Fertilizers Recommendation System for Disease Prediction

PROJECT REPORT

Submitted by

Team ID: PNT2022TMID52565

ESHA A(CITC1905012)

KOUSIKA M (CITC1905024)

SHUBASHINI J (CITC1905049)

ASIYA BEGAM M (CITC2005202)

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

COIMBATORE INSTITUTE OF TECHNOLOGY

(Government Aided Autonomous Institution Affiliated to Anna University)

COIMBATORE-641 014

1 INTRODUCTION

Fertilizer Recommendation system for disease Prediction is a simple ML and DL based website which recommends the best crop to grow, fertilizers to use and the diseases caught by your crops.

1.1 PROJECT OVERVIEW:

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. An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

1.2 PUPRPOSE:

The main purpose of this project is used to test the Vegitables and fruits of the plant's sample and identify the diseases. Then provide the recommended fertilizer for that disease. The process starts with the user has to take an image of the affected leaves and then uploading that image. It scans the leaves with the help of the CNN layer and machine learning technique. Machine learning is particularly effective in detecting and recognizing plant illnesses, and it can provide early disease sign identification. Plant disease specialists can examine the digital photos processed with digital image processing to identify blights on plants. computer vision and image processing applications Processing methods merely help farmers throughout all regions. It detects the type of that disease and finds the recommended fertilizer which should be used for that disease.

Traditional approaches depend on experts, encounters, and guides, but the bulk of them are expensive, time-consuming, and labor-intensive, and it might be challenging to precisely identify them. As a result, it seems crucial for trade and biology in agriculture that a quick and accurate method be used to identify plant infections. If the illness is not correctly detected, disease control measures could be a waste of time and money and result in further plant loss. A deep learning-based model is what our project suggests, and it will be trained using images of crop leaves that are both healthy and diseased that are taken from a dataset. The model will accomplish its objective by grouping images of leaves into harmful categories based on flaw patterns.

2 LITERATURE SURVEY:

2.1 EXISTING PROBLEM:

In our case When a pathogen that is already present or invades successfully to plant host tissues and cells results in plant disease. It is important to fix the problem because Plant diseases reduce the amount of food available to humans by ultimately interfering with crop yields. This can cause inadequate food for humans which result in starvation or death in the worst cases.

2.2 REFERENCES:

[1] Dr.P. Pandi Selvi, P. Poornima (2021). "Soil Based Fertilizer Recommendation System for Crop Disease Prediction System".

[2] Devdatta A. Bondre, Mr. Santosh Mahagaonkar, (2019). "Prediction of Crop Yield and Fertilizer Recommendation Using Machine Learning Algorithms".

[3] Suma V, R Amog Shetty, Rishab F Tated, Sunku Rohan, Triveni S Pujar, (2019). "CNN-based Leaf Disease Identification and Remedy Recommendation System".

[4] Limin Chuan, Ping He, Mirasol F. Pampolino, Adrian M. Johnston, Jiyun Jin, Xinpeng Xu, Shicheng Zhao, Shaojun Qiu, and Wei Zhou, (2013). "Establishing a Scientific Basis for Fertilizer Recommendations for Wheat in China".

[5] R. Neela, P.(2019) "Fertilizers Recommendation System For Disease Prediction In Tree Leave" International journal of scientific & technology research volume 8, issue 11, november 2019.

2.3 PROBLEM STATEMENT DEFINITION:

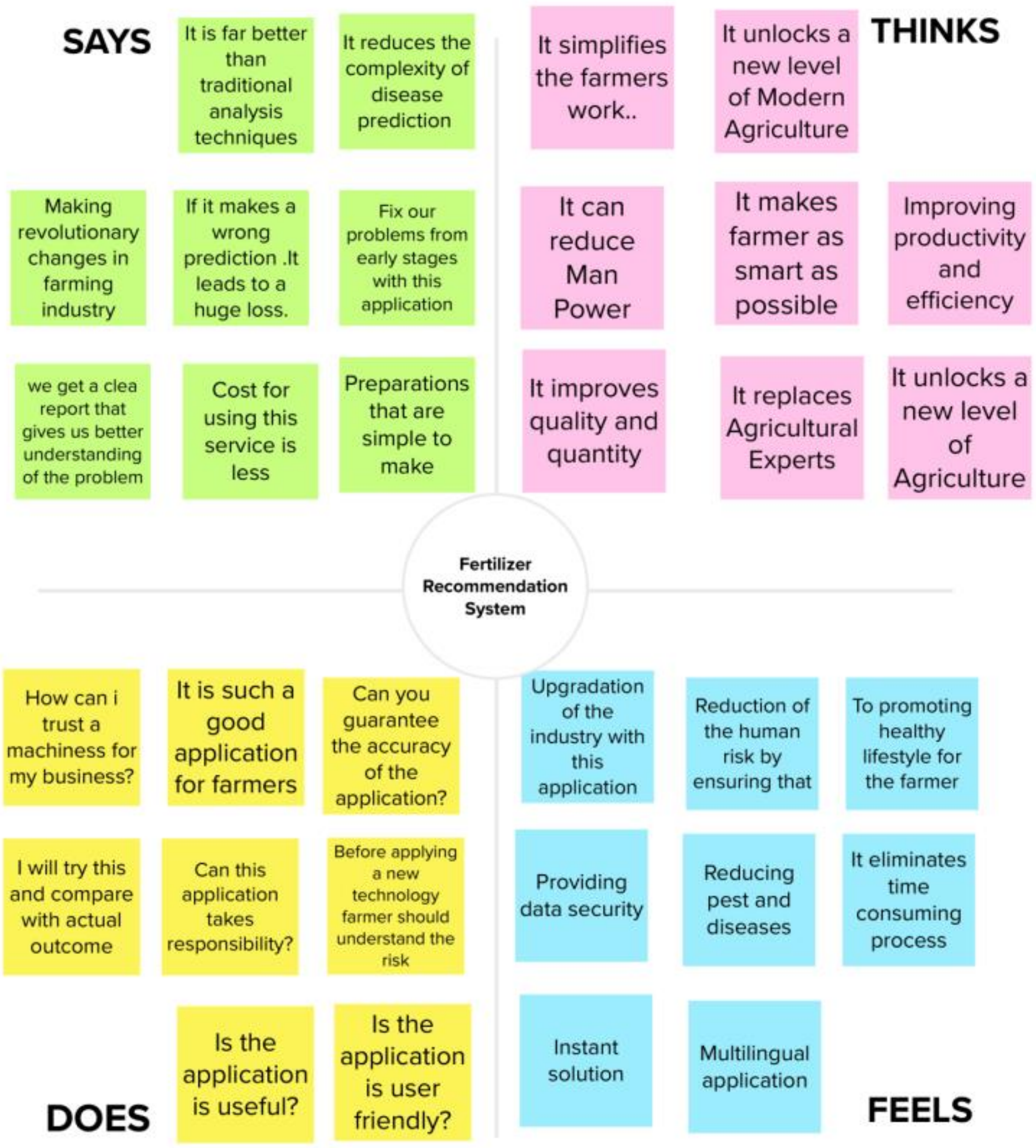
Agriculture is the most important sector in today's life. Most plants are affected by a wide variety of bacterial and fungal diseases. Diseases in plants placed a major constraint on the production and a major threat to food security. Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality. In recent years, the number of diseases in 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.

An automated system is introduced to identify different diseases in plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases. The disease can be found easily in the early stage by looking at changes in the color of the leaves. So, without knowing about the correct disease they use some fertilizersand it doesn’t cure the disease properly. This fertilizer recommendation system helps to findthe accurate disease and helps them to cure the disease and increase in the growth of plants.

3 IDEATION AND PROPOSED SOLUTION:

3.1 EMPATHY MAP CANVAS:

An empathy map is a collaborative tool team can use to gain a deeper insight into their customers.



SAYS

It is far better than traditional analysis techniques

It reduces the complexity of disease prediction

Making revolutionary changes in farming industry

If it makes a wrong prediction .It leads to a huge loss.

Fix our problems from early stages with this application

we get a clea report that gives us better understanding of the problem

Cost for using this service is less

Preparations that are simple to make

THINKS

It simplifies the farmers work..

It unlocks a new level of Modern Agriculture

It can reduce Man Power

It makes farmer as smart as possible

Improving productivity and efficiency

It improves quality and quantity

It replaces Agricultural Experts

It unlocks a new level of Agriculture

Fertilizer Recommendation System

DOES

How can i trust a machiness for my business?

It is such a good application for farmers

Can you guarantee the accuracy of the application?

I will try this and compare with actual outcome

Can this application takes responsibility?

Before applying a new technology farmer should understand the risk

Is the application is useful?

Is the application is user friendly?

FEELS

Upgradation of the industry with this application

Reduction of the human risk by ensuring that

To promoting healthy lifestyle for the farmer

Providing data security

Reducing pest and diseases

It eliminates time consuming process

Instant solution

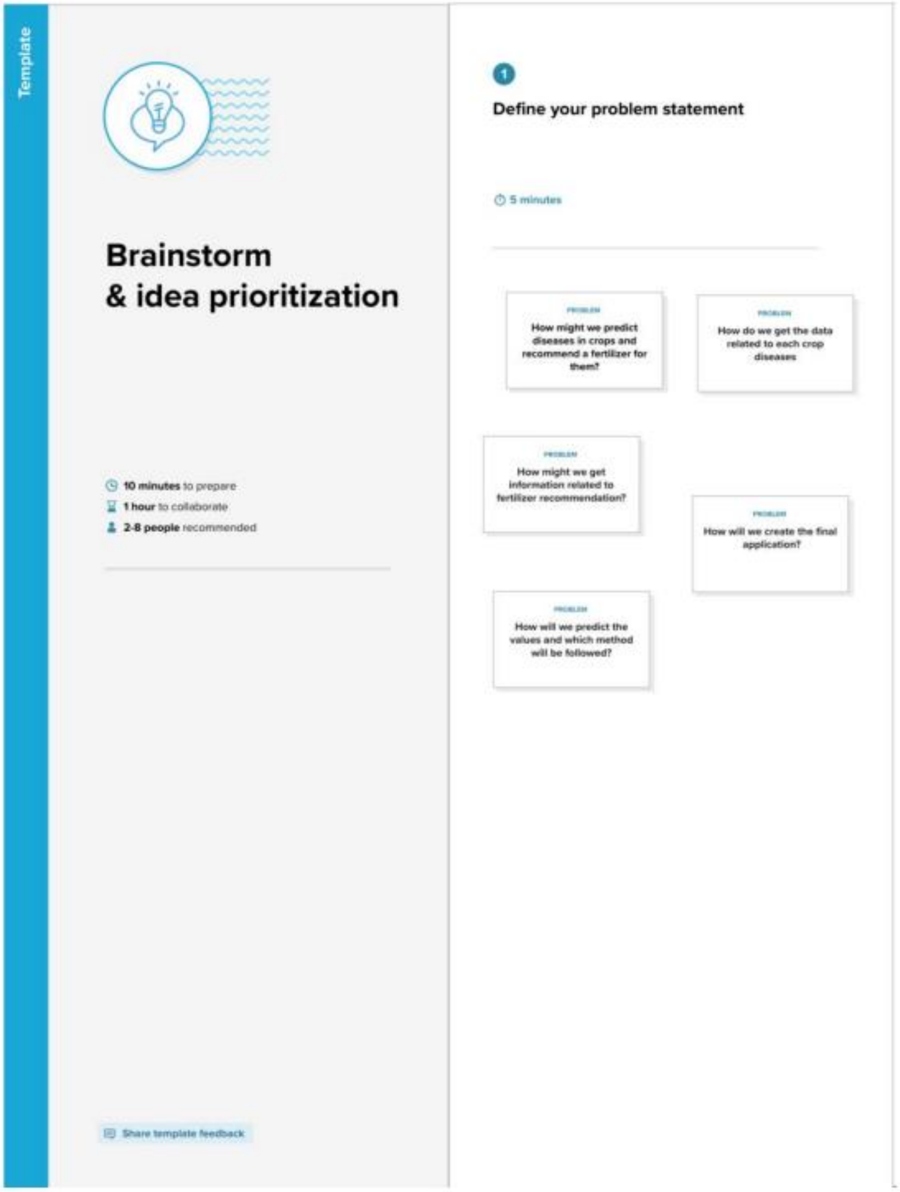
Multilingual application

Fig 3.1 Empathy Map

3.2 IDEATION AND BRAINSTROMING:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem-solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

STEP1: Team gathering, Collaboration, select the Problem statement



Template

Brainstorm
& idea prioritization

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

1

Define your problem statement

5 minutes

PROBLEM
How might we predict diseases in crops and recommend a fertilizer for them?

PROBLEM
How do we get the data related to each crop diseases

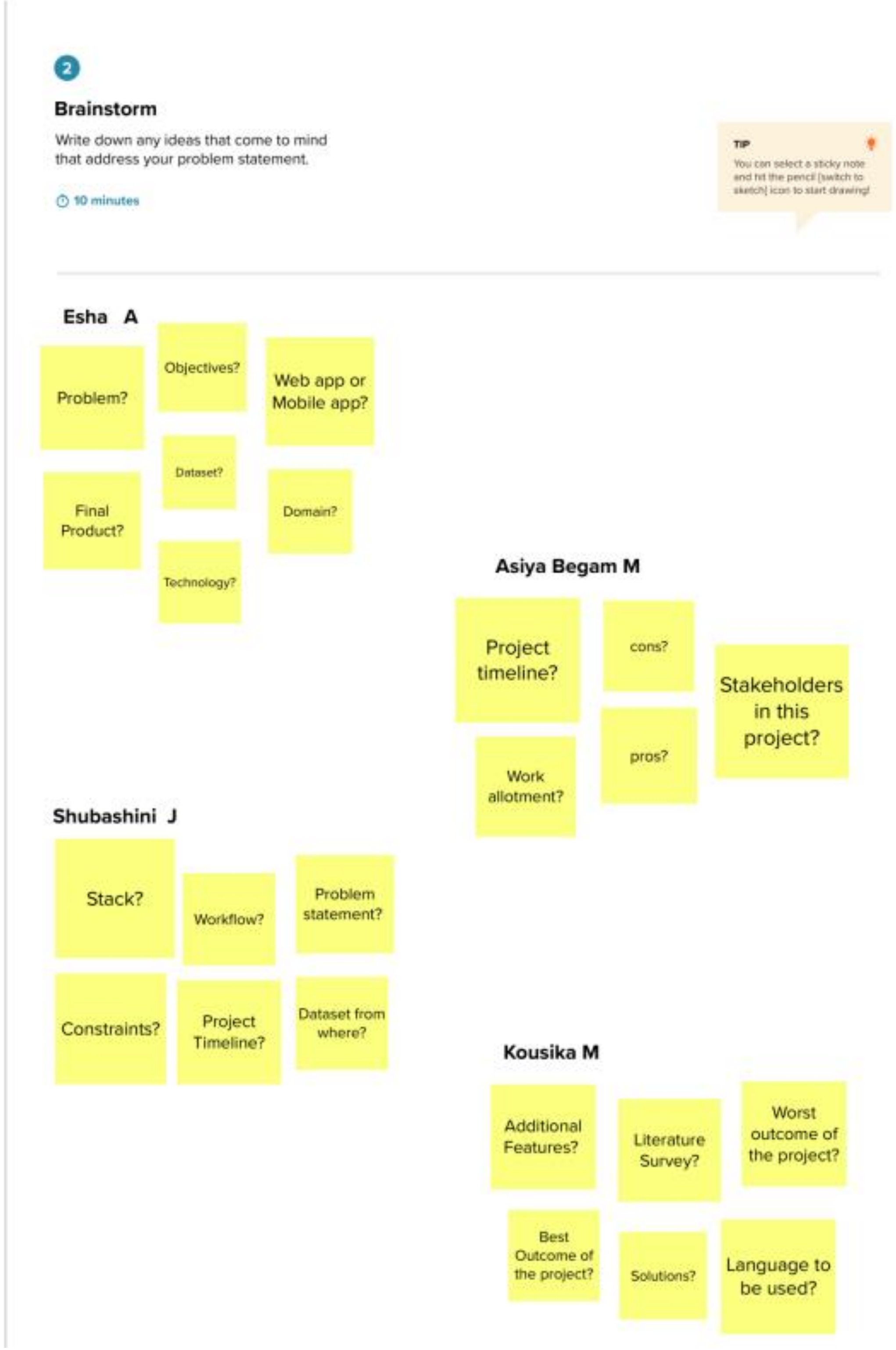
PROBLEM
How might we get information related to fertilizer recommendation?

PROBLEM
How will we create the final application?

PROBLEM
How will we predict the values and which method will be followed?

Fig 3.2 Brainstorming step-1

STEP 2: Brainstorm, Idea listing, Grouping



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Esha A

Problem?

Objectives?

Web app or Mobile app?

Final Product?

Dataset?

Domain?

Technology?

Asiya Begam M

Project timeline?

cons?

Stakeholders in this project?

Work allotment?

pros?

Shubashini J

Stack?

Workflow?

Problem statement?

Constraints?

Project Timeline?

Dataset from where?

Kousika M

Additional Features?

Literature Survey?

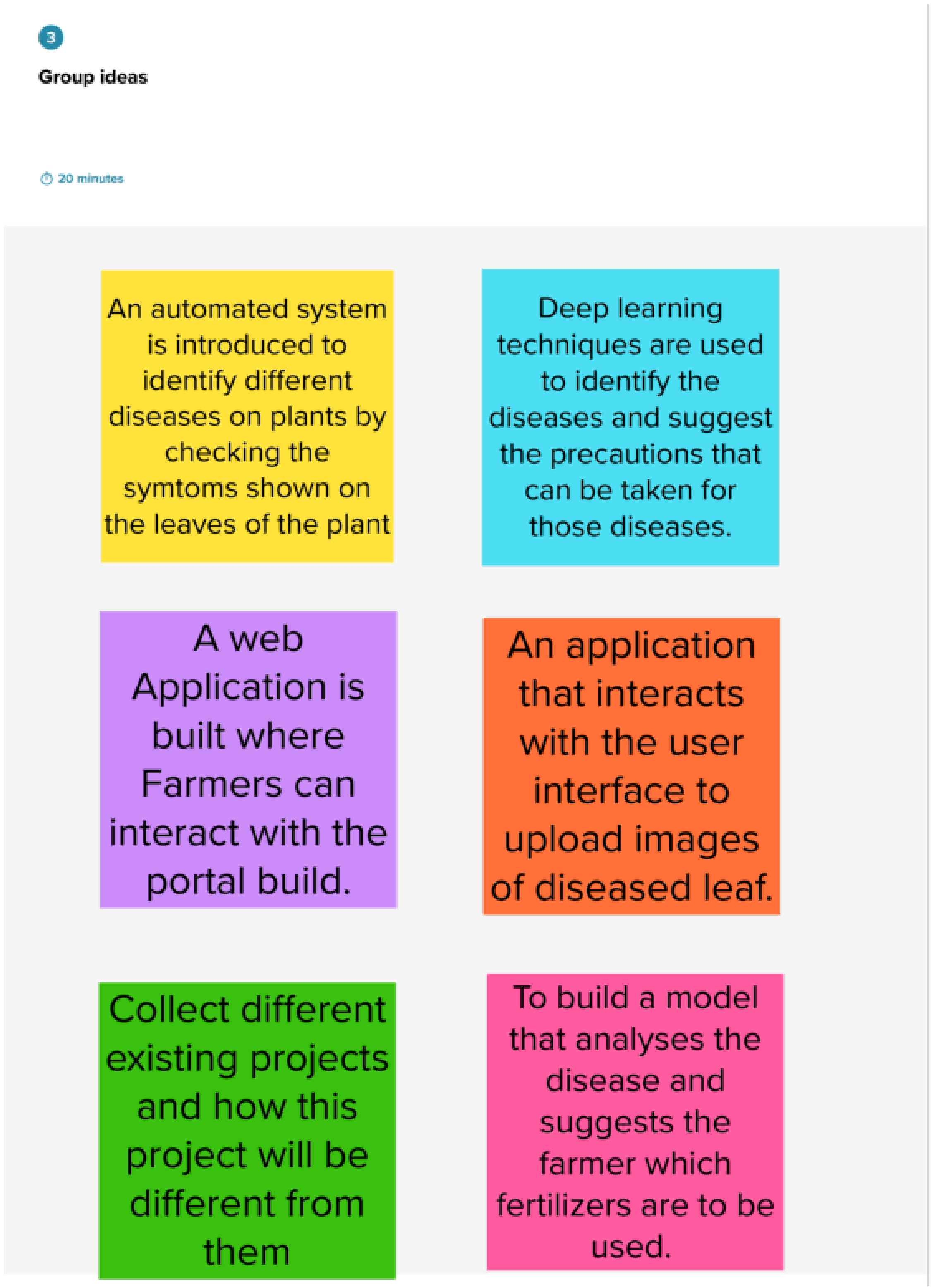
Worst outcome of the project?

Best Outcome of the project?

Solutions?

Language to be used?

Fig 3.3 step-2



3

Group ideas

20 minutes

An automated system is introduced to identify different diseases on plants by checking the symtoms shown on the leaves of the plant

Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

A web Application is built where Farmers can interact with the portal build.

An application that interacts with the user interface to upload images of diseased leaf.

Collect different existing projects and how this project will be different from them

To build a model that analyses the disease and suggests the farmer which fertilizers are to be used.

Fig 3.4 Group ideas

STEP 3: Idea Prioritization

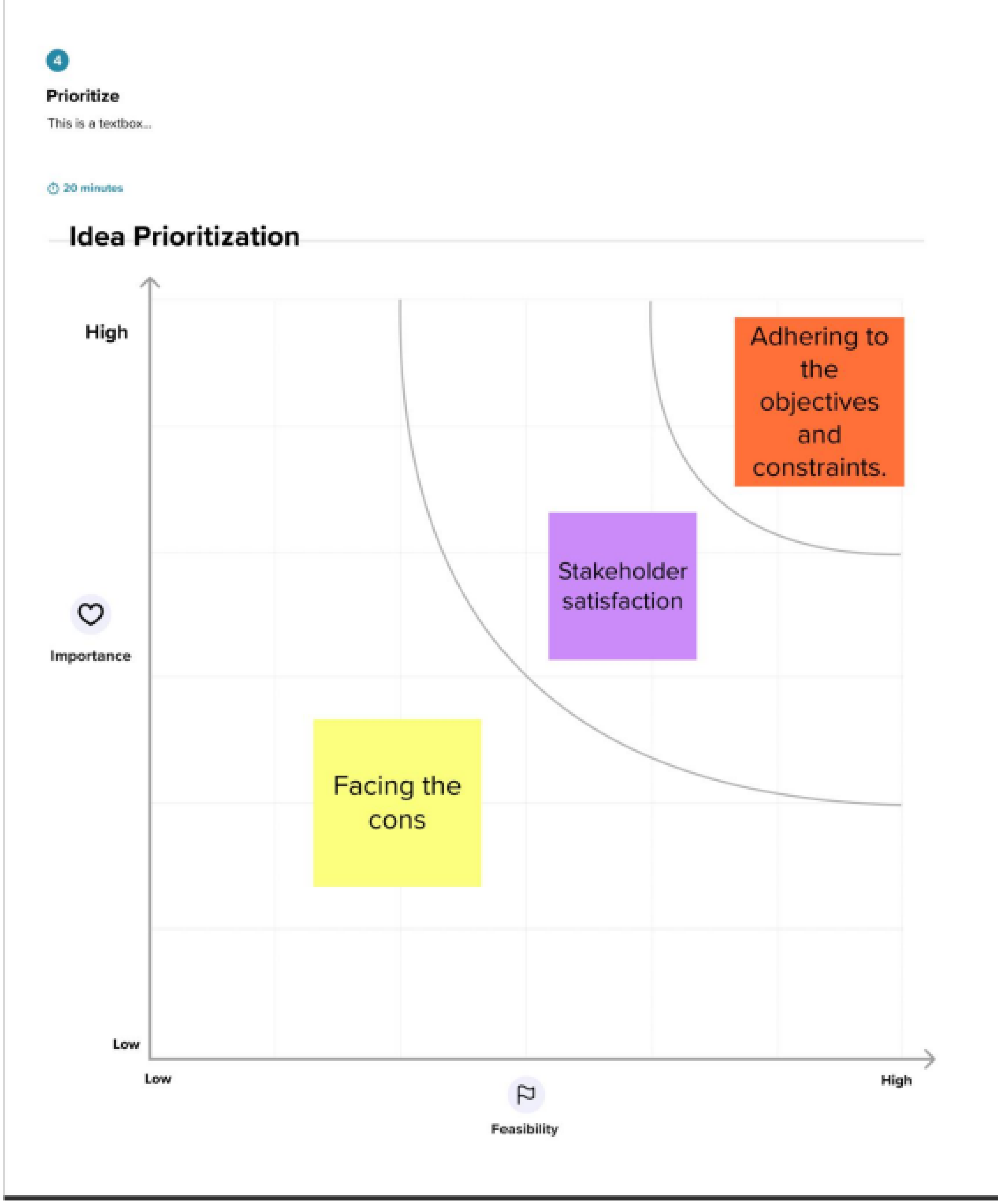
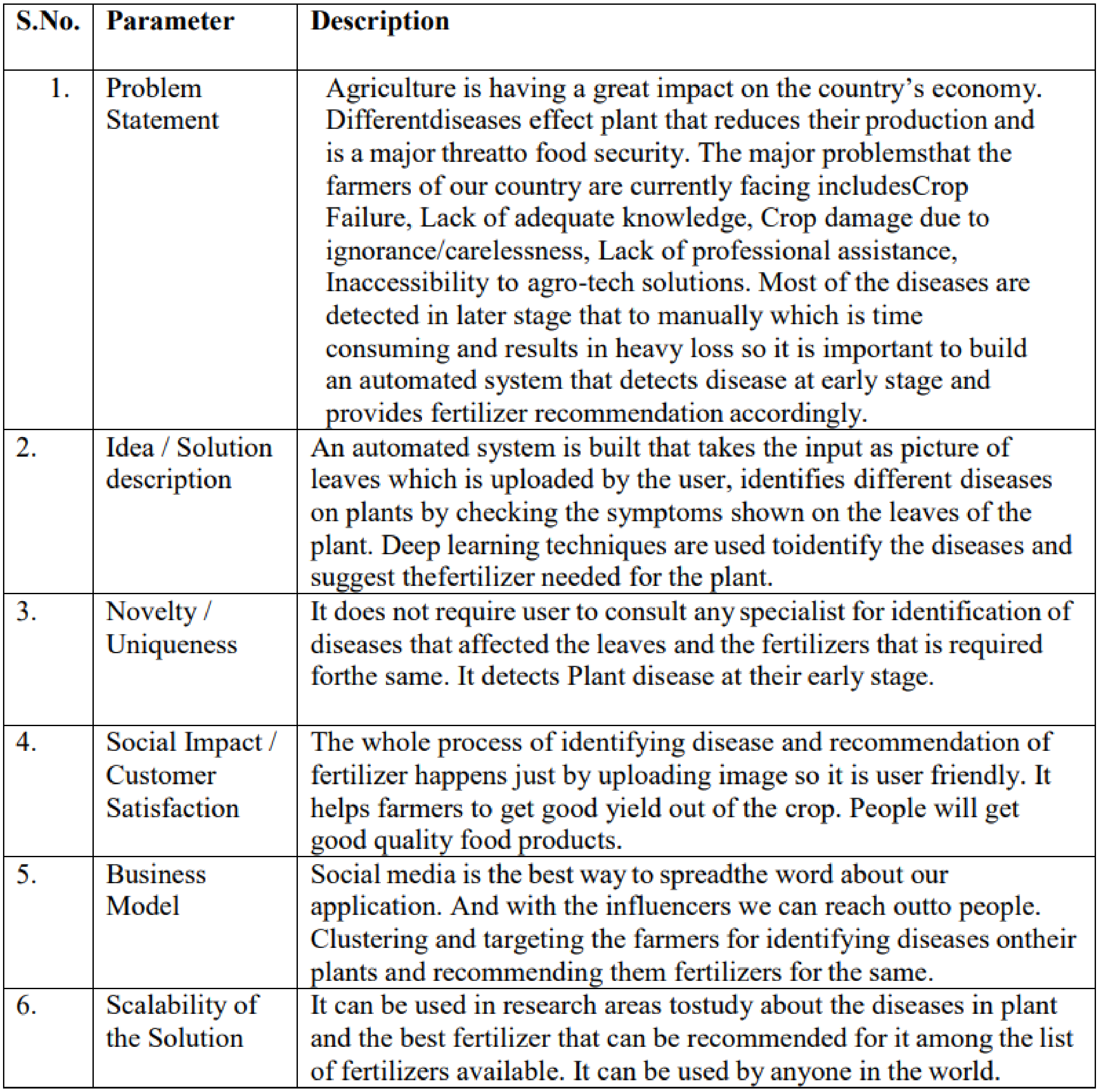


Fig 3.5 step-3

3.3 PROPOSED SOLUTION:

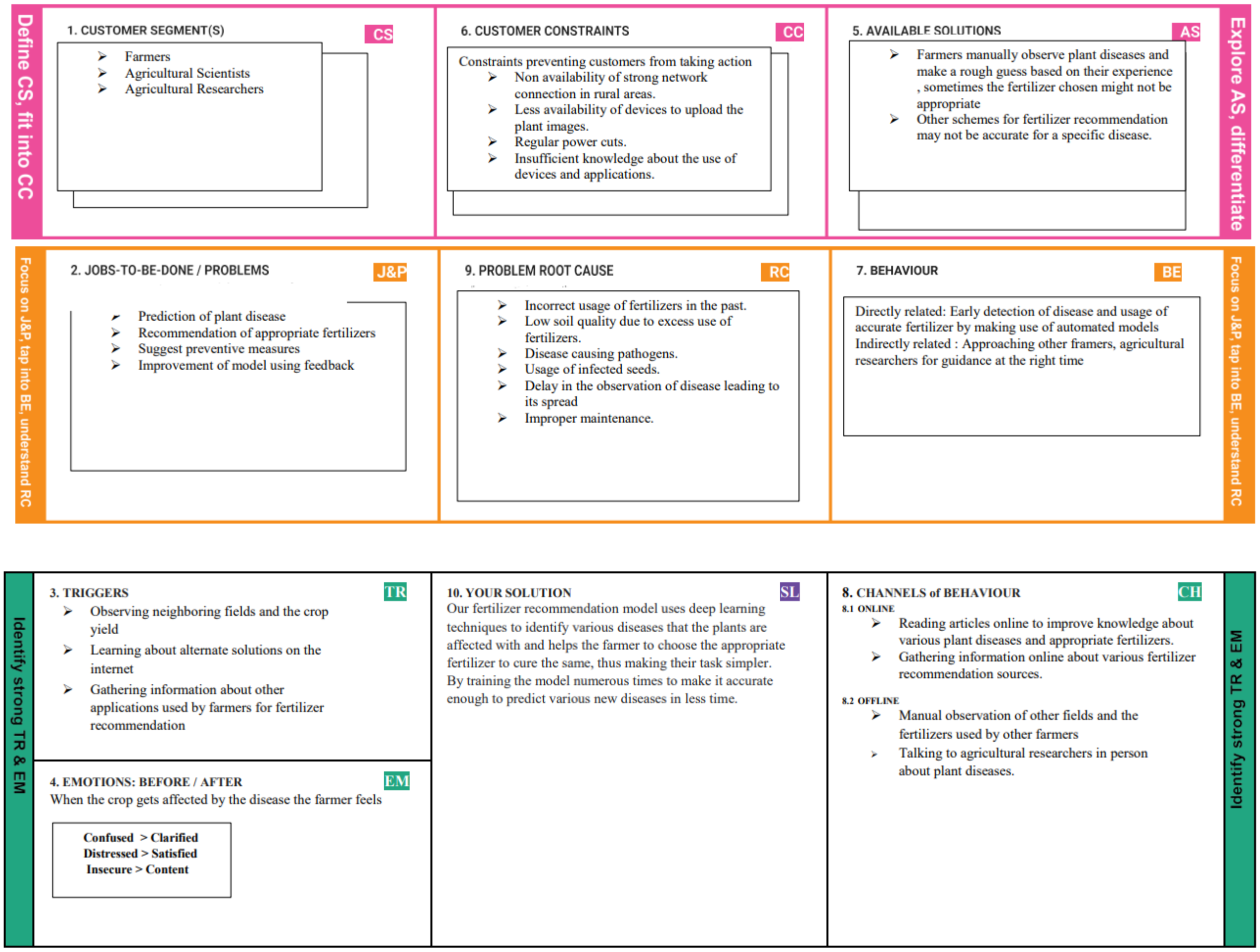
The solution to the problem is machine learning. This system recommends the best crop to grow on your land based on the nutritional value of the soil and along with the climate of that region. It also recommends the best fertilizer for each crop, which is a challenging task. The cultivation recommendation features a database of soil nitrogen, phosphorus and potassium for modern agriculture. The ensemble technique is used to create a recommendation model that combines multiple machine- learning predictions. The models recommend the right harvest based on the value of the soil and the best fertilizer.



S.No.	Parameter	Description
1.	Problem Statement	Agriculture is having a great impact on the country's economy. Differentdiseases effect plant that reduces their production and is a major threatto food security. The major problemsthat the farmers of our country are currently facing includesCrop Failure, Lack of adequate knowledge, Crop damage due to ignorance/carelessness, Lack of professional assistance, Inaccessibility to agro-tech solutions. Most of the diseases are detected in later stage that to manually which is time consuming and results in heavy loss so it is important to build an automated system that detects disease at early stage and provides fertilizer recommendation accordingly.
2.	Idea / Solution description	An automated system is built that takes the input as picture of leaves which is uploaded by the user, identifies different diseases on plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used toidentify the diseases and suggest thefertilizer needed for the plant.
3.	Novelty / Uniqueness	It does not require user to consult any specialist for identification of diseases that affected the leaves and the fertilizers that is required forthe same. It detects Plant disease at their early stage.
4.	Social Impact / Customer Satisfaction	The whole process of identifying disease and recommendation of fertilizer happens just by uploading image so it is user friendly. It helps farmers to get good yield out of the crop. People will get good quality food products.
5.	Business Model	Social media is the best way to spreadthe word about our application. And with the influencers we can reach outto people. Clustering and targeting the farmers for identifying diseases ontheir plants and recommending them fertilizers for the same.
6.	Scalability of the Solution	It can be used in research areas tostudy about the diseases in plant and the best fertilizer that can be recommended for it among the list of fertilizers available. It can be used by anyone in the world.

3.4 PROBLEM SOLUTION FIT:

Problem-Solution canvas is a tool for entrepreneurs, marketers and corporate innovators, which help them, identify solutions with higher chances for a solution adoption, reduce time spent on solution testing and get a better overview of current situation.



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

- Farmers
- Agricultural Scientists
- Agricultural Researchers

6. CUSTOMER CONSTRAINTS CC

Constraints preventing customers from taking action

- Non availability of strong network connection in rural areas.
- Less availability of devices to upload the plant images.
- Regular power cuts.
- Insufficient knowledge about the use of devices and applications.

5. AVAILABLE SOLUTIONS AS

- Farmers manually observe plant diseases and make a rough guess based on their experience , sometimes the fertilizer chosen might not be appropriate
- Other schemes for fertilizer recommendation may not be accurate for a specific disease.

Explore AS, differentiate

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

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

- Prediction of plant disease
- Recommendation of appropriate fertilizers
- Suggest preventive measures
- Improvement of model using feedback

9. PROBLEM ROOT CAUSE RC

- Incorrect usage of fertilizers in the past.
- Low soil quality due to excess use of fertilizers.
- Disease causing pathogens.
- Usage of infected seeds.
- Delay in the observation of disease leading to its spread
- Improper maintenance.

7. BEHAVIOUR BE

Directly related: Early detection of disease and usage of accurate fertilizer by making use of automated models
Indirectly related : Approaching other framers, agricultural researchers for guidance at the right time

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

Identify strong TR & EM

3. TRIGGERS TR

- Observing neighboring fields and the crop yield
- Learning about alternate solutions on the internet
- Gathering information about other applications used by farmers for fertilizer recommendation

4. EMOTIONS: BEFORE / AFTER EM

When the crop gets affected by the disease the farmer feels

Confused > Clarified
Distressed > Satisfied
Insecure > Content

10. YOUR SOLUTION SL

Our fertilizer recommendation model uses deep learning techniques to identify various diseases that the plants are affected with and helps the farmer to choose the appropriate fertilizer to cure the same, thus making their task simpler. By training the model numerous times to make it accurate enough to predict various new diseases in less time.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

- Reading articles online to improve knowledge about various plant diseases and appropriate fertilizers.
- Gathering information online about various fertilizer recommendation sources.

8.2 OFFLINE

- Manual observation of other fields and the fertilizers used by other farmers
- Talking to agricultural researchers in person about plant diseases.

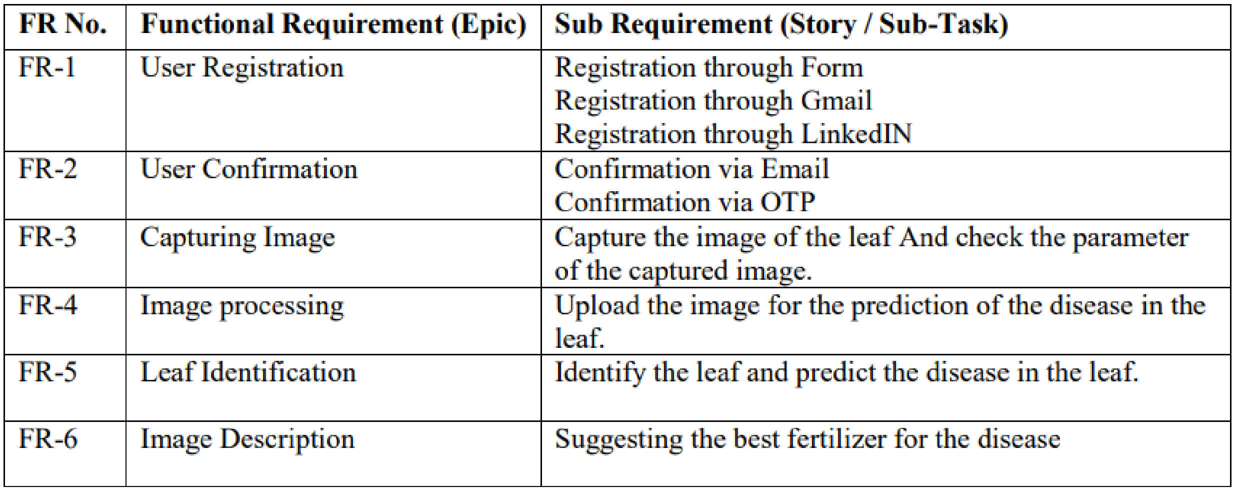
Identify strong TR & EM

Fig 3.6 solution fit

4 REQUIREMENT ANALYSIS:

4.1 FUNCTIONAL REQUIREMENT

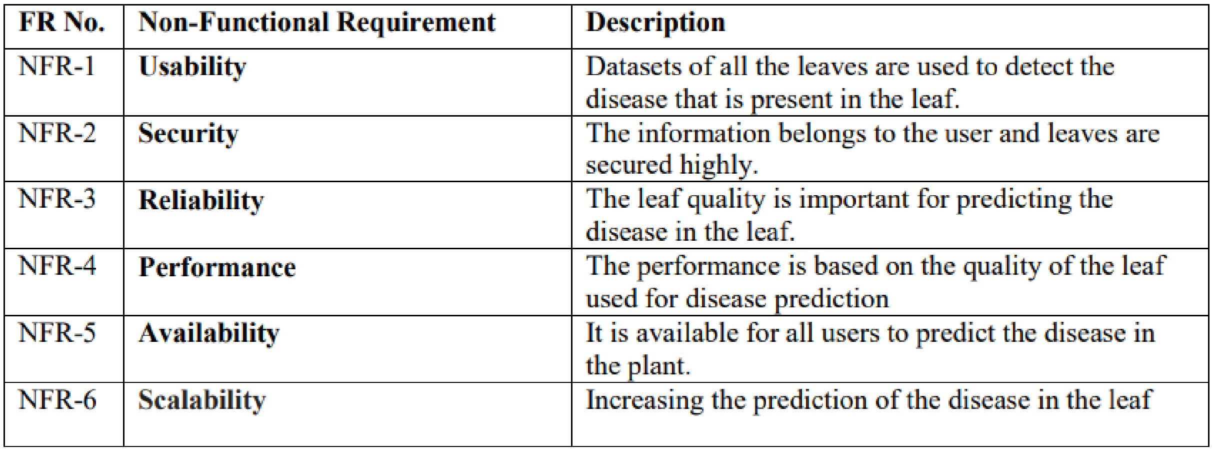
Following are the functional requirements of the proposed solution.



FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
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.
FR-5	Leaf Identification	Identify the leaf and predict the disease in the leaf.
FR-6	Image Description	Suggesting the best fertilizer for the disease

4.2 NON-FUNCTIONAL REQUIREMENT

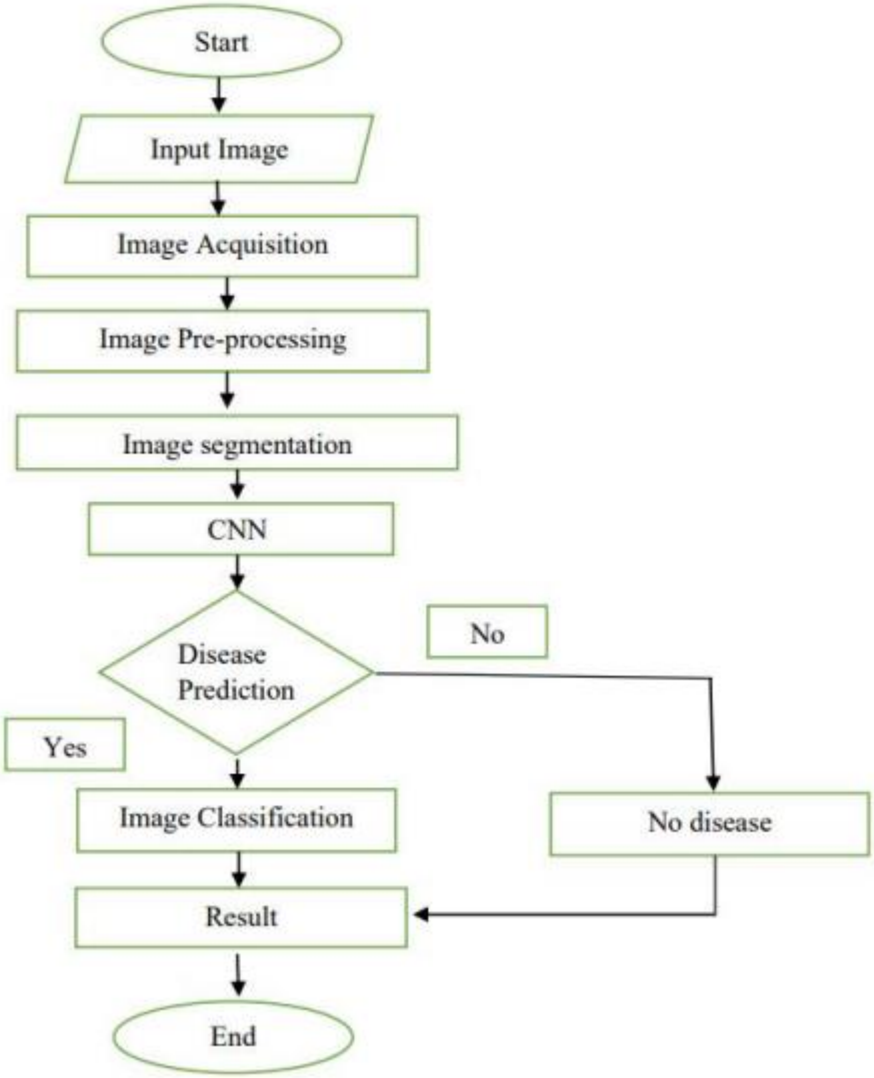
Following are the non-functional requirements of the proposed solution.



FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Datasets of all the leaves are used to detect the disease that is present in the leaf.
NFR-2	**Security**	The information belongs to the user and leaves are secured highly.
NFR-3	**Reliability**	The leaf quality is important for predicting the disease in the 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 users 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



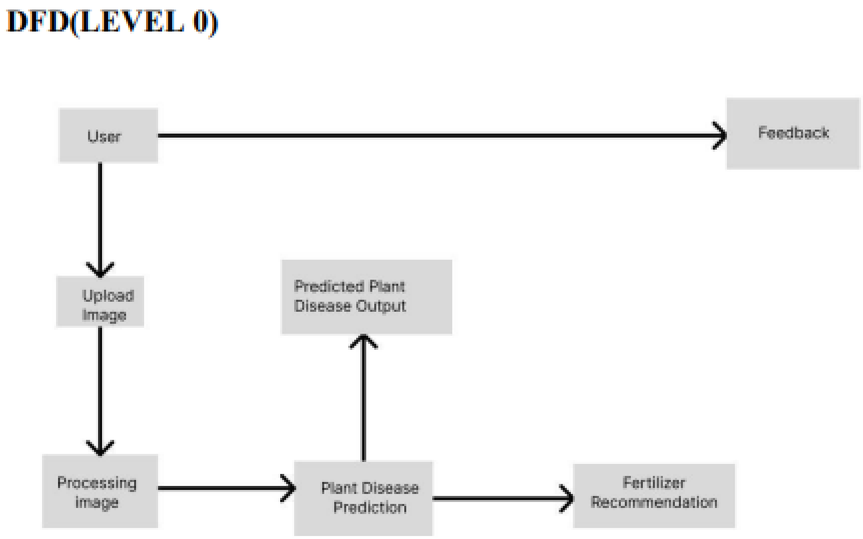


Fig 5.1 data flow diagram

5.2 SOLUTIONS AND TECHNICAL ARCHITECTURE

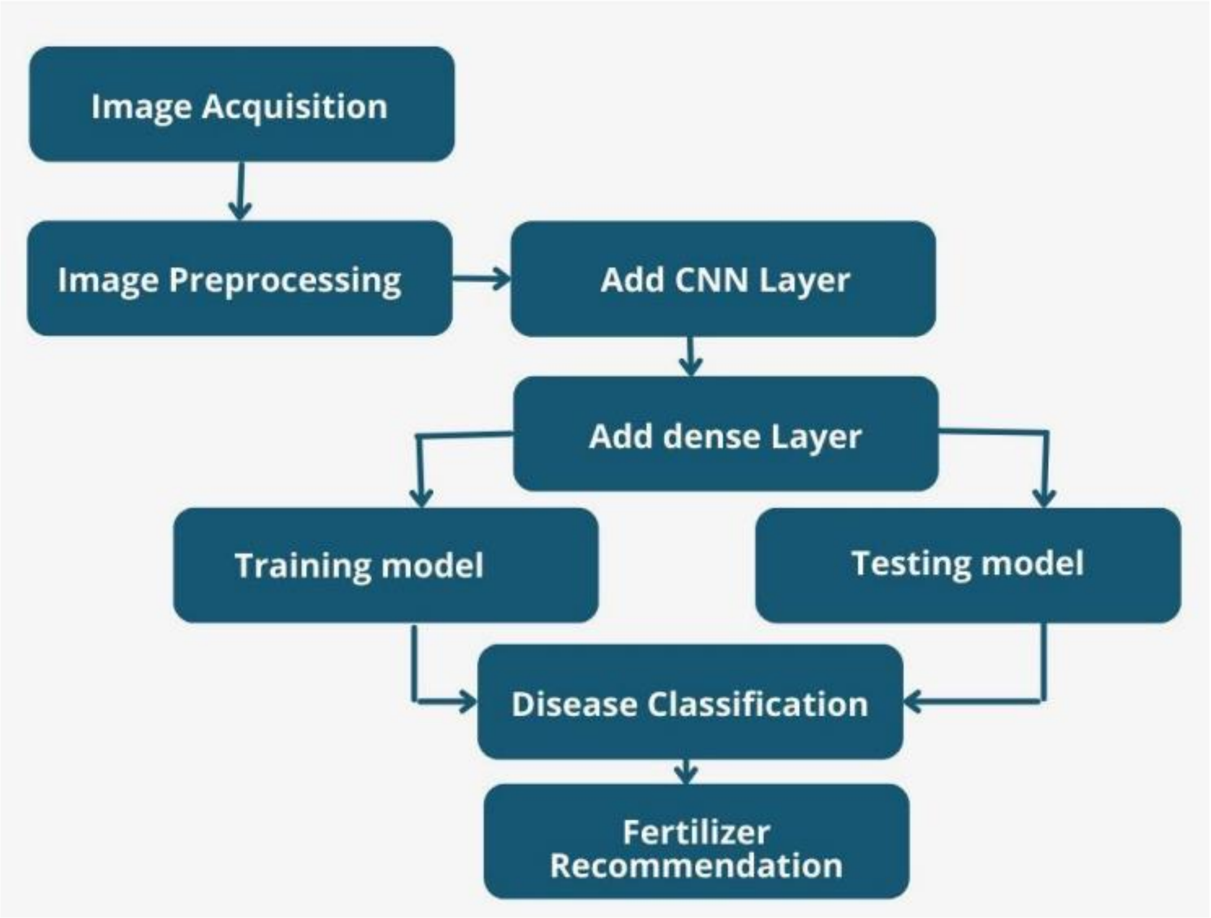
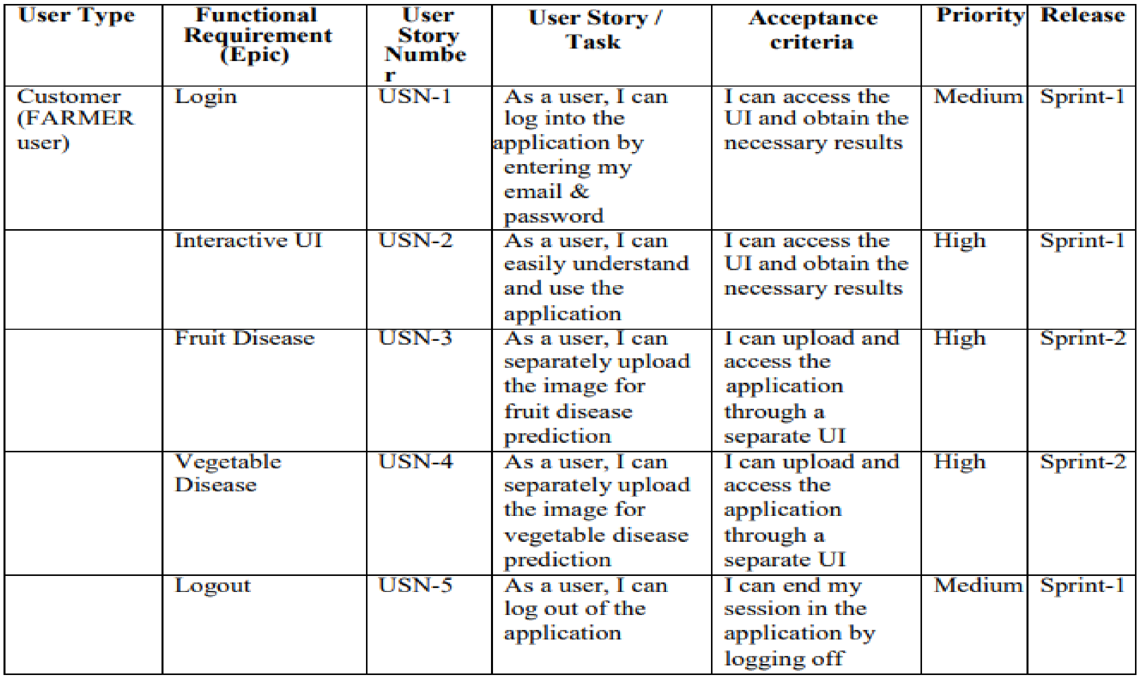


Fig 5.2 technical architecture

Project design is an early phase of the project lifecycle where ideas, processes, resources, and deliverables are planned out. A project design comes before a project plan as it’s a broad overview whereas a project plan includes more detailed information. A project design is the process of outlining all of a project's stages and creating a project plan. It includes a strategy of ideas, resources and processes to achieve project goals and keep within a budget and deadline. Project managers could add flowcharts, sketches, photo impressions and prototypes to help fully outline the project. Project managers present the project plan to senior stakeholders and investors to get final approval before beginning the project. In many cases, project managers create more than one pan for each project so stakeholders can choose which one they think would work best for the project.

5.3 USER STORIES

A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer.

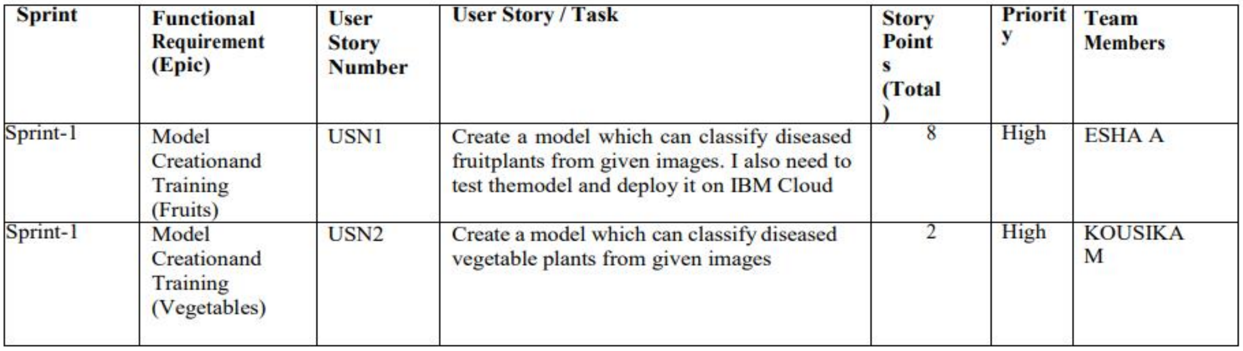


User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (FARMER user)	Login	USN-1	As a user, I can log into the application by entering my email & password	I can access the UI and obtain the necessary results	Medium	Sprint-1
	Interactive UI	USN-2	As a user, I can easily understand and use the application	I can access the UI and obtain the necessary results	High	Sprint-1
	Fruit Disease	USN-3	As a user, I can separately upload the image for fruit disease prediction	I can upload and access the application through a separate UI	High	Sprint-2
	Vegetable Disease	USN-4	As a user, I can separately upload the image for vegetable disease prediction	I can upload and access the application through a separate UI	High	Sprint-2
	Logout	USN-5	As a user, I can log out of the application	I can end my session in the application by logging off	Medium	Sprint-1

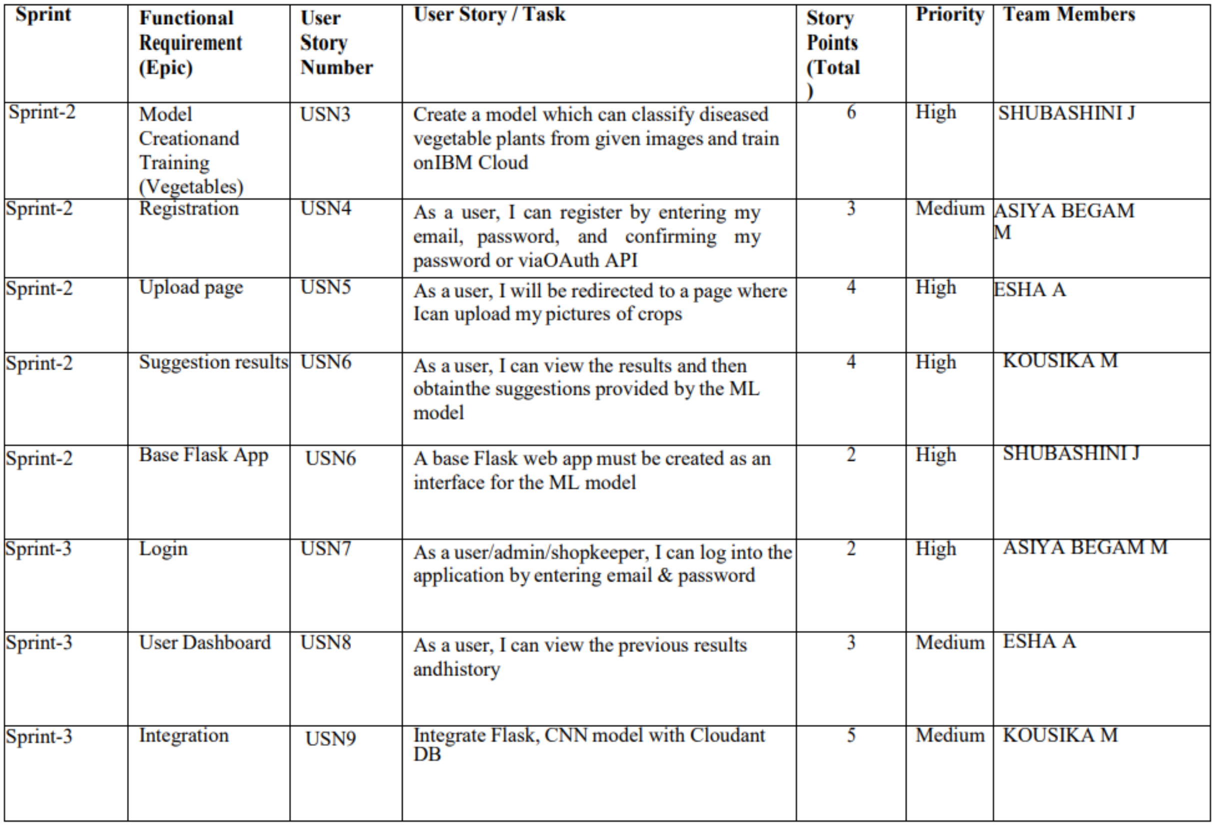
6. PROJECT PLANNING AND SCHEDULING:

6.1 SPRINT PLANNING AND ESTIMATION:

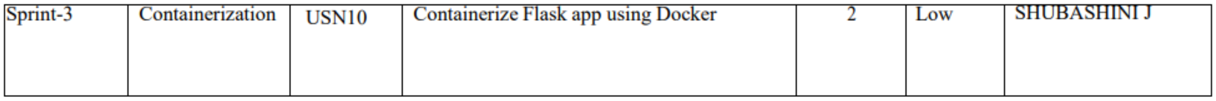
The objective of the Estimation would be to consider the User Stories for the Sprint by Priority and by the Ability of the team to deliver during the Time Box of the Sprint.



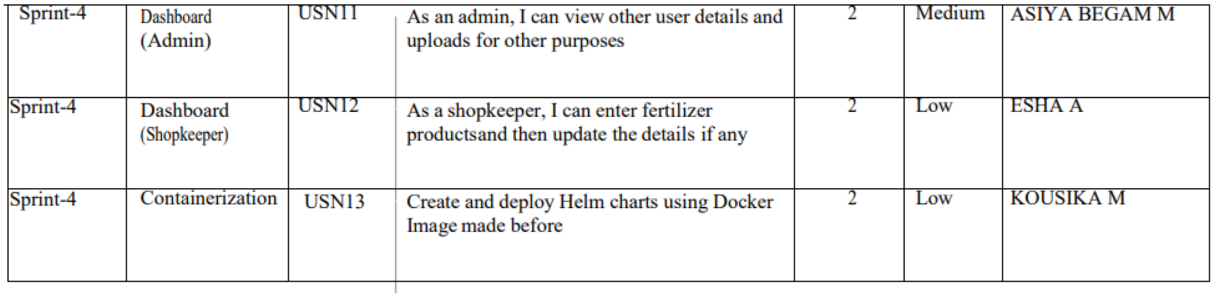
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points (Total)	Priority	Team Members
Sprint-1	Model Creationand Training (Fruits)	USN1	Create a model which can classify diseased fruitplants from given images. I also need to test themodel and deploy it on IBM Cloud	8	High	ESHA A
Sprint-1	Model Creationand Training (Vegetables)	USN2	Create a model which can classify diseased vegetable plants from given images	2	High	KOUSIKA M



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points (Total)	Priority	Team Members
Sprint-2	Model Creationand Training (Vegetables)	USN3	Create a model which can classify diseased vegetable plants from given images and train onIBM Cloud	6	High	SHUBASHINI J
Sprint-2	Registration	USN4	As a user, I can register by entering my email, password, and confirming my password or viaOAuth API	3	Medium	ASIYA BEGAM M
Sprint-2	Upload page	USN5	As a user, I will be redirected to a page where Ican upload my pictures of crops	4	High	ESHA A
Sprint-2	Suggestion results	USN6	As a user, I can view the results and then obtainthe suggestions provided by the ML model	4	High	KOUSIKA M
Sprint-2	Base Flask App	USN6	A base Flask web app must be created as an interface for the ML model	2	High	SHUBASHINI J
Sprint-3	Login	USN7	As a user/admin/shopkeeper, I can log into the application by entering email & password	2	High	ASIYA BEGAM M
Sprint-3	User Dashboard	USN8	As a user, I can view the previous results andhistory	3	Medium	ESHA A
Sprint-3	Integration	USN9	Integrate Flask, CNN model with Cloudant DB	5	Medium	KOUSIKA M

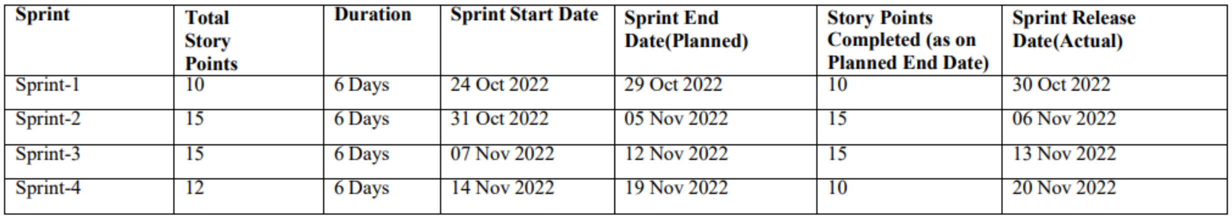


Sprint-3	Containerization	USN10	Containerize Flask app using Docker	2	Low	SHUBASHINI J



Sprint-4	Dashboard (Admin)	USN11	As an admin, I can view other user details and uploads for other purposes	2	Medium	ASIYA BEGAM M
Sprint-4	Dashboard (Shopkeeper)	USN12	As a shopkeeper, I can enter fertilizer productsand then update the details if any	2	Low	ESHA A
Sprint-4	Containerization	USN13	Create and deploy Helm charts using Docker Image made before	2	Low	KOUSIKA M

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

7 CODING & SOLUTIONING

7.1 FEATURE 1

7.1.1 DATASET

Two datasets will be used, we will be creating two models one to detect vegetable leaf diseases like tomato, potato, and pepper plants and the second model would be for fruit diseases like corn, peach, and apple.

7.1.2 IMAGE PROCESSING

Before training the model, you have to pre-process the images and then feed them onto themodel for training. We make use of the Keras ImageDataGenerator class for image pre-processing.

Image Pre-processing includes the following main tasks

- Import ImageDataGenerator Library.
- Configure ImageDataGenerator Class.
- Applying ImageDataGenerator functionality to the trainset and test set.

Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset. The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the Image Data Generator class.

7.1.3 MODEL BUILDING FOR DISEASE PREDICTION

For model building, we are following the below steps

- Import the libraries
- Initializing the model

- Add CNN layers
- Add dense layer
- Train and Save the model

7.1.4 IMPORT THE LIBRARIES

Here we have Imported the libraries that are required to initialize the neural network layer, and create and add different layers to the neural network model.

```
from keras.models import Sequential

from keras.layers import Dense

from keras.layers import Convolution2D

from keras.layers import MaxPooling2D

from keras.layers import Flatten
```

7.1.5 ADD CNN AND CONVOLUTION LAYER

We will be adding three layers for CNN

- Convolution layer
- Pooling layer
- Flattening layer

The first layer of the neural network model, the convolution layer will be added. To create a convolution layer, Convolution2D class is used. It takes a number of feature detectors, feature detector size, expected input shape of the image, and activation function as arguments. This layer applies feature detectors on the input image and returns a feature map (features from the image).

Activation Function:

These are the functions that help us to decide if we need to activate the node or not. These functions introduce non-linearity in the networks.

- **model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation ='relu'))**

7.1.6 TRAIN AND SAVE THE MODEL

After adding all the required layers, the model is compiled, for this step, the loss function, optimizer,and metrics for evaluation can be passed as arguments

- **model.compile(optimizer='adam', loss ="categorical_crossentropy" , metrics =['accuracy'])**

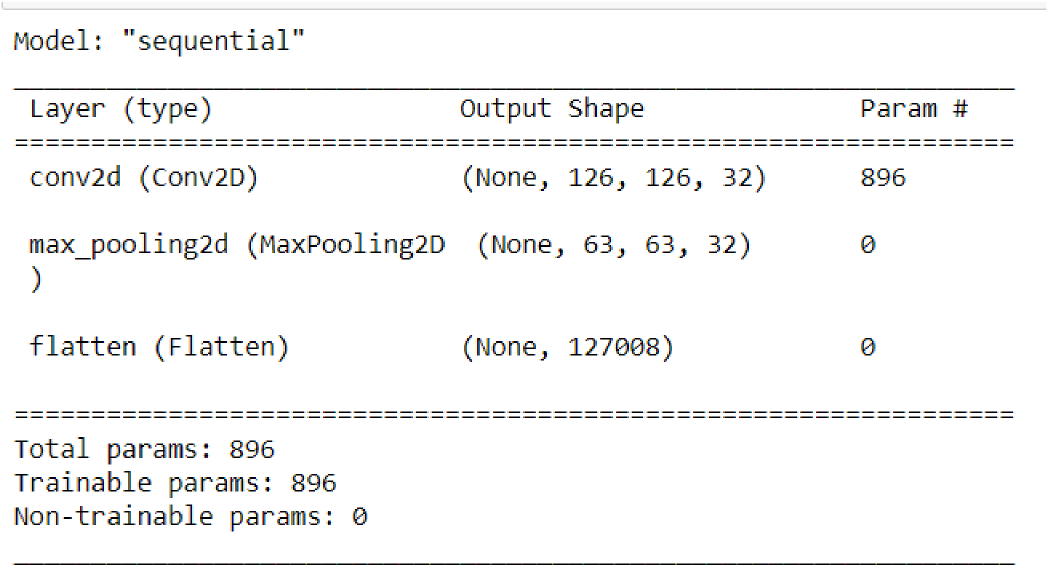
Fit the neural network model with the train and test set

- **model.fit(x_train,epochs=20,steps_per_epoch=89,validation_data= x_test, validation_steps = 27)**

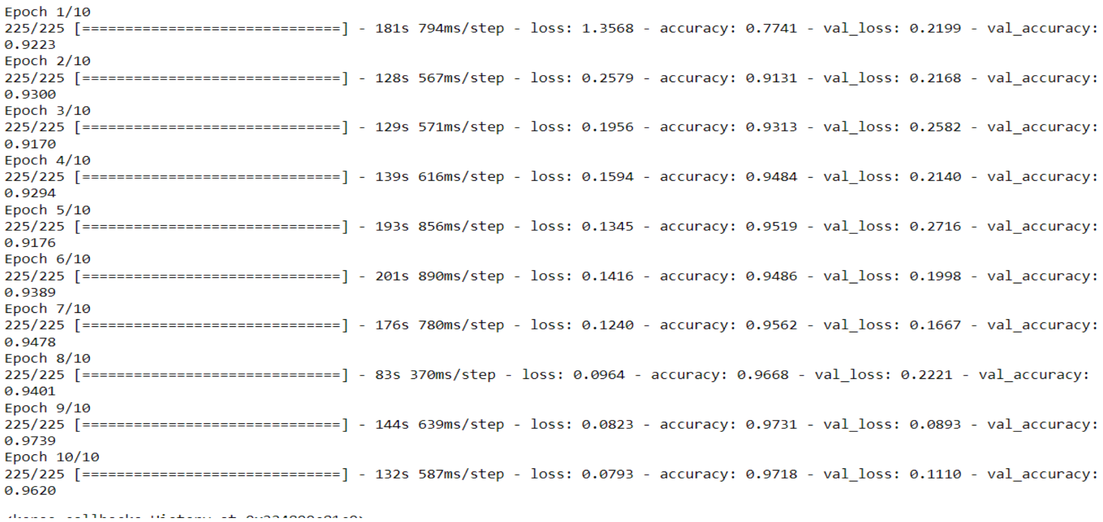
The weights are to be saved for future use. The weights are saved in as .h5 file using save().

- **model.save("fruit.h5")**

model.summary() can be used to see all parameters and shapes in each layer in our models



```
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
_________________________________________________________________
```



```
Epoch 1/10
225/225 [==============================] - 181s 794ms/step - loss: 1.3568 - accuracy: 0.7741 - val_loss: 0.2199 - val_accuracy:
0.9223
Epoch 2/10
225/225 [==============================] - 128s 567ms/step - loss: 0.2579 - accuracy: 0.9131 - val_loss: 0.2168 - val_accuracy:
0.9300
Epoch 3/10
225/225 [==============================] - 129s 571ms/step - loss: 0.1956 - accuracy: 0.9313 - val_loss: 0.2582 - val_accuracy:
0.9170
Epoch 4/10
225/225 [==============================] - 139s 616ms/step - loss: 0.1594 - accuracy: 0.9484 - val_loss: 0.2140 - val_accuracy:
0.9294
Epoch 5/10
225/225 [==============================] - 193s 856ms/step - loss: 0.1345 - accuracy: 0.9519 - val_loss: 0.2716 - val_accuracy:
0.9176
Epoch 6/10
225/225 [==============================] - 201s 890ms/step - loss: 0.1416 - accuracy: 0.9486 - val_loss: 0.1998 - val_accuracy:
0.9389
Epoch 7/10
225/225 [==============================] - 176s 780ms/step - loss: 0.1240 - accuracy: 0.9562 - val_loss: 0.1667 - val_accuracy:
0.9478
Epoch 8/10
225/225 [==============================] - 83s 370ms/step - loss: 0.0964 - accuracy: 0.9668 - val_loss: 0.2221 - val_accuracy:
0.9401
Epoch 9/10
225/225 [==============================] - 144s 639ms/step - loss: 0.0823 - accuracy: 0.9731 - val_loss: 0.0893 - val_accuracy:
0.9739
Epoch 10/10
225/225 [==============================] - 132s 587ms/step - loss: 0.0793 - accuracy: 0.9718 - val_loss: 0.1110 - val_accuracy:
0.9620

<keras.callbacks.History at 0x224800a81c0>
```

7.2 FEATURE 2

7.2.1 APPLICATION BUILDING

After the model is built, we will be integrating it into a web application so that normal users can also use it. The new users need to initially register in the portal. After registration users can login to browse the images to detect the disease.

In this section, you have to build

- HTML pages - front end
- Python script - Server-side script

7.2.2 BUILD PYTHON CODE

After the model is built, we will be integrating it into a web application so that normal users canalso use it. The user needs to browse the images to detect the disease.

Activity 1: Build a flask application

Step 1: Load the required packages

```
from_future_import division, print_functionimport os

import numpy as np

import cv2

# Keras

from tensorflow.keras.models import load_model

from tensorflow.keras.preprocessing.image import img_to_array
```

Step 2: Initializing the flask app and loading the model flask applications must create an application instance. The web server passes all the requests it receives from clients to objects for handling using a protocol for WSG.

```
        from flask  import Flask
```

app = Flask (name) (An application instance is an object of class Flask.)

```
        app = Flask( name )

        MODEL_PATH = 'fruit.h5'
```

MODEL LOADING

```
model = load_model(MODEL_PATH) model.make_predict_function()
default_image_size = (128, 128)
labels=["Apple_Black_rot","Apple_healthy","Corn_(maize)  healthy",
"Corn_(maize)_______________________Northern_Leaf_Blight","Peach
Bacterial_spot","Peach healthy"]

def convert_image_to_array(image_dir):

        try:

           image = cv2.imread(image_dir)

           if image is not None:

                   image = cv2.resize(image, default_image_size)

                   return img_to_array(image)

          else:

                   return np.array([])

        except Exception as e:
```

```
        print(f"Error : {e}")

        return None

def model_predict(file_path, model):

    x = convert_image_to_array(file_path)

    x=np.expand_dims(x, axis=0)

    preds = model.predict(x)

    return preds
```

Step 3: Configure the home page

Routes and View Functions in Flask Framework Instance

Clients send requests to the webserver, in turn, sends them to the Flask application instance. The instance needs to know what code needs to run for each URL requested and map URLs to Python functions. The association between a URL and the function that handles it is called a route. The most convenient way to define a route in a Flask application is through the (app.route). Decorator exposed by the application instance, which registers the 'decorated function,' decorators are python feature that modifies the behavior of a function.

```
 @app.route("/", methods=['GET'])

def index():
    return render_template("index.html", query="")
```

Step 4: Pre-process the frame and run

Pre-process the captured frame and given it to the model for prediction. Based on the prediction the output text is generated and sent to the HTML to display.

Request

To process incoming data in Flask, you need to use the request object, including mime-type, IPaddress, and data. HEAD: Un-encrypted data sent to server w/o response.

GET

Sends data to the server requesting a response body.

POST

Read form inputs and register a user, send HTML data to the server are methods handled by the route. Flask attaches methods to each route so that different view functions can handle different request methods to the same URL.

```
@app.route("/", methods=['GET', 'POST'])
def upload():

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

                if = request.files['file']

                basepath = os.path.dirname( file_)

                file_path=os.path.join(basepath,'uploads',secure_filename(f.filename))

                f.save(file_path) preds = model_predict(file_path, model)

                preds = np.argmax(preds)

                results= labels[preds]

        return render_template('index.html', prediction_text=result)

        return None
```

Server Startup

The application instance has a ‘run’ method that launches flask’s integrateddevelopment webserver –

```
if_name_== "_main_":

    app.run(debug=True)
```

Output:

```
* Serving Flask app 'app'

* Debug mode: on

* Running on http://127.0.0.1:5000
```

7.2.3 BUILD HTML PAGES

home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>home page</title>
  <style>
    body{
    margin: 0;
    padding: 0;
    }
    .container{
      padding: 30px 70px 30px 70px;
     left: 20px;
     right:20px;
      background-color:rgb(163, 192, 120);
      font-size: 20pt;
      font-family: 'Times New Roman';



    }

    .card{
      font: optional;
      display: flex;

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

      background-color:black;



    }
    #abc{
      color: white;
   }

  </style>
</head>
<body><div class="menu">
  <ul>
```

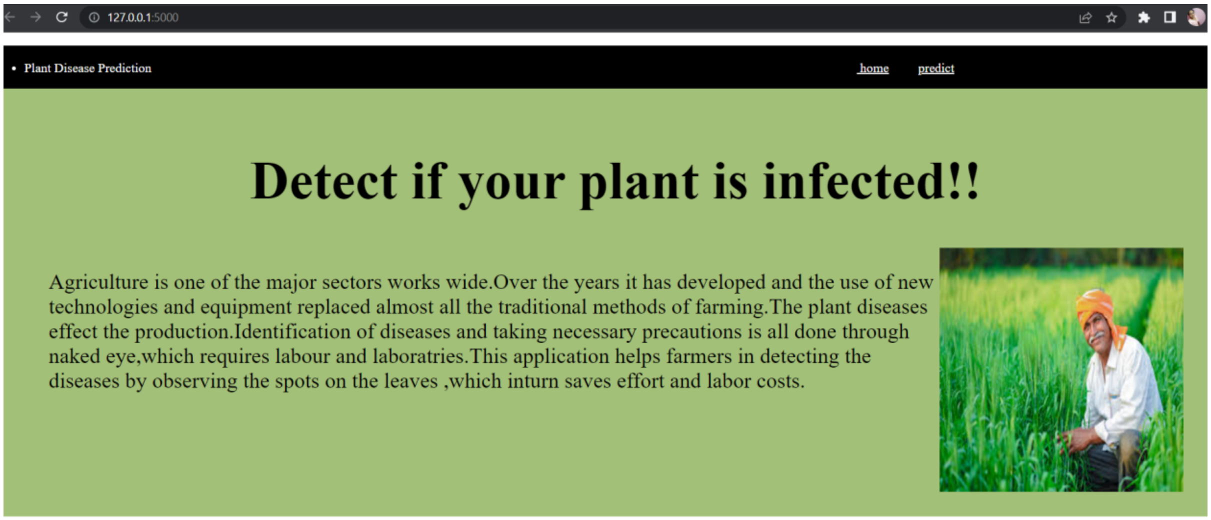
```
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;


&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;

    <li id="abc"> Plant Disease
Prediction&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb
sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href="{{
url_for('home') }}" id="abc"> home</a>

        &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href="{{
url_for('prediction') }}" id="abc">predict</a></li></ul>


    <div class="container" >
```

```
        <h1 id="h1"><center><b> Detect if your plant is infected!! </b></center></h1>
        <div class="card" >
        <p > Agriculture is one of the major sectors works wide.Over the years it has developed and the use of new technologies and equipment replaced almost all the traditional methods of farming.The plant  diseases effect theproduction. Identification of diseases and taking necessary precautions is all done through naked eye,which requires labour and laboratries.This application helps farmers in detecting the diseases by observing the spots on the leaves ,which inturn saves effort and labor costs.</p>
        <img src="{{url_for('static', filename='img.jpg')}}" height="300" width="300">
    </div>
    </div>
    </div>
</body>
</html>
```

predict.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>predict</title>
</head>

<style>
    .container{
        display: flex;
        padding: 60px 70px 60px 70px;
    }
    .card{
        padding: 70px 80px 70px 80px;
    }
    .menu{
        padding: 10px 10px 10px 10px;
        background-color: black;
        color: white;
        font-size: 15pt;
    }
</style>
<body>
    <div class="menu">
        <ul ><li>Plant disease Prediction</li></ul></div>
    <div class="container">

        <img src="{{url_for('static', filename='img1.jpg')}}">
        <div class="card">
        <form action="{{url_for('predict')}}" method="POST">
            <h1>Drop in the image to get the Prediction </h1><br><br>
            <label><select name="Fruit" id="plant">
                <option value="fruit" id="fruit">Fruit</option>
                <option value="vegitable" id="vig">vegitable</option>
                </select>
            </label><br><br><br>
            <input  id="default-btn" type="file" name=""
onchange="document.getElementById('output').src=window.URL.createObjectURL
(this.files[0])"><br><br><br>
            <img src="" id="output">
```

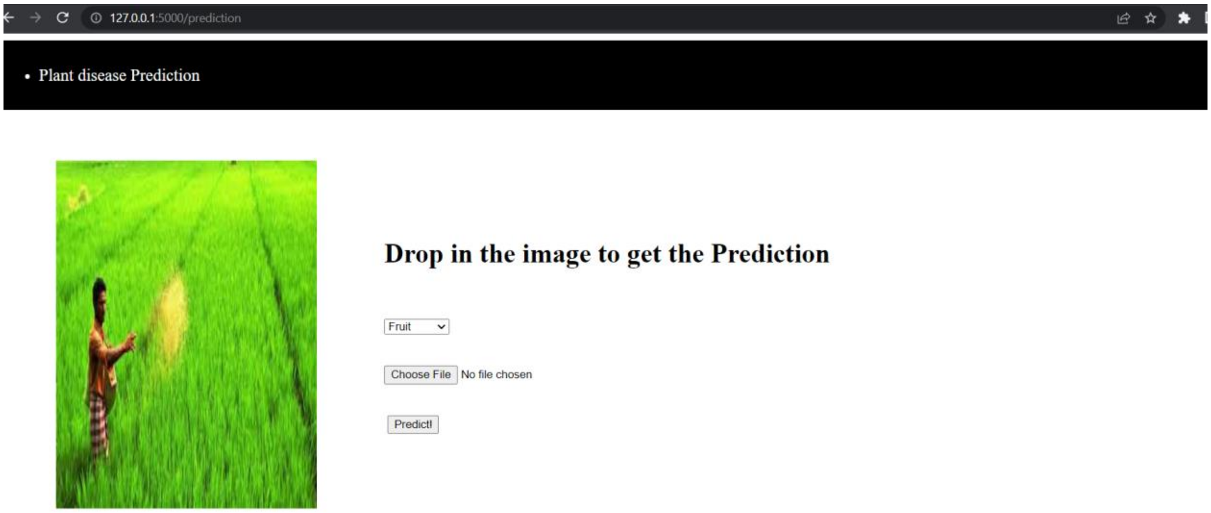
```
                <button id="button" onclick ="display()" >Predict!</button><br><br>



        </form>


</body>
</html>
```



- Plant Disease Prediction

Detect if your plant is infected!!

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

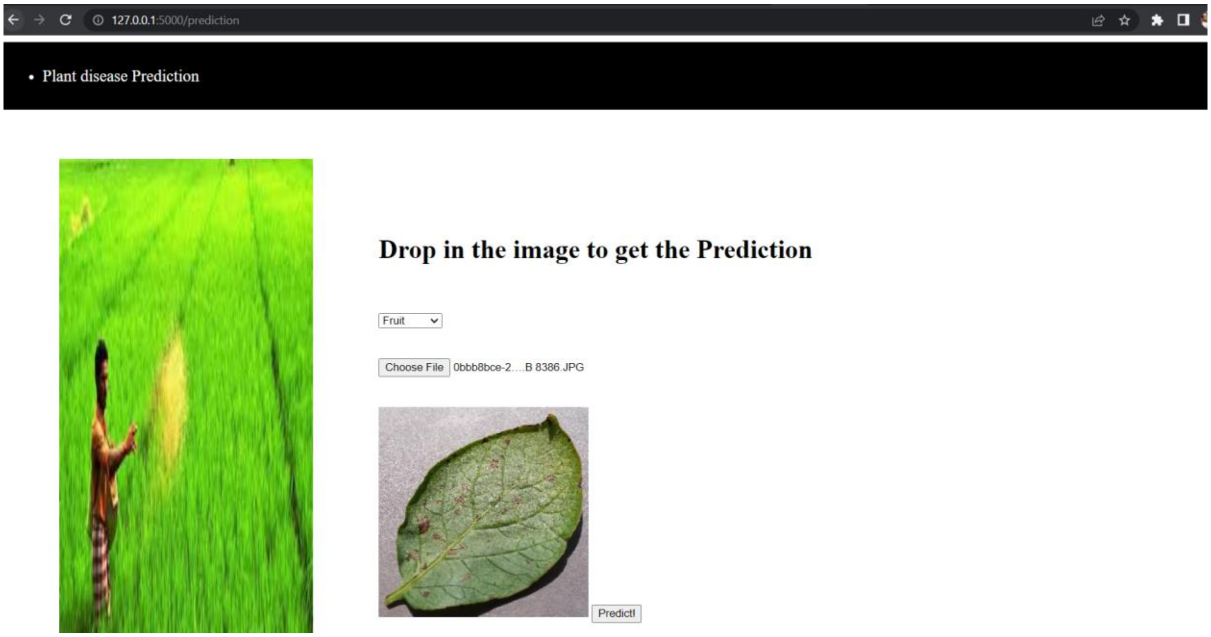


- Plant disease Prediction

Drop in the image to get the Prediction

Fruit

No file chosen



127.0.0.1:5000/prediction

- Plant disease Prediction

Drop in the image to get the Prediction

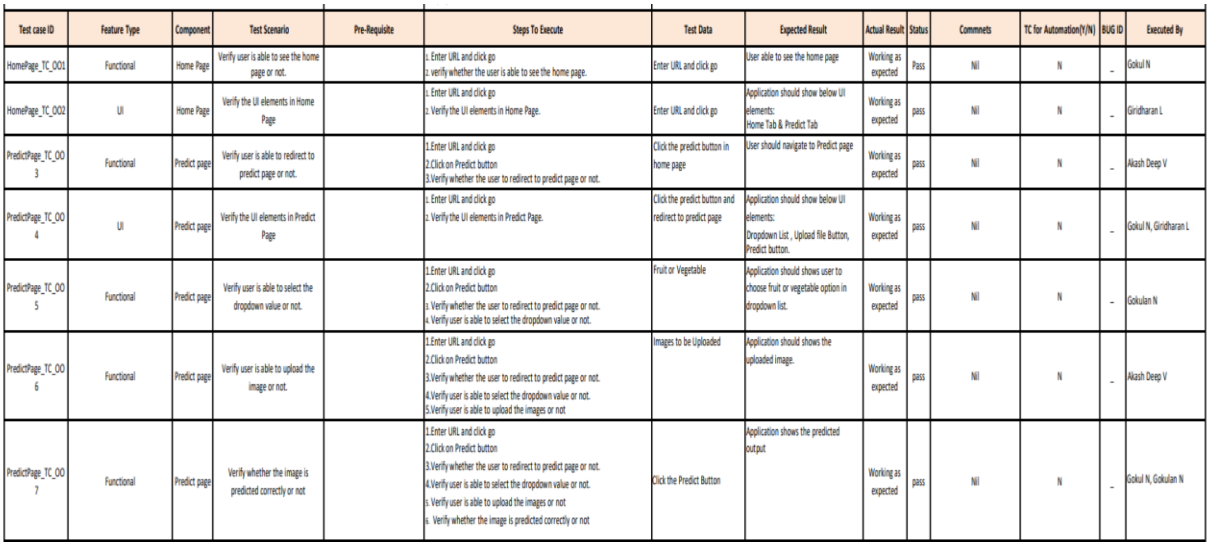
Fruit

Choose File 0bbb8bce-2....B 8386 JPG

Fig 7.1 Output image

8 TESTING

8.1 TEST CASE



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_OO1	Functional	Home Page	Verify user is able to see the home page or not.		1. Enter URL and click go 2. verify whether the user is able to see the home page.	Enter URL and click go	User able to see the home page	Working as expected	Pass	Nil	N	_	Gokul N
HomePage_TC_OO2	UI	Home Page	Verify the UI elements in Home Page		1. Enter URL and click go 2. Verify the UI elements in Home Page.	Enter URL and click go	Application should show below UI elements: Home Tab & Predict Tab	Working as expected	pass	Nil	N	_	Giridharan L
PredictPage_TC_OO3	Functional	Predict page	Verify user is able to redirect to predict page or not.		1.Enter URL and click go 2.Click on Predict button 3.Verify whether the user to redirect to predict page or not.	Click the predict button in home page	User should navigate to Predict page	Working as expected	pass	Nil	N	_	Akash Deep V
PredictPage_TC_OO4	UI	Predict page	Verify the UI elements in Predict Page		1. Enter URL and click go 2. Verify the UI elements in Predict Page.	Click the predict button and redirect to predict page	Application should show below UI elements: Dropdown List , Upload file Button, Predict button.	Working as expected	pass	Nil	N	_	Gokul N, Giridharan L
PredictPage_TC_OO5	Functional	Predict page	Verify user is able to select the dropdown value or not.		1.Enter URL and click go 2.Click on Predict button 3. Verify whether the user to redirect to predict page or not. 4. Verify user is able to select the dropdown value or not.	Fruit or Vegetable	Application should shows user to choose fruit or vegetable option in dropdown list.	Working as expected	pass	Nil	N	_	Gokulan N
PredictPage_TC_OO6	Functional	Predict page	Verify user is able to upload the image or not.		1.Enter URL and click go 2.Click on Predict button 3.Verify whether the user to redirect to predict page or not. 4.Verify user is able to select the dropdown value or not. 5.Verify user is able to upload the images or not	Images to be Uploaded	Application should shows the uploaded image.	Working as expected	pass	Nil	N	_	Akash Deep V
PredictPage_TC_OO7	Functional	Predict page	Verify whether the image is predicted correctly or not		1.Enter URL and click go 2.Click on Predict button 3.Verify whether the user to redirect to predict page or not. 4.Verify user is able to select the dropdown value or not. 5. Verify user is able to upload the images or not 6. Verify whether the image is predicted correctly or not	Click the Predict Button	Application shows the predicted output	Working as expected	pass	Nil	N	_	Gokul N, Gokulan N

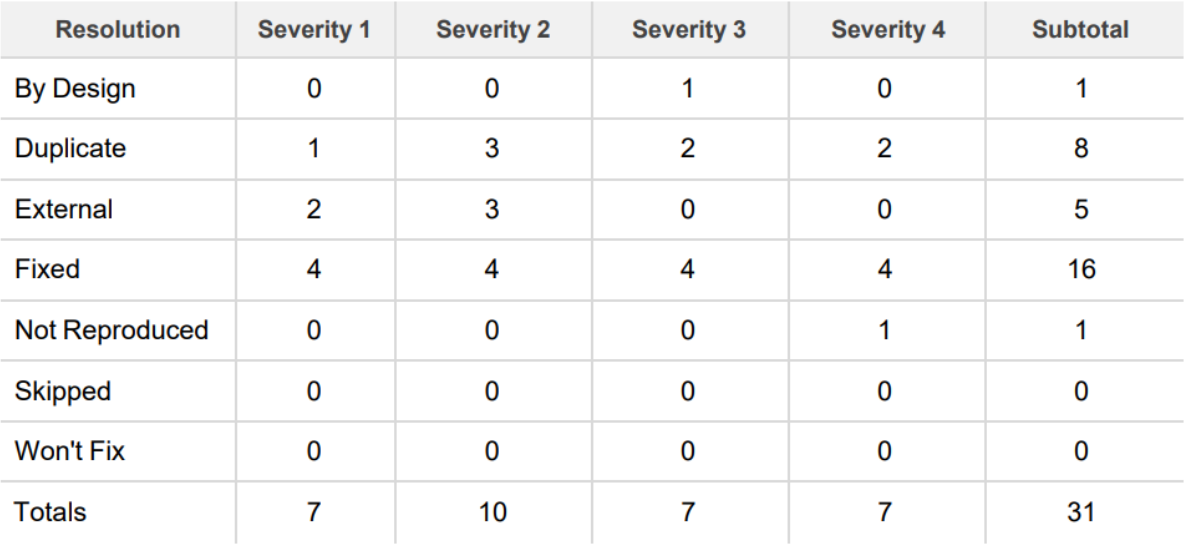
8.2 USER ACCEPTANCE TESTING

Purpose of Document

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

Defect Analysis

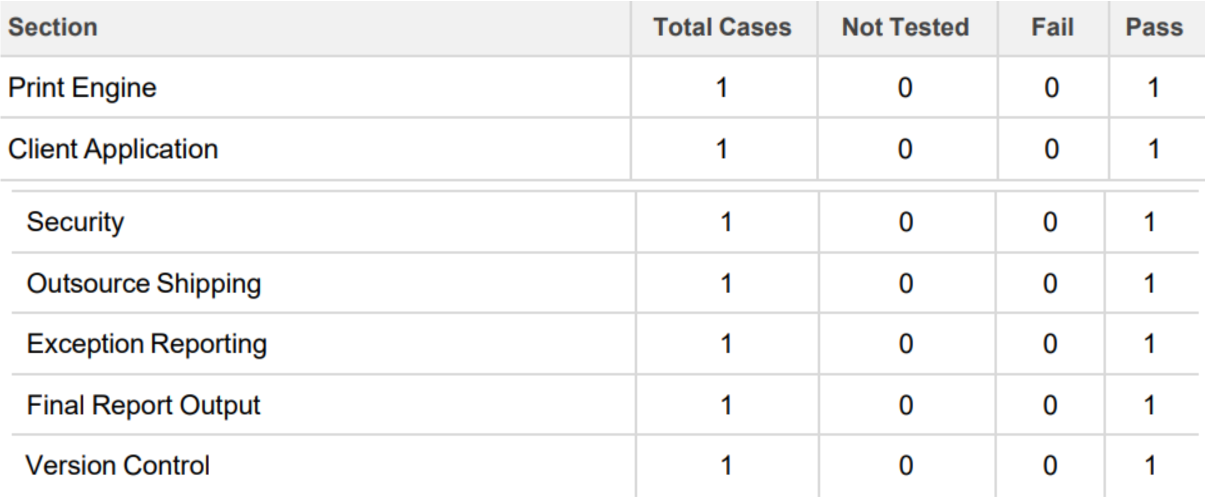
This report shows the number of resolved or closed bugs at each severity level, and how they were resolved



Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	0	0	1	0	1
Duplicate	1	3	2	2	8
External	2	3	0	0	5
Fixed	4	4	4	4	16
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	7	10	7	7	31

Test Case Analysis

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

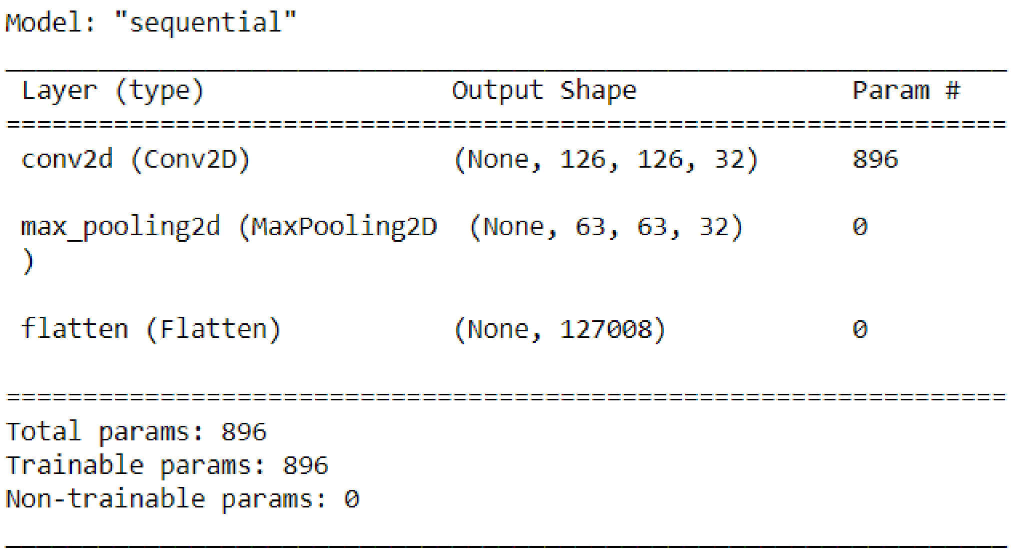


Section	Total Cases	Not Tested	Fail	Pass
Print Engine	1	0	0	1
Client Application	1	0	0	1
Security	1	0	0	1
Outsource Shipping	1	0	0	1
Exception Reporting	1	0	0	1
Final Report Output	1	0	0	1
Version Control	1	0	0	1

9 RESULT

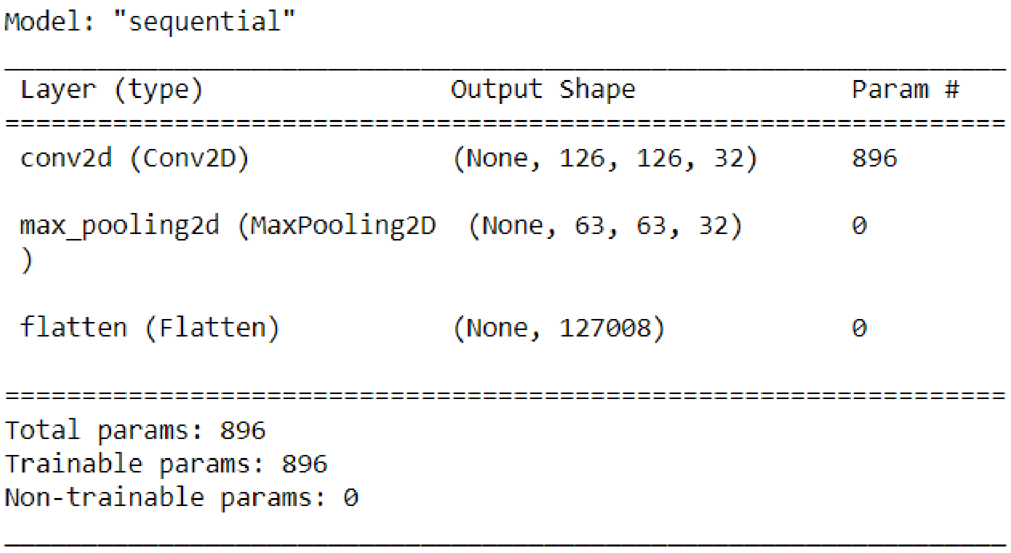
9.1 PERFORMANCE METRICS

VEGETABLE



```
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
_________________________________________________________________
```

FRUIT

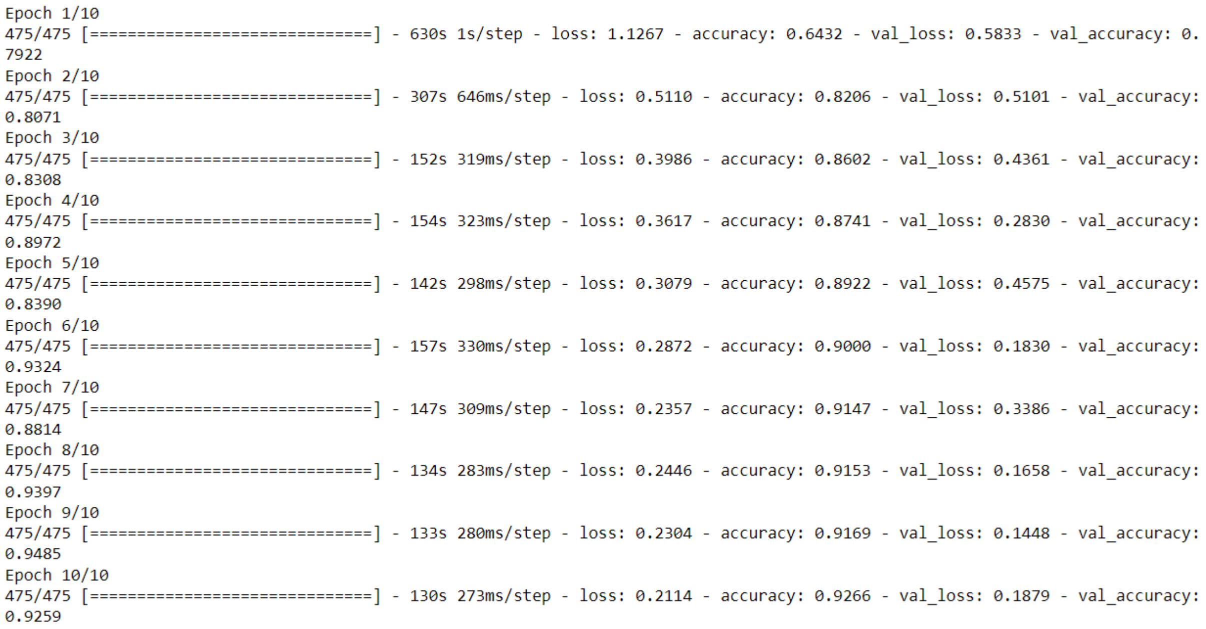


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 126, 126, 32)      896       
                                                                 
 max_pooling2d (MaxPooling2D  (None, 63, 63, 32)       0         
 )                                                               
                                                                 
 flatten (Flatten)           (None, 127008)            0         
                                                                 
=================================================================
Total params: 896
Trainable params: 896
Non-trainable params: 0
_________________________________________________________________
```

PARAMETER ACCURACY

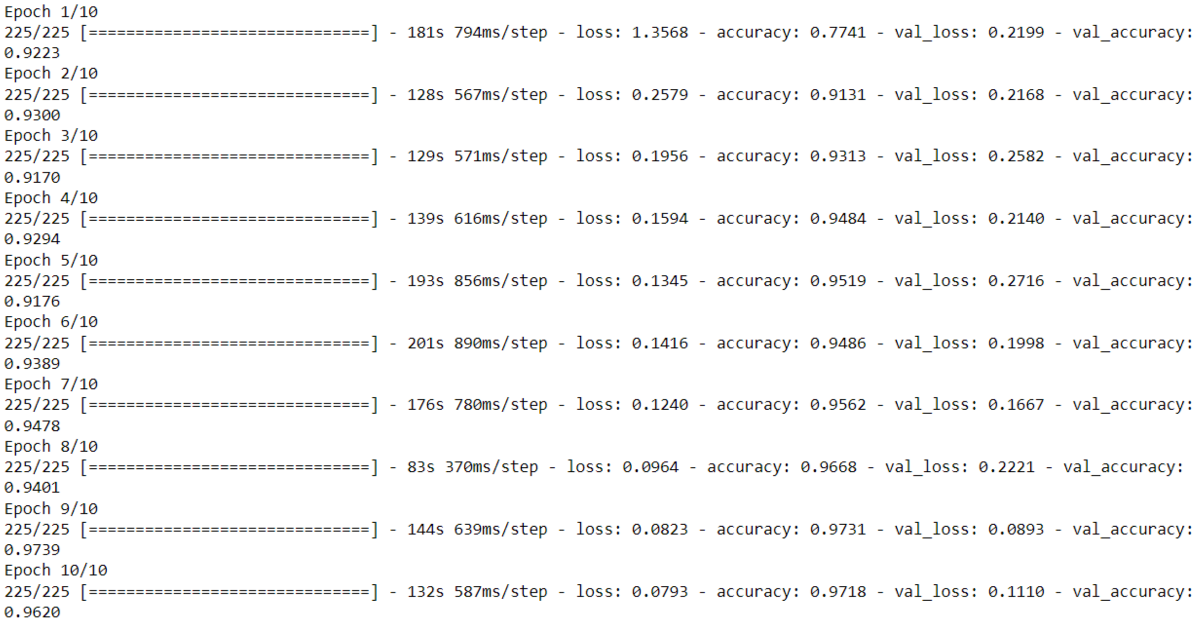
- Training Accuracy
- Validation Accuracy

VEGETABLE



```
Epoch 1/10
475/475 [==============================] - 630s 1s/step - loss: 1.1267 - accuracy: 0.6432 - val_loss: 0.5833 - val_accuracy: 0.
7922
Epoch 2/10
475/475 [==============================] - 307s 646ms/step - loss: 0.5110 - accuracy: 0.8206 - val_loss: 0.5101 - val_accuracy:
0.8071
Epoch 3/10
475/475 [==============================] - 152s 319ms/step - loss: 0.3986 - accuracy: 0.8602 - val_loss: 0.4361 - val_accuracy:
0.8308
Epoch 4/10
475/475 [==============================] - 154s 323ms/step - loss: 0.3617 - accuracy: 0.8741 - val_loss: 0.2830 - val_accuracy:
0.8972
Epoch 5/10
475/475 [==============================] - 142s 298ms/step - loss: 0.3079 - accuracy: 0.8922 - val_loss: 0.4575 - val_accuracy:
0.8390
Epoch 6/10
475/475 [==============================] - 157s 330ms/step - loss: 0.2872 - accuracy: 0.9000 - val_loss: 0.1830 - val_accuracy:
0.9324
Epoch 7/10
475/475 [==============================] - 147s 309ms/step - loss: 0.2357 - accuracy: 0.9147 - val_loss: 0.3386 - val_accuracy:
0.8814
Epoch 8/10
475/475 [==============================] - 134s 283ms/step - loss: 0.2446 - accuracy: 0.9153 - val_loss: 0.1658 - val_accuracy:
0.9397
Epoch 9/10
475/475 [==============================] - 133s 280ms/step - loss: 0.2304 - accuracy: 0.9169 - val_loss: 0.1448 - val_accuracy:
0.9485
Epoch 10/10
475/475 [==============================] - 130s 273ms/step - loss: 0.2114 - accuracy: 0.9266 - val_loss: 0.1879 - val_accuracy:
0.9259
```

FRUIT



```
Epoch 1/10
225/225 [==============================] - 181s 794ms/step - loss: 1.3568 - accuracy: 0.7741 - val_loss: 0.2199 - val_accuracy:
0.9223
Epoch 2/10
225/225 [==============================] - 128s 567ms/step - loss: 0.2579 - accuracy: 0.9131 - val_loss: 0.2168 - val_accuracy:
0.9300
Epoch 3/10
225/225 [==============================] - 129s 571ms/step - loss: 0.1956 - accuracy: 0.9313 - val_loss: 0.2582 - val_accuracy:
0.9170
Epoch 4/10
225/225 [==============================] - 139s 616ms/step - loss: 0.1594 - accuracy: 0.9484 - val_loss: 0.2140 - val_accuracy:
0.9294
Epoch 5/10
225/225 [==============================] - 193s 856ms/step - loss: 0.1345 - accuracy: 0.9519 - val_loss: 0.2716 - val_accuracy:
0.9176
Epoch 6/10
225/225 [==============================] - 201s 890ms/step - loss: 0.1416 - accuracy: 0.9486 - val_loss: 0.1998 - val_accuracy:
0.9389
Epoch 7/10
225/225 [==============================] - 176s 780ms/step - loss: 0.1240 - accuracy: 0.9562 - val_loss: 0.1667 - val_accuracy:
0.9478
Epoch 8/10
225/225 [==============================] - 83s 370ms/step - loss: 0.0964 - accuracy: 0.9668 - val_loss: 0.2221 - val_accuracy:
0.9401
Epoch 9/10
225/225 [==============================] - 144s 639ms/step - loss: 0.0823 - accuracy: 0.9731 - val_loss: 0.0893 - val_accuracy:
0.9739
Epoch 10/10
225/225 [==============================] - 132s 587ms/step - loss: 0.0793 - accuracy: 0.9718 - val_loss: 0.1110 - val_accuracy:
0.9620
```

10 ADVANTAGES & DISADVANTAGES

10.1 ADVANTAGES

Farmers can interact with the portal build

- Interacts with the user interface to upload images of diseased leafOur model-built analyses the Disease and suggests the farmer with fertilizers are to be used
- It is easy to maintain.
- It is user-friendly.
- The system can easily detect the leaf from the image.
- It will also detect which type of leaf it is.
- It will suggest the recommended fertilizer for that disease quickly with in a minute of time.

10.2 DISADVANTAGES

- More training samples - more speed of computing distances sensitive irrelevant inputsso expensive test every time
- It is slower in execution speed and long training time.
- Sometimes it can predict the wrong disease which may cause difficulty for farmers.
- Recommending the wrong fertilizers can damage crops.
- It requires more samples to prepare the application and if any wrong updates that make crop damage.
- Previously yield is predicted on the bases of the farmers prior experience but now weather conditions may change drastically so they cannot guess the yield.

11 CONCLUSION

We have proposed an automated system to identify and classify the disease caused in plants at an earlier stage with pest management. to detect and identification of various diseases, we usethe convolutional neural network (CNN) and deep learning. The result from can be used to identify the disease with a highly accurate and suggested solution. A high-performance model is obtained by using the best

hyperparameters and good training data. The final model will give high accuracy for the given data. An application to detect, control, and monitor plant disease helps the farmer to reduce their work as well as time. This application helps the farmer to reduce their effort, and also helps in increasing the farm of production. The proposed method helps to find the plant disease and in monitoring the several environmental conditions the status of the leaf has been identified with the help of neural network classification. Then the environmental circumstances such as temperature, humidity, and moisture have been monitored the environmental condition is abnormal, then the pump will automatically. This project gives the executed results on different disease classification techniques that can be used for plant leaf disease detection a. Therefore, related diseases for these plants were taken for identification.With very less computational effort the optimum results were obtained, which also shows the efficiency of the proposed algorithm in the recognition and classification of the leaf diseases. Another advantage of using this method is that plant diseases can be identified at an early stage or the initial stage. By using this concept, disease identification is done for all kinds of leaves and also the user can know the affected area of the leaf in percentage by identifying the disease properly the user can rectify the problem very easily.

12 FUTURE SCOPE

- This system can be enhanced in the future by using the trained model in android apps tomake it more feasible and efficient.
- In the future, the use of more advanced algorithms can be implemented into the system to showhigh accuracy and less process time.
- Using the camera we can implement the system in continuous monitoring of crops and plants for detecting the texture of plants for more early detection of plants.
- After the leaf undergoes detection, the disease is identified, and checked whether the leaf can be cured under certain conditions or not, and fertilizers are recommended according to the leaf.

13 APPENDIX

13.1 GITHUB AND PROJECT DEMO LINK

GITHUB LINK: https://github.com/IBM-EPBL/IBM-Project-40045-1660621951

PROJECT DEMO LINK:
https://drive.google.com/file/d/1fxs7ptI6zh7NTbCOZARKZ7AmYKjnprrY/view?usp=s