DR. MAHALINGAM COLLEGE OF ENGINEERING AND TECHNOLOGY

FERTILIZERS RECOMMENDATION SYSTEM FOR DISEASE PREDICTION

DONE BY

TEAM ID: PNT2022TMID08602

A.V.RANJANI (727619BCS041)

G.SANTHIYA(727619BCS043)

M.SNEHA KARTHIKA(727619BCS065)

S.MANOJ(727620BCS317)

INDEX

1. INTRODUCTION

1.1 Project Overview

Agriculture is the main source of income and survival in India for majority population. Agriculture is done from ages. Hence a rich collection of agricultural past data is available. Information Technology can be used to process such a large amount of data and then for recommendation. Various data mining techniques can be used for finding recommendations about crops and fertilizers.

Agriculture is significant to a nation's ability to innovate. All nations are built on agriculture, which provides food and raw materials. Thousands of individuals work in the agricultural sector, which is directly tied to agricultural products. However, in recent days the field was deteriorating as a result of different natural disasters. One of the main things that lower the yield of the food crops in terms of both quality and quantity is plant disease, especially on the leaves. Early detection of plant diseases is crucial as they impair the growth of their particular species.

1.2 Purpose

Plant illnesses are more common than other types of diseases, so we need a system to detect them and also to suggest the right fertilizer depending on the disease forecast. By using a digital image of the plant leaves. Machine learning algorithms are very good at recognizing and diagnosing plant diseases. Digital images are processed using a variety of image-processing techniques in order to recover crucial information needed for disease classification.

2. LITERATURE SURVEY

2.1 Existing Problem

I. Artificial Intelligence in Agriculture

(**Authors: Gouravmoy Bannerjee, Uditendu Sarkar, Swarup Das, Indrajit Ghosh - 2022)**

The domain of agriculture faces many challenges such as disease and pest infestation, improper soil treatment, inadequate drainage, and irrigation, and many more. These lead to severe crop loss along with environmental hazards due to excessive use of chemicals. Several researches have been conducted to address these issues. The field of artificial intelligence with its rigorous learning capabilities have become a key technique for solving different agriculture related problems. Systems are being developed to assist the agricultural experts for better solutions throughout the world. Issues pertaining to soil and irrigation management are very vital in agriculture. Improper irrigation and soil management lead to crop loss and degraded quality. Crop diseases are also a matter of grave concern to a farmer. Significant expertise and experience are required in order to detect an ailing plant and to take necessary steps for recovery. Computer aided systems are being used worldwide to diagnose the diseases and to suggest control measures. At the counter side the things which is required either the physical or software cost eventually reaches the maximum extreme some may not be affordable.

II. Fertilizer Recommendation Methods for Precision Agriculture

(Authors: Humberto Martins Beneduzzi, Eduardo Godoy de Souza, Wendel Kaian Oliveira Moreira, Ricardo Sobjak – 2022)

Spatial variability management of soil chemical attributes is one of the approaches to be employed in the face of the constant challenge of increasing agricultural yield to meet world demand. In this sense, precision agriculture has as one of its tools the application of inputs at varying rates, which seeks to determine the ideal amount of fertilizer at each point of the crop, contrary to the conventional recommendation approach based on average values. In this context, this work studied the fertilizer recommendation methods used in site-specific nutrient management and the calculation methodologies for N, P, and K recommendations. Software for recommending fertilizers with a focus on precision agriculture. We sought to identify, throughout the SLS process, which cited software had the functionality to carry out fertilizer recommendation at varying rate.

III. Prediction of Crop Yield and Fertilizer Recommendation using Machine Learning Algorithms

(Authors: Devdatta A. Bondre – 2021)

Machine learning is an emerging research field in crop yield analysis. Yield prediction is a 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.

IV. Artificial Intelligence in Agriculture

(Authors: Dr.P. Pandi Selvi , P. Poornima – 2021)

India is an agricultural country and depends on agricultural products for their wellbeing. It is agriculture that promotes the economic growth and the development of our country. But recently many problems have been faced by the farmers due to certain natural calamities. Apart from these major calamities, they were also in lack of sufficient knowledge about the nutrients present in their soil. The characteristics of the soil type vary accordingly based on its nutritive value. Not only the soil type, the climatic condition and the usage of fertilizer also play a major role. Certain varieties of crops can be cultivated based on the climatic condition in their locality and accordingly the fertilizers can be preferred. In certain situation the usage of fertilizers also affects the cultivation. Under a climatic condition the cultivation of a right crop, usage of right fertilizer to the soil gives better yield. Mostly, fertilizers were recommended based on the nutrients present

in the soil. Hence while preferring fertilizer the farmer needs to consider the soil type, the crop, and the pesticides. On using chemical fertilizer, the quality or the nutrients present in the soil was degraded, that promotes a decrease in the nutritive value of the soil. Another major factor to be considered is the disease in the crop cultivated. Identifying the disease in the plants and preferring appropriate fertilizer by the agriculturist to the farmers plays a major role. In earlier days, all these processes were carried out manually. But with the advancement of technology the entire system was digitalized. On considering all these the authors proposed a new framework that can be used in real life, which enables the farmers in solving certain problems.

2.2 References

[1] Artificial Intelligence in Agriculture International Journal of Scientific Research in Computer Science Applications and Management Studies Authors: Gouravmoy Bannerjee, Uditendu Sarkar, Swarup Das, Indrajit Ghosh.

[2] Fertilizer recommendation methods for precision agriculture- Book of Western parna state university-PR-BrazilAuthors: Humberto Martins Beneduzzi, Eduardo Godoy de Souza.

[3] Prediction of crop yield and fertilizer recommendation system using machine learning algorithms-International Journal of Engineering Applied Sciences and Technology, 2019 Vol. 4, Issue 5, ISSN No. 2455-2143, Pages 371-376 Published Online September 2019 in IJEASTAuthors: Devdatta A. BondreWendel Kaian Oliveira Moreira, Ricardo Sobjak.

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

2.3 Problem Statement Definition

Problem Statement (PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	A Farmer	Enrich my field with wholesome crops and protect them from pathogens.	I am financially not stable and reside in a rural location where agriculture department rarely visits.	Insufficient knowledge about the plants and disease of the infected plant has to be predicted to resume its fine growth	Mentally low and occasionally being a spendthrift because of bad weather or some government farm bills.

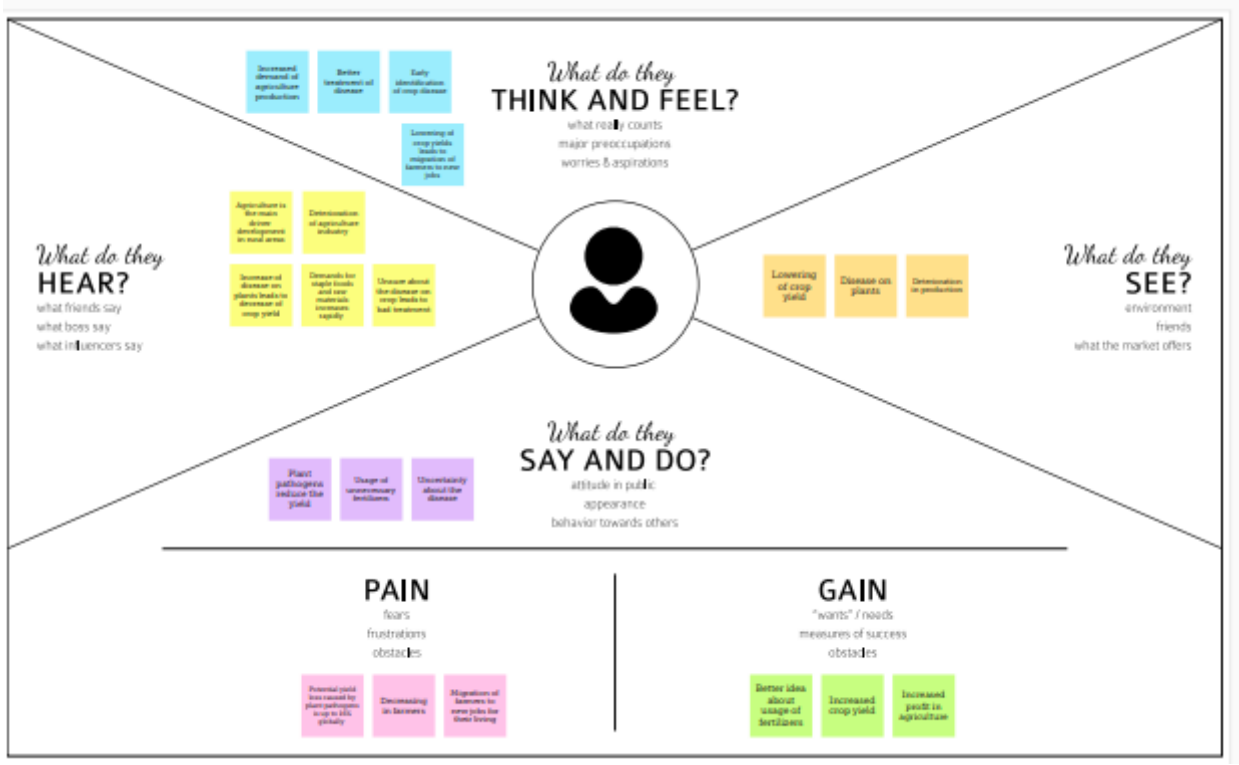
PS-2	A Harvester who cultivates different kinds of crops according to the season.	Obtain a healthy crop yield with appropriate fertilizers.	Since crop diseases are common.	Appropriate fertilizer to restore crops to a healthy state is unknown.	Low self-esteem due to my mental and financial instability and lack of sufficient understanding about all of my crops.
PS-3	A gardener who looks after the crops on a farm.	Keep the farm functional and generate a sufficient profit.	The cost of labor and fertilizers makes it difficult to maintain the farm.	Crop production is inadequate, there will be less profit or loss.	Being unable to effectively treat the diseased plant makes me feel depressed.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

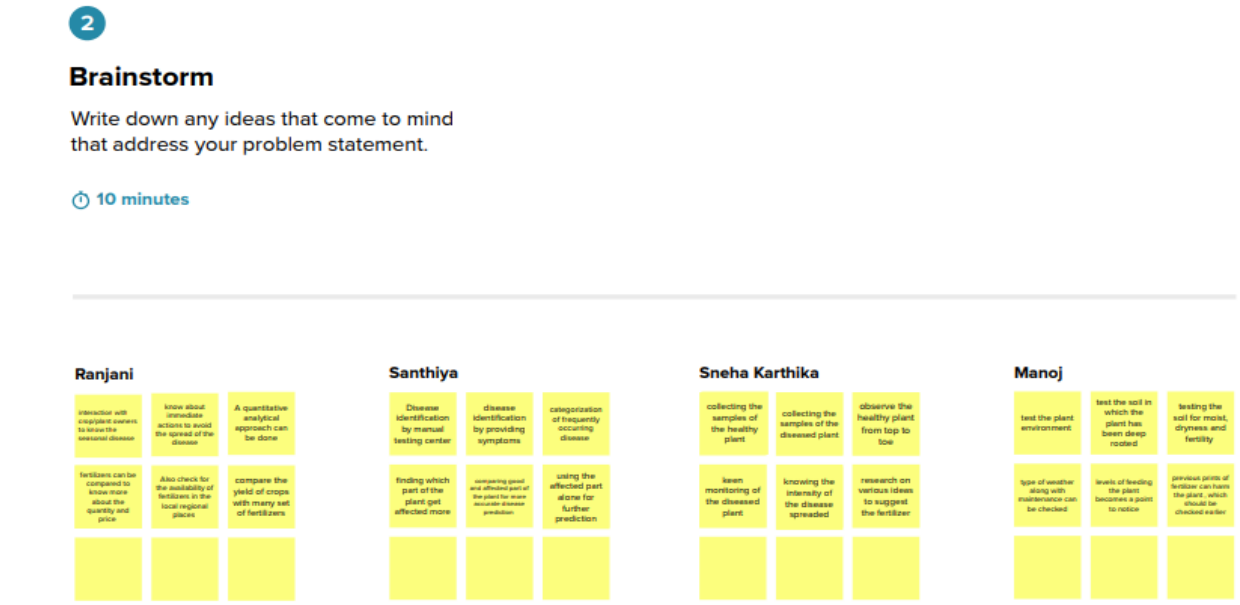
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

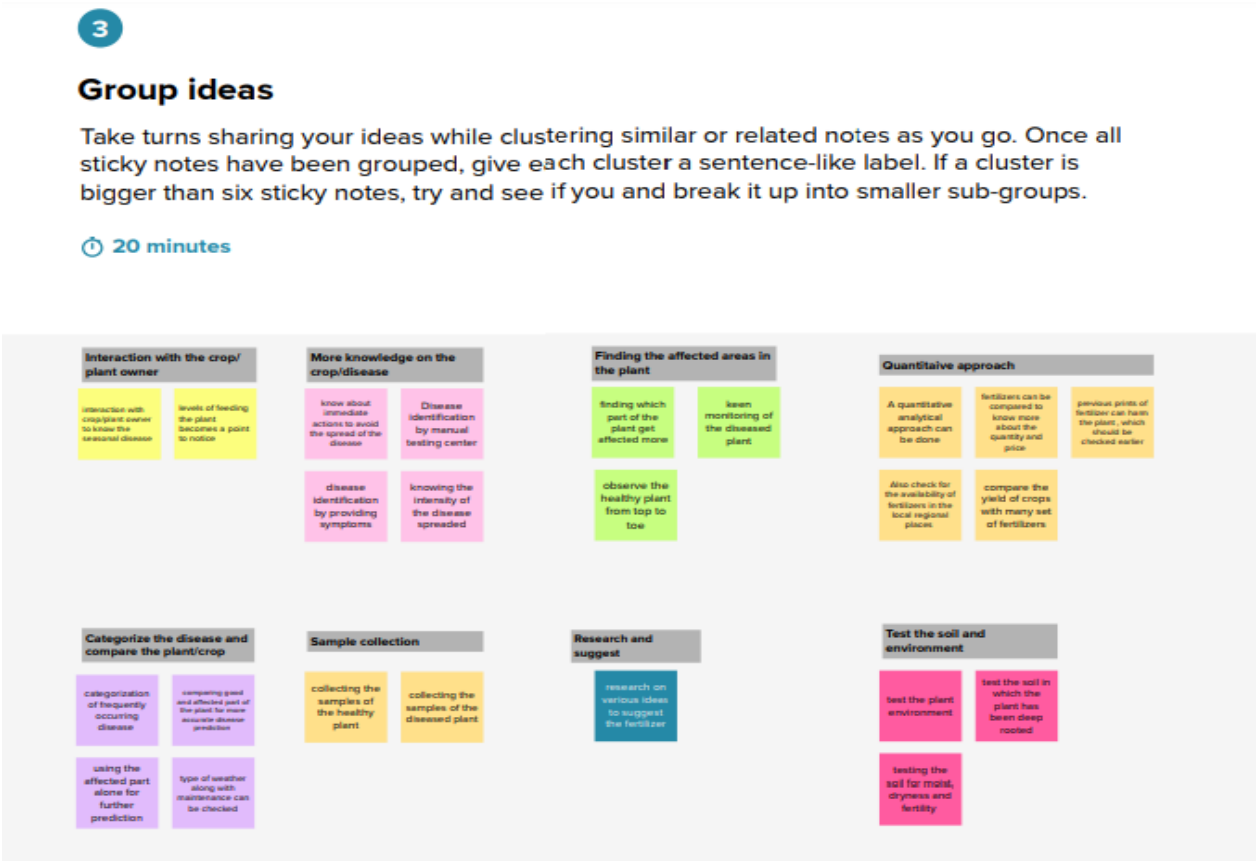
It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



3.2 Ideation & Brainstorming

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





3.3 Proposed Solution

i. Problem Statement:

Agriculture being the economy of India, crops, plants, trees, etc., should be nourished to increase the economy without a backdrop. Though there are lots of problems that arise from the root node till the leaf or bud, which are called asthe plant disease. These problems in plants can be predicted and further recommendation of fertilizers can be rendered to make the plant healthy.

ii. Idea/Solution Description:

The idea for the problem stated above can be solved by an automated disease prediction system that recommends the fertilizer to its customers. The pesticides can be recommended by comparing the current diseased pictures uploaded by the customers and a healthy picture. The recommendation of pesticides/fertilizers does not stop there but also leveraged by suggesting the shops that have the inventory of the recommended fertilizer.

iii. Novelty/Uniqueness:

The solution to be developed will have a super-fast response when comparedto the other applications on the web today. There are many applications that provide health advisory, calendar guides and guest speaker talk. But this application will predict the plant infection and will also suggest a healthymeasure of recovering it to a healthy state.

iv. Social Impact/Customer Satisfaction:

The application will be a major assistance for the farmers and people intrigued in the field of agriculture. For ex: There is a statistic that accounts on more than 15 million downloads for an app named plantix. The level of customer satisfaction will simply be immeasurable as the yield increases and the automated work together provides an utmost customer satisfaction.

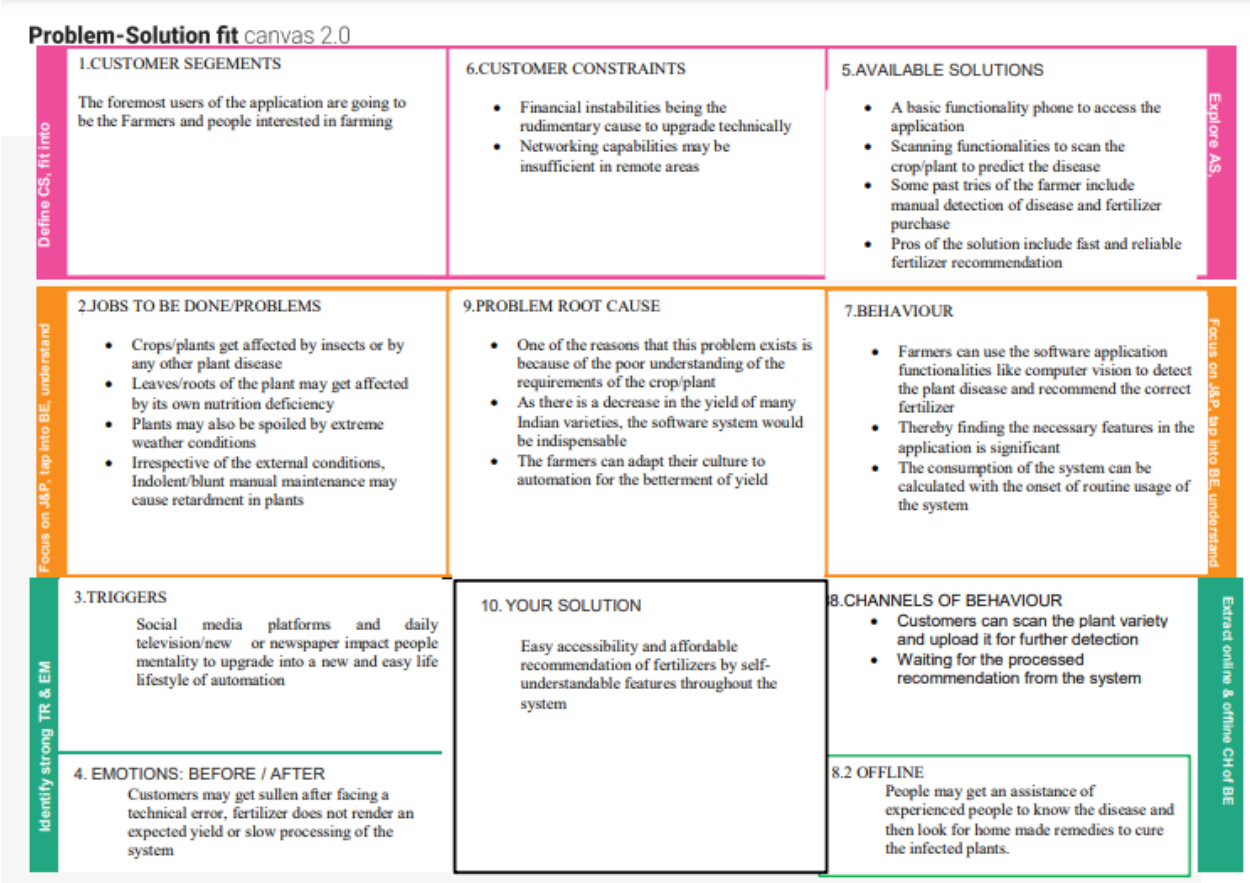
v. Business Model:

Increase in the users are directly proportional to the download rate that ultimately rises the revenue of the developer and the future growth of agriculture will always be in a boom irrespective of the technology diversities.

vi. Scalability of Solution:

The solution proposed can be used only to predict the plant disease or only to know the inventory of fertilizers in the shops nearby. The SAAS model will be easy to use interface alongside of being a franchise model.

3.4 Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

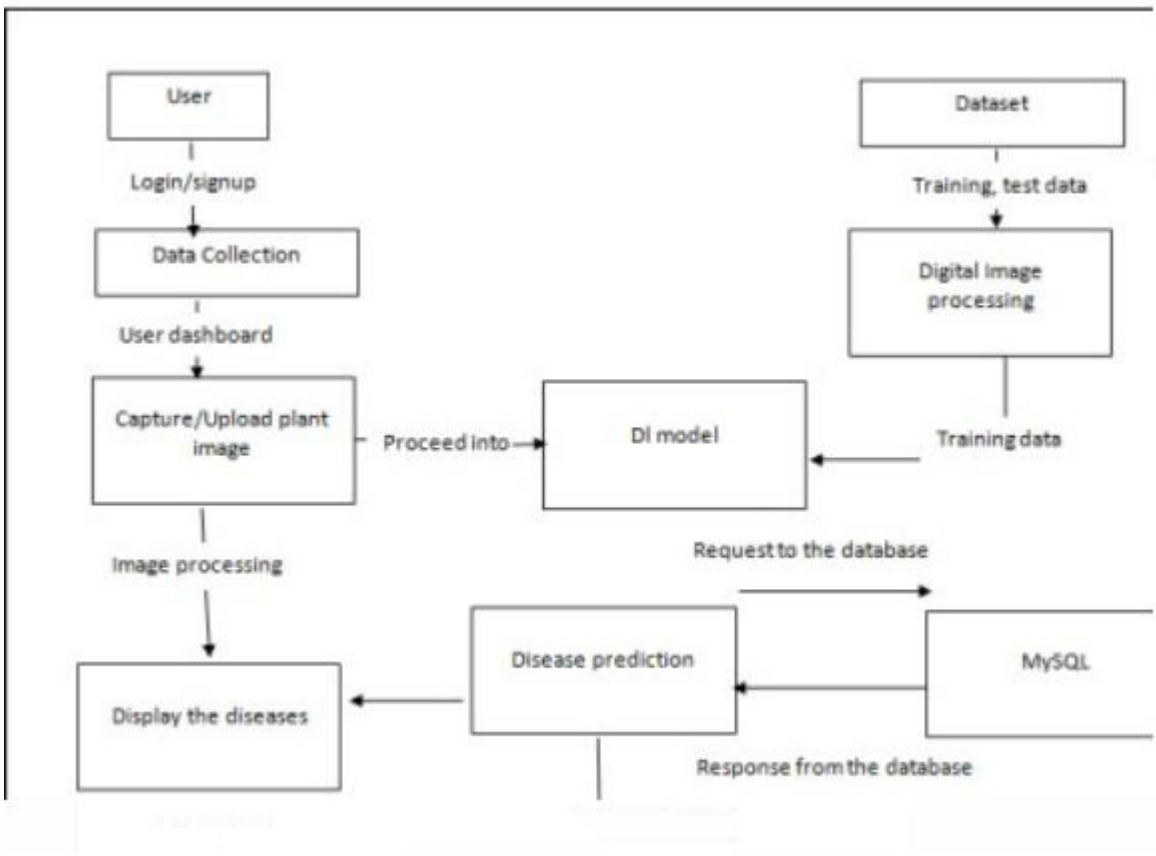
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**User Registration**	Form-based registration Profile build Rudimentary details Plant/crop information Soil and Environment data
FR-2	**User Confirmation**	Affirmation by text message
FR-3	**User engagement**	Picture capture and upload Image quality Evaluation Provoking visuals for precision
FR-4	**Disease prediction**	Digital Image processing Proper Disease prediction
FR-5	**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**	The application provides simple and clear functionality such as image capturing and earlier disease diagnosis besides alternative fertilizer suggestion.
NFR-2	**Security**	To safeguard against threats including illegal access,the application is designed in such a way that it ensures the privacy of the gathered user information.
NFR-3	**Reliability**	The model is trained with different visuals for a certain ailment, which leads to more accurate assessment of a disease, thereby making the systemmore reliable.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

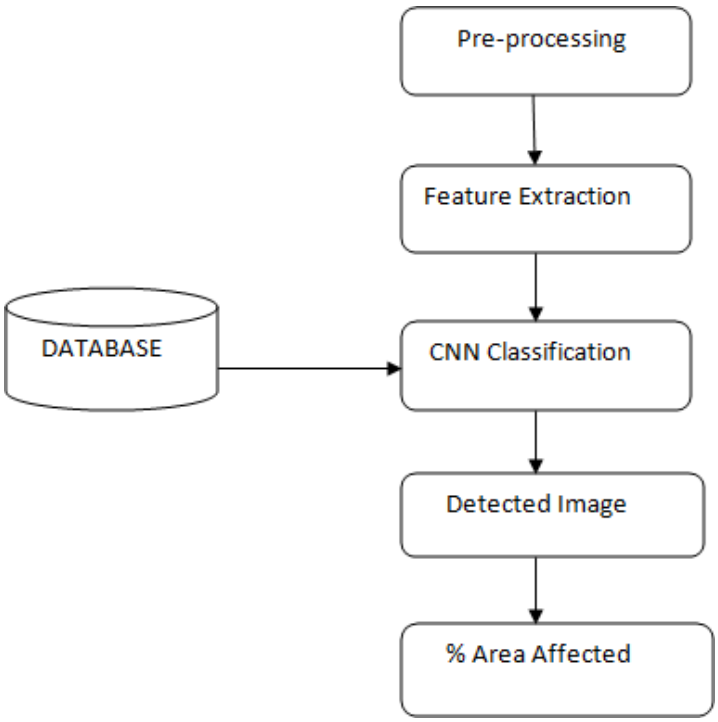


5.2 Solution & Technical Architecture

Crop disease in plants is predicted and suitable fertilizer is recommended for better yield. The images of the diseased plants are obtained and it is preprocessed against the dataset of diseased plants. Deep Learning Algorithm (CNN) is used to process the images and then it is evaluated. Then a model is built on the evaluations, it is then trained using no. of. inputs and predictions are given to the users which subsequently helps in recommending the fertilizers.

The Convolutional layers are used to classify and process the images and further helps in recommending the fertilizers. The image classification steps are:

- Image acquisition
- Preprocessing
- Segmentation
- Disease Prediction
- Fertilizer Recommendation



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Login/Signup	USN-1	As a user, I can login or create my account by providing the credentials like email-id, password, residential address and crop cultivated.	I can access my account	High	Sprint-1
Customer (Mobile user)	Home Page	USN-2	As a registered user, I can capture/upload the crop/plant image.	I am in Home/user page	High	Sprint-2
Customer (Mobile user)	Dashboard	USN-3	I can able to know the disease predicted.	Alter/Notification in home screen	High	Sprint-3
Customer (Mobile user)	Notifications	USN-4	I can also get the recommendation of the specific fertilizer to be used.	I will receive messages	High	Sprint-4
Customer (Mobile user)	Primary Mail	USN-5	Besides I am able to get an alternative fertilizes suggestion in uncertain cases where some fertilizers might be unavailable.	I will receive email	Low	Sprint-5

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint 1

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points (Total)	Priority	Team Members
Sprint-1	Model Creation and Training		Create a model which can classify diseased plant and train the model for accurate results	8	High	Ranjani Santhiya Snehakarthika Manoj
	Model Creation and Training		Create a model which can classify diseased plants from the rest of the healthy plants	2	High	Ranjani Santhiya Snehakarthika Manoj

Sprint 2 and 3

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points (Total)	Priority	Team Members
Sprint-2	Model Creation and Training		Create a model which can classify diseased plants from the healthy plant	6	High	Ranjani Santhiya Snehakarthika Manoj
	Registration	USN-1	As a user, I can register by entering my email, password, and confirming my password or viaOAuth API	3	Medium	Ranjani Santhiya Snehakarthika Manoj
	Upload page	USN-2	As a user, I will be redirected to a page where I can upload my pictures of crops	4	High	Ranjani Santhiya Snehakarthika Manoj
	Suggestion results	USN-3	As a user, I can view the results and then obtain the suggestions provided by the ML model	4	High	Ranjani Santhiya Snehakarthika Manoj.
	Base Flask App		A base Flask web app must be created as an interface for the ML model	2	High	Ranjani Santhiya Snehakarthika Manoj
Sprint-3	Login	USN-4	As a user/admin/shopkeeper, I can log into the application by entering email & password	2	High	Ranjani Santhiya Snehakarthika Manoj.
	User Dashboard	USN-5	As a user, I can view the previous results and history	3	Medium	Ranjani Santhiya Snehakarthika Manoj
	Integration		Integrate Flask, CNN model with MySql DB	5	Medium	Ranjani Santhiya Snehakarthika Manoj

Sprint 4

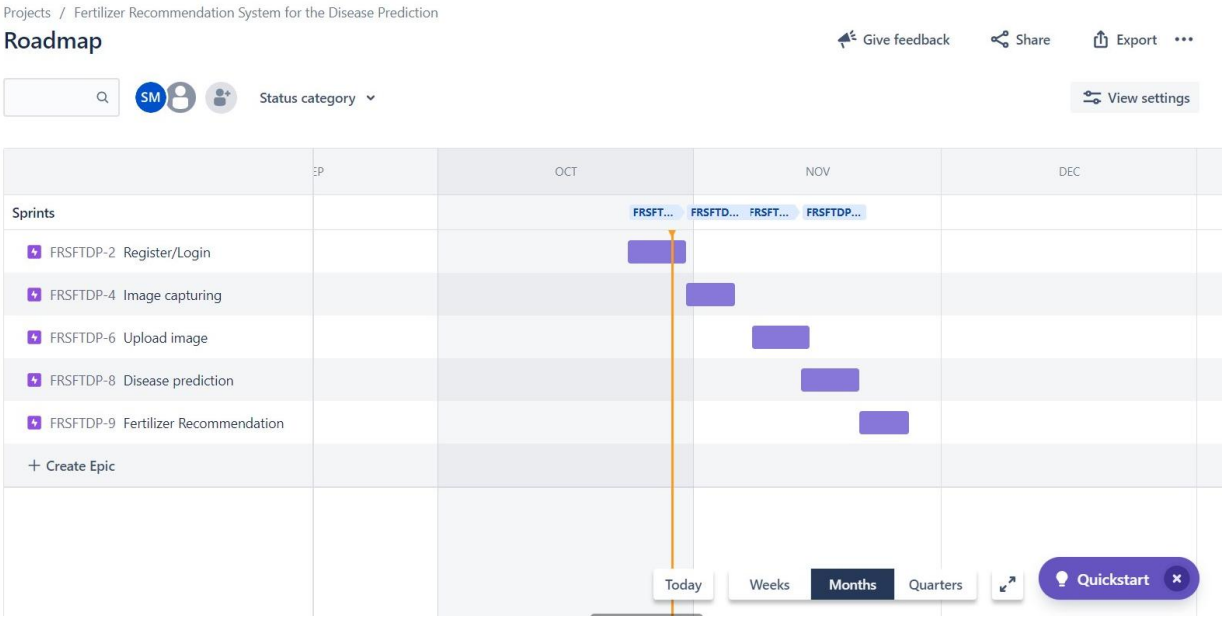
Sprint-4	Dashboard (Admin)	USN-6	As an admin, I can view other user details and uploads for other purposes	2	Medium	Ranjani Santhiya Snehakarthika Manoj
	Dashboard (Shopkeeper)	USN-7	As a shopkeeper, I can enter fertilizer products and then update the details if any	2	Low	Ranjani Santhiya Snehakarthika Manoj

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	10	30 Oct 2022
Sprint-2	15	6 Days	31 Oct 2022	05 Nov 2022	15	06 Nov 2022
Sprint-3	15	6 Days	07 Nov 2022	12 Nov 2022	15	13 Nov 2022
Sprint-4	12	6 Days	14 Nov 2022	19 Nov 2022	10	20 Nov 2022

6.3 Reports From JIRA

Roadmap



7. CODING & SOLUTIONING

7.1 Python – app_feretilizer.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

```html
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Fertilizer Recommented System and Disease Prediction</title>
  <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
  <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
  <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
  <link href="{{ 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>

   <nav class="navbar navbar-dark bg-dark">
      <div class="container">
         <a class="navbar-brand" href="#"><center><b>CROP DISEASE PREDICTION</b></center></a>
      </div>
   </nav>
   <div class="container">
      <div id="content" style="margin-top:2em">
            <div class="container">
             <div class="row">
                     <div class="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>
                              <img src="http://www.geocities.ws/senthilirtt/First.jpg"
height="50%",width="20%">
                     </div>
                     <div class="col-sm-6">
                              <div>
                                       <h4>Upload Image Here To Identify the Plant Disease</h4>
                     <form action = "http://localhost:5000/" id="upload-file" method="post"
enctype="multipart/form-data">
                              <label for="imageUpload" class="upload-label">
                                       Choose...
                              </label>
                              <input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg">
                     </form>


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

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

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

                </div>
                    </div>

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

<footer>
    <script src="{{ 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: 5px solid #F8F8F8;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
    margin-top: 1em;
    margin-bottom: 1em;
}

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

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

.upload-label{
    display: inline-block;
    padding: 12px 30px;
    background: #39D2B4;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

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

.loader {
```

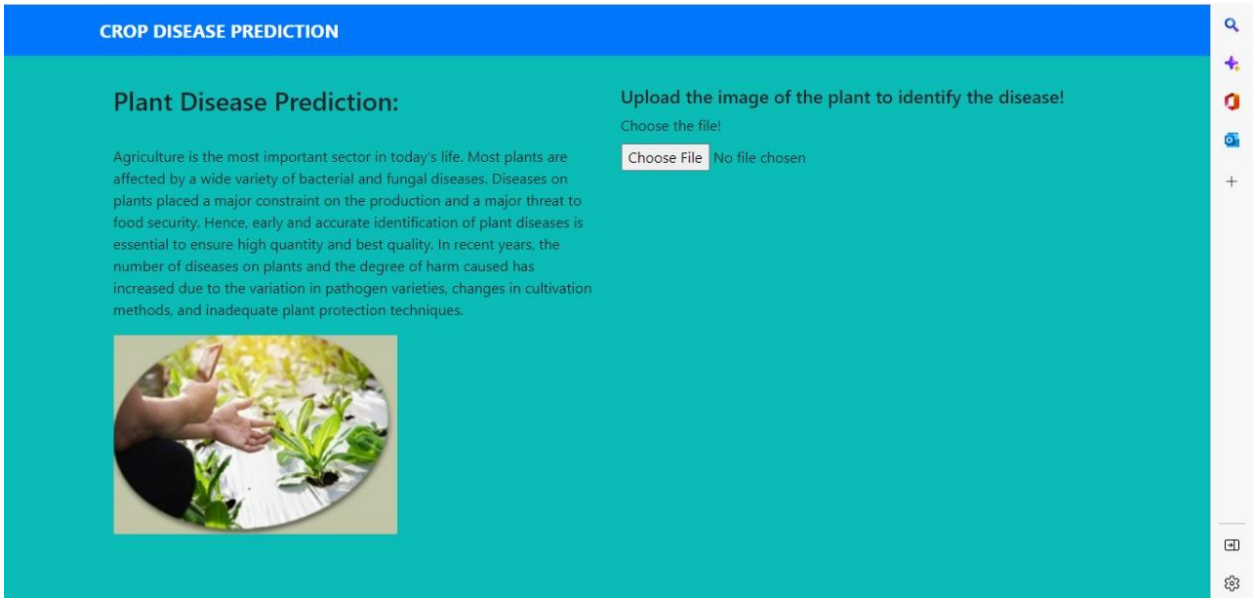
```
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #3498db; /* Blue */
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;
}

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

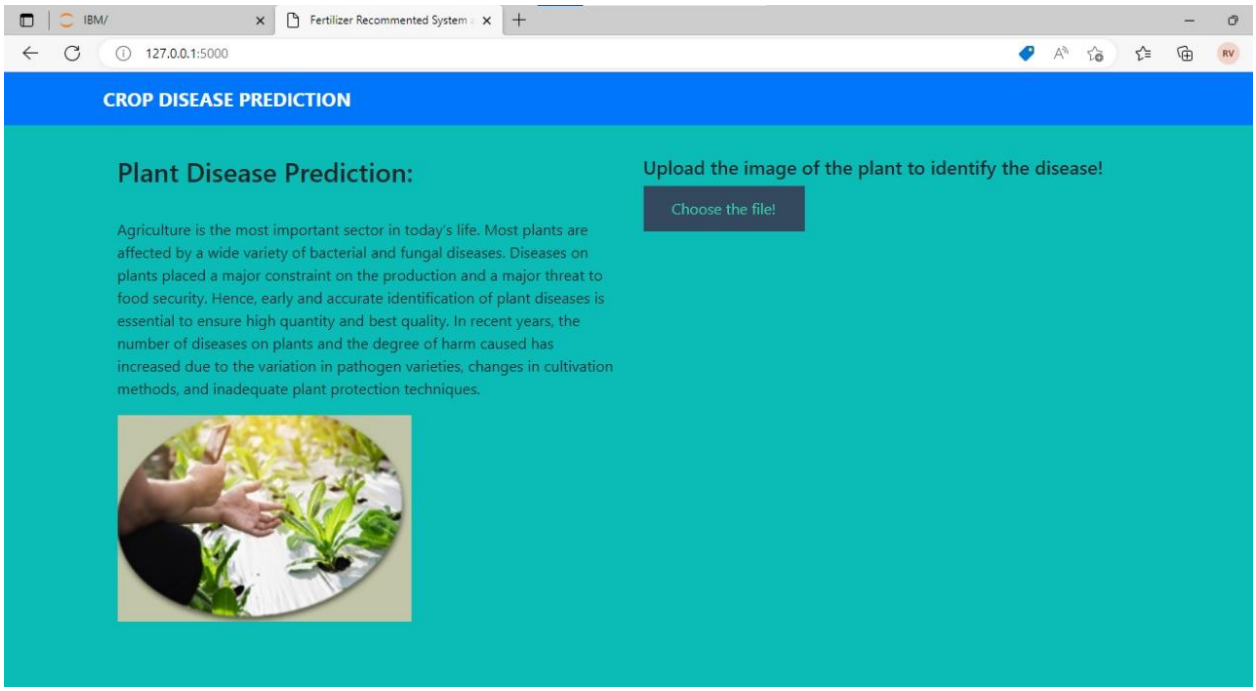
7.4 main.js

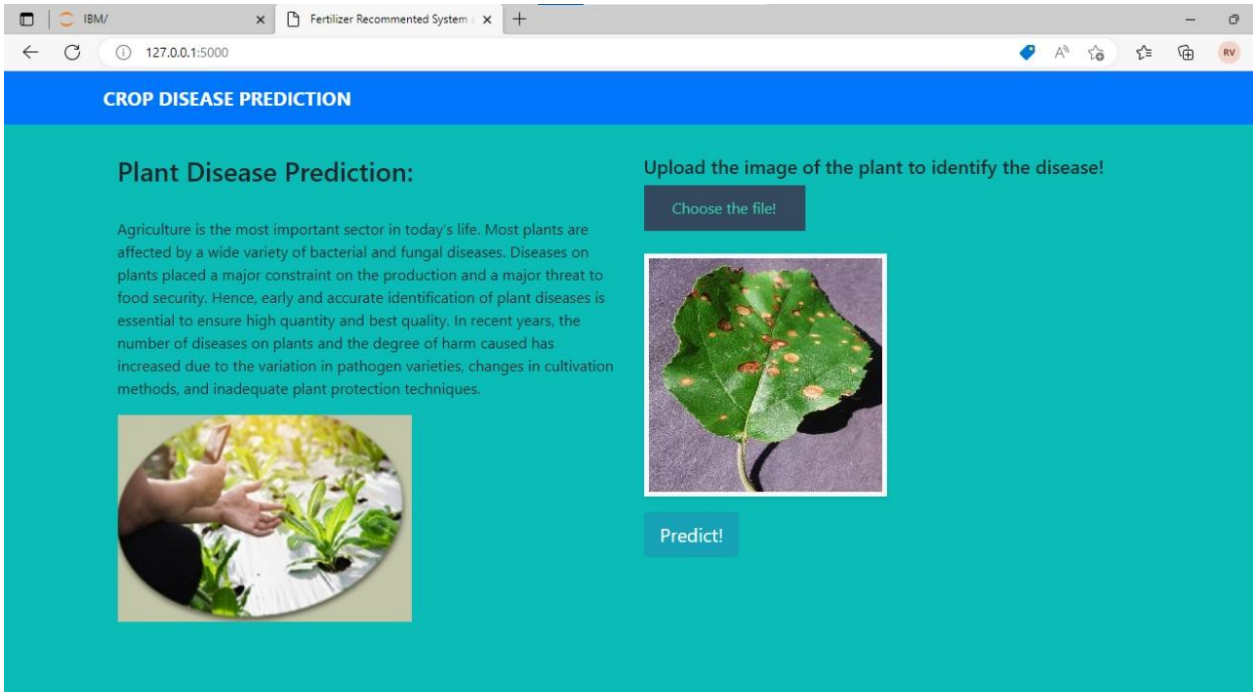
```
$(document).ready(function () {
    // Init
    $('.image-section').hide();
    $('.loader').hide();
    $('#result').hide();

    // 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!');
        },
    });
});

});
```

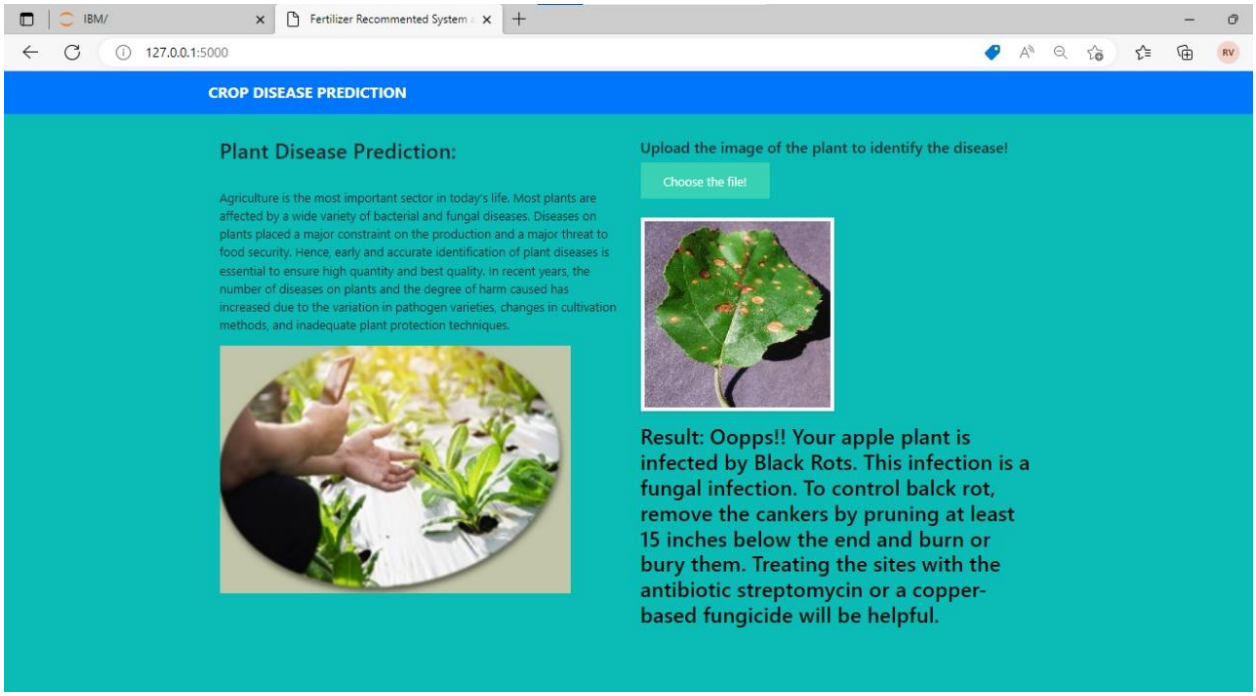
INPUT:






IBM/
Fertilizer Recommented System
127.0.0.1:5000
CROP DISEASE PREDICTION
Plant Disease Prediction:
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.
Upload the image of the plant to identify the disease!
Choose the file!
Predict!

OUTPUT:


IBM/
Fertilizer Recommented System
127.0.0.1:5000
CROP DISEASE PREDICTION
Plant Disease Prediction:
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.
Upload the image of the plant to identify the disease!
Choose the file!
Result: Oopps!! Your apple plant is infected by Black Rots. This infection is a fungal infection. To control balck rot, remove the cankers by pruning at least 15 inches below the end and burn or bury them. Treating the sites with the antibiotic streptomycin or a copper-based fungicide will be helpful.

8. TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)
Dataset_TC_001	Dataset	Data	assemble all the fruit and vegetable datasets required 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
Preprocessing_TC_002	Dataset	Data	Preprocessing will be carried out on the sample of images that was collected.	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 propoerly modified for the identification	Working as expected	Pass	NIL	Y
User interface_TC_003	Functional	Home page	Ensure that the user can access the application's main 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
Model building_TC_004	Functional	Deep Learning Model	Model building for disease predicton	Pre-processed dataset	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
Testing_TC_005	Functional	Deep Learning Model	Using the test dataset, evaluate the model.	deep learning model	1. Fit the test data to the model 2.Calculate the accuracy	Images from Dataset	Accuracy over 85%	Working as expected	Pass	NIL	Y
Prediction_TC_006	Functional	Deep Learning Model	predict the disease accurately based on training dataset	Trained model to predict disease	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	Disease identification	Working as expected	Pass	NIL	Y
Visualization_TC_00	Functional	Deep Learning Model	After identifying disease recommend appropriate fertilizer to cure the disease	Disease should be predicted	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	Fertilizer suggestion	Working as expected	Pass	NIL	Y

8.2 User Acceptance Testing

1. Purpose of Document

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. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
Leaf Spots	10	4	2	3	19
Mosaic Leaf Pattern	9	6	3	0	24
Blights	4	5	2	1	12
Yellow leaves	11	4	3	20	38
Fruit rots	3	2	1	1	6
Misshapen leaves	2	7	1	0	10
Fruits spots	5	4	2	1	11
Totals	44	31	13	13	120

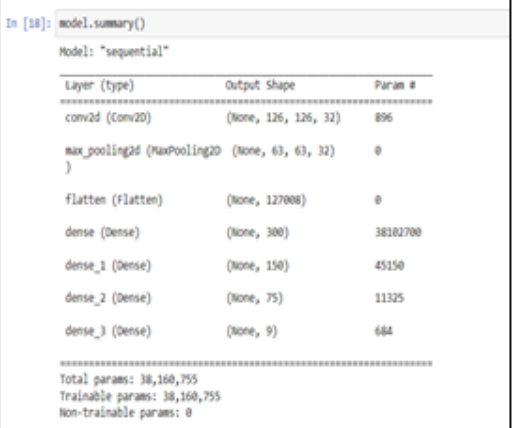
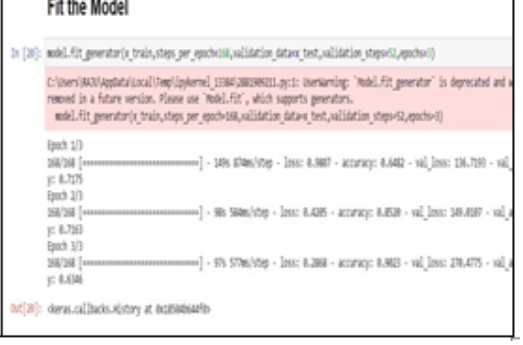
3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Leaf spots	18	0	0	18
Fruit spots	5	0	0	5
Mosaic leaf pattern	43	0	0	43
Blights	2	0	0	2
Misshapen Leaves	25	0	0	25
Yellow leaves	7	0	0	7
Fruit rots	9	0	0	9

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

i. One farm management approach which is precision agriculture can help farmers grow more crops with fewer resources. Precision agriculture powered by AI could become the next big thing in farming. Precision farming combines the best soil management practices, variable rate technology, and the most effective data management practices to help farmers maximize yields and minimize spending.

ii. AI can provide farmers with real-time insights from their fields, allowing them to identify areas that need irrigation, fertilization, or pesticide treatment. Also, innovative farming practices like vertical agriculture may help increase food production while minimizing the use of resources. The result is reduced use of herbicides, better harvest quality, higher profits, and significant cost savings.

iii. Agricultural work is hard, and labor shortages in this industry are nothing new. Farmers can solve this problem with the help of automation. Driverless tractors, smart irrigation and fertilizing systems, smart spraying, vertical farming software, and AI-based robots for harvesting are some examples of how farmers can get the work done without having to hire more people. Compared with any human farm worker, AI-driven tools are faster, harder, and more accurate.

iv. Farmers need to understand that AI is only an advanced part of simpler technologies for processing, gathering, and monitoring field data. AI requires a proper technology infrastructure for it to work. That is why even those farms that already have some technology in place can find it difficult to move forward.

v. This is also a challenge for software companies. They should approach farmers gradually, giving them simpler technology first, such as an agriculture trading platform. Once farmers get used to a less complicated solution, it will be reasonable to step it up and offer something else, including AI features.

vi. The agricultural sector in developing countries is different from the agricultural sector in Western Europe and the US. Some regions could benefit from artificial intelligence agriculture, but it may be hard to sell such technology in areas where agricultural technology is not common. Farmers will most likely need help adopting it.

vii. Therefore, tech companies hoping to do business in regions with emerging agricultural economies might need to take a proactive approach. In addition to providing their products, they will have to provide training and ongoing support for farmers and agribusiness owners who are ready to take on innovative solutions.

viii. Since there are no clear policies and regulations around the use of AI not just in agriculture but in general, precision agriculture and smart farming raises various legal issues that often remain

unanswered. Privacy and security threats like cyberattacks and data leaks may cause farmers serious problems. Unfortunately, many farms are vulnerable to these threats.

11. CONCLUSION

Different approaches and models of Deep Learning methods were explored and used in this project so that it can detect and classify plant diseases correctly through image processing of leaves of the plants. The procedure starts from collecting the images used for training, testing and validation to image preprocessing and augmentation and finally comparison of different pretrained models over their accuracy. Finally, at the end, our model detects and distinguishes between a healthy plant and different diseases and provides suitable remedies to cure the disease. This paper proposed and developed a system which uses plant leaf images to detect different types of disease in tomato crops, and provides appropriate fertilizer suggestions.

12. FUTURE SCOPE

The system successfully interprets various Diseases and is also capable of providing fertilizers suggestion for the respective disease. Furthermore, this system can be made more robust by incorporating more image dataset with wider variations like more than one leaf in a single image. An App could also be developed for the project which could make the work of the farmers easier. They could directly upload image on the app and it would tell the disease and the cure then and there. This would reduce the time and efforts. This project is limited to just one crop for now but in the future more crops and even flowers dataset can be added so that it is helpful for every agricultural need. Newer models can also be added and tried with time which may result in better accuracy and would make the model even faster.

13. APPENDIX

Source Code

Python – app_feretilizer.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

```html
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Fertilizer Recommented System and Disease Prediction</title>
  <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
  <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
  <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
```

```
   <link href="{{ 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>
      <nav class="navbar navbar-dark bg-dark">
      <div class="container">
         <a class="navbar-brand" href="#"><center><b>CROP DISEASE
PREDICTION</b></center></a>
      </div>
   </nav>
   <div class="container">
      <div id="content" style="margin-top:2em">
            <div class="container">
             <div class="row">
                    <div class="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>
                              <img src="http://www.geocities.ws/senthilirtt/First.jpg"
height="50%",width="20%">
                    </div>
                    <div class="col-sm-6">
                          <div>
                                <h4>Upload Image Here To Identify the Plant Disease</h4>
```

```
                    <form action = "http://localhost:5000/" id="upload-file" method="post"
enctype="multipart/form-data">
                        <label for="imageUpload" class="upload-label">
                            Choose...
                        </label>
                        <input type="file" name="image" id="imageUpload" accept=".png, .jpg,
.jpeg">
                    </form>

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

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

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

            </div>
                    </div>

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

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

main.css

```css
.img-preview {
    width: 256px;
    height: 256px;
    position: relative;
    border: 5px solid #F8F8F8;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
    margin-top: 1em;
    margin-bottom: 1em;
}

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

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

.upload-label{
    display: inline-block;
    padding: 12px 30px;
    background: #39D2B4;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

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

.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #3498db; /* Blue */
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;
}

@keyframes spin {
    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'C:\Users\HP\OneDrive\Desktop\Ranjani\IBM\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'C:\Users\HP\OneDrive\Desktop\Ranjani\IBM\Dataset
Plant Disease\fruit-dataset\fruit-dataset\train',target_size=(128,128),
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPool2D,Flatten
model=Sequential()
```

Adding CNN 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/HP/OneDrive/Desktop/Ranjani/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/HP/OneDrive/Desktop/Ranjani/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/HP/OneDrive/Desktop/Ranjani/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/HP/OneDrive/Desktop/Ranjani/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/HP/OneDrive/Desktop/Ranjani/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/HP/OneDrive/Desktop/Ranjani/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'C:/Users/HP/OneDrive/Desktop/Ranjani/IBM/Dataset Plant Disease/veg-dataset/veg-dataset/train_set',target_size=(128,128),batch_size=16,class_mode='categorical')
x_test=test_datagen.flow_from_directory(r'C:/Users/HP/OneDrive/Desktop/Ranjani/IBM/Dataset Plant Disease/veg-dataset/veg-dataset/test_set',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 CNN 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,epochs=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/HP/OneDrive/Desktop/Ranjani/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/HP/OneDrive/Desktop/Ranjani/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/HP/OneDrive/Desktop/Ranjani/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/HP/OneDrive/Desktop/Ranjani/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/HP/OneDrive/Desktop/Ranjani/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/HP/OneDrive/Desktop/Ranjani/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/HP/OneDrive/Desktop/Ranjani/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/HP/OneDrive/Desktop/Ranjani/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/HP/OneDrive/Desktop/Ranjani/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]])
```

GitHub Link

https://github.com/IBM-EPBL/IBM-Project-3236-1658507426

Project Demo Link

https://drive.google.com/file/d/11XN7OTR1WKGhvsTAvikidG7SGwx4arcP/view?usp=share_link