FERTILIZER RECOMMENDATION SYSTEM FOR DISEASE PREDICTION

A PROJECT REPORT

Submitted by

ROSHAN BANU.S	**(Reg.No:911519205017)**
ATCHAYA PRIYA.G	**(Reg.No:911519205004)**
SINDHU KUMARI.R	**(Reg.No:911519205021)**
SOWMIYA.M	**(Reg.No:911519205026)**

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

MOHAMED SATHAK ENGINEERING COLLEGE

KILAKARAI- 623 806

ANNA UNIVERSITY :: CHENNAI 600 025

NOVEMBER 2022

Project Report Format

1.INTRODUCTION

1.1 Project Overview

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

1.2 Purpose

This project is used to test the fruits and vegetables samples and identify the different diseases. Also, this project recommends fertilizers for predicted diseases.

2. LITERATURE SURVEY

2.1 Existing problem

Indumathi proposed a method for leaf disease detection and suggest fertilizers to cure leaf diseases. But the method involves less number of train and test sets which results in poor accuracy. Pandi selvi proposed a simple prediction method for soil-based fertilizer recommendation system for predicted crop diseases. This method gives less accuracy and prediction. Shiva reddy proposed an IoT based system for leaf disease detection and fertilizer recommendation which is based on Machine Learning techniques yields less 80 percentage accuracies.

2.2 Proposed solution

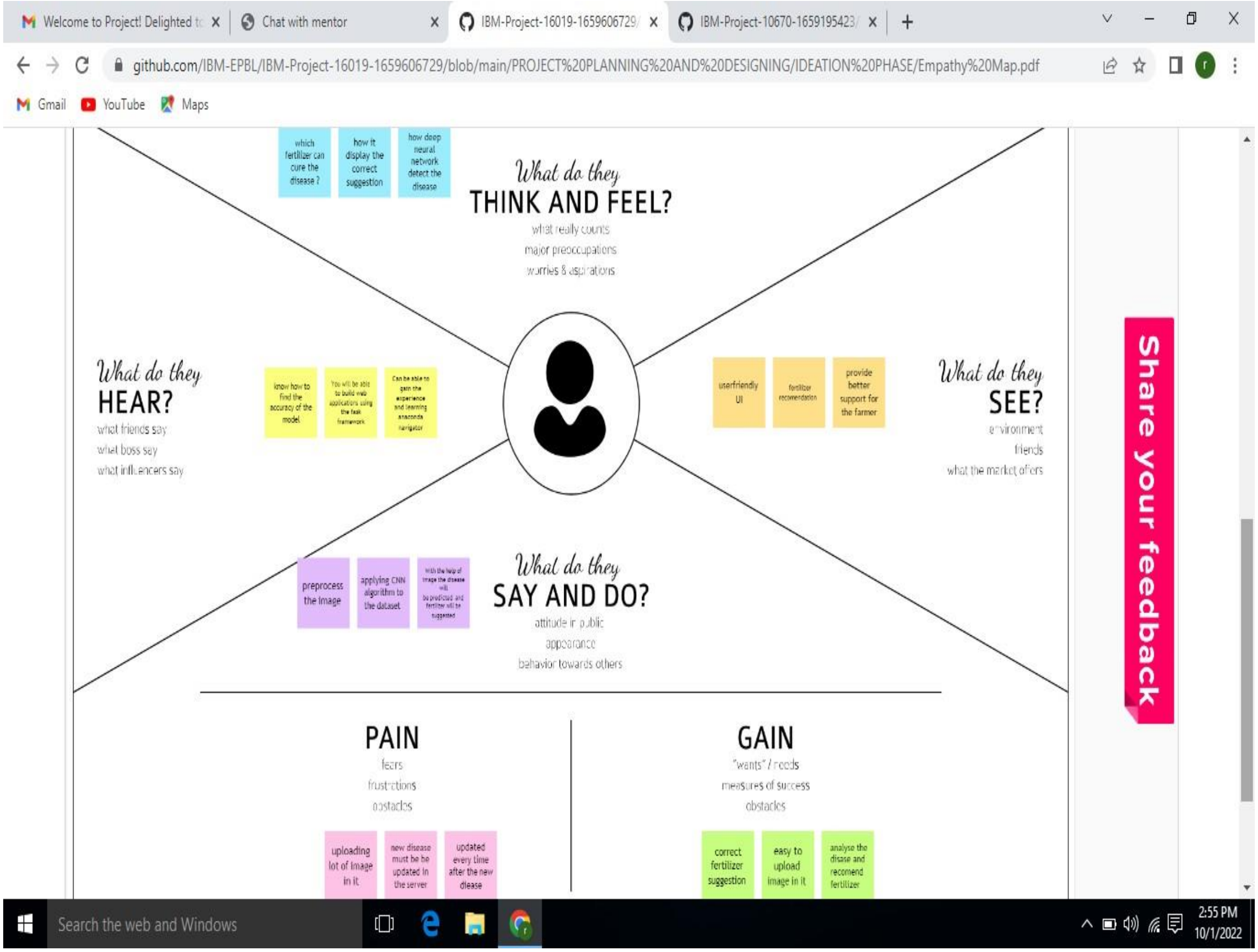
In this project work, a deep learning based neural network is used to train the collected datasets and test the same. The deep learning based neural network is CNN which gives more than 90% classification accuracies. By increasing the more number of dense layers and by modifying hyper parameters such as number of epochs, batch size, the accuracy rate can be increased to 95% to 98%.

2.3 Problem Statement

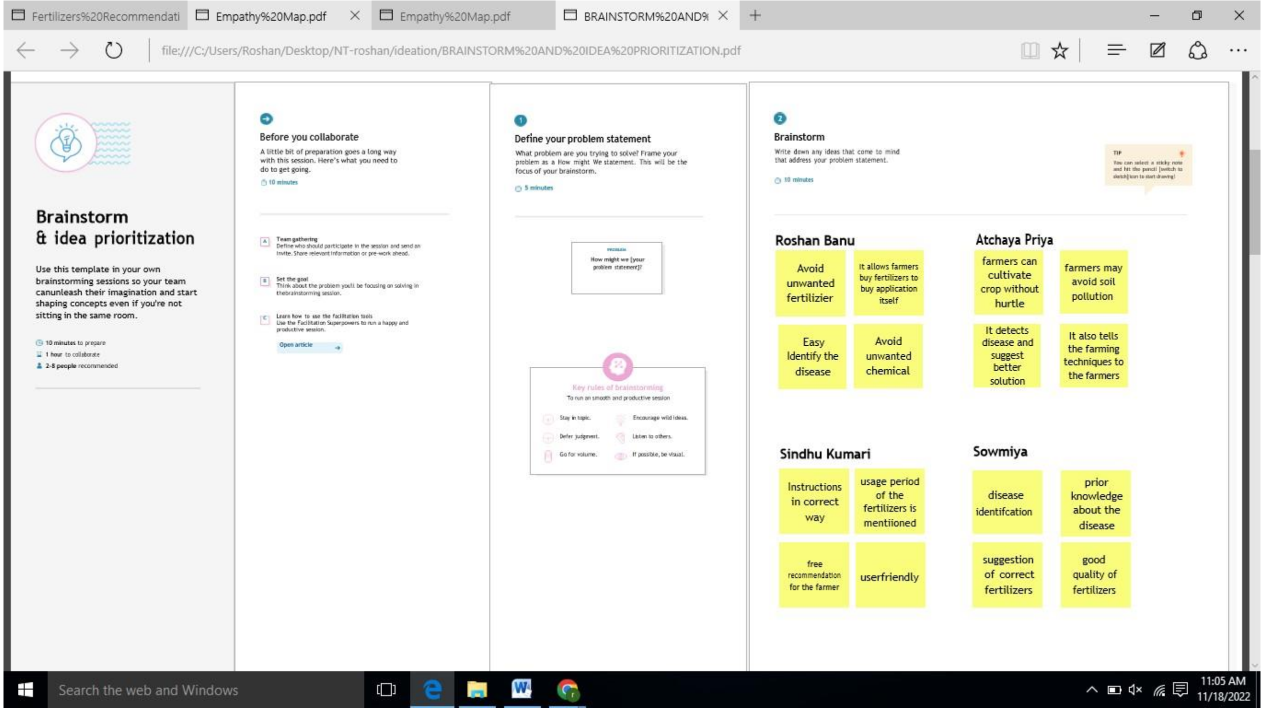
In today's world agriculture is very important for life and helps to save the natural resources around as. Doing agriculture is the very hard in current scenario because of many natural disaster are happening everyday. Most of the plants are affected by many disease due to pollution in water, air, soil. Identifying the disease is one of the huge hurtles in agriculture. Most of the plants are affected by leaf disease and its hard to find to correct fertilizer to cure. Identifying the disease in early stage is very important and easy to cure that.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas


Welcome to Project! Delighted to
Chat with mentor
IBM-Project-16019-1659606729/
IBM-Project-10670-1659195423/
github.com/IBM-EPBL/IBM-Project-16019-1659606729/blob/main/PROJECT%20PLANNING%20AND%20DESIGNING/IDEATION%20PHASE/Empathy%20Map.pdf
Gmail
YouTube
Maps
which fertilizer can cure the disease ?
how it display the correct suggestion
how deep neural network detect the disease
What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
What do they HEAR?
what friends say
what boss say
what influencers say
know how to find the accuracy of the model
You will be able to build web applications using the flask framework
Can be able to gain the experience and learning anaconda navigator
userfriendly UI
fertilizer recomendation
provide better support for the farmer
What do they SEE?
environment
friends
what the market offers
preprocess the image
applying CNN algorithm to the dataset
With the help of image the disease will be predicted and fertilizer will be suggested
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
PAIN
fears
frustrations
obstacles
uploading lot of image in it
new disease must be be updated in the server
updated every time after the new diease
GAIN
"wants" / needs
measures of success
obstacles
correct fertilizer suggestion
easy to upload image in it
analyse the disase and recomend fertilizer
Share your feedback
Search the web and Windows
2:55 PM
10/1/2022

3.2 Ideation & Brainstorming


Fertilizers%20Recommendati
Empathy%20Map.pdf
Empathy%20Map.pdf
BRAINSTORM%20AND%
file:///C:/Users/Roshan/Desktop/NT-roshan/ideation/BRAINSTORM%20AND%20IDEA%20PRIORITIZATION.pdf
Search the web and Windows
11:05 AM
11/18/2022

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team canunleash their imagination and start shaping concepts even if you're not sitting in the same room.

- 10 minutes to prepare
- 1 hour to collaborate
- 2-8 people recommended

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 thebrainstorming 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
How might we [your problem statement]?

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!

Roshan Banu

- Avoid unwanted fertilizier
- It allows farmers buy fertilizers to buy application itself
- Easy Identify the disease
- Avoid unwanted chemical

Atchaya Priya

- farmers can cultivate crop without hurtle
- farmers may avoid soil pollution
- It detects disease and suggest better solution
- It also tells the farming techniques to the farmers

Sindhu Kumari

- Instructions in correct way
- usage period of the fertilizers is mentiioned
- free recommendation for the farmer
- userfriendly

Sowmiya

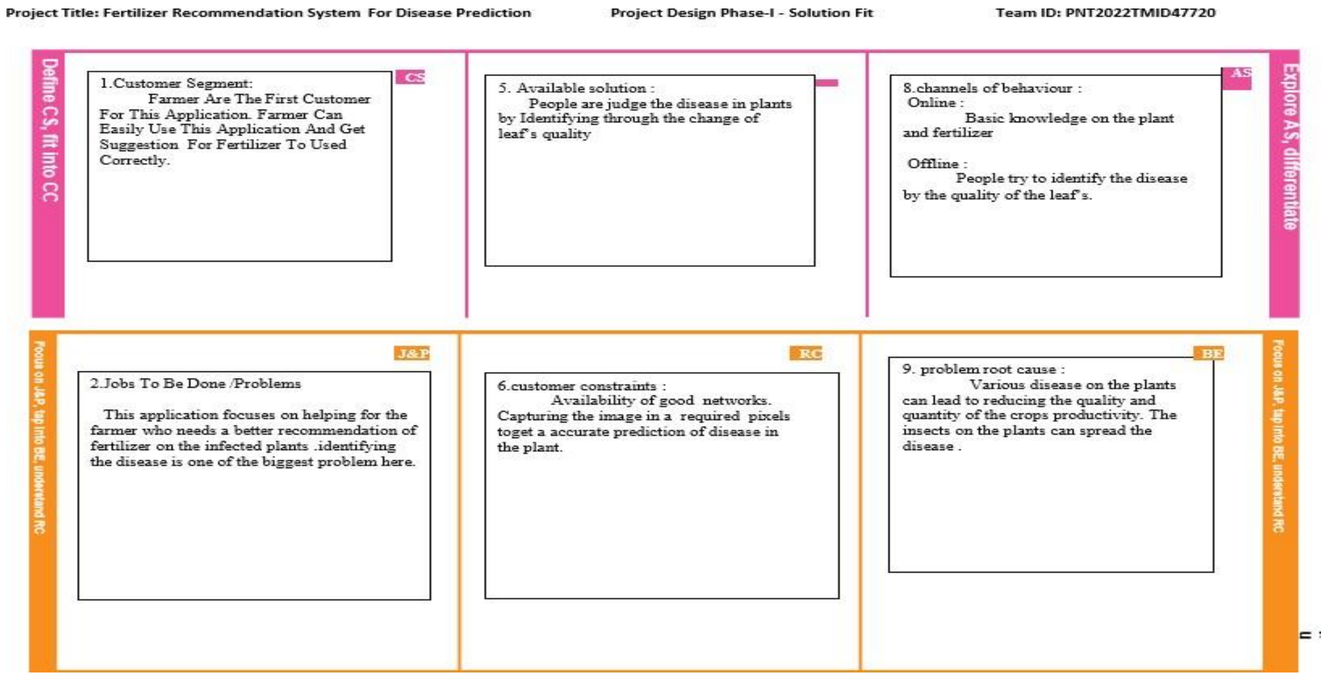
- disease identifcation
- prior knowledge about the disease
- suggestion of correct fertilizers
- good quality of fertilizers

3.3 Proposed Solution

S.No	Parameter	Description
1	Problem statement (problem to be solved)	Disease in plants reduced the quantity and quality of the plants productivity. Identifying the disease in plant is hard to find.
2	Idea/solution description	One of the solution of the problem is to identifying the disease in early stage and using the correct fertilizer.
3	Novelty / uniqueness	This application can suggest good fertilizer for the disease in the plant by recognizing the images
4	Social impact/customer satisfaction	It helps the farmer by identifying the disease in the early stage and increase the quality and quantity of crops in efficient way.
5	Business model(revenue model)	The application is recommends to farmer in subscription basis.
6	Scalability of the solution	This application can be improved

		by introducing online purchases of crops, fertilizer easily

3.4 Problem Solution Fit



Project Title: Fertilizer Recommendation System For Disease Prediction

Project Design Phase-I - Solution Fit

Team ID: PNT2022TMID47720

Define CS, fit into CC
CS
AS
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
J&P
RC
BE
Focus on J&P, tap into BE, understand RC

1.Customer Segment:
Farmer Are The First Customer For This Application. Farmer Can Easily Use This Application And Get Suggestion For Fertilizer To Used Correctly.

5. Available solution :
People are judge the disease in plants by Identifying through the change of leaf's quality

8.channels of behaviour :
Online :
Basic knowledge on the plant and fertilizer

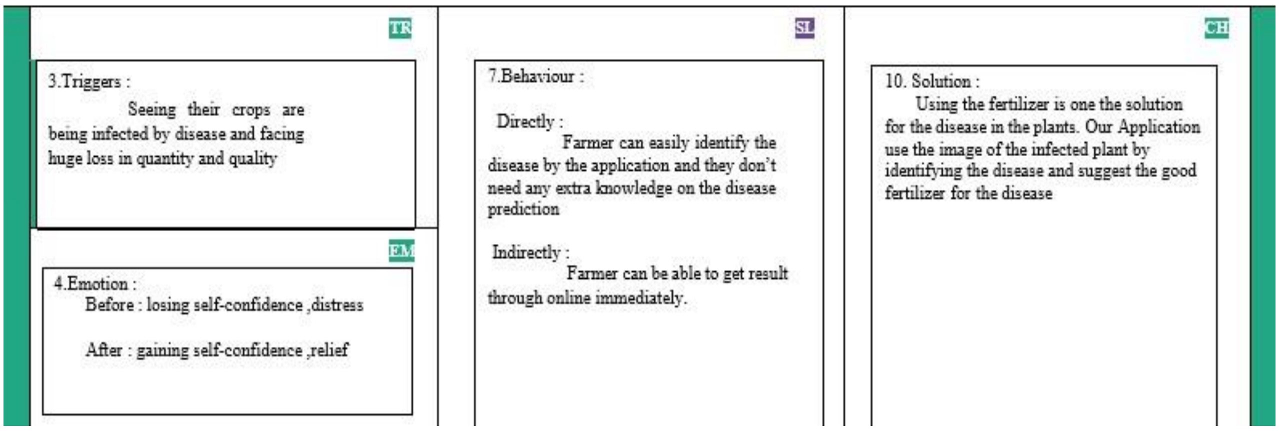
Offline :
People try to identify the disease by the quality of the leaf's.

2.Jobs To Be Done /Problems

This application focuses on helping for the farmer who needs a better recommendation of fertilizer on the infected plants .identifying the disease is one of the biggest problem here.

6.customer constraints :
Availability of good networks. Capturing the image in a required pixels toget a accurate prediction of disease in the plant.

9. problem root cause :
Various disease on the plants can lead to reducing the quality and quantity of the crops productivity. The insects on the plants can spread the disease .



3.Triggers :
Seeing their crops are being infected by disease and facing huge loss in quantity and quality

4.Emotion :
Before : losing self-confidence ,distress

After : gaining self-confidence ,relief

7.Behaviour :

Directly :
Farmer can easily identify the disease by the application and they don't need any extra knowledge on the disease prediction

Indirectly :
Farmer can be able to get result through online immediately.

10. Solution :
Using the fertilizer is one the solution for the disease in the plants. Our Application use the image of the infected plant by identifying the disease and suggest the good fertilizer for the disease

4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

Following are the functional requirements of the proposed solution .

Fr-1- User registration: Registration through form Registration through Gmail

Fr-2 -User confirmation: Confirmation via OTP Confirmation via Email

Fr-3- Capturing image: Capture the image of the leaf And check the parameter of the captured image .

Fr-4 -Image processing: Upload the image for the prediction of the disease in the leaf.

Fr-5 -Leaf identification: Identify the leaf and predict the disease in leaf.

Fr-6 -Image description: Suggesting the best fertilizer for the disease .

4.2 Non-functional requirement

Following are the non-functional requirement of the proposed solution

Nfr-1- Usability: Datasets of all the leaf is used to detecting the disease that present in the leaf.

Nfr-2- Security: The information belongs to the user and leaf are secured highly.

Nfr-3- Reliability: The leaf quality is important for the predicting the disease in leaf.

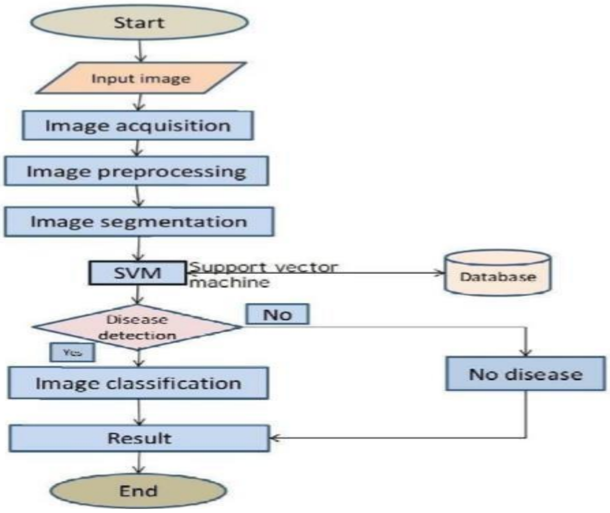
Nfr-4- Performance: The performance is based on the quality of the leaf used for disease prediction

Nfr-5- Availability :It is available for all user to predict the disease in the plant

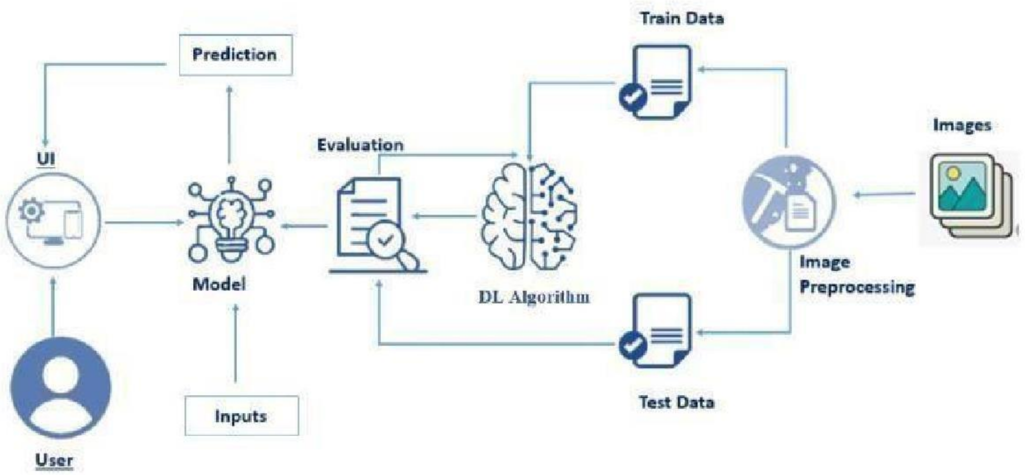
Nfr-6 –Scalability: Increasing the prediction of the disease in the leaf

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories

- Create a model which can classify diseased fruit plants from given images. I also need to test the model and deploy it on IBM Cloud.
- Create a model which can classify diseased vegetable plants from given images.
- Create a model which can classify diseased vegetable plants from given images and train on IBM Cloud.
- As a user, I can register by entering my email, password, and confirming my password or via OAuth API .
- As a user, I will be redirected to a page where I can upload my pictures of crops.
- As a user, I can view the results and then obtain the suggestions provided by the ML model.
- A base Flask web app must be created as an interface for the ML model.
- As a user/admin/shopkeeper, I can log into the application by entering email & password.
- As a user, I can view the previous results and history .
- Integrate Flask, CNN model with Cloudant DB Containerize Flask app using Docker.
- As an admin, I can view other user details and uploads for other purposes.
- As a shopkeeper, I can enter fertilizer products and then update the details if any.
- Create and deploy Helm charts using Docker Image made before.

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points (Total)
Sprint-1	Model Creation and Training (Fruits)		Create a model which can classify diseased fruit plants from given images. I also need to test the model and deploy it on IBM Cloud	8
	Model Creation and Training (Vegetables)		Create a model which can classify diseasedvegetable plants from given images	2

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points (Total)
Sprint-2	Model Creation and Training (Vegetables)		Create a model which can classify diseased vegetable plants from given images and train onIBM Cloud	6
	Registration	USN-1	As a user, I can register by entering my email, password, and confirming my password or via OAuth API	3
	Upload page	USN-2	As a user, I will be redirected to a page where Ican upload my pictures of crops	4

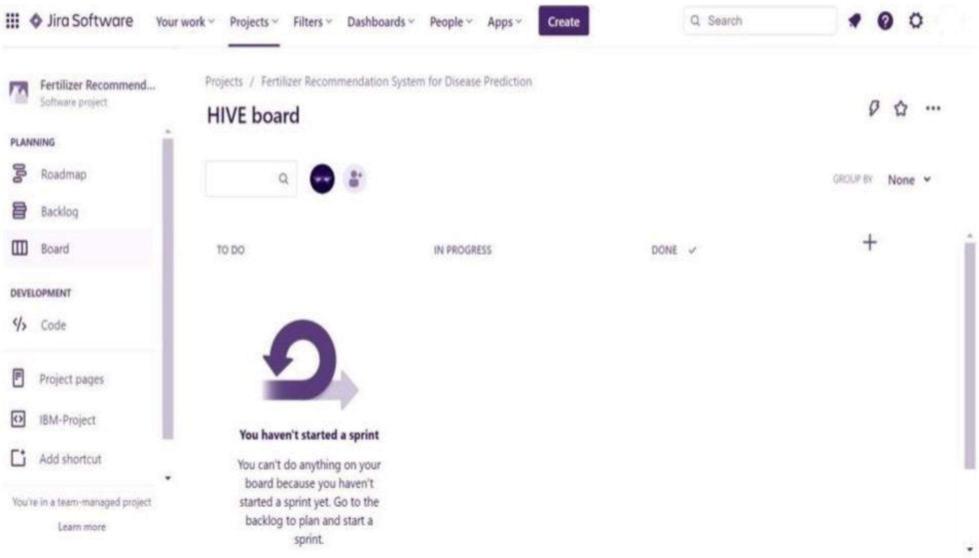
	Suggestion results	USN-3	As a user, I can view the results and then obtainthe suggestions provided by the ML model	4
	Base Flask App		A base Flask web app must be created as an interface for the ML model	2
Sprint-3	Login	USN-4	As a user/admin/shopkeeper, I can log into the application by entering email & password	2
	User Dashboard	USN-5	As a user, I can view the previous results and history	3
	Integration		Integrate Flask, CNN model with Cloudant DB	5
	Containerization		Containerize Flask app using Docker	2
Sprint-4	Dashboard (Admin)	USN-6	As an admin, I can view other user details and uploads for other purposes	2
	Dashboard (Shopkeeper)	USN-7	As a shopkeeper, I can enter fertilizer productsand then update the details if any	2
	Containerization		Create and deploy Helm charts using Docker Image made before	2

6.2 Sprint Delivery Schedule

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	10	6 Days	24 Oct 2022	29 Oct 2022	10	30 Oct 2022
Sprint-2	15	6 Days	31 Oct 2022	05 Nov 2022	15	06 Nov 2022
Sprint-3	15	6 Days	07 Nov 2022	12 Nov 2022	15	13 Nov 2022
Sprint-4	12	6 Days	14 Nov 2022	19 Nov 2022	10	20 Nov 2022

6.3 Reports from JIRA

Screenshots:



Projects / Fertilizer Recommendation System for Disease Prediction

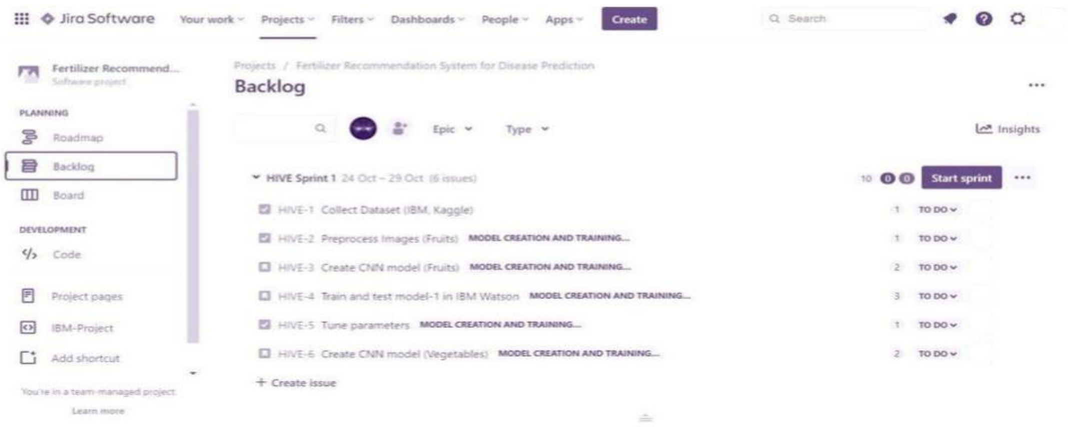
HIVE board

You haven't started a sprint

You can't do anything on your board because you haven't started a sprint yet. Go to the backlog to plan and start a sprint.

You're in a team-managed project

Learn more



Projects / Fertilizer Recommendation System for Disease Prediction

Backlog

HIVE Sprint 1 24 Oct – 29 Oct (6 issues) 10 0 0

☑ HIVE-1	Collect Dataset (IBM, Kaggle)		1	TO DO
☑ HIVE-2	Preprocess Images (Fruits)	MODEL CREATION AND TRAINING...	1	TO DO
□ HIVE-3	Create CNN model (Fruits)	MODEL CREATION AND TRAINING...	2	TO DO
□ HIVE-4	Train and test model-1 in IBM Watson	MODEL CREATION AND TRAINING...	3	TO DO
☑ HIVE-5	Tune parameters	MODEL CREATION AND TRAINING...	1	TO DO
□ HIVE-6	Create CNN model (Vegetables)	MODEL CREATION AND TRAINING...	2	TO DO

7. CODING & SOLUTIONING

7.1 Features

Feature 1:Registration

Feature 2:Login

Feature 3:User interface

Feature 4:Store database

Feature 5:Send Alert Emails to us

8. TESTING

8.1 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [Fertilizer Recommendation system for plant 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 how they 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
Total	44	31	13	32	119

3. Test Case Analysis

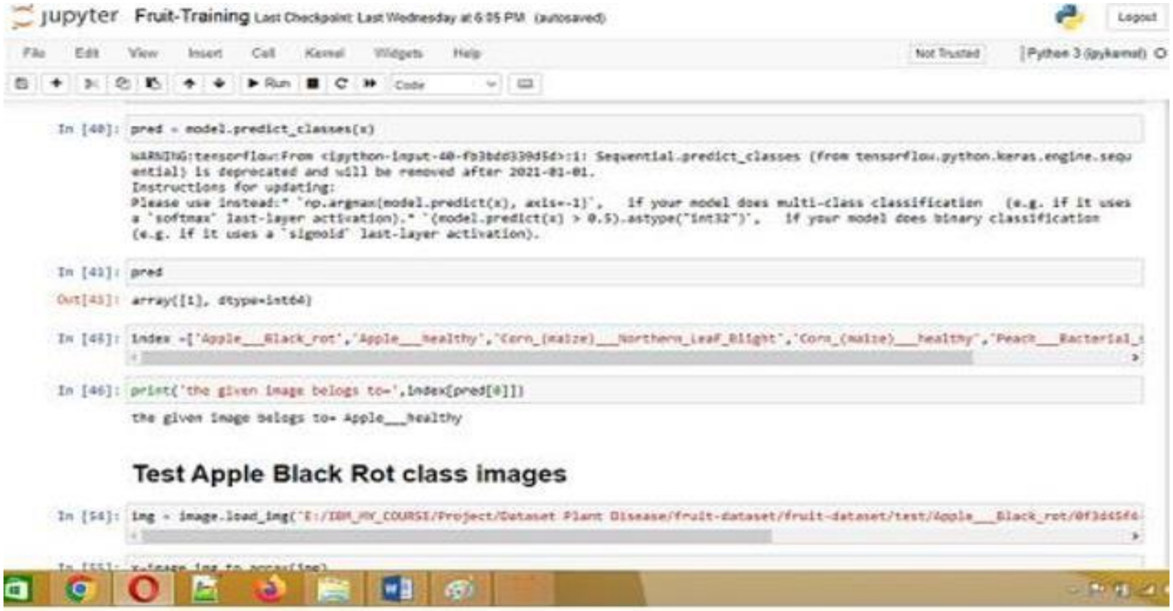
This report shows the number of test cases that have passed, failed, and untested

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

6. RESULTS

Final findings (output) of the project given below in the form of screenshot:

Training and Testing of Fruit dataset.

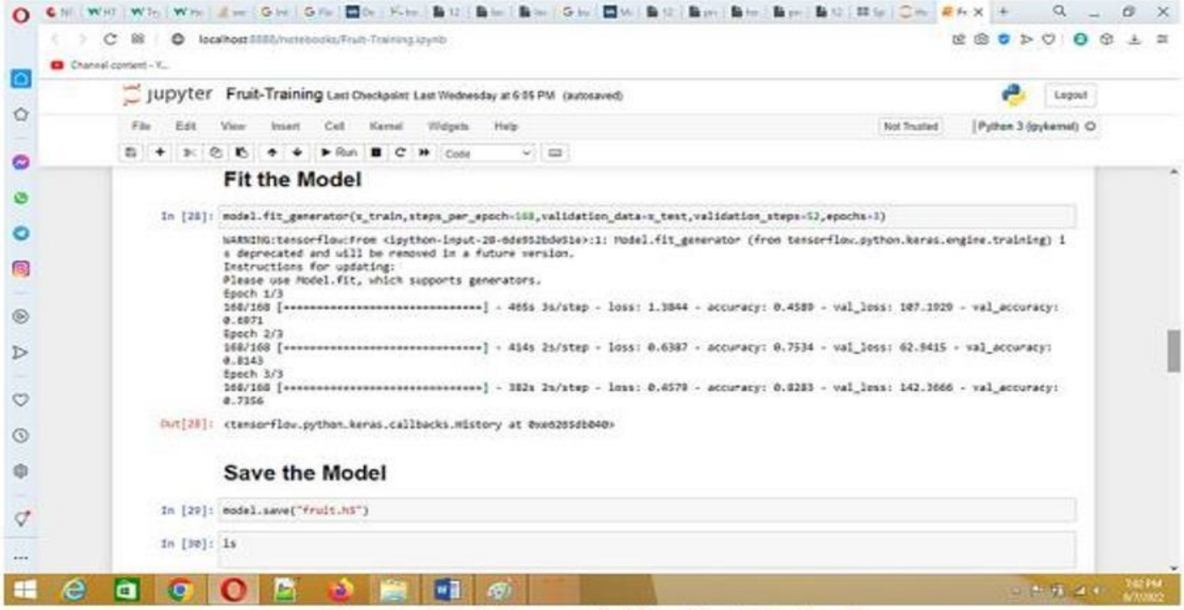


```
In [40]: pred = model.predict_classes(x)

         WARNING:tensorflow:From <ipython-input-40-fb3bdd339d5d>:1: Sequential.predict_classes (from tensorflow.python.keras.engine.sequ
         ential) is deprecated and will be removed after 2021-01-01.
         Instructions for updating:
         Please use instead:* `np.argmax(model.predict(x), axis=-1)`,   if your model does multi-class classification   (e.g. if it uses
         a `softmax` last-layer activation).* `(model.predict(x) > 0.5).astype("int32")`,   if your model does binary classification
         (e.g. if it uses a `sigmoid` last-layer activation).

In [41]: pred

Out[41]: array([1], dtype=int64)

In [45]: index =['Apple___Black_rot','Apple___healthy','Corn_(maize)___Northern_Leaf_Blight','Corn_(maize)___healthy','Peach___Bacterial_s

In [46]: print('the given image belogs to=',index[pred[0]])

         the given image belogs to= Apple___healthy
```

Test Apple Black Rot class images

```
In [54]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple___Black_rot/0f3d45f6-

In [55]: x=image.img_to_array(img)
```

Figure.6.2 Test the Fruit dataset


jupyter Fruit-Training Last Checkpoint: Last Wednesday at 6:05 PM (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Logout

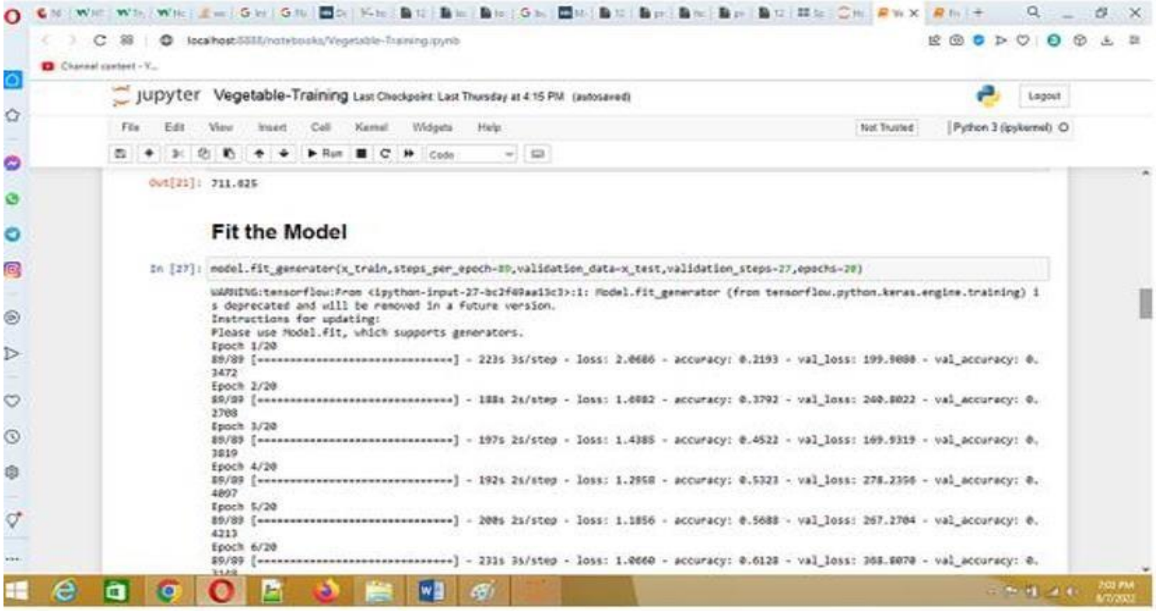
Fit the Model

```
In [28]: model.fit_generator(x_train,steps_per_epoch=168,validation_data=x_test,validation_steps=52,epochs=3)

WARNING:tensorflow:From <ipython-input-28-6de952bde51e>:1: Model.fit_generator (from tensorflow.python.keras.engine.training) i
s deprecated and will be removed in a future version.
Instructions for updating:
Please use Model.fit, which supports generators.
Epoch 1/3
168/168 [==============================] - 465s 3s/step - loss: 1.3844 - accuracy: 0.4589 - val_loss: 107.1929 - val_accuracy:
0.6071
Epoch 2/3
168/168 [==============================] - 414s 2s/step - loss: 0.6387 - accuracy: 0.7534 - val_loss: 62.9415 - val_accuracy:
0.8143
Epoch 3/3
168/168 [==============================] - 382s 2s/step - loss: 0.4579 - accuracy: 0.8283 - val_loss: 142.3666 - val_accuracy:
0.7356

Out[28]: <tensorflow.python.keras.callbacks.History at 0xe6285db040>
```

Save the Model

```
In [29]: model.save("fruit.h5")

In [30]: ls
```

Figure.6.1. Fit a model for Fruit dataset

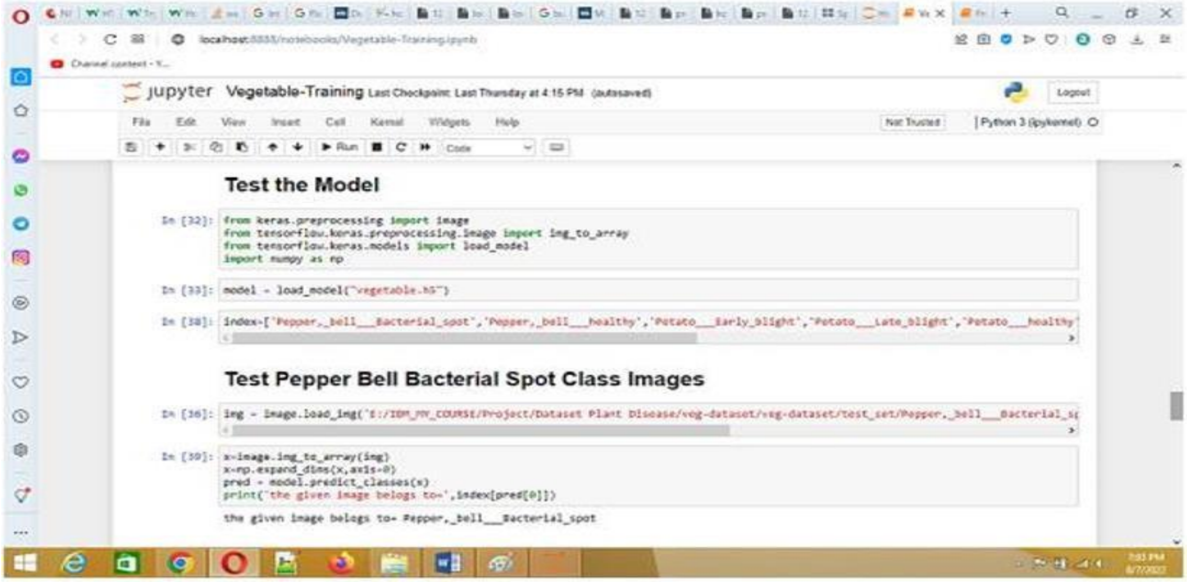

jupyter Vegetable-Training Last Checkpoint: Last Thursday at 4:15 PM (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Logout

```
Out[21]: 711.825
```

Fit the Model

```
In [27]: model.fit_generator(x_train,steps_per_epoch=89,validation_data=x_test,validation_steps=27,epochs=20)

WARNING:tensorflow:From <ipython-input-27-bc2f49aa13c3>:1: Model.fit_generator (from tensorflow.python.keras.engine.training) i
s deprecated and will be removed in a future version.
Instructions for updating:
Please use Model.fit, which supports generators.
Epoch 1/20
89/89 [==============================] - 223s 3s/step - loss: 2.0686 - accuracy: 0.2193 - val_loss: 199.9080 - val_accuracy: 0.
3472
Epoch 2/20
89/89 [==============================] - 188s 2s/step - loss: 1.6082 - accuracy: 0.3792 - val_loss: 240.8022 - val_accuracy: 0.
2708
Epoch 3/20
89/89 [==============================] - 197s 2s/step - loss: 1.4385 - accuracy: 0.4522 - val_loss: 169.9319 - val_accuracy: 0.
3819
Epoch 4/20
89/89 [==============================] - 192s 2s/step - loss: 1.2958 - accuracy: 0.5323 - val_loss: 278.2356 - val_accuracy: 0.
4097
Epoch 5/20
89/89 [==============================] - 208s 2s/step - loss: 1.1856 - accuracy: 0.5688 - val_loss: 267.2704 - val_accuracy: 0.
4213
Epoch 6/20
89/89 [==============================] - 231s 3s/step - loss: 1.0660 - accuracy: 0.6128 - val_loss: 368.8070 - val_accuracy: 0.
```

Figure.6.3. Train the Vegetable dataset


jupyter Vegetable-Training Last Checkpoint: Last Thursday at 4:15 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code

Test the Model

```
In [32]: from keras.preprocessing import image
         from tensorflow.keras.preprocessing.image import img_to_array
         from tensorflow.keras.models import load_model
         import numpy as np

In [33]: model = load_model("vegetable.h5")

In [38]: index=['Pepper,_bell__Bacterial_spot','Pepper,_bell__healthy','Potato__Early_blight','Potato__Late_blight','Potato__healthy'
```

Test Pepper Bell Bacterial Spot Class Images

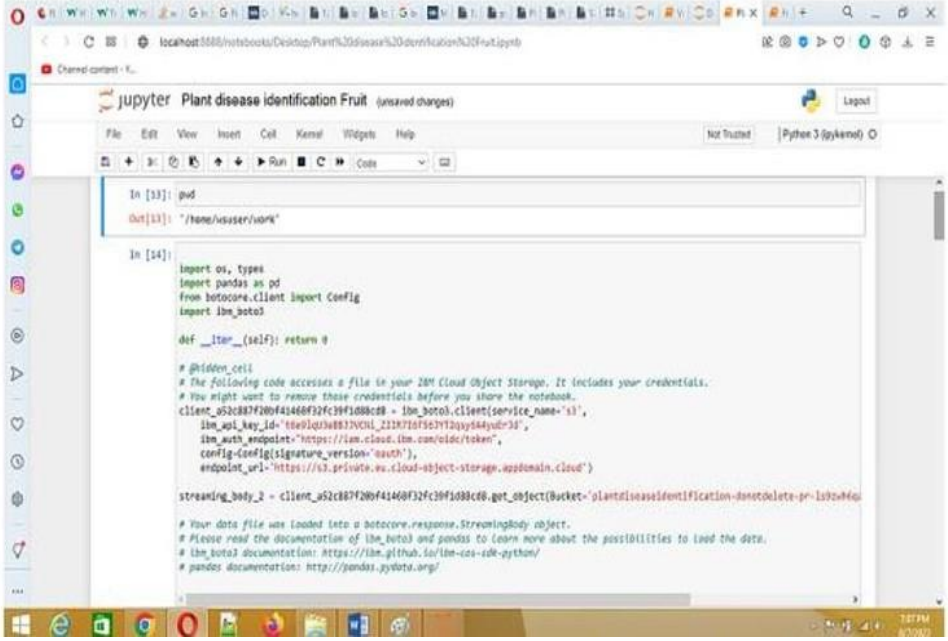
```
In [36]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/veg-dataset/veg-dataset/test_set/Pepper,_bell__Bacterial_sp

In [39]: x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)
         pred = model.predict_classes(x)
         print('the given image belogs to=',index[pred[0]])

         the given image belogs to= Pepper,_bell__Bacterial_spot
```

Figure.6.4. Test the Vegetable dataset

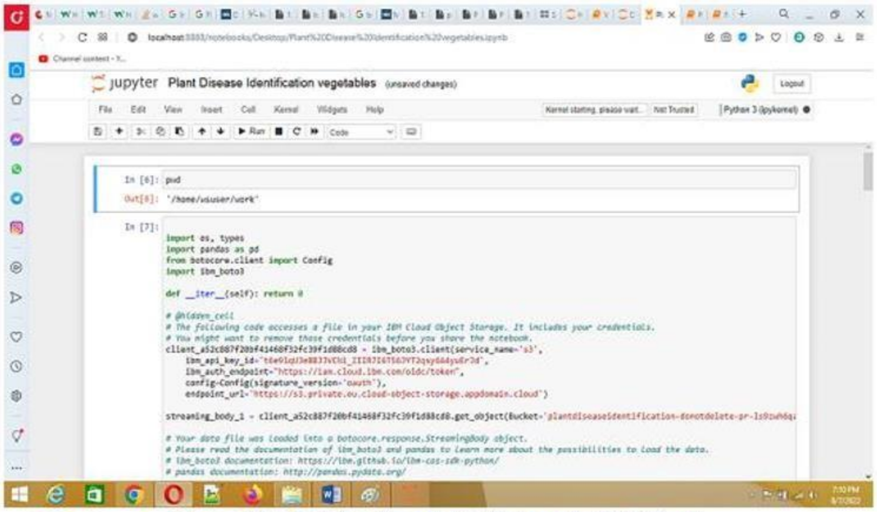
Train and Test Vegetable dataset IBM Cloud

Due to CUH limit exceeds, I have downloaded the notebooks and opened in Jupyter notebook

(i). Fruit dataset


jupyter Plant disease identification Fruit (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code

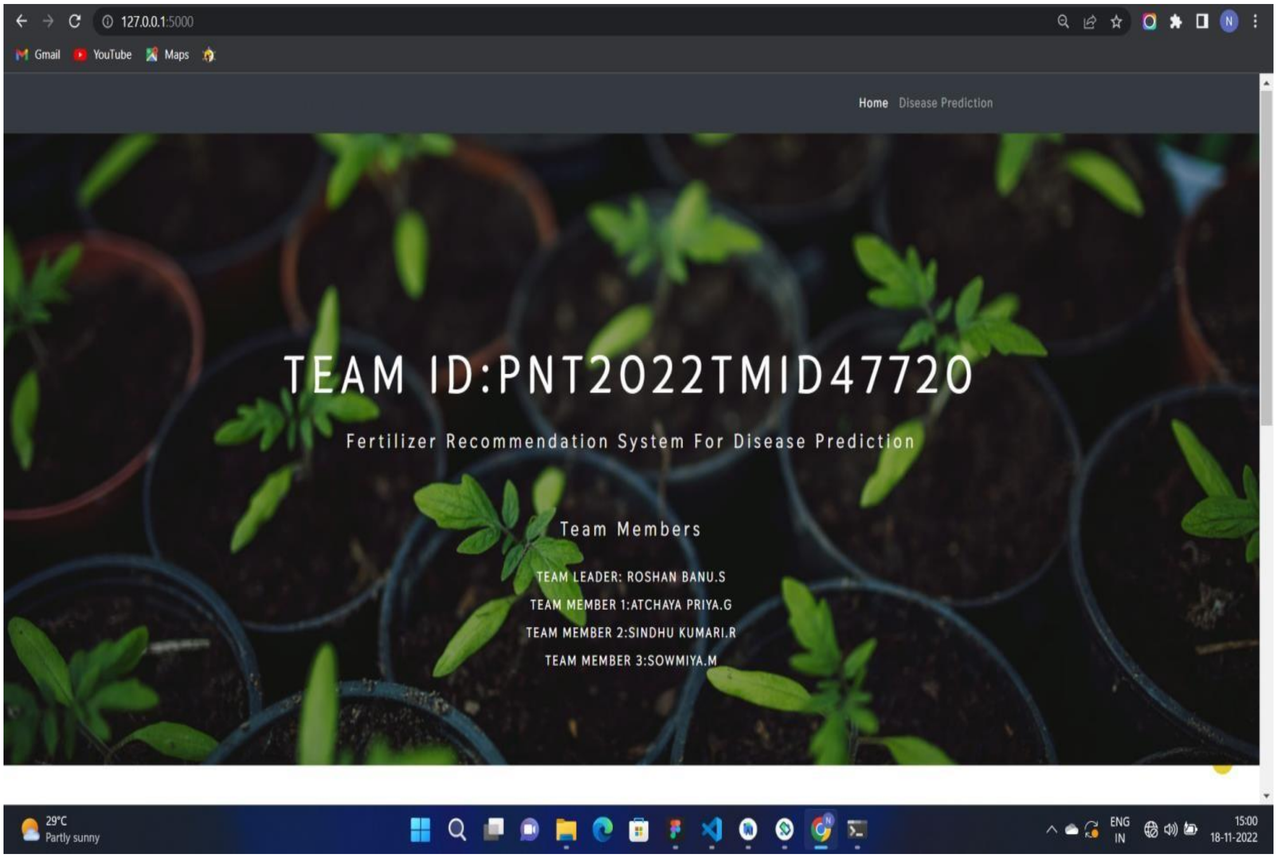
```
In [13]: pwd
Out[13]: '/home/wsuser/work'

In [14]:
         import os, types
         import pandas as pd
         from botocore.client import Config
         import ibm_boto3

         def __iter__(self): return 0

         # @hidden_cell
         # The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
         # You might want to remove those credentials before you share the notebook.
         client_a52c887f20bf41460f32fc39f1d88cd8 = ibm_boto3.client(service_name='s3',
             ibm_api_key_id='t6e9lQU3eBB3JVCHL_Z2IR7I6f5t3Yf2qsy544yuDr3d',
             ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
             config=Config(signature_version='oauth'),
             endpoint_url='https://s3.private.eu.cloud-object-storage.appdomain.cloud')

         streaming_body_2 = client_a52c887f20bf41460f32fc39f1d88cd8.get_object(Bucket='plantdiseaseidentification-donotdelete-pr-1s9zw6q

         # Your data file was loaded into a botocore.response.StreamingBody object.
         # Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
         # ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
         # pandas documentation: http://pandas.pydata.org/
```


jupyter Plant Disease Identification vegetables (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Kernel starting, please wait... Not Trusted
Python 3 (ipykernel)
Run
Code

```
In [6]: pwd
Out[6]: '/home/wsuser/work'

In [7]:
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you store the notebook.
client_a52c887f20bf41468f32fc39f1d88cd8 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='t6e91qJ3eBB3JVChl_IIIB7I6T5G7YT2qsy6Adyu6r3d',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.eu.cloud-object-storage.appdomain.cloud')

streaming_body_1 = client_a52c887f20bf41468f32fc39f1d88cd8.get_object(Bucket='plantdiseaseidentification-donotdelete-pr-ls9zwh6q

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

Figure.6.6. Training Vegetable Dataset in IBM Cloud

OUTPUT



TEAM ID:PNT2022TMID47720

Fertilizer Recommendation System For Disease Prediction

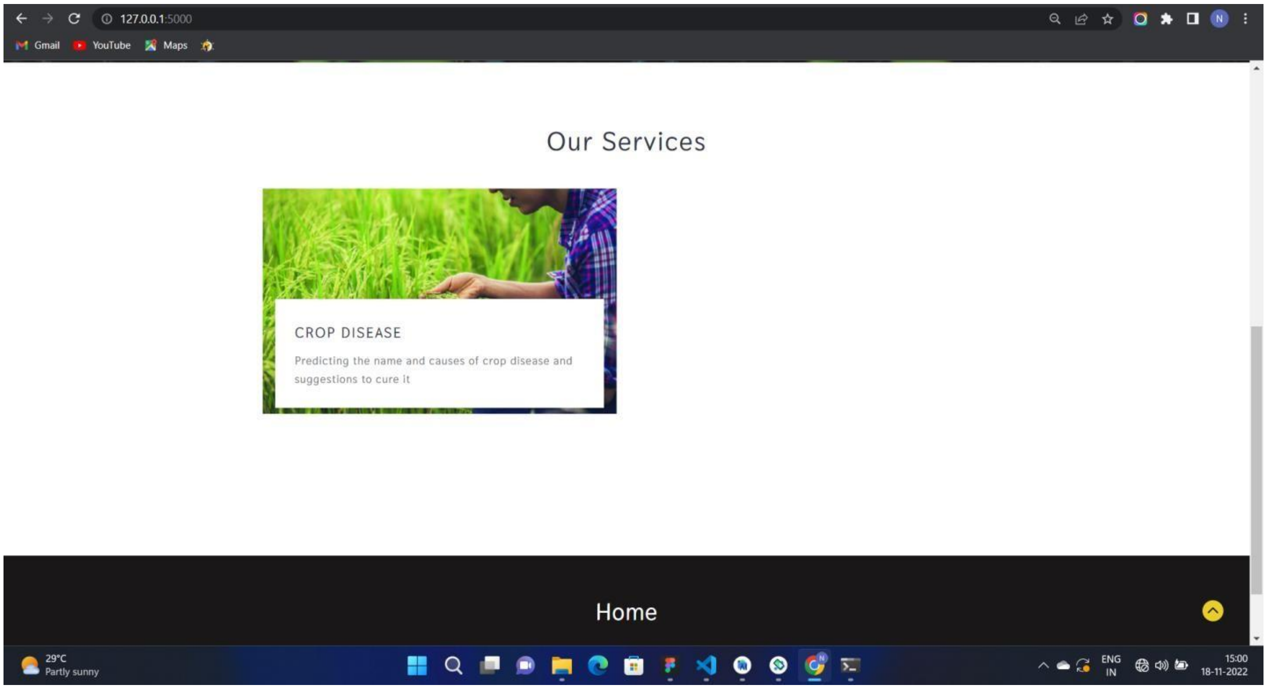
Team Members

TEAM LEADER: ROSHAN BANU.S

TEAM MEMBER 1:ATCHAYA PRIYA.G

TEAM MEMBER 2:SINDHU KUMARI.R

TEAM MEMBER 3:SOWMIYA.M

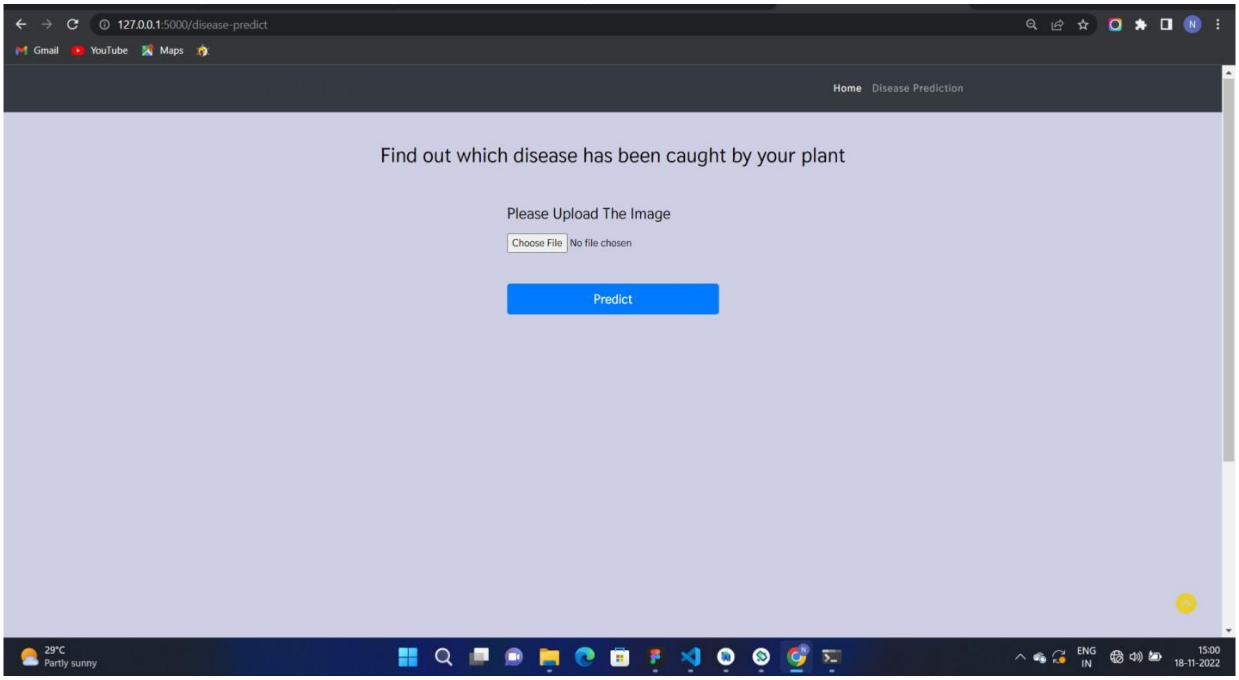


Our Services

CROP DISEASE

Predicting the name and causes of crop disease and suggestions to cure it

Home

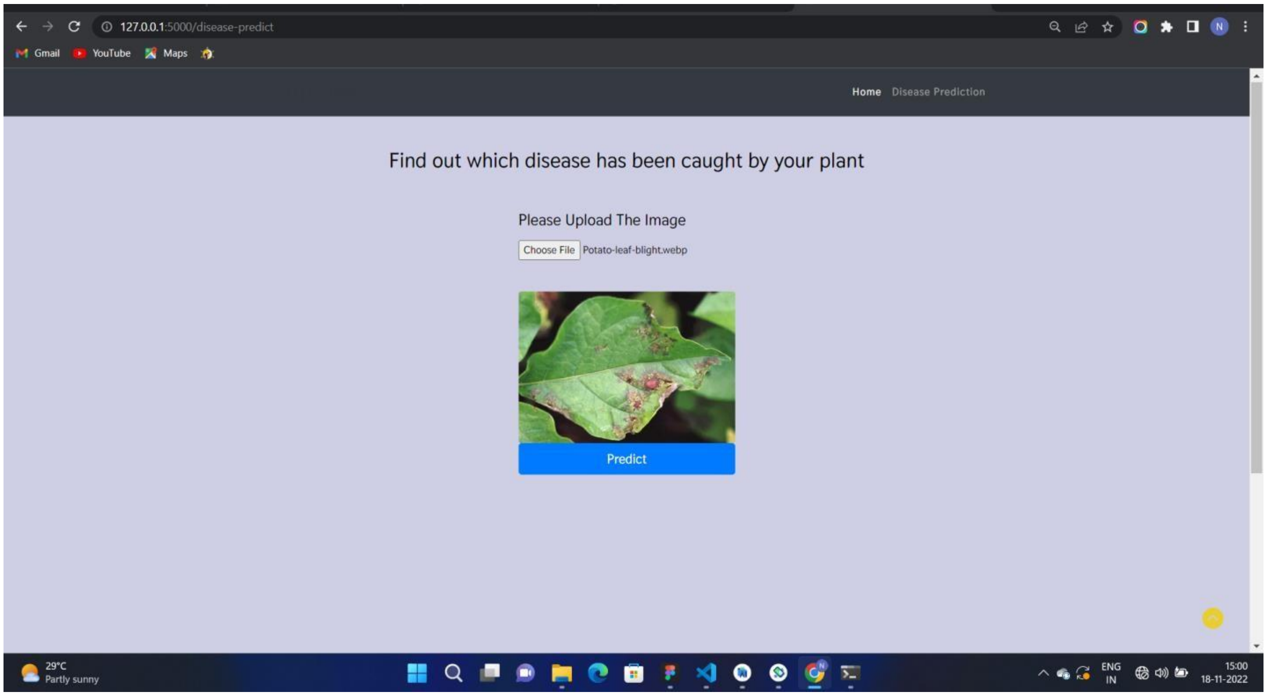


Find out which disease has been caught by your plant

Please Upload The Image

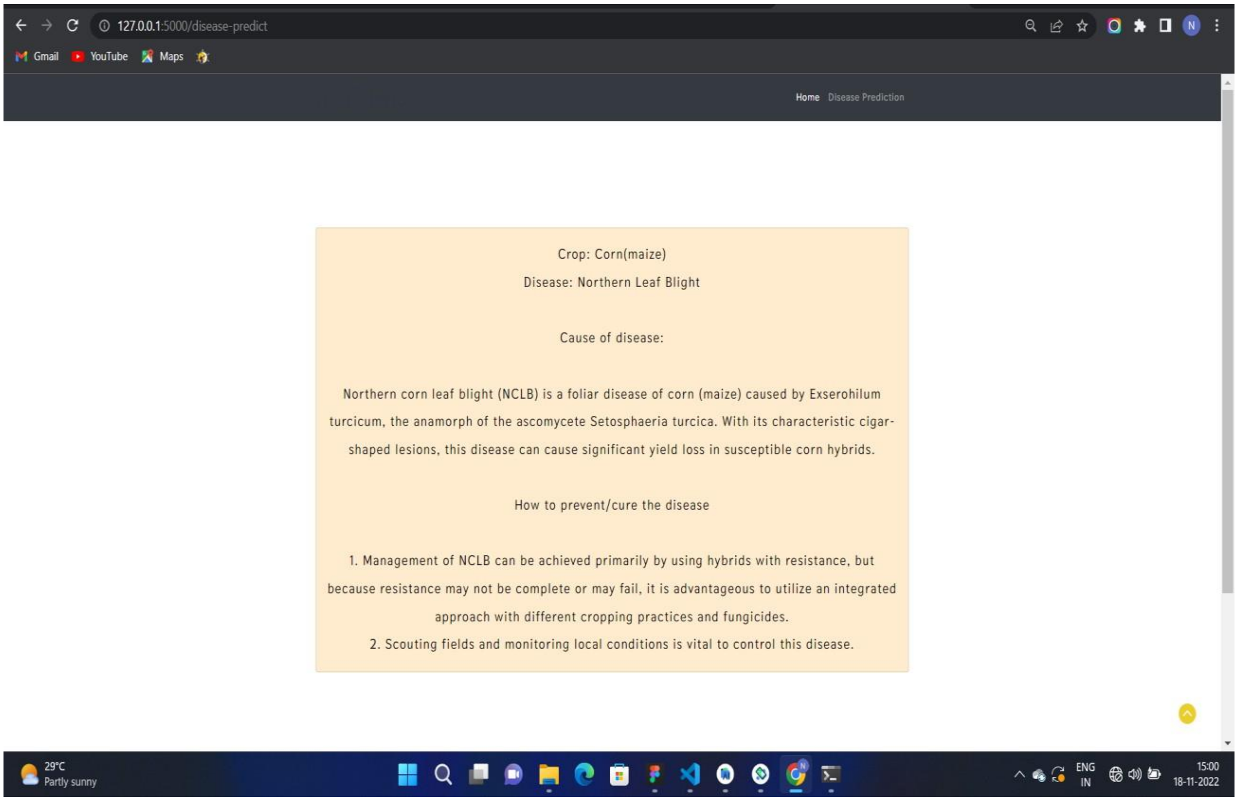
Choose File No file chosen

Predict



Find out which disease has been caught by your plant

Please Upload The Image



Crop: Corn(maize)

Disease: Northern Leaf Blight

Cause of disease:

Northern corn leaf blight (NCLB) is a foliar disease of corn (maize) caused by Exserohilum turcicum, the anamorph of the ascomycete Setosphaeria turcica. With its characteristic cigar-shaped lesions, this disease can cause significant yield loss in susceptible corn hybrids.

How to prevent/cure the disease

1. Management of NCLB can be achieved primarily by using hybrids with resistance, but because resistance may not be complete or may fail, it is advantageous to utilize an integrated approach with different cropping practices and fungicides.
2. Scouting fields and monitoring local conditions is vital to control this disease.

7.ADVANTAGES & DISADVANTAGES

List of 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.

List of 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.

8.APPLICATIONS

1. The trained network model used to classify the image patterns with high accuracy.

2. The proposed model not only used for plant disease classification but also for other image pattern classification such as animal classification.

3. This project work application involves not only image classification but also for pattern recognition.

9. CONCLUSIONS

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**.**

10. 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 Open CV 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

https://github.com/IBM-EPBL/IBM-Project-10670-1659195423

GITHUB & PROJECT DEMO LINK

https://github.com/IBM-EPBL/IBM-Project-10670-1659195423

PROJECT DEMO VIDEO LINK

https://drive.google.com/drive/folders/1ZXG3l8yAlwpyC7LYsuH_6fWg0a5KvPYw