PROJECT REPORT

Fertilizers Recommendation for Disease Prediction

Team ID: PNT2022TMID24348

Team Leader : M GUNA SEKHAR-113319104024

Team Member : V SARATH -113319104101

Team Member : M.CHARAN-113319104047

Team Member : V.HEMASAI-113319104097

Team Member :T.RAJESH -113319104094

Department of Computer Science and Engineering

Velammal Institute of Technology

Panchetti, Tiruvallur Dist, Chennai -

601204

1.INTRODUCTION

1.1 **Project Overview**

1.2 **Purpose**

2. **LITERATURE SURVEY**

2.1 **Existing problem**

2.2 **References**

2.3 **Problem Statement Definition**

3. **IDEATION & PROPOSED SOLUTION**

3.1 **Empathy Map Canvas**

3.2 **Ideation & Brainstorming**

3.3 **Proposed Solution**

3.4 **Problem Solution fit**

4. **REQUIREMENT ANALYSIS**

4.1 **Functional requirement**

4.2 **Non-Functional requirements**

5. **PROJECT DESIGN**

5.1 **Data Flow Diagrams**

5.2 **Solution & Technical Architecture**

5.3 **User Stories**

6. **PROJECT PLANNING & SCHEDULING**

6.1 **Sprint Planning & Estimation**

6.2 **Sprint Delivery Schedule**

6.3 **Reports from JIRA**

7. **CODING & SOLUTIONING**

7.1 **Python – App.py**

7.2 **Feature 1**

7.3 **Feature 2**

8. **TESTING**

8.1 **Test Cases**

8.2 **User Acceptance Testing**

9. **RESULTS**

9.1 **Performance Metrics**

10. **ADVANTAGES & DISADVANTAGES**

11. **CONCLUSION**

12. **FUTURE SCOPE**

13. **APPENDIX**

Source Code

GitHub & Project Demo Link

1.INTRODUCTION :

- 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.

1.1 Project Overview

- 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 changes in cultivation method and inadequate plant protection techniques and suggest all the precautions that can be taken for those diseases.

1.2 Purpose

- To Detect and recognize the plant diseases and to recommend fertilizer, it is necessary to identify the diseases and to recommend to get different and useful features needed for the purpose of analyzing later.

- To provide symptoms in identifying the disease at its earliest. Hence the authors proposed and implemented new fertilizers Recommendation System for Crop Disease Prediction.

2.LITREATURE SURVEY

2.1 Existing Problem

- Adequate mineral nutrition is central to crop production. However, it can also exert considerable Influence on disease development. Fertilizer application can increase or decrease development of diseases caused by different pathogens, and the mechanisms responsible are complex, including effects of nutrients on plant growth, plant resistance mechanisms and direct effects on the pathogen. The effects of mineral nutrition on plant disease and the mechanisms responsible for those effects have been dealt with comprehensively elsewhere. In India, around 40% of land is kept and grown using reliable irrigation technologies, while the rest relies on the monsoon environment for water. Irrigation decreases reliance on the monsoon, increases food security, and boosts agricultural production.

- Most research articles use humidity, moisture, and temperature sensors near the plant's root, with an external device handling all of the data provided by the sensors and transmitting it directly to an Android application. It was created to measure the approximate values of temperature, humidity and moisture sensors that were programmed into a microcontroller to manage the amount of water.

2.2 References :

- Reyes Angie .K, Juan C. Caicedo, and Jorge E. Camargo, "Fine-tuning Deep Convolutional Networks for Plant Recognition", In CLEF (Working Notes), 2015.
- Hamrouni .L, Aiadi .O, Khaldi .B and Kherfi .M.L, "Plants Species Identification using Computer Vision Techniques", Revue des Bioressources 7, no. 1, 2018.
- Naresh, Y. G., and H. S. Nagendraswamy, "Classification of medicinal plants: an approach using modified LBP with symbolic representation", Neurocomputing 173, pp: 1789-1797, 2016.

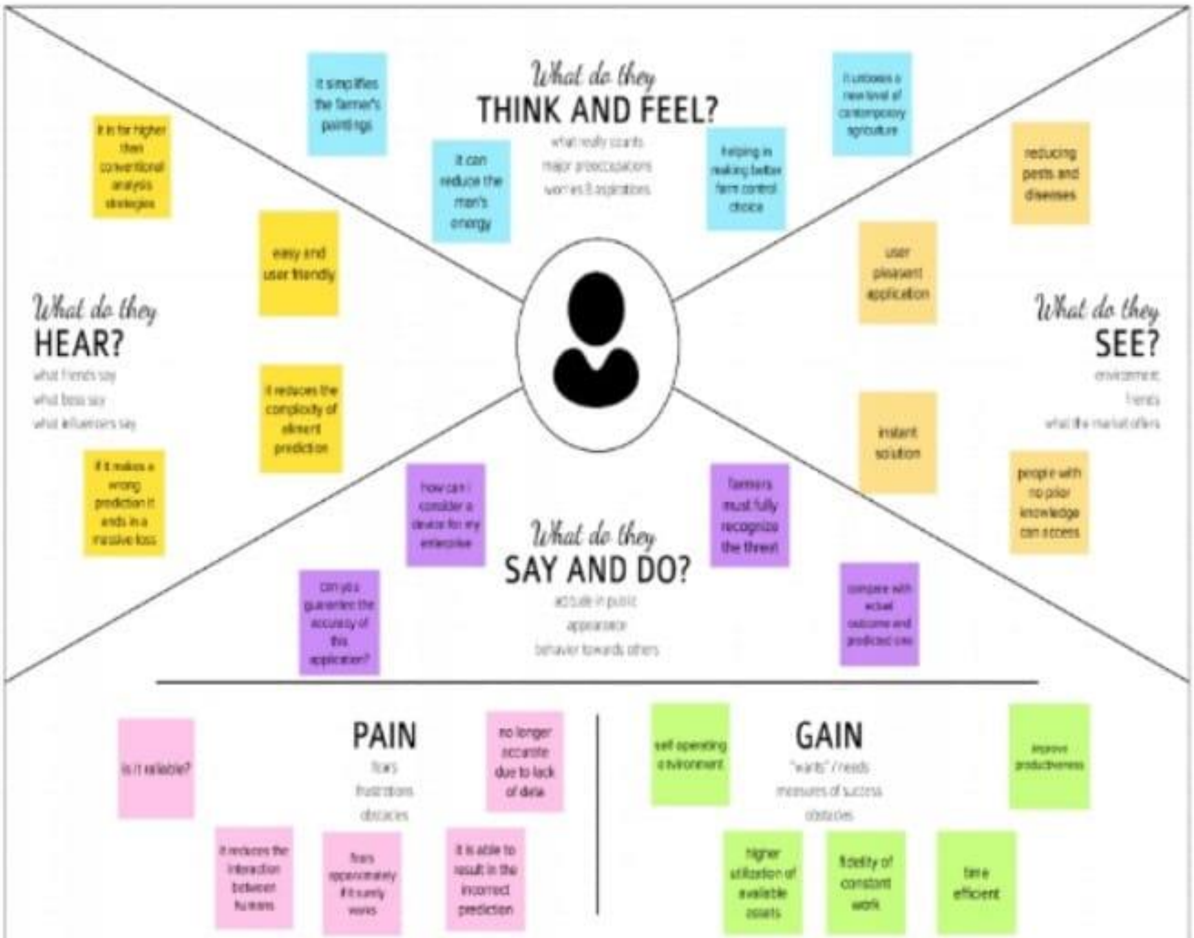
2.3 Problem Statement Definition :

- The solution to the problem is Machine learning, which is one of the applications of Artificial Intelligence, is being used to implement the proposed system. Crop recommendation is going to recommend you the best crop you can grow in your land as per the soil nutrition value and along with as per the climate in that region. And recommending the best fertilizer for every particular crop is also a challenging task. And the other and most important issue is when a plant gets caught by heterogeneous diseases that effect on less amount of agriculture production and compromises with quality as well. To overcome all these issues this recommendation has been proposed .

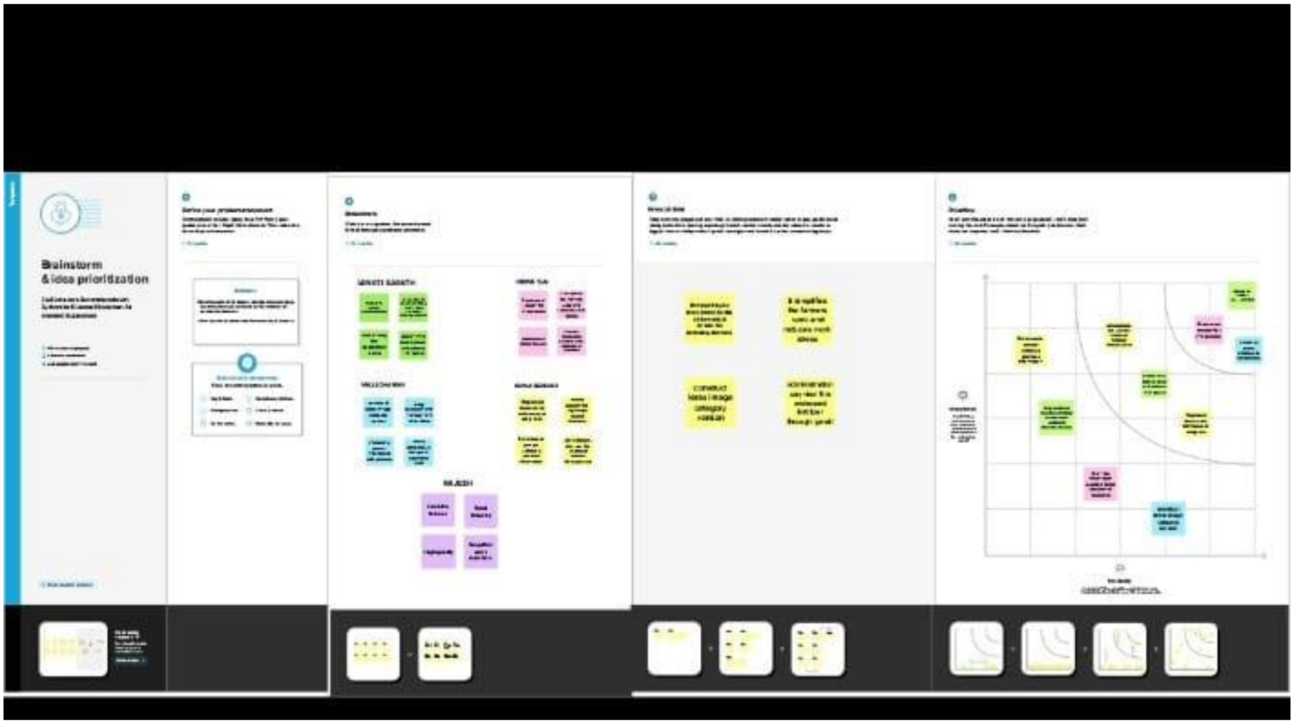
- Nowadays a lot of research and work is being implemented in the smart and modern agriculture domain. Crop recommendation is characterized by a soil database comprised of Nitrogen, Phosphorus, potassium. The ensembles technique is used to build a recommendation model that combines the prediction of multiple machine learning. Models to recommend the right crop based on soil value and the best fertilizer to use.

3.IDEATION & PROPOSED SOLUTION :

3.1 Empathy Map Canvas :



3.2 Ideation & Brainstorming :



3.3 Proposed Solution :

- The idea of the proposed solution uses Deep learning and Machine algorithm to classify leaves and identify the diseases and siggest the fertilizers. The deep learning process includes the MobileNetV2 and VGG19 training Models.

- Based on the leaf disease detected , the model recommendation for fertilizers for the prevention. The farmers and researchers are the end users get benefied by the system.

- More accurate in others. The system is more robust corporating more image data sets with wider variations. This system also estimates the probability of infected plant.
- Plant growth can be enhanced. Ensure plants are getting supplied with every nutrient they need also and multiple cross in grow in every yields for every season. It also helps people's nutritional needs.

3.4 Problem Solution Fit

Project Title: Fertilizer Recommendation System For Disease Prediction

Project Design Phase-I - Solution Fit

Team ID: PNT2022TMID24348

Define CS, fit into CC

1.Customer Segment: CS

This is used for farmer. It helps farmer to identify the disease by themself and recommand fertilizers by themselves.

5. Available solution :

People are judge the disease in plants by identifying through the change of leaves quality

8.Channels Of Behaviour: AS

Online :

Basic knowledge on the plant and fertilizer

Offline :

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

Explore AS, differentiate

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

2.Jobs To Be Done /Problems J&P

This application helps the farmer who needs recommendation of fertilizer on the plant disease. Identifying disease is the major problem

6.customer constraints : RC

Availability of good networks. Capturing the image in a required pixels to get a accurate prediction of disease in the plant.

9. Problem root cause : BE

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.

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

3.Triggers : TR

Seeing their crops are being infected by disease and facing huge loss in quantity and quality

4.Emotion : EM

Before : losing self-confidence ,distress

After : gaining self-confidence ,relief

7.Behaviour : SL

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 : CH

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 Requirements

Following are the functional requirements of the proposed solution .

Fr.no	Functional requirement	Sub requirement (story/subtask)
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 .

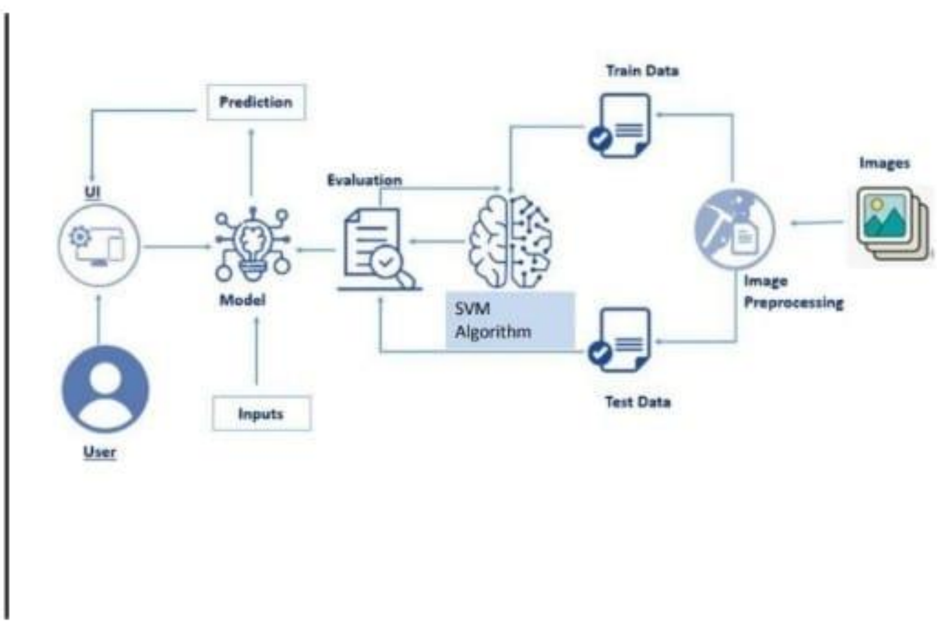
4.2 Non Functional Requirements

Following are the non-functional requirement of the proposed solution

NFr.no	Non-functional requirement	Description
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 :

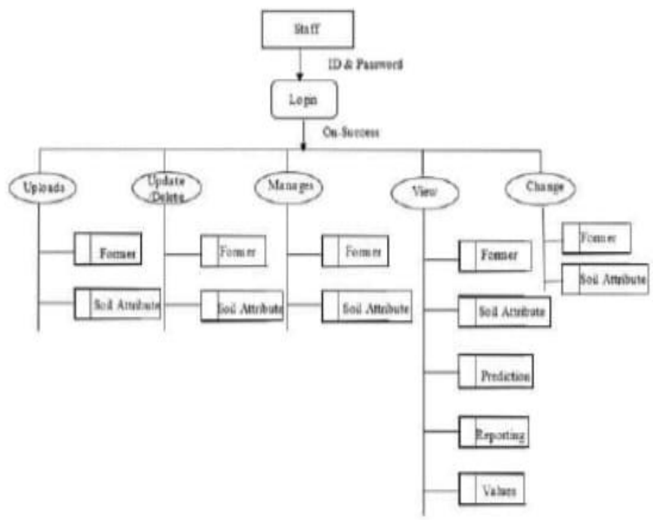
5.1 Solution & Technical Architecture



5.2 Data Flow Daigrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Example:



5.3 User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Google	I can register & access the dashboard with Google Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard					
Customer (Web user)		USN-6	As a User can view the dash board ,and this dashboard include the check roles of access and then move to the manage modules.	I can view the dashboard in this fertilizer recommendation system for disease prediction	Medium	
Customer Care Executive		USN-7				
Administrator			As a user once view the manage modules this describes the Manage system Admins and Manage Roles of User and etc..			

6.PROJECT PLANNING & SCHEDULING :

6.1 Sprint Planning and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points (Total)	Priority	Team Members
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	High	Guna sekhar, Sarath, Charan, Rajesh, Hema sai
	Model Creation and Training (Vegetables)		Create a model which can classify diseased vegetable plants from given images	2	High	Guna sekhar, Sarath, Charan, Rajesh, Hema sai

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points (Total)	Priority	Team Members
Sprint-2	Model Creation and Training (Vegetables)		Create a model which can classify diseased vegetable plants from given images and train on IBM Cloud	6	High	Guna sekhar, Sarath, Charan, Rajesh, Hema sai.
	Registration	USN-1	As a user, I can register by entering my email, password, and confirming my password or via OAuth API	3	Medium	Guna sekhar, Sarath, Charan, Rajesh, Hema sai.
	Upload page	USN-2	As a user, I will be redirected to a page where I can upload my pictures of crops	4	High	Guna sekhar, Sarath, Charan, Rajesh, Hema sai.
	Suggestion results	USN-3	As a user, I can view the results and then obtain the suggestions provided by the ML model	4	High	Guna sekhar, Sarath, Charan, Rajesh, Hema sai.
	Base Flask App		A base Flask web app must be created as an interface for the ML model	2	High	Guna sekhar, Sarath, Charan, Rajesh, Hema sai.
Sprint-3	Login	USN-4	As a user/admin/shopkeeper, I can log into the application by entering email & password	2	High	Guna sekhar, Sarath, Charan, Rajesh, Hema sai.
	User Dashboard	USN-5	As a user, I can view the previous results and history	3	Medium	Guna sekhar, Sarath, Charan, Rajesh, Hema sai.

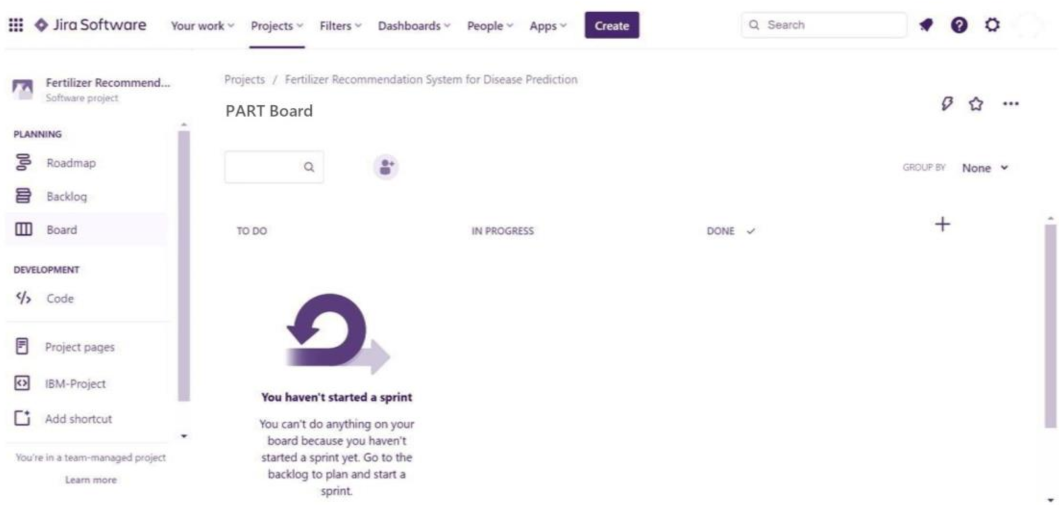
	Integration		Integrate Flask, CNN model with Cloudant DB	5	Medium	Guna sekhar, Sarath, Charan, Rajesh, Hema sai.
	Containerization		Containerize Flask app using Docker	2	Low	Guna sekhar, Sarath, Charan, Rajesh, Hema sai
Sprint-4	Dashboard (Admin)	USN-6	As an admin, I can view other user details and uploads for other purposes	2	Medium	Guna sekhar, Sarath, Charan, Rajesh, Hema sai
	Dashboard (Shopkeeper)	USN-7	As a shopkeeper, I can enter fertilizer products and then update the details if any	2	Low	Guna sekhar, Sarath, Charan, Rajesh, Hema sai
	Containerization		Create and deploy Helm charts using Docker Image made before	2	Low	Guna sekhar, Sarath, Charan, Rajesh, Hema sai

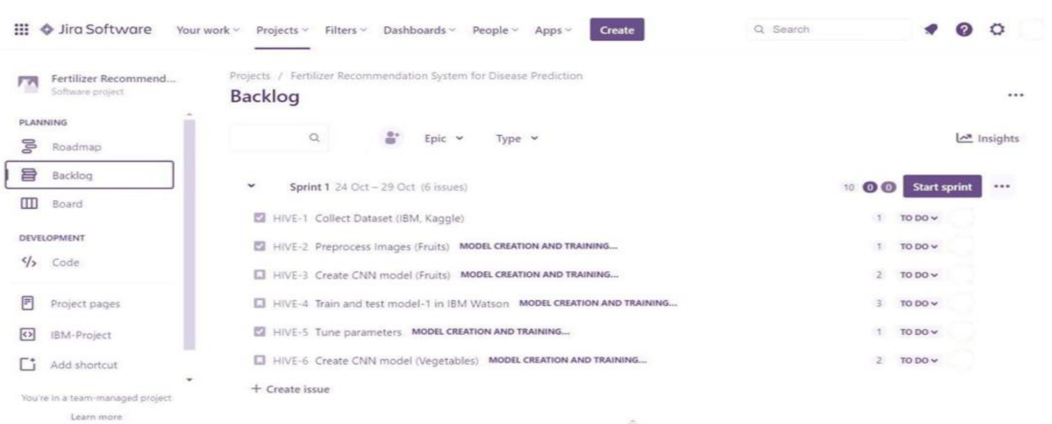
6.2 Sprint Delivery Schedule

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

NOTE: Burndown charts, Velocity to be updated dynamically after end of sprints **Roadmap:**

6.3 Reports from JIRA





7.CODING & SOLUTIONING:

7.1 Python – App.py :

```
import numpy as np

import pandas as pd

from utils.disease import disease_dic

from utils.fertilizer import fertilizer_dic

import requests

import config

import pickle

import io

import torch

from torchvision import transforms

from PIL import Image

from utils.model import ResNet9

import os

disease_classes = ['Apple___Apple_scab',

        'Apple___Black_rot',

        'Apple___Cedar_apple_rust',

        'Apple___healthy',

        'Blueberry___healthy',

        'Cherry_(including_sour)___Powdery_mildew',

        'Cherry_(including_sour)___healthy',

        'Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot',

        'Corn_(maize)__Common_rust',

        'Corn_(maize)___Northern_Leaf_Blight',

        'Corn_(maize)___healthy',

        'Grape___Black_rot',
```

```
            'Grape__Esca(Black_Measles)',
            'Grape__Leaf_blight(Isariopsis_Leaf_Spot)',
            'Grape___healthy',
            'Orange__Haunglongbing(Citrus_greening)',
            'Peach___Bacterial_spot',
            'Peach___healthy',
            'Pepper,bell__Bacterial_spot',
            'Pepper,bell__healthy',
            'Potato___Early_blight',
            'Potato___Late_blight',
            'Potato___healthy',
            'Raspberry___healthy',
            'Soybean___healthy',
            'Squash___Powdery_mildew',
            'Strawberry___Leaf_scorch',
            'Strawberry___healthy',
            'Tomato___Bacterial_spot',
            'Tomato___Early_blight',
            'Tomato___Late_blight',
            'Tomato___Leaf_Mold',
            'Tomato___Septoria_leaf_spot',
            'Tomato___Spider_mites Two-spotted_spider_mite',
            'Tomato___Target_Spot',
            'Tomato___Tomato_Yellow_Leaf_Curl_Virus',
            'Tomato___Tomato_mosaic_virus',
            'Tomato___healthy']
disease_model_path = 'models/plant_disease_model.pth'
```

```
disease_model = ResNet9(3, len(disease_classes))
disease_model.load_state_dict(torch.load(
    disease_model_path, map_location=torch.device('cpu')))
disease_model.eval()
crop_recommendation_model_path = 'models/RandomForest.pkl'
crop_recommendation_model = pickle.load(
    open(crop_recommendation_model_path, 'rb'))
def weather_fetch(city_name):
    api_key = config.weather_api_key
    base_url = "http://api.openweathermap.org/data/2.5/weather?"

    complete_url = base_url + "appid=" + api_key + "&q=" + city_name
    response = requests.get(complete_url)
    x = response.json()

    if x["cod"] != "404":
        y = x["main"]
        temperature = round((y["temp"] - 273.15), 2)
        return temperature
    else:
            return None
def predict_image(img, model=disease_model):
    transform = transforms.Compose([
        transforms.Resize(256),
        transforms.ToTensor(),
    ])
    image = Image.open(io.BytesIO(img))
```

```
    img_t = transform(image)
    img_u = torch.unsqueeze(img_t, 0)

    # Get predictions from model
    yb = model(img_u)
    # Pick index with highest probability
    _, preds = torch.max(yb, dim=1)
    prediction = disease_classes[preds[0].item()]
    # Retrieve the class label
    return prediction
app=Flask(name)
@ app.route('/crop-predict', methods=['POST'])
def crop_prediction():
    title = 'Harvestify - Crop Recommendation'
    if request.method == 'POST':
        N = int(request.form['nitrogen'])
        P = int(request.form['phosphorous'])
        K = int(request.form['pottasium'])
        ph = float(request.form['ph'])
        rainfall = float(request.form['rainfall'])

        # state = request.form.get("stt")
        city = request.form.get("city")

        if weather_fetch(city) != None:
            temperature, humidity = weather_fetch(city)
            data = np.array([[N, P, K, temperature, humidity, ph, rainfall]])
```

```
            my_prediction = crop_recommendation_model.predict(data)
            final_prediction = my_prediction[0]
            return render_template('crop-result.html', prediction=final_prediction, title=title)
        else:
            return render_template('try_again.html', title=title)
@ app.route('/fertilizer-predict', methods=['POST'])
def fert_recommend():
    title = 'Harvestify - Fertilizer Suggestion'
    crop_name = str(request.form['cropname'])
    N = int(request.form['nitrogen'])
    P = int(request.form['phosphorous'])
    K = int(request.form['pottasium'])
    # ph = float(request.form['ph'])
    df = pd.read_csv('Data/fertilizer.csv')
    nr = df[df['Crop'] == crop_name]['N'].iloc[0]
    pr = df[df['Crop'] == crop_name]['P'].iloc[0]
    kr = df[df['Crop'] == crop_name]['K'].iloc[0]
    n = nr - N
    p = pr - P
    k = kr - K
    temp = {abs(n): "N", abs(p): "P", abs(k): "K"}
    max_value = temp[max(temp.keys())]
    if max_value == "N":
        if n < 0:
            key = 'NHigh'
        else:
            key = "Nlow"
```

```
    elif max_value == "P":
        if p < 0:
            key = 'PHigh'
        else:
            key = "Plow"
    else:
        if k < 0:
            key = 'KHigh'
        else:
            key = "Klow"
            response = Markup(str(fertilizer_dic[key]))
            return render_template('fertilizer-result.html', recommendation=response, title=title)
@app.route('/disease-predict', methods=['GET', 'POST'])
def upload():
    if request.method=='POST':
        f=request.files['image']
        basepath=os.path.dirname(file)
        filepath=os.path.join(basepath,'uploads',f.filename)
        f.save(filepath)
        print('File Save')
        img=image.load_img(filepath,target_size=(128,128))
        x=image.img_to_array(img)
        print('Image to gray')
        x=np.expand_dims(x,axis=0)
        plant=request.form['plant']
        if(plant=='vegetable'):
            model=load_model("vegitable.h5")
```

```
        y=np.argmax(model.predict(x),axis=1)
        df=pd.read_excel('precautions_veg.xlsx')
    if(plant=='fruit'):
        model=load_model('fruit.h5')
        y=np.argmax(model.predict(x),axis=1)
        df=pd.read_excel('precautions_fruits.xlsx')
    return df.iloc[y[0]]['caution']
if name=='main':
  temp.run(debug=False)
```

7.2 Feature 1 :

Home.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: 0;
    padding: 0;
    }
    .container{
      padding: 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;

W   }
#h1{
    font-size: 50pt;
}
.menu{

    background-color:black;



}
#abc{
    color: white;
}
```

```
    </style>
</head>
<body><div class="menu">
    <ul>
                   
           
           
           
             &
nbsp;            &nb
sp;             
;       

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

7.3 Feature 2:

Predict.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>predict</title>
</head>

<style>
    .container{
        display: flex;
        padding: 60px 70px 60px 70px;
    }
    .card{
        padding: 70px 80px 70px 80px;
```

```
    }
    .menu{
        padding: 10px 10px 10px 10px;
        background-color: black;
        color: white;
        font-size: 15pt;
    }
</style>
<body>
    <div class="menu">
        <ul ><li>Plant disease Prediction</li></ul></div>
    <div class="container">

        <img src="img1.jpg">
        <div class="card">
        <form>
            <h1>Drop in the image to get the Prediction </h1><br><br>
            <label><select name="Fruit" id="plant">
                <option value="fruit" id="fruit">Fruit</option>
                <option value="vagitable" id="vig">vegitable</option>
                </select>
            </label><br><br><br>
            <input  id="default-btn" type="file" name=""
onchange="document.getElementById('output').src=window.URL.createObjectURL(this.files[0])
"><br><br><br>
            <img src="" id="output">

            <button id="button" onclick ="display()" >Predict!</button><br><br>
```

</form>

</body>

</html>

8.TESTING

8.1 TEST CASES:

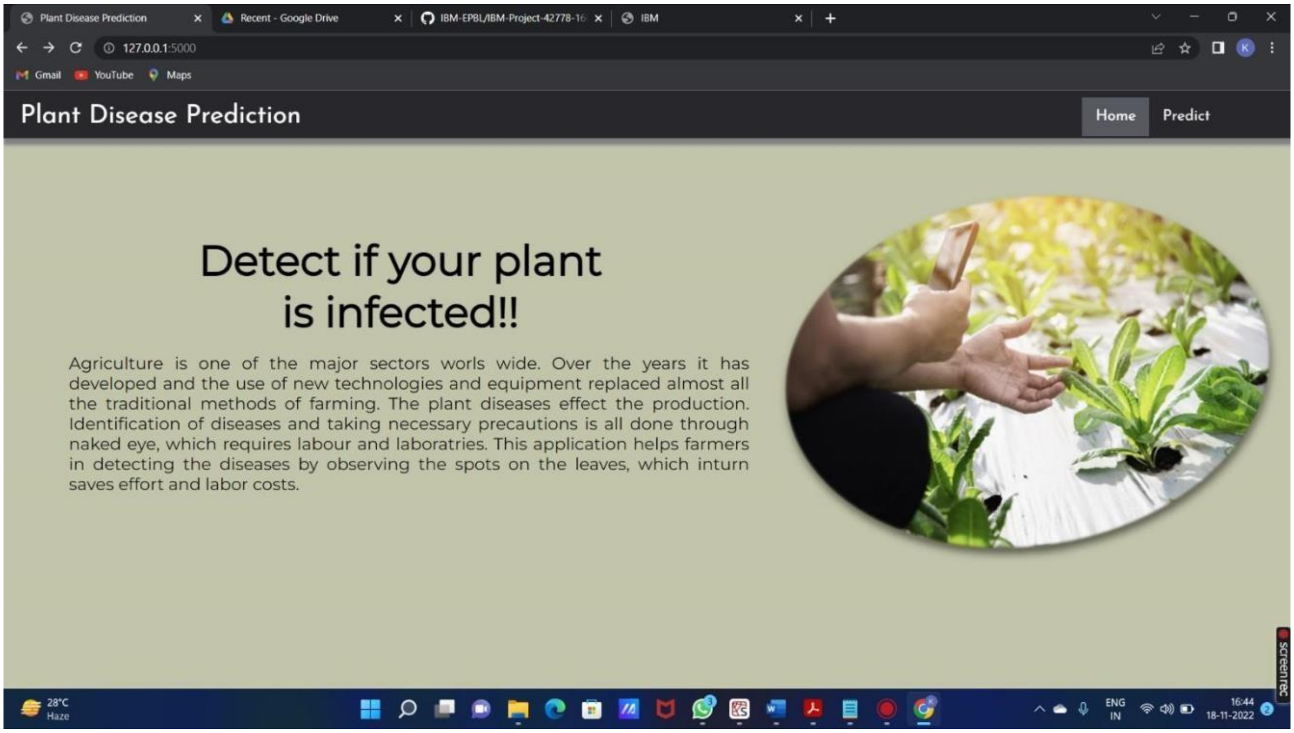
SECTION	TOTAL CASES	NOT TESTED	FAIL	PASS
Leaf spots	17	0	0	17
Mosaic Leaf Pattern	51	0	0	51
Misshapen Leaves	20	0	0	20
Yellow Leaves	7	0	0	7
Fruit Rots	9	0	0	9
Fruit Spots	4	0	0	4
Blights	2	0	0	2

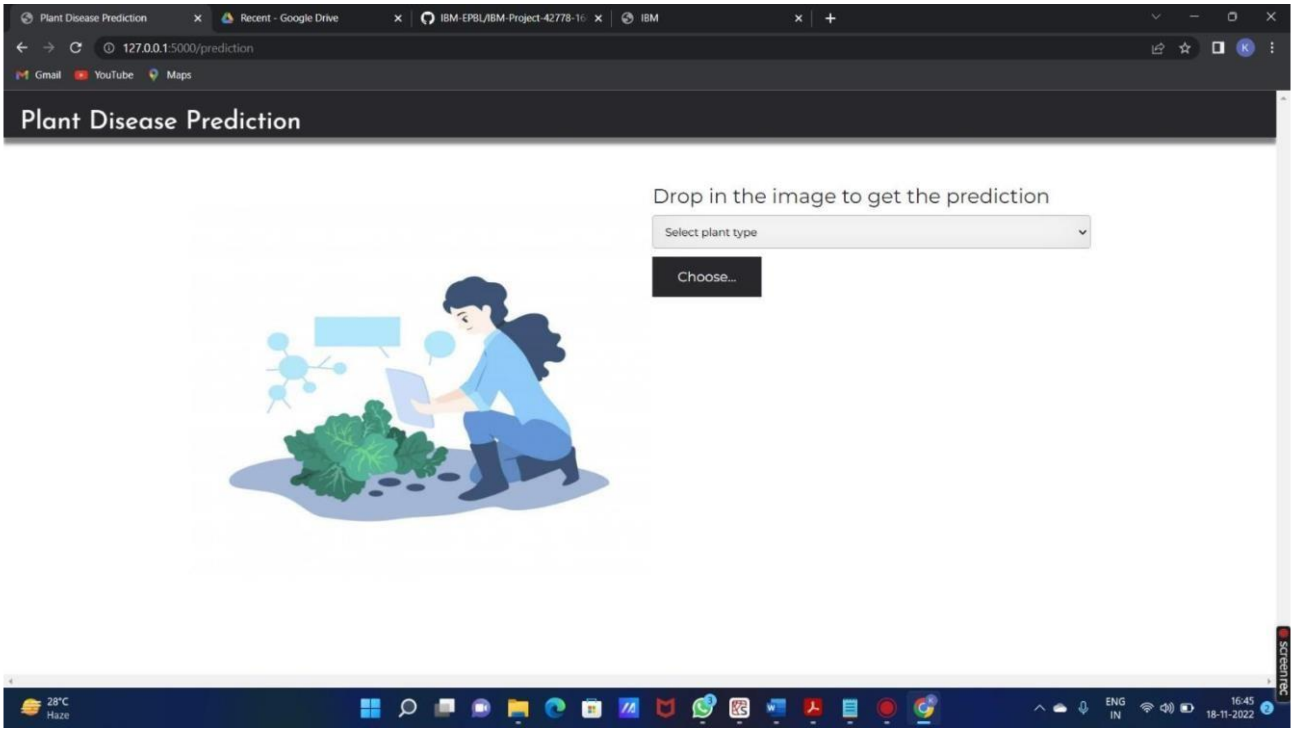
8.2 User Acceptance:

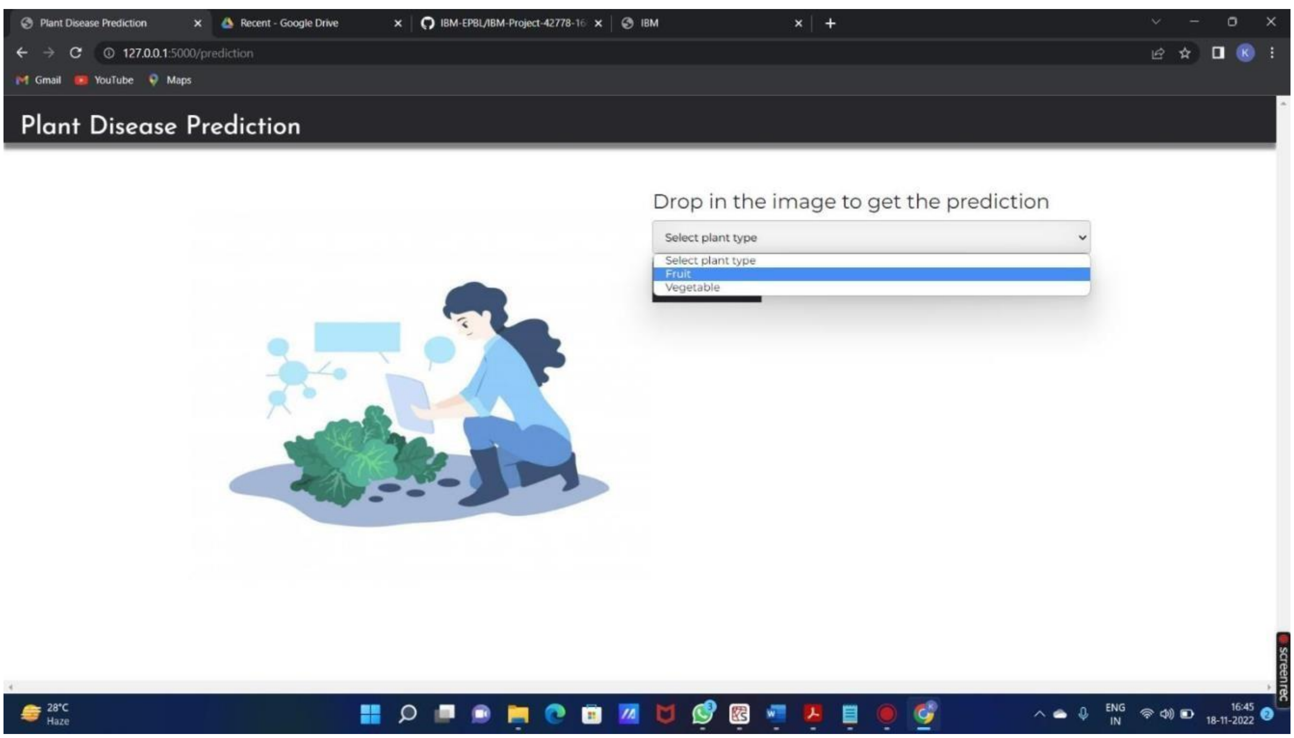
RESOLUTION	SEVERITY 1	SEVERITY 2	SEVERITY 3	SEVERITY 4	SUBTOTAL
Leaf spots	10	4	2	3	19
Mosaic Leaf Pattern	9	6	3	6	24
Misshapen Leaves	2	7	0	1	10
Yellow Leaves	11	4	3	20	38
Fruit Rots	3	2	1	0	6
Fruit Spots	5	3	1	1	10
Blights	4	5	2	1	12
Totals	44	31	13	32	119

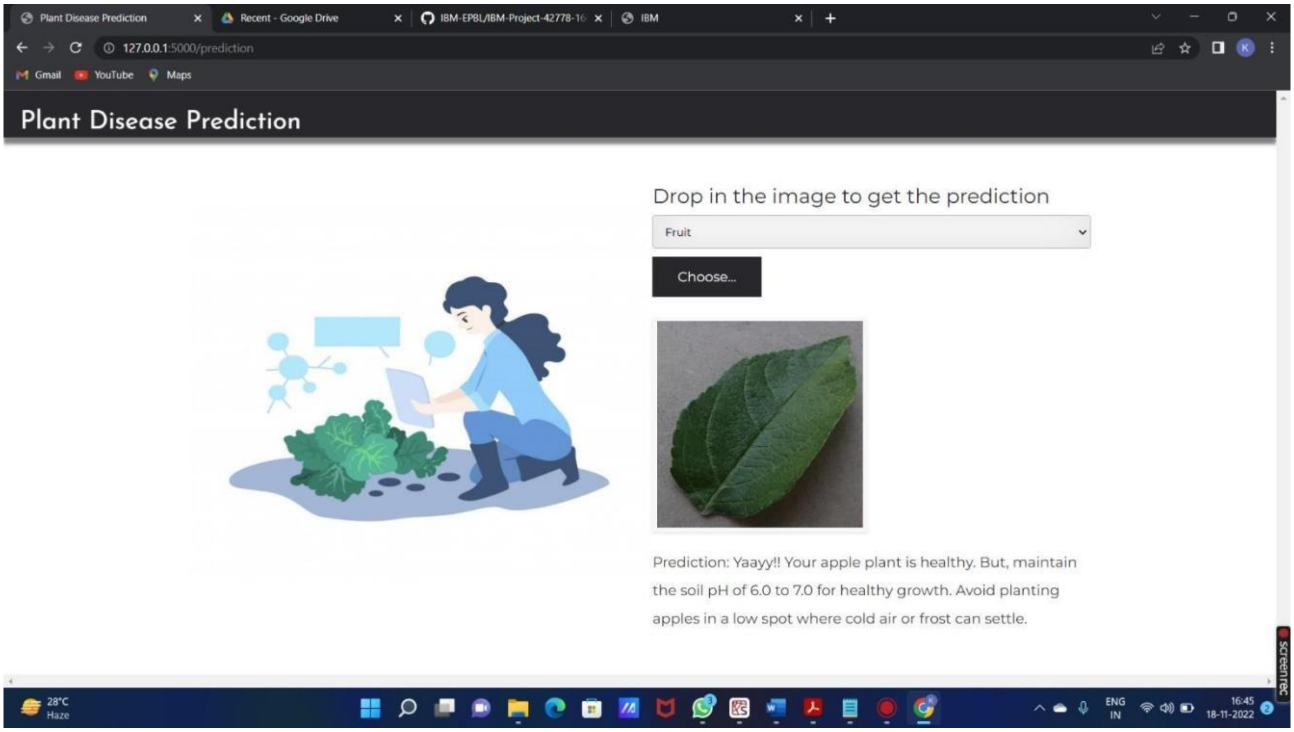
9.RESULTS

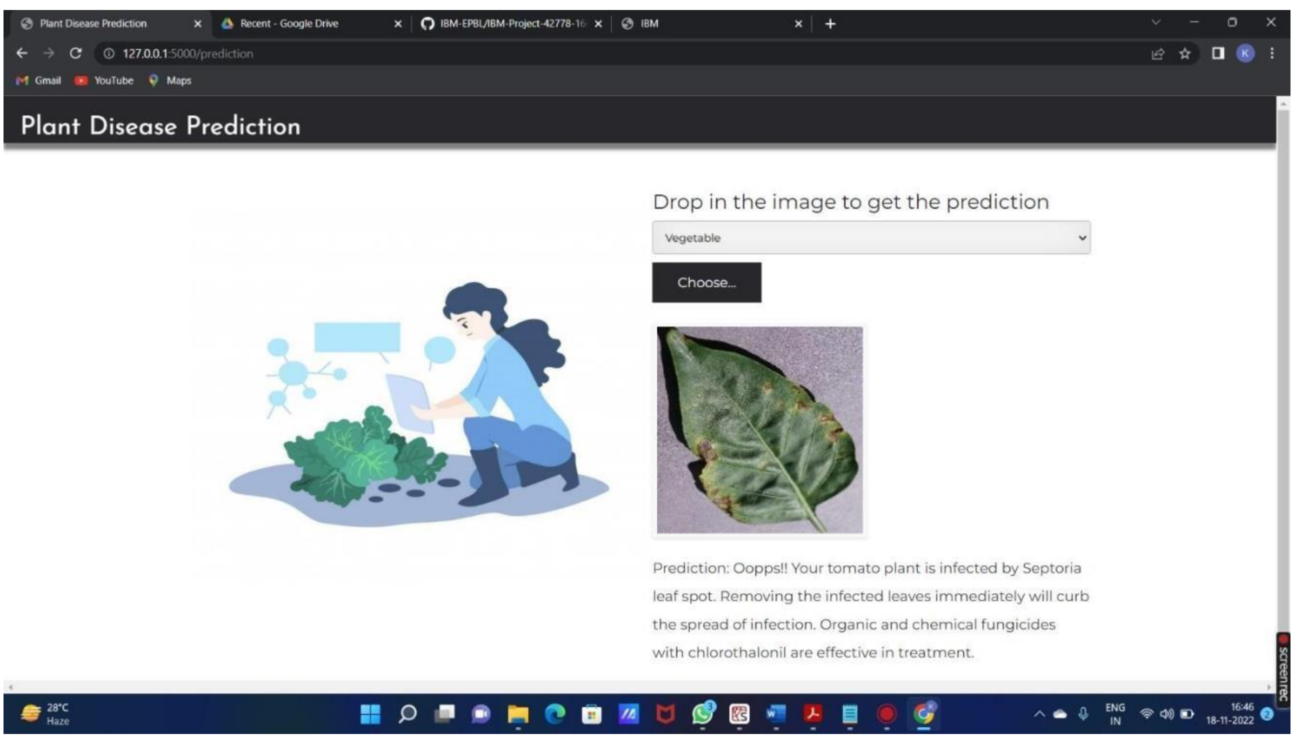
9.1 Performance Metrics




Plant Disease Prediction
Recent - Google Drive
IBM-EPBL/IBM-Project-42778-16
IBM
127.0.0.1:5000/prediction
Gmail
YouTube
Maps
Plant Disease Prediction
Drop in the image to get the prediction
Select plant type
Choose...
28°C
Haze
ENG
IN
16:45
18-11-2022


Plant Disease Prediction
Recent - Google Drive
IBM-EPBL/IBM-Project-42778-16
IBM
127.0.0.1:5000/prediction
Gmail
YouTube
Maps
Plant Disease Prediction
Drop in the image to get the prediction
Select plant type
Select plant type
Fruit
Vegetable
28°C
Haze
ENG
IN
16:45
18-11-2022


Plant Disease Prediction
Recent - Google Drive
IBM
127.0.0.1:5000/prediction
Gmail
YouTube
Maps
Plant Disease Prediction
Drop in the image to get the prediction
Fruit
Choose...
Prediction: Yaayy!! Your apple plant is healthy. But, maintain the soil pH of 6.0 to 7.0 for healthy growth. Avoid planting apples in a low spot where cold air or frost can settle.
28°C
Haze
ENG
IN
16:45
18-11-2022


Plant Disease Prediction
Recent - Google Drive
IBM
127.0.0.1:5000/prediction
Gmail
YouTube
Maps
Plant Disease Prediction
Drop in the image to get the prediction
Vegetable
Choose...
Prediction: Oopps!! Your tomato plant is infected by Septoria leaf spot. Removing the infected leaves immediately will curb the spread of infection. Organic and chemical fungicides with chlorothalonil are effective in treatment.
28°C
Haze
ENG
IN
16:46
18-11-2022

10.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.

11.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.

12.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.

13.FUTURE SCOPE :

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

14.APPENDIX

Source Code

(Jupyter notebook python code) fruit.ipynb (due to limited page size the code vegetable.ipynb uploaded in github)

```
#!/usr/bin/env python
# coding: utf-8
# In[1]: pwd
#    In[2]: cd E:/IBM_MY_COURSE/Project/Dataset     Plant    Disease/fruitdataset/fruit-
     dataset

# # Apply ImageDataGenerator functionality to Train and Test set
  # # Preprocessing   #

 In[3]: from keras.preprocessing.image

 import
```

ImageDataGenerator train_datagen =

ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizonta l_fli

p=True) test_datagen = ImageDataGenerator(rescale=1) # In[4]: pwd

In[5]: x_train = train_datagen.flow_from_directory('E:/IBM_MY_COURSE/Project/Dataset

Plant Disease/fruit-
dataset/fruitdataset/train',target_size=(128,128),batch_size=32,class_mode='cate gorical')

In[6]: x_test=test_datagen.flow_from_directory('E:/IBM_MY_COURSE/Project/Datas

et Plant Disease/fruit-dataset/fruit-dataset/test',target_size=(128,128),

batch_size=32,class_mode='categorical') # # Import the models

In[7]: from tensorflow.keras.models import Sequential from

tensorflow.keras.layers import Dense,Convolution2D,MaxPool2D,Flatten

Initializing the models 10
In[8]: model=Sequential()
Add CNN Layers
In[9]: model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

In[10]: x_train.class_indices
Add Pooling layer
In[11]: model.add(MaxPool2D(pool_size=(2,2)))
Add Flatten layer # In[12]: model.add(Flatten())
Add Dense Layer
In[21]: model.add(Dense(40, kernel_initializer='uniform',activation='relu'))

model.add(Dense(20, kernel_initializer='random_uniform',activation='relu'))

Add Output Layer # In[24]: model.add(Dense(6,activation='softmax',

kernel_initializer='random_uniform'))

Compile the model

In[25]: model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accur acy'
])

In[26]: len(x_train)

In[27]: 5384/32
Fit the Model
#In[28]:
model.fit_generator(x_train,steps_per_epoch=168,validation_data=x_test,validat
ion_st eps=52,epochs=3)

Save the Model
In[29]: model.save("fruit.h5")
In[30]: ls
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("fruit.h5")
Test Apple_Healthy Class images
In[37]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant
Disease/fruitdataset/fruit-dataset/test/Apple___healthy/00fca0da-2db3-
481bb98a9b67bb7b105c___RS_HL 7708.JPG',target_size=(128,128)) 11

In[39]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0)
In[40]: pred = model.predict_classes(x)
In[41]: pred
In[45]: index
=['Apple___Black_rot','Apple___healthy','Corn_(maize)___Northern_Leaf_Blig ht','Corn_(
maize)___healthy','Peach___Bacterial_spot','Peach___healthy']

In[46]: print('the given image belogs to=',index[pred[0]])
Test Apple Black Rot class images # In[54]: img =
image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant
Disease/fruitdataset/fruit-dataset/test/Apple___Black_rot/0f3d45f4-e121-42cda5b6-
be2f866a0574___JR_FrgE.S 2870.JPG',target_size=(128,128))

In[55]: 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]])

Test Corn Northern leaf Blight class images

In[56]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruitdataset/fruitdataset/test/Corn_(maize)___Northern_Leaf_Blight/00a14441-7a62- 4034-bc40b196aeab2785___RS_NLB 3932.JPG',target_size=(128,128))

In[57]: 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]])

Test Corn Healthy class images # In[58]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruitdataset/fruit-dataset/test/Corn_(maize)___healthy/0a68ef5a-027c41ae-b227-159dae77d3dd___R.S_HL 7969 copy.jpg',target_size=(128,128))

In[59]: 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]]) # # Test Peach Bacterial spot class images # In[60]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruitdataset/fruit-dataset/test/Peach___Bacterial_spot/00ddc106-692e4c67-b2e8-569c924caf49___Rutg._Bact.S 1228.JPG',target_size=(128,128)) 12 # In[61]: 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]])

Test Peach Healthy class images

In[62]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruitdataset/fruit-dataset/test/Peach___healthy/1a07ce54-f4fd-41cfb088-144f6bf71859___Rutg._HL 3543.JPG',target_size=(128,128))

In[63]: 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]])

Github Link :- https://github.com/IBM-EPBL/IBM-Project-54007-1661587406

Demonstration Link:- **https://drive.google.com/file/d/13hJdeq0QtiT-NrE3QFPfR45rT-kwJgXq/view?usp=drivesdk**