TEAN ID : PNT2022TMID32297

PROJECT NAME: Fertilizer Recommendation system for disease prediction

INDEX PAGE

TEAN ID : PNT2022TMID32297

PROJECT NAME: Fertilizer Recommendation system for disease prediction

Prediction Using Al

1 INTRODUCTION

1.1 Project Overview

Agriculture is the main aspect of country development. Food is considered as the basic need of human being which can be satisfied through farming. This project Presents an extensive survey of Artificial Intelligence ,by using the convolutional neural network and computer vision predict the plant disease and recommend the fertilizers for the plant.

1.2 Purpose

Detection and recognition of plant diseases using Al are very efficient in providing symptoms of identifying diseases at its earliest . Plant pathologists can analyze the digital images using digital image processing for diagnosis of plant diseases. Generally the plant disease are caused by the abnormal physiological functionalities of plants. Therefore the characteristic symptom are generated based on the differentiation between the normal physiological functionalities and abnormal physiological functionalities. The dataset is collected based on the plant disease. The collected dataset is trained and tested with deep learning neural network named Convolutional Neural Network(CNN). And also this project recommend the fertilizers based on the collected dataset and the disease.

2 LITERATURE SURVEY

2.1 Existing Problem

It is very important to recommend the fertilizers correctly based on the plant disease with a good accuracy. It should be easily for the farmers to communicate with the dealers or the fertilzer recommenders.

2.2 References

Aakanksha Rastogi, Ritika Arora, Shanu Sharma "advances in image processing for Leaf Disease Detection and Grading using Computer Vision Technology &Fuzzy Logic" International conference on signal processing and integrated network SPIN, pp 500-505.

Ms.pooja pawar, Dr.varsha turkar, Prof.pravin patil presents "algorithm for detecting crop disease early and exactly, this system is developed using image processing techniques and artificial neural network".

H.G. Wang, G. L. Li, Z. H. Ma, and X. L. Li. "Application of neural networks to image recognition of plant diseases", International Conference on Systems and Informatics, 2012.

Jayamala K. Patill and Raj Kumar, "Advances in image processing for detection of plant diseases", Journal of Advanced Bioinformatics Applications and Research, ISSN 09762604v01 2, Issue 2, pp 135-141,June-2011.

Shiva reddy proposed an 10T based system for leaf disease detection and fertilizer recommendation which is based on Machine Learning techniques yields less 80 percentage accuracies.

2.3 Problem Statement Definition

Agriculture is the most important Sector in today's life. Most of the plants are affected by a wide variety of bacterial and fungal diseases. In agricultural aspects , if the plant is affected by leaf disease then it reduces the growth and productiveness. Generally the plant diseases are caused by the abnormal physiological functionalities of plants.

3 IDEATION & PROPOSED SOLUTION

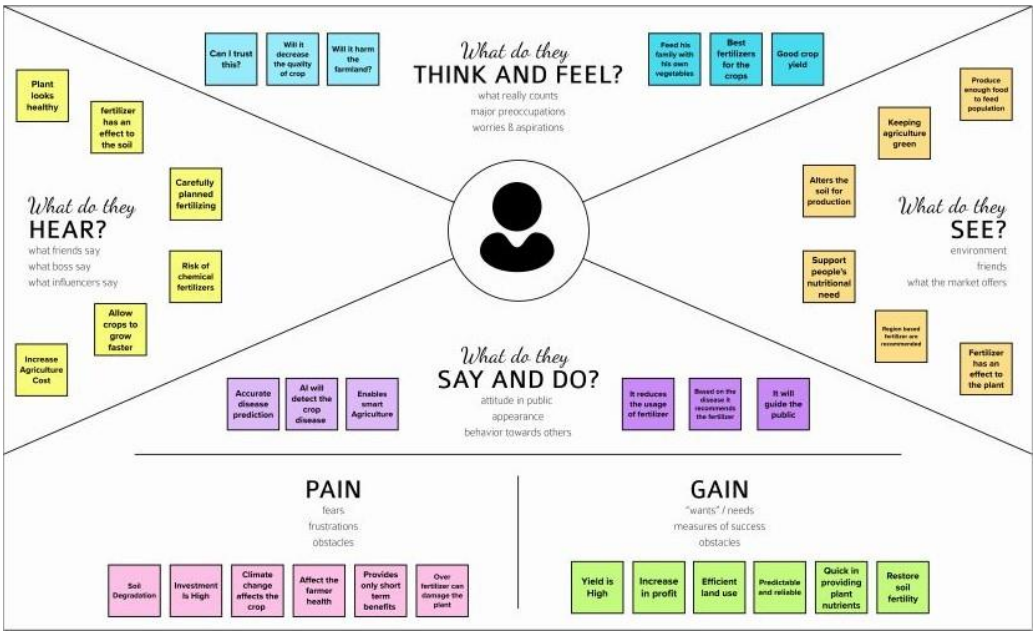
3.1 Empathy Map Canvas

TEAN ID : PNT2022TMID32297

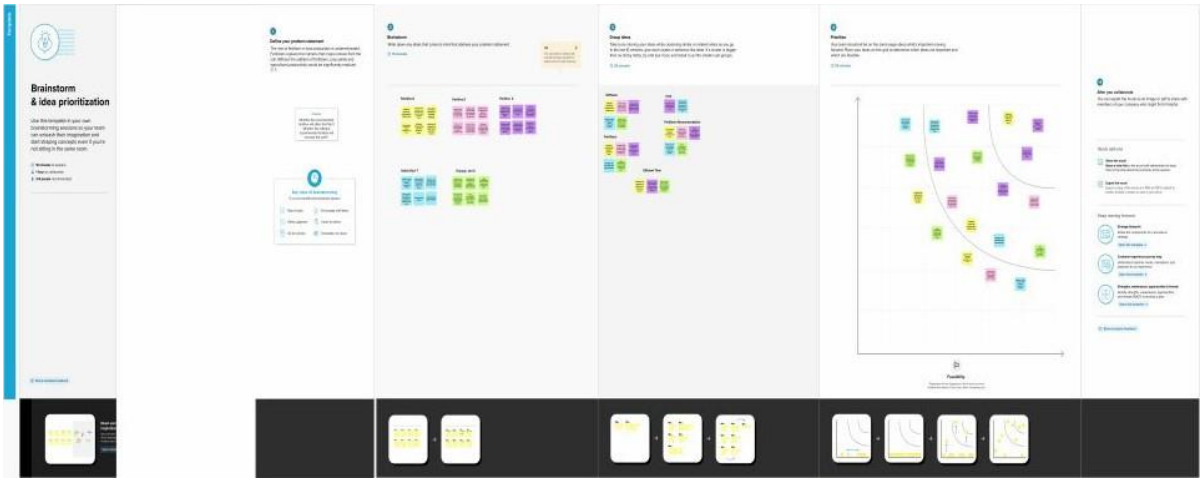
PROJECT NAME: Fertilizer Recommendation system for disease prediction

Empathy Map Canvas

Fertilizers Recommendation System For Disease Prediction



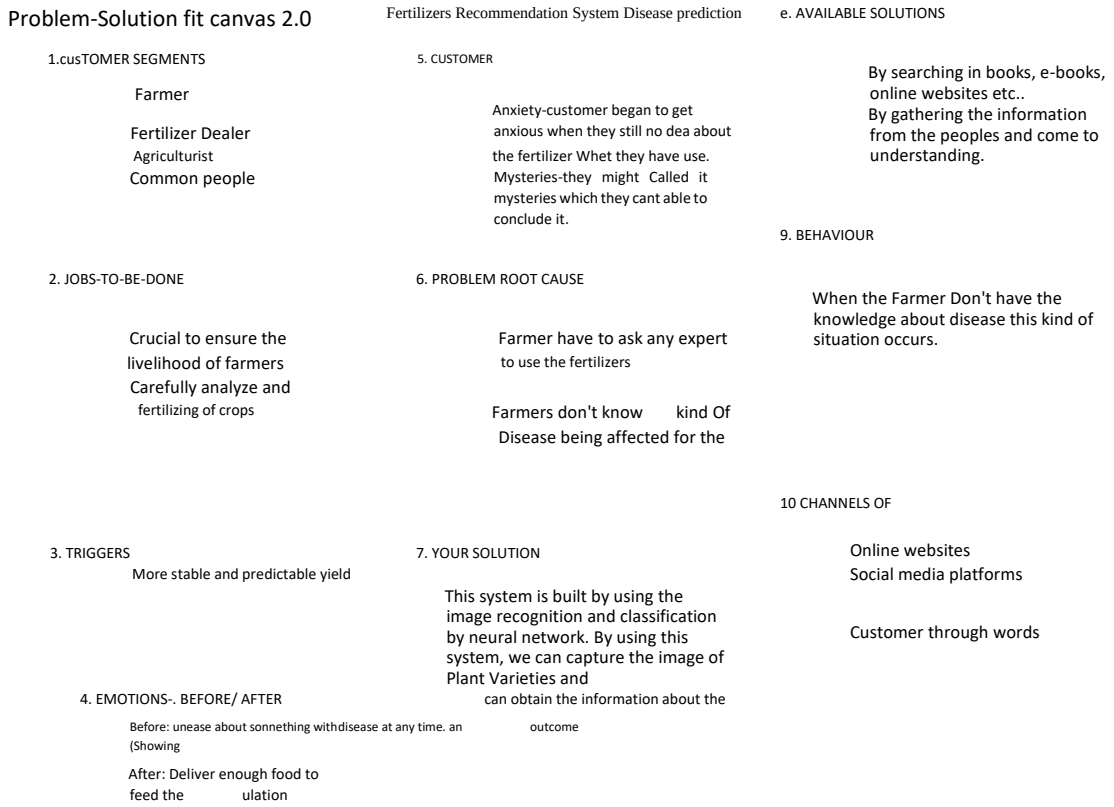
3.2 Ideation & Brainstorming



3.3 Proposed Solution

The proposed solution is to develop a prediction model using IBM Al service. Under Al, there are various deep learning techniques are available. Using Convolutional Neural Network(CNN), different diseases are identified and it recommend the fertilizers based on the predicted disease. An application is also built in an easy way for the farmers to communicate with the dealers or fertilizer retailers.

3.4 Problem Solution Fit



TEAN ID : PNT2022TMID32297

PROJECT NAME: Fertilizer Recommendation system for disease prediction

4 REQUIREMENT ANALYSIS

4.1 Functional Requirements

User Registration: Registration through form,
Registration through Gmail,
Registration through Linkedln.

User Conformation: Conformation via Email,
Conformation via OTP.

User Profile: Log in ,
Access the profile

Image Processing: Capture the image of the plant disease
Analyze the plant disease that is send by user

Prediction: Compare the image with the trained data in the model and predict the model

Recommend: Based on the predicted disease the software recommend fertilizers.

4.2 Non-Functional Requirements

Usability: Fertilizers recommendation are created and saved then these recommended fertilizers are used by the farmers.

Security: The software keeps the users information more securely.

Reliability: Creating the interactive dashboard which is easy to understand and useful for the users.

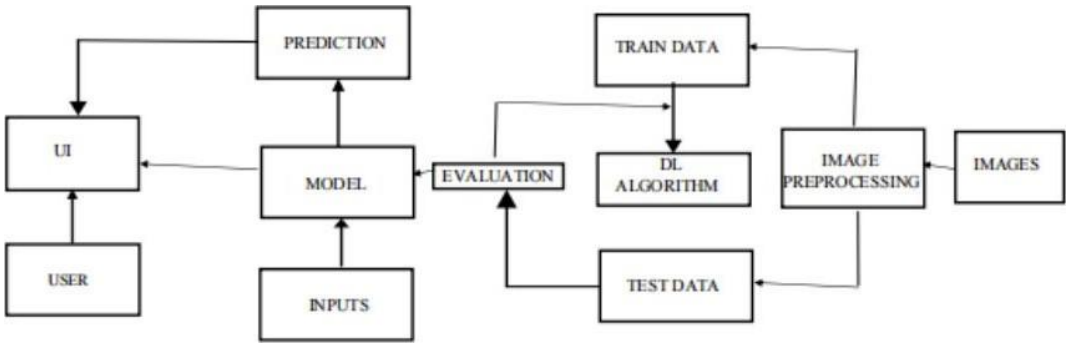
Availability: Software application is available for every user and they can easily access them.

Scalability: The proposed precision agriculture structure allows the implementation of a flexible methodology that can be adopted to different types of crops.

5 PROJECT DESIGN

5.1 Data Flow Diagram

A data flow diagram (DFD) is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement.



5.2 Solution & Technical Architecture

Solution Architecture

A solution architecture (SA) is an architectural description of specific solution. SAS combine guidance from different enterprise architecture viewpoints (business, information and technical), as well as from the enterprise solution architecture (ESA).

Technical Architecture

Technology architecture deals with the deployment of application components on technology components. A standard set of predefined technology components is provided in order to represent servers, network, workstations, and so on

TEAN ID : PNT2022TMID32297

PROJECT NAME: Fertilizer Recommendation system for disease prediction

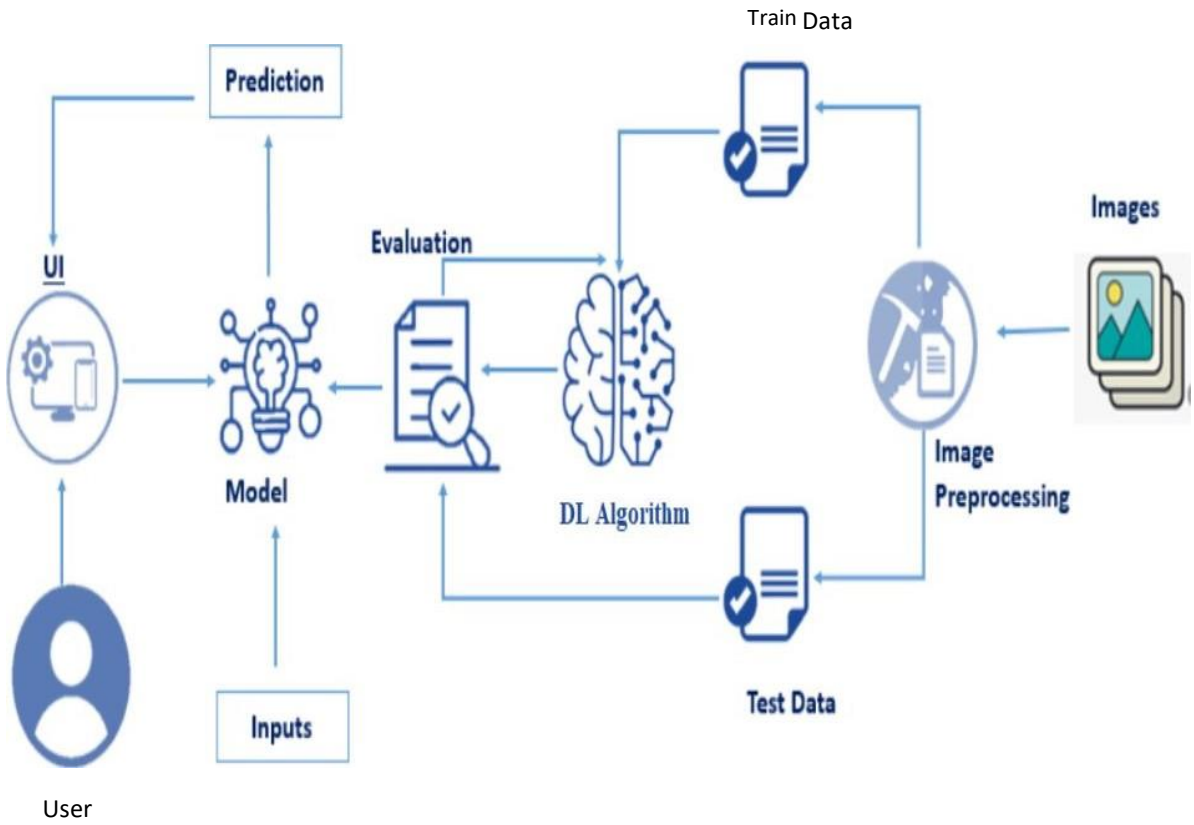


Image send by an user to the software and it is processed by image preprocessing. The image is compared with the trained and tested data by using the deep learning algorithm. The model analysis the given image and predict the output.

5.3 User Stories

ser 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- I

		-USN 2	As a user, I will recelve confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-I
		-USN 3	Asa user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		-USN 4	As a user, I can register for the application through Gmail	I can able to go through the cart	Medium	Sprint -l
	Login	-USN 5	As a user, I can log into the application by entering email & password	I can reset my password If I have forgotten my password	High	Sprint -l
	My account	-USN 6	As a user,l can view my personal information	I can edit my profile photo and email.i can logout of the application from my account	High	Sprint-2

TEAN ID : PNT2022TMID32297

PROJECT NAME: Fertilizer Recommendation system for disease prediction

ser type	Functional Requirement(Epic)	User Story Number	User Story/ Task	Acceptancecriteria	Priority	Release
Customer(Web user)	Registration	-USN 7	As a user ,I can legister for the application by entering my email ,password and confirming my password	I can upload a rofile photo and add my name to the account	Medium	Sprint -1
Customer Care Executive	Communication	-USN 8	As a user, i can maintain provide support systems forrelationships thatwith customer client I communicatecan with thequeries customersmcrease productivity	I can strong companies oftenand ease their and	High	Sprint- 2
Administrator	Chief Executive	-USN 9	As an administrator ,I can modify the list of products -so I can adjust our offerings overtime	Add or remove products.modify product images.select a category for the products.modify category taxonomy	High	Sprint -1

6 PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement	UserStory Number	User Story I Task	Story Points otal	Priority	Team Members
Sprint-I	Download the Dataset		Download the dataset for fillther Processing	8	High	Pavithm.S Küila T Pavithra K
	Image Preprocessing		Process and Fonnat the images before they used by model training and inference	2	High	Pavithra.S Kokila T Nivedha G

Sprint	Functional Requirement (Epic)	User Story Number	user Story / Task	Story Points (Total)	Priority	Team Members
Sprint-2	Model Creatiml and Training (Fruits)		Create a model which can classify' diseased vegetable plants from given images and train 0111BM Cloud		High	Pavithra S Kokila Devi T Nivedha G Pavithra K Thanuja Shri K

	Model Creation and Training (Vegetables)		Create a model which can classify diseased vegetable plants from given images	2	High	Pavithra S Kokila Devi T Nivedha G Pavithra K Thanuja Shri K
Sprint-3	Registration	USN-I	As a user, I can register by entering my email, password, and confinning my password via OAuth API		Medium	Pavithra S Kokila Devi T Nivecnia G
	Upload page	USN-2	As a user, I will be redirected to a page where I can upload my pictures of crops	4	High	Nivedha G Pavithra K Thanuja Shri K
	Suggestion results	USN-3	As a usa, I can view the results and thau obtain the suggestions provided by the NIL model	4	High	Pavithra S Kokila Devi T Nivedha G
	Base Flask App		A base Flask web app must be created as an inta%ce for ML model	2	High	kokila nevi T Nivedha G Pavithra K
	Login	USN-I	As a user/admin/shopkeeper, I can log into the application by entering email & password	2	High	Nivedha G Pavithra K Thanuja Shri K
	User Dashboard	USN-5	As a usa, I can View the previous results and		Medium	Pavithra S Kokila nevi T Nivedha G
Sprint4	Integration		IntegateF1ask, CNN model with Cloudant DB		Medium	Nivedha G Pavithra K Thanuja Shri K
	Dashboard	USN-6	As an admin, I can view Other user details and uploads for othe- purm»ses		Medium	Kokila nevi T Nivedha G Pavithra K
	Dashboard (Sho	USN-7	As a shopkeeper, I can enter fertilizer productsand then update the details if any			Pavithra S Kokila Devi T Nivedha G

6.2 Sprint Delivery Schedule

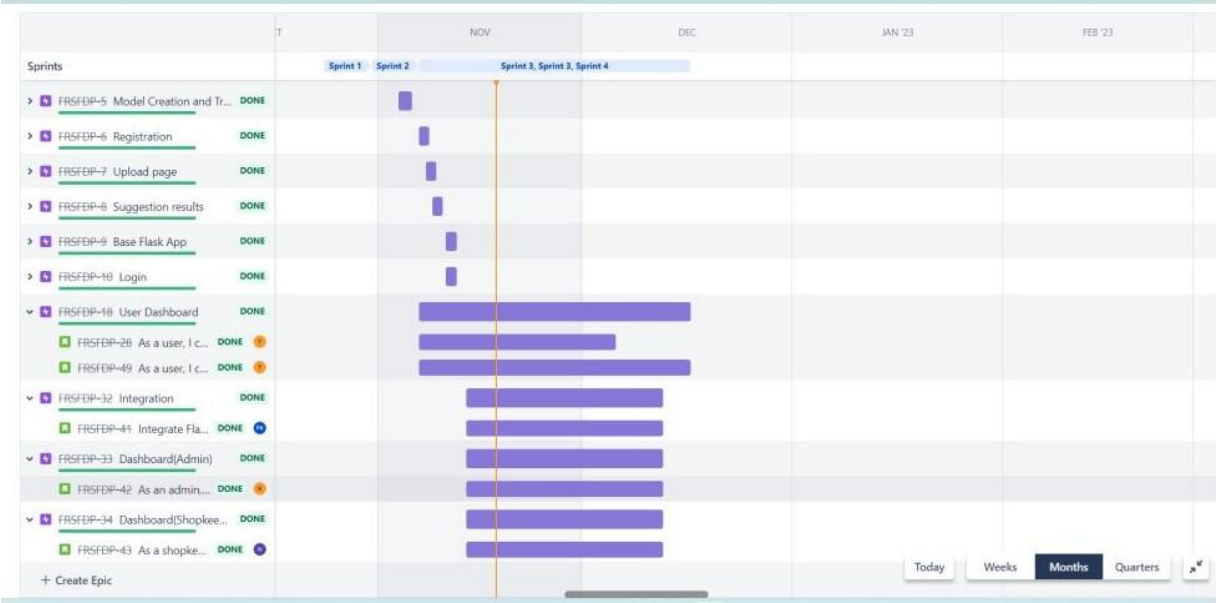
Sprint	Total Story Points	Du ration	Sprint Start Date	Sprint End Date (Planned)	Story points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-I	10	6 Days	24 Oct 2022	29 Oct 2022		29 Oct 2022
Sprint-2	10	6 Days	31 Oct2022	05 Nov 2022	10	05 Nov 2022
	18	6 Days	07 Nov 2022	12 Nov 2022	18	12 Nov 2022
Sprint—4	12	6 Days	14 Nov 2022	19 Nov 2022	12	19 Nov 2022

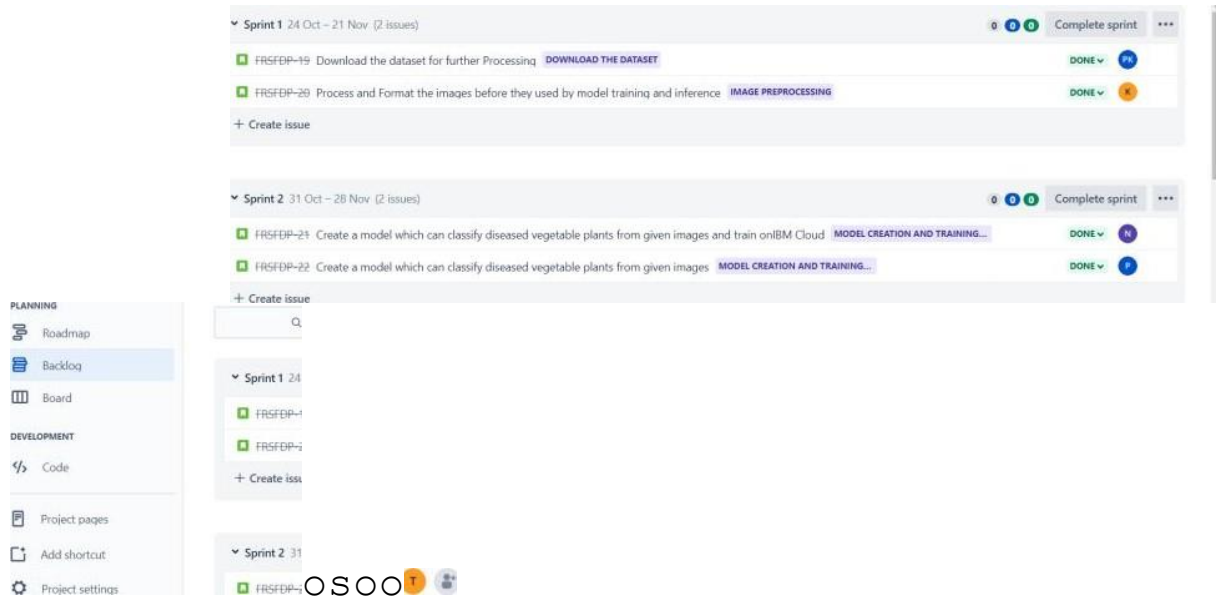
6.3 Reports

Creation of Sprint 1,2,3,4:

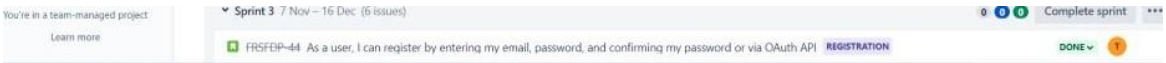
TEAN ID : PNT2022TMID32297

PROJECT NAME: Fertilizer Recommendation system for disease prediction





.00



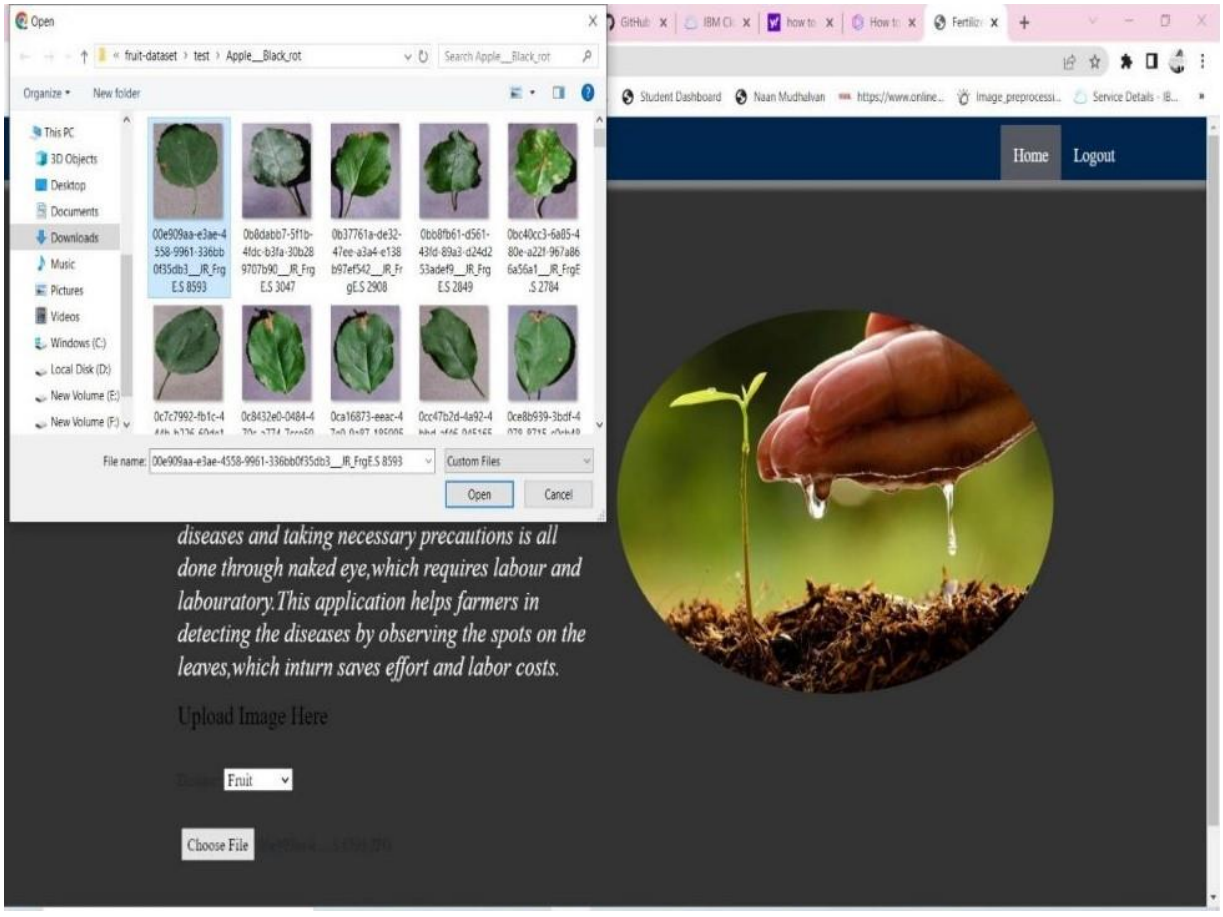
7 CODING & SOLUTIONING

7.1 Feature 1

- New user can able to register in our website.
- Verified user can able to login our website.
- Verified user cannot be able to login with invalid credentials.

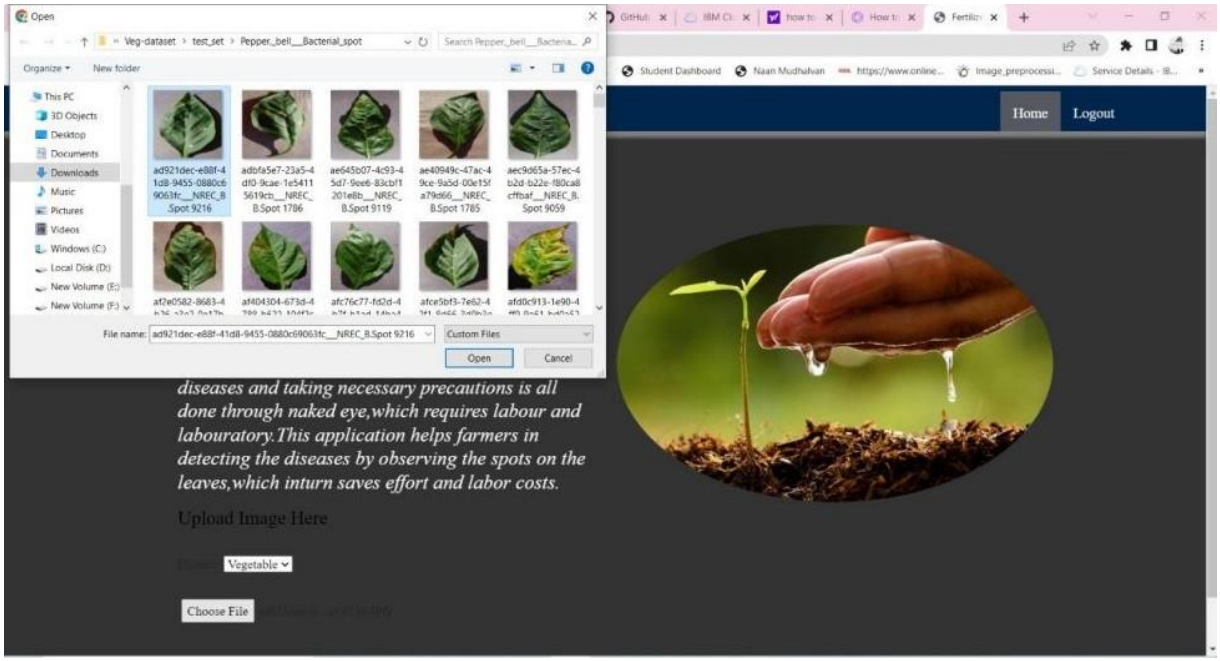
7.2 Feature 2

- User can view the index page of the Fertilizers Recommendation System for Disease Prediction using Artificial Intelligence.
- User can choose the image file (Fruit/Veg) in the index page.



TEAN ID : PNT2022TMID32297

PROJECT NAME: Fertilizer Recommendation system for disease prediction



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
LoginPage_TC_001	Functional	Login Page	Verify user is able to see the Login/Signup popup when user clicked on My	Running on http://127.0.0.1:5000	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	file:///C:/Users/LENOVO/Downloads/login.html	User should navigate to user account homepage	Working as expected	Pass	Steps are clear and simple	N	Nil	Kokila Devi T
LoginPage_TC_002	UI	Login Page	Verify the UI elements in Login/Signup popup	Running on http://127.0.0.1:5000	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button	Username: chalam@gmail password: Testing123	Application should show below UI elements: a.email text box b.password text box c.Login button with white colour	Working as expected	Pass	Good Look and feel to the user	N	Nil	Kokila Devi T
LoginPage_TC_003	UI	Register page	For New User Registration	Running on http://127.0.0.1:5000 connect with flask app	1.Enter URL and click go 2.Enter User Valid username 3.Enter Valid email in Email text box 4.Enter valid password in password text box 5.Click on Register button user already exists click on login	file:///C:/Users/LENOVO/Downloads/register.html	User should navigate to user account homepage	Working as expected	Pass	Easy and simple user - Friendly UI.	N	Nil	Pavithra S
LoginPage_TC_004	Functional	Home page	User can Select the Plant Disease(Fruit/Veg) image	Running on http://127.0.0.1:5000 connect with flask app	1.After Navigation into Home Page 2.Click on the Dropdown button to select(Fruit/Veg)	Select from the dropdown menu	User can select any one among the option	Working as expected	Pass	Clear information was given.	N	Nil	Pavithra K
LoginPage_TC_005	Functional	Home page	User can choose the image from the Dataset	Running on http://127.0.0.1:5000 connect with flask app	1.click on the choose file 2.import the image from the image files 3.click on open	choose image from the file	User can choose suitable image file	Working as expected	Pass	Clear information was given.	N	Nil	Nivedha G
LoginPage_TC_006	Functional	Home page	User can get the result from the recommendation system	Running on http://127.0.0.1:5000 connect with flask app and import H5 file	1.Click on the Predict button 2.Navigate to Predict Page 3.Result will be displayed on the Interface	Plant Disease will get Predicted.	User can get the result from the recommendation system	Working as expected	Pass	Clear information was given.	N	Nil	Thanuja Shri K

8.2 User Acceptance Testing

2. 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	Subtot
By Design	10	4	2	3	19
Duplicate	2	0	2	0	4
External	2	0	0	0	2
Fixed	10	3	2	1	16
Not Reproduced	0	1	0	0	1
Skipped	0	0	1	1	2
Won't Fix	0	0	0	0	0
Totals	24	8	7	5	44

Subtotal

3. Test Case Analysis

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

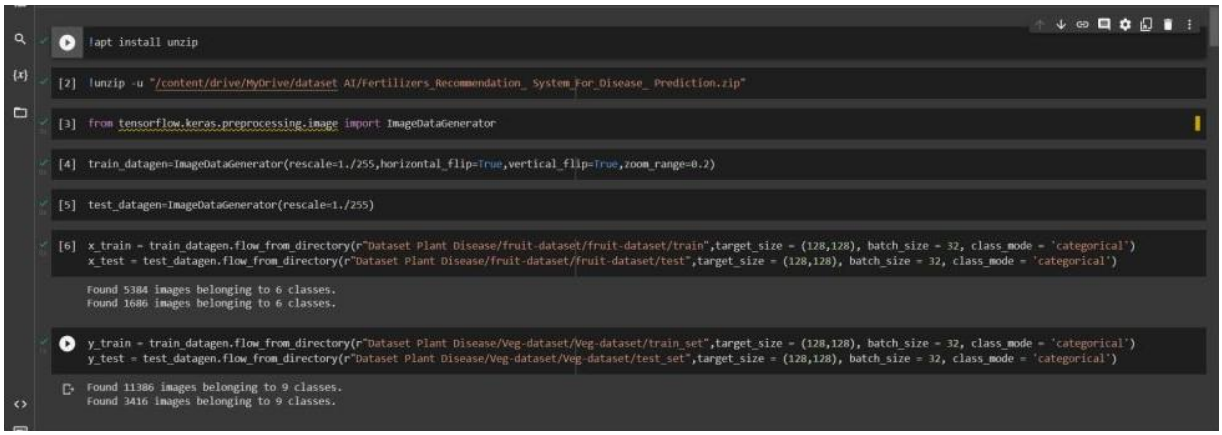
Section	Total Cases Not Tested	Fail	Pass			
Print Engine			7	0	0	7
Client Application			43	0	0	43
Security			5	0	0	5
Outsource Shipping			4	0	0	4
Exception Reporting			10			10
Final Report Output						
Version Control						

TEAN ID : PNT2022TMID32297

PROJECT NAME: Fertilizer Recommendation system for disease prediction

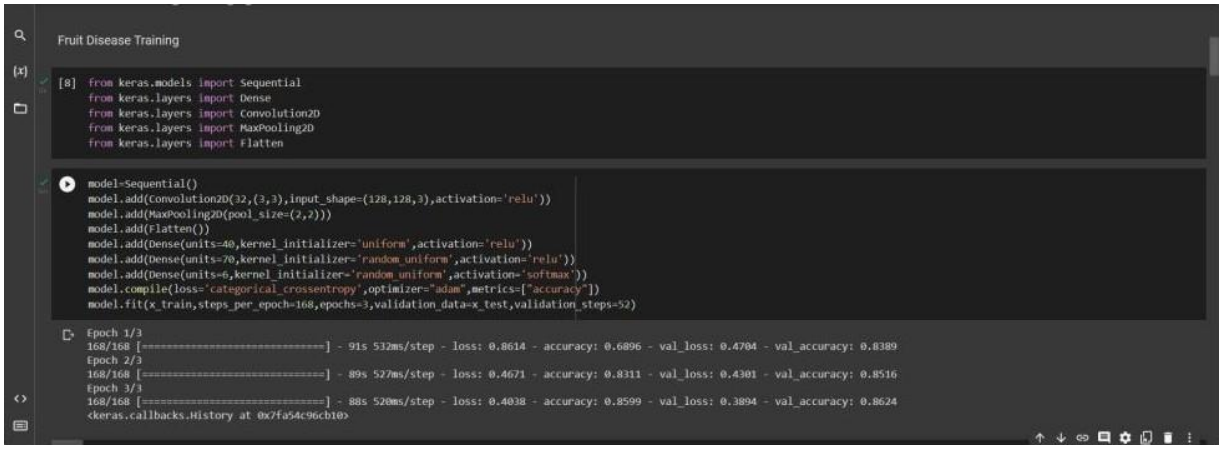
9 RESULTS

9.1 Performance Metrics



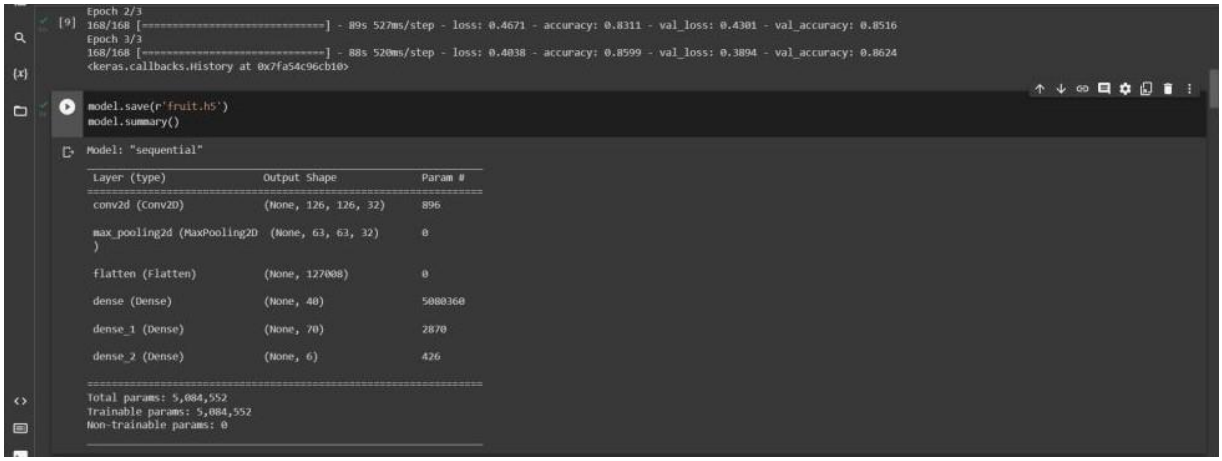
```
!apt install unzip

[2] !unzip -u "/content/drive/MyDrive/dataset AI/Fertilizers_Recommendation_ System_For_Disease_ Prediction.zip"

[3] from tensorflow.keras.preprocessing.image import ImageDataGenerator

[4] train_datagen=ImageDataGenerator(rescale=1./255,horizontal_flip=True,vertical_flip=True,zoom_range=0.2)

[5] test_datagen=ImageDataGenerator(rescale=1./255)

[6] x_train = train_datagen.flow_from_directory(r"Dataset Plant Disease/fruit-dataset/fruit-dataset/train",target_size = (128,128), batch_size = 32, class_mode = 'categorical')
    x_test = test_datagen.flow_from_directory(r"Dataset Plant Disease/fruit-dataset/fruit-dataset/test",target_size = (128,128), batch_size = 32, class_mode = 'categorical')

    Found 5384 images belonging to 6 classes.
    Found 1686 images belonging to 6 classes.

    y_train = train_datagen.flow_from_directory(r"Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set",target_size = (128,128), batch_size = 32, class_mode = 'categorical')
    y_test = test_datagen.flow_from_directory(r"Dataset Plant Disease/Veg-dataset/Veg-dataset/test_set",target_size = (128,128), batch_size = 32, class_mode = 'categorical')

    Found 11386 images belonging to 9 classes.
    Found 3416 images belonging to 9 classes.
```

Fruit Accuracy:



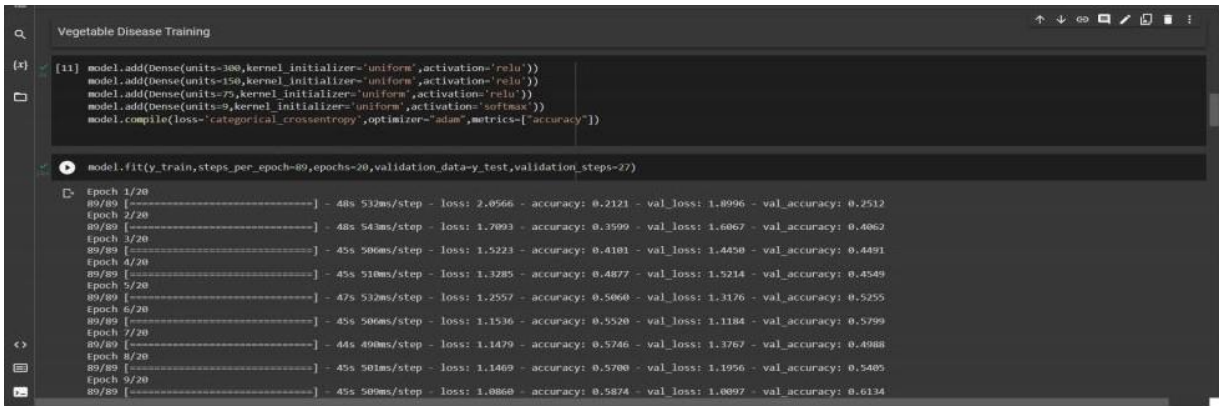
```
Fruit Disease Training

[8] 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

    model=Sequential()
    model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Flatten())
    model.add(Dense(units=40,kernel_initializer='uniform',activation='relu'))
    model.add(Dense(units=70,kernel_initializer='random_uniform',activation='relu'))
    model.add(Dense(units=6,kernel_initializer='random_uniform',activation='softmax'))
    model.compile(loss='categorical_crossentropy',optimizer="adam",metrics=["accuracy"])
    model.fit(x_train,steps_per_epoch=168,epochs=3,validation_data=x_test,validation_steps=52)

    Epoch 1/3
    168/168 [==============================] - 91s 532ms/step - loss: 0.8614 - accuracy: 0.6896 - val_loss: 0.4704 - val_accuracy: 0.8389
    Epoch 2/3
    168/168 [==============================] - 89s 527ms/step - loss: 0.4671 - accuracy: 0.8311 - val_loss: 0.4301 - val_accuracy: 0.8516
    Epoch 3/3
    168/168 [==============================] - 88s 520ms/step - loss: 0.4038 - accuracy: 0.8599 - val_loss: 0.3894 - val_accuracy: 0.8624
    <keras.callbacks.History at 0x7fa54c96cb10>
```

Fruit Summary:



```
Epoch 2/3
[9] 168/168 [==============================] - 89s 527ms/step - loss: 0.4671 - accuracy: 0.8311 - val_loss: 0.4301 - val_accuracy: 0.8516
Epoch 3/3
168/168 [==============================] - 88s 520ms/step - loss: 0.4038 - accuracy: 0.8599 - val_loss: 0.3894 - val_accuracy: 0.8624
<keras.callbacks.History at 0x7fa54c96cb10>

model.save(r'fruit.h5')
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, 70)                2870

 dense_2 (Dense)             (None, 6)                 426

=================================================================
Total params: 5,084,552
Trainable params: 5,084,552
Non-trainable params: 0
_________________________________________________________________
```

Vegetable Accuracy:



```
Vegetable Disease Training

[11] model.add(Dense(units=300,kernel_initializer='uniform',activation='relu'))
     model.add(Dense(units=150,kernel_initializer='uniform',activation='relu'))
     model.add(Dense(units=75,kernel_initializer='uniform',activation='relu'))
     model.add(Dense(units=9,kernel_initializer='uniform',activation='softmax'))
     model.compile(loss='categorical_crossentropy',optimizer="adam",metrics=["accuracy"])

     model.fit(y_train,steps_per_epoch=89,epochs=20,validation_data=y_test,validation_steps=27)

Epoch 1/20
89/89 [==============================] - 48s 532ms/step - loss: 2.0566 - accuracy: 0.2121 - val_loss: 1.8996 - val_accuracy: 0.2512
Epoch 2/20
89/89 [==============================] - 48s 543ms/step - loss: 1.7093 - accuracy: 0.3599 - val_loss: 1.6067 - val_accuracy: 0.4062
Epoch 3/20
89/89 [==============================] - 45s 506ms/step - loss: 1.5223 - accuracy: 0.4101 - val_loss: 1.4458 - val_accuracy: 0.4491
Epoch 4/20
89/89 [==============================] - 45s 510ms/step - loss: 1.3285 - accuracy: 0.4877 - val_loss: 1.5214 - val_accuracy: 0.4549
Epoch 5/20
89/89 [==============================] - 47s 532ms/step - loss: 1.2557 - accuracy: 0.5060 - val_loss: 1.3176 - val_accuracy: 0.5255
Epoch 6/20
89/89 [==============================] - 45s 506ms/step - loss: 1.1536 - accuracy: 0.5520 - val_loss: 1.1184 - val_accuracy: 0.5799
Epoch 7/20
89/89 [==============================] - 44s 490ms/step - loss: 1.1479 - accuracy: 0.5746 - val_loss: 1.3767 - val_accuracy: 0.4988
Epoch 8/20
89/89 [==============================] - 45s 501ms/step - loss: 1.1469 - accuracy: 0.5700 - val_loss: 1.1956 - val_accuracy: 0.5405
Epoch 9/20
89/89 [==============================] - 45s 509ms/step - loss: 1.0860 - accuracy: 0.5874 - val_loss: 1.0097 - val_accuracy: 0.6134
```

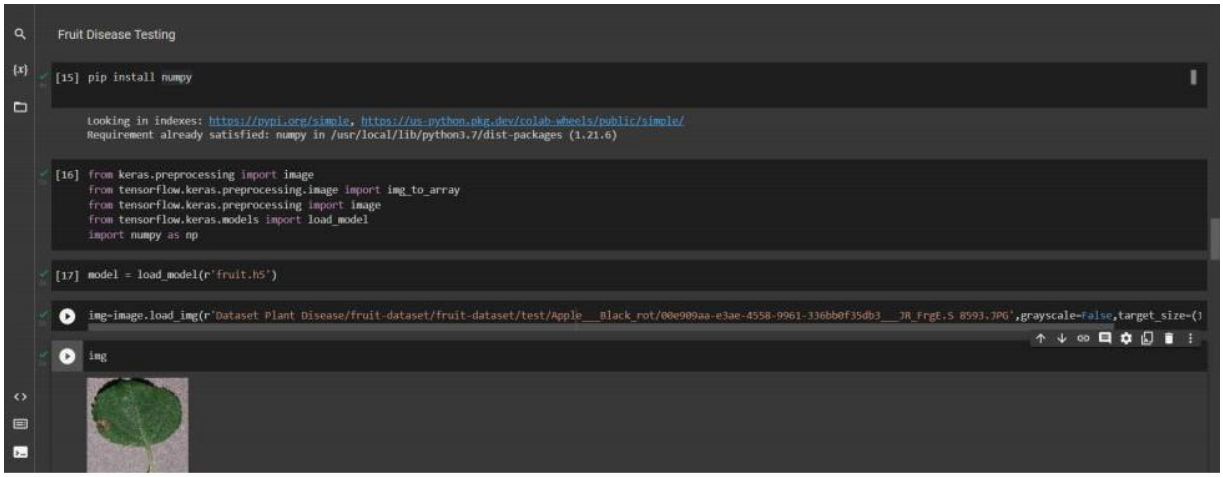
Vegetable Summary:

```
model.save(r'vegetable.h5')
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, 70)                2870

 dense_2 (Dense)             (None, 6)                 426

 dense_3 (Dense)             (None, 300)               2100

 dense_4 (Dense)             (None, 150)               45150

 dense_5 (Dense)             (None, 75)                11325

 dense_6 (Dense)             (None, 9)                 684

=================================================================
Total params: 5,143,811
Trainable params: 5,143,811
Non-trainable params: 0
_________________________________________________________________
```

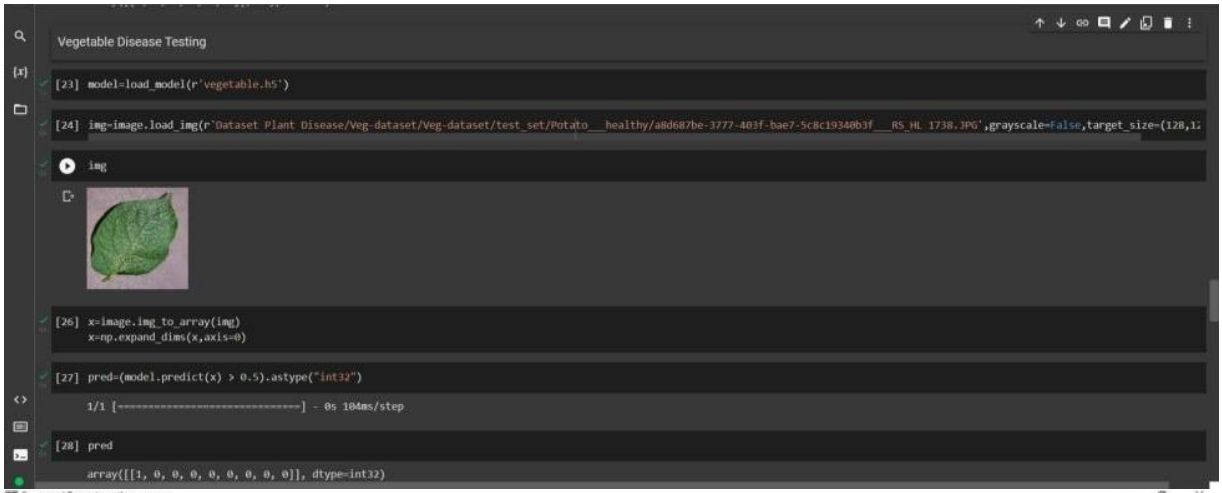
TEAN ID : PNT2022TMID32297

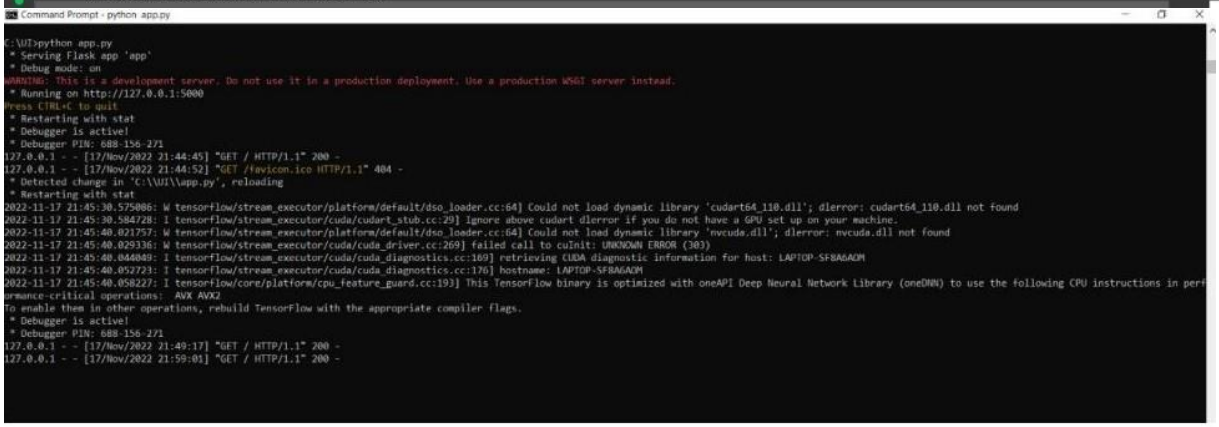
PROJECT NAME: Fertilizer Recommendation system for disease prediction

Fruit Score:



Vegetable Score:

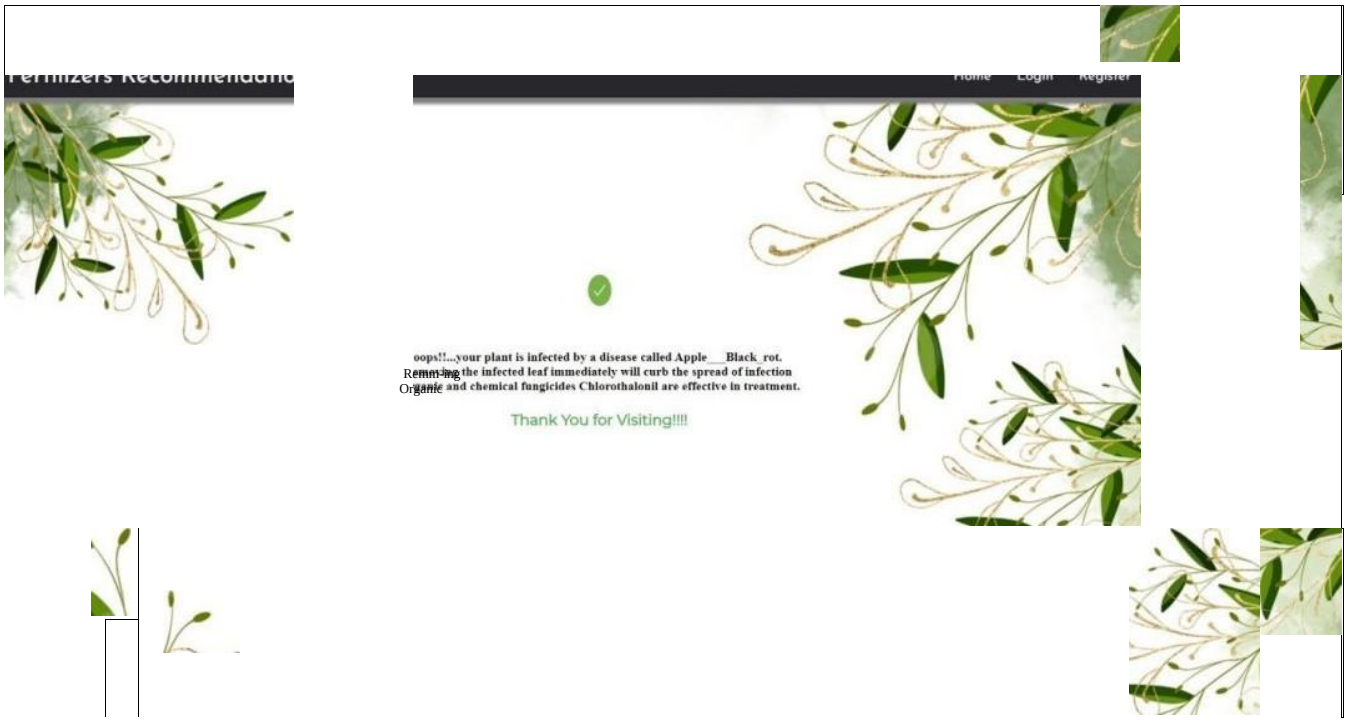




Login Register

TEAN ID : PNT2022TMID32297

PROJECT NAME: Fertilizer Recommendation system for disease prediction



10 ADVANTAGES & DISADVANTAGES 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.

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.

11 CONCLUSION

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

- The accuracy of the classification also increased by varying dense layers.
- For different batch sizes, different classification accuracies are obtained.
- The accuracy of classification increased by increasing the number of epoch. o 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 aplication software using python to exe software. The real time image classification,image recognition and video processing are possible with the help of OpenCV python library. This project work can be extended for security aplications such as finger print recognition , iris recognition and face recognition.

13 APPENDIX

Source Code

```
!apt install unzip
!unzip -u "/content/drive/MyDrive/dataset Al/Fertilizers_Recommendation_ System_For_Disease_ Prediction.zip"
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

TEAN ID : PNT2022TMID32297

PROJECT NAME: Fertilizer Recommendation system for disease prediction

train_datagen=ImageDataGenerator(rescale=
m_range=0.2) 1./255,horizontal_flip —True,
vertical_flip=True,zoo

test_datagen=ImageDataGenerator(rescale=1./255) x_train = train_datagen.flow_from_directory(r"Dataset Plant Disease/fruit-dataset/fruitdataset/train",target_size = (128,128), batch_size = 32, class_mode = 'categorical') x_test = test_datagen.flow_from_directory(r"Dataset Plant Disease/fruit-dataset/fruitdataset/test",target_size = (128,128), batch_size = 32, class_mode = 'categorical') y_train = train_datagen.flow_from_directory(r"Dataset Plant Disease/Veg-dataset/Vegdataset/train_set",target_size = (128,128), batch_size = 32, class_mode = 'categorical') y_test = test_datagen.flow_from_directory(r"Dataset Plant Disease/Veg-dataset/Vegdataset/test_set",target_size = (128,128), batch_size 32, class_mode = 'categorical')

Fruit Disease Training

```
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

model—Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2))) model.add(Flatten())
model.add(Dense(units=40,kernel_initializer='uniform',activation='relu'))
model.add(Dense(units=70,kernel_initializer='random_uniform',activation='relu')) model. ,
activation='softmax')) model. adam" , accuracy" l)
dd(Dense(units=6,kernel_initializer='random_uniform'
ompile(loss='categorical_crossentropy',optimizer="ada
model.fit(x_train,steps_per_epoch=168,epochs=3,validation_data=x_test,validation_steps=
52) model.save(r'fruit.h5') model.summary()
```

Vegetable Disease Training

```
model.add(Dense(units=300,kernel_initializer='uniform',activation='relu'))
model.add(Dense(units=150,kernel_initializer='uniform',activation='relu'))
```

dd(Dense(units=9,kernel_initializer='uniform',acti
ompile(loss='categorical_crossentropy',optimizer=

```
model.add(Dense(units=75,kernel_initializer='uniform',activation='relu')) model.
,activation='softmax')) model. "adam", accuracy"])
model.fit(y_train,steps_per_epoch=89,epochs=20,validation_data=y_test,validation_steps=27
)

model.save(r'vegetable.h5')
model.summary()
```

Fruit Disease Testing

```
pip install numpy

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

model = load_model(r'fruit.h5')
img=image.load_img(r'Dataset Plant Disease/fruit-
dataset/fruit-
```

dataset/test/Apple___Black_rot/00e909aa-e3ae-4558-9961-336bb0f35db3___
JR_FrgE.S

```
8593.JPG',grayscale=False,target_size=(128,128))

img

x=image.img_to_array(img)
x=np.expand_dims(x,axis=())
pred=(model.predict(x) > 0.5).astype("int32")
pred
```

TEAN ID : PNT2022TMID32297

PROJECT NAME: Fertilizer Recommendation system for disease prediction

Vegetable Disease Testing

```
model—load_model(r'vegetable.h5') img=image.load_img(r'Dataset
Plant Disease/Veg-dataset/Veg-
    dataset/test_set/Potato___healthy/a8d687be-3777-403f-bae7-5c8c19340b3fRS HL
1738.JPG',grayscale=False,target_size=(128,128))
img

x—image.img_to_array(img)
x=np.expand_dims(x,axis=())

pred=(model.predict(x)  >  0.5).astype("int32")
pred
```

Python - Flask (app.py)

```
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")
mode11=load_model("fruit.h5")


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

```
@app.route('/prediction')
def prediction():
   return render_template('output.html') @app.route('/predict',methods=['POST']) def
   predict():
   if request.method=='POST':
     f=request.files['image']
     basepath=os.path.dirname(_file_)
          ath=os.path.j oin(basepath,'uploads', ecure_filename(f.filename))
     f.save(file_path)
     img=image.load_img(file_path,target_size=(128,128))
     x=image.img_to_array(img)
     x=np.expand_dims(x,axis=())
     plant=request.form['plant']
     print(plant)
     if(plant=="vegetable"):
        preds=model.predict_classes(x)
     print(preds)
     df=pd.read_excel('precautions-veg.xlsx')
     print(df.iloc[preds[O]]['caution']) else:
        preds=modell.predict_classes(x)

       df=pd.read_excel('precautions-fruits.xlsx')
    print(df.iloc[preds[O]]['caution']) return
    df.iloc[preds[O]]['caution']


if name          main '':app.run(debug=True)


HTML code:<html
lang="en'

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

TEAN ID : PNT2022TMID32297

PROJECT NAME: Fertilizer Recommendation system for disease prediction

```
<meta http-equiv="X-UA-Compatible"content="
<title>Fertilizers Recommendation System for Disease prediction </title>
<link href="https://cdn.bootcss.com/bootstrap/4.O.O/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.bootcss.com/popper.js/l.12.9/umd/popper.min.js"></script>
<script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
<script src="https://cdn.bootcss.com/bootstrap/4.O.O/js/bootstrap.min.js"></script>
<link href="{{ url_for('static', filename='css/Style.css') }}"rel="style


</head>
<body style="font-family:Montserrat;background-color:#333;">


<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:white;
paddingtop:1%">Fertilizers Recommendation System</div>
<div lass="topnav-right" style="padding-top:O.5%;">


<a class="active" href=" { { url_for('login')}}">Home
<a href="{{ url_for('index')}}">Logout</a>



</div>

<!--<body style="background-color:white";>
<header class="header">
<section id="navbar">
<br>
<hl style="color:green;"></i><center>Fertilizers Recommendation
System</center></hl>
```

```
        <div class="nav--items">

          <ul>
            <li><a     href="{{     url_for('index')}}">Home</a></li>
          <li><a         { url_for('logout')}}">Logout</a></li>


                    </ul>
        </div>
      </section>
    </header>

    <div  class="container">  <div  id="content"
      ="margin-top:2em">
      <div class="container">
       <divclass="
        <div class="col-sm-6 bd" > <br><br>
          <blockquote style="font-size:40px"> Detect if your Plant is infected! ! !</blockquote>
          <p><h5><i style="color:white;font-size:25px;"> Agriculture is one of the major sectors world wide.Over the years it has developed and the use of new technologies and equipments 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 labouratory.This application helps farmers in detecting the diseases by observing the spots on the leaves,which inturn saves effort and labor costs.
</i></h5></p>
        <!---<img
src="https://www.spokanecounty.org/ImageRepository/Document?documentId=41892"
style="height:240px"class="img-rounded" alt="Gesture">-->
        <h4 style="color:black;">Upload Image Here</h4> <br>
<form action = "http://localhost:5000/" id="upload-file" method=
enctype="multipart/form-data">
          <label for="imageUpload" class="upload-label">
          </label>
          <input type="file" name="image" d="imageUpload"accept=".png, .jpg, .jpeg,.pdf">
```

TEAN ID : PNT2022TMID32297

PROJECT NAME: Fertilizer Recommendation system for disease prediction

```
        </form>

            <button class="button buttonl">submit</button>
        </div>
        <div class="col-sm-6">
          <br><br><br><br><br><br>

            <img
src="https://github.com/Yanjare/Fertilizer_recommendation_System/blob/main/templates/ho
me.j
pg?raw=true" style="height:350px" class="img-rounded"alt="Ges

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



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



</html>
```

Style.css:

```
.bg-dark        {        background-color:
   #42678c!important;

#result { color:
   #0a1c4ed1;
   .header { top:O;
```

```
    margin:Opx;
    left: Opx; right:
    Opx; position:
    fixed;
    background-
    color:#00264d;
    color: white;
    box-shadow:
    Opx 8px 4px
    grey; overflow:
    hidden; padding-
    left:20px; font-
    family: 'Josefin
    Sans'; font-size:
    2vw; width:
    100%;
    height:8%; text-
    align: center;

.topnav { overflow:
hidden; background-
color: #333;


.topnav-right a {
float: left; color:
#f2f2f2; text-align:
center; padding:
14px 16px; text-
decoration: none;
font-size: 18px;
.topnav-right a:hover
{ background-color:
#ddd; color: black;
```

TEAN ID : PNT2022TMID32297

PROJECT NAME: Fertilizer Recommendation system for disease prediction

```
.topnav-right a.active {
background-color:
#565961; color: white;



.topnav-right {
float: right;
padding-right:
IOOpx;

.login{ margin-
top: IOOpx;

body {

background-color:#ffffff; background-
repeat: no-repeat; background-size:cover;
background-position: Opx Opx;



.button { border:
none; color: white;
padding: 16px 32px;
text-align: center;
text-decoration:
none; display: inline-
block; font-size:
16px; margin: 4px
2px; transition-
duration: 0.4s;
cursor: pointer;
```

```
.buttonl { background-color: white; color: black; border: 2px solid #4CAF50;

.buttonl:hover { background-color: #4CAF50; color: white;

img { border-radius: 50%;
```

GitHub & Project Demo Link

GitHub Link:

https://github.com/1BM-EPBL/1BM-Project-1383-1658386536

Project Demo Link:

https://drive.google.com/file/d/1Gy52ZZCge2dk16qt6CX61mwJ06fPObSE/view?usp=drivesdk