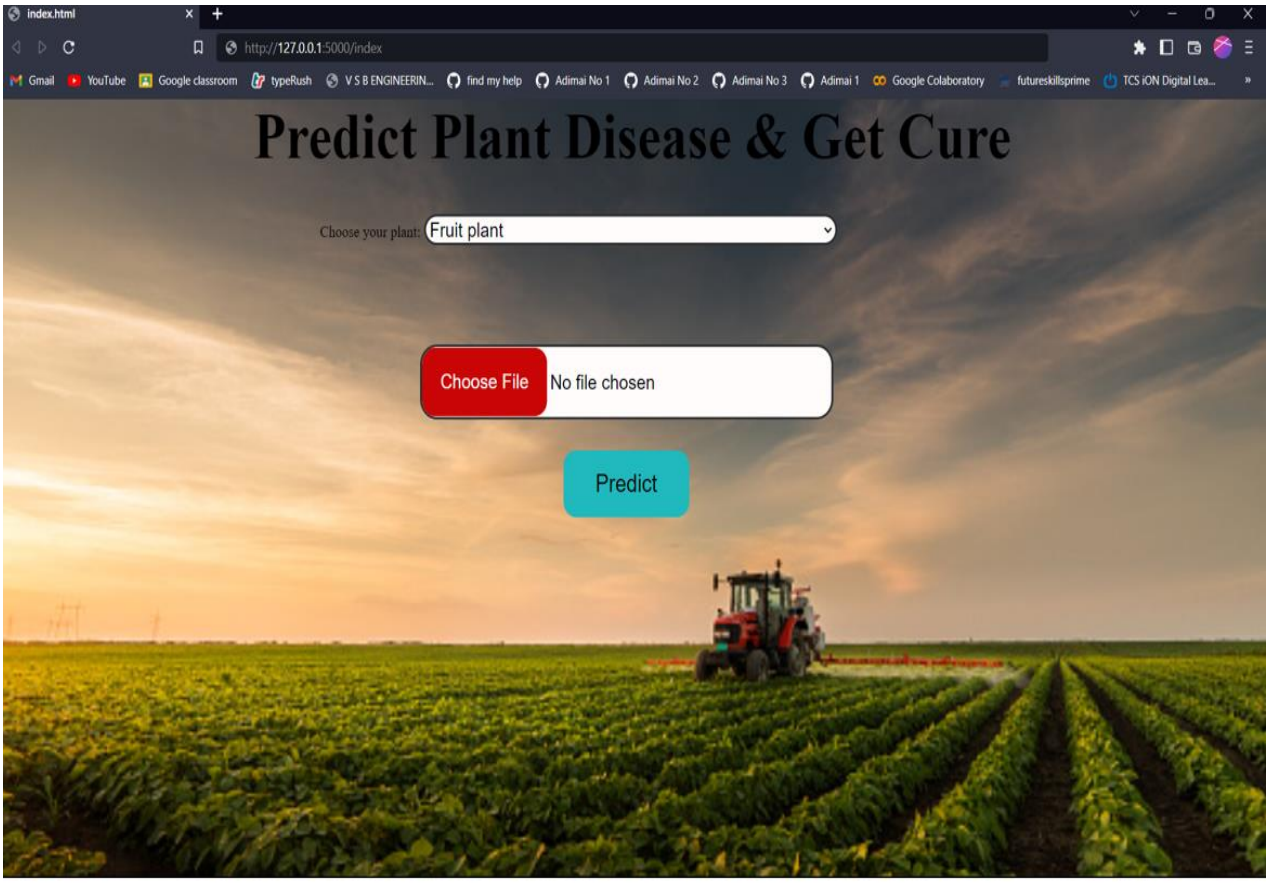
Training Fruits Dataset:



Training Vegetable Dataset:

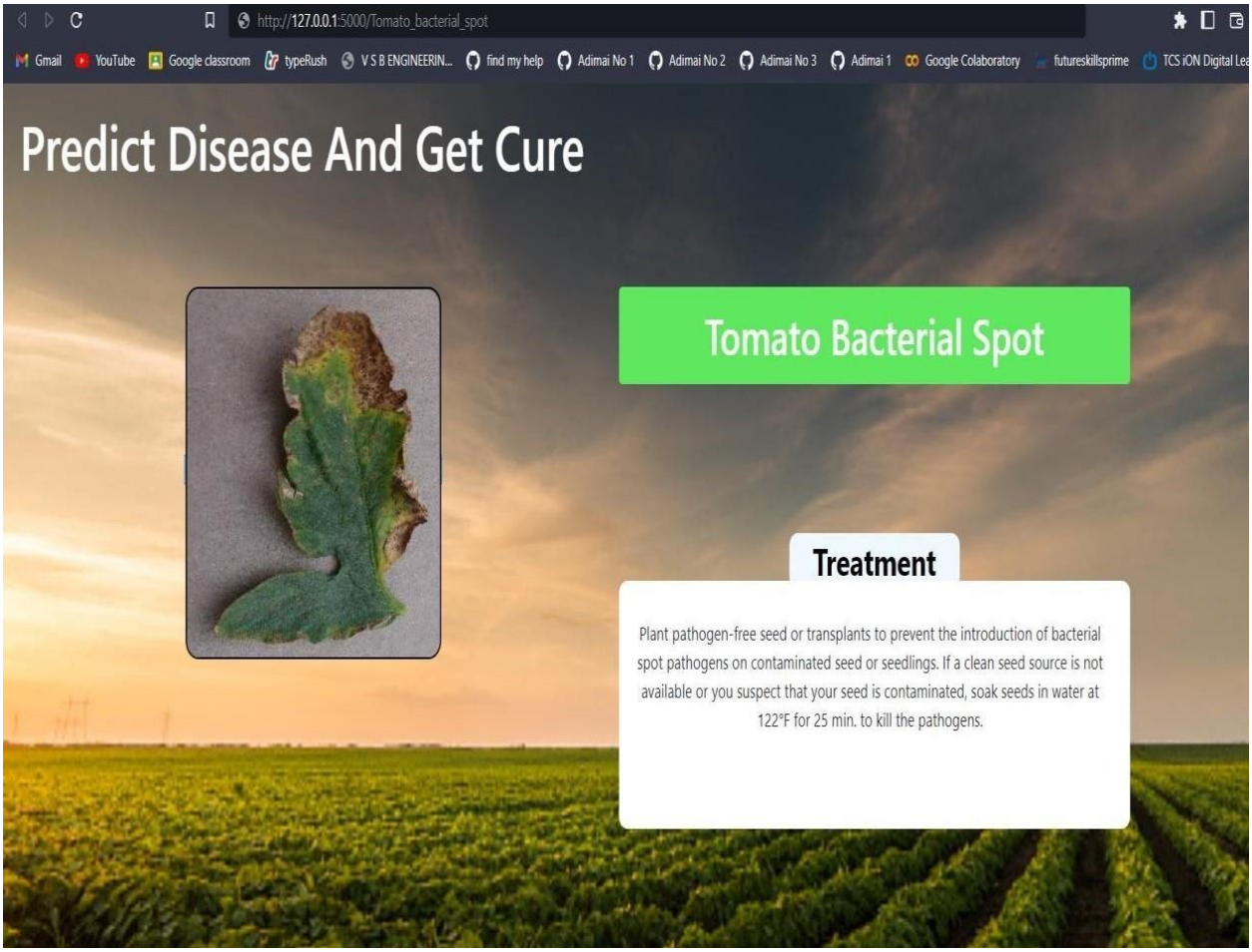


Login Page


index.html
http://127.0.0.1:5000/index
Gmail
YouTube
Google classroom
typeRush
V S B ENGINEERIN...
find my help
Adimai No 1
Adimai No 2
Adimai No 3
Adimai 1
Google Colaboratory
futureskillsprime
TCS iON Digital Lea...
Predict Plant Disease & Get Cure
Choose your plant:
Fruit plant
Choose File
No file chosen
Predict

Predicting the Fruit Leaf:

Predicting the Vegetable Leaf:



10. ADVANTAGES & DISADVANTAGES

Advantages:

1. Disease prediction is easy
2. User Friendly application
3. Saves time for the farmer
4. Cost reduction
5. No unusual cost for unwanted fertilizers
6. Pre-processing the Image with high capability
7. It gives good result to farmers

Disadvantages:

1. Language is provided only in English
2. Some may face difficulty in this software
3. Sometimes it might predict wrongly due to poor image
4. Using AI some decisions may get confused to humans
5. Internet facility is mandatory
6. If software crash occurs, correct prediction is not possible

11. CONCLUSION:

The model proposed here involves image classification of fruit datasets and vegetable datasets. The following points are observed during model testing and training:

- The accuracy of classification increased by increasing the number of epochs.
- For different batch sizes, different classification accuracies are obtained.
- The accuracies are increased by increasing more convolution layers.
- The accuracy of classification also increased by varying dense layers.
- Different accuracies are obtained by varying the size of kernel used in the convolution layer output.
- Accuracies are different while varying the size of the train and test datasets.

12. FUTURE SCOPE:

The model that is being provided in this project work can be expanded to recognize images. Using python to exe software, the complete model may be turned into application software. With the aid of the OpenCV Python package, real-time image categorization, picture recognition, and video processing are all made feasible. This project's work can be expanded to include security applications including face, iris, and figure print recognition. The system successfully interprets various Diseases and is also capable of providing fertilizers suggestion for the respective disease. Furthermore, this system can be made more robustby incorporating more image datasetwith wider variations like morethan one leaf in a single image.An App could also be developed for the projectwhichcould make the work of the farmers easier. They could directly upload image on the app and it would tell the disease and the cure then and there. This would reduce the time and efforts. This project is limited to just one crop for now but in the future more crops and even flowersdataset can be added so that it is helpfulfor every agriculturalneed. Newer models can also be added and tried with time which may result in better accuracy.

13. APPENDIX:

Source Code:

```
#Import necessary libraries
from flask import Flask, render_template, request

import numpy as np
import os

from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model

filepath =
'C:/Users/Swetha/AppData/Local/Programs/Python/Python38/Tomato_Leaf_Disease_Pre
diction/fruitdata.h5'
model = load_model(filepath)
print(model)

print("Model Loaded Successfully")

def pred_tomato_dieas(plant):
  test_image = load_img(plant, target_size = (128, 128)) # load image
  print("@@ Got Image for prediction")

  test_image = img_to_array(test_image)/255 # convert image to np array and normalize
  test_image = np.expand_dims(test_image, axis = 0) # change dimention 3D to 4D

  result = model.predict(test_image) # predict diseased palnt or not
```

```
  print('@@ Raw result = ', result)

  pred = np.argmax(result, axis=1)
  print(pred)
  if plant=="fruit":
    if pred==0:
       return "Apple__Black_rot", 'Apple__Black_rot.html'

    elif pred==1:
       return "Apple__healthy", 'Apple__healthy.html'

    elif pred==2:
       return "Corn_(maize)__healthy", 'Corn_(maize)__healthy.html'

    elif pred==3:
       return "Corn_(maize)__Northern_Leaf_Blight",
'Corn_(maize)__Northern_Leaf_Blight.html'

    elif pred==4:
       return "Peach__Bacterial_spot", 'Peach__Bacterial_spot.html'

    elif pred==5:
       return "Peach__healthy", 'Peach__healthy.html'
  elif plant=="Vegetable":
    if pred==0:
       return "Pepper,_bell__Bacterial_spot", 'Pepper,_bell__Bacterial_spot.html'

    elif pred==1:
       return "Pepper,_bell__healthy", 'Pepper,_bell__healthy.html'
```

```
    elif pred==2:
        return "Potato__Early_blight", 'Potato__Early_blight.html'

    elif pred==3:
        return "Potato__healthy", 'Potato__healthy.html'

    elif pred==4:
        return "Potato__Late_blight", 'Potato__Late_blight.html'

    elif pred==5:
        return "Tomato__Bacterial_spot", 'Tomato__Bacterial_spot.html'

    elif pred==6:
        return "Tomato__Late_blight" , 'Tomato__Late_blight.html'

    elif pred==7:
        return "Tomato__Leaf_Mold" , 'Tomato__Leaf_Mold.html'

    elif pred==8:
        return "Tomato__Septoria_leaf_spot" , 'Tomato__Septoria_leaf_spot.html'




# Create flask instance
app = Flask(__name__)
```

```
# render index.html page
@app.route("/", methods=['GET', 'POST'])
def home():
        return render_template('index.html')



# get input image from client then predict class and render respective .html page for
solution
@app.route("/predict", methods = ['GET','POST'])
def predict():
    if request.method == 'POST':
        file = request.files['image'] # fet input
        filename = file.filename
        print("@@ Input posted = ", filename)

        file_path =
os.path.join('C:/Users/Swetha/AppData/Local/Programs/Python/Python38/Tomato_Leaf_
Disease_Prediction/static/upload/', filename)
        file.save(file_path)

        print("@@ Predicting class.....")
        pred, output_page = pred_tomato_dieas(tomato_plant=file_path)

        return render_template(output_page, pred_output = pred, user_image = file_path)

# For local system & cloud
if __name__ == "__main__":
    app.run(threaded=False,port=8080)
```

GITHUB LINK

Link

https://github.com/IBM-EPBL/IBM-Project-41079-1660639233

PROJECT DEMO LINK

Link

https://drive.google.com/file/d/1kloXJXuCyQ0clILUR9d-ixdZglMhJe89/view?usp=share_link