FERTILIZER RECOMMENDATION SYSTEM FOR DISEASE PREDICTION

IBM PROJECT REPORT

Team ID :PNT2022TMID51948

SUBMITTED BY

KISHANTHINI M	- 962319104052
ABINILLA V A	- 962319104003
GOKUL S	- 962319104041
GOWTHAM K	- 962319104043

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

AMRITA COLLEGE OF ENGINEERING AND TECHNOLOGY ERACHAKULAM,
NAGERCOIL.

ANNA UNIVERSITY::CHENNAI 600 025

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Project Overview

- Preprocess the images
- Applying the CNN algorithm to the dataset
- How deep neural networks detect the disease
- Finding the accuracy of the model
- Building web applications using the Flask framework

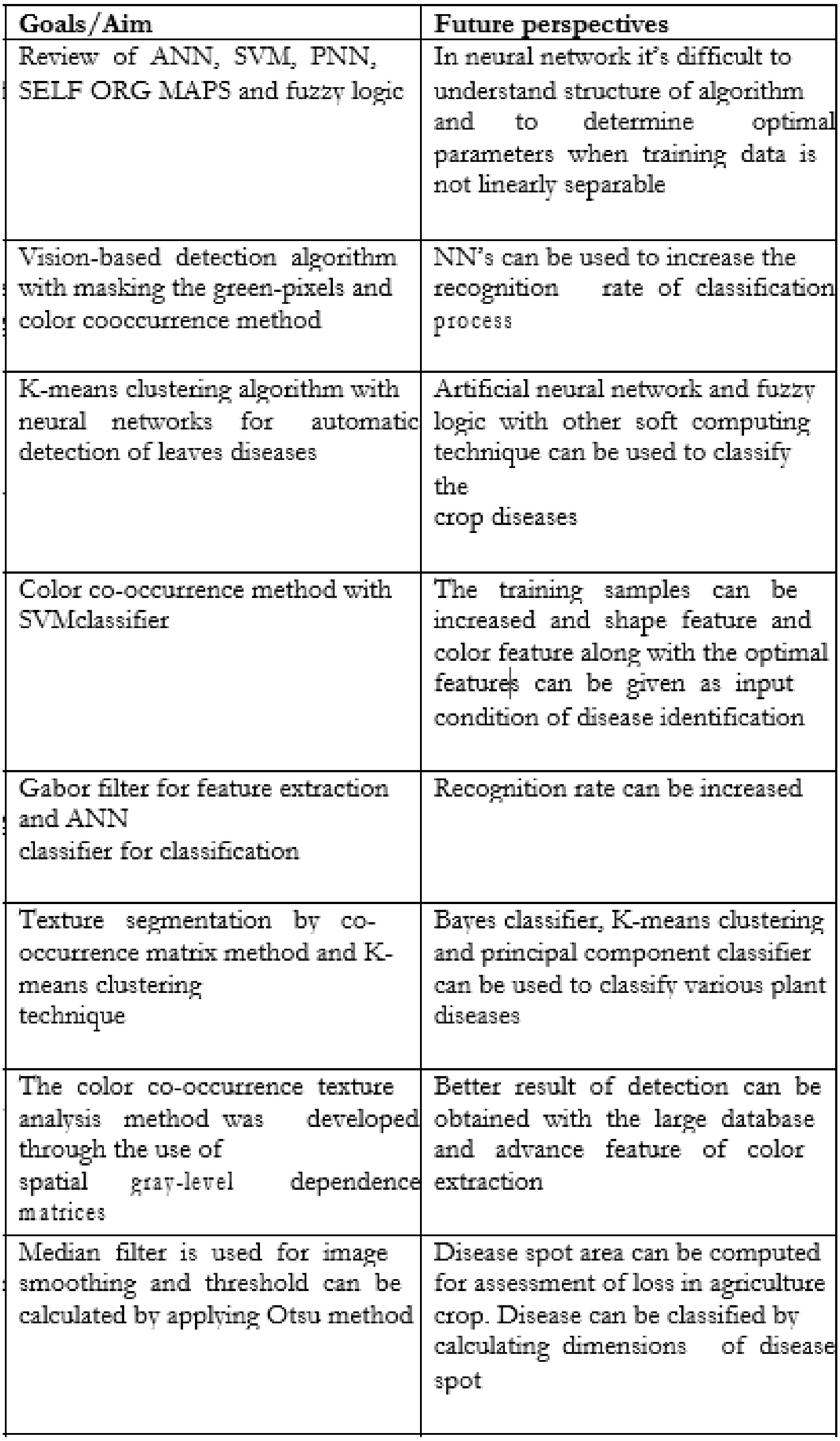
1.2 Purpose

Agriculture productivity is a key factor in economic growth. This is one of the reasons that plant disease detection is crucial in the sector of agriculture, as the presence of illness in plants is extremely common. If proper care is not done in this region, plants suffer major consequences, which have an impact on the products' quality, quantity, or productivity. The benefit of automatic plant disease detection is that it decreases the amount of labour required to monitor large crop farms and can identify disease symptoms at very early stages, such as when they first develop on plant leaves. With the prediction of the leaf disease the best fertilizers are recommended in order to prevent the leaf disease at their early stages.

2. LITERATURE SURVEY

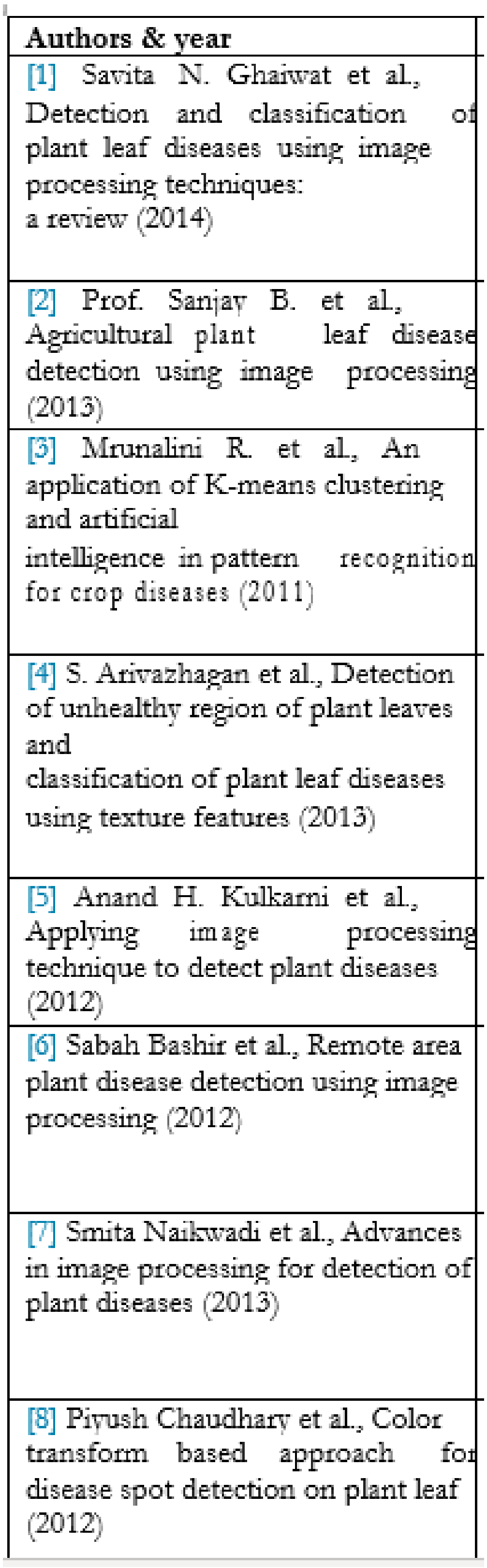
2.1 Existing Problem

2.1 Existing problem



Goals/Aim	Future perspectives
Review of ANN, SVM, PNN, SELF ORG MAPS and fuzzy logic	In neural network it's difficult to understand structure of algorithm and to determine optimal parameters when training data is not linearly separable
Vision-based detection algorithm with masking the green-pixels and color cooccurrence method	NN's can be used to increase the recognition rate of classification process
K-means clustering algorithm with neural networks for automatic detection of leaves diseases	Artificial neural network and fuzzy logic with other soft computing technique can be used to classify the crop diseases
Color co-occurrence method with SVMclassifier	The training samples can be increased and shape feature and color feature along with the optimal features can be given as input condition of disease identification
Gabor filter for feature extraction and ANN classifier for classification	Recognition rate can be increased
Texture segmentation by co-occurrence matrix method and K-means clustering technique	Bayes classifier, K-means clustering and principal component classifier can be used to classify various plant diseases
The color co-occurrence texture analysis method was developed through the use of spatial gray-level dependence matrices	Better result of detection can be obtained with the large database and advance feature of color extraction
Median filter is used for image smoothing and threshold can be calculated by applying Otsu method	Disease spot area can be computed for assessment of loss in agriculture crop. Disease can be classified by calculating dimensions of disease spot

2.2 References

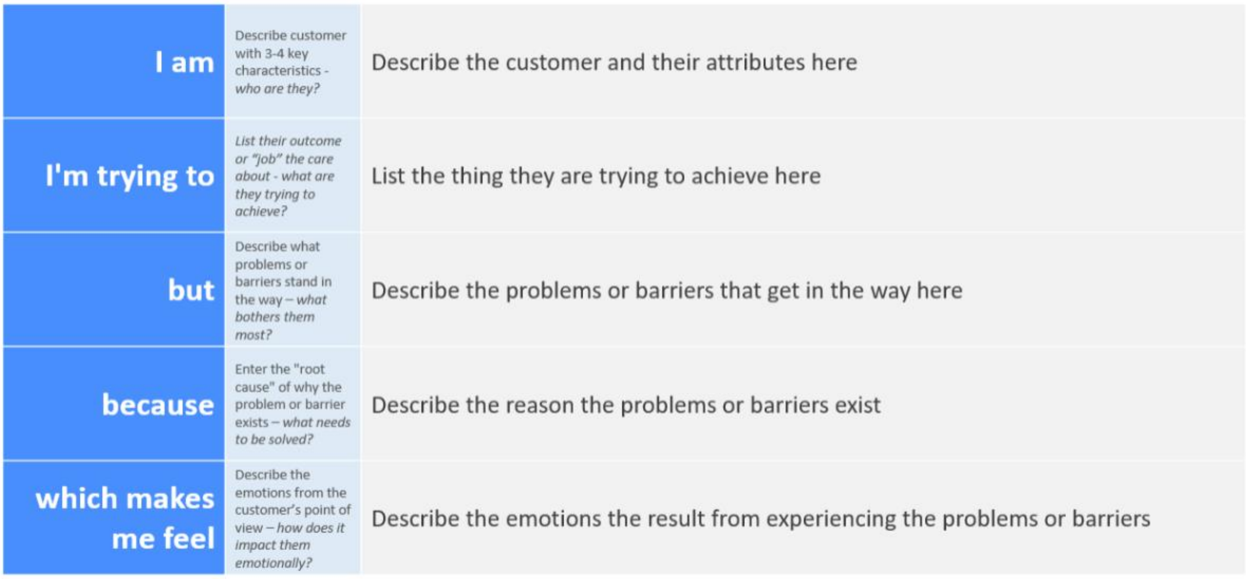


Authors & year
[1] Savita N. Ghaiwat et al., Detection and classification of plant leaf diseases using image processing techniques: a review (2014)
[2] Prof. Sanjay B. et al., Agricultural plant leaf disease detection using image processing (2013)
[3] Mrunalini R. et al., An application of K-means clustering and artificial intelligence in pattern recognition for crop diseases (2011)
[4] S. Arivazhagan et al., Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features (2013)
[5] Anand H. Kulkarni et al., Applying image processing technique to detect plant diseases (2012)
[6] Sabah Bashir et al., Remote area plant disease detection using image processing (2012)
[7] Smita Naikwadi et al., Advances in image processing for detection of plant diseases (2013)
[8] Piyush Chaudhary et al., Color transform based approach for disease spot detection on plant leaf (2012)

2.3 Problem Statement

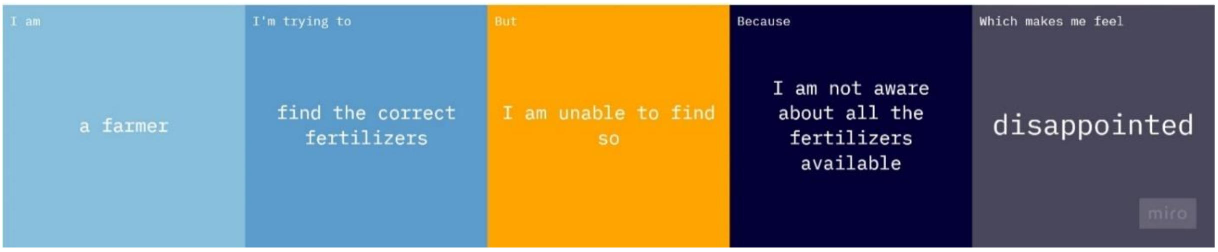
I am Suresh. I have my own land and do agriculture for my survival. I am cultivating my own land with different fruits and vegetables from past 25 years. From all these years the major problem I faced was in difficulty in Choosing Fertilizers and Controlling of Plant Disease.

- I choose to recognize the better suggestions for fertilizers and plant life with diseases.
- I am going through large losses from long time.
- I want to comprehend the end results right away.

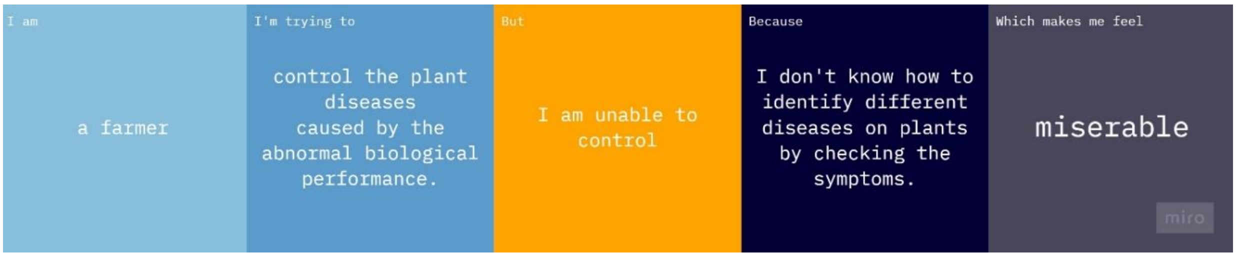


I am	Describe customer with 3-4 key characteristics - *who are they?*	Describe the customer and their attributes here
I'm trying to	*List their outcome or "job" the care about - what are they trying to achieve?*	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way – *what bothers them most?*	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists – *what needs to be solved?*	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view – *how does it impact them emotionally?*	Describe the emotions the result from experiencing the problems or barriers

Example:



I am	I'm trying to	But	Because	Which makes me feel
a farmer	find the correct fertilizers	I am unable to find so	I am not aware about all the fertilizers available	disappointed

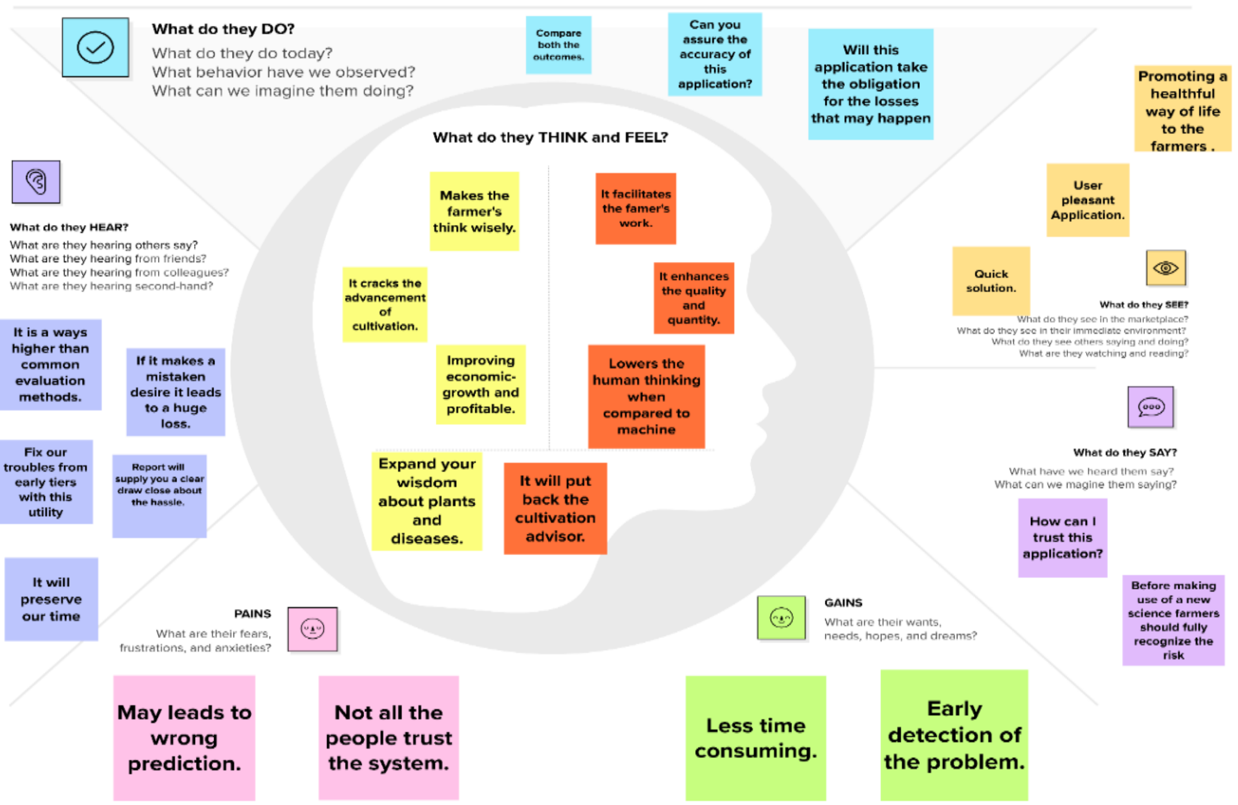


I am	I'm trying to	But	Because	Which makes me feel
a farmer	control the plant diseases caused by the abnormal biological performance.	I am unable to control	I don't know how to identify different diseases on plants by checking the symptoms.	miserable

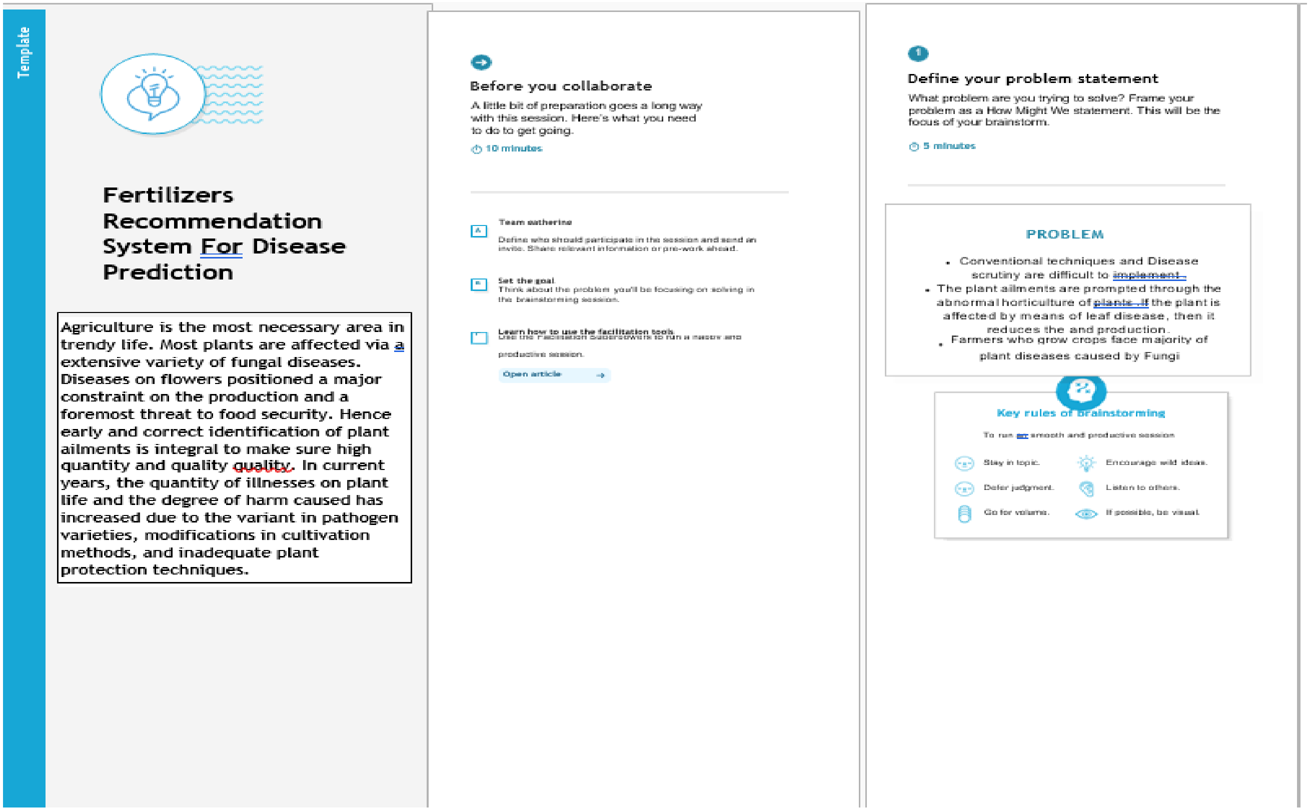
Problem Statement (PS)	**I am**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	a Farmer	find the correct fertilizers	I am unable to find so	I am not aware about all the fertilizers available	Disappointed.
PS-2	a Farmer	control the plant diseases caused by the abnormal biological performance.	I am unable to control	I don't know how to identify different diseases on plants by checking the symptoms.	Miserable.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map


What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?
Compare both the outcomes.
Can you assure the accuracy of this application?
Will this application take the obligation for the losses that may happen
Promoting a healthful way of life to the farmers .
What do they THINK and FEEL?
Makes the farmer's think wisely.
It facilitates the famer's work.
It cracks the advancement of cultivation.
It enhances the quality and quantity.
Improving economic-growth and profitable.
Lowers the human thinking when compared to machine
Expand your wisdom about plants and diseases.
It will put back the cultivation advisor.
What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?
It is a ways higher than common evaluation methods.
If it makes a mistaken desire it leads to a huge loss.
Fix our troubles from early tiers with this utility
Report will supply you a clear draw close about the hassle.
It will preserve our time
User pleasant Application.
Quick solution.
What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?
What do they SAY?
What have we heard them say?
What can we magine them saying?
How can I trust this application?
Before making use of a new science farmers should fully recognize the risk
PAINS
What are their fears, frustrations, and anxieties?
May leads to wrong prediction.
Not all the people trust the system.
GAINS
What are their wants, needs, hopes, and dreams?
Less time consuming.
Early detection of the problem.

3.2 Ideation & Brainstroming



Fertilizers Recommendation System For Disease Prediction

Agriculture is the most necessary area in trendy life. Most plants are affected via a extensive variety of fungal diseases. Diseases on flowers positioned a major constraint on the production and a foremost threat to food security. Hence early and correct identification of plant ailments is integral to make sure high quantity and quality quality. In current years, the quantity of illnesses on plant life and the degree of harm caused has increased due to the variant in pathogen varieties, modifications in cultivation methods, and inadequate plant protection techniques.

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

C Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

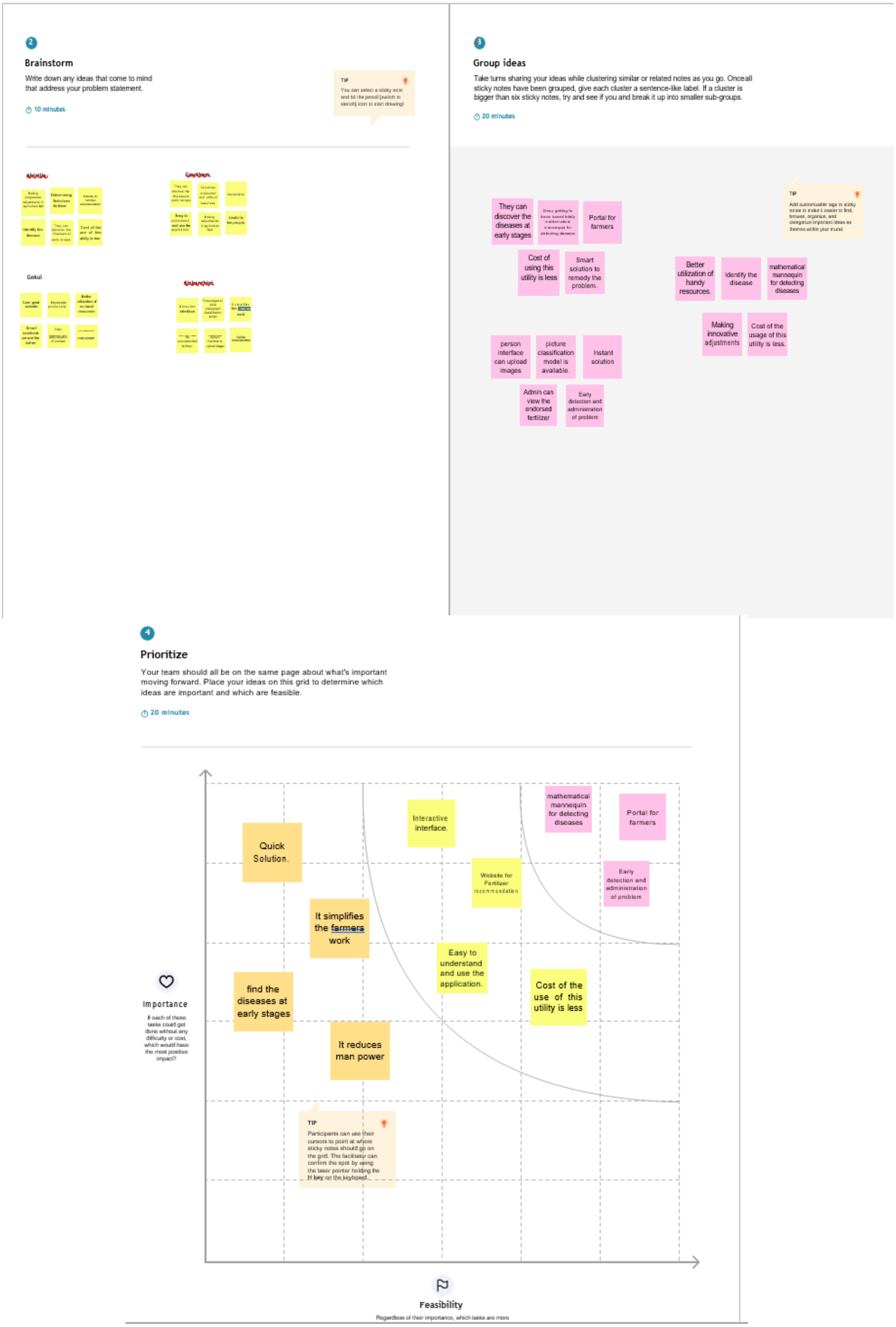
PROBLEM

- Conventional techniques and Disease scrutiny are difficult to implement .
- The plant ailments are prompted through the abnormal horticulture of plants . If the plant is affected by means of leaf disease, then it reduces the and production.
- Farmers who grow crops face majority of plant diseases caused by Fungi

Key rules of brainstorming

To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Abirami

Making progressive adjustments in agriculture field

Determining the ... fertilizer

[illegible]

Identify the disease

They can discover the diseases at early stages

Cost of the use of this utility is less

Gowtham

They can discover the diseases at early stages

Improves production and utility of resources

[illegible]

Easy to understand and use the application

Making adjustments in agriculture field

Useful to the people.

Gokul

User-great website

Improves productivity

Better utilization of on hand resources

Smart solution to unravel the bother

Early [illegible] of problem

[illegible] man power

Kishanthini

Interactive interface.

Encourages to build photograph classification model

It simplifies the farmers work

[illegible] recommended fertilizer

[illegible] interface to upload images

Fertilizer recommendation

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

They can discover the diseases at early stages

Deep getting to know based totally mathematical mannequin for detecting diseases

Portal for farmers

Cost of using this utility is less

Smart solution to remedy the problem.

person interface can upload images

picture classification model is available.

Instant solution

Admin can view the endorsed fertilizer

Early detection and administration of problem

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Better utilization of handy resources.

Identify the disease

mathematical mannequin for detecting diseases

Making innovative adjustments

Cost of the usage of this utility is less.

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Quick Solution.

It simplifies the farmers work

find the diseases at early stages

It reduces man power

Interactive interface.

Website for Fertilizer recommendation

Easy to understand and use the application.

mathematical mannequin for detecting diseases

Portal for farmers

Early detection and administration of problem

Cost of the use of this utility is less

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more

3.3 Proposed Solution

S.No.	Parameter	Description
1.	**Problem Statement (Problem to be solved)**	When plants and crops are suffering from pests it affects the agricultural production of the country. Usually, farmers or specialists watch the plants with an eye out for disease detection and identification. Based on the disease fertilizers are used to cure. However, this approach is frequently slow to process, expensive, and inaccurate.
2.	**Idea / Solution description**	A Convolution Neural Network based method for leaf disease detection is proposed. The model is trained with images of various defective and non -defective leaves of fruits and vegetables. A defective leaf image is given as input to the model. The model then predicts the leaf disease and the best suitable fertilizer is recommended.
3.	**Novelty / Uniqueness**	The Convolution Neural Network works well with the images. The Image acquisition and Pre-processing are done before feeding the images into the model to remove unwanted information present in it. After training the model using the if-else model the fertilizer is recommended based on the disease predicted by the model.
4.	**Social Impact / Customer Satisfaction**	Customers will be able to detect the leaf disease earlier before the time of harvesting. The basic needs of the farmer and gardener is satisfied. No prior knowledge of the leaf disease and fertilizer is required as the model does everything. The website is user-friendly and it takes less time to predict and recommend the disease and fertilizer than doing manually.
		Nowadays the rate of leaf disease is increasing. and also agricultural experts are in

5.	**Business Model (Revenue Model)**	fewer numbers because youngsters today rarely opt for agriculture as their job. So there is a high demand for the website that we are going to build. Therefore the rate of subscription/users increases which yield benefits to the industry.
6.	**Scalability of the Solution**	A model built to predict the disease of leaf consumes data from a large dataset and delivers prediction instantly. It serves millions of users and fits well for big data.

3.4 Problem Solution fit

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

- Farmers
- Gardeners
- Agriculture Research Scientist
- Plant Scientist

6. CUSTOMER CONSTRAINTS CC

- Poor Network Connections in rural areas
- Less Quality Image (Because of less camera quality)
- No proper knowledge of using the website
- Installation Cost of required devices (laptops,PC,mobile etc.)

5. AVAILABLE SOLUTIONS AS

The traditional method to identify diseases by visually inspecting certain features like texture, color and shape of leaves, Usually the farmers used to hire professional agriculturists to diagnose their crops for diseases.

✅ **Pros:** Comparatively Cheaper than other methods

❌ **Cons:** Time Consuming

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- Effective Detection of leaf disease using AI methods.
- Recommendation of Fertilizers based on the Disease Detected.

9. PROBLEM ROOT CAUSE RC

The reasons for leaf damage are Leaf viruses, Leaf biting insects, Spraying of pesticides, Usage of inefficient fertilizers. And also Environmental factors like soil pH, temperature, and humidity causes leaf damage.

The leaf diseases should be detected in the early stages and the best suitable fertilizer should be recommended to reduce or completely cure the diseases.

7. BEHAVIOUR BE

- At first,the farmers should upload the defective leaf to the website
- The leaf disease is predicted by the trained model.
- A suitable fertilizer for the predicted disease then suggested.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

- The failure of crops leads to a huge loss in quantity of crops.
- Quality of the crops being reduced.
- Increase in the cost of pesticides.
- Reading about more efficient methods in internet.

4. EMOTIONS: BEFORE/AFTER

BEFORE	AFTER
Difficent	Confident
Distress	Relief

10. YOUR SOLUTION SL

- As Convolution Neural Network works well with Images we used this technique to learn the patterns from the images of different leaves.
- Thus given an input image the trained model predicts the disease of the leaf.
- The predicted result is used to recommend a suitable fertilizer for the disease

→ MODEL → Disease → Fertilizer

8.CHANNELS OF BEHAVIOUR CH

8.1 ONLINE

- Reading articles online to improve knowledge about various plant diseases and appropriate fertilizers.
- Gathering information online about various fertilizer recommendation sources.

8.2 OFFLINE

- Visiting various agricultural lands to know more about types of leaf diseases and fertilizers used by the farmers.
- Talking with agricultural research scientist.

Extract online&offline CH of BE

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements:

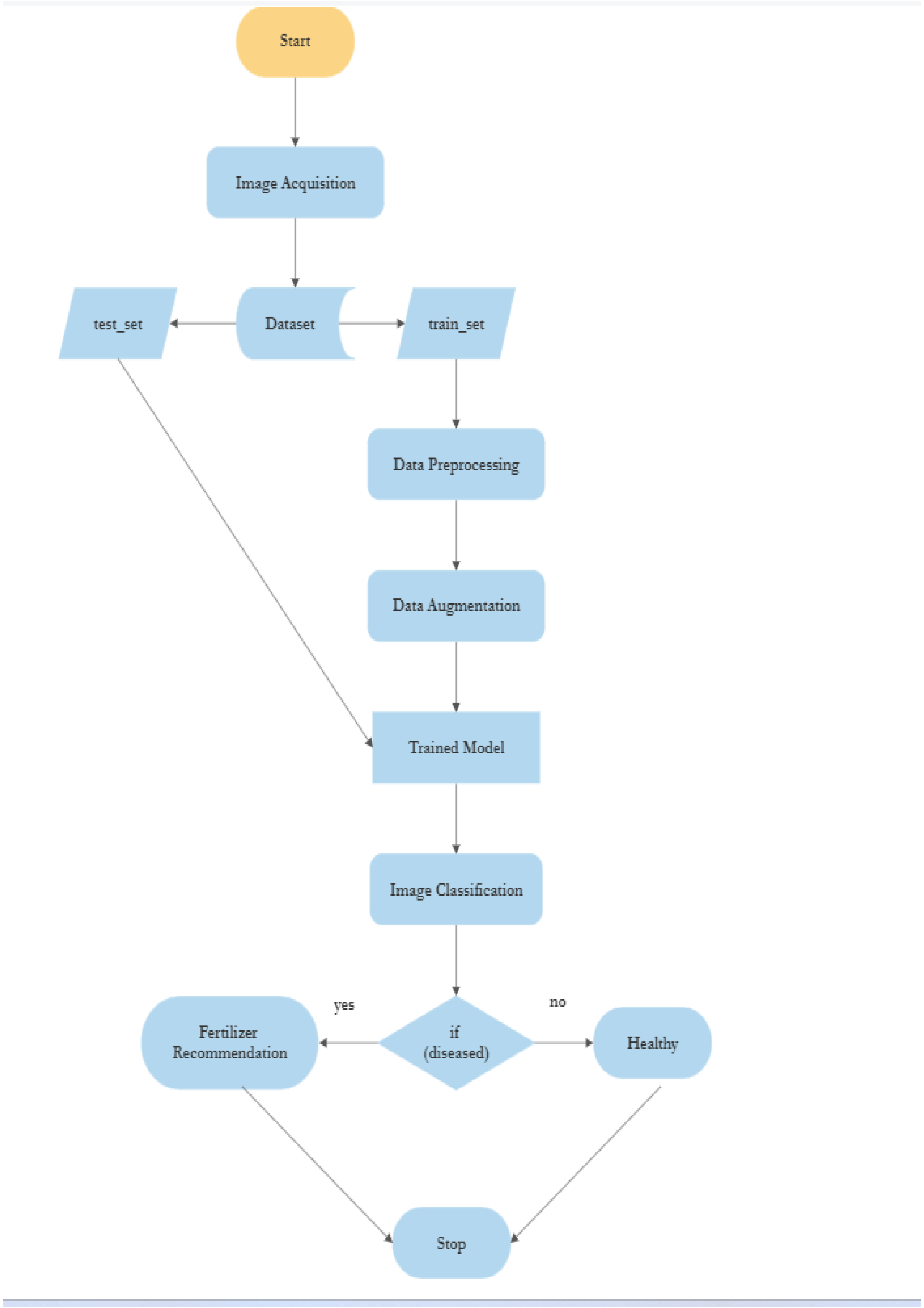
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Basic Tools	A PC, Laptop or Mobile used to access the website
FR-2	Website Information	Information about how the website works and steps to upload the images.
FR-3	Materials and Methods	A camera or similar devices are used to 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 highly secured.
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 image for disease prediction.
NFR-5	**Availability**	It is available to predict the disease in plants/leaves for all user.
NFR-6	**Scalability**	Increase the precise of disease prediction in leaf.

5.PROJECT DESIGN

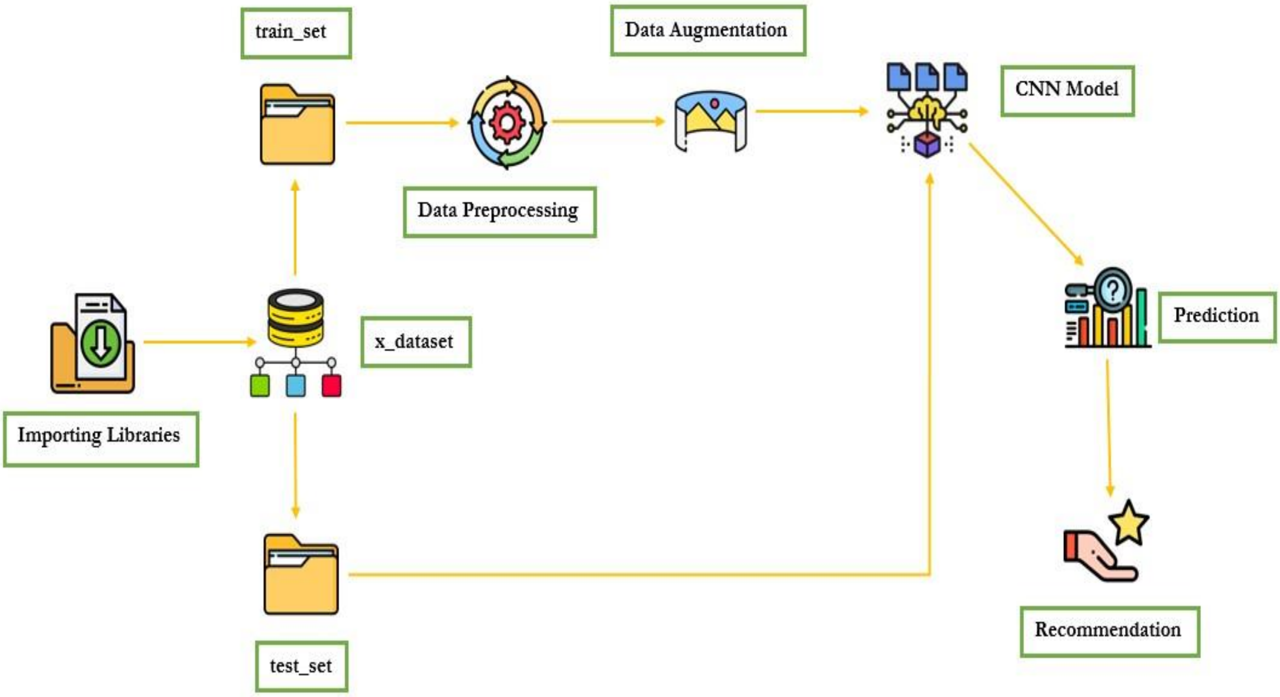
5.1 Data Flow Diagram



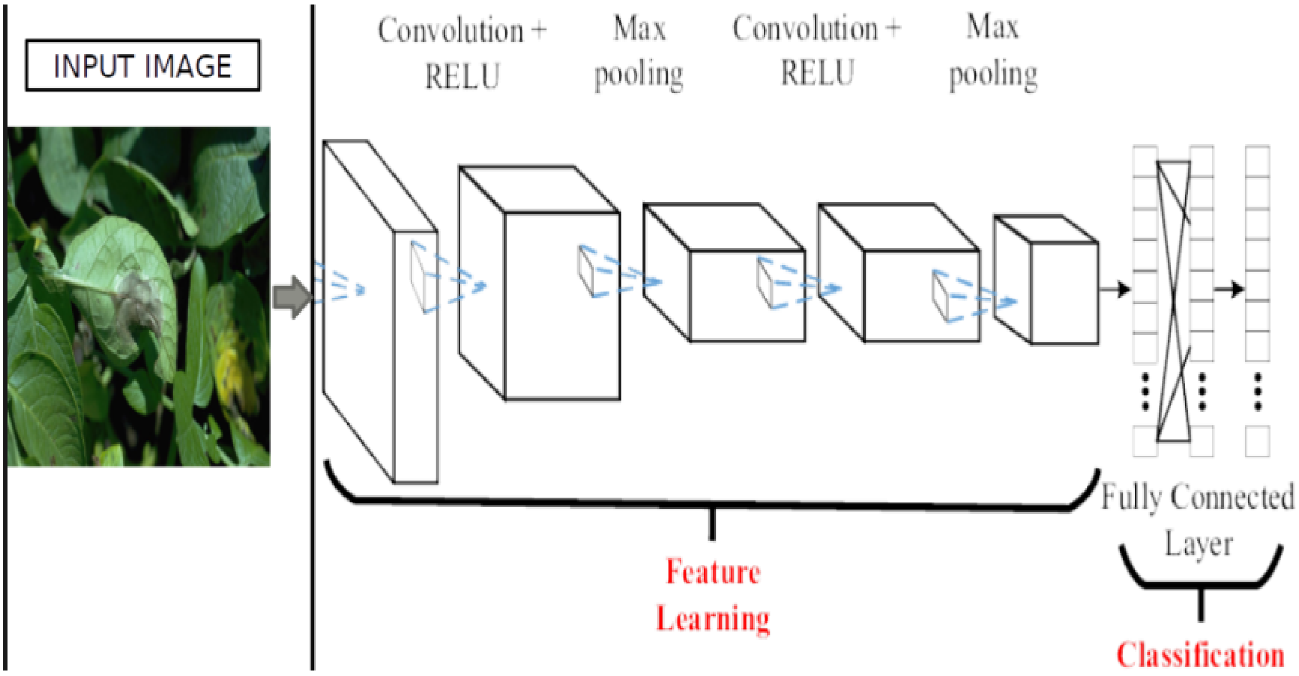
5.2 Solution & Technical Architecture

5.2.1 Solution Architecture

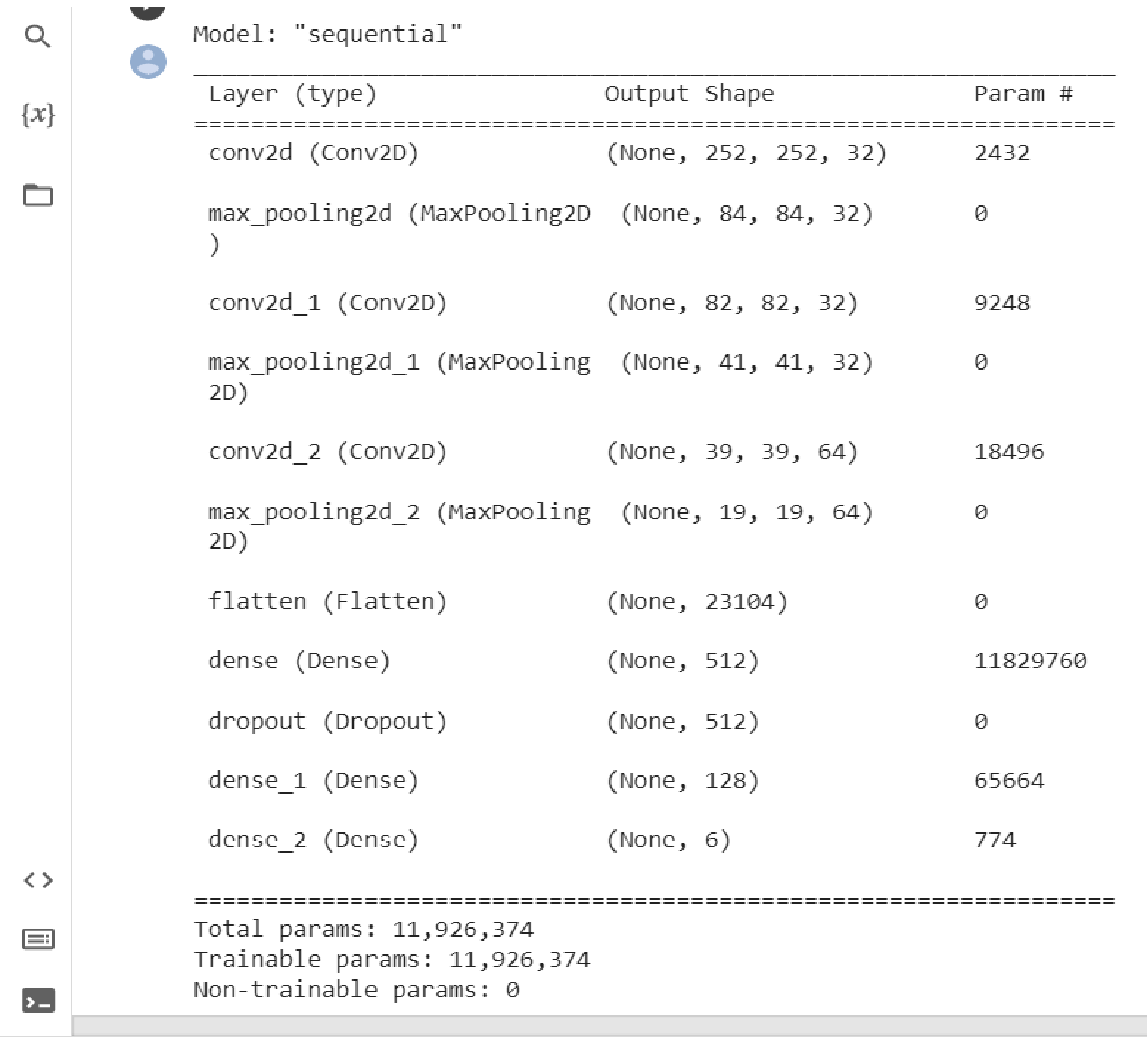
5.2.2



Technical Architecture



5.2.3 Model Architecture



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 252, 252, 32)      2432

 max_pooling2d (MaxPooling2D  (None, 84, 84, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 82, 82, 32)        9248

 max_pooling2d_1 (MaxPooling  (None, 41, 41, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 39, 39, 64)        18496

 max_pooling2d_2 (MaxPooling  (None, 19, 19, 64)       0
 2D)

 flatten (Flatten)           (None, 23104)             0

 dense (Dense)               (None, 512)               11829760

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 128)               65664

 dense_2 (Dense)             (None, 6)                 774

=================================================================
Total params: 11,926,374
Trainable params: 11,926,374
Non-trainable params: 0
```

5.3 User Stories

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 my password.	I can access my account / dashboard	High	**Sprint-1**
	Login	USN-2	As a user, I can log into the application by entering email & password	I can login using my Email ID accounts or user credentials	High	**Sprint-1**

	Dashboard	USN-3	As a user, I can view the page of the application where I can upload my images and the fertilizer should be recommended	I can access my account/ dashboard	High	**Sprint-2**
Customer (Webuser)	Registration	USN-4	As a user, I can login to web dashboard just Like website dashboard	I can register using my username and password	High	**Sprint-3**
	Login	USN-5	As a user, I can login to my web dashboard with the login credentials	I can login using my User credentials	High	**Sprint-3**
	Dashboard	USN-6	As a user, I can view the web application where i can upload my images and the fertilizer should be recommended	I can access my account/ dashboard	High	**Sprint-4**
		USN-7	As a user, the fertilizer recommended to me should be of higher accuracy	I can access myaccount/ dashboard	High	**Sprint-4**
Administrator	Login	USN-8	As a admin, I can login to the website using my login credentials	I can login to the website using my login credentials	High	**Sprint-4**
	Dashboard	USN-9	As a admin, I can view the dashboard of the application	I can access my dashboard	High	**Sprint-4**

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-1	Data Collection		Download the dataset, Two datasets	5	High	Kishanthini M, Abinilla V A, Gokul S,

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			will be used, we will be creating two models one to detect vegetable leaf diseases like tomato, potato, and pepper plants and the second model would be for fruits diseases like corn, peach, and apple.			Gowtham K
	Image Preprocessing		Before training the model you have to preprocess the images and then feed them on to the model for training. We make use of Keras Image DataGenerator class for image preprocessing.	5	Medium	Kishanthini M, Abinilla V A, Gokul S, Gowtham K
Sprint-2	Model Building For Fruit Disease Prediction		Create a CNN Model which can classify the type of fruit leaf disease from given image. (fruit-dataset)	7.5	High	Kishanthini M, Abinilla V A, Gokul S, Gowtham K
	Model Building For Vegetable Disease Prediction		Create a CNN Model which can classify the type of vegetable	4.5	High	Kishanthini M, Abinilla V A, Gokul S, Gowtham K

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			leaf disease from given image. (vegetable-dataset)			
Sprint-3	Testing Both The Models		The model had been successfully trained. The model is to be tested with different images to know if it is working correctly.	5	High	Kishanthini M, Abinilla V A, Gokul S, Gowtham K
	IBM Cloud Registration		Register for the IBM Cloud to use the service of the IBM.	5	High	Kishanthini M, Abinilla V A, Gokul S, Gowtham K
	Train & Deploy The Model On IBM		Train the model on IBM Cloud & Deploy it using the Watson Studio Service.	5	High	Kishanthini M, Abinilla V A, Gokul S, Gowtham K
Sprint-4	Application Building		The build model is then integrated into a web application so that normal users can also use it. Build a flask app using HTML and Python	4	Medium	Kishanthini M, Abinilla V A, Gokul S, Gowtham K
	Home page	USN-1	As a user I can know about how the application works	4	High	Kishanthini M, Abinilla V A, Gokul S, Gowtham K

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
	Prediction Page	USN-2	As a user I can access the application and upload the images of leaf and get my fertizer recommended	4	High	Kishanthini M, Abinilla V A, Gokul S, Gowtham K

6.2 Sprint Delivery Plan

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	20	6 Days	24 Oct 2022	29 Oct 2022	20	02 Nov 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	15	06 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	15	13 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	10	20 Nov 2022

7. CODING & SOLUTIONING

7.1 Feature 1

CNN Model

Fruit-Training.ipynb

```
from keras_preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D,MaxPool2D,Flatten
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import glob
from keras.layers import Dense,Dropout,Flatten
from keras.layers import Conv2D,MaxPooling2D,Activation,AveragePooling2D,BatchNor
malization
from keras.preprocessing import image
from tensorflow.keras.utils import load_img
from tensorflow.keras.utils import img_to_array
import numpy as np
import matplotlib.pyplot as plt
from tensorflow import keras
```

Data Collection & Preprocessing

```
#Rescaling , Zooming ,Horizontal Flip Data Augmentation
train_datagen = ImageDataGenerator(rescale = 1./255,zoom_range = 0.2,horizontal_f
lip = True,vertical_flip = False)
test_datagen = ImageDataGenerator(rescale = 1./255)


train_dir ="/content/drive/MyDrive/Project/Dataset Plant Disease/fruit-
dataset/fruit-dataset/train/"
test_dir="/content/drive/MyDrive/Project/Dataset Plant Disease/fruit-
dataset/fruit-dataset/test/"

train_samples =get_files(train_dir)
num_classes=len(glob.glob(train_dir+"/*"))
test_samples=get_files(test_dir)
print(num_classes,"Classes")
print(train_samples,"Train images")
print(test_samples,"Test images")


# set height and width and color of input image.
img_width,img_height =256,256
input_shape=(img_width,img_height,3)
batch_size =32
train_generator = train_datagen.flow_from_directory(train_dir,
                                target_size=(img_width,img_height),batch_size=bat
ch_size)
```

```
test_generator=test_datagen.flow_from_directory(test_dir,shuffle=True,target_size
=(img_width,img_height),batch_size=batch_size)


# The name of the 6 diseases.
train_generator.class_indices
```

Model Training

```
# CNN building.
model = Sequential()
model.add(Conv2D(32, (5, 5),input_shape=input_shape,activation='relu'))
model.add(MaxPooling2D(pool_size=(3, 3)))
model.add(Conv2D(32, (3, 3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, (3, 3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(512,activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(128,activation='relu'))
model.add(Dense(num_classes,activation='softmax'))
model.summary()


from keras.preprocessing import image
from tensorflow.keras.utils import load_img
from tensorflow.keras.utils import img_to_array
import numpy as np
img1 = load_img('/home/wsuser/work/fruit-
dataset/train/Corn_(maize)___Northern_Leaf_Blight/8e0669bb-1959-4f28-b98d-
da3c2d85396c___RS_NLB 3678.JPG')
plt.imshow(img1);
#preprocess image
img1 = load_img('/home/wsuser/work/fruit-
dataset/train/Corn_(maize)___Northern_Leaf_Blight/8e0669bb-1959-4f28-b98d-
da3c2d85396c___RS_NLB 3678.JPG', target_size=(256, 256))
img = img_to_array(img1)
img = img/255
img = np.expand_dims(img, axis=0)


validation_generator = train_datagen.flow_from_directory(
                       train_dir, # same directory as training data
                       target_size=(img_height, img_width),
                       batch_size=batch_size)
opt=keras.optimizers.Adam(lr=0.001)
model.compile(optimizer=opt,loss='categorical_crossentropy',metrics=['accuracy'])
nb_epoch = 10
train=model.fit_generator(train_generator,epochs=nb_epoch,steps_per_epoch=train_g
enerator.samples//batch_size,validation_data=validation_generator,validation_step
s=validation_generator.samples // batch_size,verbose=1)
```

```
# Save model
from keras.models import load_model
model.save('fruit.h5')
```

Testing the Model

```
# Save model
from keras.models import load_model
model.save('fruit.h5')


img = image.load_img(r"/content/drive/MyDrive/Project/Dataset Plant Disease/fruit
-dataset/fruit-dataset/test/Apple___Black_rot/00e909aa-e3ae-4558-9961-
336bb0f35db3___JR_FrgE.S 8593.JPG")
img = image.load_img(r"/content/drive/MyDrive/Project/Dataset Plant Disease/fruit
-dataset/fruit-dataset/test/Apple___Black_rot/00e909aa-e3ae-4558-9961-
336bb0f35db3___JR_FrgE.S 8593.JPG",target_size=(64,64))
x = image.img_to_array(img)
x = np.expand_dims(x,axis=0)
y = np.argmax(model.predict(x),axis=1)
print(y)
```

Vegetable-Training.ipynb

```
# Import Libraries
import os
import glob
import matplotlib.pyplot as plt
import numpy as np
# Keras API
import keras
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten
from keras.layers import Conv2D,MaxPooling2D,Activation,AveragePooling2D,BatchNor
malization
from keras.preprocessing.image import ImageDataGenerator
```

Data Collection & Preprocessing

```
# My data is in google drive.
train_dir ="/content/drive/MyDrive/Project/Dataset Plant Disease/Veg-dataset/Veg-
dataset/train_set/"
test_dir="/content/drive/MyDrive/Project/Dataset Plant Disease/Veg-dataset/Veg-
dataset/test_set"


# function to get count of images
def get_files(directory):
  if not os.path.exists(directory):
    return 0
  count=0
  for current_path,dirs,files in os.walk(directory):
    for dr in dirs:
      count+= len(glob.glob(os.path.join(current_path,dr+"/*")))
  return count

train_samples =get_files(train_dir)
num_classes=len(glob.glob(train_dir+"/*"))
test_samples=get_files(test_dir)
print(num_classes,"Classes")
print(train_samples,"Train images")
print(test_samples,"Test images")

# Pre-processing data with parameters.
train_datagen=ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)

# set height and width and color of input image.
img_width,img_height =256,256
input_shape=(img_width,img_height,3)
batch_size =32
train_generator =train_datagen.flow_from_directory(train_dir,
                                target_size=(img_width,img_height),batch_size=bat
ch_size)
test_generator=test_datagen.flow_from_directory(test_dir,shuffle=True,target_size
=(img_width,img_height),batch_size=batch_size)

# The name of the 12 diseases.
train_generator.class_indices
```

Model Training

```
# CNN building.
model = Sequential()
model.add(Conv2D(32, (5, 5),input_shape=input_shape,activation='relu'))
model.add(MaxPooling2D(pool_size=(3, 3)))
model.add(Conv2D(32, (3, 3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, (3, 3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(512,activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(128,activation='relu'))
model.add(Dense(num_classes,activation='softmax'))
model.summary()

from keras.preprocessing import image
from tensorflow.keras.utils import load_img
from tensorflow.keras.utils import img_to_array
import numpy as np
img1 = load_img('/content/drive/MyDrive/Project/Dataset Plant Disease/Veg-
dataset/Veg-dataset/train_set/Tomato___Late_blight/013f987a-9371-4763-a104-
ea6f326e584b___GHLB2 Leaf 8556.JPG')
plt.imshow(img1);
#preprocess image
img1 = load_img('/content/drive/MyDrive/Project/Dataset Plant Disease/Veg-
dataset/Veg-dataset/train_set/Tomato___Late_blight/013f987a-9371-4763-a104-
ea6f326e584b___GHLB2 Leaf 8556.JPG', target_size=(256, 256))
img = img_to_array(img1)
img = img/255
img = np.expand_dims(img, axis=0)


validation_generator = train_datagen.flow_from_directory(
                       train_dir, # same directory as training data
                       target_size=(img_height, img_width),
                       batch_size=batch_size)
opt=keras.optimizers.Adam(lr=0.001)
model.compile(optimizer=opt,loss='categorical_crossentropy',metrics=['accuracy'])
nb_epoch = 15
train=model.fit_generator(train_generator,epochs=nb_epoch,steps_per_epoch=train_g
enerator.samples//batch_size,validation_data=validation_generator,validation_step
s=validation_generator.samples // batch_size,verbose=1)

# Save model
from keras.models import load_model
model.save('vegetable.h5')
```

Model Testing

```
# Loading model and predict.
from keras.models import load_model
model=load_model('vegetable.h5')
# Mention name of the disease into list.
Classes = ["Pepper,_bell___Bacterial_spot",
 "Pepper,_bell___healthy",
 "Potato___Early_blight",
 "Potato___Late_blight",
 "Potato___healthy",
 "Tomato___Bacterial_spot",
 "Tomato___Late_blight",
 "Tomato___Leaf_Mold",
 "Tomato___Septoria_leaf_spot"]

import numpy as np
import matplotlib.pyplot as plt
# Pre-Processing test data same as train data.
img_width=256
img_height=256
```

```
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy
'])
from keras.preprocessing import image
def prepare(img_path):
    img = load_img(img_path, target_size=(256, 256))
    x =   img_to_array(img)
    x = x/255
    return np.expand_dims(x, axis=0)

result = model.predict([prepare('/content/drive/MyDrive/Project/Dataset Plant Dis
ease/Veg-dataset/Veg-dataset/test_set/Potato___Early_blight/b7883606-5157-4dc1-
b965-fc10f8fe1796___RS_Early.B 7598.JPG')])
disease= load_img('/content/drive/MyDrive/Project/Dataset Plant Disease/Veg-
dataset/Veg-dataset/test_set/Potato___Early_blight/b7883606-5157-4dc1-b965-
fc10f8fe1796___RS_Early.B 7598.JPG')
#plt.imshow(disease)
ind = np.argmax(result)
ind
print (Classes[ind])
```

7.2 Feature 2

Application Building Using flask

app.ipynb

```
import requests
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
import numpy as np
import pandas as pd
import tensorflow as tf
from flask import Flask, request, render_template, redirect, url_for
import os
from werkzeug.utils import secure_filename
from tensorflow.python.keras.backend import set_session

app=Flask(__name__)
model = load_model("vegetable.h5")
model1 = load_model("fruit.h5")


@app.route('/')
#home page
def home():
    return render_template('index.html')

#prediction page
@app.route('/prediction.html')
def prediction():
    return render_template('prediction.html')


@app.route('/predict',methods=['GET','POST'])
def predict():
    if request.method == 'POST':
        #getting file from post request
        f=request.files['image']
        #save the files to uploads
        basepath = os.path.dirname(os.getcwd())
        file_path        =        os.path.join(basepath,        '\\Users\\Inspiron15
3000\\Desktop\\Application Building\\uploads', secure_filename(f.filename))
        f.save(file_path)
        img = image.load_img(file_path, target_size=(256,256))
        x=image.img_to_array(img)
        x=np.expand_dims(x,axis=0)
        plant=request.form['plants']
        print(plant)
        p = ""
        disease = ""
        caution = ""
        if(plant=="Vegetable"):
            y = np.argmax(model.predict(x),axis=1)
            df=pd.read_excel('precautions - veg.xlsx')
            caution = df.iloc[y[0]]['caution']
            p = df.iloc[y[0]]['plant']
            disease = df.iloc[y[0]]['disease']
        else:
            y = np.argmax(model1.predict(x),axis=1)
            df=pd.read_excel('precautions - fruits.xlsx')
            caution = df.iloc[y[0]]['caution']
            p = df.iloc[y[0]]['plant']
            disease = df.iloc[y[0]]['disease']

        return render_template('predict.html', plant=p , disease = disease , caution
= caution)


if __name__ ==  "__main__":
    app.run(debug=False)
```

html pages:

index.html

```
 <!DOCTY
 PE
 html>
           <html lang="en">
             <head>
               <meta charset="UTF-8" />
               <meta http-equiv="X-UA-Compatible" content="IE=edge" />
               <meta   name="viewport"   content="width=device-width,   initial-
           scale=1.0" />
               <title>Home</title>
               <link rel="stylesheet" href="/static/css/style.css" />
               <script
           src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.
           js"></script>

                <script   src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
           integrity="sha384-
           q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
           crossorigin="anonymous"></script>
```

```
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/pop
per.min.js" integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap
.min.js" integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
  <script
src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
    <script src="https://cdn.tailwindcss.com"></script>
    <script>
      tailwind.config = {
        theme: {
          extend: {
            colors: {
              header: "#28272C",
              bodyColor: "#C1C5A8",
            },
          },
        },
      };
    </script>
  </head>
  <body>
    <nav>
      <div
        class="flex flex-col items-center w-full justify-center bg-
header font-semibold h-15"
      >
        <div class="">
          <h1 class="text-white pt-10 text-2xl tracking-wide ml-8">
            FERTILIZER RECOMMENDATION SYSTEM FOR DISEASE PREDICTION
          </h1>

           <h4 class="text-white mt-2 text-center text-2xl tracking-
wide ml-8">
            GET YOUR LEAF DISEASE PREDICTED IN SECONDS!!

          </h4>
        </div>

        <ul class="flex justify-end p-4 pr-20 flex-row w-full">
          <li>
            <a class="text-white mr-8 hover:text-gray-400 duration-
200" href="/"
              >HOME</a
            >
          </li>
          <li>
            <a
              class="text-white hover:text-gray-400 duration-200"
              href="prediction.html"
              >PREDICITION</a
            >
          </li>
        </ul>
      </div>
    </nav>

    <section class="h-screen bg-bodyColor">
      <div
        class="flex flex-col pt-24 w-full px-12 items-center justify-
center space-x-6 md:flex-row"
      >
        <div class="w-3/6">
          <h1 class="text-7xl font-bold text-gray-800">
            Detect if your plant is infected!!
          </h1>
          <p class="text-lg pt-8 text-xl text-justify text-gray-600">
            Agriculture is one of the major sectors worls wide. Over
the years
            it has developed and the use of new technologies and
equipment
            replaced almost all the traditional methods of farming.
The plant
            diseases effect the production. Identification of diseases
and
            taking necessary precautions is all done through naked
eye, which
            requires labour and laboratries. This application helps
farmers in
            detecting the diseases by observing the spots on the
leaves, which
            inturn saves effort and labor costs.
          </p>
        </div>
        <div class="w-3/6">
          <img
            class="rounded-lg"
            src="/static/images/homeImage.jpg"
            alt="Plant"
          />
        </div>
      </div>
    </section>
```

```
            </body>
          </html>
```

predict.html

```
<!DOCTY
PE
html>
          <html lang="en">
            <head>
              <meta charset="UTF-8" />
              <meta http-equiv="X-UA-Compatible" content="IE=edge" />
              <meta   name="viewport"   content="width=device-width,   initial-
          scale=1.0" />
              <title>Prediction</title>
              <script src="https://cdn.tailwindcss.com"></script>
              <script
          src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.
          js"></script>

               <script   src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
          integrity="sha384-
          q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
          crossorigin="anonymous"></script>
            <script
          src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/pop
          per.min.js"                                          integrity="sha384-
          UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
          crossorigin="anonymous"></script>
            <script
          src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap
          .min.js"                                             integrity="sha384-
          JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
          crossorigin="anonymous"></script>
            <script
          src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
              <script>
                tailwind.config = {
                  theme: {
                    extend: {
                      colors: {
                        header: "#28272C",
                        bodyColor: "#C1C5A8",
                      },
                    },
                  },
                };
              </script>
            </head>
```

```
<body>
    <nav>
      <div
        class="flex flex-col items-center w-full justify-center bg-
header font-semibold h-15"
      >
        <div class="">
          <h1 class="text-white pt-10 text-2xl tracking-wide ml-8">
            FERTILIZER RECOMMENDATION SYSTEM FOR DISEASE PREDICTION
          </h1>
<h4 class="text-white mt-2 text-center text-2xl tracking-wide ml-8">
            GET YOUR LEAF DISEASE PREDICTED IN SECONDS!!
          </h4>
        </div>

        <ul class="flex justify-end p-4 pr-20 flex-row w-full">
          <li>
            <a class="text-white mr-8 hover:text-gray-400 duration-
200" href="/"
              >HOME</a
            >
          </li>
          <li>
            <a
              class="text-white hover:text-gray-400 duration-200"
              href="prediction.html"
              >PREDICITION</a
            >
          </li>
        </ul>
      </div>
    </nav>

    <section>
      <div class="flex flex-col justify-center h-screen sm:flex-row">
        <div class="w-1/2 pt-12 ml-20">
          <img
            src="/static/images/fertilizer.jpg"
            alt="FertilzerPrediction"
          />
        </div>
        <div class="flex flex-col w-1/2">


          <form
            id="form"
            action="/predict"
            method="POST"
            enctype="multipart/form-data"
          >
      <!--      <select
              name="plants"
              class="bg-gray-100 border w-4/6 border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5"
            >
              <option>Choose the type</option>
              <option value="fruit">Fruit</option>
              <option value="Vegetable">Vegetable</option>
            </select>
            <input
              type="file"
              name="image"
              accept="image/*"
              onchange="loadFile(event)"
              class="block w-full pt-4 text-sm text-gray-500 file:mr-4 file:py-2 file:px-4 file:rounded-full file:border-0 file:text-sm file:font-semibold file:bg-blue-100 file:text-blue-700 hover:file:bg-blue-200"
            /> -->

            <img class="w-1/2 pt-8 pb-8 rounded-lg" id="output" />


            <p id="predict-content" class="text-lg text-justify w-3/4">
              Prediction:  Oopps!! Your {{plant}} plant is infected by {{disease}}. {{caution}}
            </p>

   <!--
<button
              id="predict-btn"
              type="submit"
              class="text-white bg-blue-600 hover:bg-blue-500 w-32 focus:ring-4 focus:ring-blue-300 font-medium rounded-lg text-md px-5 py-2.5 mr-2 mb-2"
            >
              Predict
            </button> -->

          </form>
        </div>
      </div>
    </section>
    <script>
      //For image preview
      var loadFile = function (event) {
        var reader = new FileReader();
        reader.onload = function () {
          var output = document.getElementById("output");
          output.src = reader.result;
        };
        reader.readAsDataURL(event.target.files[0]);
      };
    </script>
  </body>
</html>
```

prediction.html

```
<!DOCTY
PE
html>
          <html lang="en">
            <head>
              <meta charset="UTF-8" />
              <meta http-equiv="X-UA-Compatible" content="IE=edge" />
              <meta   name="viewport"   content="width=device-width,   initial-
          scale=1.0" />
              <title>Prediction</title>
              <script src="https://cdn.tailwindcss.com"></script>
              <script
          src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.
          js"></script>

               <script   src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
          integrity="sha384-
          q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
          crossorigin="anonymous"></script>
            <script
          src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/pop
          per.min.js"                                        integrity="sha384-
          UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
          crossorigin="anonymous"></script>
            <script
          src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap
          .min.js"                                           integrity="sha384-
          JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
          crossorigin="anonymous"></script>
            <script
          src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
              <script>
                tailwind.config = {
                  theme: {
                    extend: {
                      colors: {
                        header: "#28272C",
                        bodyColor: "#C1C5A8",
                      },
                    },
                  },
                };
              </script>
            </head>
            <body>
              <nav>
                <div
                  class="flex flex-col items-center w-full justify-center bg-
          header font-semibold h-15"
                >
                  <div class="">
                    <h1 class="text-white pt-10 text-2xl tracking-wide ml-8">
                      FERTILIZER RECOMMENDATION SYSTEM FOR DISEASE PREDICTION
                    </h1>
          <h4 class="text-white mt-2 text-center text-2xl tracking-wide ml-8">
                      GET YOUR LEAF DISEASE PREDICTED IN SECONDS!!
                    </h4>
                  </div>

                  <ul class="flex justify-end p-4 pr-20 flex-row w-full">
                    <li>
                      <a class="text-white mr-8 hover:text-gray-400 duration-
          200" href="/"
                        >HOME</a
                      >
                    </li>
                    <li>
                      <a
                        class="text-white hover:text-gray-400 duration-200"
                        href="prediction.html"
                        >PREDICITION</a
                      >
                    </li>
                  </ul>
                </div>
              </nav>

              <section>
                <div class="flex flex-col justify-center h-screen sm:flex-row">
                  <div class="w-1/2 pt-12 ml-20">
                    <img
                      src="/static/images/sectionImage.png"
                      alt="FertilzerPrediction"
                    />
                  </div>
                  <div class="flex flex-col w-1/2">
                    <h3 class="mb-2 pt-12 text-lg font-medium text-gray-900">
                    </h3>



                    <form
                      id="form"
                      action="/predict"
                      method="POST"
                      enctype="multipart/form-data"
                    >
                      <select
                        name="plants"
                        id="plants"
                        class="bg-gray-100 border w-4/6 border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5"
                      >
                        <option>Choose the type</option>
                        <option value="fruit">Fruit</option>
                        <option value="Vegetable">Vegetable</option>
                      </select>
                      <input
                        type="file"
                        name="image"
                        accept="image/*"
                        onchange="loadFile(event)"
                        class="block w-full pt-4 text-sm text-gray-500 file:mr-4 file:py-2 file:px-4 file:rounded-full file:border-0 file:text-sm file:font-semibold file:bg-blue-100 file:text-blue-700 hover:file:bg-blue-200"
                      />
                      <img class="w-1/2 pt-8 pb-8 rounded-lg" id="output" />
                      <button
                        id="predict-btn"
                        type="submit"
                        class="text-white bg-blue-600 hover:bg-blue-500 w-32 focus:ring-4 focus:ring-blue-300 font-medium rounded-lg text-md px-5 py-2.5 mr-2 mb-2"
                      >
                        Predict
                      </button>
                    </form>
                  </div>
                </div>
              </section>
              <script>
                //For image preview
                var loadFile = function (event) {
                  var reader = new FileReader();
                  reader.onload = function () {
                    var output = document.getElementById("output");
                    output.src = reader.result;
                  };
                  reader.readAsDataURL(event.target.files[0]);
                };

              </script>
            </body>
          </html>
```

8. TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
HomePage_TC_OO1	Functional	Home Page	Verify user is able to see the home page or not.		1. Enter URL and click go 2. verify whether the user is able to see the home page.	Enter URL and click go	User able to see the home page	Working as expected	Pass	Nil	N	_	Muhammad Marzooq A
HomePage_TC_OO2	UI	Home Page	Verify the UI elements in Home Page		1. Enter URL and click go 2. Verify the UI elements in Home Page.	Enter URL and click go	Application should show below UI elements: Home Tab & Predict Tab	Working as expected	pass	Nil	N	_	Ranjithkumar.U
PredictPage_TC_OO 3	Functional	Predict page	Verify user is able to redirect to predict page or not.		1.Enter URL and click go 2.Click on Predict button 3.Verify whether the user to redirect to predict page or not.	Click the predict button in home page	User should navigate to Predict page	Working as expected	pass	Nil	N	_	Surya.P
PredictPage_TC_OO 4	UI	Predict page	Verify the UI elements in Predict Page		1. Enter URL and click go 2. Verify the UI elements in Predict Page.	Click the predict button and redirect to predict page	Application should show below UI elements: Dropdown List , Upload file Button, Predict button.	Working as expected	pass	Nil	N	_	Muhammad Marzooq.A, Romario.S
PredictPage_TC_OO 5	Functional	Predict page	Verify user is able to select the dropdown value or not.		1.Enter URL and click go 2.Click on Predict button 3. Verify whether the user to redirect to predict page or not. 4. Verify user is able to select the dropdown value or not.	Fruit or Vegetable	Application should shows user to choose fruit or vegetable option in dropdown list.	Working as expected	pass	Nil	N	_	Surya.P
PredictPage_TC_OO 6	Functional	Predict page	Verify user is able to upload the image or not.		1.Enter URL and click go 2.Click on Predict button 3.Verify whether the user to redirect to predict page or not. 4.Verify user is able to select the dropdown value or not. 5.Verify user is able to upload the images or not	Images to be Uploaded	Application should shows the uploaded image.	Working as expected	pass	Nil	N	_	Ranjithkumar.U
PredictPage_TC_OO 7	Functional	Predict page	Verify whether the image is predicted correctly or not		1.Enter URL and click go 2.Click on Predict button 3.Verify whether the user to redirect to predict page or not. 4.Verify user is able to select the dropdown value or not. 5. Verify user is able to upload the images or not 6. Verify whether the image is predicted correctly or not	Click the Predict Button	Application shows the predicted output	Working as expected	pass	Nil	N	_	Muhammad Marzooq.A, Surya.P

8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [Fertilizers Recommendation System for Disease Prediction] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and howthey were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
Leaf spots	10	4	2	3	19
Mosaic leaf pattern	9	6	3	6	24
Misshapen leaves	2	7	0	1	10
Yellow leaves	11	4	3	20	38
Fruit rots	3	2	1	0	6
Fruit spots	5	3	1	1	10
Blights	4	5	2	1	12
Totals	44	31	13	32	119

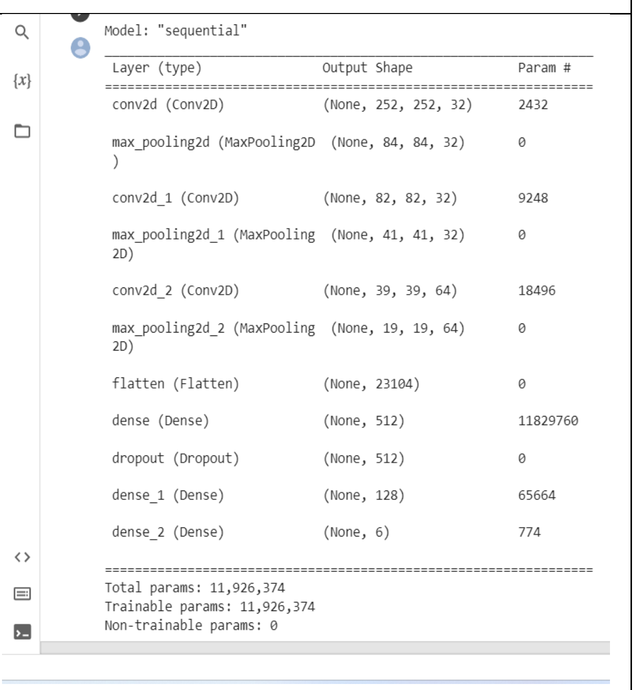
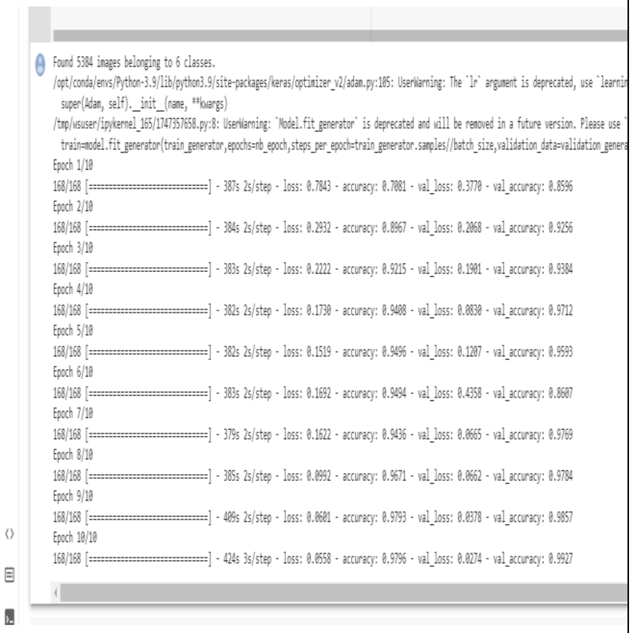
3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Leaf spots	17	0	0	17
Mosaic leaf pattern	51	0	0	51
Misshapen leaves	20	0	0	20
Yellow leaves	7	0	0	7
Fruit rots	9	0	0	9
Fruit spots	4	0	0	4
Blights	2	0	0	2

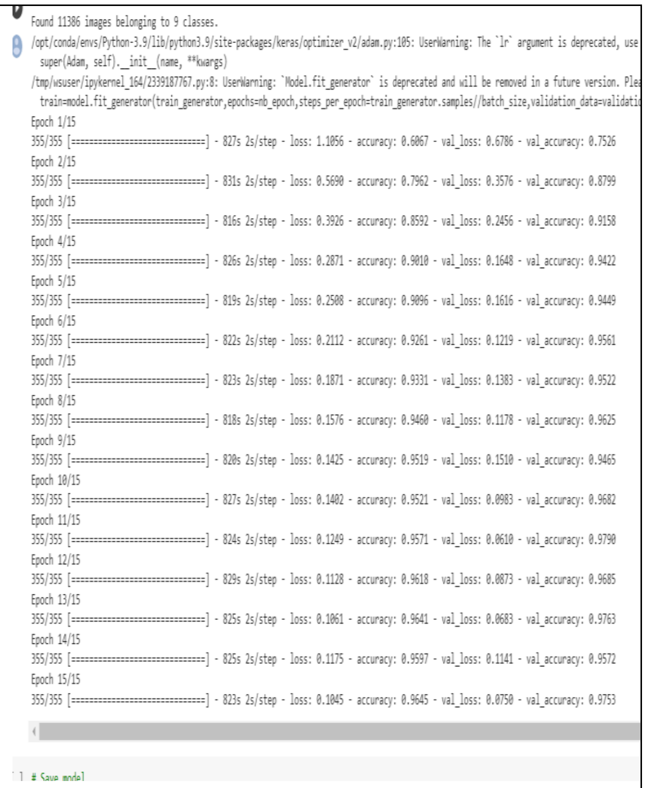
9. RESULTS

9.1 Performance Metrices

S.No.	Parameter	Values	Screenshot
1.	Model Summary	**Conv2D layers** – 32 Features are extracted from the input image **Relu** – Rectified linear activation to fire up neurons **Flatten layer**- 4D Array is converted into 1D array. **Dropout Layer** – It makes some neurons to relax **Softmax activation** – For the multiclassification output	
2.	Accuracy	**Training Accuracy –** i) Fruit dataset – **97.96%** **Validation Accuracy –** i) Fruit dataset – **99.27%** **Training Accuracy –**	i)**Fruit Dataset**  **ii)Vegetable Dataset**

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 252, 252, 32)      2432

 max_pooling2d (MaxPooling2D  (None, 84, 84, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 82, 82, 32)        9248

 max_pooling2d_1 (MaxPooling  (None, 41, 41, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 39, 39, 64)        18496

 max_pooling2d_2 (MaxPooling  (None, 19, 19, 64)       0
 2D)

 flatten (Flatten)           (None, 23104)             0

 dense (Dense)               (None, 512)               11829760

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 128)               65664

 dense_2 (Dense)             (None, 6)                 774

=================================================================
Total params: 11,926,374
Trainable params: 11,926,374
Non-trainable params: 0
```

```
Found 5384 images belonging to 6 classes.
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/keras/optimizer_v2/adam.py:105: UserWarning: The `lr` argument is deprecated, use `learnin
  super(Adam, self).__init__(name, **kwargs)
/tmp/wsuser/ipykernel_165/1747357658.py:8: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `
  train=model.fit_generator(train_generator,epochs=nb_epoch,steps_per_epoch=train_generator.samples//batch_size,validation_data=validation_genera
Epoch 1/10
168/168 [==============================] - 387s 2s/step - loss: 0.7843 - accuracy: 0.7081 - val_loss: 0.3770 - val_accuracy: 0.8596
Epoch 2/10
168/168 [==============================] - 384s 2s/step - loss: 0.2932 - accuracy: 0.8967 - val_loss: 0.2068 - val_accuracy: 0.9256
Epoch 3/10
168/168 [==============================] - 383s 2s/step - loss: 0.2222 - accuracy: 0.9215 - val_loss: 0.1901 - val_accuracy: 0.9384
Epoch 4/10
168/168 [==============================] - 382s 2s/step - loss: 0.1730 - accuracy: 0.9408 - val_loss: 0.0830 - val_accuracy: 0.9712
Epoch 5/10
168/168 [==============================] - 382s 2s/step - loss: 0.1519 - accuracy: 0.9496 - val_loss: 0.1207 - val_accuracy: 0.9593
Epoch 6/10
168/168 [==============================] - 383s 2s/step - loss: 0.1692 - accuracy: 0.9494 - val_loss: 0.4358 - val_accuracy: 0.8607
Epoch 7/10
168/168 [==============================] - 379s 2s/step - loss: 0.1622 - accuracy: 0.9436 - val_loss: 0.0665 - val_accuracy: 0.9769
Epoch 8/10
168/168 [==============================] - 385s 2s/step - loss: 0.0992 - accuracy: 0.9671 - val_loss: 0.0662 - val_accuracy: 0.9784
Epoch 9/10
168/168 [==============================] - 409s 2s/step - loss: 0.0601 - accuracy: 0.9793 - val_loss: 0.0378 - val_accuracy: 0.9857
Epoch 10/10
168/168 [==============================] - 424s 3s/step - loss: 0.0558 - accuracy: 0.9796 - val_loss: 0.0274 - val_accuracy: 0.9927
```

		ii)Vegetable dataset – **96.45%** **Validation Accuracy** – ii)Vegetable dataset – **97.53%**	

```
Found 11386 images belonging to 9 classes.
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/keras/optimizer_v2/adam.py:105: UserWarning: The `lr` argument is deprecated, use
  super(Adam, self).__init__(name, **kwargs)
/tmp/wsuser/ipykernel_164/2339187767.py:8: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Plea
  train=model.fit_generator(train_generator,epochs=nb_epoch,steps_per_epoch=train_generator.samples//batch_size,validation_data=validatio
Epoch 1/15
355/355 [==============================] - 827s 2s/step - loss: 1.1056 - accuracy: 0.6067 - val_loss: 0.6786 - val_accuracy: 0.7526
Epoch 2/15
355/355 [==============================] - 831s 2s/step - loss: 0.5690 - accuracy: 0.7962 - val_loss: 0.3576 - val_accuracy: 0.8799
Epoch 3/15
355/355 [==============================] - 816s 2s/step - loss: 0.3926 - accuracy: 0.8592 - val_loss: 0.2456 - val_accuracy: 0.9158
Epoch 4/15
355/355 [==============================] - 826s 2s/step - loss: 0.2871 - accuracy: 0.9010 - val_loss: 0.1648 - val_accuracy: 0.9422
Epoch 5/15
355/355 [==============================] - 819s 2s/step - loss: 0.2508 - accuracy: 0.9096 - val_loss: 0.1616 - val_accuracy: 0.9449
Epoch 6/15
355/355 [==============================] - 822s 2s/step - loss: 0.2112 - accuracy: 0.9261 - val_loss: 0.1219 - val_accuracy: 0.9561
Epoch 7/15
355/355 [==============================] - 823s 2s/step - loss: 0.1871 - accuracy: 0.9331 - val_loss: 0.1383 - val_accuracy: 0.9522
Epoch 8/15
355/355 [==============================] - 818s 2s/step - loss: 0.1576 - accuracy: 0.9460 - val_loss: 0.1178 - val_accuracy: 0.9625
Epoch 9/15
355/355 [==============================] - 820s 2s/step - loss: 0.1425 - accuracy: 0.9519 - val_loss: 0.1510 - val_accuracy: 0.9465
Epoch 10/15
355/355 [==============================] - 827s 2s/step - loss: 0.1402 - accuracy: 0.9521 - val_loss: 0.0983 - val_accuracy: 0.9682
Epoch 11/15
355/355 [==============================] - 824s 2s/step - loss: 0.1249 - accuracy: 0.9571 - val_loss: 0.0610 - val_accuracy: 0.9790
Epoch 12/15
355/355 [==============================] - 829s 2s/step - loss: 0.1128 - accuracy: 0.9618 - val_loss: 0.0873 - val_accuracy: 0.9685
Epoch 13/15
355/355 [==============================] - 825s 2s/step - loss: 0.1061 - accuracy: 0.9641 - val_loss: 0.0683 - val_accuracy: 0.9763
Epoch 14/15
355/355 [==============================] - 825s 2s/step - loss: 0.1175 - accuracy: 0.9597 - val_loss: 0.1141 - val_accuracy: 0.9572
Epoch 15/15
355/355 [==============================] - 823s 2s/step - loss: 0.1045 - accuracy: 0.9645 - val_loss: 0.0750 - val_accuracy: 0.9753

[ ] # Save model
```

\

10. ADVANTAGES & DISADVANTAGES

10.1 ADVANTAGES

- The proposed model here produces very high accuracy of classification.
- Very large datasets can also be trained and tested.
- Images of very high can be resized within the proposed itself.
- No need to consult any specialists for leaf disease
- Fully automated system.
- Early Detection of leaf Disease

10.2 DISADVANTAGES

- For training and testing, the proposed model requires very high computational

time.

- The neural network architecture used in this project work has high complexity.
- When a plant is infected with multiple diseases the system may not predict all the diseases due to the mixed symptoms

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 proposed model in this project work can be extended to image recognition. The entire model can be converted to application software using python to exe software. The real time image classification, image recognition and video processing are possible with help OpenCV python library. This project work can be extended for security applications such as figure print recognition, iris recognition and face recognition.

13. APPENDIX

Source Code :

Final Code - Google Drive or

https://drive.google.com/drive/folders/1FlLBApkLZrVLpYcLLW5ni4GJTIwjgiOF

GitHub & Project Demo Link

Github Repo : https://github.com/IBM-EPBL/IBM-Project-39247-1660402498

Project Demo Link :
https://drive.google.com/file/d/1cKpCEEL75PpcsFF_kSKyr2gmaN_85idt/view