Project Report

1.INTRODUCTION

1.1 Project Overview

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

1.2 Purpose

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

2.LITERATURE SURVEY

2.1 Existing problem

[1] The proposed method uses SVM to classify tree leaves, identify the disease and suggest the fertilizer. The proposed method is compared with the existing CNN based leaf disease prediction. The proposed SVM technique gives a better result when compared to existing CNN. For the same set of images, F-Measure for CNN is 0.7and 0.8 for SVM, the accuracy of identification of leaf disease of CNN is 0.6 and SVM is 0.8.

Advantages : The prediction and diagnosing of leaf diseases are depending on the segmentation such as segmenting the healthy tissues from diseased tissues of leaves.

Disadvantages : This further research is implementing the proposed algorithm with the existing public datasets. Also, various segmentation algorithms can be implemented to improve accuracy. The proposed algorithm can be modified further to identify the disease that affects the various plant organs such as stems and fruits.

[2] Detection of Leaf Diseases and Classification using Digital Image Processing International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), IEEE, 2017.

Advantages: The system detects the diseases on citrus leaves with 90% accuracy.

Disadvantages: System only able to detect the disease from citrus leaves. The main objective of this paper is image analysis & classification techniques for detection of leaf diseases and classification. The leaf image is firstly preprocessed and then does the further work. K-Means Clustering used for image segmentation and then system extract the GLCM features from disease detected images. The disease classification done through the SVM classifier.

2.2 References

[1] SVM – Praline B.Padol, Anjali A.Yadav, 2016 conference on advances in Signal processing(CASP).

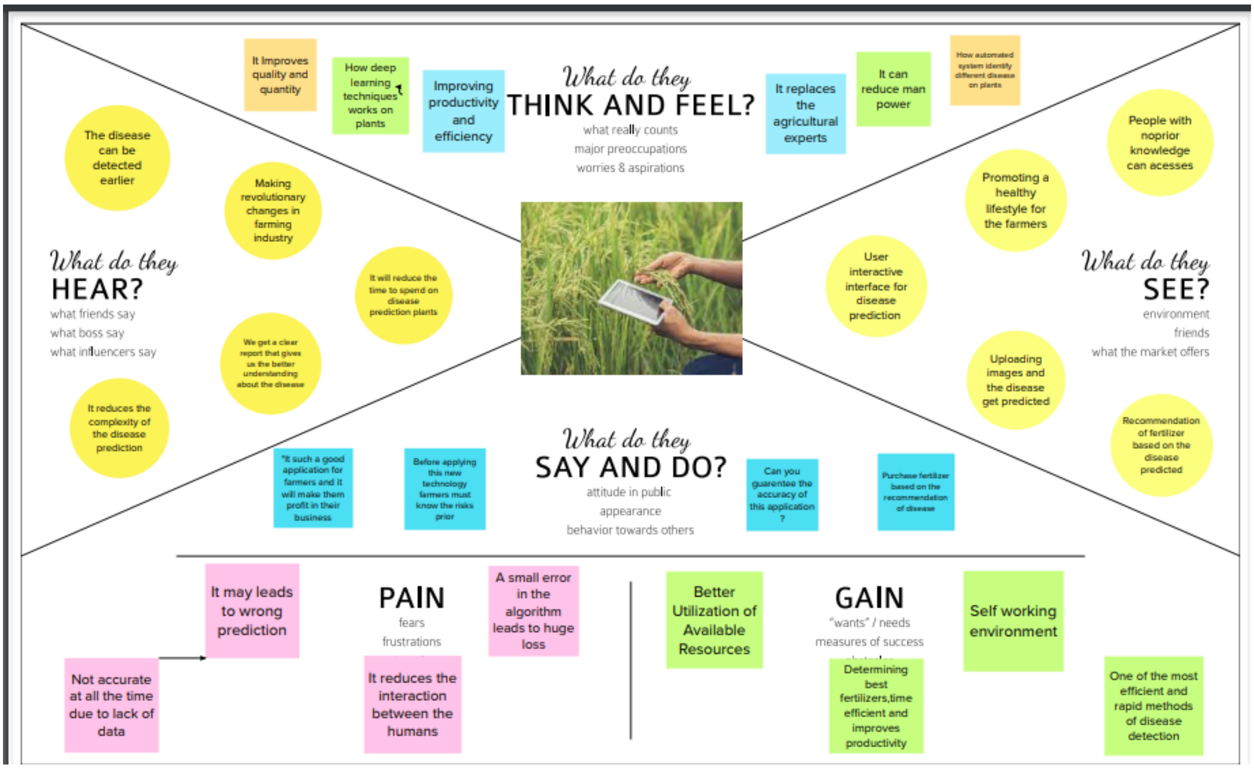
[2] Semi-automatic leaf disease detection and classification system for soybean culture IET Image Processing, 2018

2.3 Problem Statement Definitions

The global agricultural powerhouse is considered as a key for human lives. In the recent years due to the presence of multiple fertilizers in the market, so the farmers get confused also there is lots of unpredicted diseases in the farming. To overcome this problem, we recommend our trained model to predict the disease exactly and to recommend the fertilizer to cure the disease.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

It Improves quality and quantity

How deep learning techniques works on plants

Improving productivity and efficiency

It replaces the agricultural experts

It can reduce man power

How automated system identify different disease on plants

What do they HEAR?
what friends say
what boss say
what influencers say

The disease can be detected earlier

Making revolutionary changes in farming industry

It will reduce the time to spend on disease prediction plants

We get a clear report that gives us the better understanding about the disease

It reduces the complexity of the disease prediction

What do they SEE?
environment
friends
what the market offers

People with noprior knowledge can accesses

Promoting a healthy lifestyle for the farmers

User interactive interface for disease prediction

Uploading images and the disease get predicted

Recommendation of fertilizer based on the disease predicted

What do they SAY AND DO?
attitude in public
appearance
behavior towards others

"It such a good application for farmers and it will make them profit in their business

Before applying this new technology farmers must know the risks prior

Can you guarantee the accuracy of this application ?

Purchase fertilizer based on the recommendation of disease

PAIN
fears
frustrations

It may leads to wrong prediction

Not accurate at all the time due to lack of data

A small error in the algorithm leads to huge loss

It reduces the interaction between the humans

GAIN
"wants" / needs
measures of success

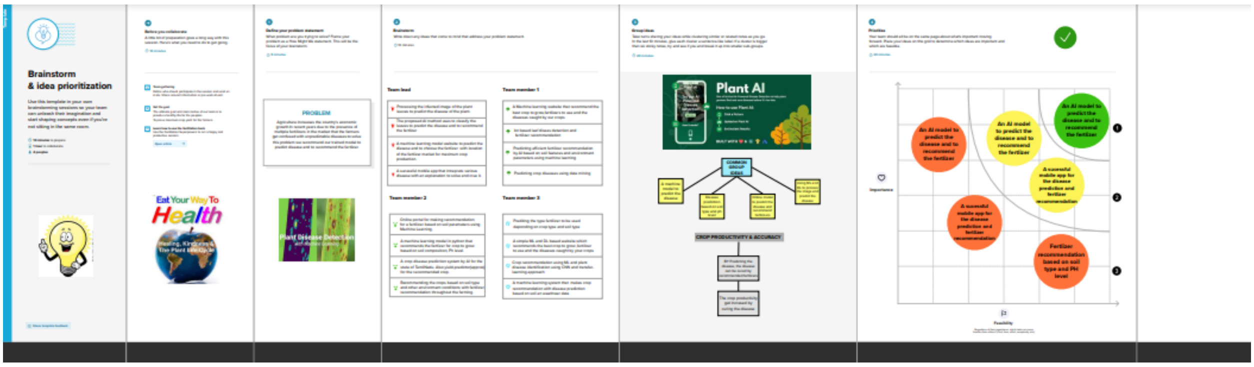
Better Utilization of Available Resources

Self working environment

Determining best fertilizers,time efficient and improves productivity

One of the most efficient and rapid methods of disease detection

3.2 Ideation & Brainstorming

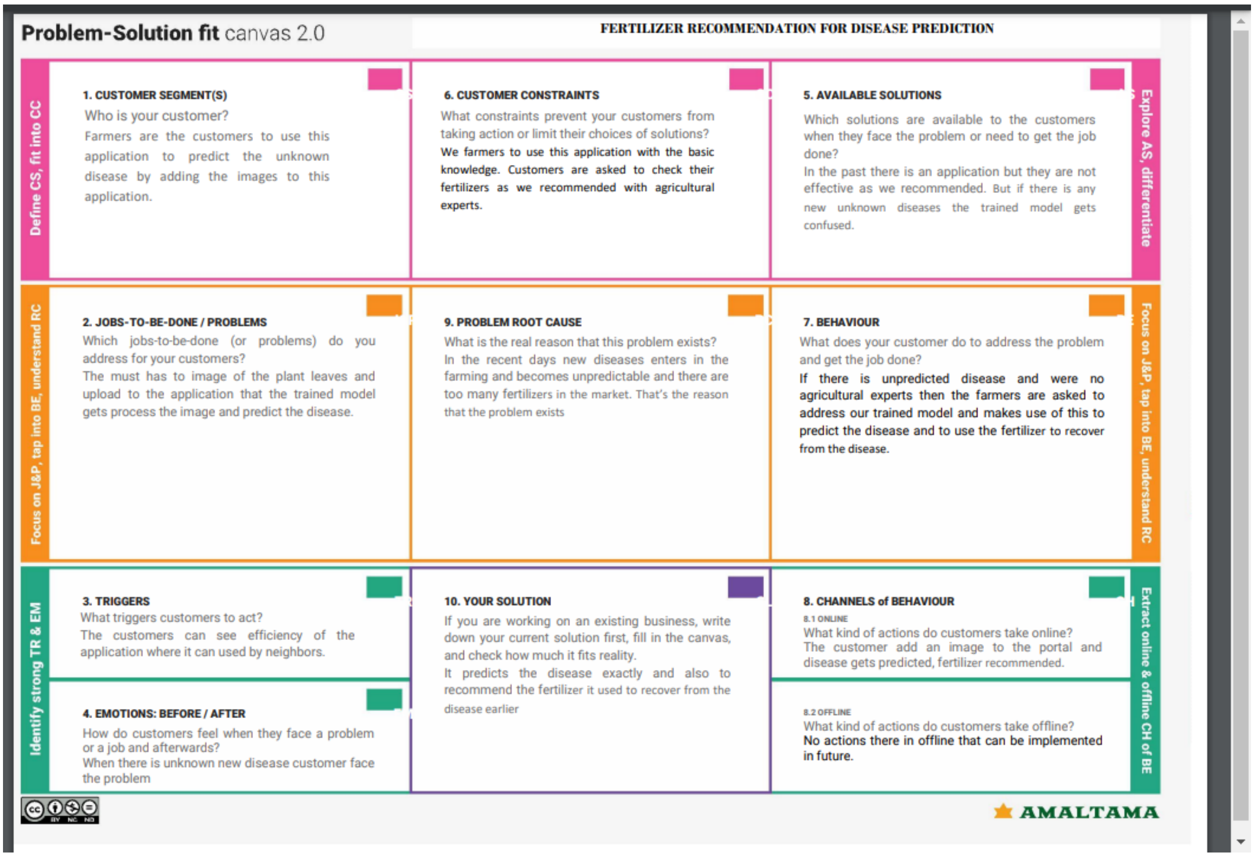

Brainstorm & idea prioritization
Eat Your Way To Health
PROBLEM
Plant Disease Detection
Plant AI
CROP PRODUCTIVITY & ACCURACY

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (problem to be solved)	- In the recent years due spread of multiple diseases in the farming and the availability of multiple fertilizers in the market the farmers get confused to predict the disease and to use the fertilizers.
2.	Idea / Solution Description	- When the user inputs an image of a diseased plant leaf, the application predicts the type of disease, display the result along with the little background about the disease and suggestions to cure it. - Deep learning techniques are used to identify the diseases and suggest appropriate fertilizer that can be taken for those diseases.

3.	Novelty / Uniqueness	1. The classification technique is divided into the following steps: a. Image acquisition b. Pre-processing 2. Instant solutions for farmer's queries.
4.	Social Impact / Customer satisfaction	1. Fertilizers are a supplements supply to soil nutrients, build up soil fertility in order to support plant nutrients and increase plants productivity. 2. Nowadays, artificial intelligence and sensor technology play a vital role in the agriculture field. The use of excess insecticides and fertilizers in farming poses a risk to human health.
5.	Business Model (Revenue Model)	1. Helpline support for resolving app related issues. 2. Service availability depends on the plan subscribed by the farmers.

3.4 Problem Solution Fit



Problem-Solution fit canvas 2.0

FERTILIZER RECOMMENDATION FOR DISEASE PREDICTION

Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

Who is your customer?

Farmers are the customers to use this application to predict the unknown disease by adding the images to this application.

6. CUSTOMER CONSTRAINTS

What constraints prevent your customers from taking action or limit their choices of solutions?

We farmers to use this application with the basic knowledge. Customers are asked to check their fertilizers as we recommended with agricultural experts.

5. AVAILABLE SOLUTIONS

Which solutions are available to the customers when they face the problem or need to get the job done?

In the past there is an application but they are not effective as we recommended. But if there is any new unknown diseases the trained model gets confused.

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS

Which jobs-to-be-done (or problems) do you address for your customers?

The must has to image of the plant leaves and upload to the application that the trained model gets process the image and predict the disease.

9. PROBLEM ROOT CAUSE

What is the real reason that this problem exists?

In the recent days new diseases enters in the farming and becomes unpredictable and there are too many fertilizers in the market. That's the reason that the problem exists

7. BEHAVIOUR

What does your customer do to address the problem and get the job done?

If there is unpredicted disease and were no agricultural experts then the farmers are asked to address our trained model and makes use of this to predict the disease and to use the fertilizer to recover from the disease.

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

Identify strong TR & EM

3. TRIGGERS

What triggers customers to act?

The customers can see efficiency of the application where it can used by neighbors.

4. EMOTIONS: BEFORE / AFTER

How do customers feel when they face a problem or a job and afterwards?

When there is unknown new disease customer face the problem

10. YOUR SOLUTION

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.

It predicts the disease exactly and also to recommend the fertilizer it used to recover from the disease earlier

8. CHANNELS of BEHAVIOUR

8.1 ONLINE

What kind of actions do customers take online?

The customer add an image to the portal and disease gets predicted, fertilizer recommended.

8.2 OFFLINE

What kind of actions do customers take offline?

No actions there in offline that can be implemented in future.

Extract online & offline CH of BE

4.REQUIREMENT ANALYSIS

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	No Registration needed.
FR-2	Image	Image of the affected plant leaf is needed to predict the disease
FR-3	PC / Desktop or Mobile	Device needed to run the application

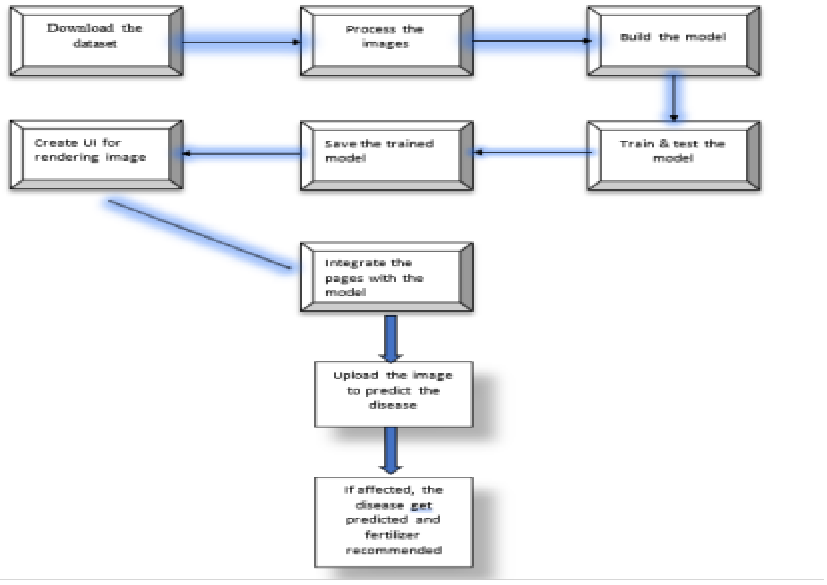
4.2 Non-Functional Requirement

Following are the non-functional requirements of the proposed solution.

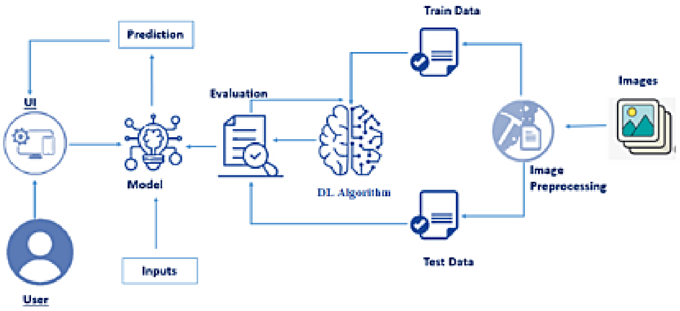
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Used to predict disease affected on plants
NFR-2	**Security**	No security needed while using this application
NFR-3	**Reliability**	Reliable on anytime anywhere with the environment
NFR-4	**Performance**	High performance to predict disease which is difficult to predict by the farmers
NFR-5	**Availability**	Available for customers on both web and mobile
NFR-6	**Scalability**	Scalable with predicting diseases which is known or defined earlier, but not for the new diseases

5.PROJECT DESIGN

5.1 Data Flow Diagram


Download the dataset
Process the images
Build the model
Train & test the model
Save the trained model
Create UI for rendering image
Integrate the pages with the model
Upload the image to predict the disease
If affected, the disease get predicted and fertilizer recommended

5.2 Solution & Technical Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Dashboard	USN-1	As a user, I can use this user-friendly application without any registration and login	I can access this dashboard	High	Sprint-2
	Predict page	USN-2	As a user, I can upload the image of the infected plant leaf	I can access this predict page	Medium	Sprint-3
	Solution	USN-3	As a user, I can get the solution with the trained model	I can access this on predict page	Medium	Sprint-3
Customer (Web user)	Dashboard	USN-1	As a user, I can use this user-friendly web application without any registration and login	I can access this dashboard	High	Sprint-2

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning

1. Planning is a crucial role in project management because it allows team members to schedule their time on the project.

2. This activity demonstrates how the team members assigned and completed various tasks!

3. In Project we can Split into the Four Step of Phrases are

➢ Phrase 1: Information Collection and Requirement Analysis
➢ Phrase 2: Project Planning and Developing Modules
➢ Phrase 3: Implementing the High Accuracy Machine Learning Algorithm to Perform
➢ Phrase 4: Deploying the Model on Cloud and Testing the Model and UI Performance

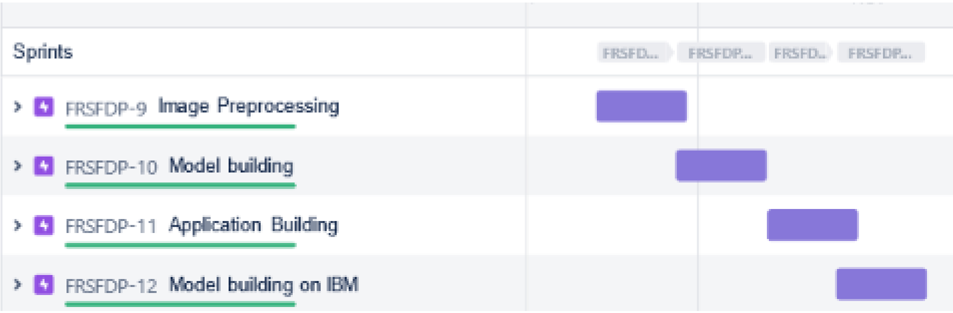
6.1 Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Image Processing	USN-1	As a user, I can process and retrieve the useful information about the images.	1	Low	Karthikeyan/ Shankar
Sprint-2	Model Building for Fruit Disease Prediction	USN-2	As a user, I can be able to predict fruit disease using this model.	1	Medium	Gokul Prasath/ Karthikeyan
Sprint-2	Model Building for Vegetable Disease Prediction	USN-3	As a user, I can be able to predict vegetable disease using this model.	2	Medium	Karthikeyan/ Karthikeyan
Sprint-3	Application Building	USN-4	As a user, I can see a web page for Fertilizers Recommendation System for Disease Prediction.	2	High	Gokul Prasath/ Karthikeyan/ Shankar/ Karthikeyan
Sprint-4	un the Model on IBM Cloud	USN-5	As a user, I can save the information about Fertilizer and crops on IBM cloud.	2	High	Gokul Prasath/

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	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	04 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	10 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	17 Nov 2022

6.3 Reports From JIRA

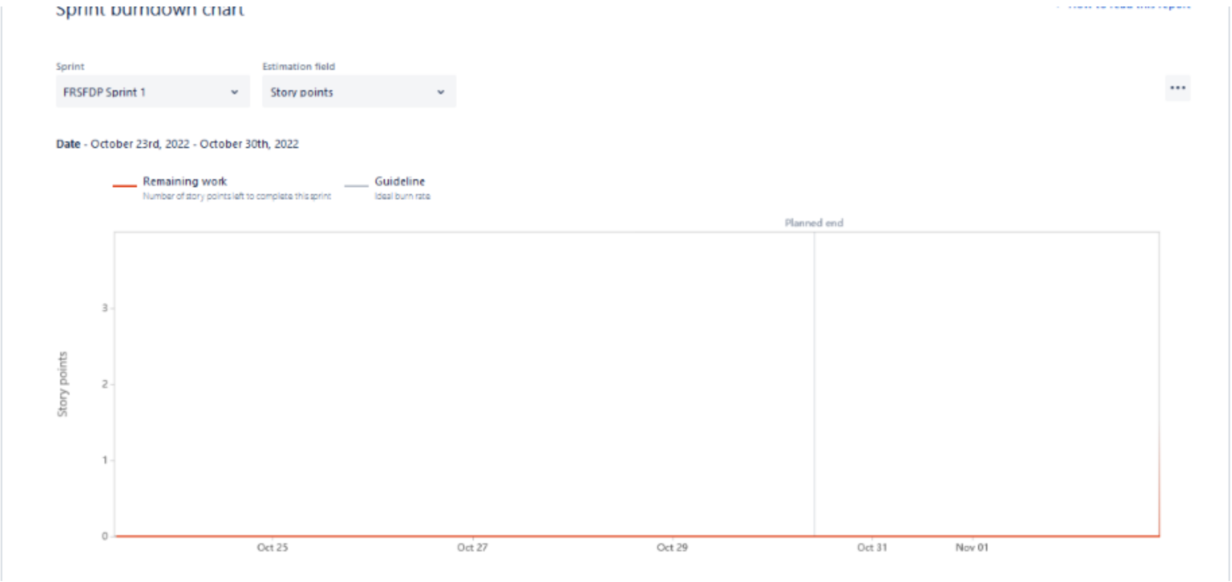


FRSFDP-9 Image Preprocessing

FRSFDP-10 Model building

FRSFDP-11 Application Building

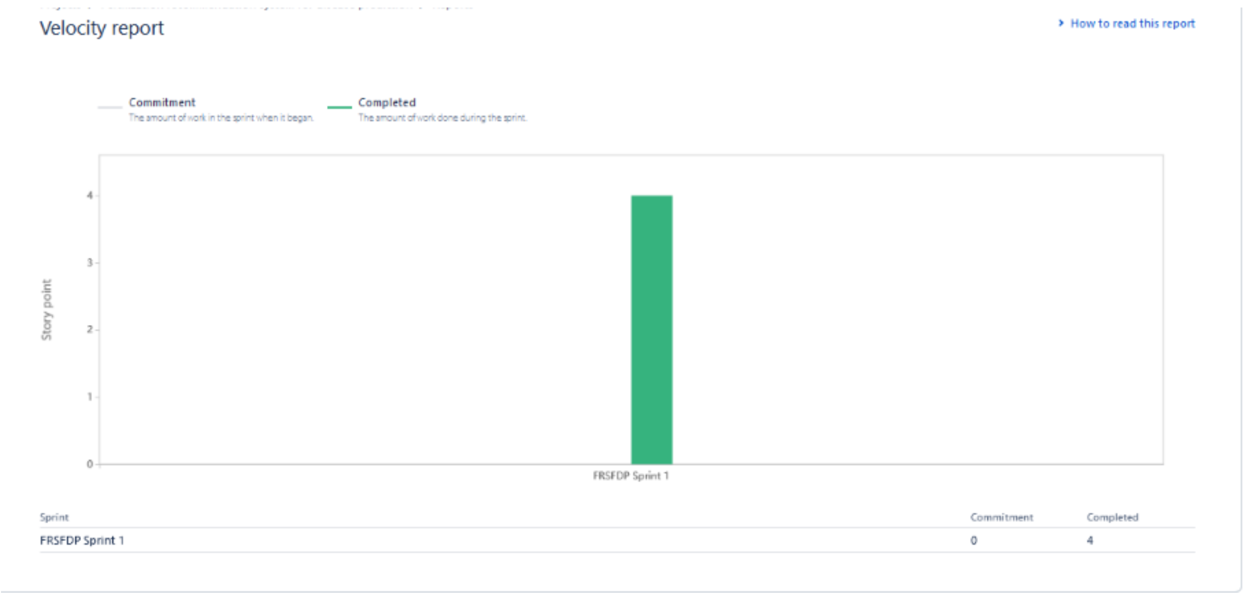
FRSFDP-12 Model building on IBM


Sprint burndown chart

Sprint: FRSFDP Sprint 1

Estimation field: Story points

Date - October 23rd, 2022 - October 30th, 2022



Velocity report

› How to read this report

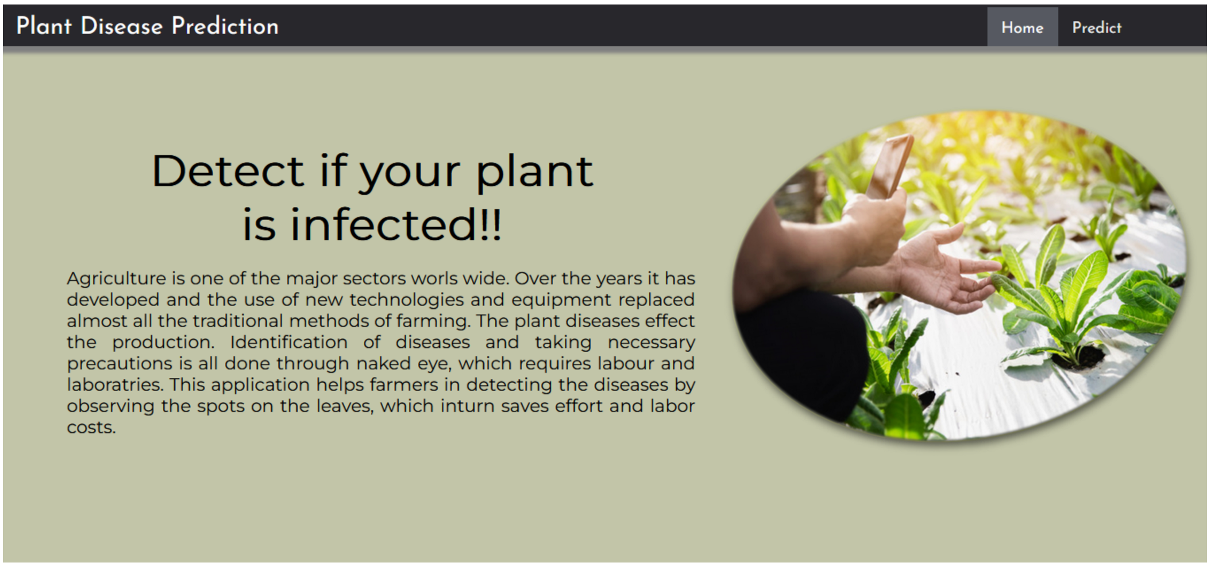
Commitment
The amount of work in the sprint when it began.
Completed
The amount of work done during the sprint.
Story point
4
3
2
1
0
FRSFDP Sprint 1

Sprint	Commitment	Completed
FRSFDP Sprint 1	0	4

7.CODING & SOLUTIONING

Home

Basic information about the website that we have created and deployed.

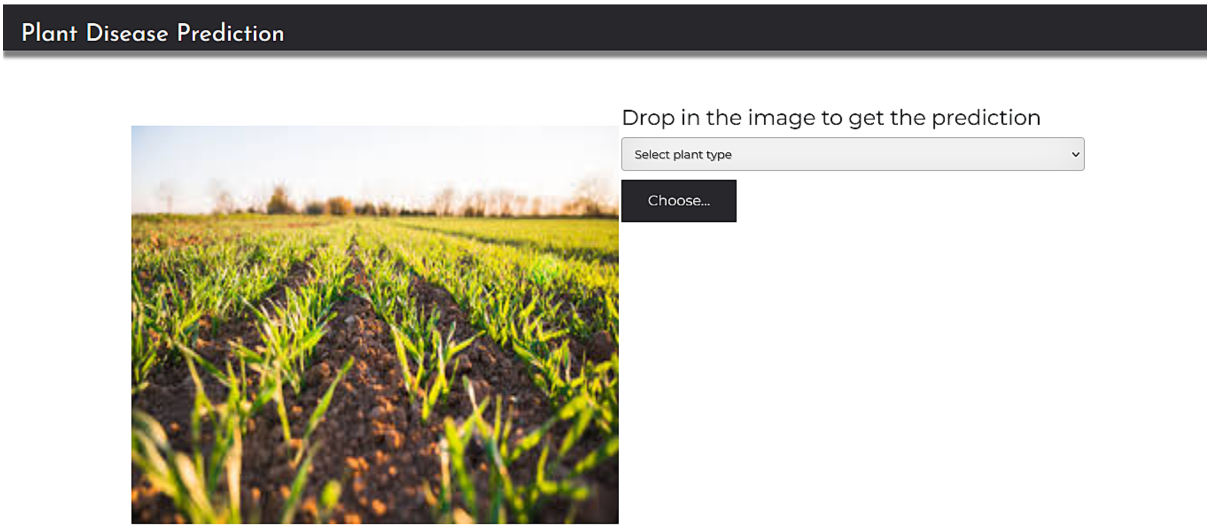


Detect if your plant is infected!!

Agriculture is one of the major sectors worls wide. Over the years it has developed and the use of new technologies and equipment replaced almost all the traditional methods of farming. The plant diseases effect the production. Identification of diseases and taking necessary precautions is all done through naked eye, which requires labour and laboratries. This application helps farmers in detecting the diseases by observing the spots on the leaves, which inturn saves effort and labor costs.

Predict

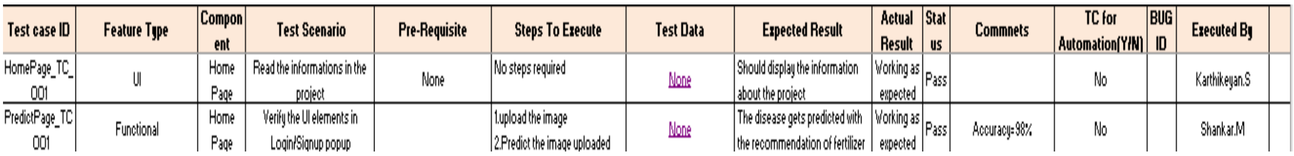
In the prediction page the infected plant leaf gets predicted and the precautions are measured.



Drop in the image to get the prediction

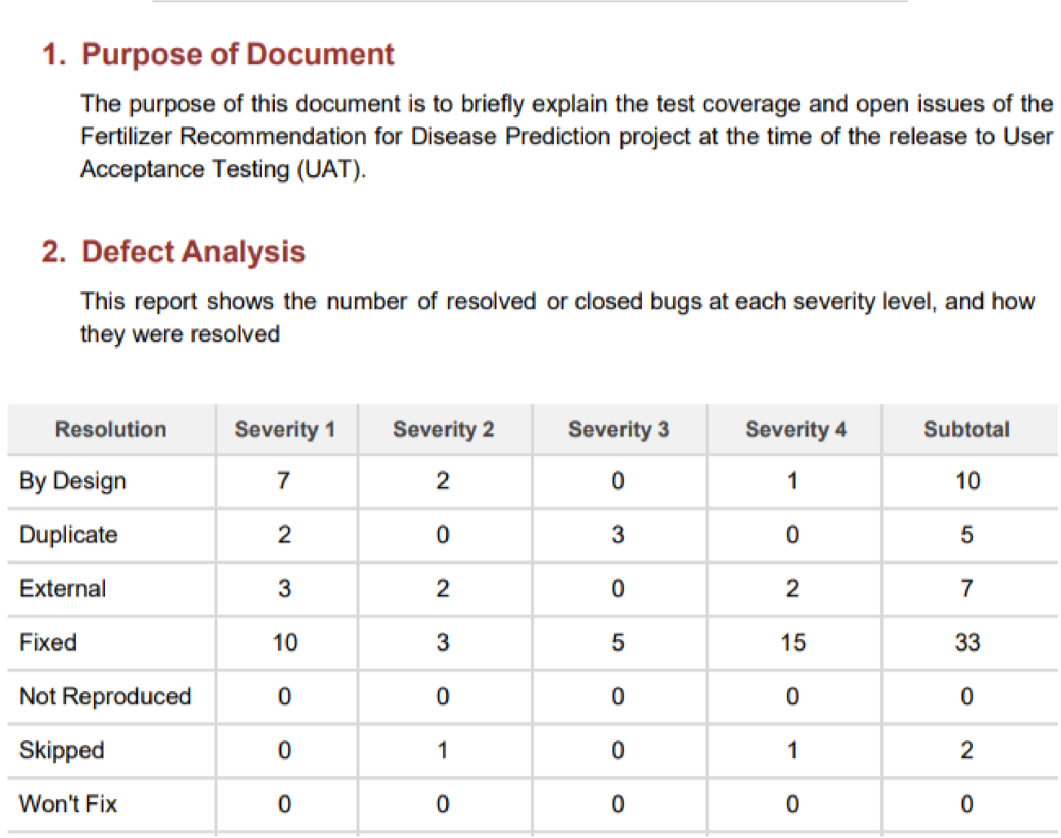
8. **TESTING**

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)	BUG ID	Executed By
HomePage_TC_001	UI	Home Page	Read the informations in the project	None	No steps required	None	Should display the information about the project	Working as expected	Pass		No		Karthikeyan.S
PredictPage_TC001	Functional	Home Page	Verify the UI elements in Login/Signup popup		1.upload the image 2.Predict the image uploaded	None	The disease gets predicted with the recommendation of fertilizer	Working as expected	Pass	Accuracy=98%	No		Shankar.M

8.2 User Acceptance Testing



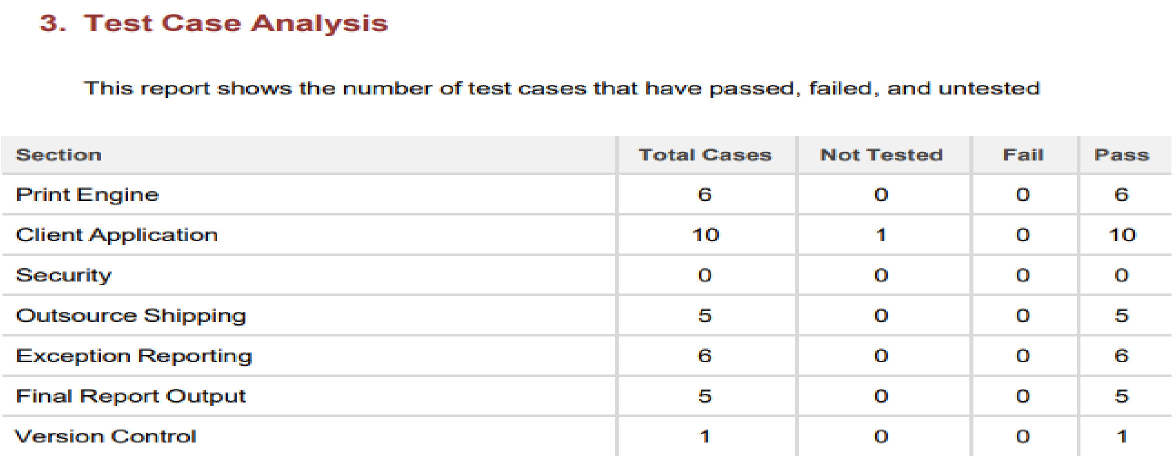
1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Fertilizer Recommendation for Disease Prediction project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	7	2	0	1	10
Duplicate	2	0	3	0	5
External	3	2	0	2	7
Fixed	10	3	5	15	33
Not Reproduced	0	0	0	0	0
Skipped	0	1	0	1	2
Won't Fix	0	0	0	0	0



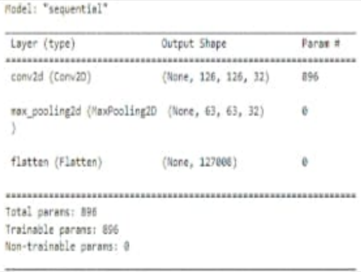
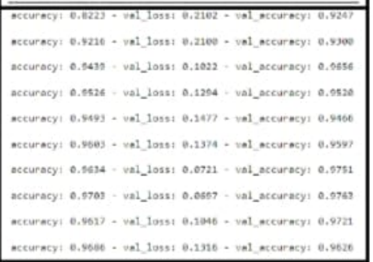
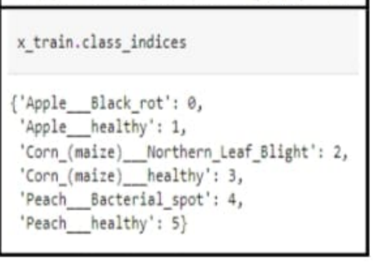
3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	6	0	0	6
Client Application	10	1	0	10
Security	0	0	0	0
Outsource Shipping	5	0	0	5
Exception Reporting	6	0	0	6
Final Report Output	5	0	0	5
Version Control	1	0	0	1

9.**RESULTS**

9.1Performance Metrics

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Model value - 896	 Model: "sequential" Layer (type) Output Shape Param # conv2d (Conv2D) (None, 126, 126, 32) 896 max_pooling2d (MaxPooling2D) (None, 63, 63, 32) 0 flatten (Flatten) (None, 127008) 0 Total params: 896 Trainable params: 896 Non-trainable params: 0
2.	Accuracy	Training Accuracy – 0.9686 Validation Accuracy – 0.9626	 accuracy: 0.8223 - val_loss: 0.2102 - val_accuracy: 0.9247 accuracy: 0.9216 - val_loss: 0.2100 - val_accuracy: 0.9300 accuracy: 0.9439 - val_loss: 0.1022 - val_accuracy: 0.9656 accuracy: 0.9526 - val_loss: 0.1294 - val_accuracy: 0.9520 accuracy: 0.9493 - val_loss: 0.1477 - val_accuracy: 0.9466 accuracy: 0.9603 - val_loss: 0.1374 - val_accuracy: 0.9597 accuracy: 0.9634 - val_loss: 0.0721 - val_accuracy: 0.9751 accuracy: 0.9703 - val_loss: 0.0667 - val_accuracy: 0.9743 accuracy: 0.9617 - val_loss: 0.1046 - val_accuracy: 0.9721 accuracy: 0.9686 - val_loss: 0.1316 - val_accuracy: 0.9626
3.	Confidence Score (Only Yolo Projects)	Class Detected Confidence Score - 96	 x_train.class_indices {'Apple___Black_rot': 0, 'Apple___healthy': 1, 'Corn_(maize)___Northern_Leaf_Blight': 2, 'Corn_(maize)___healthy': 3, 'Peach___Bacterial_spot': 4, 'Peach___healthy': 5}

10.ADVANTAGES & DISADVANTAGES

Advantages:

✓ The system helps to compute the disease severity.

✔ It allows us to predict which crops would be appropriate for a given climate. Using the weather and disease related data sets, the crop quality can also be improved.

✔ The prediction and diagnosing of leaf diseases are depending on the segmentation such as segmenting the healthy tissues from diseased tissues of leaves.

Disadvantages:

✔ Due to the changing climatic conditions, accurate results cannot be predicted by this system.

✔ System only able to detect the disease from citrus leaves.

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 vide processing are possible with help OpenCV python library This project work can be extended for security applications such as figure print recognition, inis recognition and face recognition.

13.APPENDIX

13.1 SOURCE CODE

Home.html

```
<!DOCTYPE html>
<html >

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title> Plant Disease Prediction</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
<link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Josefin Sans' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>
<style>
.header {
                                        top:0;
                                        margin:0px;
                                        left: 0px;
                                        right: 0px;
                                        position: fixed;
                                        background-color: #28272c;
                                        color: white;
                                        box-shadow: 0px 8px 4px grey;
```

```
                              overflow: hidden;
                              padding-left:20px;
                              font-family: 'Josefin Sans';
                              font-size: 2vw;
                              width: 100%;
                              height:8%;
                              text-align: center;
                      }
                      .topnav {
  overflow: hidden;
  background-color: #333;
}

.topnav-right a {
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 18px;
}

.topnav-right a:hover {
  background-color: #ddd;
  color: black;
}

.topnav-right a.active {
  background-color: #565961;
  color: white;
}

.topnav-right {
  float: right;
  padding-right:100px;
}

body {

  background-color:#ffffff;
  background-repeat: no-repeat;
  background-size:cover;
  background-position: 0px 0px;
  }
  .button {
  background-color: #28272c;
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  border-radius: 12px;
}
.button:hover {
  box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
}
form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}

input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  display: inline-block;
  margin-bottom:18px;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

button {
  background-color: #28272c;
  color: white;
  padding: 14px 20px;
  margin-bottom:8px;
        border: none;
        cursor: pointer;
        width: 15%;
        border-radius:4px;
      }

```

```
button:hover {
  opacity: 0.8;
}

.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}

.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
}

img.avatar {
  width: 30%;
  border-radius: 50%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 100%;
  }
}

```

```
.home{
   margin:80px;

  width: 84%;
  height: 500px;
  padding-top:10px;
  padding-left: 30px;
}
.login{
   margin:80px;
   box-sizing: content-box;
  width: 84%;
  height: 420px;
  padding: 30px;
  border: 10px solid blue;
}
.left,.right{
 box-sizing: content-box;
 height: 400px;
 margin:20px;
 border: 10px solid blue;
}

.mySlides {display: none;}
img {vertical-align: middle;}

/* Slideshow container */
.slideshow-container {
  max-width: 1000px;
  position: relative;
  margin: auto;
}

/* Caption text */
.text {
  color: #f2f2f2;
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom: 8px;
  width: 100%;
  text-align: center;
}
/* The dots/bullets/indicators */
.dot {
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
}

.active {
  background-color: #717171;
}

/* Fading animation */
.fade {
  -webkit-animation-name: fade;
  -webkit-animation-duration: 1.5s;
  animation-name: fade;
  animation-duration: 1.5s;
}

@-webkit-keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
  .text {font-size: 11px}
}
</style>
</head>
```

```
        <body style="font-family:'Times New Roman', Times, serif;background-color:#C2C5A8;">

        <div class="header">
         <div style="width:50%;float:left;font-size:2vw;text-align:left;color:white; padding-top:1%">Plant Disease Prediction</div>
          <div class="topnav-right"style="padding-top:0.5%;">
            <a class="active" href="{{ url_for('home')}}">Home</a>
            <a href="{{ url_for('prediction')}}">Predict</a>
          </div>
        </div>

        <div style="background-image:url("./static/images/images.jpg");">
        <div style="width:60%;float:left;">
        <div style="font-size:50px;font-family:Montserrat;padding-left:20px;text-align:center;padding-top:10%;">
        <b>Detect if your plant<br> is infected!!</b></div><br>
        <div style="font-size:20px;font-family:Montserrat;padding-left:70px;padding-right:30px;text-align:justify;">Agriculture is one of the major sectors worls wide. Over the years it has developed and the use of new technologies and equipment replaced almost all the traditional methods of farming. The plant diseases effect the production. Identification of diseases and taking necessary precautions is all done through naked eye, which requires labour and laboratries. This application helps farmers in detecting the diseases by observing the spots on the leaves, which inturn saves effort and labor costs.</div><br><br>
        </div>
        </div>
        <div style="width:40%;float:right;"><br><br>
        <img src="{{url_for('static',filename='images/12456.png')}}" style="max-height:100%;max-width:100%;">

        </div>
```

```
</div>

<div class="home">

<br>

</div>

<script>
var slideIndex = 0;
showSlides();

function showSlides() {
  var i;
  var slides = document.getElementsByClassName("mySlides");
  var dots = document.getElementsByClassName("dot");
  for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }
  slideIndex++;
  if (slideIndex > slides.length) {slideIndex = 1}
  for (i = 0; i < dots.length; i++) {
    dots[i].className = dots[i].className.replace(" active", "");
  }
  slides[slideIndex-1].style.display = "block";
  dots[slideIndex-1].className += " active";
  setTimeout(showSlides, 2000); // Change image every 2 seconds
}
</script>
</body>
</html>
```

predict.html

```
<!DOCTYPE html>
<html >

<head>
```

```
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title> Plant Disease Prediction</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<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='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Josefin Sans' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>
<link href="{{ url_for('static', filename='css/final.css') }}" rel="stylesheet">
<style>
.header {
            top:0;
            margin:0px;
            left: 0px;
```

```
                    right: 0px;
                    position: fixed;
                    background-color: #28272c;
                    color: white;
                    box-shadow: 0px 8px 4px grey;
                    overflow: hidden;
                    padding-left:20px;
                    font-family: 'Josefin Sans';
                    font-size: 2vw;
                    width: 100%;
                    height:8%;
                    text-align: center;
            }
            .topnav {
  overflow: hidden;
  background-color: #333;
}

.topnav-right a {
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 18px;
}

.topnav-right a:hover {
  background-color: #ddd;
  color: black;
}

.topnav-right a.active {
  background-color: #565961;
  color: white;
}

.topnav-right {
  float: right;
  padding-right:100px;
}
```

```

.login{
margin-top:-70px;
}
body {

  background-color:#ffffff;
  background-repeat: no-repeat;
  background-size:cover;
  background-position: 0px 0px;
  }
.login{
  margin-top:100px;
}

.container {
  margin-top:40px;
  padding: 16px;
}
select {
  width: 100%;
  margin-bottom: 10px;
  background: rgba(255,255,255,255);
  border: none;
  outline: none;
  padding: 10px;
  font-size: 13px;
  color: #000000;
  text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
  border: 1px solid rgba(0,0,0,0.3);
  border-radius: 4px;
  box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);
  -webkit-transition: box-shadow .5s ease;
  -moz-transition: box-shadow .5s ease;
      -o-transition: box-shadow .5s ease;
      -ms-transition: box-shadow .5s ease;
      transition: box-shadow .5s ease;
}

</style>
```

```
</head>

<body style="font-family:Montserrat;overflow:scroll;">

<div class="header">
 <div style="width:50%;float:left;font-size:2vw;text-align:left;color:white; padding-top:1%">Plant Disease Prediction</div>
  <div class="topnav-right" style="padding-top:0.5%;">
  </div>
</div>
<div class="container">
        <div id="content" style="margin-top:2em">
          <div class="container">
           <div class="row">
                <div class="col-sm-6 bd" >

                 <br>
                    <img src="{{url_for('static',filename='images/images.jpg')}}" style="height:450px;width:550px"class="img-rounded" alt="Gesture">
                </div>
                <div class="col-sm-6">
                    <div>
                        <h4>Drop in the image to get the prediction </h4>
                <form action = "" id="upload-file" method="post" enctype="multipart/form-data">
                    <select name="plant">

                        <option value="select" selected>Select plant type</option>
                        <option value="fruit">Fruit</option>
                        <option value="vegetable">Vegetable</option>
            </select><br>
                    <label for="imageUpload" class="upload-label" style="background: #28272c;">
                        Choose...
                    </label>
                    <input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg">
```

```html
            </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" style="background: #28272c;">Predict!</button>
                </div>
            </div>

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

            <h3>
                <span id="result" style="font-size:17px; "> </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>
```

final.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: #28272c;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

.upload-label:hover{
    background: #C2C5A8;
    color: #39D2B4;
}

.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #28272c; /* 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('Prediction: '+data);
                console.log('Success!');
            },
        });
    });

});
```

app.py

```
import requests
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
import numpy as np
import pandas as pd
import tensorflow as tf
from flask import Flask, request, render_template, redirect, url_for
import os
from werkzeug.utils import secure_filename
from tensorflow.python.keras.backend import set_session

app = Flask(__name__)
global sess

global graph
graph=tf.compat.v1.get_default_graph()



```

```python
model = load_model("./Models/fruit.h5")
model1=load_model("./Models/vegetable.h5")


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


@app.route('/prediction')
def prediction():
    return render_template('predict.html')

@app.route('/predict',methods=['POST'])

def predict():
    if request.method == 'POST':

        f = request.files['image']


        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)
        img = image.load_img(file_path, target_size=(128, 128))
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)

        plant=request.form['plant']
        print(plant)

        if(plant=="vegetable"):
            preds = model.predict(x)
            preds = np.argmax(preds)
            print(preds)
            df=pd.read_excel('precautions - veg.xlsx')
            print(df.iloc[preds]['caution'])
        else:
            preds = model1.predict(x)
            preds = np.argmax(preds)
            df=pd.read_excel('precautions - fruits.xlsx')
            print(df.iloc[preds]['caution'])
        return df.iloc[preds]['caution']

if __name__ == "__main__":
    app.run(debug=False)
```

Github Link : https://github.com/IBM-EPBL/IBM-Project-12792-1659493073

Demo video link:https://drive.google.com/file/d/13b2s7LvJzIA4F36V6--OecD-1knpq9qF/view?usp=share_link