FERTILIZER RECOMMENDATION SYSTEM FOR PLANT DISEASES PREDICTION

TEAM ID : PNT2022TMID39322

TEAM MEMBERS:

V.D.Vinisha

S.Kaviya

S.Iswarya

A.Akalya

INDEX

1. INTRODUCTION

1.1 PROJECT OVERVIEW:

Fertilizer recommendation for plant diseases project is for farmer who is lack of their knowledge in plant diseases and how to give correct fertilizer for affected plant leaf. we use deep learning for predict diseases ,this plays major role in future scope . Lots of scope in this project now a days. Agriculture is most important in day to day activity .so our project have lots of scope for our project.

1.2 PURPOSE:

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.

2. LITERATURE SURVEY:

2.1 EXISTING PROBLEM:

YEAR	TITLE & AUTHOR	TECHNIQUE	PROBLEM STATEMENT	PROS & CONS
2018	User centered design of fertilizer recommendation System for small holder farmer By nikil mallareddy	Site specific nutrient management (ssnm)	Soil health is major problem in india. Imbalanced fertilizer use, a negative consumption of fertilizer these cause severe soil damage and plant death.	**Pros:** Using ssnm technique soil type has been identified and proper fertilizer has been recommended for soil and plant . **CONS:** Lack of soil testing sevices. poor soil health due to low fertilizer use. Excess fertilizer causes effect to environment.
2021	Plant disease identification using CNN . BY Mustafa abdo mohammed alhammadi,prof .amol ashok bhilare	**CNN**	Leaf of plant had been infected or attacked by some disease, the other areas had been exposed to be infected.It will decrease leaf yield and it also reduce farmer income	**PROS:** Low cost, low power consumption ,high accuracy the sensor has excellent sensitivity with quick response time. **CONS:** It only tell about diseases not about prevention.
2021	Soil based fertilizer recommendation system for crop diseases prediction system. BY Dr. p. pandi selvi, p.poornima	Long and short term memory algorithm ,sensor.	It is very much useful to analyze the soil nutrient type efficiently,kind of leaf diseases pesent in crop and predict fertilizer in proficient manner	**PROS**: It has check soil type plant type and then only recommend fertilizer diseased crop. **CONS**: Certain files regrading leaf diseases or soil type or fertilizer may not be updated.
2022	Precision agriculture using Machine Learning and IOT. BY Atharva labhasetwar Venkata Narayana bommanabonia Kundan patil	SVM , Logistic regression , Random forest,precision agriculture.	The most revelant problem faced by farmer is that they do not use the appropriate crop for their land and fertilizer .this is major fault .	**Pros:** Improve farm management efficiency by adjusting field/crop treatment .It will reduce excessive chemical usage in crop production. **CONS:** Accuracy depend upon input dataset . Complexity grows with data.

2.2 REFERENCES:

1] Chohan, Murk & Khan, Adil & Chohan, Rozina & Hassan, Muhammad. (2020). Plant Disease Detection using Deep Learning. International Journal of Recent

Technology and Engineering. 9. 909-914. 10.35940/ijrte.A2139.059120.

[2] Mohanty SP, Hughes DP. Using Deep Learning for Image-Based Plant Disease Detection. Front Plant Sci. 2016 Sep 22;7:1419. doi: 10.3389/fpls.2016.01419.

PMID: 27713752; PMCID: PMC5032846.

[3] Vijeta Shrivastava, Pushpanjali, Samreen Fatima, Indrajit Das, " Plant leaf diseases detection and classification using machine learning", International

Journal of Latest Trends in Engineering and Technology , Vol.10, Issue.2, April 2018.

[4] Ferentinos, Konstantinos. (2018). “Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture.” 145. 311-318.

10.1016/j.compag.2018.01.009.

[5] Guo, Yan, et al. “Plant Disease Identification Based on Deep Learning Algorithm in Smart Farming.” Discrete Dynamics in Nature and Society, Hindawi, 18

Aug. 2020.

[6] Arsenovic Marko, Karanovic Mirjana, Sladojevic S. “Solving Current Limitations of Deep Learning Based Approaches for Plant Disease

Detection.” Symmetry. 2019; 11(7):939

2.3 PROBLEM STATEMENT:

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.

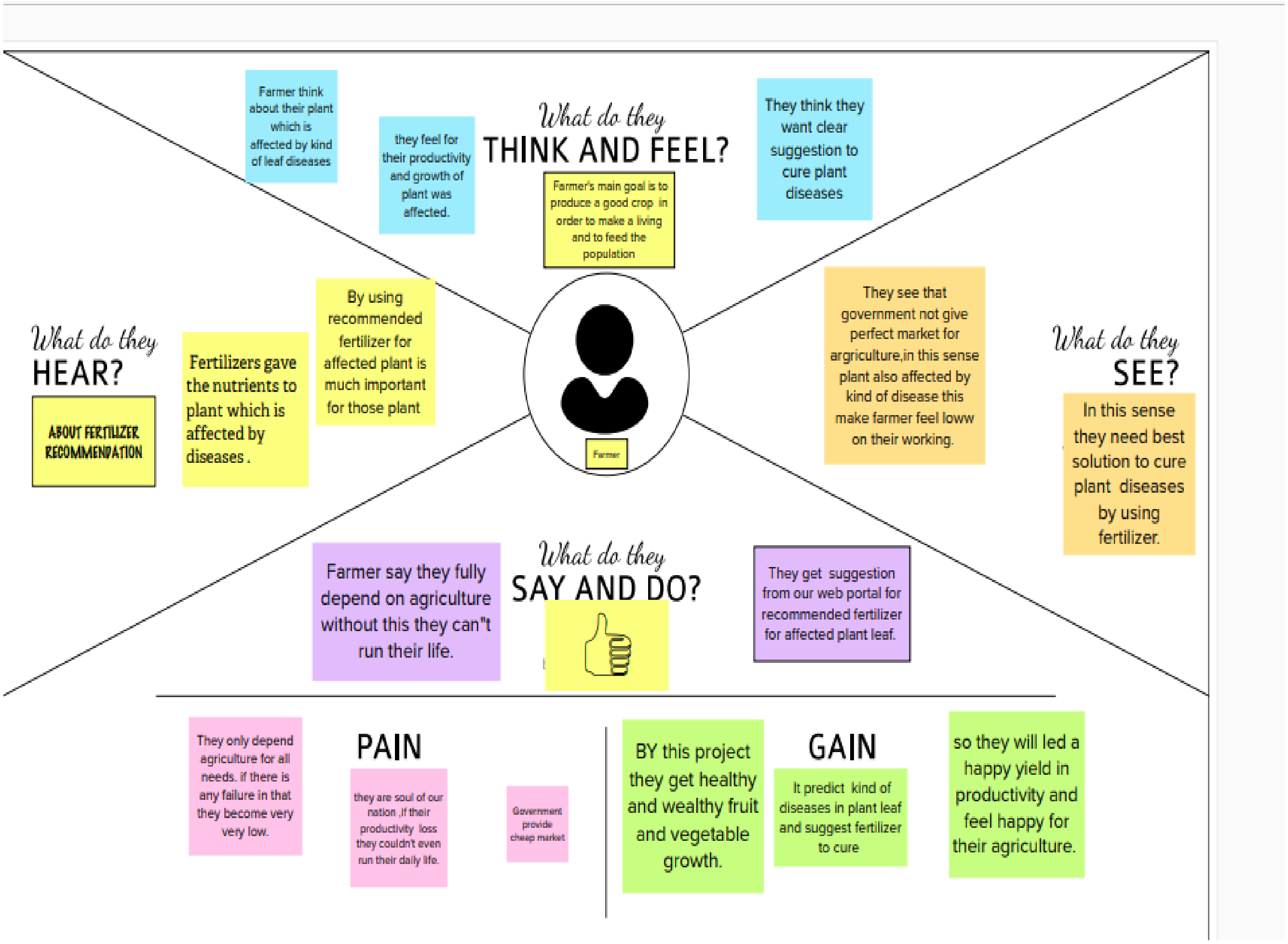
Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality.

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.

Finding leaf disease is important to yield and quality

3 . IDEATION & PROPOSED SOLUTION :

3.1 Empathy Map Canvas:



What do they THINK AND FEEL?

Farmer think about their plant which is affected by kind of leaf diseases

they feel for their productivity and growth of plant was affected.

Farmer's main goal is to produce a good crop in order to make a living and to feed the population

They think they want clear suggestion to cure plant diseases

What do they HEAR?

ABOUT FERTILIZER RECOMMENDATION

Fertilizers gave the nutrients to plant which is affected by diseases .

By using recommended fertilizer for affected plant is much important for those plant

Farmer

What do they SEE?

They see that government not give perfect market for argriculture,in this sense plant also affected by kind of disease this make farmer feel loww on their working.

In this sense they need best solution to cure plant diseases by using fertilizer.

What do they SAY AND DO?

Farmer say they fully depend on agriculture without this they can"t run their life.

They get suggestion from our web portal for recommended fertilizer for affected plant leaf.

PAIN

They only depend agriculture for all needs. if there is any failure in that they become very very low.

they are soul of our nation ,if their productivity loss they couldn't even run their daily life.

Government provide cheap market

GAIN

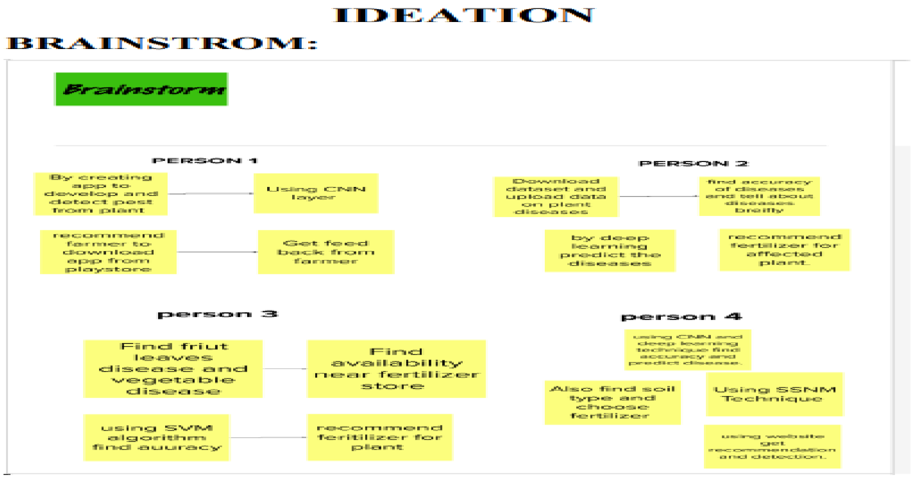
BY this project they get healthy and wealthy fruit and vegetable growth.

It predict kind of diseases in plant leaf and suggest fertilizer to cure

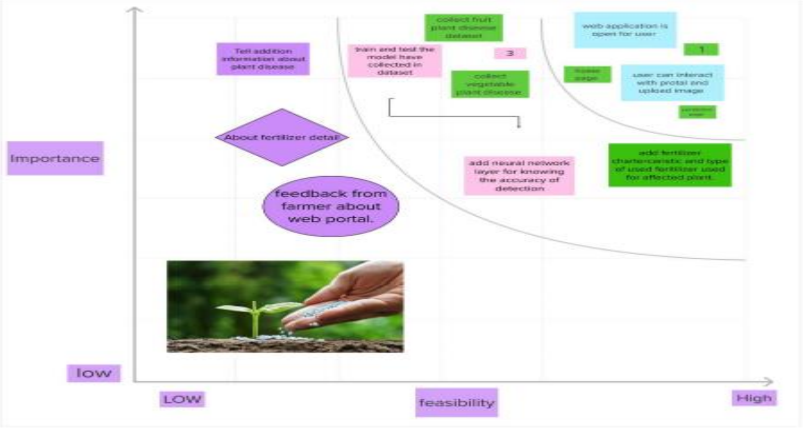
so they will led a happy yield in productivity and feel happy for their agriculture.

3.2 Ideation & Brainstorming :

Brainstorming:


IDEATION
BRAINSTROM:
Brainstorm
PERSON 1
By creating app to develop and detect pest from plant
Using CNN layer
recommend farmer to download app from playstore
Get feed back from farmer
PERSON 2
Download dataset and upload data on plant diseases
find accuracy of diseases and tell about diseases breifly
by deep learning predict the diseases
recommend fertilizer for affected plant.
person 3
Find friut leaves disease and vegetable disease
Find availability near fertilizer store
using SVM algorithm find auuracy
recommend feritilizer for plant
person 4
using CNN and deep learning technique find accuracy and predict disease.
Also find soil type and choose fertilizer
Using SSNM Technique
using website get recommendation and detection.

IDEATION:


Importance
low
LOW
feasibility
High
Tell addition information about plant disease
About fertilizer detail
feedback from farmer about web portal.
train and test the model have collected in dataset
collect fruit plant disease dataset
3
collect vegetable plant disease
add neural network layer for knowing the accuracy of detection
web application is open for user
1
home page
user can interact with protal and upload image
add fertilizer chartercaristic and type of used fertilizer used for affected plant.

3.3 Proposed Solution :

SOLUTION DESCRIPTION:

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.

These different leaf symptoms and diseases are predicted in image processing ,this can be down by CNN layer the image is analyzed and processed, and the fertilizer needed for plant growth is predicted.

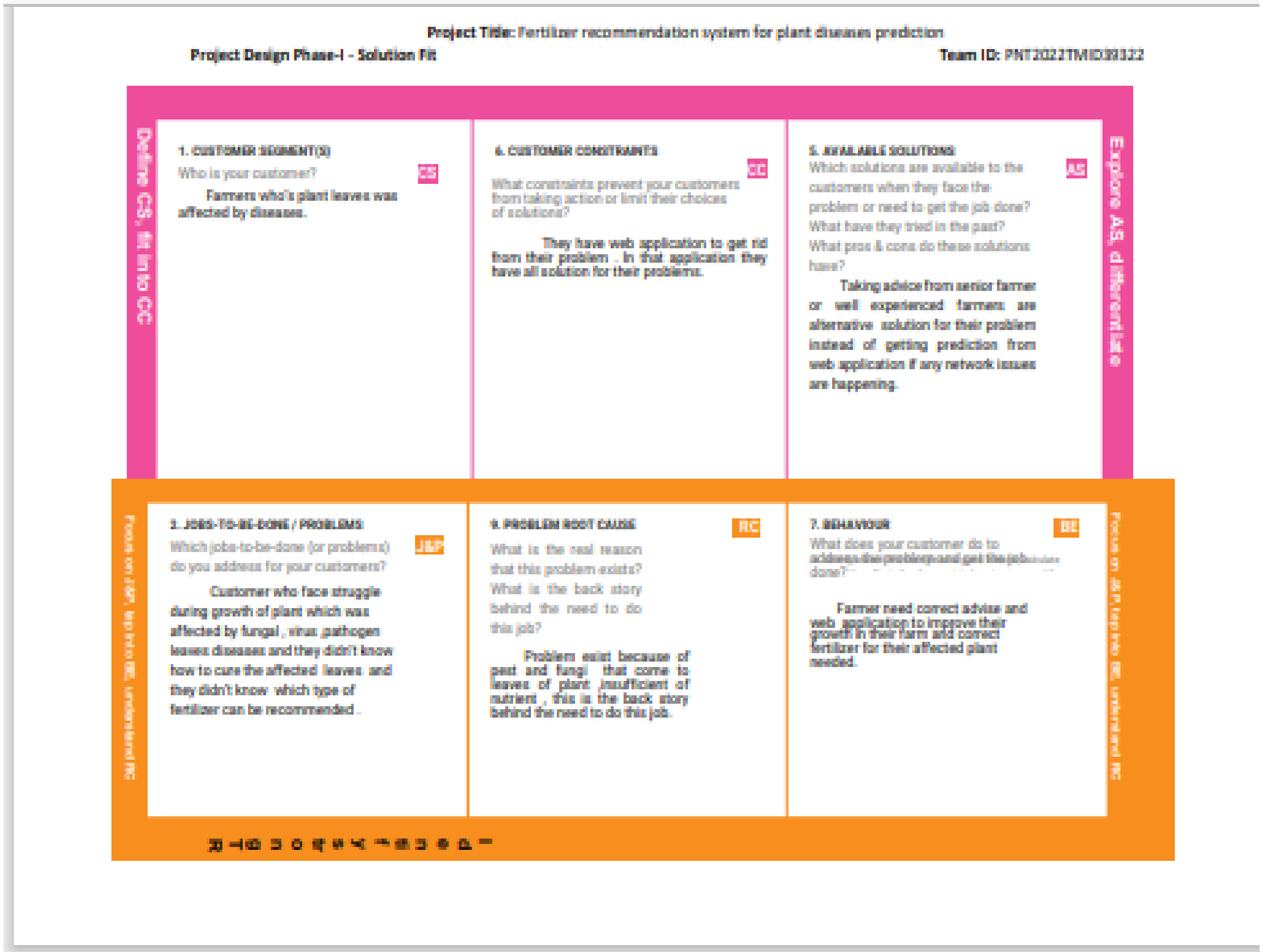
NOVELTY:

The global fertilizer market was valued at $184.60 billion in 2021,and is projected to reach $251.57 billion by 2030, growing at a CAGR of 3.55% from 2022 to 2030.

For fertilizer, market is segmented into application ,type and form.

Now a days, Without the addition of fertilizer crop yield and agriculture productivity would be significantly reduced.

3.4 Problem Solution fit :



Project Title: Fertilizer recommendation system for plant diseases prediction

Project Design Phase-I - Solution Fit **Team ID:** PNT2022TMID39322

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

Farmers who's plant leaves was affected by diseases.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions?

They have web application to get rid from their problem . In that application they have all solution for their problems.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have?

Taking advice from senior farmer or well experienced farmers are alternative solution for their problem instead of getting prediction from web application if any network issues are happening.

Explore AS, differentiate

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

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

Which jobs-to-be-done (or problems) do you address for your customers?

Customer who face struggle during growth of plant which was affected by fungal , virus ,pathogen leaves diseases and they didn't know how to cure the affected leaves and they didn't know which type of fertilizer can be recommended .

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?

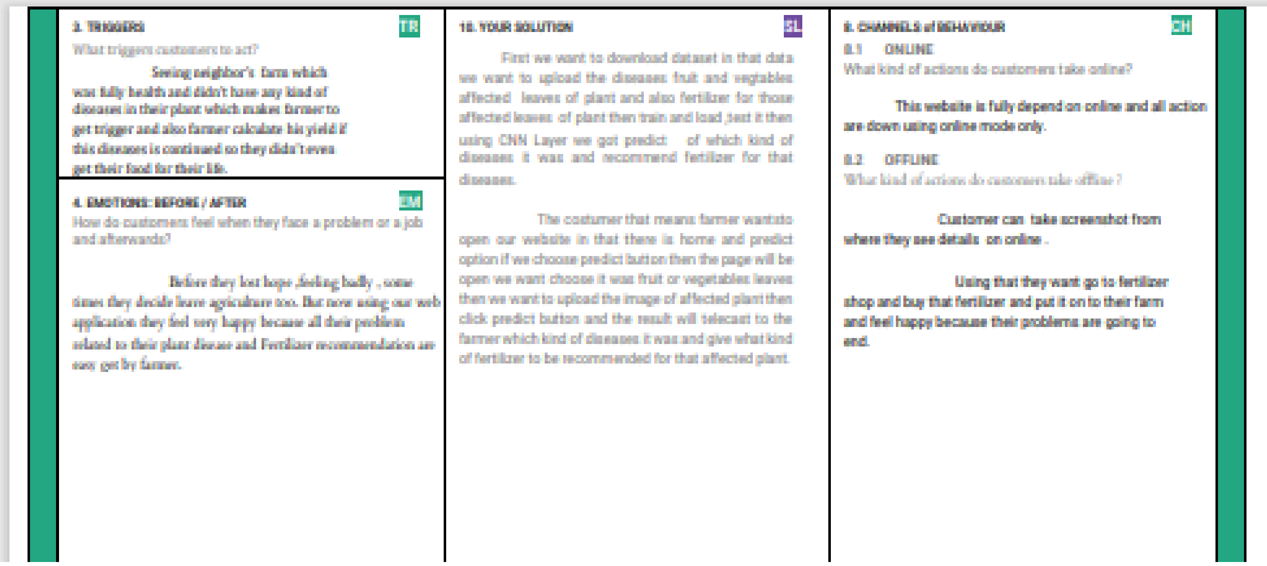
Problem exist because of pest and fungi that come to leaves of plant ,insufficient of nutrient , this is the back story behind the need to do this job.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?

Farmer need correct advise and web application to improve their growth in their farm and correct fertilizer for their affected plant needed.

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



3. TRIGGERS TR

What triggers customers to act?

Seeing neighbor's farm which was fully health and didn't have any kind of diseases in their plant which makes farmer to get trigger and also farmer calculate his yield if this diseases is continued so they didn't even get their food for their life.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?

Before they lost hope ,feeling badly , some times they decide leave agriculture too. But now using our web application they feel very happy because all their problem related to their plant disease and Fertilizer recommendation are easy get by farmer.

10. YOUR SOLUTION SL

First we want to download dataset in that data we want to upload the diseases fruit and vegtables affected leaves of plant and also fertilizer for those affected leaves of plant then train and load ,test it then using CNN Layer we got predict of which kind of diseases it was and recommend fertilizer for that diseases.

The costumer that means farmer wantsto open our website in that there is home and predict option if we choose predict button then the page will be open we want choose it was fruit or vegetables leaves then we want to upload the image of affected plant then click predict button and the result will telecast to the farmer which kind of diseases it was and give what kind of fertilizer to be recommended for that affected plant.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

What kind of actions do customers take online?

This website is fully depend on online and all action are down using online mode only.

8.2 OFFLINE

What kind of actions do customers take offline ?

Customer can take screenshot from where they see details on online .

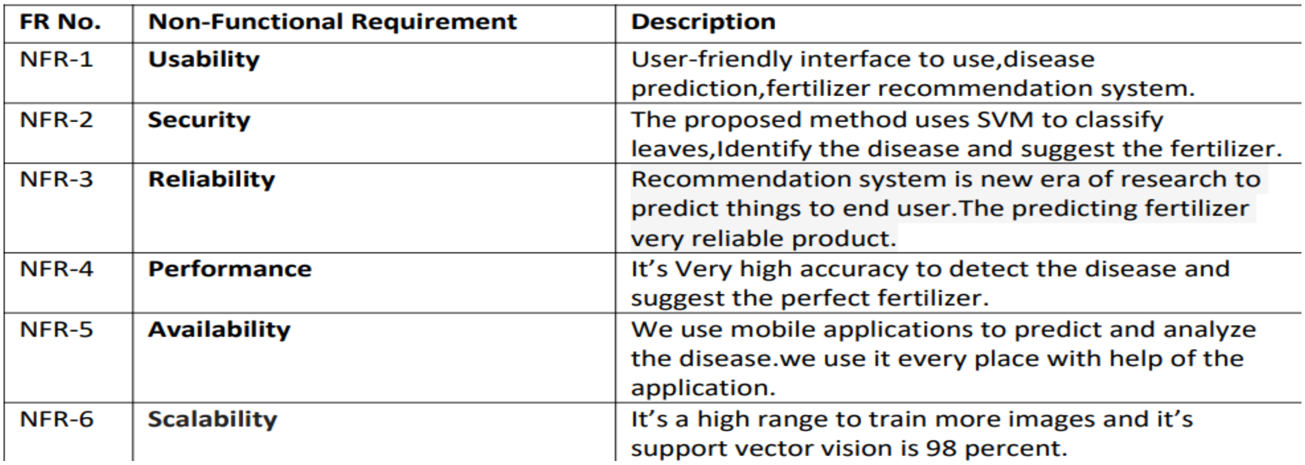
Using that they want go to fertilizer shop and buy that fertilizer and put it on to their farm and feel happy because their problems are going to end.

4. REQUIREMENT ANALYSIS :

4.1 Functional requirement:

FR	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR1	HOME	About agriculture and motivation on agriculture.
FR2	Predict page	User want choose fruit or vegetable option and press upload a image button.
FR3	RESULT	When user upload the image of affected plant it will predict and tell about which fertilizer is best for affected plant.

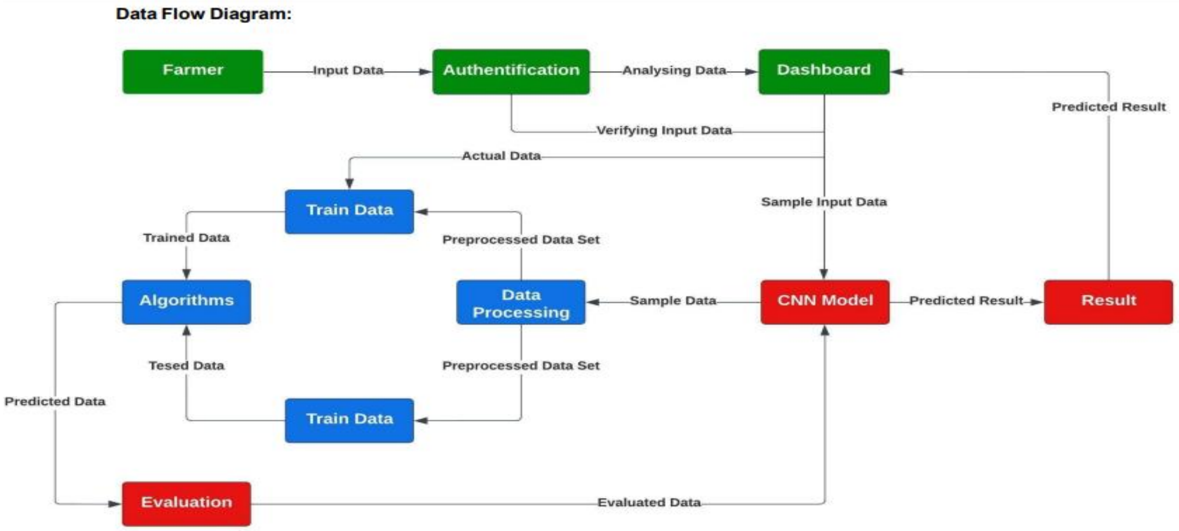
4.2 Non-Functional requirement:



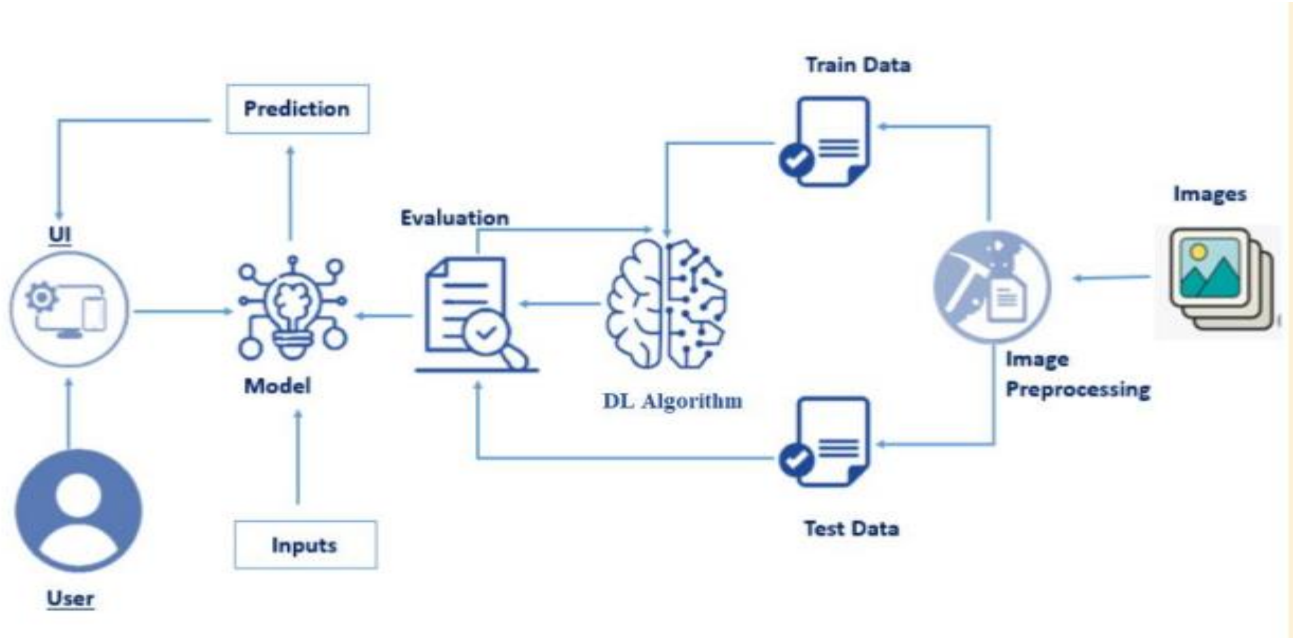
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	User-friendly interface to use,disease prediction,fertilizer recommendation system.
NFR-2	**Security**	The proposed method uses SVM to classify leaves,Identify the disease and suggest the fertilizer.
NFR-3	**Reliability**	Recommendation system is new era of research to predict things to end user.The predicting fertilizer very reliable product.
NFR-4	**Performance**	It's Very high accuracy to detect the disease and suggest the perfect fertilizer.
NFR-5	**Availability**	We use mobile applications to predict and analyze the disease.we use it every place with help of the application.
NFR-6	**Scalability**	It's a high range to train more images and it's support vector vision is 98 percent.

5. PROJECT DESIGN :

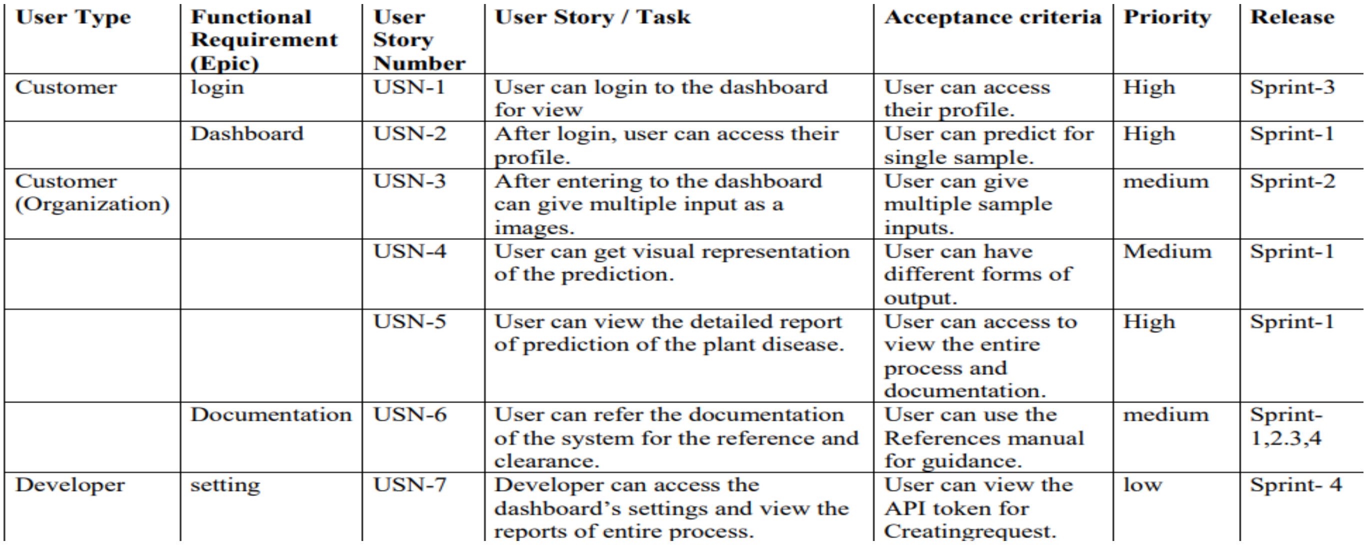
5.1 Data Flow Diagrams :



5.1 Solution & Technical Architecture :



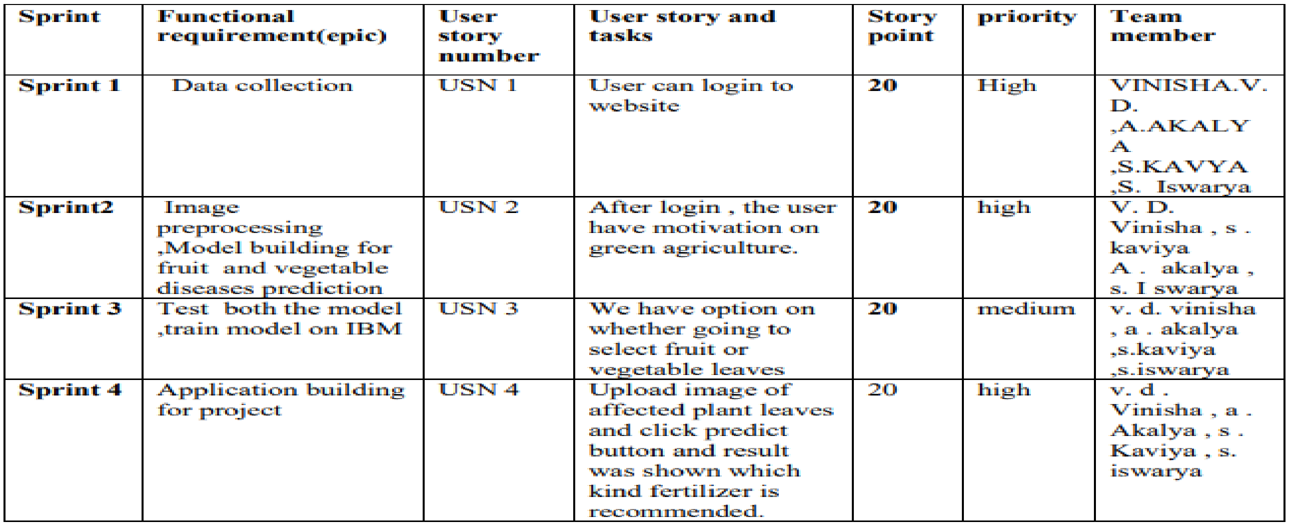
5.3 User Stories :



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	login	USN-1	User can login to the dashboard for view	User can access their profile.	High	Sprint-3
	Dashboard	USN-2	After login, user can access their profile.	User can predict for single sample.	High	Sprint-1
Customer (Organization)		USN-3	After entering to the dashboard can give multiple input as a images.	User can give multiple sample inputs.	medium	Sprint-2
		USN-4	User can get visual representation of the prediction.	User can have different forms of output.	Medium	Sprint-1
		USN-5	User can view the detailed report of prediction of the plant disease.	User can access to view the entire process and documentation.	High	Sprint-1
	Documentation	USN-6	User can refer the documentation of the system for the reference and clearance.	User can use the References manual for guidance.	medium	Sprint-1,2.3,4
Developer	setting	USN-7	Developer can access the dashboard's settings and view the reports of entire process.	User can view the API token for Creatingrequest.	low	Sprint- 4

6. PROJECT PLANNING & SCHEDULING :

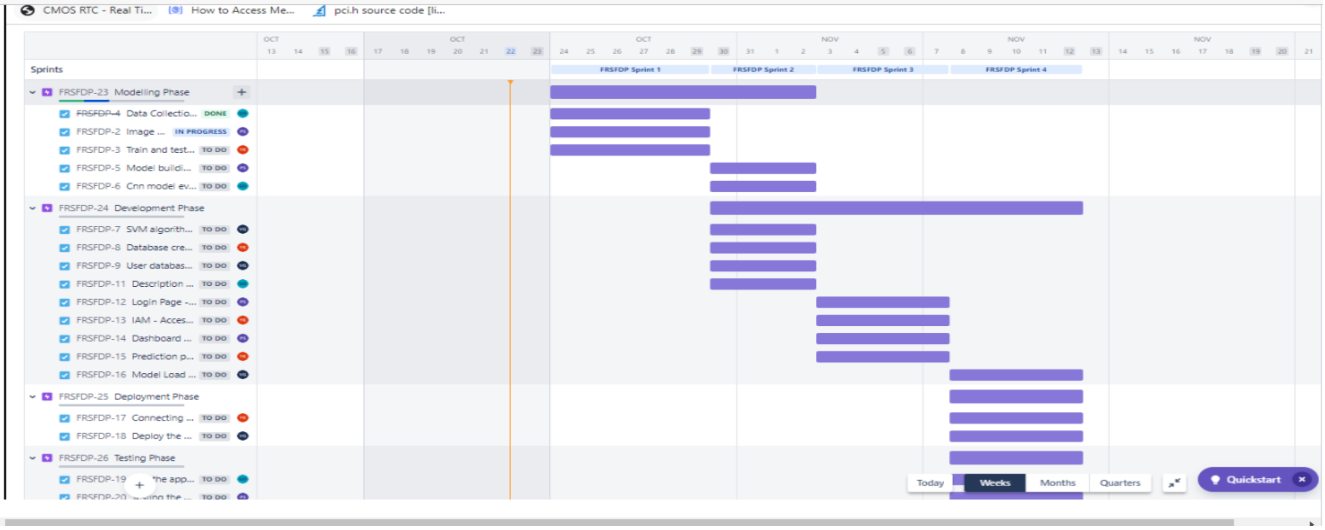
6.1Sprint Delivery Schedule :



Sprint	Functional requirement(epic)	User story number	User story and tasks	Story point	priority	Team member
Sprint 1	Data collection	USN 1	User can login to website	20	High	VINISHA.V. D. ,A.AKALYA ,S.KAVYA ,S. Iswarya
Sprint2	Image preprocessing ,Model building for fruit and vegetable diseases prediction	USN 2	After login , the user have motivation on green agriculture.	20	high	V. D. Vinisha , s . kaviya A . akalya , s. I swarya
Sprint 3	Test both the model ,train model on IBM	USN 3	We have option on whether going to select fruit or vegetable leaves	20	medium	v. d. vinisha , a . akalya ,s.kaviya ,s.iswarya
Sprint 4	Application building for project	USN 4	Upload image of affected plant leaves and click predict button and result was shown which kind fertilizer is recommended.	20	high	v. d . Vinisha , a . Akalya , s . Kaviya , s. iswarya

Sprint	Total story points	duration	Sprint start date	Sprint end date (planned)	Story point completed (as on planned end date)	Sprint release date(actual)
Sprint 1	20	6 days	24-oct -2022	29-oct-2022		29-oct-2022
Sprint 2	20	6 days	31-oct-2022	05-nov-2022		05-nov-2022
Sprint 3	20	6 days	07-nov-2022	12-nov-2022		12-nov-2022
Sprint 4	20	6 days	14-nov-2022	19-nov-2022		19-nov-2022

6.2 Reports from JIRA :



Sprints

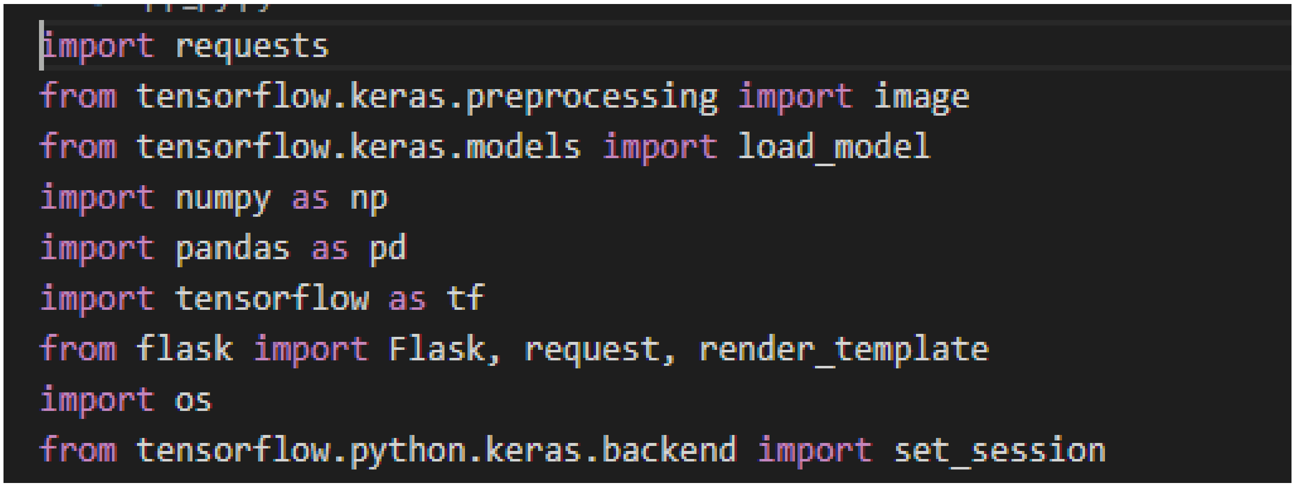
FRSFDP-23 Modelling Phase
- ~~FRSFDP-4~~ Data Collectio... DONE
- FRSFDP-2 Image ... IN PROGRESS
- FRSFDP-3 Train and test... TO DO
- FRSFDP-5 Model buildi... TO DO
- FRSFDP-6 Cnn model ev... TO DO

FRSFDP-24 Development Phase
- FRSFDP-7 SVM algorith... TO DO
- FRSFDP-8 Database cre... TO DO
- FRSFDP-9 User databas... TO DO
- FRSFDP-11 Description ... TO DO
- FRSFDP-12 Login Page -... TO DO
- FRSFDP-13 IAM - Acces... TO DO
- FRSFDP-14 Dashboard ... TO DO
- FRSFDP-15 Prediction p... TO DO
- FRSFDP-16 Model Load ... TO DO

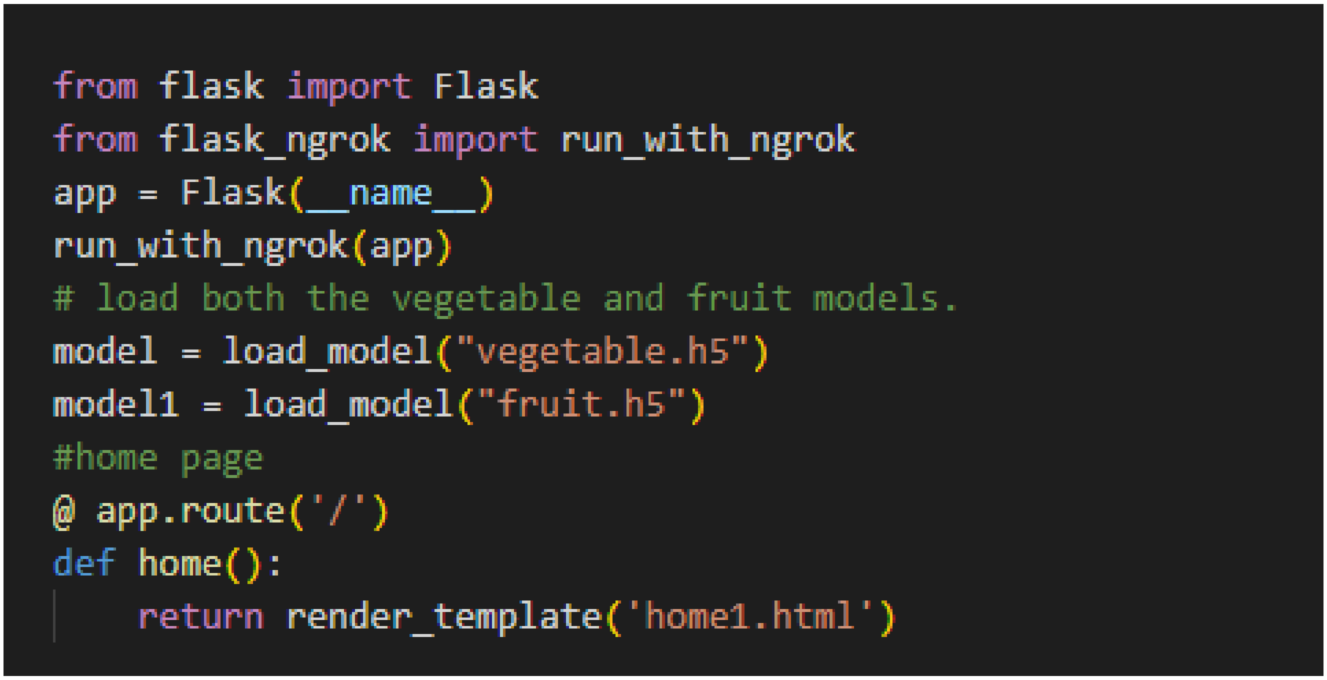
FRSFDP-25 Deployment Phase
- FRSFDP-17 Connecting ... TO DO
- FRSFDP-18 Deploy the ... TO DO

FRSFDP-26 Testing Phase
- FRSFDP-19 ...the app... TO DO
- FRSFDP-20 ...ing the ... TO DO

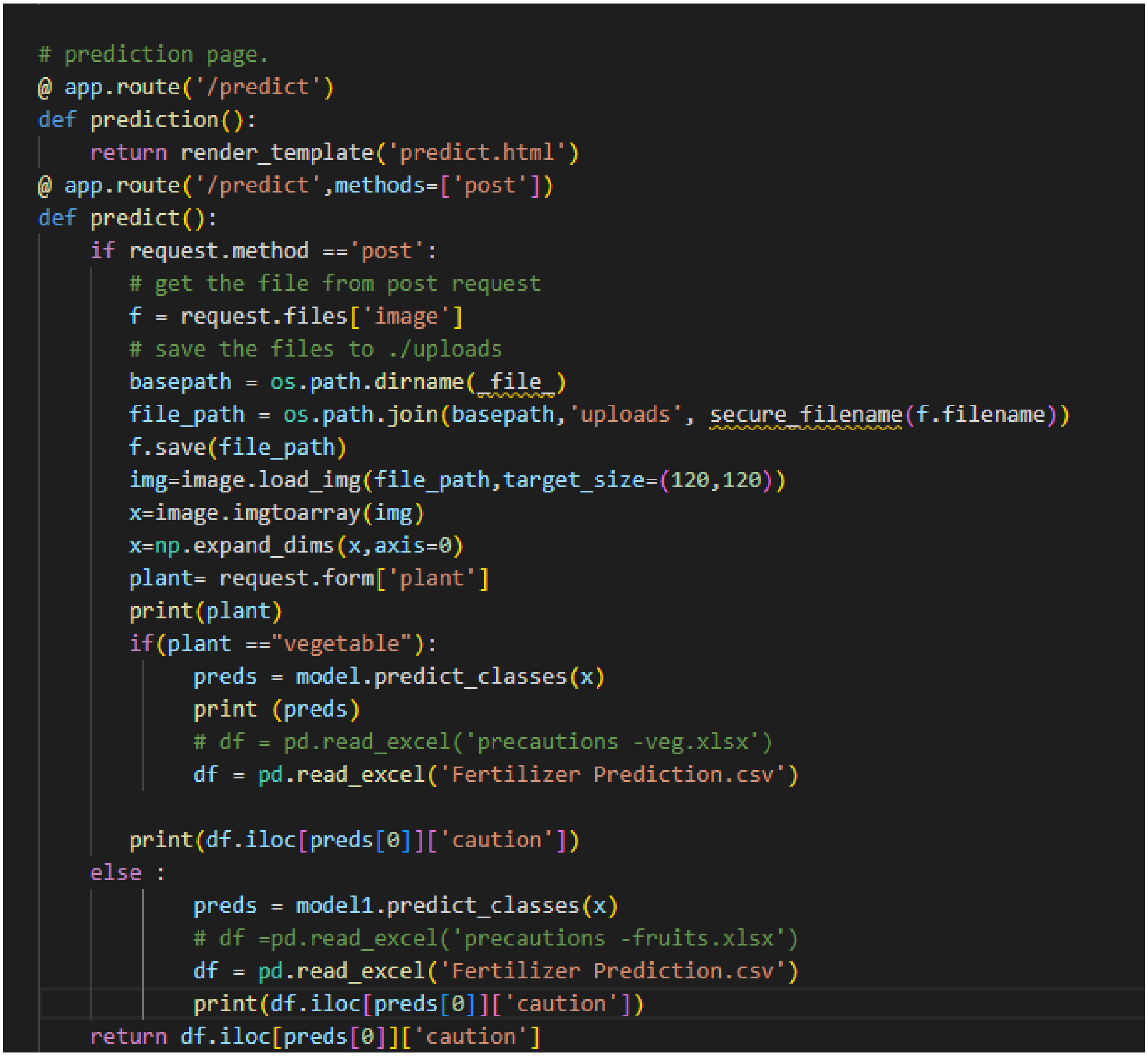
7. CODING & SOLUTIONING:



```
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
import os
from tensorflow.python.keras.backend import set_session
```



```
from flask import Flask
from flask_ngrok import run_with_ngrok
app = Flask(__name__)
run_with_ngrok(app)
# load both the vegetable and fruit models.
model = load_model("vegetable.h5")
model1 = load_model("fruit.h5")
#home page
@ app.route('/')
def home():
    return render_template('home1.html')
```



```
# prediction page.
@ app.route('/predict')
def prediction():
    return render_template('predict.html')
@ app.route('/predict',methods=['post'])
def predict():
    if request.method =='post':
       # get the file from post request
       f = request.files['image']
       # save the files to ./uploads
       basepath = os.path.dirname(_file_)
       file_path = os.path.join(basepath,'uploads', secure_filename(f.filename))
       f.save(file_path)
       img=image.load_img(file_path,target_size=(120,120))
       x=image.imgtoarray(img)
       x=np.expand_dims(x,axis=0)
       plant= request.form['plant']
       print(plant)
       if(plant =="vegetable"):
            preds = model.predict_classes(x)
            print (preds)
            # df = pd.read_excel('precautions -veg.xlsx')
            df = pd.read_excel('Fertilizer Prediction.csv')

       print(df.iloc[preds[0]]['caution'])
    else :
            preds = model1.predict_classes(x)
            # df =pd.read_excel('precautions -fruits.xlsx')
            df = pd.read_excel('Fertilizer Prediction.csv')
            print(df.iloc[preds[0]]['caution'])
    return df.iloc[preds[0]]['caution']
```

8. TESTING

8.1 TEST CASES:

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

Section Total	Total Cases	Not Tested	Fail	Pass
Home page	10	0	4	6
Predict page	20	0	15	5
Upload	7	0	2	5
result	6	0	4	2

8.2 USER ACCEPTANCE TESING:

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

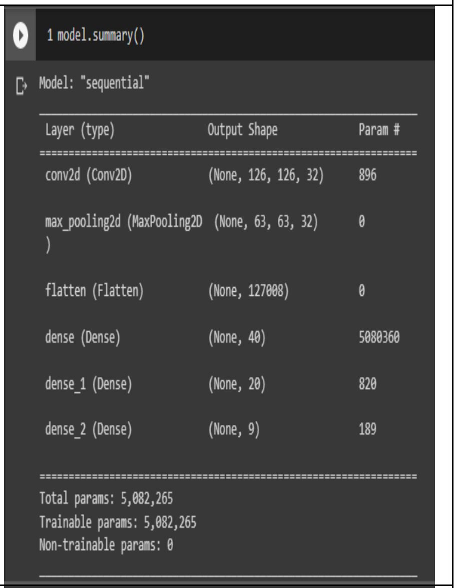
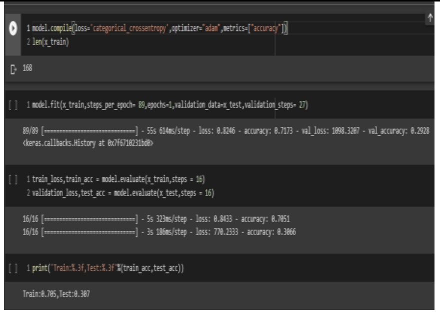
Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won’t Fix	0	5	2	1	8
Totals	24	14	13	26	77

9.RESULT:

9.1 PERFORMANCES METRICS:

S.NO	PARAMETER	VALUE	SCREENSHOTS
1	Model Summary	conv2d (Conv2D) (None, 126, 126, 32) 896 max_pooling2d (MaxPooling2D (None, 63, 63, 32) 0) flatten (Flatten) (None, 127008) 0 dense (Dense) (None, 40) 5080360 dense_1 (Dense) (None, 20) 820 dense_2 (Dense) (None, 9) 189 === Total params: 5,082,265 Trainable params: 5,082,265 Non-trainable params: 0	 model.summary() Model: "sequential" ____ Layer (type) Output Shape Param # === conv2d (Conv2D) (None, 126, 126, 32) 896 max_pooling2d (MaxPooling2D (None, 63, 63, 32) 0) flatten (Flatten) (None, 127008) 0 dense (Dense) (None, 40) 5080360 dense_1 (Dense) (None, 20) 820 dense_2 (Dense) (None, 9) 189 === Total params: 5,082,265 Trainable params: 5,082,265 Non-trainable params: 0 ____
2	Accuracy	16/16 [=============================] - 5s 323ms/step - loss: 0.8433 - accuracy: 0.7051 16/16 [=============================] - 3s 186ms/step - loss: 770.2333 - accuracy: 0.3066 print('Train:%.3f,Test:%.3f' %(train_acc,test_acc)) Train:0.705,Test:0.307	 model.compile(loss='categorical_crossentropy',optimizer="adam",metrics=["accuracy"]) len(x_train) 168 model.fit(x_train,steps_per_epoch= 89,epochs=1,validation_data=x_test,validation_steps= 27) 89/89 [==============================] - 55s 614ms/step - loss: 0.8246 - accuracy: 0.7173 - val_loss: 1098.3207 - val_accuracy: 0.2928 <keras.callbacks.History at 0x7f6710231bd0> train_loss,train_acc = model.evaluate(x_train,steps = 16) validation_loss,test_acc = model.evaluate(x_test,steps = 16) 16/16 [==============================] - 5s 323ms/step - loss: 0.8433 - accuracy: 0.7051 16/16 [==============================] - 3s 186ms/step - loss: 770.2333 - accuracy: 0.3066 print('Train:%.3f,Test:%.3f'%(train_acc,test_acc)) Train:0.705,Test:0.307
3	Confidence Score (Only Yolo Projects)	Class Detected - 9 Confidence Score -9	

10. ADVANTAGES &DISADVANTAGES:

ADVANTAGES:

- Farmers can interact with the portal build
- Interacts with the user interface to upload images of diseased leaf
- Our model built analyses the Disease and suggests the farmer with fertilizers are to be used .
- This one is very much usefull to farmers who didn’t know more about farming crop.
- This application is user friendly application to farmers.
- We can use this application from any region of world.

DISADVANTAGES:

- Error may happen when was user try to upload and give predict button some times.
- Need internet connection to predict.
- Some times complicated to maintain this application.

11. CONCLUSION:

Even though there are various methods for detecting and classifying plant diseases using automatic or computer vision, research into this field has been lacking. In addition, there are few commercial options, with the exception of those focusing on the identification of plant species via photographs. Over the last few years, there has been tremendous progress in the performance of convolutional neural networks. The new generation of convolutional neural networks (CNNs) has shown promising results in the field of image recognition. A novel approach to automatically classifying and detecting plant diseases from leaf images was examined through this project utilizing deep learning techniques. With an accuracy of 90%, the developed model could distinguish healthy leaves from eight diseases that could be observed visually. On the basis of this high level of performance, it becomes apparent that convolutional neural networks are highly suitable for automatic diagnosis and detection of plants.

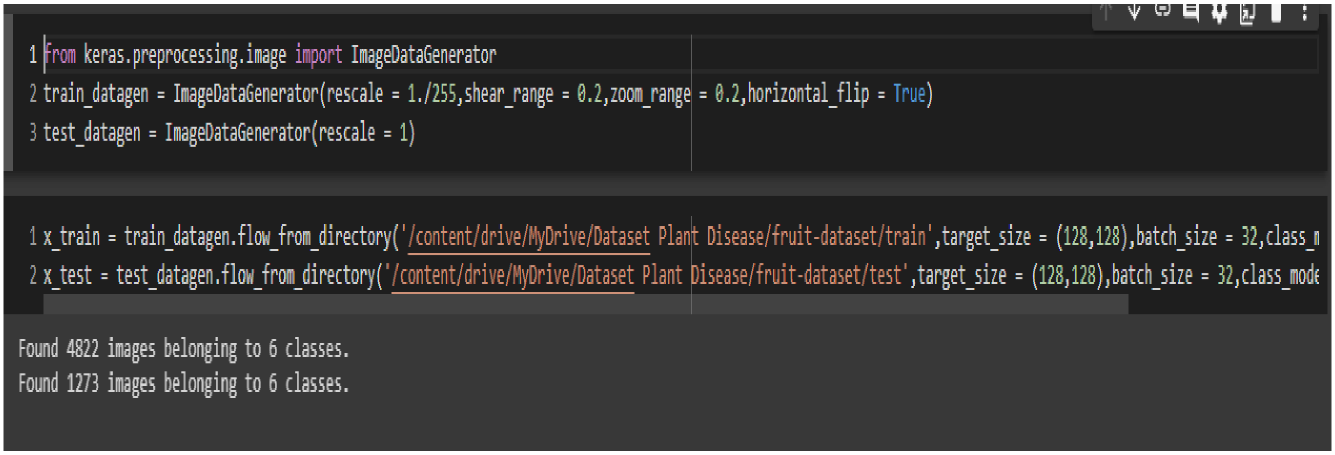
12. FUTURE SCOPE:

The main goal for the future project is to develop a complete system comprising a trained model on the server, as well as an application for mobile phones that display recognized diseases in fruits, vegetables, and other plants based on photographs taken from the phone camera. This application will aid farmers by facilitating the recognition and treatment of plant diseases in a timely manner and help them make informed decisions when utilizing chemical pesticides

13. APPENDIX

SOURCE CODE:

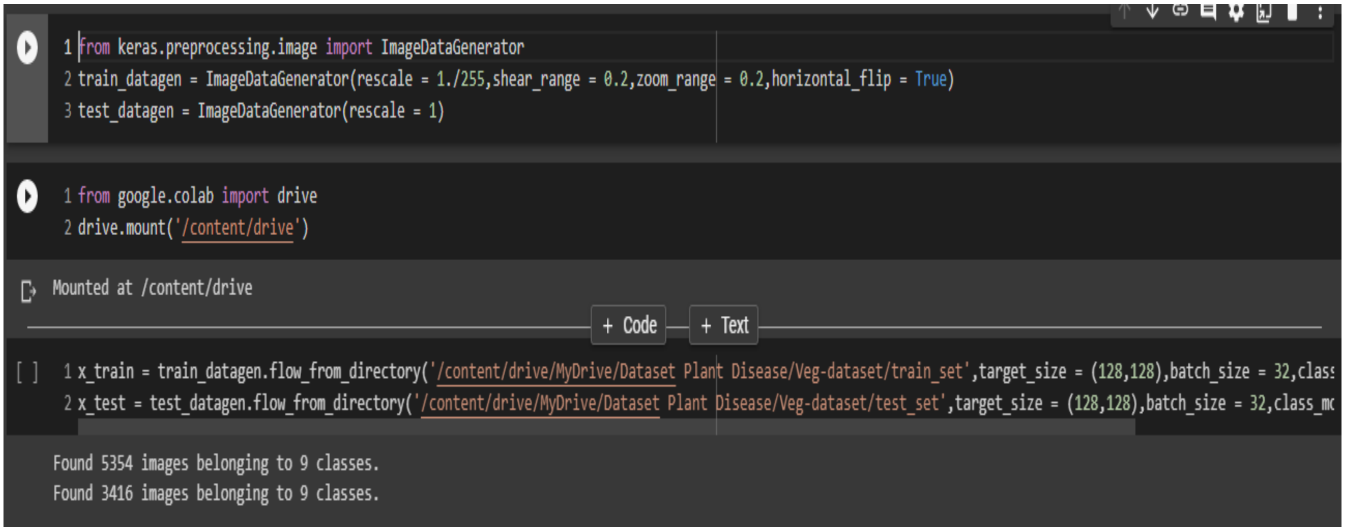
Image preprocessing:



```
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('/content/drive/MyDrive/Dataset Plant Disease/fruit-dataset/train',target_size = (128,128),batch_size = 32,class_m
x_test = test_datagen.flow_from_directory('/content/drive/MyDrive/Dataset Plant Disease/fruit-dataset/test',target_size = (128,128),batch_size = 32,class_mode
```

```
Found 4822 images belonging to 6 classes.
Found 1273 images belonging to 6 classes.
```



```
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)
```

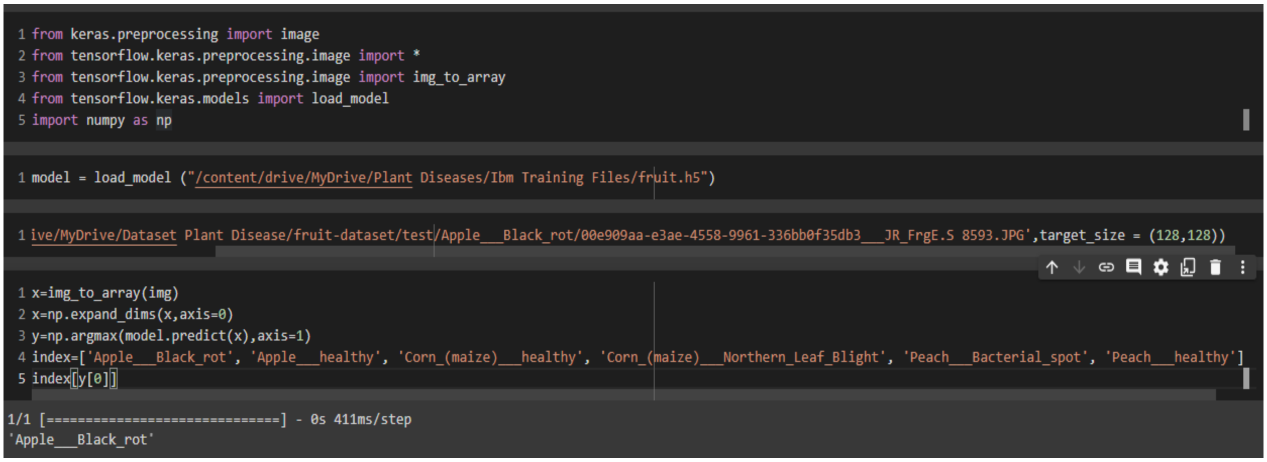
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
x_train = train_datagen.flow_from_directory('/content/drive/MyDrive/Dataset Plant Disease/Veg-dataset/train_set',target_size = (128,128),batch_size = 32,class
x_test = test_datagen.flow_from_directory('/content/drive/MyDrive/Dataset Plant Disease/Veg-dataset/test_set',target_size = (128,128),batch_size = 32,class_mc
```

```
Found 5354 images belonging to 9 classes.
Found 3416 images belonging to 9 classes.
```

Model building :



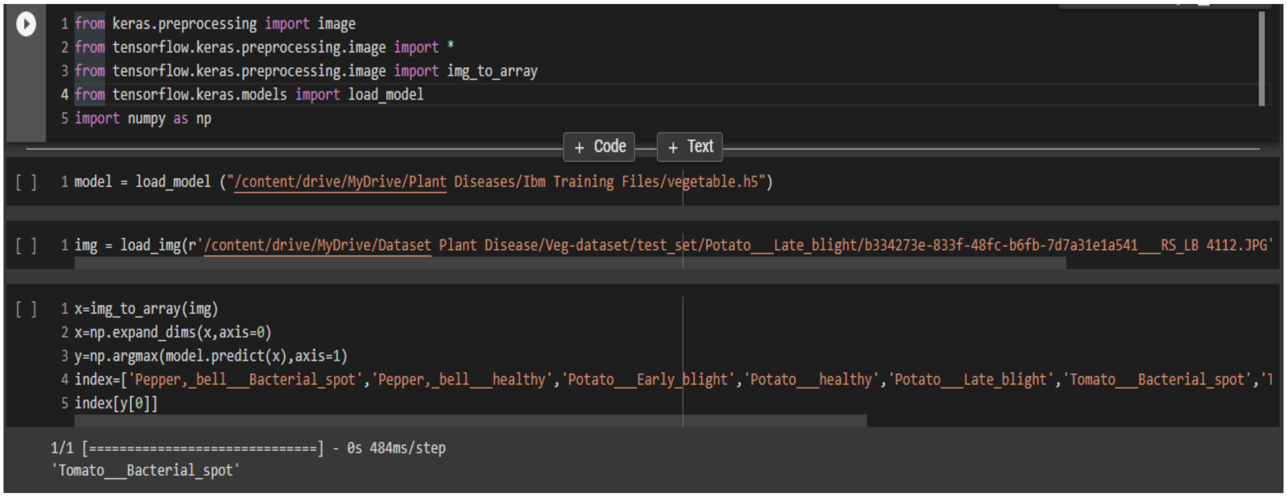
```python
from keras.preprocessing import image
from tensorflow.keras.preprocessing.image import *
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
import numpy as np
```

```python
model = load_model ("/content/drive/MyDrive/Plant Diseases/Ibm Training Files/fruit.h5")
```

```python
ive/MyDrive/Dataset Plant Disease/fruit-dataset/test/Apple___Black_rot/00e909aa-e3ae-4558-9961-336bb0f35db3___JR_FrgE.S 8593.JPG',target_size = (128,128))
```

```python
x=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]]
```

```
1/1 [==============================] - 0s 411ms/step
'Apple___Black_rot'
```



```python
from keras.preprocessing import image
from tensorflow.keras.preprocessing.image import *
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
import numpy as np
```

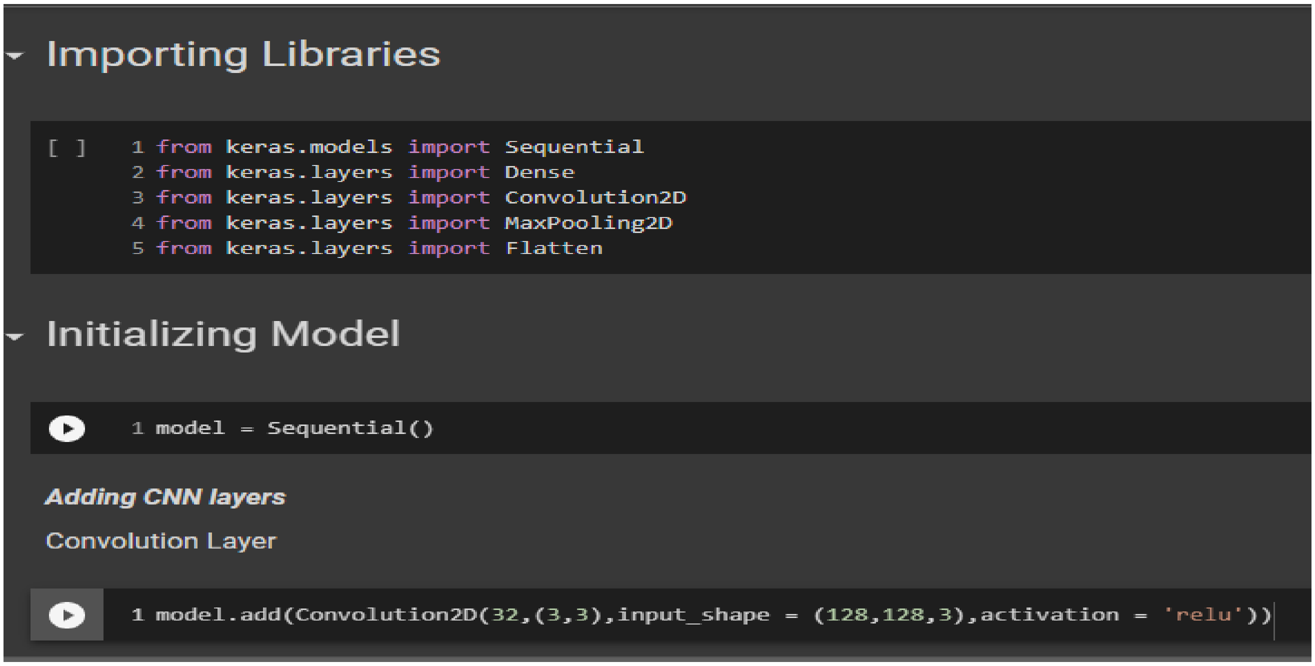
```python
model = load_model ("/content/drive/MyDrive/Plant Diseases/Ibm Training Files/vegetable.h5")
```

```python
img = load_img(r'/content/drive/MyDrive/Dataset Plant Disease/Veg-dataset/test_set/Potato___Late_blight/b334273e-833f-48fc-b6fb-7d7a31e1a541___RS_LB 4112.JPG'
```

```python
x=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','T
index[y[0]]
```

```
1/1 [==============================] - 0s 484ms/step
'Tomato___Bacterial_spot'
```

Testing:



Importing Libraries

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
```

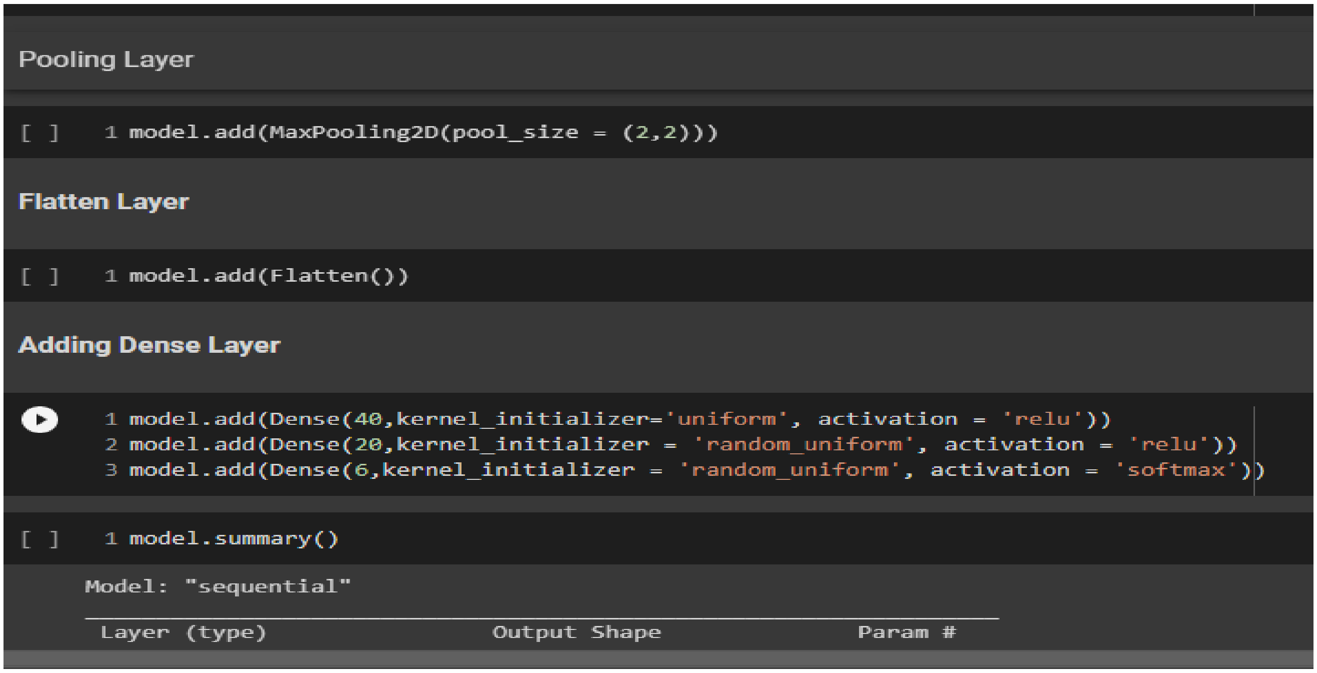
Initializing Model

```
model = Sequential()
```

Adding CNN layers

Convolution Layer

```
model.add(Convolution2D(32,(3,3),input_shape = (128,128,3),activation = 'relu'))
```



Pooling Layer

```
model.add(MaxPooling2D(pool_size = (2,2)))
```

Flatten Layer

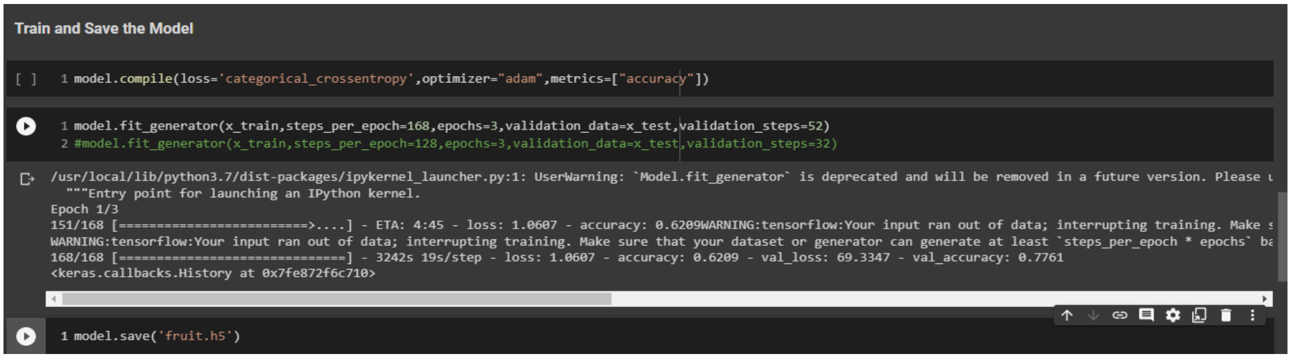
```
model.add(Flatten())
```

Adding Dense Layer

```
model.add(Dense(40,kernel_initializer='uniform', activation = 'relu'))
model.add(Dense(20,kernel_initializer = 'random_uniform', activation = 'relu'))
model.add(Dense(6,kernel_initializer = 'random_uniform', activation = 'softmax'))
```

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
```



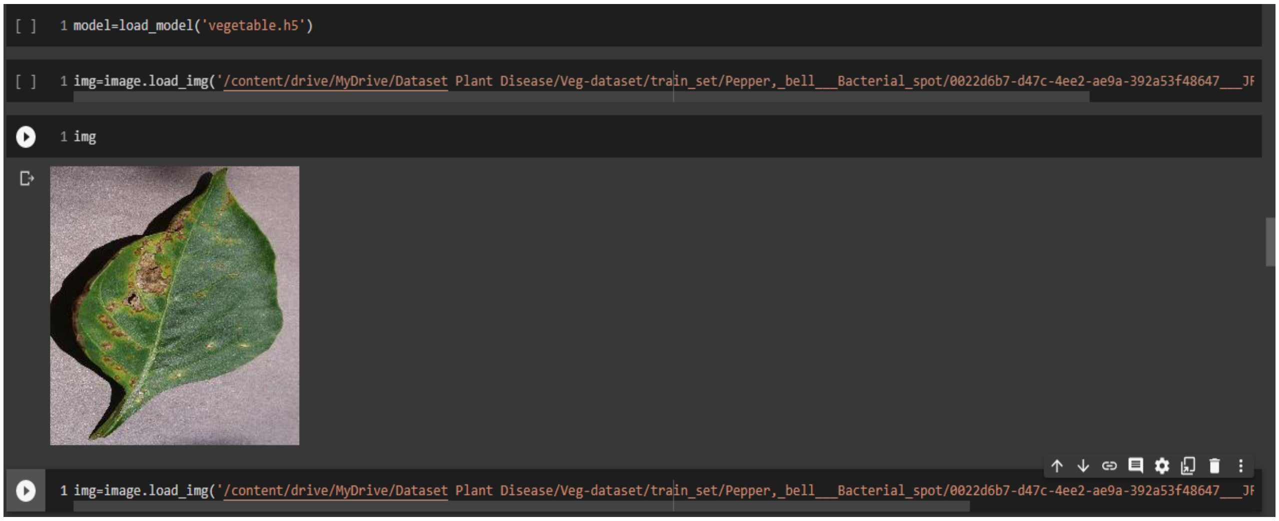
Train and Save the Model

```
[ ]  model.compile(loss='categorical_crossentropy',optimizer="adam",metrics=["accuracy"])
```

```
model.fit_generator(x_train,steps_per_epoch=168,epochs=3,validation_data=x_test,validation_steps=52)
#model.fit_generator(x_train,steps_per_epoch=128,epochs=3,validation_data=x_test,validation_steps=32)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please u
  """Entry point for launching an IPython kernel.
Epoch 1/3
151/168 [=========================>....] - ETA: 4:45 - loss: 1.0607 - accuracy: 0.6209WARNING:tensorflow:Your input ran out of data; interrupting training. Make s
WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least `steps_per_epoch * epochs` ba
168/168 [==============================] - 3242s 19s/step - loss: 1.0607 - accuracy: 0.6209 - val_loss: 69.3347 - val_accuracy: 0.7761
<keras.callbacks.History at 0x7fe872f6c710>
```

```
model.save('fruit.h5')
```



```
[ ]  model=load_model('vegetable.h5')
```

```
[ ]  img=image.load_img('/content/drive/MyDrive/Dataset Plant Disease/Veg-dataset/train_set/Pepper,_bell___Bacterial_spot/0022d6b7-d47c-4ee2-ae9a-392a53f48647___JF
```

```
img
```

```
img=image.load_img('/content/drive/MyDrive/Dataset Plant Disease/Veg-dataset/train_set/Pepper,_bell___Bacterial_spot/0022d6b7-d47c-4ee2-ae9a-392a53f48647___JF
```

Home 1.html

```
<!DOCTYPE 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 page</title><style>body{margin: 0padding:
0}.containerpadding: 30px 70px 30px 70px;left: 20px;right:20px;background-
color:rgb(163, 192, 120);font-size: 20pt;font-family: 'Times New Roman';}.card{font:
optional;display: flex;}#h1{font-size: 50pt}.menu{background-color:black;}#abc{color:
white:}</style></head><body><div
class="menu"<ul>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;
```

```
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<li id="abc"> plant Disease Prediction
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href="firstpage.html" id="abc"> home</a>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href="{{ url_for('predict') }}"
id="abc">predict</a></li></ul>

<div class="container" >

<h1 id="h1"><center><b> Detect if your plant is infected!! </b></center></h1>

<div class="card" >

<p > Agriculture is one of the major sectors works 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>

<img src="img.jpg" height="300" width="300">

</div>

</div>

</div>




</body>
```

```
</html>
```

APP.PY:

```
        f = request.files['image']
```

```
    app.run()
```

GITHUB ACCOUNT:

IBM-EPBL/**IBM-Project-31401-1660200074**