PROJECT REPORT

Fertilizers Recommendation for Disease Prediction

Team ID: PNT2022TMID24340

Team Leader : Chennareddy Preethi - 113319104016

Team Member : Vakati Harshitha -113319104098

Team Member : Parvathareddy Jhansi - 113319104059

Team Member : Kamireddy Gnapika -113319104036

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.

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**.**

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**.**

LITREATURE SURVEY

Existing Problem

- Detection and recognition of plant diseases using machine learning 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. Application of computer vision and image processing strategies simply assist farmers in all of the regions of agriculture. Generally, the plant diseases are caused by the abnormal physiological functionalities of plants.
- Therefore, the characteristic symptoms are generated based on the differentiation between normal physiological functionalities and abnormal physiological functionalities of the plants. Mostly, the plant leaf diseases are caused by Pathogens which are positioned on the stems of the plants.
- These different symptoms and diseases of leaves are predicted by different methods in image processing. These different methods includedifferent fundamental processes like segmentation, feature extraction and classification and so on. Mostly, the prediction and diagnosis of leaf diseases are depending on the segmentation such as segmenting the healthy tissues from diseased tissues of leaves.

References :

[1] Semi-automatic leaf disease detection and classification system for soybean culture IET Image Processing, 2018

[2] Cloud Based Automated Irrigation And Plant Leaf Disease Detection System Using An Android Application. International Conference on Electronics, Communication and Aerospace Technology, ICECA 2017.

[3] Duan Yan-e, Design of Intelligent Agriculture Management Information System Based on IOT ‖, IEEE,4th, Fourth International reference on Intelligent Computation Technology and Automation, 2011 https://ieeexplore.ieee.org/document/5750779

[4] R. Neela, P. Fertilizers Recommendation System For Disease Prediction In Tree Leave International journal of scientific & technology research volume 8, issue 11, November 2019 http://www.ijstr.org/final-print/nov2019/Fertilizers-Recommendation-System-For-Disease-PredictionIn-Tree-Leave.pdf .

[5] Swapnil Jori1, Rutuja Bhalshankar2, Dipali Dhamale3, Sulochana Sonkamble , Healthy Farm: Leaf Disease Estimation and Fertilizer Recommendation System using Machine Learning, International Journal of All Research Education and Scientific Methods (IJARESM), ISSN: 2455-6211

Problem Statement Definition :

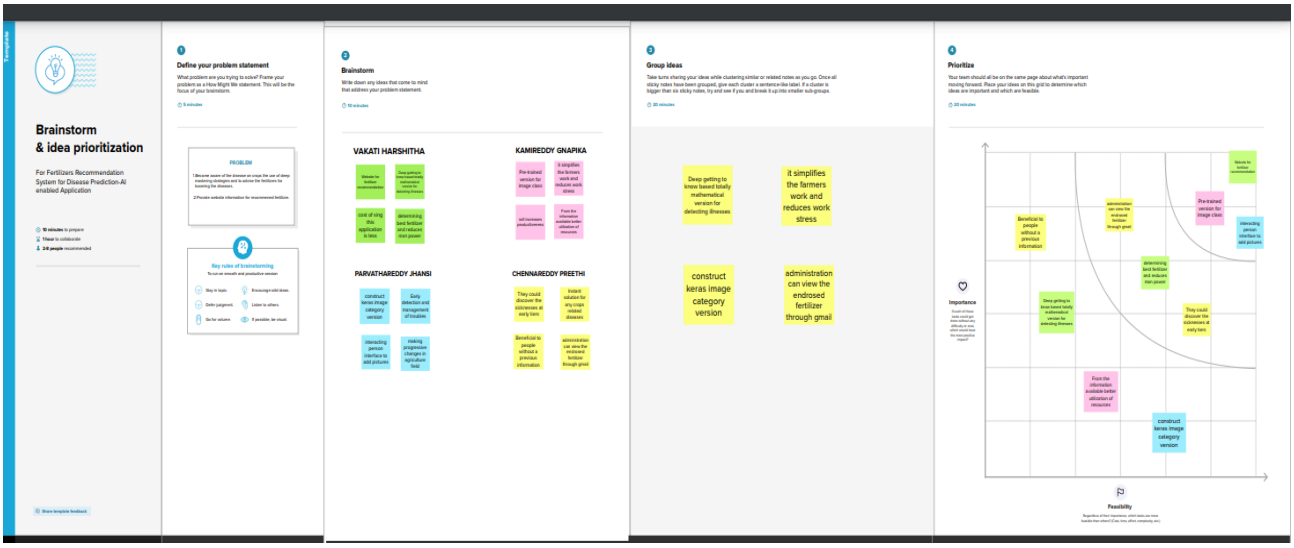
- The solution to the problem is Machine learning, which is one of the applications of Artificial Intelligence, is being used to implement the proposed system. Crop recommendation is going to recommend you the best crop you can grow in your land as per the soil nutrition value and along with as per the 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.

- **IDEATION & PROPOSED SOLUTION :**

Empathy Map Canvas :

Ideation & Brainstorming :



Proposed Solution :

- The idea of the proposed solution uses Deep learning and Machine algorithm to classify leaves and identify the diseases and suggest 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 benefited by the system.

- More accurate in others. The system is more robust corporation 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.

Problem Solution Fit

Project Title: Fertilizers Recommendation System For Disease Prediction **Project Design Phase-I - Solution Fit Template** **Team ID:** PNT2022TMID24340

Define CS, fit into CC

1.CUSTOMER SEGMENT(S) CS

Farmers are the customers

for the application.

6. CUSTOMER CONSTRAINTS CC

Good network connection is

required for the application to work

properly.

5. AVAILABLE SOLUTIONS AS

Farmers can use the fertilizers

based on the knowledge they have.

The prediction may not always be correct. Hence they may not get expected yield all the time.

Explore AS, differentiate

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

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

This application helps farmers by recommending the fertilizer to be used on the infected crop. The job is to identify the type of disease occurred in the crop.

9. PROBLEM ROOT CAUSE RC

Due to the lack of proper knowledge about the crop diseases and unaware of the amount and what kind of fertilizers need to be used to prevent the diseases .

7. BEHAVIOUR BE

Directly related:

Customers can get the proper fertilizers to be used for the disease prevention and does not require any knowledge on the fertilizers.

Indirectly related:

They need not wait for so long in order to get the recommendation

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

Identify strong TR & EM

3. TRIGGERS TR

As the application predicts the disease and offers the best solution by recommending the suitable fertilizer.

4. EMOTIONS: BEFORE / AFTER EM

Before: As they are not aware of the preventive methods, crop gets destroyed due to the disease and there is huge loss and there efforts gets wasted.

After: By taking the necessary steps in preventing the disease the crop yield will be high .

10. YOUR SOLUTION SL

Recommending the correct fertilizer to be used is the solution for preventing the disease in the plant. Our application suggests the good fertilizer for the disease.

8.CHANNELS OF BEHAVIOUR

Online: Providing online assistance to the farmer,regarding the knowledge about the application, and by giving more knowledge about AI and assistance about the fertilizer recommendation.

Offline: By conducting Awareness campus to teach the importance and advantages of the automation and AI which helps in the fertilizer recommending in the development of agriculture.

Identify strong TR & EM

REQUIREMENT ANALYSIS :

Functional Requirements

Functional requirement :

Following are the functional requirements of the proposed solution .

Fr.no	Functional requirement	Sub requirement (story/subtask)
Fr-1	User registration	Registration through form or Registration through Gmail
Fr-2	User confirmation	Confirmation via OTP or Confirmation via Email
Fr-3	Log in to system	Check credentials Confirmation via otp
Fr-4	Manage Modules	Manage System Admins Manage Roles of user Manage User Permission
Fr-5	Capturing image	Capture the image of the leaf And check the parameter of the captured image .
Fr-6	Image processing	Upload the image for the prediction of the disease in the leaf.
Fr-7	Leaf identification	Identify the leaf and predict the disease in leaf.

Non Functional Requirements

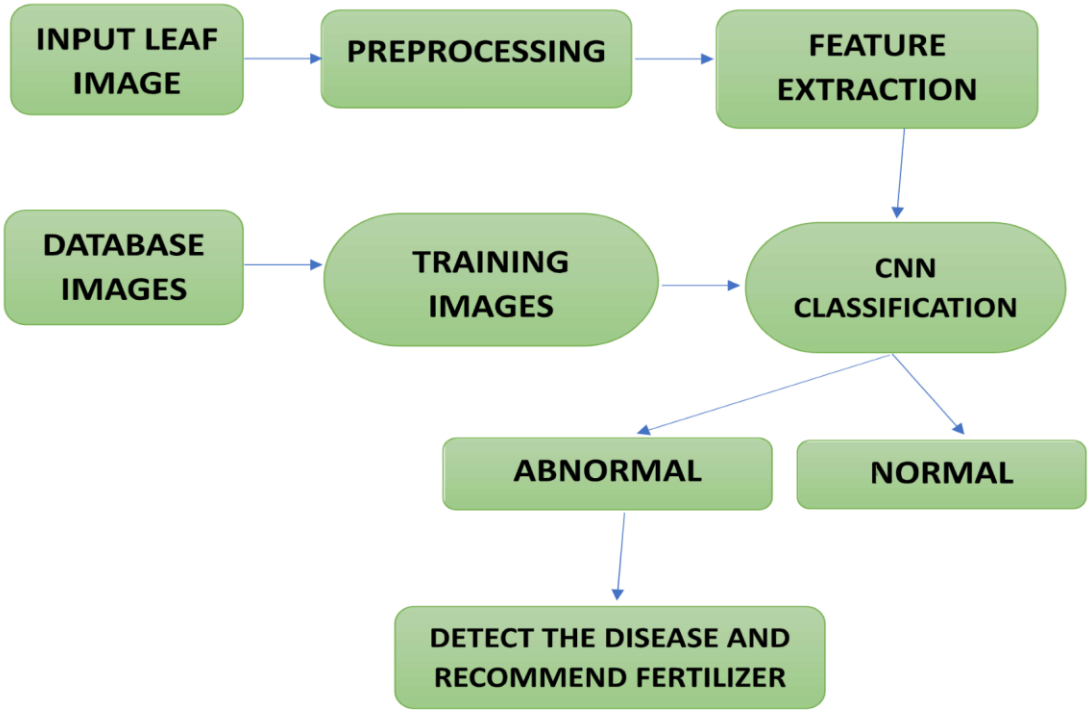
Non-functional requirement:

Following are the non-functional requirement of the proposed solution

Nfr.no	Non-functional requirement	Description
Nfr-1	Usability	• Usability covers features including ease of learning effectiveness in usage, memory retention. • Datasets of all the leaf is used to detecting the disease that present in the leaf.
Nfr-2	Security	• Private and sensitive information must be kept secure all the times. • The information belongs to the user and leaf are secured highly.
Nfr-3	Reliability	• A superior cost of reliability trade off is achieved with shared protection. • The leaf quality is important for the predicting the disease in leaf.
Nfr-4	Performance	• It will be more effective to monitor farming operations overall if integrated sensors are used analyze leaf condition.

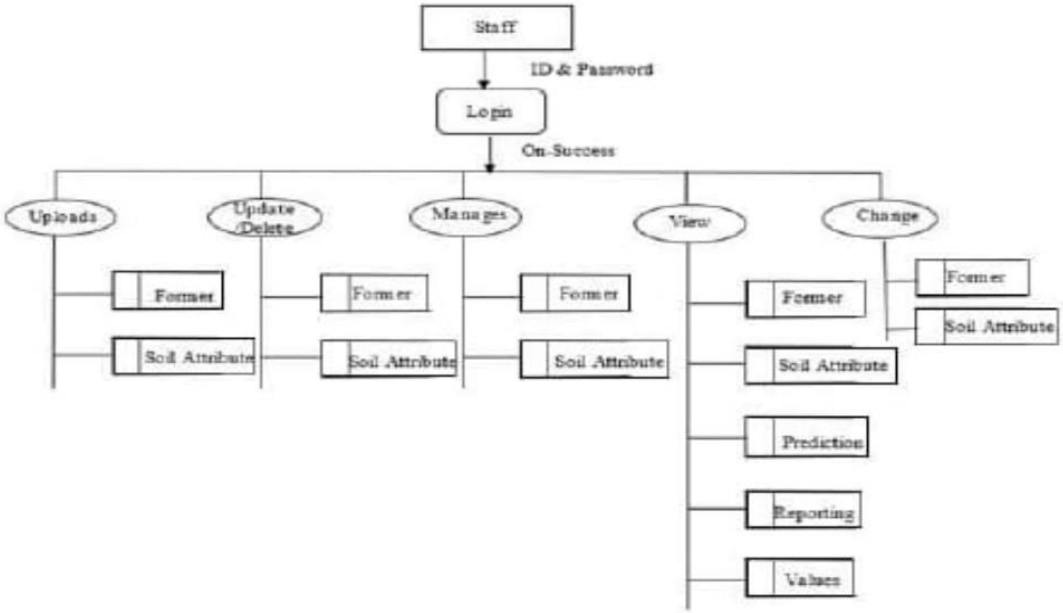
PROJECT DESIGN :

Solution & Technical Architecture



Data Flow Daigrams

Example:



User Stories

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..			

PROJECT PLANNING & SCHEDULING :

Sprint Planning and Estimation

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

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	VakatiHarshitha, ParvathareddyJhansi, Kamireddy Gnapika, Chennareddy Preethi
	Model Creation and Training (Vegetables)		Create a model which can classify diseased vegetable plants from given images	2	High	VakatiHarshitha, ParvathareddyJhansi, Kamireddy Gnapika, Chennareddy Preethi

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	VakatiHarshitha, ParvathareddyJhansi, Kamireddy Gnapika, Chennareddy Preethi
	Registration	USN-1	As a user, I can register by entering my email, password, and confirming my password or via OAuth API	3	Medium	VakatiHarshitha, ParvathareddyJhansi, Kamireddy Gnapika, Chennareddy Preethi
	Upload page	USN-2	As a user, I will be redirected to a page where I can upload my pictures of crops	4	High	VakatiHarshitha, ParvathareddyJhansi, Kamireddy Gnapika, Chennareddy Preethi
	Suggestion results	USN-3	As a user, I can view the results and then obtain the suggestions provided by the ML model	4	High	VakatiHarshitha, ParvathareddyJhansi, Kamireddy Gnapika, Chennareddy Preethi
	Base Flask App		A base Flask web app must be created as an interface for the ML model	2	High	VakatiHarshitha, ParvathareddyJhansi, Kamireddy Gnapika, Chennareddy Preethi
Sprint-3	Login	USN-4	As a user/admin/shopkeeper, I can log into the application by entering email & password	2	High	VakatiHarshitha, ParvathareddyJhansi, Kamireddy Gnapika, Chennareddy Preethi
	User Dashboard	USN-5	As a user, I can view the previous results and history	3	Medium	VakatiHarshitha, ParvathareddyJhansi, Kamireddy Gnapika, Chennareddy Preethi
	Integration		Integrate Flask, CNN model with Cloudant DB	5	Medium	VakatiHarshitha, ParvathareddyJhansi, Kamireddy Gnapika, Chennareddy Preethi
	Containerization		Containerize Flask app using Docker	2	Low	VakatiHarshitha, ParvathareddyJhansi, Kamireddy Gnapika, Chennareddy Preethi

Sprint-4	Dashboard (Admin)	USN-6	As an admin, I can view other user details and uploads for other purposes	2	Medium	VakatiHarshitha, ParvathareddyJhansi, Kamireddy Gnapika, Chennareddy Preethi
	Dashboard (Shopkeeper)	USN-7	As a shopkeeper, I can enter fertilizer products and then update the details if any	2	Low	VakatiHarshitha, ParvathareddyJhansi, Kamireddy Gnapika, Chennareddy Preethi
	Containerization		Create and deploy Helm charts using Docker Image made before	2	Low	VakatiHarshitha, ParvathareddyJhansi, Kamireddy Gnapika, Chennareddy Preethi

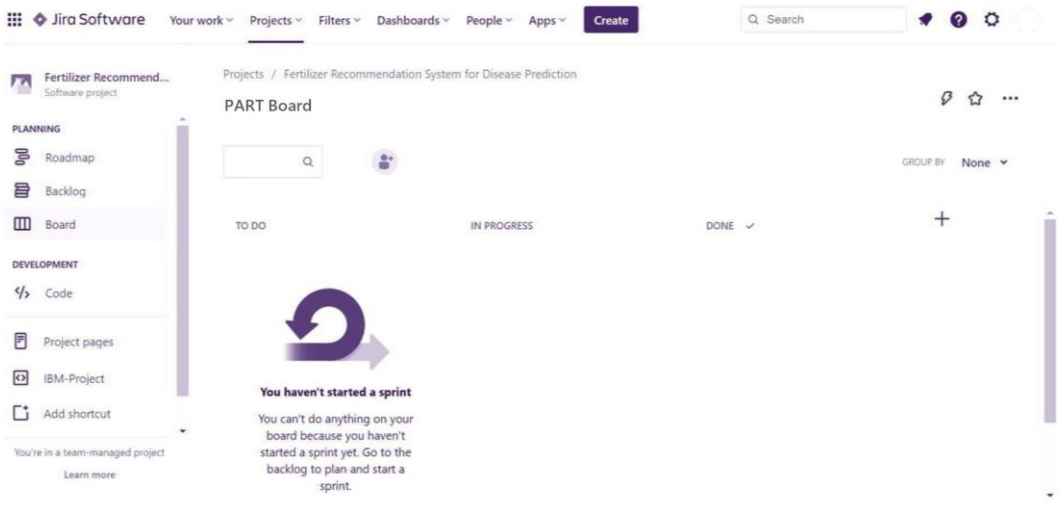
Sprint Delivery Schedule

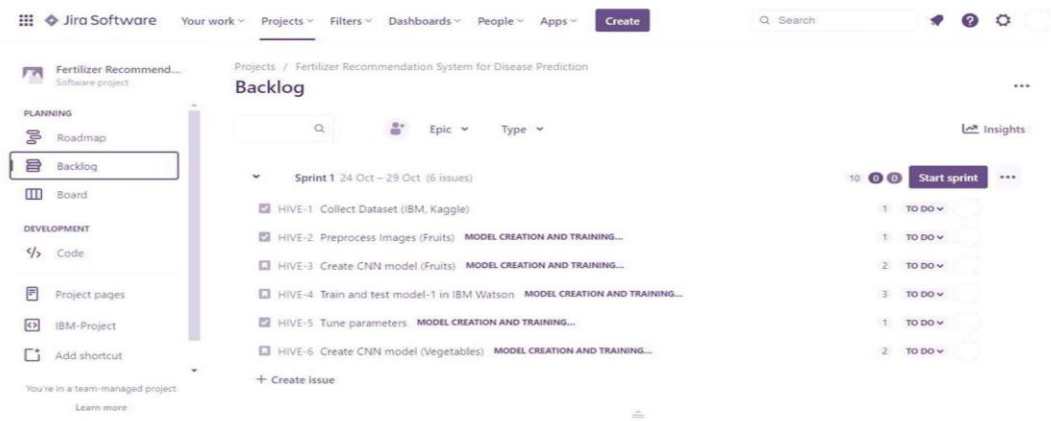
Project Tracker, Velocity & Burndown Chart: (4 Marks)

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:**

Reports from JIRA





CODING & SOLUTIONING:

Python-App.py:

```
from flask
import Flask,
render_template,
request, Markup
                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)
```

Feature 1:
Home.html:

```
<!
DO
CT
YP
E
ht
```

```
ml
>
      <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;

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

                  background-color:black;


              }
              #abc{
                  color: white;
```

```
        }

    </style>
</head>
<body><div class="menu">
    <ul>
                   
           
           
           
              &nbs
p;              &n
bsp;              
   

              &nbs
p;            
    <li id="abc"> plant Disease
Prediction           
           
           
              &nbs
p;              &n
bsp;              
              &nbs
p;              &n
bsp;              
              &nbs
p;              &n
bsp;              
              &nbs
p;              &n
bsp;              
     <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>
```

Feature 2:

Predict.html:

```
<!DOCTY
PE
html>
          <html lang="en">
          <head>
              <meta charset="UTF-8">
              <meta http-equiv="X-UA-Compatible" content="IE=edge">
              <meta name="viewport" content="width=device-width, initial-scale=1.0">
              <title>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.file
s[0])"><br><br><br>
            <img src="" id="output">

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



        </form>



</body>
</html>
```

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.

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.

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.

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.

APPENDIX

Source Code

```
 ls
In [ ]:
 pwd
In [ ]:
 from tensorflow.keras.preprocessing.image import ImageDataGenerator
In [ ]:
 train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,vertic
 al_flip=False)
In [ ]:
 test_datagen=ImageDataGenerator(rescale=1./255)
In [ ]:
 ls
In [ ]:
```

```
 x_train=train_datagen.flow_from_directory(r"C:\Users\Vakati_Harshitha\Desktop\FILES\data_fo
 r_ibm\Fertilizers_Recommendation_ System_For_Disease_ Prediction\Dataset Plant Disease\fruit-
 dataset\fruit-dataset\train",target_size=(128,128),
                                  class_mode='categorical',batch_size=24)
x_test=test_datagen.flow_from_directory(r"C:\Users\Vakati_Harshitha\Desktop\FILES\data_for_ib
m\Fertilizers_Recommendation_ System_For_Disease_ Prediction\Dataset Plant Disease\fruit-
dataset\fruit-dataset\test",target_size=(128,128),
                                 class_mode='categorical',batch_size=24)
 from tensorflow.keras.models import Sequential
 from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten
In [ ]:
 model=Sequential()
In [ ]:
 model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
In [ ]:
 model.add(MaxPooling2D(pool_size=(2,2)))
 model.add(Flatten())
 model.summary()
In [ ]:
 32*(3*3*3+1)
 model.add(Dense(300,activation='relu'))
 model.add(Dense(150,activation='relu'))
In [ ]:
 model.add(Dense(6,activation='softmax'))
 model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
 len(x_train)

 model.fit(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_tes
 t),epochs=10)
In [ ]:
 model.save('fruitdata.h5')
 import numpy as np
 from tensorflow.keras.models import load_model
 from tensorflow.keras.preprocessing import image
In [ ]:
 model=load_model('fruitdata.h5')
In [ ]:
 img=image.load_img(r"C:\Users\Vakati_Harshitha\Desktop\FILES\data_for_ibm\Fertilizers_Rec
 ommendation_ System_For_Disease_ Prediction\Dataset Plant Disease\fruit-dataset\fruit-
 dataset\test\Apple___healthy\00fca0da-2db3-481b-b98a-9b67bb7b105c___RS_HL 7708.jpg")
In [ ]:
 img

img=image.load_img(r"C:\Users\Vakati_Harshitha\Desktop\FILES\data_for_ibm\Fertilizers_Reco
mmendation_ System_For_Disease_ Prediction\Dataset Plant Disease\fruit-dataset\fruit-
```

```
dataset\test\Apple___healthy\00fca0da-2db3-481b-b98a-9b67bb7b105c___RS_HL
7708.jpg",target_size=(128,128))
img
 x=image.img_to_array(img)
In [ ]:
 X
 x=np.expand_dims(x,axis=0)
In [ ]:
 x

y=np.argmax(model.predict(x),axis=1)

x_train.class_indices

 index=['Apple___Black_rot','Apple___healthy','Corn_(maize)___Northern_Leaf_Blight','Corn_(
 maize)___healthy','Peach___Bacterial_spot','Peach___healthy']
In [ ]:
 index[y[0]]
```

Github Link: https://github.com/IBM-EPBL/IBM-Project-54039-1661587887.git

Project Demo Link:
https://drive.google.com/file/d/13bOOaORbSRsikjLwGShq5Atvln_Bvnfu/view?usp=drivesdk