TABLE OF CONTENTS

CHAPTER NO.	LIST OF TITLE	PAGE NO.
1	**INTRODUCTION**	6
	1.1 Project Overview	6
	1.2 Purpose	7
2	**LITERATURE SURVEY**	8
	2.1 Existing problem	8
	2.2 References	9
	2.3 Problem Statement Definition	13
3	**IDEATION & PROPOSED SOLUTION**	14
	3.1 Empathy Map Canvas	14
	3.2 Ideation & Brainstorming	15
	3.3 Proposed Solution	18
	3.4 Problem Solution fit	19
4	**REQUIREMENTS ANALYSIS**	20
	4.1 Functional requirement	20
	4.2 Non- Functional requirement	20
5	**PROJECT DESIGN**	21
	5.1 Data Flow Diagrams	21
	5.2 Solution & Technical Architecture	22

	5.3 User Stories	24
6	**PROJECT PLANNING & SCHEDULING**	25
	6.1 Sprint Planning & Estimation	25
	6.2 Reports from JIRA	28
	6.3 Velocity Report	28
7	**CODING & SOLUTIONING**	29
	7.1 Feature 1	29
	7.2 Feature 2	29
8	**TESTING**	31
	8.1 Test Cases	31
	8.2 User Acceptance Testing	33
9	**RESULTS**	34
10	**ADVANTAGES & DISADVANTANGES**	35
11	**CONCLUSION**	36
12	**FUTURE SCOPE**	37
13	**APPENDIX**	38
	13.1 Source Code	38
	13.2 Screenshots	40
14	**REFERENCES**	41

S NO.	LIST OF FIGURES	PAGE NO.
1	3.1 Empathy Map Canvas	14
2	3.2.1 Ideation	15
3	3.2.2 Brainstorming	16
4	3.2.3 Idea Prioritization	17
5	3.4 Problem Solution Fit	19
6	5.1 Data Flow Diagram	21
7	5.2 Solution & Technical Architecture	22
8	6.1.1 Sprint 1	25
9	6.1.2 Sprint 2	25
10	6.1.3 Sprint 3	26
11	6.1.4 Sprint 4	27
12	7.2 Frontend	29
13	6.2 Reports from JIRA	28
14	6.3 Velocity Report	28
15	8.1.1 Test Cases	31
16	8.1.2 Test Cases	31
17	8.1.3 Fruit Output	32
18	8.1.4 Vegetable Output	32
19	9.1 Fruit	34
20	9.2 Vegetable	34
21	13.2 Screenshots	40

S NO.	LIST OF TABLES	PAGE NO.
1	3.3 Proposed Solution	18
2	4.1 Functional requirements	20
3	4.2 Non - Functional requirements	20
4	5.2.1 Components & Technologies	22
5	5.2.2 Application Characteristics	23
6	5.3 User Stories	24
7	6.1.1 Sprint Delivery Schedule	27

S.NO.	LIST OF ABBREVIATION
1	**CNN** - Convolutional Neural Network
2	**SVM** - Support Vector Machine
3	**ML** - Machine Learning
4	**HSD** - Hyper Spectral Data
5	**ANN** - Artificial Neural Network

1. INTRODUCTION

1.1 Project Overview

Detection and recognization of plant diseases using deep learning are very efficient in providing symptoms of identifying diseases at its earliest. Plant pathologists can analyze the digital images using digital image processing for diagnosis of plant diseases.

Application of computer vision and image processing strategies simply assist farmers in all of the regions of agriculture. Generally, the plant diseases are caused by the abnormal physiological functionalities of plants. Therefore, the characteristic symptoms are generated based on the
differentiation between normal physiological functionalities and abnormal physiological functionalities of the plants. Mostly, the plant leaf diseases are caused by Pathogens which are positioned on the stems of the plants.

These different symptoms and diseases of leaves are predicted by different methods in image processing. These different methods include different fundamental processes like image preprocessing, feature model building and testing the model, the prediction and diagnosis of leaf diseases are depending on the segmentation such as image preprocessing, model building, training and testing the model.

A new approach is used for classifying and recognizing the health state of the plantation and immediately generating treatment solutions on the go. There are two contributions in this paper; convolution Neural Network (CNN) is used to classify and recognize different classes of the plant images, detect plant diseases and determine the rate of growth of plantations at the same time respectively by extracting all features during the analysis and these features are automatically learned to facilitate the development of systems that learn from one end to the other. Also, the results are visualized to enable generation of treatment plans for immediate application on the plantations and to also provide insight on how the models perceive the leaves on the images.

The novelty of this approach is in its simplicity and ability to detect different features during analysis accurately and efficiently both the healthy leaves and its background images are aligned with other classes which effectively enables the model to distinguish healthy from unhealthy plants while mapping the problem areas on the farm.

1.2 Purpose

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

2. LITERATURE SURVEY

2.1 Existing problem

The current approach for detecting plant disease is simple naked eye observation by plant experts, which can be used to detect and identify plant diseases.

In these circumstances, the suggested technique is useful for tracking vast fields of crops. Furthermore, in some nations, farmers lack adequate facilities or are unaware that they can contact experts. As a result, consulting experts is not only more expensive but also more time consuming. In those circumstances, the suggested technique for tracking a large number of plants would be useful.

The use of an automated disease detection technique is advantageous in detecting a plant disease at an early stage. Plant diseases manifest themselves in various parts of the plant, such as the leaves. It takes a long time to manually diagnose plant disease using leaf photographs.

As a result, computational methods must be developed to automate the process of disease detection and classification using leaf images.

2.2 References

Paper 1: Analysis on Prediction of Plant Leaf diseases using Deep Learning
Author : S. Nandhini
Published Year : 2021

The economy of the country and life of the human beings are depends upon Agriculture. Improper maintenance and lack of awareness may leads to the loss of productivity and quantity of the agricultural products.

The problems like infections on the plants and diseases caused to the plants may have the solutions that can be provided by the agro-scientists.

Deep learning architectures are used to solve these kind of problems and give a solution for the plants by recommending a fertilizers of the affected plants.

As the global population continues to burgeon, increasing overall crop production is becoming imperative to ensure food safety for everyone. However, a myriad of plant diseases can sever the supply of essential crops. To tackle these diseases, it is first important to identify these diseases and monitor them on a large scale.

To solve this problem, Tensorflow with Keras and OpenCV are used to build a detection system. Since automated farms tend to be large and spread out, the classifier has to be executed and controlled from a cloud computing environment.

By using this system, we can achieve a 93% accuracy and it is also effective method for protecting a agriculture.

Paper 2 : A Machine Learning Technique for Identification of Plant Diseases in Leaves
Author : Ms. Deepa , Ms. Rashmi N
Published Year : 2021

The plant disease is main reason to reduce the income to the farmers. Currently scientist to research to find the best mechanism to the find the diseases in plant.This paper attempts to approach machine learning techniques.

The first step is collect the data set into the villages.Then preprocess the images and extract the images in standard deviation using matlab commands.

Dataset is used for training is taken from internet.Then the selected images to be clustered and measure the distance the diseases in leaves. Finally classifies the images in SVM(Support Vector Machine).It shows the disease actual name perfectly.

Increasingly, HSI is used along with Machine Learning for detecting plant diseases. These methods can be used in a variety of agriculture related applications.

Equally importantly, ML has been used with HSI in training three very effective models for successfully detecting leaf rust disease.

This method when used in combination with multispectral imagine was able to achieve the fairly high accuracy of 89.3%. In some cases, SVMs that were based on "hyper-spectral data" were able to increase their accuracy to 86%.

Paper 3 : Deep Learning model for early prediction of plant disease
Author : Ms. Rubini PE
Published Year : 2020

The growth of the country's economy is based on the agriculture. The farmers are facing many issues like change in weather, climatic changes, plants affected by the disease.

And also there is solution like weather forecasting, disease prediction, phenology identification for those problems.Monitoring the plants for identifying the disease on the daily manner is not an easy task, so there is alternate solutions for disease detection of the plants.

Deep learning mechanism is used to identify whether the plant is healthy or unhealthy and what kind of disease is caused and also its recommends a solution for that disease. The samples of the leaves are collected and the images are pre-processed.

The main intent of the proposed work is to recognize and identify whether a leaf is diseased or healthy and mention the type of disease to the farmer.

In CNN, the neural network is very deep, even though there is a statement saying “deeper network provides greater accuracy”, but it is very hard to train the model due to vanishing gradient.

Trained models are tested on the validation set using GPU. There are 14000 training images and 3000 validation images. All of them are labeled. The deep learning model was able to classify with an accuracy of 95-97 percent.

The accuracy can be increased when trained with a vast number of images and by adopting pre-trained CNN models.

Paper 4 : Coconut Disease Prediction System Using Image Processing and Deep Learning Techniques
Author : Ms. Dhapitha Nesarajan, Ms. Lokini Kunalan
Published Year : 2020

Crop production is most important economy of a every country.Our main intension is enhance the farming method into new modern technology.The continuous plant monitor using AI to avoid crop production down.In recent times crop production is heavily down.

Because the plant diseases are increased. We develop a mobile application to suggest suitable fertilizers to the predicted disease.In this project first we have collect the input images.

Then images are to be extracted like RGB to grayscale. The extracted images to be segmented. After extraction images sent it to the SVM classifier.It classifies the images and suggest actual fertilizers and using open CV for continuous plan monitor and finally app shows the message.

Insects and pest attacks are major threats to the overall growth of crops. Pest attacks are rapidly raising crops like coconuts.

Coconut caterpillar and black beetle are some of the major pests and bagworm, coconut scale and leaf miners are the minor pests which attack the leaves of the coconut. One method to protect the crop is an early pest detection and suitable pesticides should be sprayed at an early stage. This protects the crop from severe pest attacks to increase the coconut cultivation and get more cost-effective benefits.

SVM and CNN algorithms are selected with the accuracy of 93.54% and 93.72%, which used to identify the pest diseases and nutrient deficiency respectively.

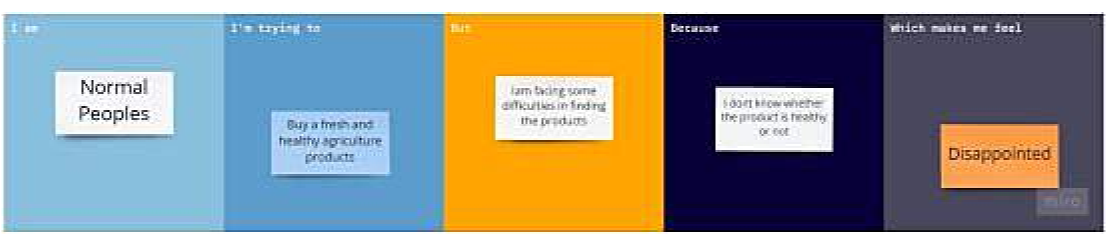
2.3 Problem Statement Definition

PROBLEM-1:

PROBLEM-2:

PROBLEM-3:

PROBLEM-4:



3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

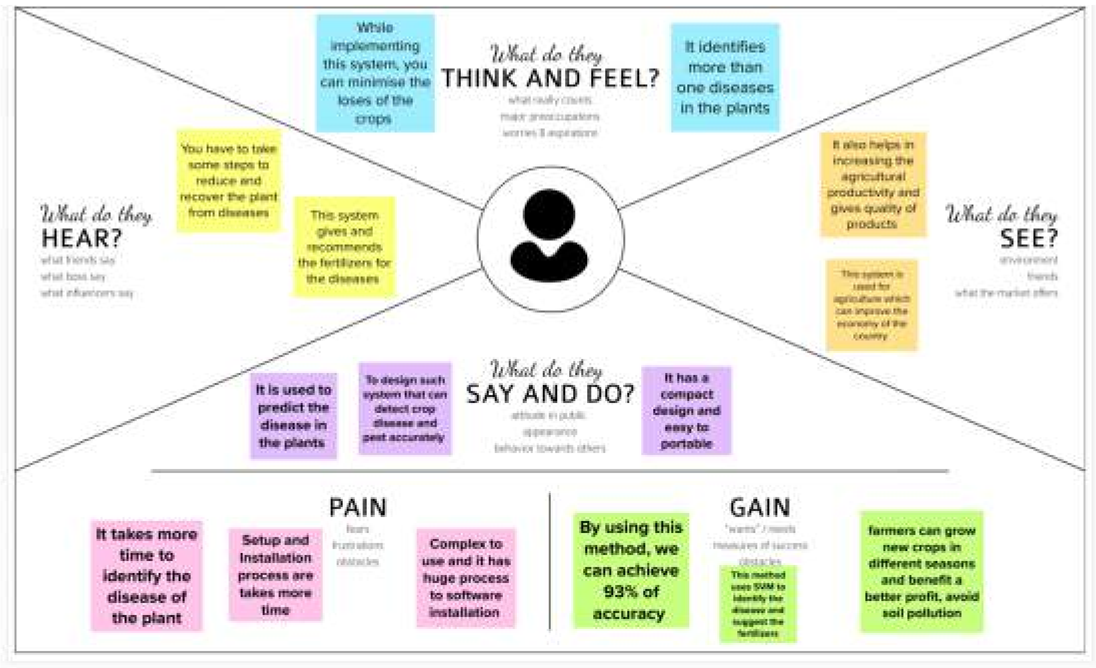


Figure.3.1 Empathy Map Canvas

Empathy map explains the how the implementation of the system is occurs and how effectively it identifies the plant disease.

It explains the steps and procedures of the process and also about the benefits ans limitations of the system.

3.2 Ideation & Brainstorming

Ideation

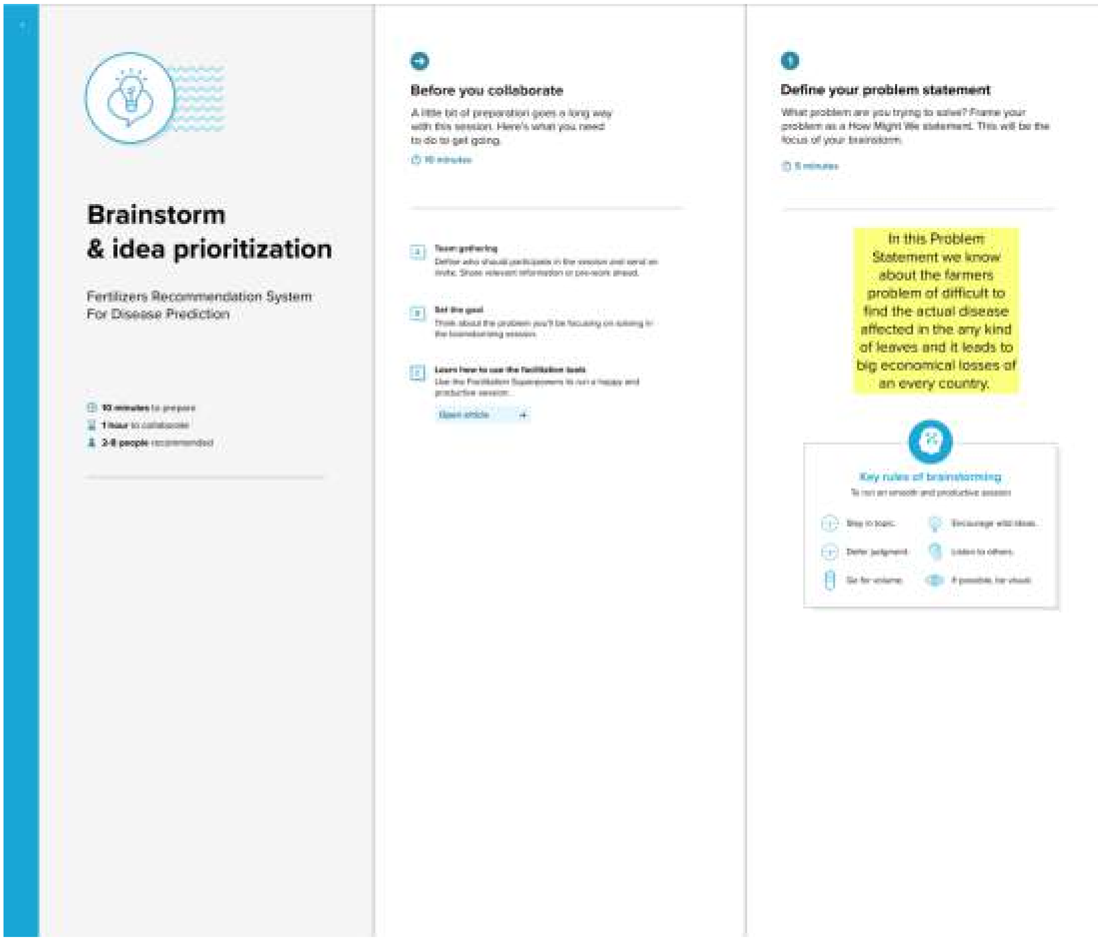


Figure.3.2.1 Ideation

Brainstorming

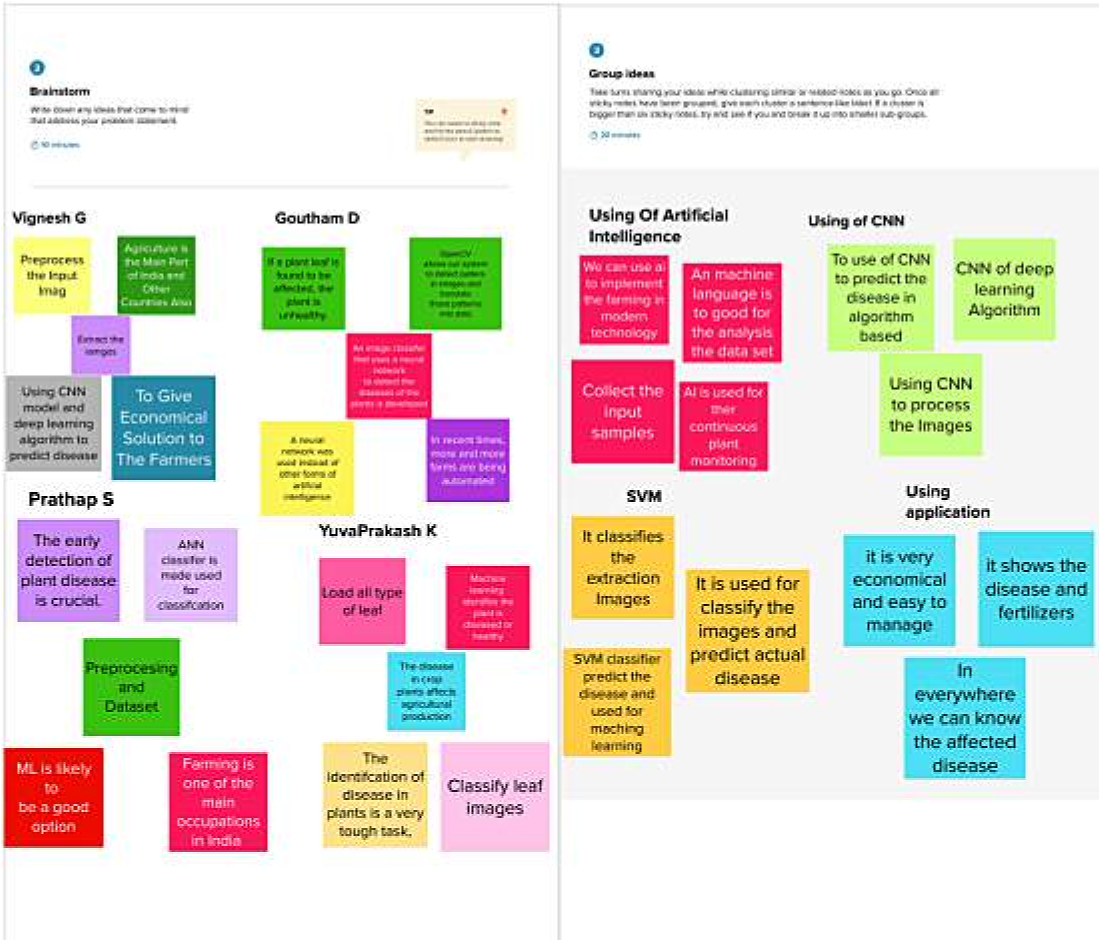


Figure.3.2.2 Brainstroming

Ideation and Brainstorm explains the step by step procedures of the system and overview of the entire process.

Idea Prioritization

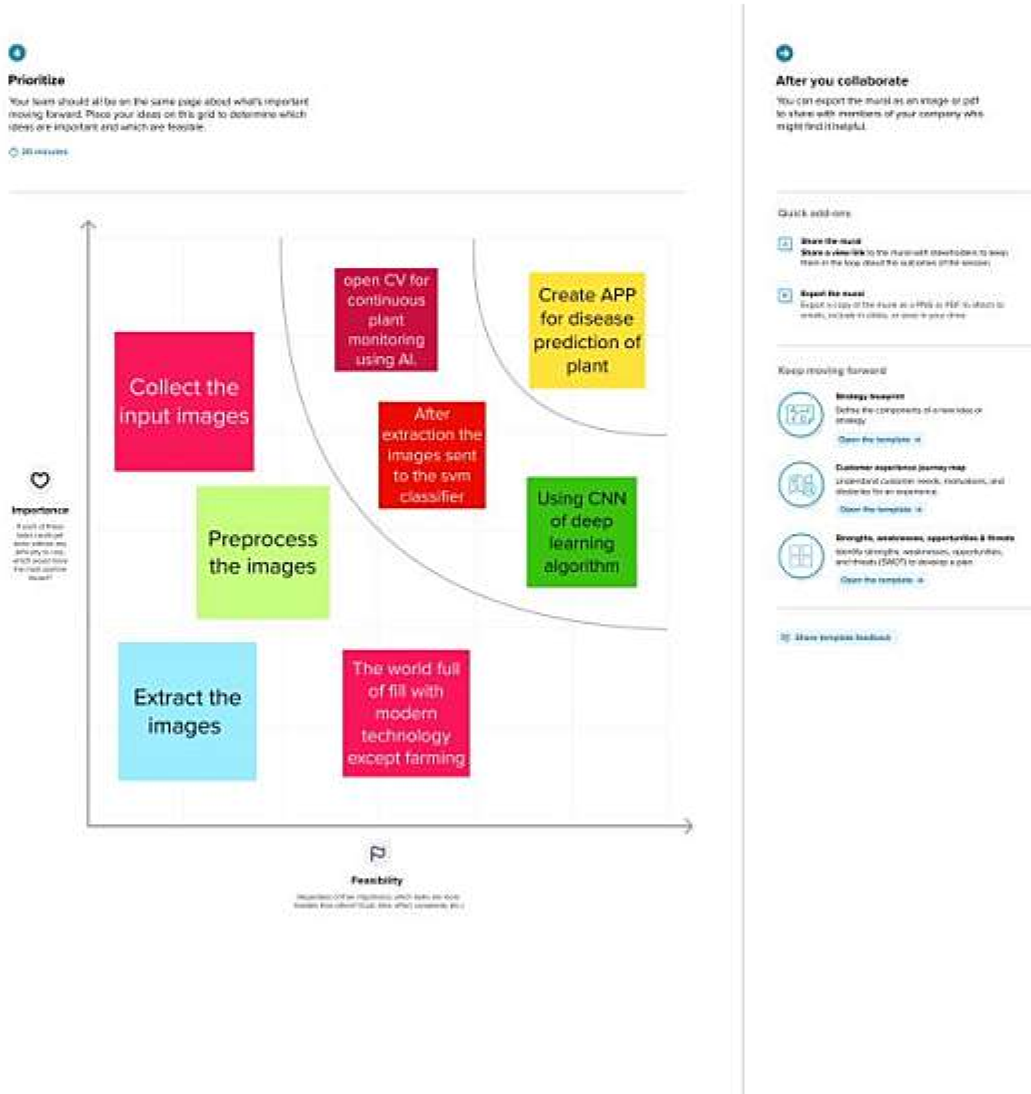


Figure.3.2.3 Idea Prioritization

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	In this project many to be solved such as identification of disease, Predict the fertilizers.The economic losses to be solved after the prediction.Increase the crop production.Hereafter the good quality of products will be received by an customer.
2.	Idea / Solution description	To avoid the people predict disease and put the wrong fertilizers,it will leads to crop decreases.So we can use application it will suggest the perfect fertilizer for the predicted disease.
3.	Novelty / Uniqueness	We use an application to predict the disease and suggest the fertilizers after the image captured.
4.	Social Impact / Customer Satisfaction	-Poor people will benefited.Because they were no money to have predict disease and implementaion. -Its is common application for all farmers can easily accessible. -It will create a big impact to the fake fertilizers marketization.Because farmers have to find the actual fertilizers . -Recent times the human disease will spread via foods.It will increase the people food safety.Because the fruits and vegetables will be pure & disease free.
5.	Business Model (Revenue Model)	To use this method increase the revenue to farmers.Because the crop production will increased.
6.	Scalability of the Solution	-Easy to managable for all people.While it shows the fertilizers in regional languages. -Huge losses decreased. -The farmers can predict everywhere and anytime.

Table.3.3 Proposed Solution

3.4 Problem Solution fit

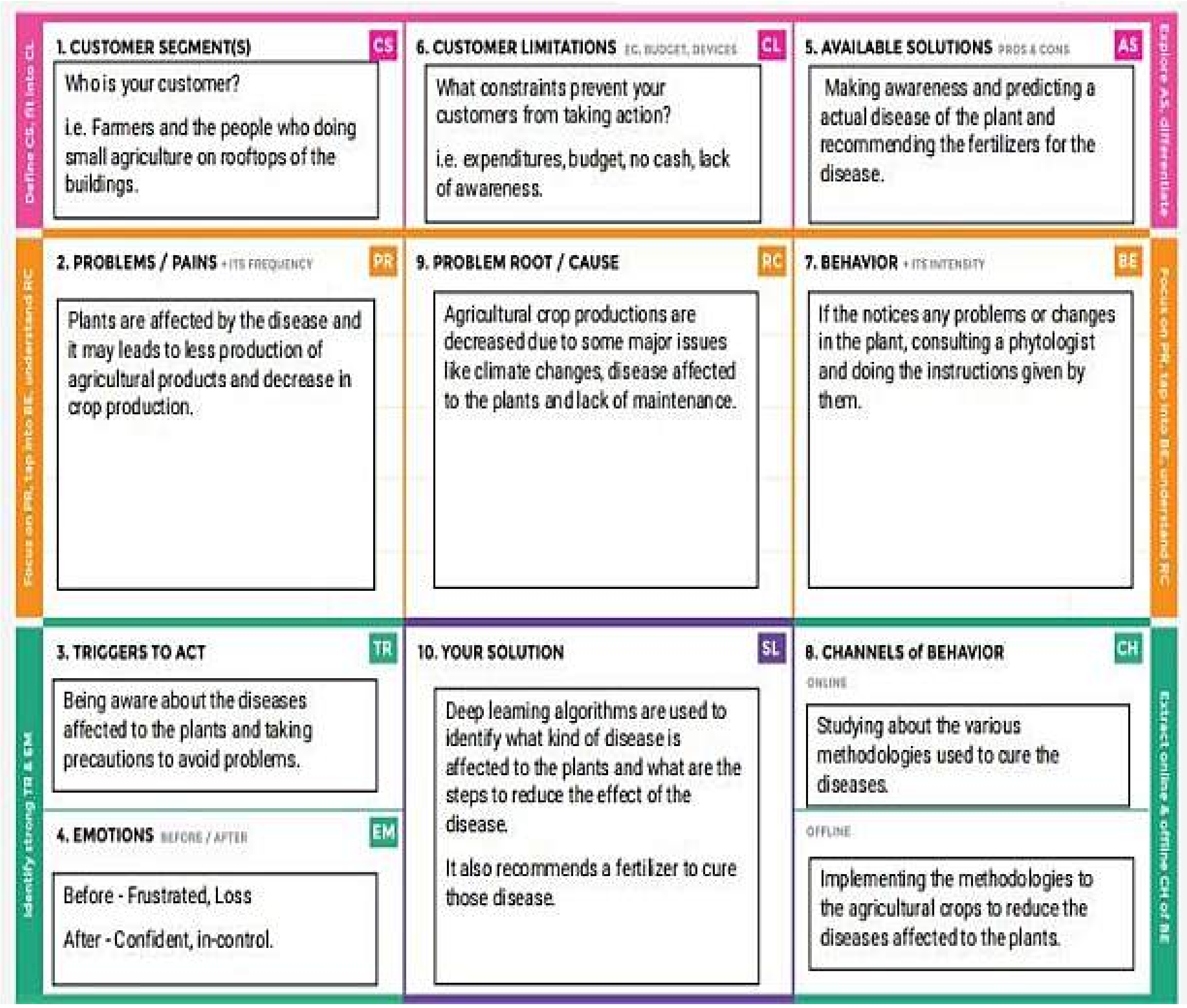


Figure.3.4 Problem Solution fit

Problem solution fit explains the problems and issues faced by the customers(Farmers), issues may be happened due to some reasons like irregular maintenance and lack of awareness and etc..

It provides the possible way to minimize the issues and also provides the solutions for those problems.

It also conveys the emotions of the customers before they using this system and after they using this system.

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 Login**	Via Former/User
FR-2	**User Dashboard**	Single Sample Prediction Multiple Sample Prediction Image Sample Analysis
FR-3	**Prediction Generation**	Disease Report Fertilizer Suggestion

Table.4.1 Functional requirements

4.2 Non-Functional requirements

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	User-friendly interface to use,disease prediction,fertilizer recommendation system.
NFR-2	**Security**	The proposed method uses SVM to classify leaves, Identify the disease and suggest the fertilizer.
NFR-3	**Reliability**	Recommendation system is the new era of research to predict the thinga to the end users. The predicting fertilizer is very reliable product.
NFR-4	**Performance**	It’s Very high accuracy to detect the disease and suggest the perfect fertilizer.
NFR-5	**Availability**	We use mobile applications to predict and analyzethe disease.we use it every place with help of the application.
NFR-6	**Scalability**	It’s a high range to train more images and it’s support vector vision is 98 percent.

Table.4.2 Non-Functional requirements

5.PROJECT DESIGN

5.1 Data Flow Diagrams

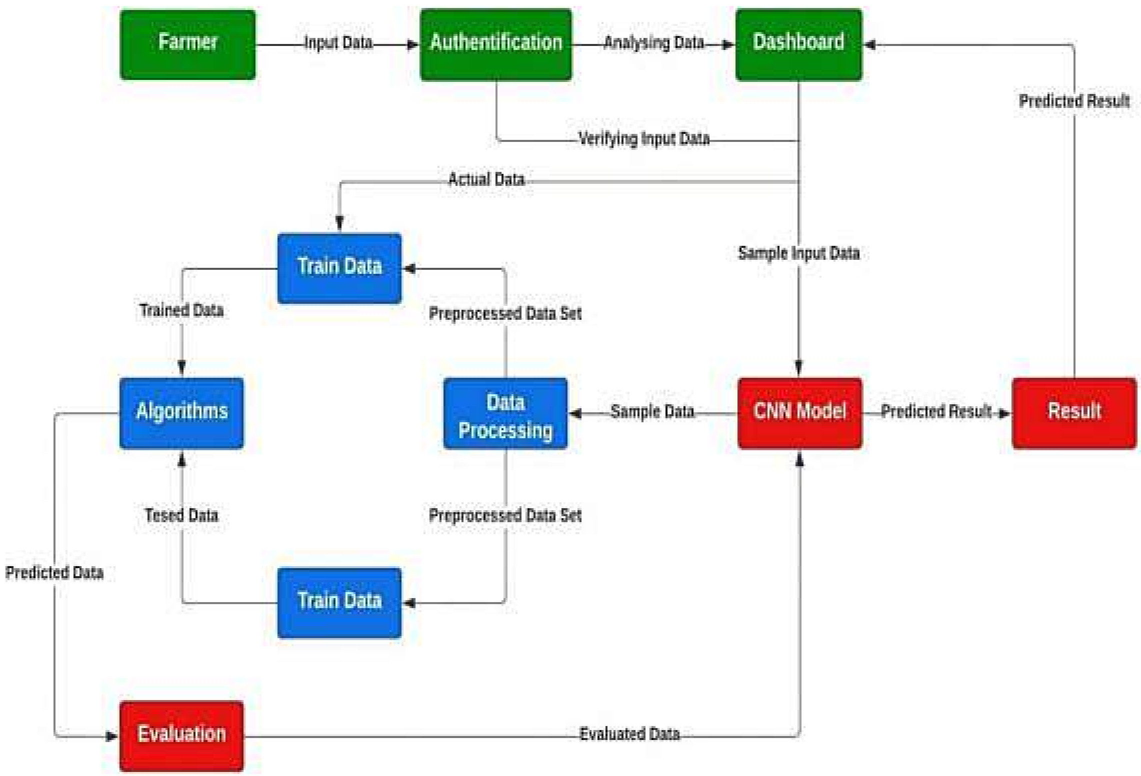


Figure.5.1 Data Flow Diagrams

Data Flow Diagrams are the diagrammatical representation of the plant disease predicting process.

It contains the entire process from initial part to final part.

5.2 Solution & Technical Architecture

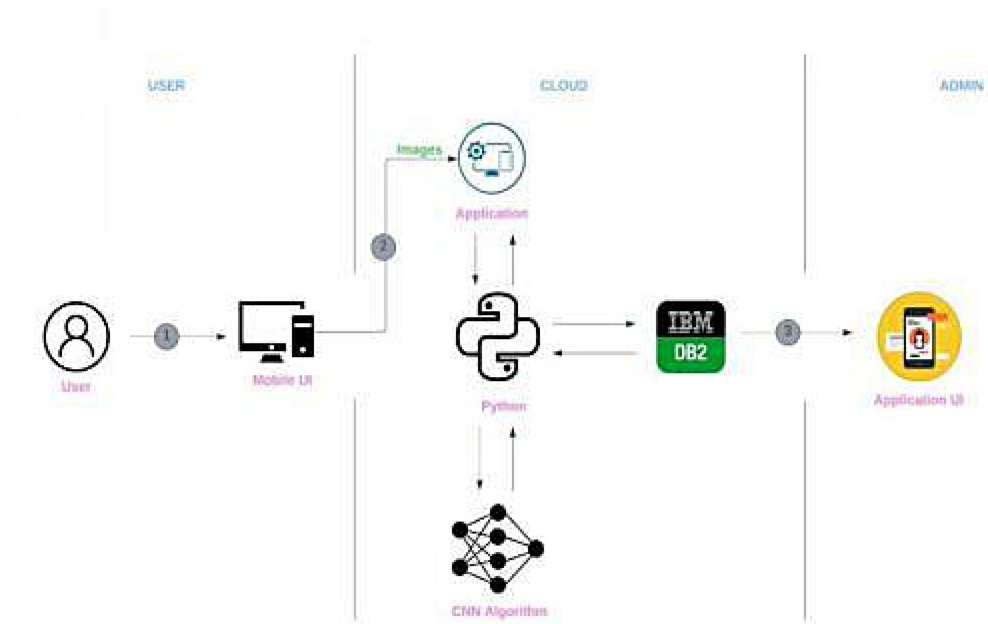


Figure.5.2 Solution & Technical Architecture

S.No	Component	Description	Technology
1.	Mobile Phone	User interacts with mobile phone to predict the Fertilizers.	HTML, CSS, python,etc.
2.	Process	It process the set of images to preprocessed and to be trained and tested.	Python
3.	Cloud Database	The IBM cloud database contains non structural data such as dataset and disease affected images.	IBM Cloudant etc.
4.	File Storage	The input files to be stored as IBM cloud and after it will show the recent files.	IBM Block Storage or Other Storage Service or Local Filesystem.
5.	Deep Learning Model	The deep learning model to use of image classification and image segmentation,prediction.	Algorithms-Support Vector Machine,etc.
6.	Infrastructure (Server / Cloud)	Application deployment on local mobile systems	Cloud servers and other cloud services.

Table.5.2.1 Components & Technologies

S.No	Characteristics	Description	Technology
1.	Open-Source Frame works	Backend Framework,Frontend Framework,RDS	Python,Figma.
2.	Security Implementation	No need for high security.It is a open source web application anybody can use to predict their plant diseases.	There is no any specific technology needed.
3.	Scalable Architecture	Support large number of images to be accessed using data framing.	Numpy,pandas.
4.	Availability	Availability increased by using application load balancers.It will reduce load of the application.	IBM cloud network and security.
5.	Performance	The prediction goes to 1000 predicts in nano second	ICnn model and flask framework

Table.5.2.2 Application Characteristics

Solution and Technical architecture includes the requirements of the system and components like software, tools, algorithms used in this system.

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Login	USN-1	User can view to the home page	User can access their profile	High	Sprint-3
	Dashboard	USN-2	After input, user can access their prediction interface	User can predict for single sample	High	Sprint-3
Customer (Organization)		USN-3	After entering to the dashboardcan give multiple input as a images	User can give multiple sampleinputs	Medium	Sprint-2
		USN-4	User can get visual representation of the prediction	User can have different forms of output	High	Sprint-3
		USN-5	User can view the detailed report of prediction of the plant disease	User can access to view the entire process and documentation	Medium	Sprint-4
	Documentation	USN-6	User can refer the documentation of the system forthe reference and clearence	User can use the References manual for guidance	Medium	Sprint-1,2,3,4
Developer	Settings	USN-7	Developer can access the dashboard’s settings and view the reports of the entire process	User can view the API token for creating request	Low	Sprint-4

Table.5.3 User Stories

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

SPRINT 1

Phase : Modelling and Development Phase

Task : Data collection, Image preprocessing.

```
testing_set = test_data_gen.flow_from_directory(path2,
                                                target_size=(128,128),
                                                batch_size=32,
                                                class_mode='categorical',
                                                shuffle=True,
                                                color_mode='rgb',
                                                subset = 'validation')

    Found 3770 images belonging to 6 classes.
    Found 503 images belonging to 6 classes.
```

Figure.6.1.1 Sprint 1

SPRINT 2

Phase : Modelling and Development Phase

Task : Model Building, Testing the Model.

```
In [6]: from tensorflow.keras.models import Sequential
        from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten

In [7]: model=Sequential()

In [8]: model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
        model.add(MaxPooling2D(pool_size=(2,2)))
        model.add(Flatten())

        32*(3*3*3+1)
        model.add(Dense(300,activation='relu'))
        model.add(Dense(150,activation='relu'))
        model.add(Dense(6,activation='softmax'))
        model.summary()

In [5]: x_train=train_datagen.flow_from_directory(path_train,target_size=(128,128),
                                                  class_mode='categorical',batch_size=24)
        x_test=test_datagen.flow_from_directory(path_test,target_size=(128,128),
                                                  class_mode='categorical',batch_size=24)

        Found 11386 images belonging to 9 classes.
        Found 3416 images belonging to 9 classes.

In [6]: model = Sequential()

In [7]: # Step 1 - Convolution
        model.add(Conv2D(32, (3, 3), input_shape = (128, 128, 3), activation = 'relu'))

        # Step 2 - Pooling
        model.add(MaxPooling2D(pool_size = (2, 2)))

        # Adding a second convolutional layer
        model.add(Conv2D(32, (3, 3), activation = 'relu'))
        model.add(MaxPooling2D(pool_size = (2, 2)))

        # Step 3 - Flattening
        model.add(Flatten())

        # Step 4 - Full connection
        model.add(Dense(units = 300, activation = 'relu'))
        model.add(Dense(units = 150, activation = 'relu'))
        model.add(Dense(units = 9, activation = 'softmax'))

In [8]: model.summary()
```

Figure.6.1.2 Sprint 2

SPRINT 3

Phase : Modelling and Development Phase

Task : Home page and Input page, Flask Web Application Framework.

```
<!DOCTYPE html>
<html>
  <head>
    <link href="https://fonts.googleapis.com/css?family=Inter&display=swap" rel="stylesheet" />
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    </head>
    <body>
      <div class="v157_102">
        <div class="v157_104"></div>
        <div class="v157_105">
          <span class="v157_106">Plantect.</span>
          <span class="v157_107">Plantect is an automated system introduced to identify different diseases on plant.</span>
          <a href="/predict2"><button class="button button1">Go -></button></a>
        <div class="v157_108">
         <div class="v157_109"></div>
        </div>
        </div>
      </div>

      </div>
        </body>
        </html>
        <br/><br/>
        <style>*
```

```
model = load_model(r"Uploads\Vegetable .h5")
model1 = load_model(r"Uploads\fruitdata.h5")
print(model)

print("Model Loaded Successfully")

# Create flask instance
app = Flask(__name__)

# render index.html page
@app.route("/")
def home():
    return render_template('home.html')

@app.route("/predict2", methods=['GET', 'POST'])
def predict2():
    return render_template('predict2.html')


# render index.html page
@app.route("/predict1", methods=['GET', 'POST'])
def predict1():
    return render_template('predict1.html')
```

Figure.6.1.3 Sprint 3

SPRINT 4

Phase : Modelling and Development Phase

Task : Testing the Back - end and Front - end.

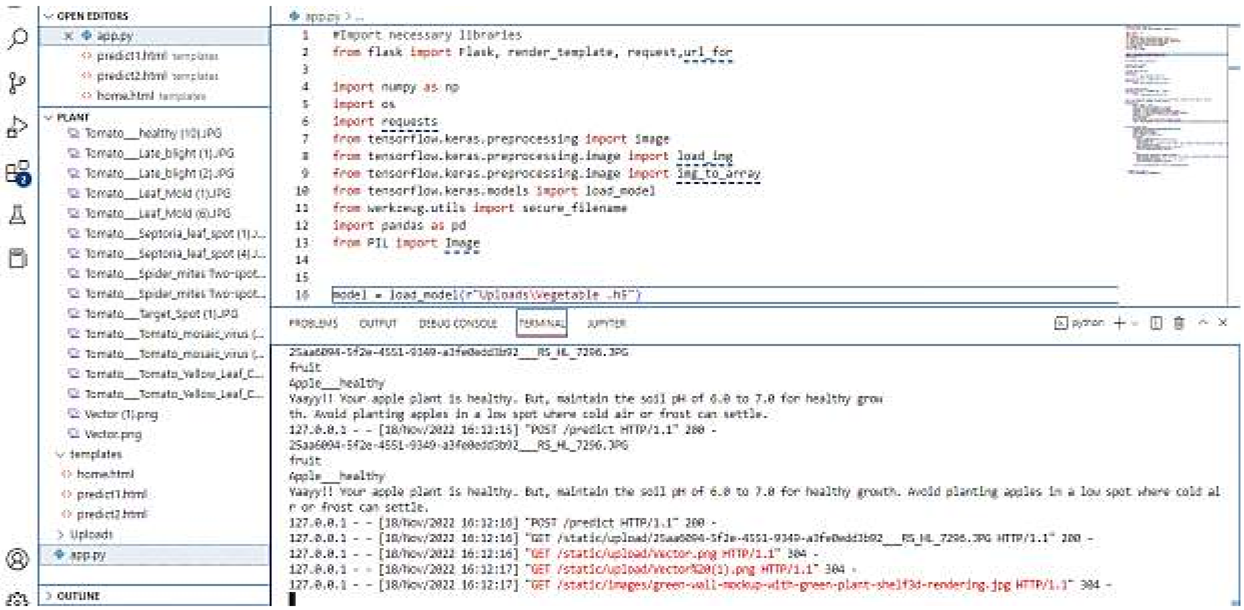


Figure.6.1.4 Sprint 4

Sprint Delivery Schedule :

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)
Sprint-1	12	6 Days	24 Oct 2022	29 Oct 2022	29 Oct 2022
Sprint-2	21	4 Days	30 Oct 2022	02 Nov 2022	02 Nov 2022
Sprint-3	09	5 Days	03 Nov 2022	07 Nov 2022	07 Nov 2022
Sprint-4	24	5 Days	08 Nov 2022	12 Nov 2022	12 Nov 2022

Table.6.1 Sprint Delivery Schedule

6.2 Reports from JIRA

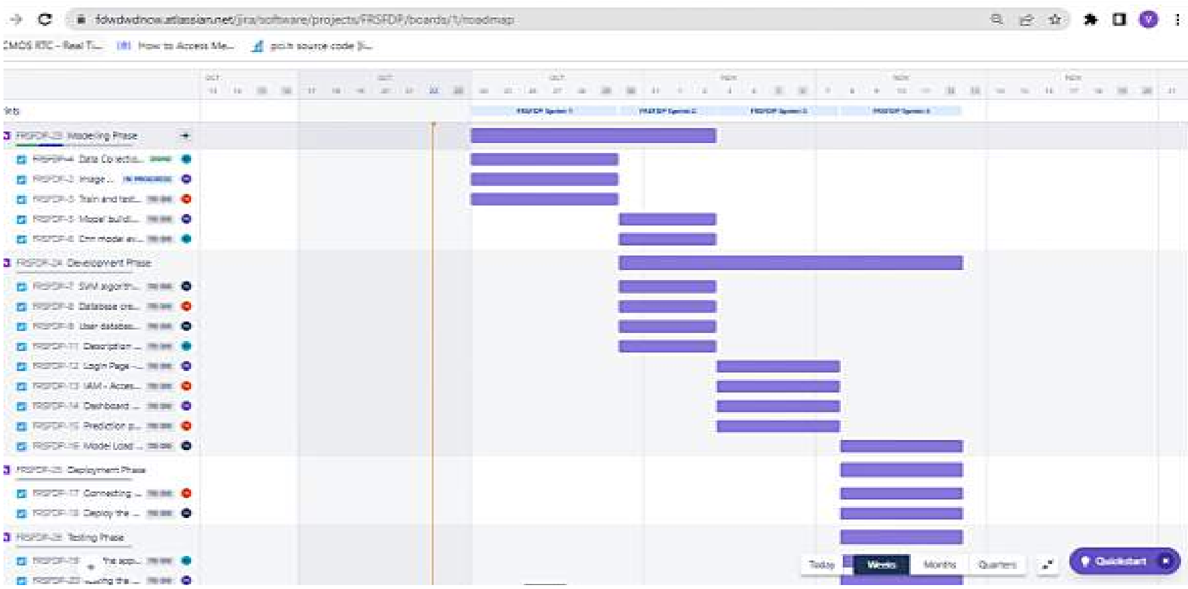


Figure 6.2 Reports from JIRA

6.3 Velocity Report

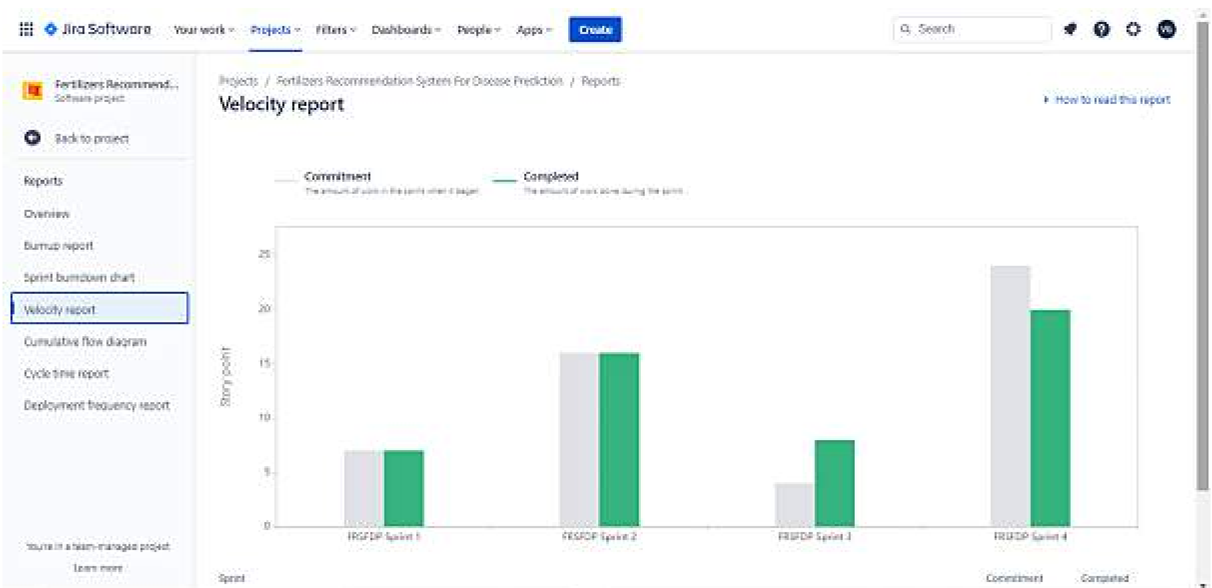


Figure.6.3 Velocity Report

7. CODING & SOLUTIONING

7.1 Feature 1

BACKEND :

- ➤ Creating the models for fruits and vegetable leaves.
- ➤ Save each model in (.h5) format.
- ➤ All the models are integrated using app.py file

7.2 Feature 2

FRONTEND :

- ➤ There are three HTML files
- ➤ “home.html” is the main file (Homepage)
- ➤ “predict2.html” page shows the prediction and fertilizer recommendation
- ➤ “app.py” is used to connect the frontend and backend files

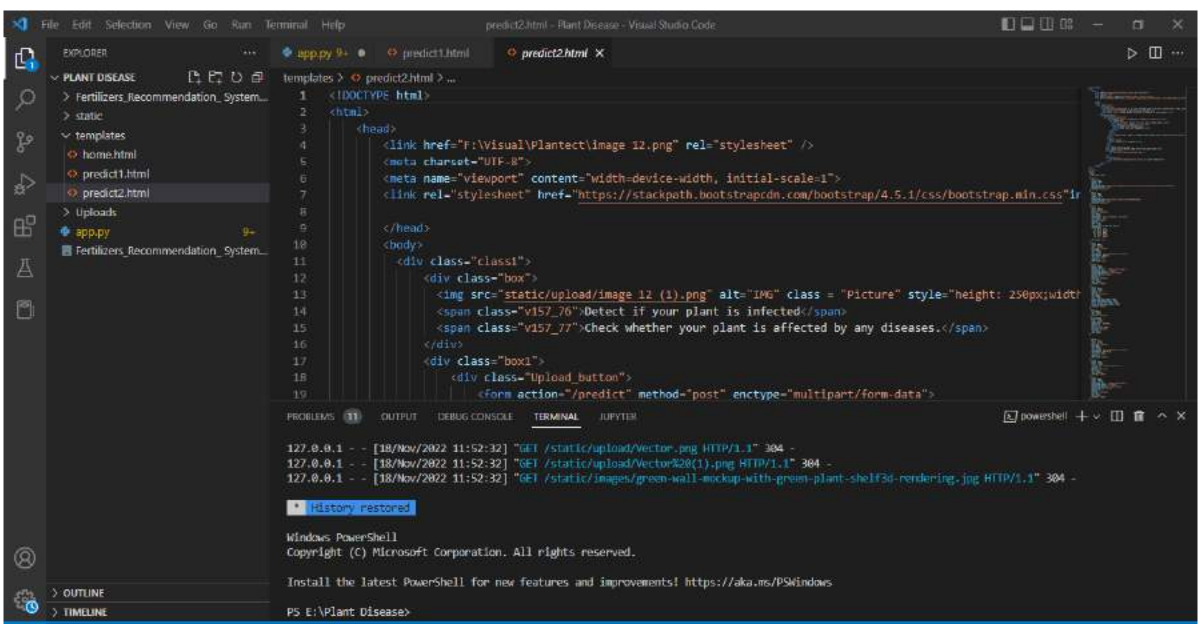
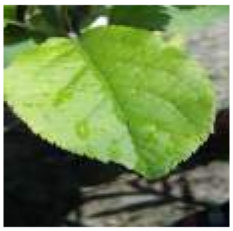


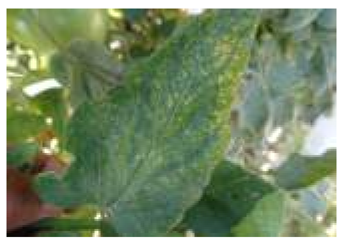
Figure7.2 Frontend

DATASET :

Fruit



Vegetable


GDrive Link :

https://drive.google.com/drive/folders/1tl0sq84Z2jhtp4gU8lEDYBFzLOZiZ6GH?usp=share_link

8. TESTING

8.1 Test Cases

Figure.8.1.1 Test Cases

Figure.8.1.2 Test Cases

Fruit :

Figure.8.1.3 Fruit Output

Vegetable :

Figure.8.1.4 Vegetable Output

8.2 User Acceptance

Different types of image datasets are provided, and then those are used to identify the affected area in leaves. Then different types of image-processing techniques are applied to them.

The process those images, to get different and useful features needed for the purpose of analyzing later-Plant leaf disease identification is especially needed to predict both the quality and quantity of the First segmentation step primarily based on a mild polygonal leaf model is first achieved and later used to guide the evolution of an energetic contour.

Combining global shape descriptors given by the polygonal model with local curvature based features, the leaves are then classified overleaf datasets.

In this research work introduce a method designed to deal with the obstacles raised by such complex images, for simple and plant leaves. A first segmentation step based on graph-cut approach is first performed and later used to guide the evolution of leaf boundaries, and implement classification algorithm to classify the diseases and recommend the fertilizers to affected leaves.

9. RESULTS

The accuracy for the fruit leaf disease prediction is upto 96% and accuracy for the vegetable leaf disease prediction is upto 92% .

Also,it will recommends the suitable fertilizers for the disease.

Fruit :

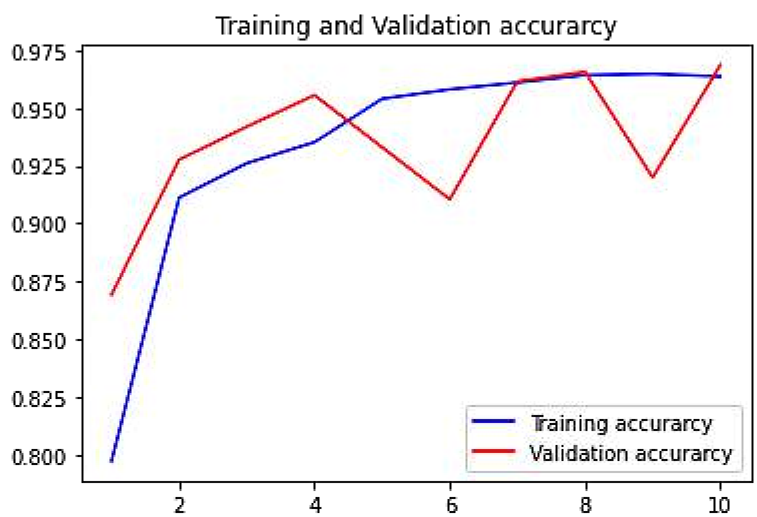


Figure.9.1 Fruit

Vegetable :

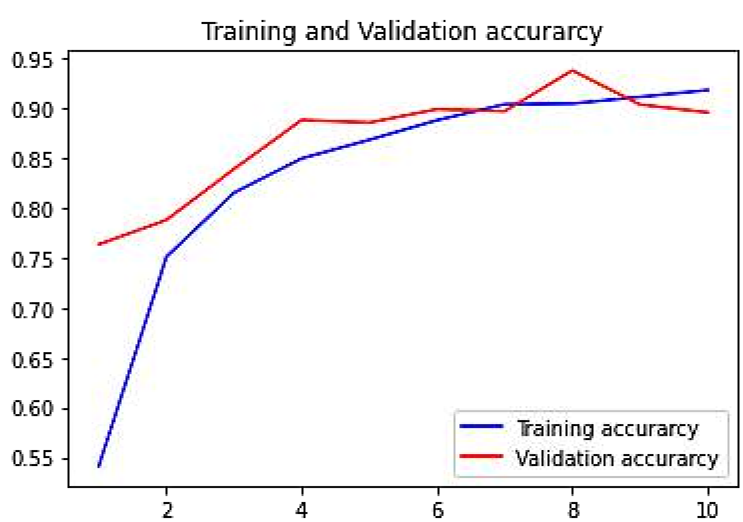


Figure.9.2 Vegetable

10. ADVANTAGES & DISADVANTAGES

Advantages :

- Detects related images with a low-cost camera and opencv.
- Opencv aids in the efficient analysis of images and videos.
- It also reduces a large work of monitoring in big farms of crops, and at very early stage itself it detects the symptoms of diseases.
- 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.

Disadvantages :

- Consumption of space is very high.
- 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

Deep learning methods are used to train the model, which aids in making appropriate disease decisions. To contain infected diseases, the farmer is advised to use pesticides as a cure.

In the future, the proposed scheme could be expanded to provide additional facilities such as nearby government markets, pesticide price lists, and a nearby open market, among others.

This paper presents a review of various disease classification strategies for plant disease detection and providing a solution by recommending fertilizers for the diseases.

Different approaches and models of Deep Learning methods were explored and used in this project so that it can detect and classify plant diseases correctly through image processing of leaves of the plants.

The procedure starts from collecting the images used for training, testing and validation to image preprocessing and augmentation and finally comparison of different pretrained models over their accuracy. Finally, at the end , our model detects and distinguishes between a healthy plant and different diseases and provides suitable remedies so as to cure the disease.

This paper proposed and developed a system which uses plant leaf images to detect different types of disease in fruit and vegetable,it provides appropriate fertilizer suggestions.

12. FUTURE SCOPE

The prediction of crop yield based on location and proper implementation of algorithms have proved that the higher crop yield can be achieved.

From above work we conclude that for soil classification Random Forest is good with accuracy 86.35% compare to Support Vector Machine. For crop yield prediction Support Vector Machine is good with accuracy 99.47% compare to Random Forest algorithm.

The work can be extended further to add following functionality. Mobile application can be build to help farmers by uploading image of farms. Crop diseases detection using image processing in which user get pesticides based on disease images. Implement Smart Irrigation System for farms to get higher yield.

The system successfully interprets various Diseases and is also capable of providing fertilizers suggestion for the respective disease. Further more, this system can be made more robust by incorporating more image dataset with wider variations like more than one leaf in a single image.

An App could also be developed for the project which could make the work of the farmers easier.They could directly upload image on the app and it would tell the disease and the cure then and there.This would reduce the time and efforts.

13. APPENDIX

13.1 Souce Code

app.py

```
#Import necessary libraries
from flask import Flask, render_template, request,url_for

import numpy as np
import os
import requests
import base64
import io
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
from werkzeug.utils import secure_filename
import pandas as pd
from PIL import Image

model = load_model(r"Uploads\Vegetable .h5")
model1 = load_model(r"Uploads\fruitdata.h5")
print(model)

print("Model Loaded Successfully")

# Create flask instance
app = Flask(__name__)

# render index.html page
@app.route("/")
def home():
    return render_template('home.html')

@app.route("/predict2", methods=['GET', 'POST'])
def predict2():
    return render_template('predict2.html')

# render index.html page
@app.route("/predict1", methods=['GET', 'POST'])
```

```
def predict1():
        return render_template('predict1.html')


# get input image from client then predict class and render respective .html page for solution
@app.route("/predict", methods = ['GET','POST'])
def predict():
    if request.method == 'POST':
        file = request.files['image1'] # fet input
        filename = secure_filename(file.filename)
        basepath = os.path.abspath(os.path.dirname(__file__))
        file_path = os.path.join(basepath,r'static\upload',filename)
        img_path = os.path.join(r'static\upload',filename)
        file.save(file_path)
        print(filename)
        Prediction_image,output=leaves(Plant_image=file_path)
        return
render_template('predict1.html',pred_output=Prediction_image,Disease=output,value=img_path,flag=True)


def leaves(Plant_image):
        img=image.load_img(Plant_image,target_size=(128,128))
        x=image.img_to_array(img)
        x=np.expand_dims(x,axis=0)
        plant = request.form.get('Plant')
        print(plant)
        if(plant == "vegetable"):
            prediction=np.argmax(model.predict(x),axis=1)
            print(prediction)

index=['Pepper,_bell___Bacterial_spot','Pepper,_bell___healthy','Potato___Early_blight','Potato___Late_blight
','Potato___healthy','Tomato___Bacterial_spot','Tomato___Late_blight','Tomato___Leaf_Mold','Tomato___Se
ptoria_leaf_spot']
            print(index[prediction[0]])
            df=pd.read_excel(r'Uploads\precautions - veg.xlsx')
            print(df.iloc[prediction[0]]['caution'])

        else:
            prediction=np.argmax(model1.predict(x),axis=1)

index=['Apple___Black_rot','Apple___healthy','Corn_(maize)___Northern_Leaf_Blight','Corn_(maize)___hea
lthy','Peach___Bacterial_spot','Peach___healthy']
```

```
        print(index[prediction[0]])
        df=pd.read_excel(r'Uploads\precautions - fruits.xlsx')
        print(df.iloc[prediction[0]]['caution'])

      return df.iloc[prediction[0]]['caution'],index[prediction[0]]

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

13.2 Screenshots

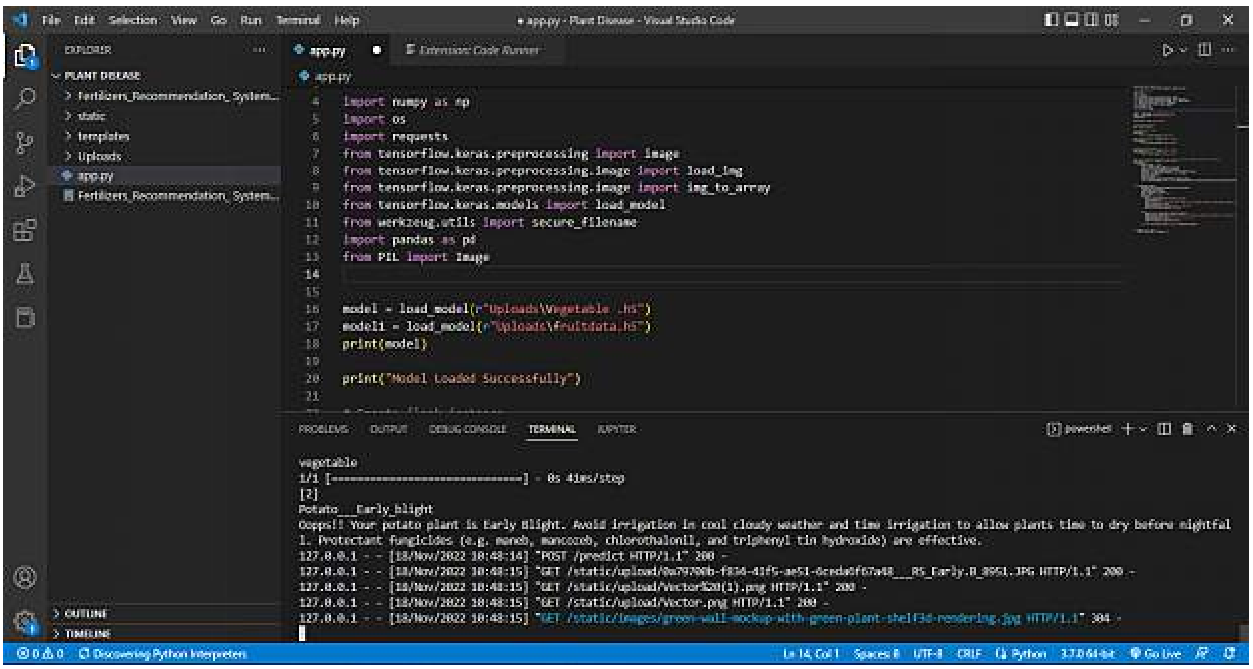


Figure.13.2 Screenshots

Github Link : https://github.com/IBM-EPBL/IBM-Project-39867-1660557591

Youtube Link : https://youtu.be/Z4B2CZCm0Nc

14. References

1. S. Nandhini , “Analysis on Prediction of Plant Leaf diseases using Deep Learning ” , 2021.
2. Ms. Deepa , Ms. Rashmi N , “A Machine Learning Technique for Identification of Plant Diseases in Leaves” , 2021.
3. Ms. Rubini PE , “Deep Learning model for early prediction of plant disease”, 2020.
4. Ms. Dhapitha Nesarajan, Ms. Lokini Kunalan , “Coconut Disease Prediction System Using Image Processing and Deep Learning Techniques” , 2020.
5. Ms. J.Amara, Mr. B.Bouaziz, “A Deep Learning-Based Approach for Banana Leaf Diseases Classification”, 2018.
6. Mr. Hyeon Park,Mr. Eun JeeSook,Mr. Se-Han Kim, “Crops Disease Diagnosing using Image-based Deep Learning Mechanism,” 2017.
7. Mr. B. Leff,Mr. N. Ramankutty, and Mr. J. A. Foley, “Geographic distribution of major crops across the world" , 2017.