Fertilizers Recommendation System for Disease Prediction

IBM

PROJECT REPORT

Submitted by

TEAM ID: PNT2022TMID42619

SUCITHA AS - 711119104067

DIVYA R - 711119104703

JENISH MANOVA D - 711119104713

REJIN R - 711119104714

In partial fulfilment for the award of the degree

Of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

JANSONS INSTITUTE OF TECHNOLOGY,

KARUMATHAMPATTI

Table Of Contents

1.INTRODUCTION

1.1 Project Overview

The problem of efficient plant disease protection is closely related to the problems of sustainable agriculture Inexperienced pesticide usage can cause the development of long-term resistance of the pathogens, severely reducing the ability to fight back. Timely and accurate diagnosis of plant diseases is one of the pillars of precision agriculture. It is crucial to prevent unnecessary waste of financial and other resources, thus achieving healthier production in this changing environment, appropriate and timely disease identification including early prevention has never been more important. There are several ways to detect plant pathologies. Some diseases do not have any visible symptoms, or the effect becomes noticeable too late to act, and in those situations, a sophisticated analysis is obligatory. However, most diseases generate some kind of manifestation in the visible spectrum, so the naked eye examination of a trained professional is the prime technique adopted in practice for plant disease detection. In order to achieve accurate plant disease diagnostics a plant pathologist should possess good observation skills so that one can identify characteristic symptoms.

1.2 Purpose

Variations in symptoms indicated by diseased plants may lead to an improper diagnosis since amateur gardeners and hobbyists could have more difficulties determining it than a professional plant pathologist. An automated system designed to help identify plant diseases by the plant's appearance and visual symptoms could be of great help to amateurs in the gardening process and also trained professionals as a verification system in disease diagnostics. Advances in computer vision present an opportunity to expand and enhance the practice of precise plant protection and extend the market of computer vision applications in the field of precision agriculture.

2.LITERATURE SURVEY

2.1 Existing problem

It should proposed a method for leaf disease detection and suggest fertilizers to cure leaf diseases. But the method involves less number of train and test sets which results in poor accuracy. It proposed a simple prediction method for soil-based fertilizer recommendation system for predicted crop diseases. This method gives less accuracy and prediction. IoT based system for leaf disease detection and fertilizer recommendation which is based on Machine Learning techniques yields less 80 percentage accuracies.

2.2 References

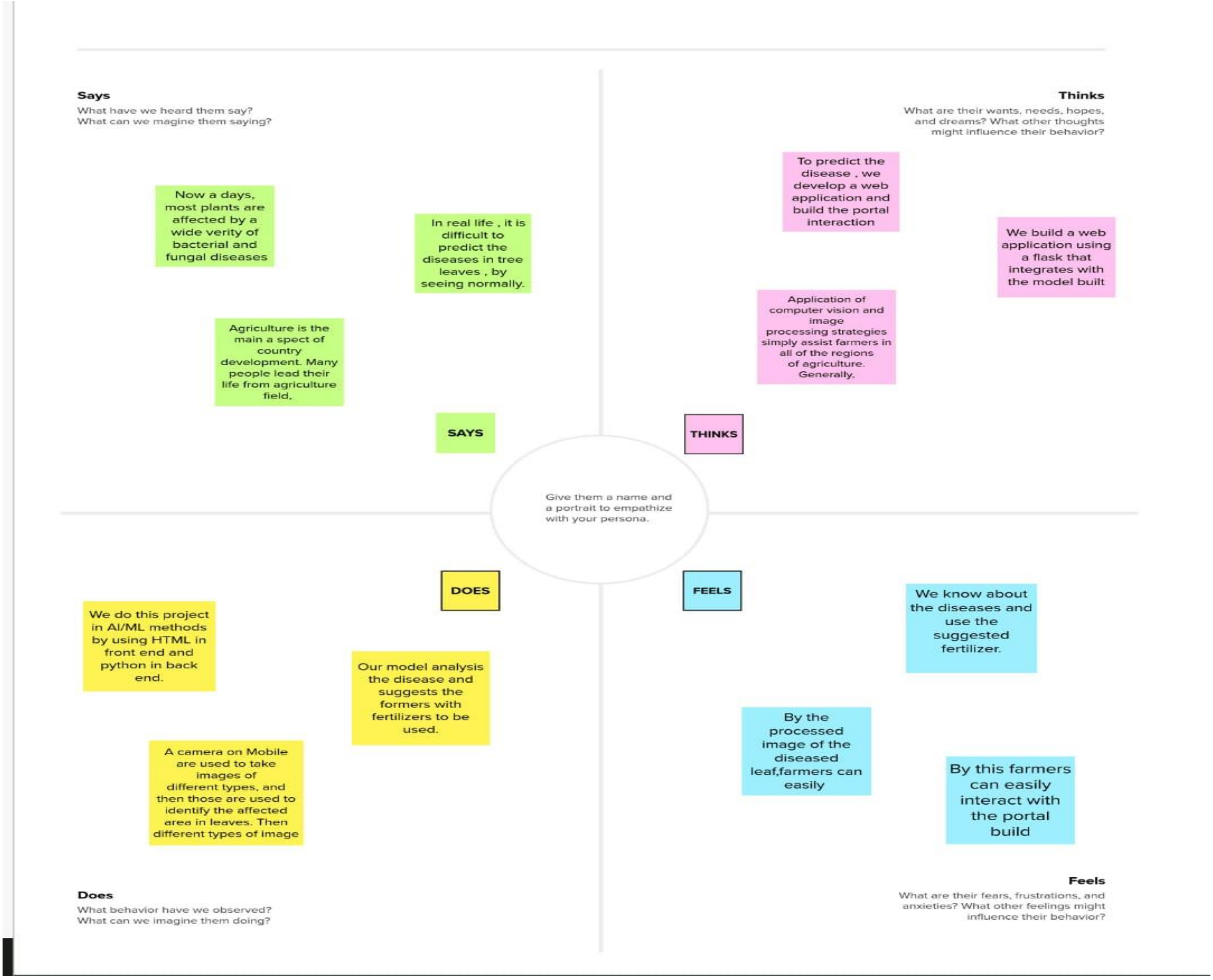
Neural Network Based Fertilizers Recommendation System For Disorder Classification And Prediction In Petal Images. This methodology requires experts who can recognize varieties in leaf shading. Ordinarily a similar malady is characterized by a few specialists as a different sickness. This arrangement is exorbitant, in light of the fact that it requires nonstop expert management

2.3 Problem Statement Definition

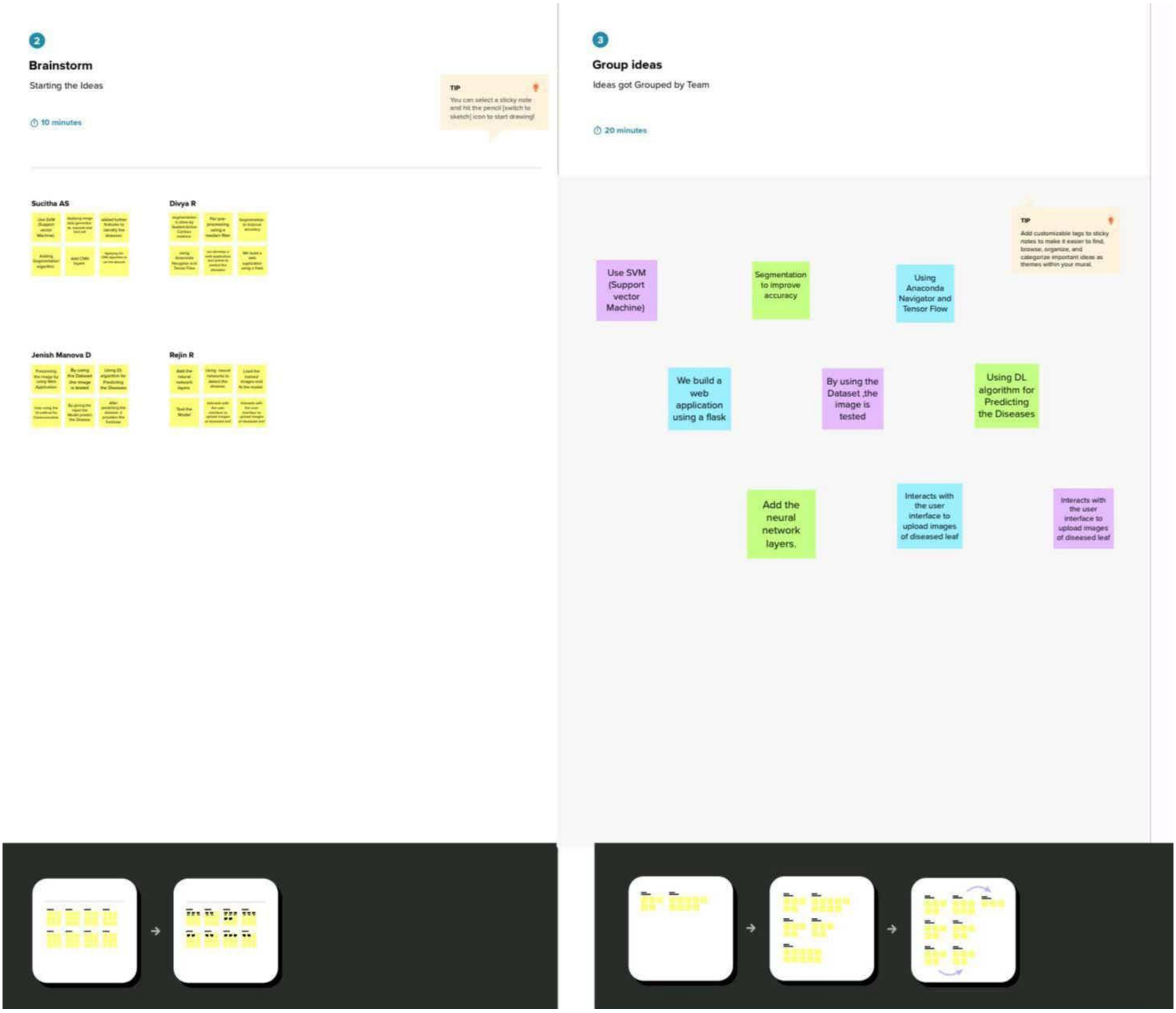
Agriculture is the most important sector in today's life. Most of the plants are affected by a wide variety of bacterial and fungal diseases. In agricultural aspects, if the plant is affected by leaf diseases then it reduces the growth and productiveness. Generally, the plant diseases are caused by the abnormal physiological functionalities of plants.

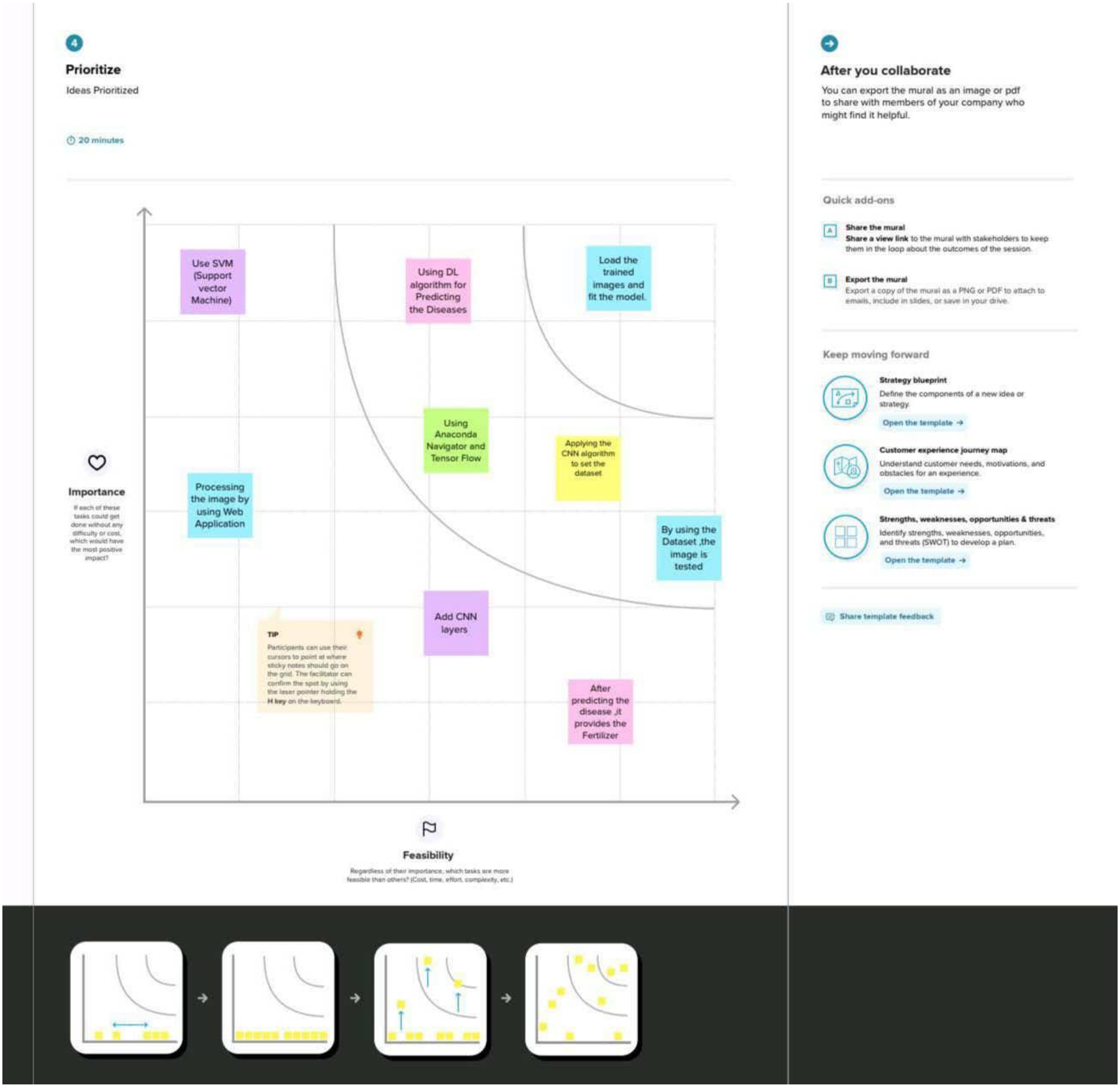
3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming


2
Brainstorm
Starting the Ideas
10 minutes
TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!
Sucitha AS
Divya R
Jenish Manova D
Rejin R
3
Group ideas
Ideas got Grouped by Team
20 minutes
TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.
Use SVM (Support vector Machine)
Segmentation to improve accuracy
Using Anaconda Navigator and Tensor Flow
We build a web application using a flask
By using the Dataset ,the image is tested
Using DL algorithm for Predicting the Diseases
Add the neural network layers.
Interacts with the user interface to upload images of diseased leaf
Interacts with the user interface to upload images of diseased leaf



4
Prioritize
Ideas Prioritized
20 minutes
Use SVM (Support vector Machine)
Using DL algorithm for Predicting the Diseases
Load the trained images and fit the model.
Using Anaconda Navigator and Tensor Flow
Applying the CNN algorithm to set the dataset
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Processing the image by using Web Application
By using the Dataset ,the image is tested
Add CNN layers
TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.
After predicting the disease ,it provides the Fertilizer
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)
After you collaborate
You can export the mural as an image or pdf to share with members of your company who might find it helpful.
Quick add-ons
Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.
Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.
Keep moving forward
Strategy blueprint
Define the components of a new idea or strategy.
Open the template →
Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →
Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →
Share template feedback

3.3 Proposed Solution

S.No.	Parameter	Description
1	Problem Statement (Problem to be solved)	Most of the Indian population depends on agriculture for their livelihood
2	Idea / Solution description	To overcome all these issues this recommendation has been proposed. Nowadays a lot of research and work is being implemented in the smart and modern agriculture domain.
3	Novelty / Uniqueness	Predicting the fertilizers, analyzing the disease in a tap makes the life of farmers easy with minimal subscription would provide an acceptable return for the organization
4	Social Impact / Customer Satisfaction	Models to recommend the right crop based on soil value and the best fertilizer to use.
5	Business Model (Revenue Model)	This action adds a lot of value to the company and the business in society. Many new technologies, such as Machine Learning and Deep Learning, are being implemented.
6	Scalability of the Solution	The ensembles technique is used to build a recommendation model. Models to recommend the right crop based on soil value and the best fertilizer to use

3.4 Problem Solution fit

Project Title: Fertilizer recommendation system | **Project Design Phase-I - Solution Fit Template** | **Team ID:** PNT2022TMID42619

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) C

Farmers are the customers here. They can easily use this application to identify the disease of the crops and the usage of fertilizer

6. CUSTOMER CONSTRAINTS CC

The image should be in the required pixels and clarity in order to get the accurate results. Good network connection should be available.

5. AVAILABLE SOLUTIONS A

Many of the people easily determine or judge the disease of the plants by seeing it. This may lead to the decay of the plants and crops which may occur in the past. The pros and cons are sometimes the crops may adapt the fertilizer and sometimes it is not.

Explore AS, differentiate

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

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

The application mainly focused on helping the farmers. The farmers who need better identification and suggestion on the disease of the plants and the fertilizers

9. PROBLEM ROOT CAUSE RC

There are so many diseases introducing day by day in our life. Those diseases mainly affect the crop's condition and the quality of the crops get reduced and eventually the quantity. The weather condition and the pests can cause various disease.

7. BEHAVIOUR BE

Directly related:

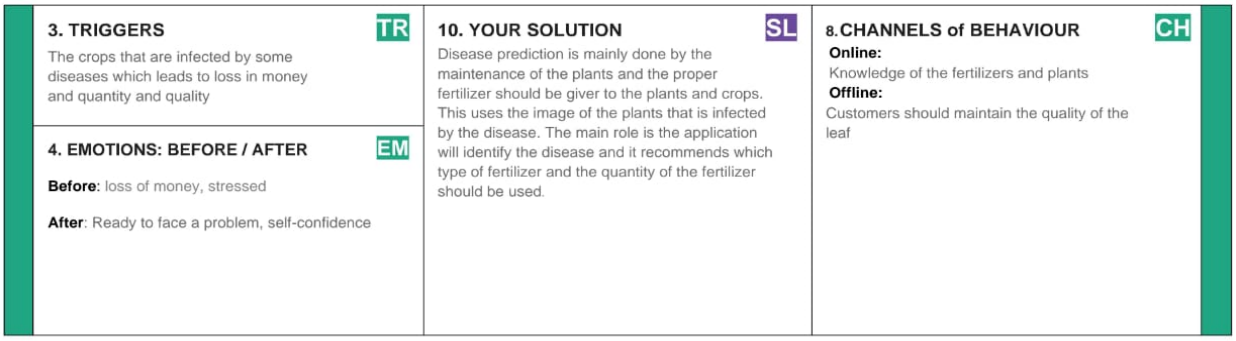
Farmers can identify the disease and they didn't want the knowledge of the plants for the prediction of the diseases.

Indirectly:

Farmers easily able to get the results through online fast and secure.

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

Agriculture is one field which has a high impact on life and economic status of human beings. Improper management leads to loss in agricultural products. Farmers lack the knowledge of disease and hence they produce less production. Farmers are unable to explain disease properly on call need to analysis the image of affected area of disease. Though, images and videos of crops provide better view and agro scientists can provide a better solution to resolve the issues related to healthy crop yet it not been informed to farmers. It is required to note that if the productivity of the crop is not healthy, it has high risk of providing good and healthy nutrition. Due to the improvement and development in technology where devices are smart enough to recognize and detect plant diseases. Recognizing illness can prompt faster treatment in order to lessen the negative impacts on harvest. This paper therefore focus upon plant disease detection using image processing approach. This work utilizes an open dataset of pictures of unhealthy and solid plants, where convolution system and semi supervised techniques are used to characterize crop species and detect the sickness. Farmers can interact with portal build interact with user interface to upload images of diseases leaf the image will be processing and train data to the algorithm. These type of algorithm may evaluate the processing of image and predict to the user interface.



4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

SI NO	Functional Requirement (Epic)	Sub Requirement (Story / SubTask)
FR-1	User Registration	Registration through web portal Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Capturing image	Capture the image of the leaf and check the parameter of the captured image
FR-4	Image processing	Upload the image for the prediction of the disease in the leaf and process the image using the trained data
FR-5	Leaf identification	Identify the leaf and predict the disease in leaf.
FR-6	Image description	Suggesting the best fertilizer for the disease.

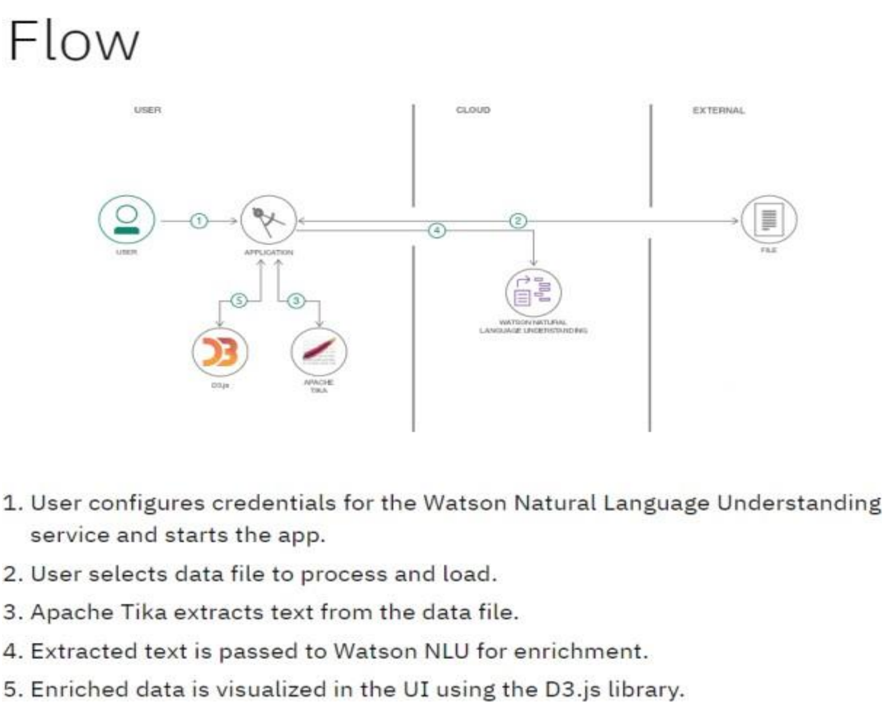
4.2 NON-FUNCTIONAL REQUIREMENT

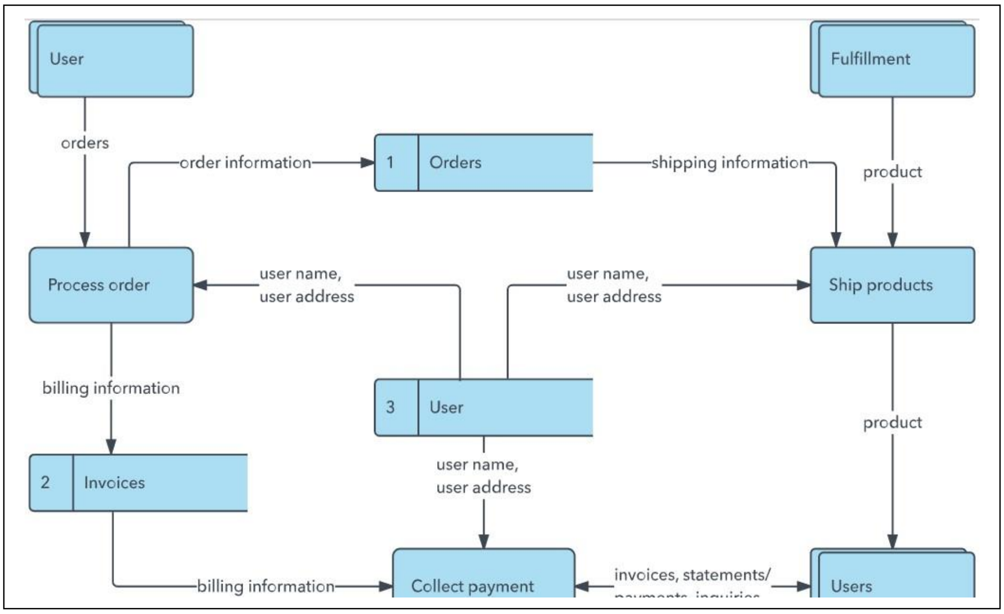
NFR NO	Non-Functional Requirement	Description
NFR-1	Usability	Datasets of all the leaf is used to detecting the disease that present in the leaf.
NFR-2	Security	The information belongs to the user and leaf are secured highly
NFR-3	Reliability	The leaf quality is important for the predicting the disease in leaf.
NFR-4	Performance	The performance is based on the quality of the leaf used for disease prediction.
NFR-5	Availability	It is available for all user to predict the disease in the plant.
NFR-6	Scalability	Increasing the prediction of the disease in the leaf

5. PROJECT DESIGN

5.1 Data Flow Diagrams

Example: DFD Level 0 (Industry Standard)

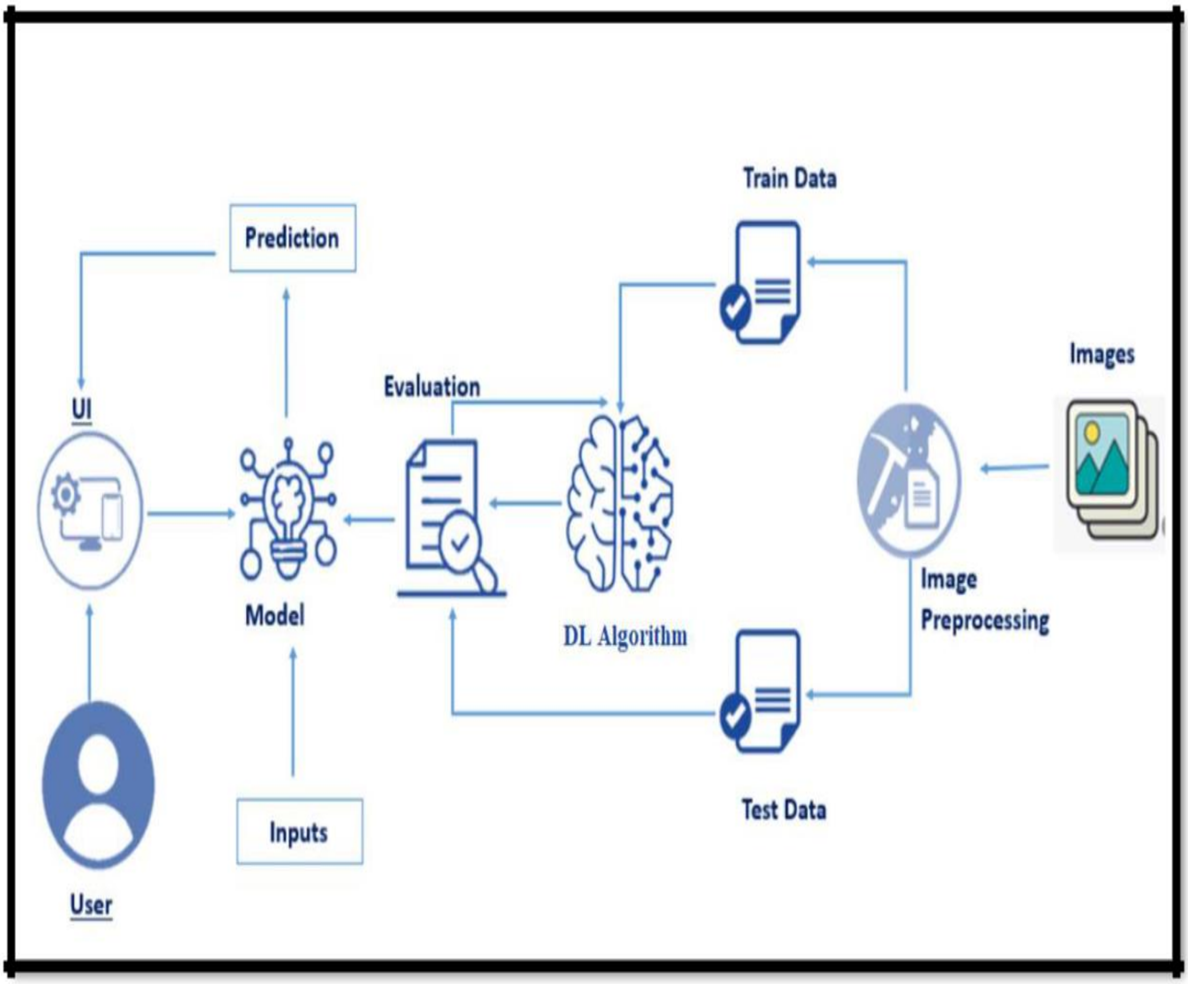




From this technical architecture and solutions the user can take the picture of an image and the image will be processing the data the train data and test data are should be using in AI algorithm. The evaluated image could predict the disease and gives an solution for the attacked leaf.

5.2 Solution and Technical Architecture

The proposed method uses 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.



5.3 User Stories

The user can register in the account and may use mail id and password to sss the page and make useful for farmers. The image can be processed and data should be held by it. If you predict the images can be taken an photo and predict button should be made. Then the prediction will appear. If the leaf is healthy it shows healthy. If the leaf is not healthy it shoes the prediction for fertilizer.

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

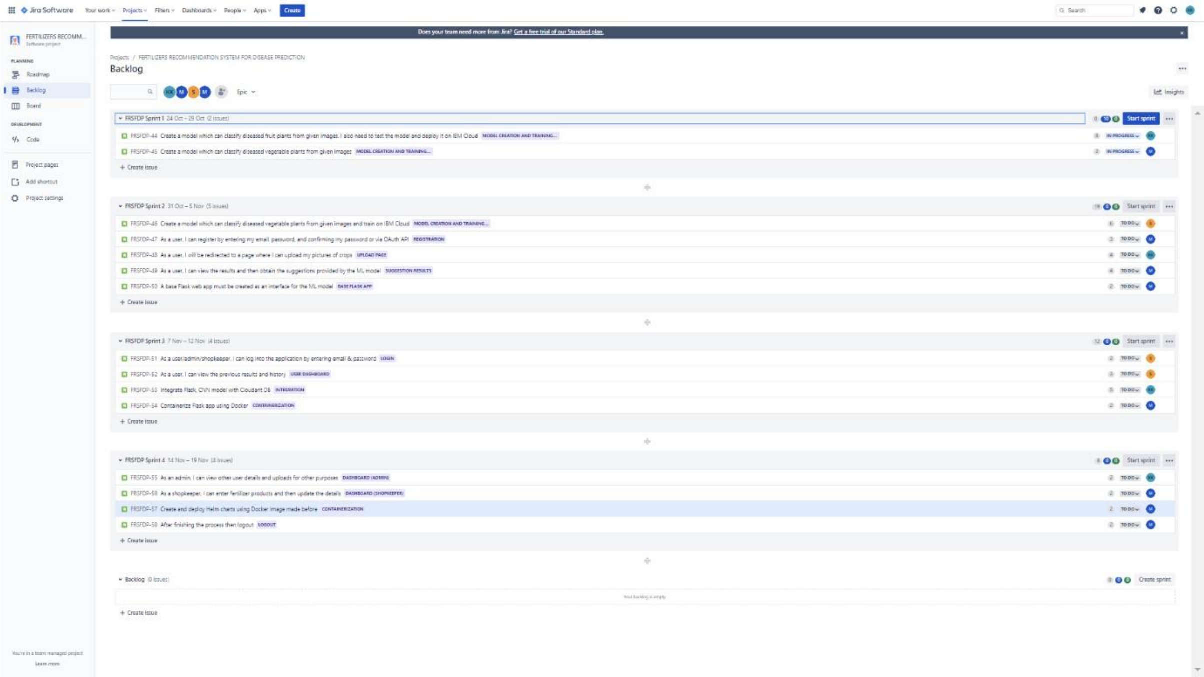
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points (Total)	Priority	Team Members
Sprint1	Model Creation and Training (Fruits)	USN-1	Create a model which can classify diseased fruit plants from given images. I also need to test the model and deploy it on IBM Cloud	8	High	Sucitha
	Model Creation and Training (Vegetables)	USN-2	Create a model which can classify diseased vegetable plants from given images	2	Medium	Jenish Manova
Sprint2	Model Creation and Training (Vegetables)	USN-3	Create a model which can classify diseased vegetable plants from given images and train on IBM Cloud	6	High	Rejin
	Registration	USN-4	As a user, I can register by entering my email, password, and confirming my password or via OAuth API	3	High	Divya
	Upload page	USN-5	As a user, I will be redirected to a page where I can upload my pictures of crops	4	High	Jenish Manova
	Suggestion results	USN-6	As a user, I can view the results and then obtain the suggestions provided by the ML model	4	High	Sucitha
	Base Flask App	USN-7	A base Flask web app must be created as an interface for the ML model	2	High	Rejin
Sprint3	Login	USN-8	As a user/admin/shopkeeper, I can log into the application by entering email & password	2	High	Divya
	User Dashboard	USN-9	As a user, I can view the previous results and history	3	Medium	Sucitha
	Integration	USN-10	Integrate Flask, CNN model with Cloudant DB	5	Medium	Jenish Manova

	Containerization	USN-11	Containerize Flask app using Docker	2	Low	Rejin
Sprint4	Dashboard (Admin)	USN-12	As an admin, I can view other user details and uploads for other purposes	2	Medium	Sucitha
	Dashboard (Shopkeeper)	USN-13	As a shopkeeper, I can enter fertilizer products and then update the details	2	Low	Rejin
	Containerization	USN-14	Create and deploy Helm charts using Docker Image made before	2	Low	Divya
	Logout	USN-15	After finishing the process then logout	2	Low	Jenish Manova

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	29 Oct 2022	03 Nov 2022	10	03 Nov 2022
Sprint-2	15	6 Days	04 Nov 2022	11 Nov 2022	15	10 Nov 2022
Sprint-3	15	6 Days	14 2022	20 Nov 2022	15	19 Nov 2022
Sprint-4	12	6 Days	19 Nov 2022	22 Nov 2022	10	22 Nov 2022

6.3 Reports from JIRA



7.CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

```
<!DOCTYPE
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>A flying astronaut</title>
                          <style>img[alt="www.000webhost.com"]{display: none}

                    *{
                       margin: 0;
                       padding: 0;
                     }
                     body{
```

```
  background-color: #102037;
  overflow: hidden;
}
@-webkit-keyframes snow {
  0% { opacity: 0; transform: translateY(0px); }
  20%{ opacity: 1;}
  100% { opacity: 1; transform: translateY(650px); }
}
@-moz-keyframes snow {
  0% { opacity: 0; transform: translateY(0px); }
  20%{ opacity: 1;}
  100% { opacity: 1; transform: translateY(650px); }
}
@keyframes snow {
  0% { opacity: 0; transform: translateY(0px); }
  20%{ opacity: 1;}
  100% { opacity: 1; transform: translateY(650px); }
}
@-webkit-keyframes astronaut{
  0%{
    transform: rotate(0deg);
  }
  100%{
    transform: rotate(360deg);
  }
}
.box-of-star1,
.box-of-star2,
.box-of-star3,
.box-of-star4{
  width: 100%;
  position: absolute;
  z-index: 10;
  left: 0;
  transform: translateY(650px);
}
.box-of-star1{
  -webkit-animation: snow 5s linear infinite;
}
.box-of-star2{
```

```
  -webkit-animation: snow 5s -1.64s linear infinite;
}
.box-of-star3{
  -webkit-animation: snow 5s -2.30s linear infinite;
}
.box-of-star4{
  -webkit-animation: snow 5s -3.30s linear infinite;
}
.star{
  width: 3px;
  height: 3px;
  border-radius: 50%;
  background-color: #FFF;
  position: absolute;
  z-index: 10;
  opacity: .7;
}
.star:before{
  content: "";
  width: 6px;
  height: 6px;
  border-radius: 50%;
  background-color: #FFF;
  position: absolute;
  z-index: 10;
  top: 40px;
  left: 70px;
  opacity: .7;
}
.star:after{
  content: "";
  width: 8px;
  height: 8px;
  border-radius: 50%;
  background-color: #FFF;
  position: absolute;
  z-index: 10;
  top: 8px;
  left: 170px;
  opacity: .9;
```

```
}
.star-position1{
  top: 30px;
  left: 20px;
}
.star-position2{
  top: 110px;
  left: 250px;
}
.star-position3{
  top: 60px;
  left: 570px;
}
.star-position4{
  top: 120px;
  left: 900px;
}
.star-position5{
  top: 20px;
  left: 1120px;
}
.star-position6{
  top: 90px;
  left: 1280px;
}
.star-position7{
  top: 30px;
  left: 1480px;
}
.astronaut{
  width: 250px;
  height: 300px;
  position: absolute;
  z-index: 11;
  top: calc(50% - 150px);
  left: calc(50% - 125px);
  -webkit-animation: astronaut 5s linear infinite;
}
.schoolbag{
  width: 100px;
```

```
    height: 150px;
    position: absolute;
    z-index: 1;
    top: calc(50% - 75px);
    left: calc(50% - 50px);
    background-color: #94b7ca;
    border-radius: 50px 50px 0 0 / 30px 30px 0 0;
  }
  .head{
    width: 97px;
    height: 80px;
    position: absolute;
    z-index: 3;
    background: -webkit-linear-gradient(left, #e3e8eb 0%,
#e3e8eb 50%, #fbfdfa 50%, #fbfdfa 100%);
    border-radius: 50%;
    top: 34px;
    left: calc(50% - 47.5px);
  }
  .head:after{
    content: "";
    width: 60px;
    height: 50px;
    position: absolute;
    top: calc(50% - 25px);
    left: calc(50% - 30px);
    background: -webkit-linear-gradient(top, #15aece 0%,
#15aece 50%, #0391bf 50%, #0391bf 100%);
    border-radius: 15px;
  }
  .head:before{
    content: "";
    width: 12px;
    height: 25px;
    position: absolute;
    top: calc(50% - 12.5px);
    left: -4px;
    background-color: #618095;
    border-radius: 5px;
    box-shadow: 92px 0px 0px #618095;
```

```
  }
  .body{
    width: 85px;
    height: 100px;
    position: absolute;
    z-index: 2;
    background-color: #fffbff;
    border-radius: 40px / 20px;
    top: 105px;
    left: calc(50% - 41px);
    background: -webkit-linear-gradient(left, #e3e8eb 0%,
#e3e8eb 50%, #fbfdfa 50%, #fbfdfa 100%);
  }
  .panel{
    width: 60px;
    height: 40px;
    position: absolute;
    top: 20px;
    left: calc(50% - 30px);
    background-color: #b7cceb;
  }
  .panel:before{
    content: "";
    width: 30px;
    height: 5px;
    position: absolute;
    top: 9px;
    left: 7px;
    background-color: #fbfdfa;
    box-shadow: 0px 9px 0px #fbfdfa, 0px 18px 0px #fbfdfa;
  }
  .panel:after{
    content: "";
    width: 8px;
    height: 8px;
    position: absolute;
    top: 9px;
    right: 7px;
    background-color: #fbfdfa;
    border-radius: 50%;
```

```
  box-shadow: 0px 14px 0px 2px #fbfdfa;
}
.arm{
  width: 80px;
  height: 30px;
  position: absolute;
  top: 121px;
  z-index: 2;
}
.arm-left{
  left: 30px;
  background-color: #e3e8eb;
  border-radius: 0 0 0 39px;
}
.arm-right{
  right: 30px;
  background-color: #fbfdfa;
  border-radius: 0 0 39px 0;
}
.arm-left:before,
.arm-right:before{
  content: "";
  width: 30px;
  height: 70px;
  position: absolute;
  top: -40px;
}
.arm-left:before{
  border-radius: 50px 50px 0px 120px / 50px 50px 0 110px;
  left: 0;
  background-color: #e3e8eb;
}
.arm-right:before{
  border-radius: 50px 50px 120px 0 / 50px 50px 110px 0;
  right: 0;
  background-color: #fbfdfa;
}
.arm-left:after,
.arm-right:after{
  content: "";
```

```
  width: 30px;
  height: 10px;
  position: absolute;
  top: -24px;
}
.arm-left:after{
  background-color: #6e91a4;
  left: 0;
}
.arm-right:after{
  right: 0;
  background-color: #b6d2e0;
}
.leg{
    width: 30px;
    height: 40px;
    position: absolute;
    z-index: 2;
    bottom: 70px;
}
.leg-left{
    left: 76px;
    background-color: #e3e8eb;
    transform: rotate(20deg);
}
.leg-right{
    right: 73px;
    background-color: #fbfdfa;
    transform: rotate(-20deg)
}
.leg-left:before,
.leg-right:before{
    content: '';
    width: 50px;
    height: 25px;
    position: absolute;
    bottom: -26px;
}
.leg-left:before{
    left: -20px;
```

```
      background-color: #e3e8eb;
      border-radius: 30px 0 0 0 ;
  }
.button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
}
.button1 {width: 250px;}
  .leg-right:before{
      right: -20px;
      background-color: #e3e8eb;
      border-radius: 0 30px 0 0 ;
  }
</style>
</head>
<body style="color:white;">
<br>
<br>
<br>
    <div class="box-of-star1">
        <div class="star star-position1"></div>
        <div class="star star-position2"></div>
        <div class="star star-position3"></div>
        <div class="star star-position4"></div>
        <div class="star star-position5"></div>
        <div class="star star-position6"></div>
        <div class="star star-position7"></div>
      </div>
      <div class="box-of-star2">
        <div class="star star-position1"></div>
        <div class="star star-position2"></div>
        <div class="star star-position3"></div>
```

```html
    <div class="star star-position4"></div>
    <div class="star star-position5"></div>
    <div class="star star-position6"></div>
    <div class="star star-position7"></div>
  </div>
  <div class="box-of-star3">
    <div class="star star-position1"></div>
    <div class="star star-position2"></div>
    <div class="star star-position3"></div>
    <div class="star star-position4"></div>
    <div class="star star-position5"></div>
    <div class="star star-position6"></div>
    <div class="star star-position7"></div>
  </div>
  <div class="box-of-star4">
    <div class="star star-position1"></div>
    <div class="star star-position2"></div>
    <div class="star star-position3"></div>
    <div class="star star-position4"></div>
    <div class="star star-position5"></div>
    <div class="star star-position6"></div>
    <div class="star star-position7"></div>
  </div>

<center>

 <h1>Leaf Disease Detection</h1>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>

<h3>A leaf spot is a limited, discoloured, diseased area of a
leaf that is caused by fungal, bacterial or viral plant diseases,
or by injuries from nematodes, insects, environmental factors,
toxicity or herbicides.</h3>
```

```
        <button class="button button1"> <a href="/my-
link/">Press to Analyse</a></button>
</center>
    </body>

</html>
```

7.2 Feature 2

```
import tensorflow as tf
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dense, Dropout, Activation, Flatten,
Conv2D, MaxPooling2D
import pickle
from keras.models import model_from_json
from keras.models import load_model
import matplotlib.pyplot as plt
import numpy as np
# Opening the files about data
X = pickle.load(open("X.pickle", "rb"))
y = pickle.load(open("y.pickle", "rb"))

# normalizing data (a pixel goes from 0 to 255)
X = X/255.0

# Building the model
# Building the model
model = Sequential()
# 3 convolutional layers
model.add(Conv2D(32, (3, 3), input_shape = X.shape[1:]))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2,2)))
```

```
model.add(Dropout(0.25))

# 2 hidden layers
model.add(Flatten())
model.add(Dense(128))
model.add(Activation("relu"))



model.add(Dense(64))
model.add(Activation("relu"))

# The output layer with 13 neurons, for 13 classes
model.add(Dense(15))
model.add(Activation("softmax"))

# Compiling the model using some basic parameters
model.compile(loss="sparse_categorical_crossentropy",
                optimizer="adam",
                metrics=["accuracy"])
y=np.array(y)
# Training the model, with 40 iterations
# validation_split corresponds to the percentage of images used for the validation phase compared to all the images
history = model.fit(X, y, batch_size=10, epochs=15, validation_split=0.2)

# Saving the model
model_json = model.to_json()
with open("model.json", "w") as json_file :
    json_file.write(model_json)

model.save_weights("model.h5")
print("Saved model to disk")

model.save('CNN.model')

# Printing a graph showing the accuracy changes during the training phase
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
```

```
val_loss = history.history['val_loss']
acc=np.array(acc)
val_acc=np.array(val_acc)
loss=np.array(loss)
val_loss=np.array(val_loss)
epochs_range = range(15)

plt.figure(figsize=(15, 15))
plt.subplot(2, 2, 1)
plt.plot(epochs_range, acc, label='Training Accuracy')
plt.plot(epochs_range, val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(2, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```

8. TESTING

8.1 Test Cases

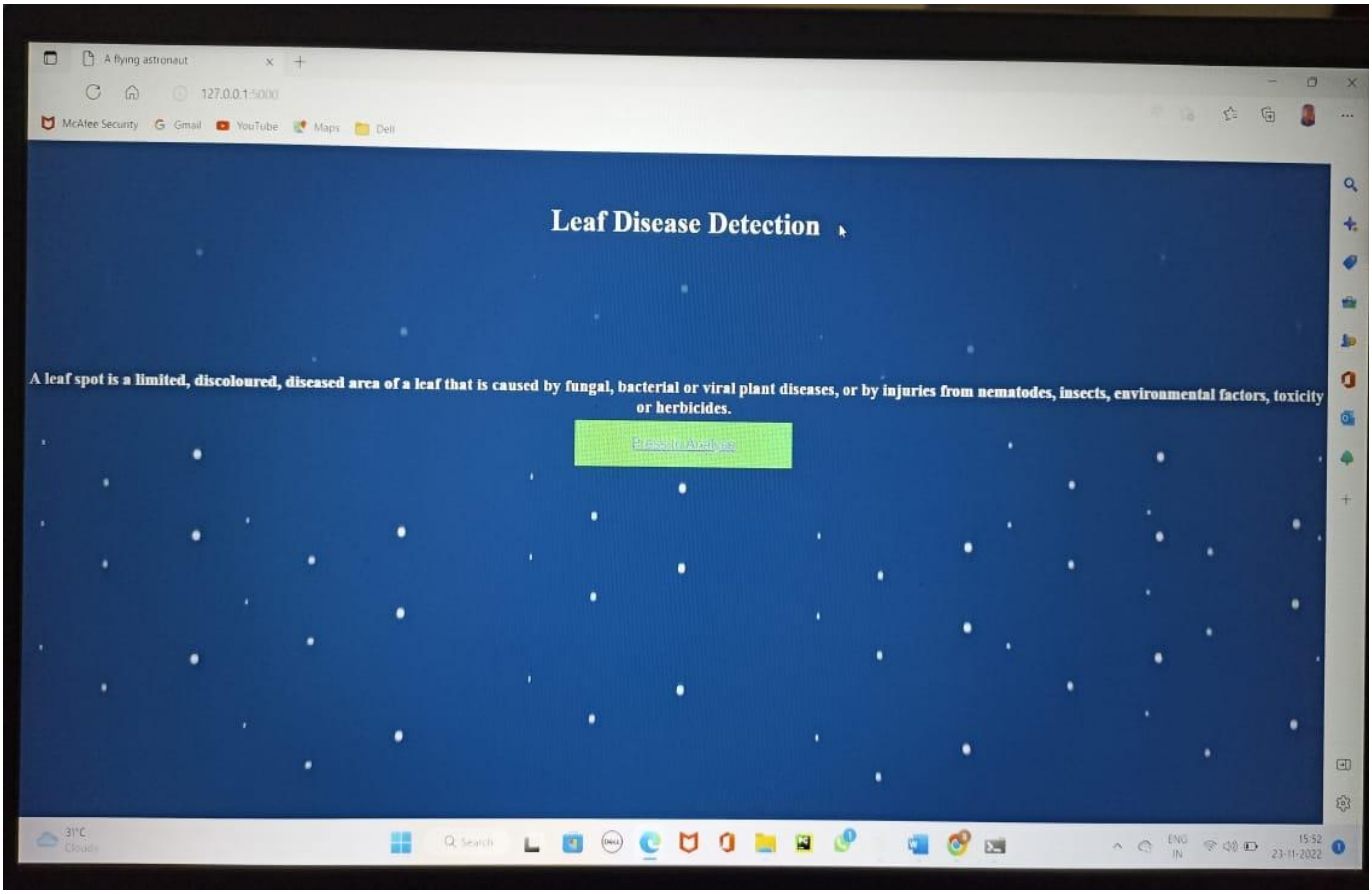
SECTION	**TOTAL CASES**	**NOT TESTED**	**FAIL**	**PASS**
Leaf spots	17	0	0	17
Mosaic Leaf Pattern	51	0	0	51
Misshapen Leaves	20	0	0	20
Yellow Leaves	7	0	0	7
Fruit Rots	9	0	0	9
Fruit Spots	4	0	0	4
Blights	2	0	0	2

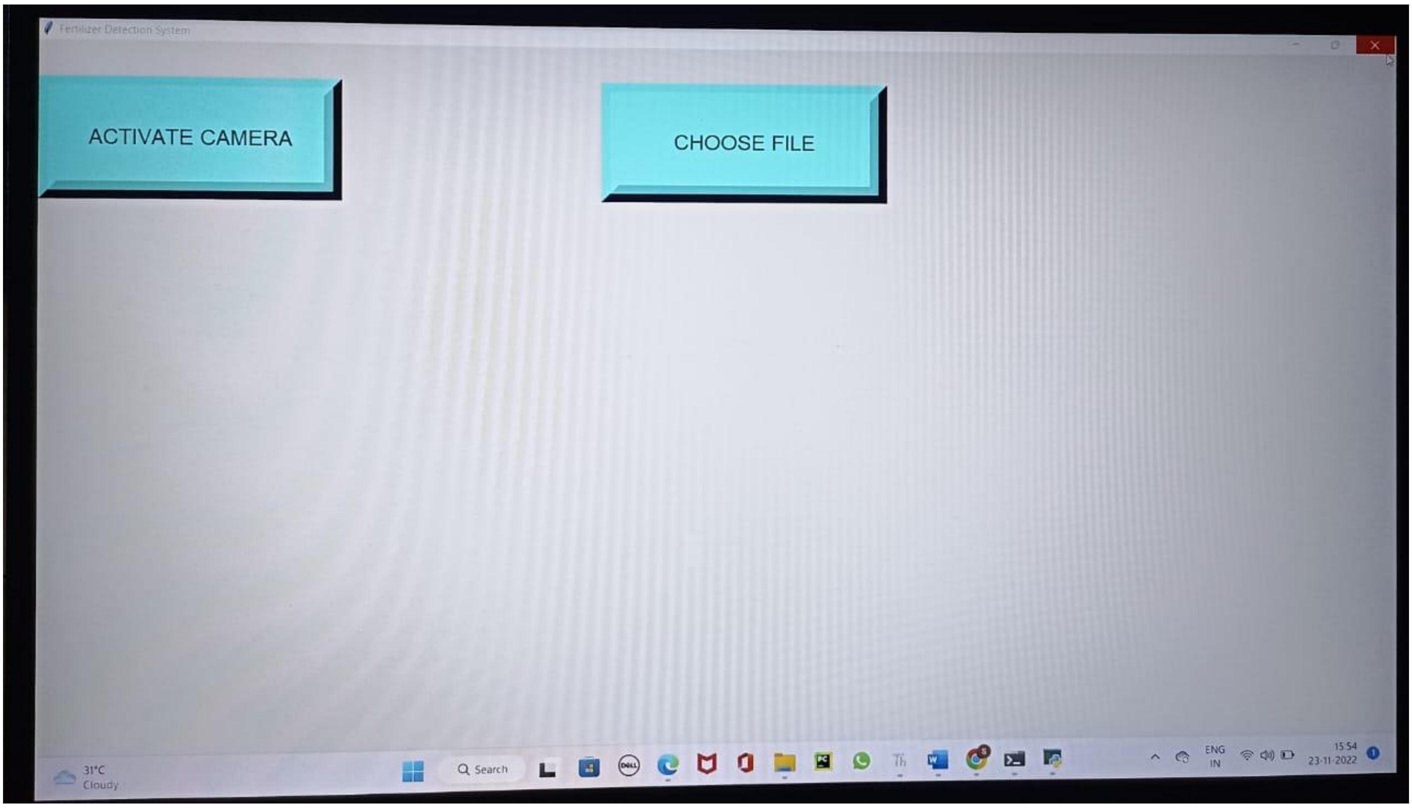
8.2 User Acceptance Testing

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

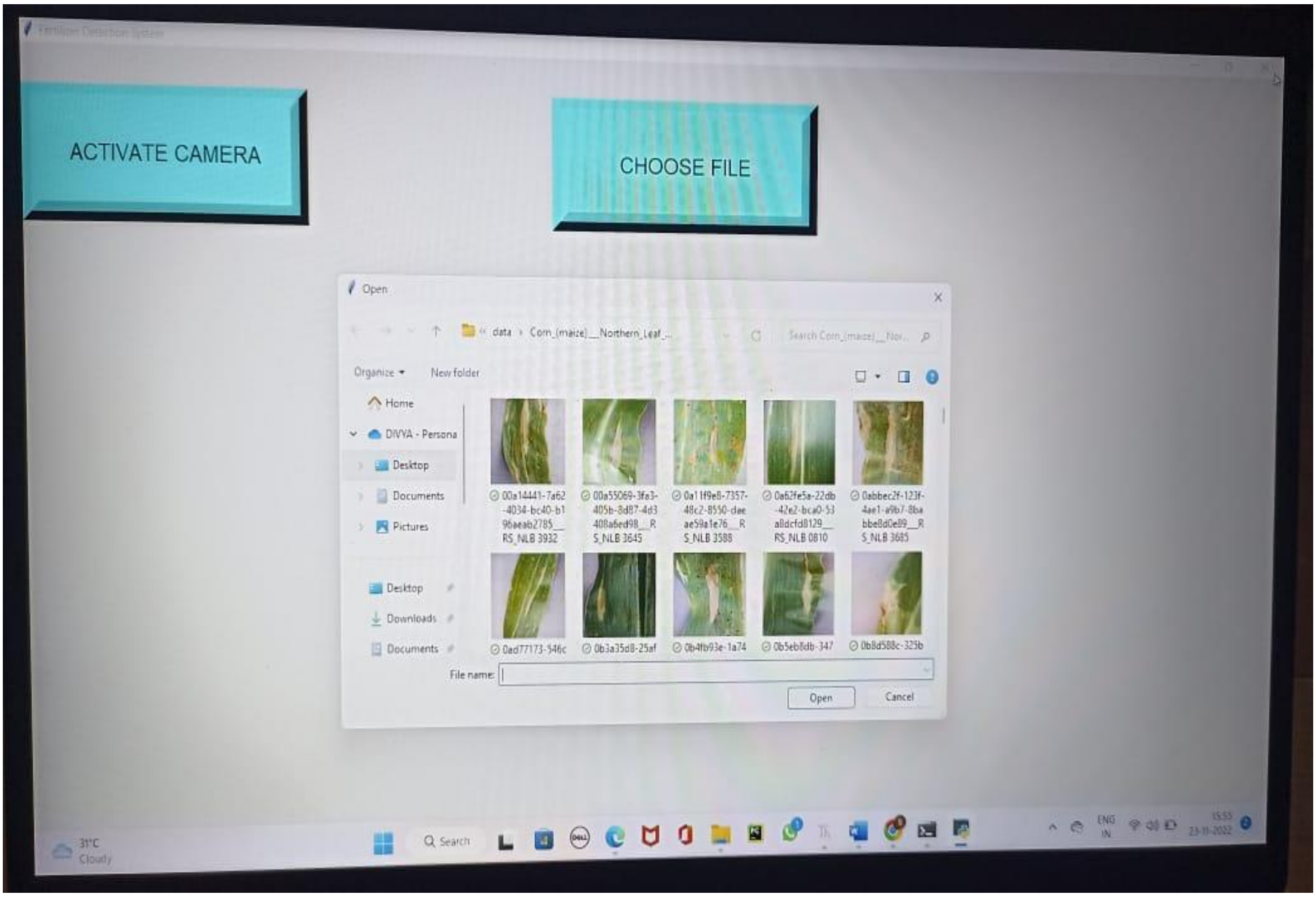
9. RESULTS

9.1 Performance Metrics

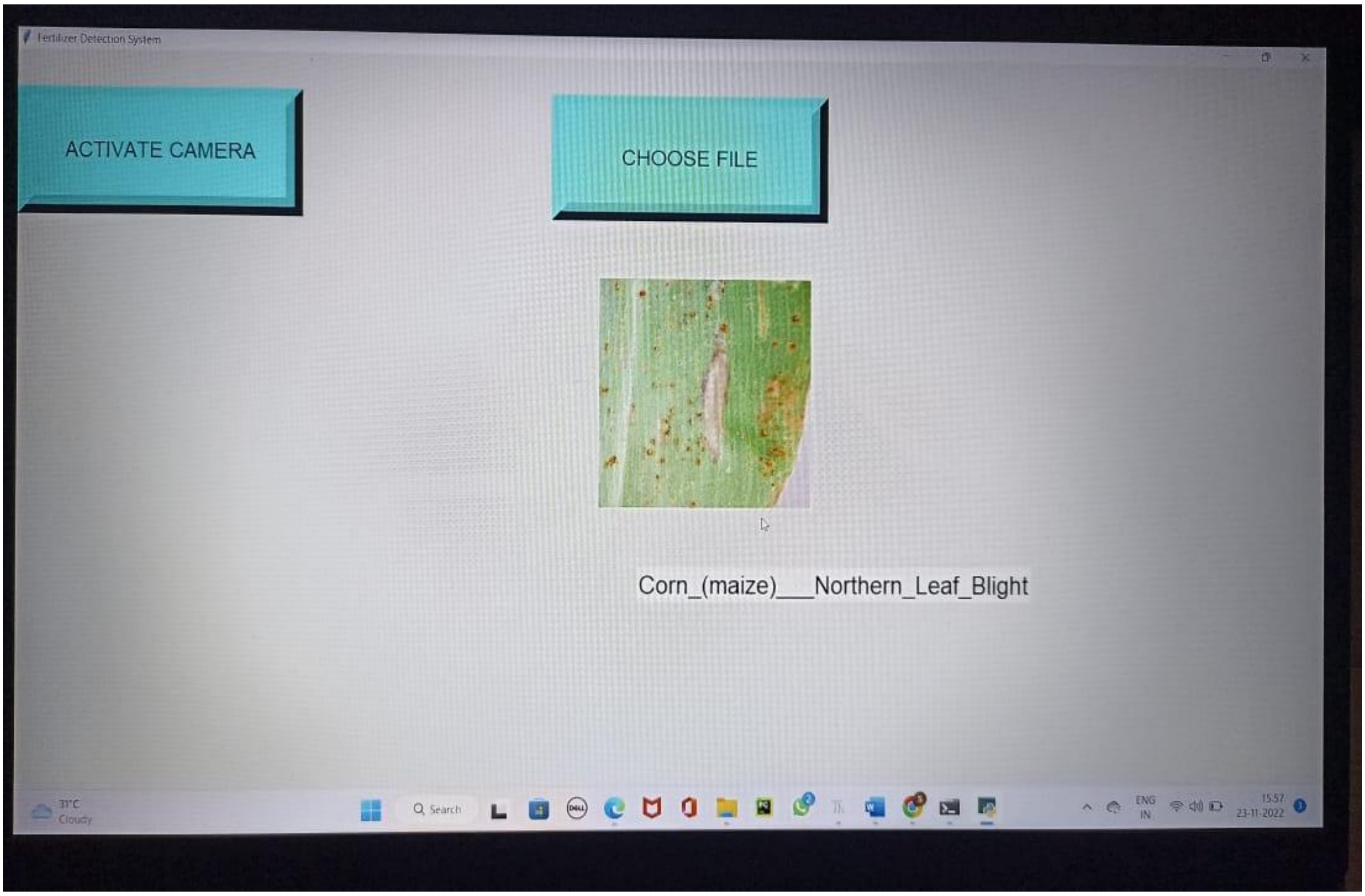


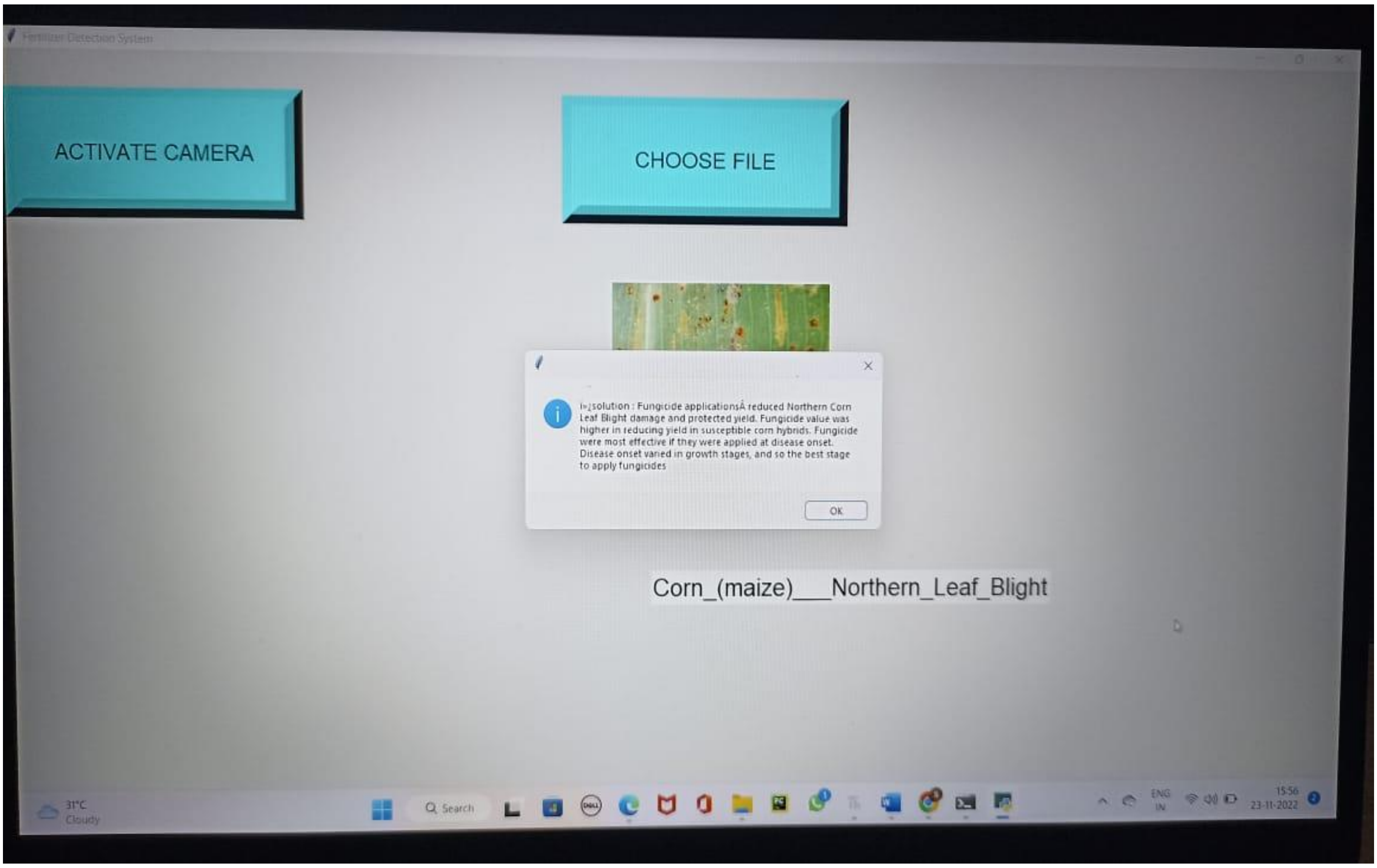


Fertilizer Detection System
ACTIVATE CAMERA
CHOOSE FILE



ACTIVATE CAMERA
CHOOSE FILE
Open
Organize
New folder
Home
Desktop
Documents
Pictures
Downloads
File name:
Open
Cancel


Fertilizer Detection System
ACTIVATE CAMERA
CHOOSE FILE
Corn_(maize)___Northern_Leaf_Blight


Fertilizer Detection System
ACTIVATE CAMERA
CHOOSE FILE
ï»¿solution : Fungicide applicationsÂ reduced Northern Corn Leaf Blight damage and protected yield. Fungicide value was higher in reducing yield in susceptible corn hybrids. Fungicide were most effective if they were applied at disease onset. Disease onset varied in growth stages, and so the best stage to apply fungicides
OK
Corn_(maize)___Northern_Leaf_Blight

10. ADVANTAGES & DISADVANTAGES

List of advantages

- The proposed model here produces very high accuracy of classification.
- Very large datasets can also be trained and tested.
- Images of very high can be resized within the proposed itself.

List of disadvantages

- For training and testing, the proposed model requires very high computational time.

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

11. CONCLUSION

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

- The accuracy of classification increased by increasing the number of epochs.
- For different batch sizes, different classification accuracies are obtained.
- The accuracies are increased by increasing more convolution layers.
- The accuracy of classification also increased by varying dense layers.
- Different accuracies are obtained by varying the size of kernel used in the convolution layer output.
- Accuracies are different while varying the size of the train and test datasets.

12. FUTURE SCOPE

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

13.APPENDIX

Source Code:

```
import numpy as np
import matplotlib.pyplot as plt
import cv2
import os
import tensorflow as tf
from keras.models import load_model
from tkinter import *
import tkinter.messagebox
import PIL.Image
import PIL.ImageTk
from tkinter import filedialog
from tkinter import filedialog
import csv
file = open('solution.csv')

type(file)
csvreader = csv.reader(file)
header = []
header = next(csvreader)

CATEGORIES = ["Apple___Black_rot",
"Apple___healthy","Corn_(maize)___healthy","Corn_(maize)___Northern_Leaf
_Blight","Peach___Bacterial_spot","Peach___healthy","Pepper,_bell___Bacteria
l_spot","Pepper,_bell___healthy","Potato___Early_blight","Potato___healthy","
Potato___Late_blight","Tomato___Bacterial_spot","Tomato___Late_blight","To
mato___Leaf_Mold"]



root = Tk()
root.title("Fertilizer Detection System")
root.state('zoomed')
root.configure(bg='#D3D3D3')
root.resizable(width = True, height = True)
value = StringVar()
panel = Label(root)
model = tf.keras.models.load_model("CNN.model")
```

```
# import the opencv library
import cv2

def Camera():
    # define a video capture object
    vid = cv2.VideoCapture(0, cv2.CAP_DSHOW)

    while(True):

        # Capture the video frame
        # by frame
        ret, frame = vid.read()

        # Display the resulting frame
        cv2.imshow('frame', frame)
        cv2.imwrite(r'C:\Users\divya\OneDrive\Desktop\JIT-
FERTILIZER\main.jpg',frame)

        # the 'q' button is set as the
        # quitting button you may use any
        # desired button of your choice
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    # After the loop release the cap object
    vid.release()
    # Destroy all the windows
    cv2.destroyAllWindows()

def prepare(file):
    IMG_SIZE = 150
    img_array = cv2.imread(file,1)
    #img_array=cv2.equalizeHist(img_array)
    #ret,img_array = cv2.threshold(img_array,170,155,cv2.THRESH_BINARY)
    #img_array = cv2.Canny(img_array, threshold1=50, threshold2=10)
    #img_array = cv2.medianBlur(img_array,1)
    #cv2.imshow("hello",img_array)
    new_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
    new_array=np.expand_dims(new_array, axis=0)
```

```
    return new_array

def detect(filename):
    prediction = model.predict(prepare(filename))
    prediction = list(prediction[0])
    print(prediction)
    l=CATEGORIES[prediction.index(max(prediction))]
    print(CATEGORIES[prediction.index(max(prediction))])
    value.set(CATEGORIES[prediction.index(max(prediction))])
    i=int(prediction.index(max(prediction)))
    j=0
    import csv
    file = open('solution.csv')
    type(file)
    csvreader = csv.reader(file)
    header = []
    header = next(csvreader)
    for row in csvreader:
        if j == int(prediction.index(max(prediction))):
            x=header[0]+" : "+row[0]
            tkinter.messagebox.showinfo("",x)
        j=j+1


def ClickAction(event=None):
    filename = filedialog.askopenfilename()
    img = PIL.Image.open(filename)
    img = img.resize((250,250))
    img = PIL.ImageTk.PhotoImage(img)
    global panel
    panel = Label(root, image = img)
    panel.image = img
    panel = panel.place(relx=0.435,rely=0.3)
    detect(filename)

button = Button(root, text='ACTIVATE CAMERA', font=(None, 18),
activeforeground='red', bd=20, bg='cyan', relief=RAISED, height=3, width=20,
command=Camera)
button = button.place(relx=0, rely=0.05)
```

```
button = Button(root, text='CHOOSE FILE', font=(None, 18),
activeforeground='red', bd=20, bg='cyan', relief=RAISED, height=3, width=20,
command=ClickAction)
button = button.place(relx=0.40, rely=0.05)
result = Label(root, textvariable=value, font=(None, 20))
result = result.place(relx=0.465,rely=0.7)
root.mainloop()
```

```
<!DOCTYPE
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>A flying astronaut</title>
    <style>img[alt="www.000webhost.com"]{display: none}

*{
  margin: 0;
  padding: 0;
 }
 body{
  background-color: #102037;
  overflow: hidden;
 }
 @-webkit-keyframes snow {
  0% { opacity: 0; transform: translateY(0px); }
  20%{ opacity: 1;}
  100% { opacity: 1; transform: translateY(650px); }
 }
 @-moz-keyframes snow {
  0% { opacity: 0; transform: translateY(0px); }
  20%{ opacity: 1;}
  100% { opacity: 1; transform: translateY(650px); }
 }
 @keyframes snow {
  0% { opacity: 0; transform: translateY(0px); }
```

```
  20%{ opacity: 1;}
  100% { opacity: 1; transform: translateY(650px); }
}
@-webkit-keyframes astronaut{
  0%{
    transform: rotate(0deg);
  }
  100%{
    transform: rotate(360deg);
  }
}
.box-of-star1,
.box-of-star2,
.box-of-star3,
.box-of-star4{
  width: 100%;
  position: absolute;
  z-index: 10;
  left: 0;
  transform: translateY(650px);
}
.box-of-star1{
  -webkit-animation: snow 5s linear infinite;
}
.box-of-star2{
  -webkit-animation: snow 5s -1.64s linear infinite;
}
.box-of-star3{
  -webkit-animation: snow 5s -2.30s linear infinite;
}
.box-of-star4{
  -webkit-animation: snow 5s -3.30s linear infinite;
}
.star{
  width: 3px;
  height: 3px;
  border-radius: 50%;
  background-color: #FFF;
  position: absolute;
  z-index: 10;
```

```
  opacity: .7;
}
.star:before{
  content: "";
  width: 6px;
  height: 6px;
  border-radius: 50%;
  background-color: #FFF;
  position: absolute;
  z-index: 10;
  top: 40px;
  left: 70px;
  opacity: .7;
}
.star:after{
  content: "";
  width: 8px;
  height: 8px;
  border-radius: 50%;
  background-color: #FFF;
  position: absolute;
  z-index: 10;
  top: 8px;
  left: 170px;
  opacity: .9;
}
.star-position1{
  top: 30px;
  left: 20px;
}
.star-position2{
  top: 110px;
  left: 250px;
}
.star-position3{
  top: 60px;
  left: 570px;
}
.star-position4{
  top: 120px;
```

```
  left: 900px;
 }
 .star-position5{
  top: 20px;
  left: 1120px;
 }
 .star-position6{
  top: 90px;
  left: 1280px;
 }
 .star-position7{
  top: 30px;
  left: 1480px;
 }
 .astronaut{
  width: 250px;
  height: 300px;
  position: absolute;
  z-index: 11;
  top: calc(50% - 150px);
  left: calc(50% - 125px);
  -webkit-animation: astronaut 5s linear infinite;
 }
 .schoolbag{
  width: 100px;
  height: 150px;
  position: absolute;
  z-index: 1;
  top: calc(50% - 75px);
  left: calc(50% - 50px);
  background-color: #94b7ca;
  border-radius: 50px 50px 0 0 / 30px 30px 0 0;
 }
 .head{
  width: 97px;
  height: 80px;
  position: absolute;
  z-index: 3;
  background: -webkit-linear-gradient(left, #e3e8eb 0%,
#e3e8eb 50%, #fbfdfa 50%, #fbfdfa 100%);
```

```
    border-radius: 50%;
    top: 34px;
    left: calc(50% - 47.5px);
  }
  .head:after{
    content: "";
    width: 60px;
    height: 50px;
    position: absolute;
    top: calc(50% - 25px);
    left: calc(50% - 30px);
    background: -webkit-linear-gradient(top, #15aece 0%,
#15aece 50%, #0391bf 50%, #0391bf 100%);
    border-radius: 15px;
  }
  .head:before{
    content: "";
    width: 12px;
    height: 25px;
    position: absolute;
    top: calc(50% - 12.5px);
    left: -4px;
    background-color: #618095;
    border-radius: 5px;
    box-shadow: 92px 0px 0px #618095;
  }
  .body{
    width: 85px;
    height: 100px;
    position: absolute;
    z-index: 2;
    background-color: #fffbff;
    border-radius: 40px / 20px;
    top: 105px;
    left: calc(50% - 41px);
    background: -webkit-linear-gradient(left, #e3e8eb 0%,
#e3e8eb 50%, #fbfdfa 50%, #fbfdfa 100%);
  }
  .panel{
    width: 60px;
```

```
  height: 40px;
  position: absolute;
  top: 20px;
  left: calc(50% - 30px);
  background-color: #b7cceb;
}
.panel:before{
  content: "";
  width: 30px;
  height: 5px;
  position: absolute;
  top: 9px;
  left: 7px;
  background-color: #fbfdfa;
  box-shadow: 0px 9px 0px #fbfdfa, 0px 18px 0px #fbfdfa;
}
.panel:after{
  content: "";
  width: 8px;
  height: 8px;
  position: absolute;
  top: 9px;
  right: 7px;
  background-color: #fbfdfa;
  border-radius: 50%;
  box-shadow: 0px 14px 0px 2px #fbfdfa;
}
.arm{
  width: 80px;
  height: 30px;
  position: absolute;
  top: 121px;
  z-index: 2;
}
.arm-left{
  left: 30px;
  background-color: #e3e8eb;
  border-radius: 0 0 0 39px;
}
.arm-right{
```

```
  right: 30px;
  background-color: #fbfdfa;
  border-radius: 0 0 39px 0;
}
.arm-left:before,
.arm-right:before{
  content: "";
  width: 30px;
  height: 70px;
  position: absolute;
  top: -40px;
}
.arm-left:before{
  border-radius: 50px 50px 0px 120px / 50px 50px 0 110px;
  left: 0;
  background-color: #e3e8eb;
}
.arm-right:before{
  border-radius: 50px 50px 120px 0 / 50px 50px 110px 0;
  right: 0;
  background-color: #fbfdfa;
}
.arm-left:after,
.arm-right:after{
  content: "";
  width: 30px;
  height: 10px;
  position: absolute;
  top: -24px;
}
.arm-left:after{
  background-color: #6e91a4;
  left: 0;
}
.arm-right:after{
  right: 0;
  background-color: #b6d2e0;
}
.leg{
    width: 30px;
```

```
        height: 40px;
        position: absolute;
        z-index: 2;
        bottom: 70px;
    }
    .leg-left{
        left: 76px;
        background-color: #e3e8eb;
        transform: rotate(20deg);
    }
    .leg-right{
        right: 73px;
        background-color: #fbfdfa;
        transform: rotate(-20deg)
    }
    .leg-left:before,
    .leg-right:before{
        content: '';
        width: 50px;
        height: 25px;
        position: absolute;
        bottom: -26px;
    }
    .leg-left:before{
        left: -20px;
        background-color: #e3e8eb;
        border-radius: 30px 0 0 0 ;
    }
  .button {
    background-color: #4CAF50; /* Green */
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin: 4px 2px;
    cursor: pointer;
  }
```

```
.button1 {width: 250px;}
  .leg-right:before{
      right: -20px;
      background-color: #e3e8eb;
      border-radius: 0 30px 0 0 ;
  }
</style>
</head>
<body style="color:white;">
<br>
<br>
<br>
    <div class="box-of-star1">
        <div class="star star-position1"></div>
        <div class="star star-position2"></div>
        <div class="star star-position3"></div>
        <div class="star star-position4"></div>
        <div class="star star-position5"></div>
        <div class="star star-position6"></div>
        <div class="star star-position7"></div>
      </div>
      <div class="box-of-star2">
        <div class="star star-position1"></div>
        <div class="star star-position2"></div>
        <div class="star star-position3"></div>
        <div class="star star-position4"></div>
        <div class="star star-position5"></div>
        <div class="star star-position6"></div>
        <div class="star star-position7"></div>
      </div>
      <div class="box-of-star3">
        <div class="star star-position1"></div>
        <div class="star star-position2"></div>
        <div class="star star-position3"></div>
        <div class="star star-position4"></div>
        <div class="star star-position5"></div>
        <div class="star star-position6"></div>
        <div class="star star-position7"></div>
      </div>
      <div class="box-of-star4">
```

```
        <div class="star star-position1"></div>
        <div class="star star-position2"></div>
        <div class="star star-position3"></div>
        <div class="star star-position4"></div>
        <div class="star star-position5"></div>
        <div class="star star-position6"></div>
        <div class="star star-position7"></div>
      </div>

<center>

 <h1>Leaf Disease Detection</h1>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>

<h3>A leaf spot is a limited, discoloured, diseased area of a
leaf that is caused by fungal, bacterial or viral plant diseases,
or by injuries from nematodes, insects, environmental factors,
toxicity or herbicides.</h3>
        <button class="button button1"> <a href="/my-
link/">Press to Analyse</a></button>
</center>
      </body>

</html>
```

GitHub & Project Demo Link:

GitHub: https://github.com/IBM-EPBL/IBM-Project-18250-1659681882

Project Demo Link: https://youtu.be/XUiu0xwagjE