Fertilizers Recommendation System for Disease Prediction

Team ID: PNT2022TMID08563

NAVEENPRASANTH S (727619BCS002)

SURENDAR M (727619BCS044)

VISHNU BALA GURU M (727619BCS056)

SREEDHARSON (727619BCS070)

INDEX

13. APPENDIX

Source Code

GitHub & Project Demo Link

1. INTRODUCTION

1.1 Project Overview

Agriculture is the most crucial one for the human existence. Crop disease prediction helps in enhancing the growth of agricultural crops and booming up the agriculture. Plant disease prediction helps in the detection and recognition of the plant diseases. The images of plants are captured and analyzed for certain symptoms using Computer vision and image processing. By identifying the disease, the deficit nutrients that lead to the disease are found. Based on the available data on fertilizers are recommended for curing the disease and also suggests some effective measures for increasing the crop yield. 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

The plant diseases may lead to the reduction in the yield of crops. The main objective of the project is to test the fruits and vegetables samples and detect the crop diseases effectively and suggesting the suitable fertilizers for diagnosing the disease. The project also aims at guiding the farmers with the proper choice of the fertilizers that are required to counter the deficiency of the nutrients that cause the disease.

2. LITERATURE SURVEY

2.1 Existing Problem

I. Soil Based Fertilizer Recommendation System for crop disease prediction (Dr.P.PandiSelvi, P. Poornima – 2021)

Agriculture is the heart and life of most Indians. But in recent days, the field was going down due to various natural calamities. In order to overcome the problem, various issues in this field need to be addressed. The soil type, fertilizer recommendation, diseases in plants and leaves. All these features need to be considered. Our proposed system was organized in such a way, to analyse the soil type, diseases in the leaves and finally to recommend the appropriate fertilizer to the farmers, that may be of great help to them. Plant disease, especially on leaves, is one of the major factors that reduce the yield in both quality and quantity of the food crops. Finding the leaf disease is an important role to preserve agriculture. Smart analysis and Comprehensive prediction model in agriculture helps the farmer to yield right crop at the right time. The main benefits of the proposed system are as follows: Yield right crop at the right time, Balancing the crop production, control plant disease, Economic growth, and planning to reduce the crop scarcity. Hence to Detect and recognize the plant diseases and to recommend fertilizer it is necessary to provide symptoms in identifying the disease at its earliest. Hence the authors proposed and implemented new fertilizers Recommendation System for crop disease prediction.

II. Study on prognostication of crop diseases using Artificial Intelligence (B. S. Eleena, Meghana Mangipudi, K. Apoorva - Computer Science - Asian Journal of Research in Computer Science – 2022)

It is universally accepted fact that crop diseases are one of the major threats in agriculture that ultimately result in drastic reduction of food supply. The present project study aims to use artificial intelligence in building a model which is integrated with a user-friendly web application. The web application is created using the Python-based Django framework. This user interface allows the user to choose a crop name and upload an image of a leaf wherein the trained model then begins the process of feature extraction on the image and tries to make an accurate prediction. The final result is displayed to the user confirming whether the crop may be “healthy” or the “diseased “and even the name of the disease that infects the plant will be displayed. The application also suggests a suitable treatment to combat the disease. Thus, the scope of this project study is very scalable as it can be easily

be used by amateur gardeners as well as by farmers. The model itself can also be extended to include more plant types along with any new diseases which may arise due to factors like climate change, pest - resistance etc.

III. Machine learning

Machine Learning is an emerging research field in crop yield analysis. Yield prediction is a very important issue in agriculture. Any farmer is interested in knowing how much yield he is about to expect. In the past, yield prediction was performed by considering farmer's experience on field and crop. The yield prediction is a major issue that remains to be solved based on available data. Machine learning techniques are the better choice for this purpose. Different Machine learning techniques are used and evaluated in agriculture for estimating the future year's crop production. This paper proposes and implements a system to predict crop yield from previous data. This is achieved by applying machine learning algorithms like Support Vector Machine and Random Forest on agriculture data and recommends fertilizer suitable for every crop. The paper focuses on creation of a prediction model which may be used for future prediction of crop yield. It presents a brief analysis of crop yield prediction using machine learning techniques.

2.2 References

[1]. P. Pandi Selvi, P. Poornima, "Soil Based Fertilizer Recommendation System for Crop Disease Prediction System", International Journal of Engineering Trends and Applications (IJETA) – Volume 8 Issue 2, Mar-Apr 2021 .

[2]. Plant Disease Detection Using Image Processing and Machine Learning Pranesh Kulkarni1 , Atharva Karwande1 , Tejas Kolhe1 , Soham Kamble1 , Akshay Joshi1 , Medha Wyawahare1 1 Department of Electronics and Telecommunication, Vishwakarma Institute of Technology. https://arxiv.org/ftp/arxiv/papers/2106/2106.10698.pdf

[3]. Zhen Nan Liu, et al., (2018), In this paper, authors have compared different machine learning algorithms for calculating, Standardised Precipitation Index (SPI) and SPEI. After data collection, Extreme learning methods, Online sequential extreme learning machine, Selfadaptive evolutionary extreme learning machine. Authors claimed that all

three algorithms can be applied successfully on drought forecasting. However, OS-ELM and SADE- ELM performs better than ELM.

[4]. K. G. Liakos et al., (2018) presented a comprehensive review of research dedicated to machine learning applications in agriculture domain. Various parameters on which work was analysed were: crop management, livestock management, water management and soil management. ML models have applied for crop yield prediction and disease detection. ML based detection can be extracted without the need of fusion of data from other resources. Author claims that farm management systems are evlvong into real artificial intelligent systems, with the ultimate scope of production improvement. Author motivates to use ML for the benefit of agriculture as it is the basic need amongst all other needs for survival.

[5]. S. D. Khirade, A. B. Patil, "Plant Disease Detection Using Image Processing", 2015 International Conference on Computing Communication Control and Automation, 2015, pp. 768-771, doi: 10.1109/ICCUBEA.2015.153

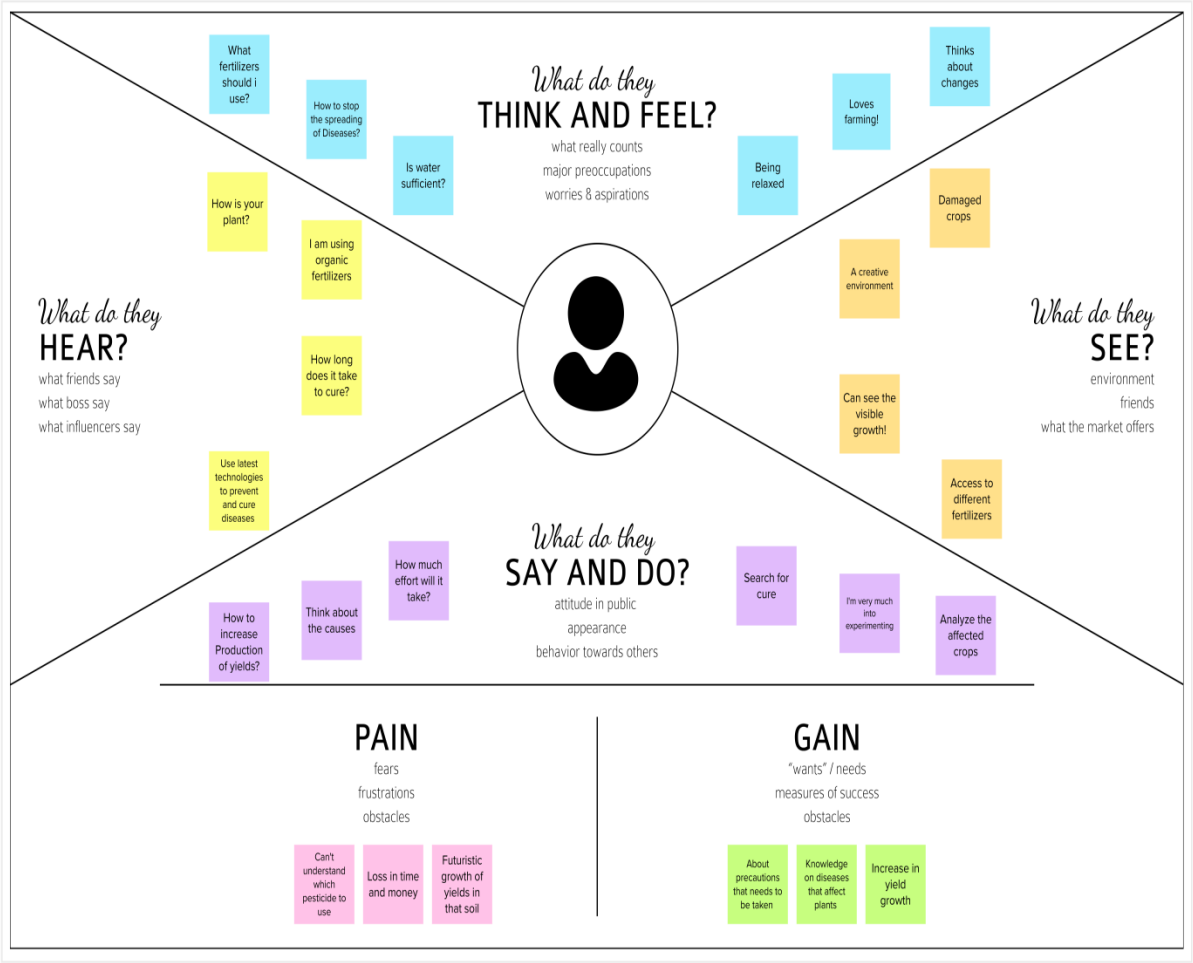
2.3 Problem Statement Definition

Problem Statement (PS)	**I am**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	Farmer	Find the cure for the diseases.	Due to lack of adequate knowledge	I didn't have enough resources	Dejected and displeased
PS-2	Corporate	Share my knowledge about diseases and fertilizers	I wasn't able to share my knowledge	I don't have good platform	Helpless and lost

3. IDEATION & PROPOSED SOLUTION

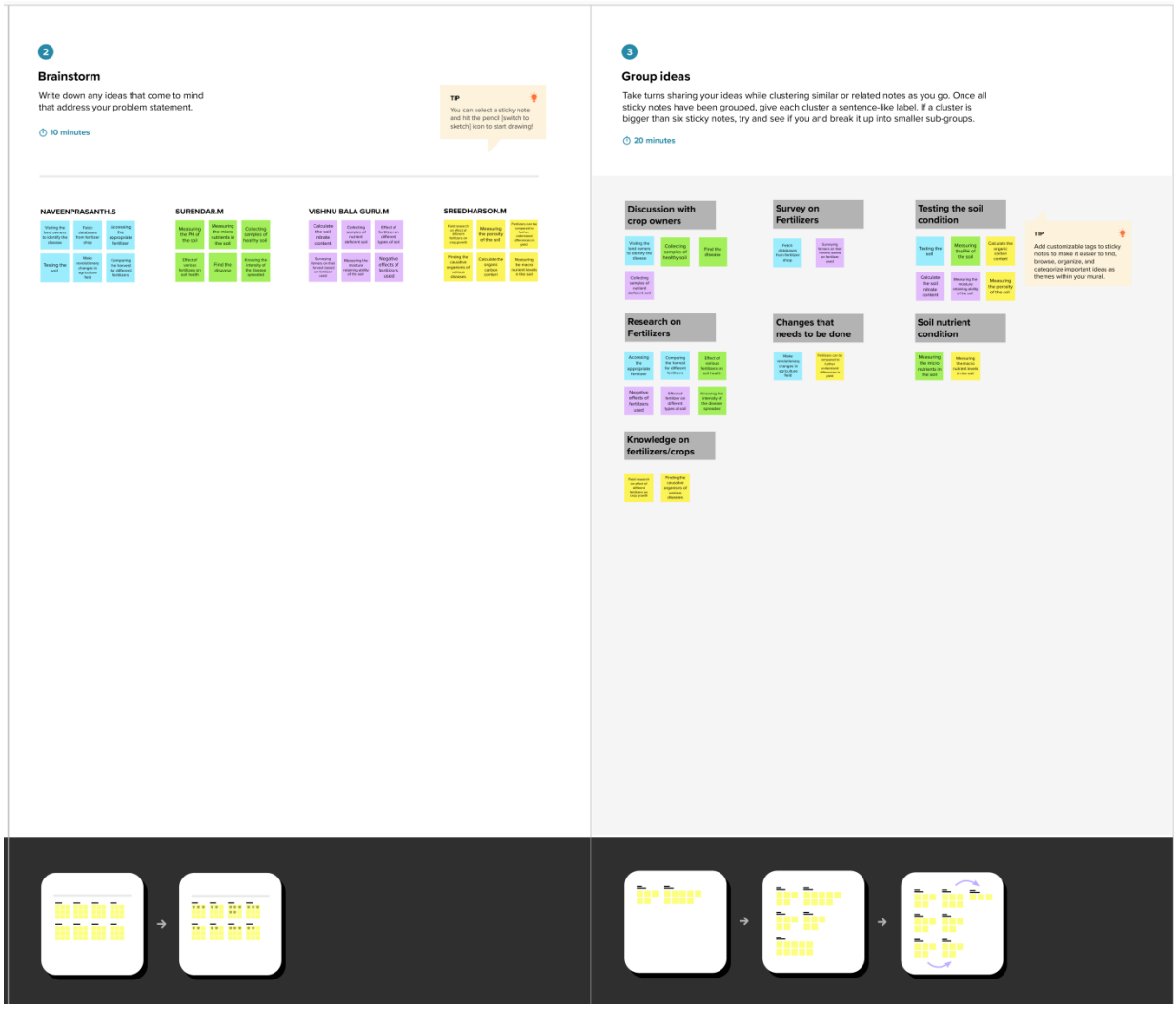
3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users.Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



3.2 Ideation & Brainstorming

Ideation and Brainstorming are performed to generate ideas and solutions. Brainstorming is a group activity unlike ideation.



4

Prioritize

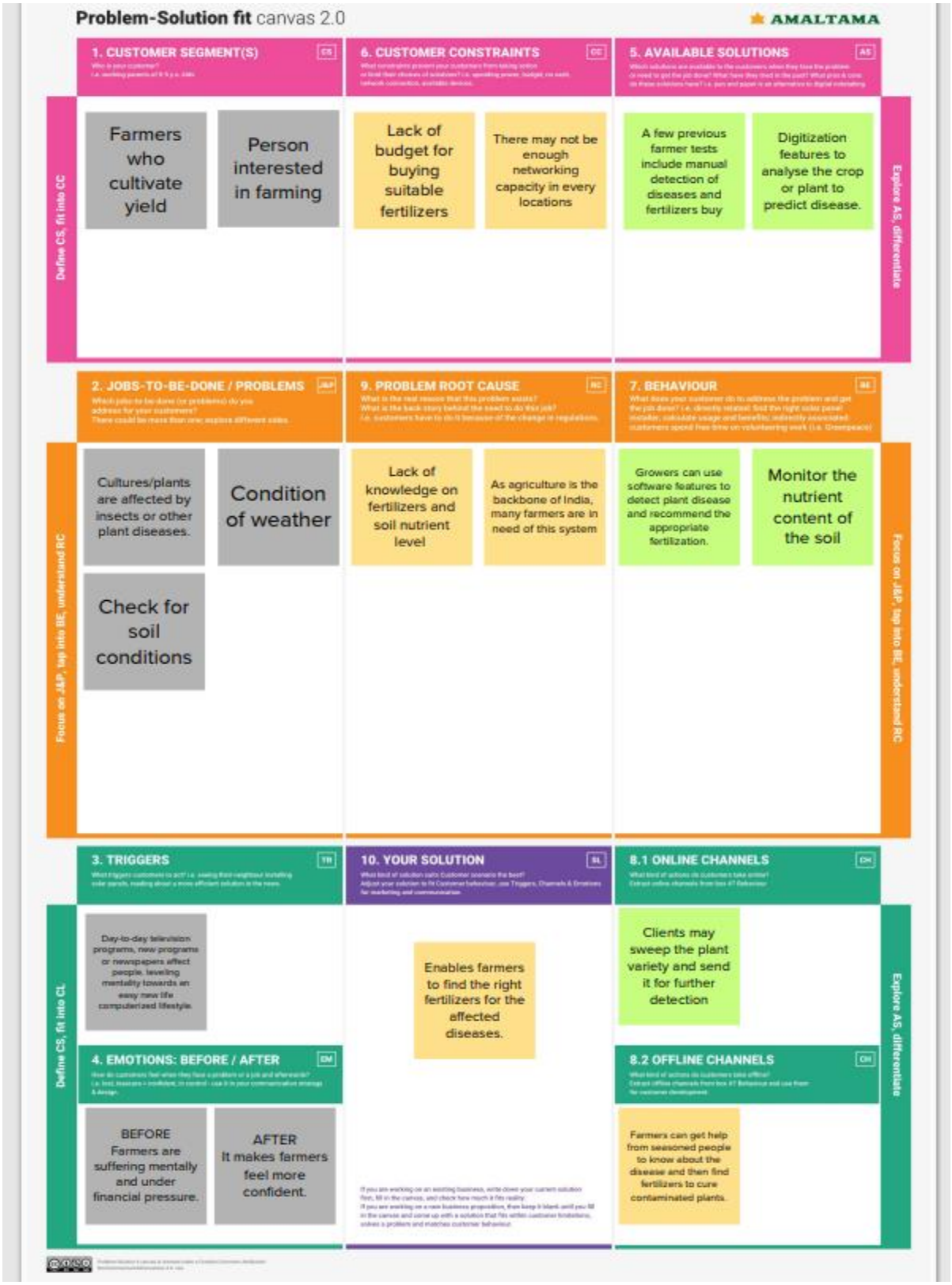
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	In India, the agriculture industry is extremely vital and crucial for economic and social development and jobs. In India, the agricultural sector provides a living for almost 48% of the population. But now a days farmers are facing the problems like disease that affect the crop yield. So this system helps the farmers by recommending the fertilizers for the appropriate diseases.
2.	Idea / Solution description	By developing the user friendly fertilizer recommendation system using artificial intelligence for the above mentioned problem. The fertilizers can be recommended by checking the soil condition, warmness and comparing the previously used pesticides.
3.	Novelty / Uniqueness	This solution will be a user friendly, and fast system to the farmers. And it helps the farmers by collecting the samples of crop that were affected by diseases.
4.	Social Impact / Customer Satisfaction	This solution will be a major support to farmers and those interested in farming. Also it increases the yield productivity.
5.	Business Model (Revenue Model)	Analyzing the disease and recommending the fertilizers in a tap makes farming life easy with minimal subscriptions would provide a acceptable organizational performance.
6.	Scalability of the Solution	The proposed solution can only be used to predict plant disease or only to track fertilizer inventory at nearby stores.

3.4 Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

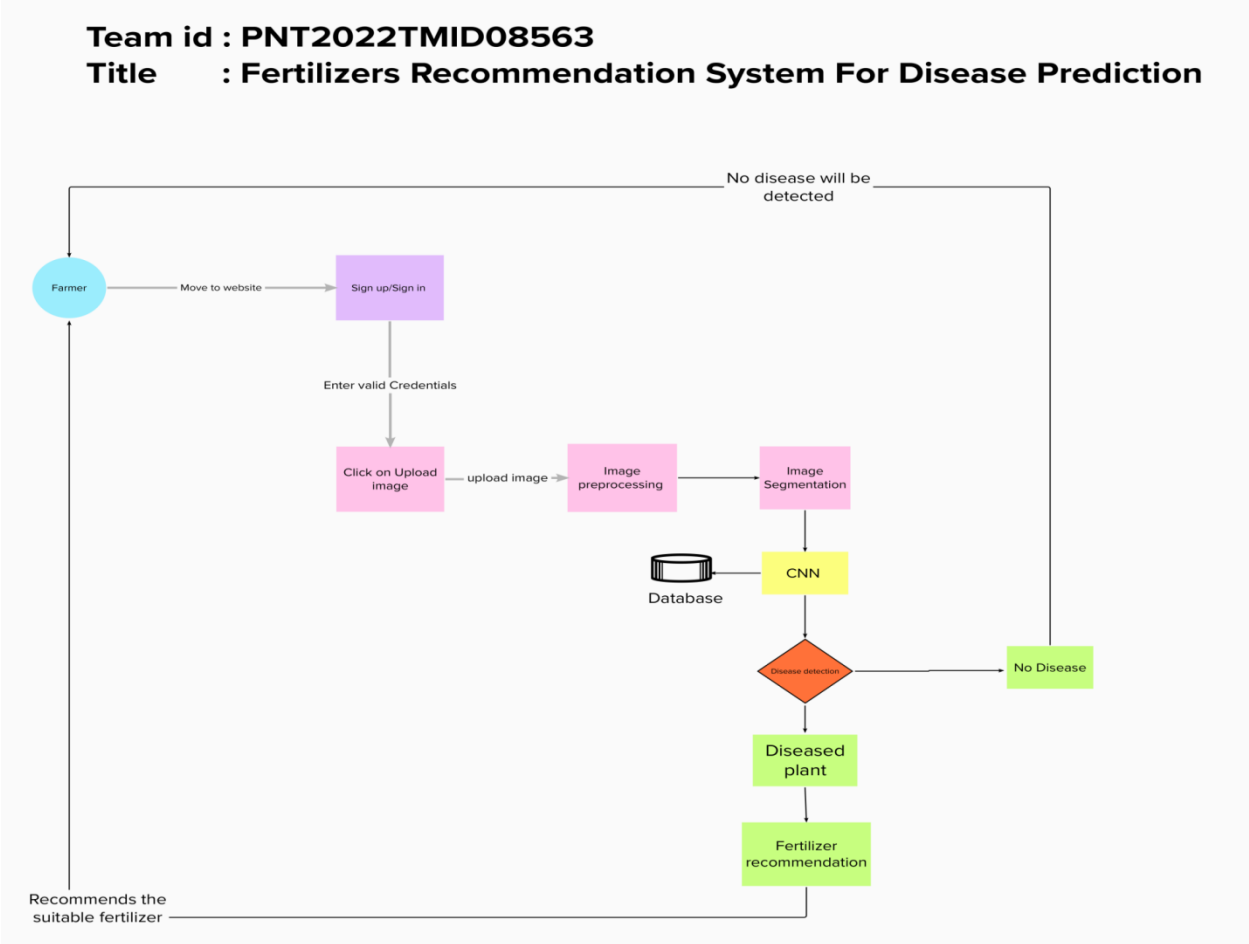
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Profile build
FR-2	User Confirmation	Confirmation via OTP
FR-3	User Login	Login with user name Login with password
FR-4	Upload images	Capture the image of the affected Crop Upload the image of the affected Crop
FR-5	Disease prediction	The disease forecasting model is trained with a large dataset and rolled out to predict disease by review of loaded images.
FR-6	Recommendation	Suggesting the right fertilizer for the anticipated disease Advising the alternative fertilizers for ease

4.2 Non-Functional Requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Website is designed to be sensitive and user friendly so it can be used on any device and from Whomsoever.
NFR-2	**Security**	User data is confidential and the user account is checked with the email ID provided at secure
NFR-3	**Reliability**	In the event of problems such as late replies, it will be corrected at stand on trust
NFR-4	**Performance**	The user has nothing to wait a long time from capturing and downloading until predicting and recommendationing.
NFR-5	**Availability**	The software may be used by anyone, regardless of the farmer's location or other network features.
NFR-6	**Scalability**	The system is designed so that it can withstand large numbers of users at any given moment

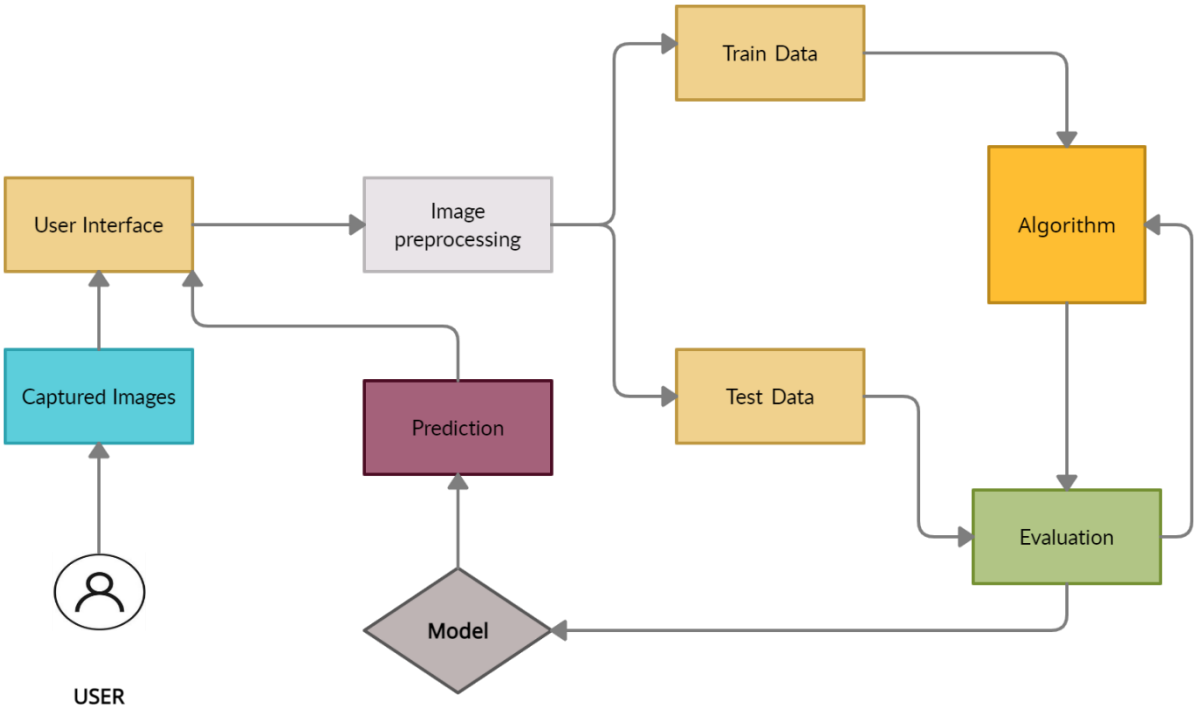
5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture

Solution architecture is the process of developing solutions based on predefined processes, guidelines and best practices with the objective that the developed solution fits within the enterprise architecture in terms of information architecture, system portfolios, integration requirements, etc.



5.3 User Stories

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
	Login	USN-2	As a user, I can log into the application by entering email & password	Must have valid credentials	High	Sprint-1
	Dashboard	USN-3	As a user, I can view the page of the application where I can upload my images and the fertilizer should be recommended	I can access my account / dashboard	High	Sprint-2
Customer (Web user)	Registration	USN-4	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-3
	Login	USN-5	As a user, I can login to my web dashboard with the login credentials	I can login by using my valid User credentials	High	Sprint-3
	Dashboard	USN-6	As a user, I can view the web application where i can upload my images for getting the suggestion of the fertilizer	I can access my account/ dashboard	High	Sprint-4
	Dataset	USN-7	As a web user, I want to view the dataset so that I can know which plants are there.	The plant types should be of different species and of wide variety.	Low	Sprint-1
Administrator	Login	USN-8	As a admin, I can login to the website using my login credentials	I can access my account/ dashboard	High	Sprint-4
	Notification	USN-9	Admin will recommend the fertilizers for certain plant diseases.	Notification by Gmail	High	Sprint-4
	Access Control	USN-10	Admin can control the access of users.	Access permission for Users	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

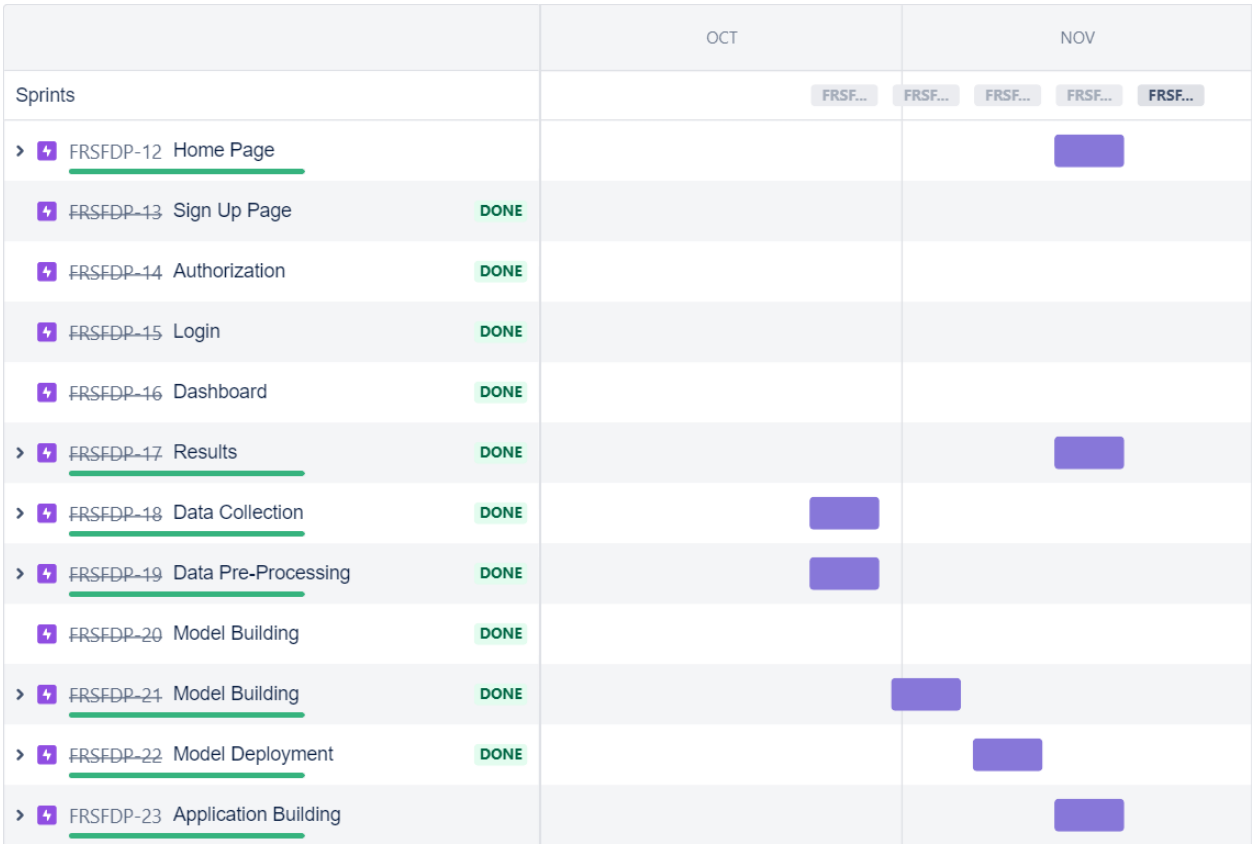
Sprint	FunctionalRequirement(Epic)	UserNumberStory	UserStory/Task	Story Points	Priority	Team Members
Sprint-4	Home	USN-1	As a user, I can view the home page which has description of the plant disease as well as options to sign up or log in.	3	High	Naveenprasanth, Surendar, Sreedharson, VishnuBalaGuru
Sprint-4	Results	USN-2	As a user, I can receive a disease in addition to recommendations on what I should do now.	2	High	Naveenprasanth, Surendar, Sreedharson, VishnuBalaGuru
Sprint-1	Data Collection	USN-3	I need to collect data (images of crops and with disease stages).	5	High	Naveenprasanth, Surendar, Sreedharson, VishnuBalaGuru
Sprint-1	Data Preprocessing	USN-4	I need to clean my data and prepare it for model building by doing pre-processing activities such as resizing etc.	8	High	Naveenprasanth, Surendar, Sreedharson, VishnuBalaGuru
Sprint-2	Model Building	USN-5	I need to build the model using Convolutional Neural Network (CNN).	13	High	Naveenprasanth, Surendar, Sreedharson, VishnuBalaGuru
Sprint-3	Model Deployment	USN-6	I need to deploy the Machine Learning model that was built.	13	Medium	Naveenprasanth, Surendar, Sreedharson, VishnuBalaGuru
Sprint-4	Application Building	USN-7	I need to build the website for the application using HTML, CSS etc.	8	High	Naveenprasanth, Surendar, Sreedharson, VishnuBalaGuru

6.2 Sprint Delivery Schedule

Sprint	Total StoryPoints	Duration	SprintStartDate	Sprint EndDate	Story PointsCompleted(as on Planned EndDate)	SprintReleaseDate(Actual)
Sprint-1	13	6 Days	24 Oct 2022	29 Oct 2022	13	31 Oct2022
Sprint-2	13	6 Days	31 Oct 2022	05 Nov 2022	13	07 Nov 2022
Sprint-3	13	6 Days	07 Nov 2022	12 Nov 2022	13	14 Nov 2022
Sprint-4	13	6 Days	14 Nov 2022	19 Nov 2022	13	19 Nov 2022

6.3 Reports From JIRA

Roadmap



Burndown Report

Velocity Report

7. CODING & SOLUTIONING

7.1 Python – app_final.py

```python
import numpy as np
import os
import pandas as pd
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from flask import Flask, render_template, request

app = Flask(__name__)

model = load_model("fruit.h5")

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

@app.route('/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)
        img=image.load_img(filepath,target_size=(128,128))
        x=image.img_to_array(img)
        x=np.expand_dims(x,axis=0)
        pred=np.argmax(model.predict(x),axis=1)
        df = pd.read_excel('precautions - fruits.xlsx')
        print(df.iloc[pred[0]]['caution'])

    return df.iloc[pred[0]]['caution']
if __name__=='__main__':
    app.run(debug=False)

```

7.2 Index.html

```
<htmllang="en">

<head>
<metacharset="UTF-8">
<metaname="viewport"content="width=device-width, initial-scale=1.0">
<metahttp-equiv="X-UA-Compatible"content="ie=edge">
<title>Fertilizer Recommented System and Disease Prediction</title>
<linkhref="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"rel="stylesheet">
<scriptsrc="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
<scriptsrc="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
<scriptsrc="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
<linkhref="{{url_for('static',filename='css/main.css')}}"rel="stylesheet">
  <style>

.bg-dark {
```

```
background-color: #7CFC00!important;
      }
#result {
color: #000f;
      }
  body
{
background-image: url("https://wallpaperaccess.com/full/327148.jpg");
background-size: cover;
}

  </style>
</head>

<body>

<navclass="navbar navbar-dark bg-dark">
<divclass="container">
<aclass="navbar-brand"href="#"><center><b>CROP DISEASE PREDICTION</b></center></a>
</div>
</nav>
<divclass="container">
<divid="content"style="margin-top:2em">
            <divclass="container">
            <divclass="row">
                  <divclass="col-sm-6 bd">
                  <h3>Plant Disease Prediction: </h3>
                  <br>
                  <p>Agriculture is one of the major sectors world 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 affect the production , Identification of disesases and taking necessary precautions are done through naked eye , which requires labour and labratries . This application helps farmers in detecting thr diseases by observing the spots and the leaves , which in turn saves effort and labour costs .</p>

  <imgsrc="http://www.geocities.ws/senthilirtt/First.jpg"height="50%",width="20%">
                  </div>
                  <divclass="col-sm-6">
                        <div>
                              <h4>Upload Image Here To Identify the Plant Disease</h4>
                  <formaction="http://localhost:5000/"id="upload-file"method="post"enctype="multipart/form-data">
                        <labelfor="imageUpload"class="upload-label">
                              Choose...
                        </label>
                        <inputtype="file"name="image"id="imageUpload"accept=".png, .jpg, .jpeg">
                  </form>


                  <divclass="image-section"style="display:none;">
                        <divclass="img-preview">
                              <divid="imagePreview">
                              </div>
                        </div>
                        <div>
                              <buttontype="button"class="btnbtn-info btn-lg "id="btn-predict">Predict!</button>
                        </div>
                  </div>
```

```
                    <divclass="loader"style="display:none;"></div>

                    <h3>
                            <spanid="result"></span>
                    </h3>

            </div>
                    </div>

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

<footer>
<scriptsrc="{{url_for('static',filename='js/main.js')}}"type="text/javascript"></script>
</footer>

</html>
```

7.3 main.css

```
.img-preview {
width: 256px;
height: 256px;
position: relative;
border: 5pxsolid#F8F8F8;
box-shadow: 0px2px4px0pxrgba(0,0,0,0.1);
margin-top: 1em;
margin-bottom: 1em;
}

.img-preview>div {
width: 100%;
height: 100%;
background-size: 256px256px;
background-repeat: no-repeat;
background-position: center;
}

input[type="file"] {
display: none;
}

.upload-label{
display: inline-block;
padding: 12px30px;
background: #39D2B4;
color: #fff;
font-size: 1em;
transition: all.4s;
cursor: pointer;
}

.upload-label:hover{
background: #34495E;
color: #39D2B4;
}
```

```
.loader {
border: 8pxsolid#f3f3f3; /*Light grey*/
border-top: 8pxsolid#3498db; /*Blue*/
border-radius: 50%;
width: 50px;
height: 50px;
animation: spin1slinearinfinite;
}

@keyframesspin {
0%{ transform: rotate(0deg); }
100%{ transform: rotate(360deg); }
}
```

7.4 main.js

```
$(document).ready(function () {
    // Init
    $('.image-section').hide();
    $('.loader').hide();
    $('#result').hide();

    // Upload Preview
    function readURL(input) {
        if (input.files&&input.files[0]) {
            var reader = new FileReader();
reader.onload = function (e) {
                $('#imagePreview').css('background-image', 'url(' + e.target.result + ')');
                $('#imagePreview').hide();
                $('#imagePreview').fadeIn(650);
            }
reader.readAsDataURL(input.files[0]);
        }
    }
    $("#imageUpload").change(function () {
        $('.image-section').show();
        $('#btn-predict').show();
        $('#result').text('');
        $('#result').hide();
readURL(this);
    });

    // Predict
    $('#btn-predict').click(function () {
        var form_data = new FormData($('#upload-file')[0]);

        // Show loading animation
        $(this).hide();
        $('.loader').show();

        // Make prediction by calling api /predict
$.ajax({
            type: 'POST',
            url: '/predict',
            data: form_data,
contentType: false,
            cache: false,
processData: false,
```

```
            async: true,
            success: function (data) {
               // Get and display the result
               $('.loader').hide();
               $('#result').fadeIn(600);
               $('#result').text(' Result:  ' + data);
               console.log('Success!');
            },
         });
      });

});
```

8. TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
Dataset_TC_001	Dataset	Data	Firstly, Collect all the necessary datasets for the project	Dataset	1.Login into the IBM dashboard 2.Navigate to the guided projects tab 3.Download the required dataset	Sample images from dataset	Dataset should be downloaded successfully	Working as expected	Pass	NIL	Y	NIL	Naveenprasanth
Preprocessing_TC_002	Dataset	Data	From the collected sample of images preprocessing will be done	Dataset	1.Open the Anaconda navigator 2.Import the required packages 3.Based on the samples the images are preprocessed	Sample images from dataset	Images are rescaled and properly modified for the identification	Working as expected	Pass	NIL	Y	NIL	Naveenprasanth, Sreetharson
User interface_TC_003	Functional	Home page	Verify user is able to view application home page	UI design	1.Have a rough draft of home page 2.Using HTML, CSS design a aesthetic home page layout	open index.html in browser	User should navigate to the appropriate webpage	Working as expected	Pass	NIL	Y	NIL	Naveenprasanth, Sreetharson, Surendar
Model building_TC_004	Functional	Deep Learning Model	The Model for the prediction is going to be built	Image preprocessing	1.Construct the basic work of the model 2.Add Dense layer for training the model 3.Save the model and train it	Images from Dataset	Training accuracy of over 85%	Working as expected	Pass	NIL	Y	NIL	Naveenprasanth, Sreetharson
Testing_TC_005	Functional	Deep Learning Model	To Test the Model with the Test Dataset	Model Building	1. Fit the test data to the model 2.Calculate the accuracy	Images from Dataset	Accuracy over 85%	Working as expected	Pass	NIL	Y	NIL	Naveenprasanth, Surendar, Vishnu Bala Guru
Prediction_TC_006	Functional	Deep Learning Model	To make predictions based on the trained dataset and predict the diseases	Testing	1.Train and test the model for the prediction of diseases 2.on the basis of trained dataset, the msystem will identify the disease of the plant	Images from Dataset	Identification of diseases	Working as expected	Pass	NIL	Y	NIL	Naveenprasanth, Sreetharson
Visualization_TC_007	Functional	Deep Learning Model	After identifying disease we shall find out the suitable fertilizer to cure the disease	Prediction of disease	1.Disease will be identified 2.On the basis of trained dataset, the system will suggest the suitable fertilizer for curing the crop disease	csv file	Suggesting fertilizer	Working as expected	Pass	NIL	Y	NIL	Naveenprasanth

8.2 User Acceptance Testing

1. PurposeofDocument

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

2. DefectAnalysis

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

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
Leaf Spots	10	4	2	3	19
Mosaic Leaf Pattern	9	6	3	0	18
Blights	4	5	2	1	12
Yellow leaves	11	4	3	20	38
Fruit rots	3	2	1	1	7
Misshapen leaves	2	7	1	0	10
Fruits spots	5	4	2	1	12
Totals	44	32	14	26	116

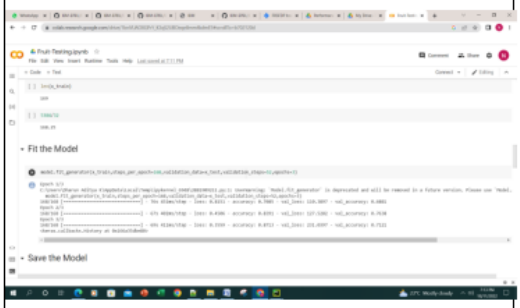
3. TestCaseAnalysis

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

Section	TotalCases	Not Tested	Fail	Pass
Leaf spots	19	0	0	19
Fruits spots	12	0	0	12
Mosaic leaf pattern	18	0	0	18
Blights	12	0	0	12
Misshapen Leaves	10	0	0	10
Yellow leaves	38	0	0	38
Fruit rots	7	0	0	7

9. RESULTS

9.1 Performance Metrics

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Total params: 38,160,755 Trainable params: 38,160,755 Non-Trainable params: 0	
2.	Accuracy	Training Accuracy –97.55 Validation Accuracy – 96.45	

10. ADVANTAGES & DISADVANTAGES

Advantages

- The proposed model here produces very high accuracy of classification.
- Very large datasets can also be trained and tested.
- No need to consult any specialists.
- Images of very high can be resized within the proposed itself.

Disadvantages

- For training and testing, the proposed model requires very high computational time.
- Works only on the pretrained diseases.
- When a plant is infected with multiple diseases the system may not predict all the diseases due to the mixed symptoms.

- The neural network architecture used in this project work has high complexity.

11. CONCLUSION

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

12. 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.

13. APPENDIX

Source Code

Python – app_final.py

```python
import numpy as np
import os
import pandas as pd
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from flask import Flask, render_template, request

app = Flask(__name__)

model = load_model("fruit.h5")

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

@app.route('/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)
        img=image.load_img(filepath,target_size=(128,128))
        x=image.img_to_array(img)
        x=np.expand_dims(x,axis=0)
        pred=np.argmax(model.predict(x),axis=1)
        df = pd.read_excel('precautions - fruits.xlsx')
        print(df.iloc[pred[0]]['caution'])

    return df.iloc[pred[0]]['caution']
if __name__=='__main__':
    app.run(debug=False)

```

index.html

```
<htmllang="en">

<head>
<metacharset="UTF-8">
<metaname="viewport"content="width=device-width, initial-scale=1.0">
<metahttp-equiv="X-UA-Compatible"content="ie=edge">
<title>Fertilizer Recommented System and Disease Prediction</title>
<linkhref="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"rel="stylesheet">
<scriptsrc="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
<scriptsrc="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
<scriptsrc="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
<linkhref="{{url_for('static',filename='css/main.css')}}"rel="stylesheet">
  <style>

.bg-dark {
background-color: #7CFC00!important;
      }
```

```
#result {
color: #000f;
      }
  body
{
background-image: url("https://wallpaperaccess.com/full/327148.jpg");
background-size: cover;
}

  </style>
</head>

<body>

<navclass="navbar navbar-dark bg-dark">
<divclass="container">
<aclass="navbar-brand"href="#"><center><b>CROP DISEASE PREDICTION</b></center></a>
</div>
</nav>
<divclass="container">
<divid="content"style="margin-top:2em">
            <divclass="container">
            <divclass="row">
                        <divclass="col-sm-6 bd">
                        <h3>Plant Disease Prediction: </h3>
                        <br>
                        <p>Agriculture is one of the major sectors world 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 affect the production , Identification of disesases and taking necessary precautions are done through naked eye , which requires labour and labratries . This application helps farmers in detecting thr diseases by observing the spots and the leaves , which in turn saves effort and labour costs .</p>

  <imgsrc="http://www.geocities.ws/senthilirtt/First.jpg"height="50%",width="20%">
                        </div>
                        <divclass="col-sm-6">
                                    <div>
                                                <h4>Upload Image Here To Identify the Plant Disease</h4>
                        <formaction="http://localhost:5000/"id="upload-file"method="post"enctype="multipart/form-data">
                                    <labelfor="imageUpload"class="upload-label">
                                                Choose...
                                    </label>
                                    <inputtype="file"name="image"id="imageUpload"accept=".png, .jpg, .jpeg">
                        </form>


                        <divclass="image-section"style="display:none;">
                                    <divclass="img-preview">
                                                <divid="imagePreview">
                                                </div>
                                    </div>
                                    <div>
                                                <buttontype="button"class="btnbtn-info btn-lg "id="btn-predict">Predict!</button>
                                    </div>
                        </div>

                        <divclass="loader"style="display:none;"></div>
```

```
                    <h3>
                        <spanid="result"></span>
                    </h3>

                </div>
                    </div>

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

<footer>
<scriptsrc="{{url_for('static',filename='js/main.js')}}"type="text/javascript"></script>
</footer>

</html>
```

main.css

```
.img-preview {
width: 256px;
height: 256px;
position: relative;
border: 5pxsolid#F8F8F8;
box-shadow: 0px2px4px0pxrgba(0,0,0,0.1);
margin-top: 1em;
margin-bottom: 1em;
}

.img-preview>div {
width: 100%;
height: 100%;
background-size: 256px256px;
background-repeat: no-repeat;
background-position: center;
}

input[type="file"] {
display: none;
}

.upload-label{
display: inline-block;
padding: 12px30px;
background: #39D2B4;
color: #fff;
font-size: 1em;
transition: all.4s;
cursor: pointer;
}

.upload-label:hover{
background: #34495E;
color: #39D2B4;
}
```

```
.loader {
border: 8pxsolid#f3f3f3; /*Light grey*/
border-top: 8pxsolid#3498db; /*Blue*/
border-radius: 50%;
width: 50px;
height: 50px;
animation: spin1slinearinfinite;
}

@keyframesspin {
0%{ transform: rotate(0deg); }
100%{ transform: rotate(360deg); }
}
```

main.js

```
$(document).ready(function () {
    // Init
    $('.image-section').hide();
    $('.loader').hide();
    $('#result').hide();

    // Upload Preview
    function readURL(input) {
        if (input.files&&input.files[0]) {
            var reader = new FileReader();
reader.onload = function (e) {
                $('#imagePreview').css('background-image', 'url(' + e.target.result + ')');
                $('#imagePreview').hide();
                $('#imagePreview').fadeIn(650);
            }
reader.readAsDataURL(input.files[0]);
        }
    }
    $("#imageUpload").change(function () {
        $('.image-section').show();
        $('#btn-predict').show();
        $('#result').text('');
        $('#result').hide();
readURL(this);
    });

    // Predict
    $('#btn-predict').click(function () {
        var form_data = new FormData($('#upload-file')[0]);

        // Show loading animation
        $(this).hide();
        $('.loader').show();

        // Make prediction by calling api /predict
$.ajax({
            type: 'POST',
            url: '/predict',
            data: form_data,
contentType: false,
            cache: false,
processData: false,
```

```
            async: true,
            success: function (data) {
               // Get and display the result
               $('.loader').hide();
               $('#result').fadeIn(600);
               $('#result').text(' Result:  ' + data);
               console.log('Success!');
            },
         });
      });

   });
```

Fruit Model

Image Pre-processing

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen =
ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1)
x_train = train_datagen.flow_from_directory(r' Dataset Plant Disease/fruit-dataset/fruit-
dataset/train ',target_size=(128,128),batch_size=32,class_mode='categorical')
x_test=test_datagen.flow_from_directory(r' Dataset Plant Disease/fruit-dataset/fruit-dataset/test
',target_size=(128,128),
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPool2D,Flatten
model=Sequential()
```

Adding Layers

```
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
x_train.class_indices
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(40, kernel_initializer='uniform',activation='relu'))
model.add(Dense(20, kernel_initializer='random_uniform',activation='relu'))
model.add(Dense(6,activation='softmax', kernel_initializer='random_uniform'))
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
len(x_train)
5384/32
```

Fitting The Model

```
model.fit_generator(x_train,steps_per_epoch=168,validation_data=x_test,validation_steps=52,epochs=3)
model.save("fruit.h5")
ls
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
model = load_model("fruit.h5")
```

Testing Model

```
import keras.utils as image
img = image.load_img(r'C:/Users/RAJU/OneDrive/Desktop/Sathiya/IBM/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple___healthy/00fca0da-2db3-481b-b98a-9b67bb7b105c___RS_HL 7708.JPG',target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x),axis=1)
pred
index
=['Apple___Black_rot','Apple___healthy','Corn_(maize)___Northern_Leaf_Blight','Corn_(maize)___healthy','Peach___Bacterial_spot','Peach___healthy']
print('the given image belogs to=',index[pred[0]])
img = image.load_img(r'C:Users/RAJU/OneDrive/Desktop/Sathiya/IBM/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple___Black_rot/0f3d45f4-e121-42cd-a5b6-be2f866a0574___JR_FrgE.S 2870.JPG',target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x),axis=1)
print('the given image belogs to=',index[pred[0]])
img = image.load_img(r'C:/Users/RAJU/OneDrive/Desktop/Sathiya/IBM/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Corn_(maize)___Northern_Leaf_Blight/00a14441-7a62-4034-bc40-b196aeab2785___RS_NLB 3932.JPG',target_size=(128,128))
x=image.img_to_array(img)
```

```
x=np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x),axis=1)
print('the given image belogs to=',index[pred[0]])
img = image.load_img(r'C:/Users/RAJU/OneDrive/Desktop/Sathiya/IBM/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Corn_(maize)___healthy/0a68ef5a-027c-41ae-b227-159dae77d3dd___R.S_HL 7969 copy.jpg',target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x),axis=1)
print('the given image belogs to=',index[pred[0]])
img = image.load_img(r'C:/Users/RAJU/OneDrive/Desktop/Sathiya/IBM/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Peach___Bacterial_spot/00ddc106-692e-4c67-b2e8-569c924caf49___Rutg._Bact.S 1228.JPG',target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x),axis=1)
print('the given image belogs to=',index[pred[0]])
img = image.load_img(r'C:/Users/RAJU/OneDrive/Desktop/Sathiya/IBM/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Peach___healthy/1a07ce54-f4fd-41cf-b088-144f6bf71859___Rutg._HL 3543.JPG',target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x),axis=1)
print('the given image belogs to=',index[pred[0]])
```

Vegetable Model

Image Pre-processing

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1)
x_train = train_datagen.flow_from_directory(r' Dataset Plant Disease/veg-dataset/veg-dataset/train ',target_size=(128,128),batch_size=16,class_mode='categorical')
```

```
x_test=test_datagen.flow_from_directory(r' Dataset Plant Disease/veg-dataset/veg-dataset/train
',target_size=(128,128),
batch_size=16,class_mode='categorical')
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPool2D,Flatten
model=Sequential()
```

Adding Layers

```
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
x_train.class_indices
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(300, kernel_initializer='uniform',activation='relu'))
model.add(Dense(150, kernel_initializer='uniform',activation='relu'))
model.add(Dense(75, kernel_initializer='uniform',activation='relu'))
model.add(Dense(9,activation='softmax', kernel_initializer='uniform'))
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
len(x_train)
11386/16
```

Fitting The Model

```
model.fit_generator(x_train,steps_per_epoch=89,validation_data=x_test,validation_steps=27,epoch
s=20)
model.save("vegetable.h5")
ls
model.summary()
```

Test The Model

```
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
model = load_model("vegetable.h5"
```

```
index=['Pepper,_bell___Bacterial_spot','Pepper,_bell___healthy','Potato___Early_blight','Potato__
_Late_blight','Potato___healthy','Tomato___Bacterial_spot','Tomato___Late_blight','Tomato___Le
af_Mold','Tomato___Septoria_leaf_spot']
import keras.utils as image
img = image.load_img(r'C:/Users/RAJU/OneDrive/Desktop/Sathiya/IBM/Dataset Plant
Disease/veg-dataset/veg-dataset/test_set/Pepper,_bell___Bacterial_spot/ad921dec-e88f-41d8-9455-
0880c69063fc___NREC_B.Spot 9216.JPG',target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x),axis=1)
print('the given image belogs to=',index[pred[0]])
img = image.load_img(r'C:/Users/RAJU/OneDrive/Desktop/Sathiya/IBM/Dataset Plant
Disease/veg-dataset/veg-dataset/test_set/Pepper,_bell___healthy/b45d62a2-3de1-411b-8f88-
ab52195b6dda___JR_HL 7639.JPG',target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x),axis=1)
print('the given image belogs to=',index[pred[0]])
img = image.load_img(r'C:/Users/RAJU/OneDrive/Desktop/Sathiya/IBM/Dataset Plant
Disease/veg-dataset/veg-dataset/test_set/Potato___Early_blight/b6220993-c51f-48fa-bee9-
fb5cb89cc0c4___RS_Early.B 7467.JPG',target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x),axis=1)
print('the given image belogs to=',index[pred[0]])
img = image.load_img(r'C:/Users/RAJU/OneDrive/Desktop/Sathiya/IBM/Dataset Plant
Disease/veg-dataset/veg-dataset/test_set/Potato___Late_blight/d20fef22-7016-4115-9930-
a66b32c718c6___RS_LB 4417.JPG',target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x),axis=1)
print('the given image belogs to=',index[pred[0]])
```

```
img = image.load_img(r'C:/Users/RAJU/OneDrive/Desktop/Sathiya/IBM/Dataset Plant Disease/veg-dataset/veg-dataset/test_set/Potato___healthy/ff700844-68ad-4e99-8427-58a39c07f817___RS_HL 1860.JPG',target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x),axis=1)
print('the given image belogs to=',index[pred[0]])
img = image.load_img(r'C:/Users/RAJU/OneDrive/Desktop/Sathiya/IBM/Dataset Plant Disease/veg-dataset/veg-dataset/test_set/Tomato___Bacterial_spot/b433dcf3-c57d-4f5e-9116-5aaeecbaef01___GCREC_Bact.Sp 3715.JPG',target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x),axis=1)
print('the given image belogs to=',index[pred[0]])
img = image.load_img(r'C:/Users/RAJU/OneDrive/Desktop/Sathiya/IBM/Dataset Plant Disease/veg-dataset/veg-dataset/test_set/Tomato___Late_blight/b19055d9-6a5d-4b73-90a5-3b3685ac344a___GHLB2 Leaf 8729.JPG',target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x),axis=1)
print('the given image belogs to=',index[pred[0]])
img = image.load_img(r'C:/Users/RAJU/OneDrive/Desktop/Sathiya/IBM/Dataset Plant Disease/veg-dataset/veg-dataset/test_set/Tomato___Leaf_Mold/d8f27502-b864-4da3-9dd5-b7eccf2e8cbc___Crnl_L.Mold 6708.JPG',target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x),axis=1)
print('the given image belogs to=',index[pred[0]])
img = image.load_img(r'C:/Users/RAJU/OneDrive/Desktop/Sathiya/IBM/Dataset Plant Disease/veg-dataset/veg-dataset/test_set/Tomato___Septoria_leaf_spot/c3233f4f-a6f5-4bfb-a6bf-bd5e5d045f7a___JR_Sept.L.S 2728.JPG',target_size=(128,128))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = np.argmax(model.predict(x),axis=1)
```

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

Git Hub Link

https://github.com/IBM-EPBL/IBM-Project-7861-1658901018

Project Demo Link

https://drive.google.com/file/d/10YXiMlwkBJ1Z8XArPFLIcB4_tEMChE_R/view?usp=share_link