### Fertilizers Recommendation System For Disease Prediction

**NALAIYA THIRAN PROJECT BASED LEARNING**

**On**

**PROFESSIONAL READINESS FOR INNOVATION,**

**EMPLOYABILITY AND ENTREPRENEURSHIP**

**A PROJECT REPORT**

GUTTA VENKATA SRINIVAS 19104064

GADDAM VENKATA SAI 19104052

GADDE TEJA 19104053

ASHAY KRISHNA 19104027

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**  
**HINDUSTHAN COLLEGE OF ENGINEERING AND TECHNOLOGY**

Approved by AICTE, New Delhi, Accredited with ‘A’ Grade by NAAC  
**(An Autonomous Institution, Affiliated to Anna University, Chennai)**  
 **COIMBATORE – 641 032**

November 2022



***Hindusthan College of Engineering And Technology***

Approved by AICTE, New Delhi, Accredited with ‘A’ Grade by NAAC**(an Autonomous Institution, Affiliated to Anna University,**

**Chennai)**Valley Campus, Pollachi Highway, Coimbatore – 641 032

**INTERNAL MENTOR**

**Mr. Bharathi Kannan, M.E.,M.Tech**

Assistant Professor,

Department of Computer Science and Engineering

Hindusthan College of Engineering and Technology,

Coimbatore-641 032

**INDUSTRY MENTOR**

**DURGA PRASAD (IBM)**

**ANNA UNIVERSITY: CHENNAI 600 025**  **BONAFIDE CERTIFICATE**

# Certified that project report “FERTILIZER RECOMMENDATION SYSTEM FOR DISEASE PREDICTION” is the bonafide work of “GUTTA VENKATA SRINIVAS, GADDE TEJA , GADDAM VENKATA SAI, ASHAY KRISHNA who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE SIGNATURE**

**Dr.S SHANKAR ,M.E ,Phd MR.BHARATHI KANNAN,M.E**

**HEAD OF THE DEPARTMENT SUPERVISOR**

Professor and Head, Associate Professor,

Department of CSE, Department of CSE,

Hindusthan College Of Hindusthan College Of

Engineering and Technology, Engineering and Technology,

Coimbatore-641032 Coimbatore-641032

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted for Project Viva-Voice conducted on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

----------------------------------- -------------------------------------

**INTERNAL EXAMINER**   **EXTERNAL EXAMINER**

## TABLE OF CONTENTS

**CHAPTER NO TITLE**

**1**   **INTRODUCTION**

**1.1 [Project Overview](#_TOC_250002)**

**1.2 Purpose**

**2**   **LITERATURE SURVEY**

**2.1 Existing Problem**

**2.2 References**

**2.3 Problem Statement Definition**

## 3 **IDEATION & PROPOSED SOLUTION**

## **3.1 Empathy Map Canvas**

**3.2 Ideation & Brainstorming**

**3.3 Proposed Solution**

**3.4 Problem Solution Fit**

## 4 REQUIREMENT ANALYSIS

**4.1 Functional Requirement**

**4.2 Non-Functional Requirement**

## 5 PROJECT DESIGN

## **5.1** **Data Flow Diagrams**

**5.2 Solution & TechnicalArchitecture**

**5.3 User Stories**

## 6 PROJECT PLANNING & SCHEDULING

**6.1 Sprint Planning &Estimation**

**6.2 Sprint Delivery Schedule**

**6.3 Reports from JIRA**

**7 CODING & SOLUTIONING**

**7.1 Feature 1**

**7.2 Feature 2**

**8 TESTING**

**8.1 Test Cases**

**8.2 User Acceptance Testing**

9  **RESULTS**

**9.1 Performance Metrics**

**10 ADVANTAGES & DISADVANTAGES**

**11 CONCLUSION**

**12 FUTURE SCOPE**

**13 APPENDIX**

**Source Code**

**Github &Project Demo Link**

**1. INTRODUCTION**

Agriculture is the main aspect of country development. Many people lead their life from the agriculture field, which is fully related to agricultural products. Plant disease, especially on leaves, is one of the major factors of reductions in both quality and quantity of the food crops. In agricultural aspects, if the plant is affected by leaf disease then it reduces the growth of the agricultural level. Finding the leaf disease is an important role of agriculture preservation. After preprocessing using a median filter, segmentation is done by Guided Active Contour method and finally, the leaf disease is identified by using Support Vector Machine. The disease-based similarity measure is used for fertiliser recommendation.

Detection and recognition of plant diseases using machine learning are very efficient in providing symptoms of identifying diseases at its earliest. Plant pathologists can analyse 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, 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 segmentation, feature extraction.

**1.1 Project Overview**

Agriculture is the main aspect of country development. Many people lead their life from the agriculture field, which is fully related to agricultural products. Plant disease, especially on leaves, is one of the major factors of reductions in both quality and quantity of the food crops. In agricultural aspects, if the plant is affected by leaf disease then it reduces the growth of the agricultural level. Finding the leaf disease is an important role of agriculture preservation. After preprocessing using a median filter, segmentation is done by Guided Active Contour method and finally, the leaf disease is identified by using Support Vector Machine. The disease-based similarity measure is used for fertiliser recommendation.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 Purpose**

ThePurpose of this project to review the previous of Researchers on the Fertilisers recommend System For Disease Prediction. This chapter will present the Detection and recognition of plant diseases using machine learning are very efficient in providing symptoms of identifying diseases at its earliest.

Isin kaye, F. 0, & Erute, E. D. (2022) investigated identifying A Smartphone-based Plant Disease Detection and Treatment Recommendation System using Machine Learning Techniques. Transactions on Machine Learning and Artificial Intelligence Plant disease, specifically on Leaves have become a source of serious concern in the agricultural sector as it always causes damage to crops and hence a reduction in the quantity and quality of food production.

However, speedy discovery and accurate identification of these diseases could assist in developing early treatment approaches while significantly reducing economic loss

**2. LITERATURE SURVEY**

**2.1 Existing problem**

Isinkaye, F. O., & Erute, E. D. A Smartphone-based Plant Disease Detection and Treatment Recommendation System using Machine Learning Techniques Transactions on Machine Learning and Artificial Intelligence Folasade Olubusola Isinkaye Department of ComputerScience, Ekiti State University, Ado-Ekiti, Nigeria Emmanuel Damilola Erute Department ofComputer Science, Ekiti State University, Ado-Ekiti, Nigeria(2022)

202DOl:10.14738/timilai. 101.11313.

Pawar, M., & Chillarge, G. (2018). Soil toxicity prediction and recommendation system using data mining in precision agriculture. In 2018 3rd international conference for convergence in technology (12CT) (pp. 1-5). IEEE.

Panigrahi, K. P, Das, H., Sahoo, A. K., & Moharana, S. C. (2020). Maize leaf disease detection and classification using machine learning algorithms. In Progress in Computing, Analytics and Networking (pp. 659-669). Springer, Singapore Hossain, M. A., & Siddique, M. N. A. (2020). Online fertiliser recommendation system (OFRS): A step towards precision agriculture and optimised fertiliser usage by smallholder farmers in Bangladesh. European Journal of Environment and Earth Sciences, 1(4).

Sladojevic, Srdjan, Marko Arsenovic, AndrasAnderla, DubravkoCulibrk, and DarkoStefanovic. "Deep neural networks based recognition of plant diseases by leaf image classification." Computational intelligence and neuroscience2016 (2016).

Uzal, Lucas C., Guillermo L. Grinblat, Rafael Namías, Mónica G. Larese, J. S. Bianchi, E. N. Morandi, and Pablo M. Granitto. "Seed-per-pod estimation for plant breeding using deep learning." Computers and electronics in agriculture 150 (2018): 196-204.

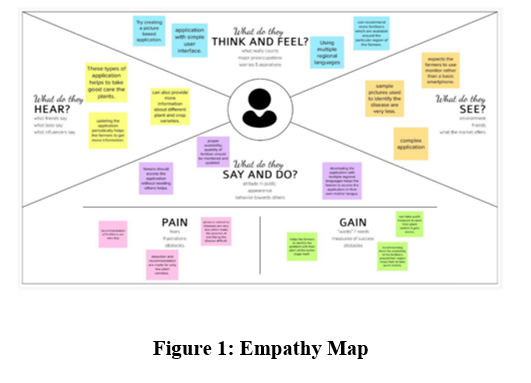
Lee, Sue Han, Chee Seng Chan, Simon Joseph Mayo, and Paolo Remagnino. "How deep learning extracts and learns leaf features for plant classification." Pattern Recognition 71 (2017): 1-13.

**2.2 Problem Statement definition**

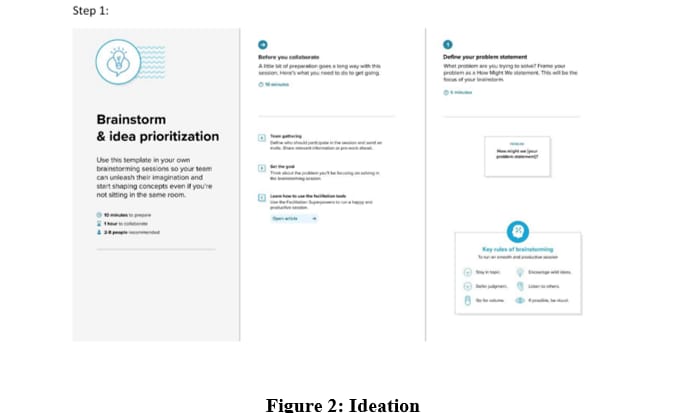


**3.IDEATION PHASE**

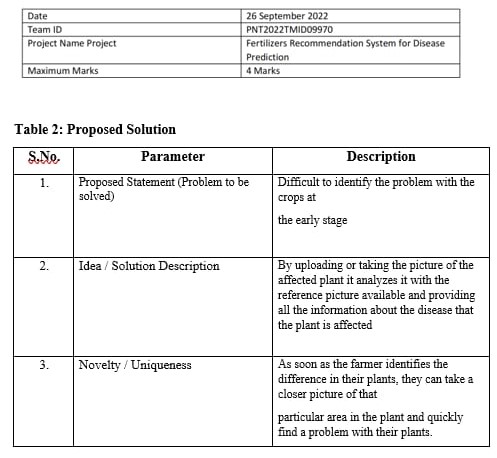
**3.1 Empathy Map**

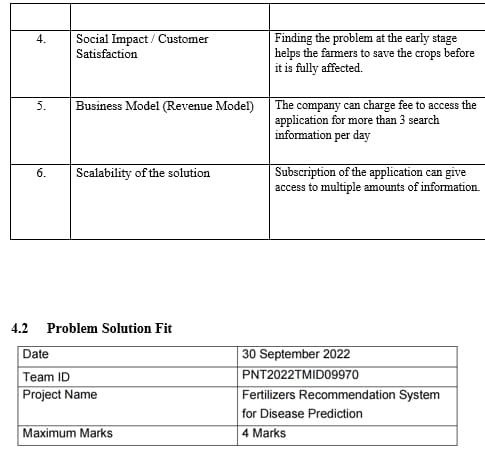


**3.2 Ideation**



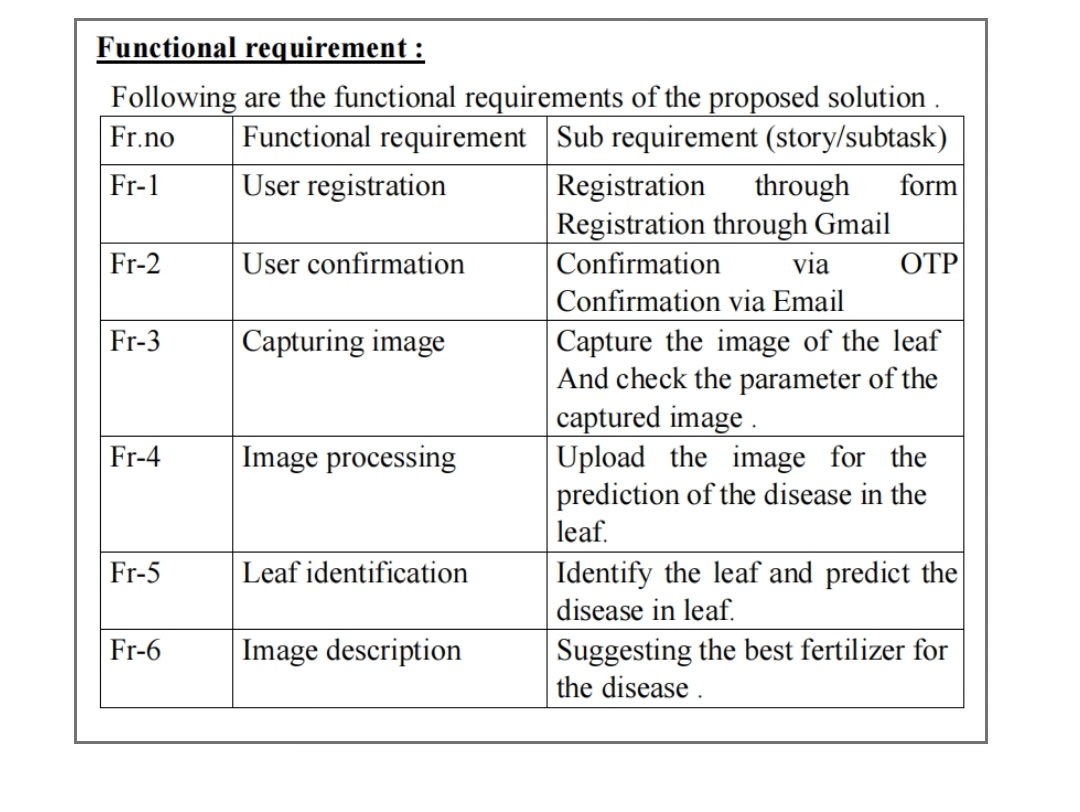
**3.3 Proposed Solution**



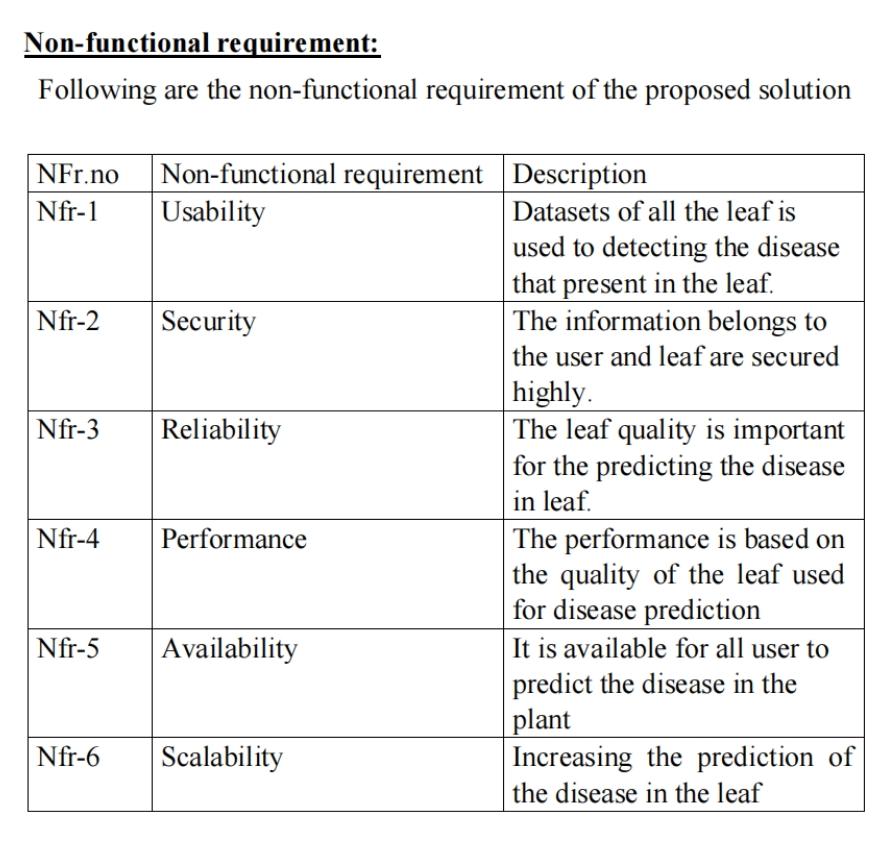
**3.4 Problem Solution fit** 

**4.REQUIREMENT ANALYSIS**

**4.1 FUNCTIONAL REQUIREMENT**

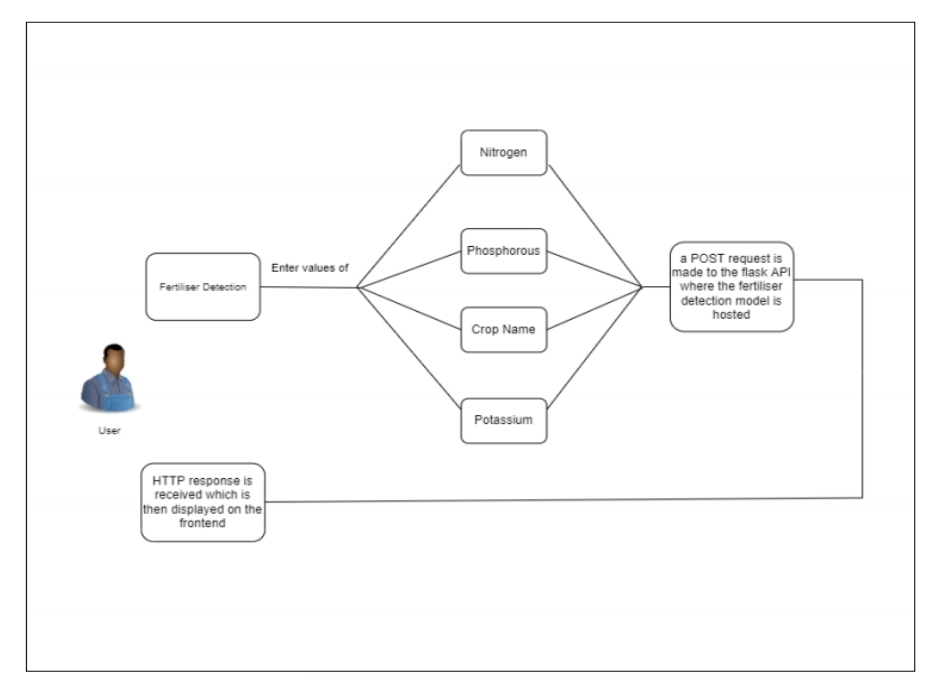


**4.2 Non-Functional Requirement**



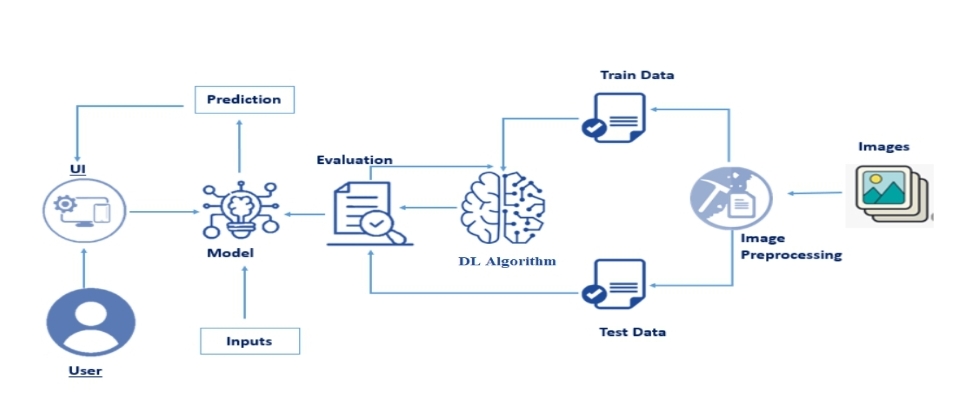
**5.PROJECT DESIGN**

**5.1 Data Flow Diagram**

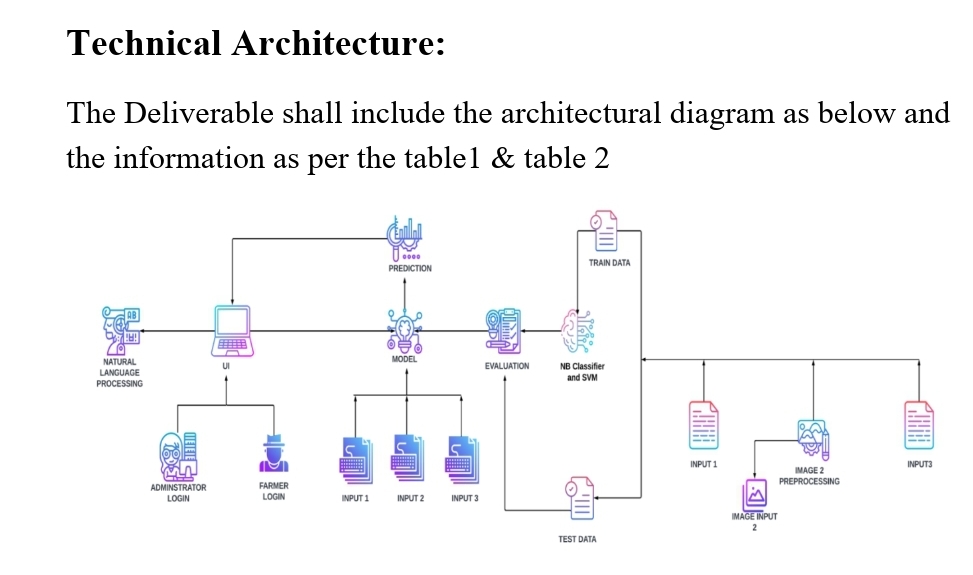


**5.2 Solution and Technical Architecture**

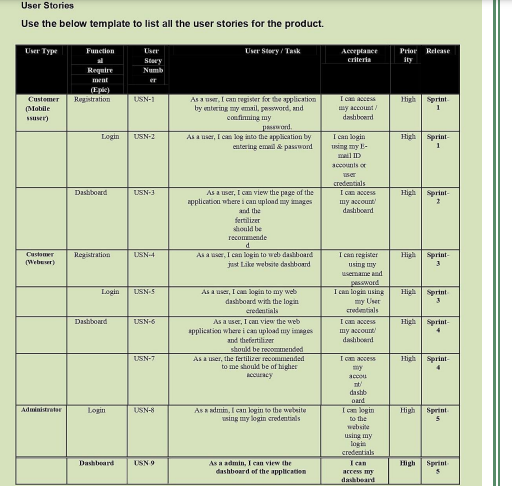
**Solution Architecture**



**Technical Architecture**

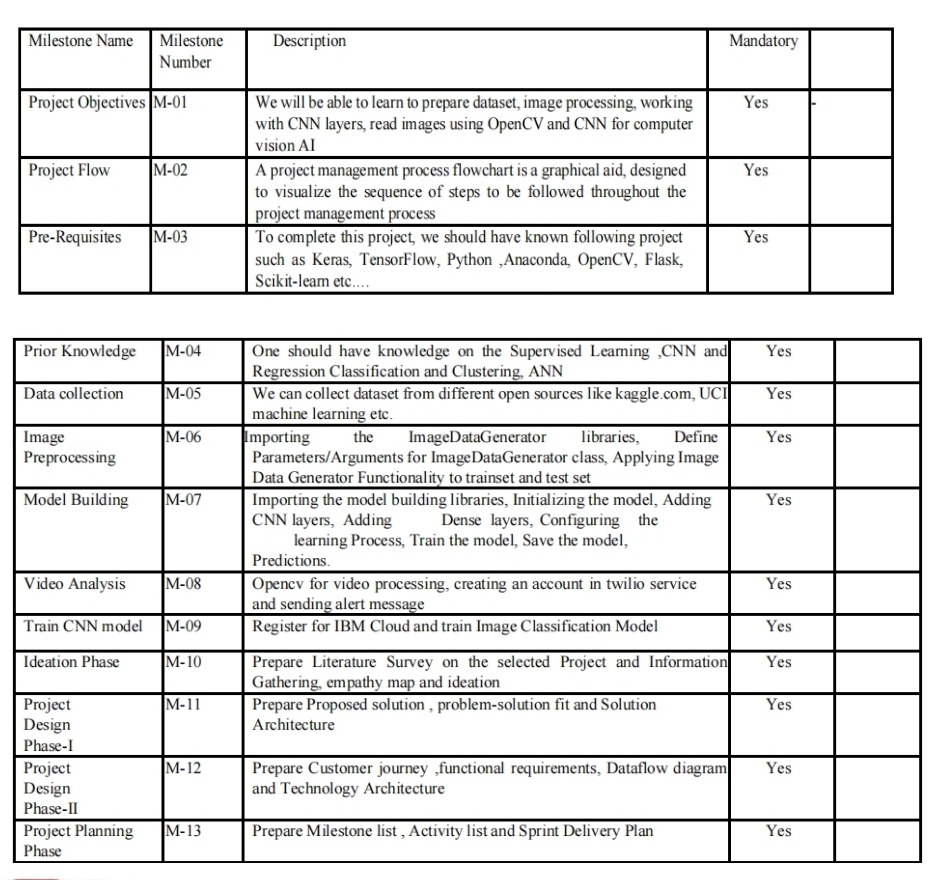
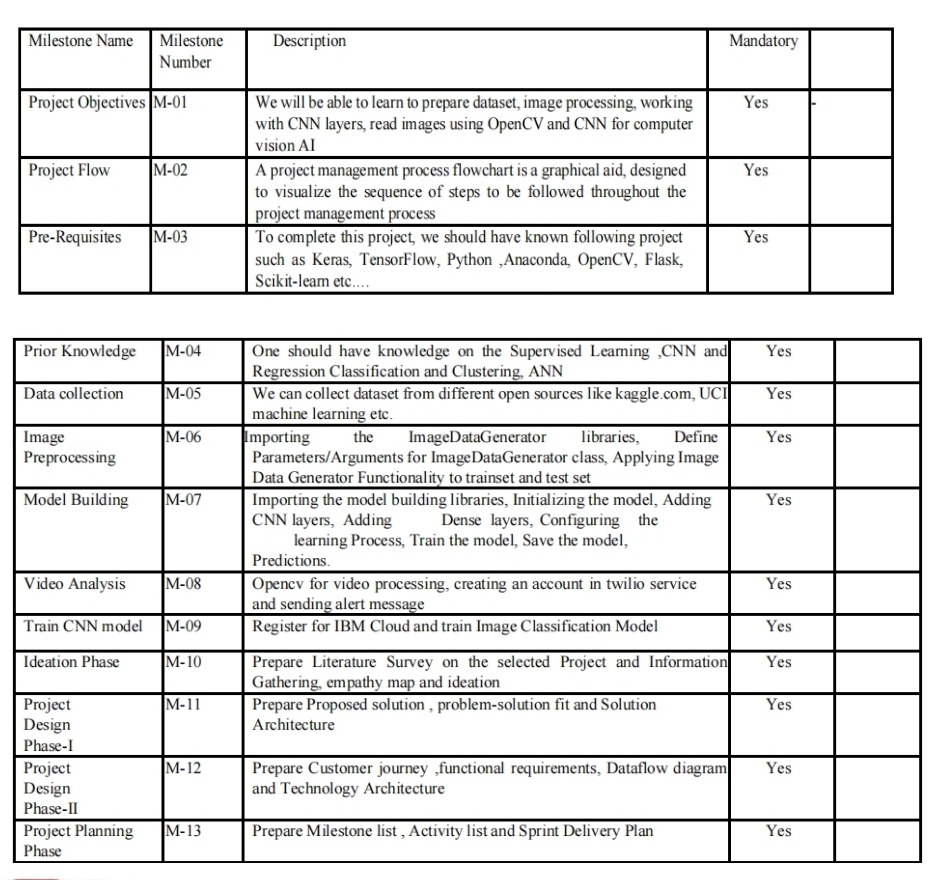


**5.3 User Stories**



**6. PROJECT PLANNING PHASE**

**6.1 Milestone and Activity List**



**6.2 Sprint Delivery and Schedule**

Use the below template to create product backlog and sprint schedule

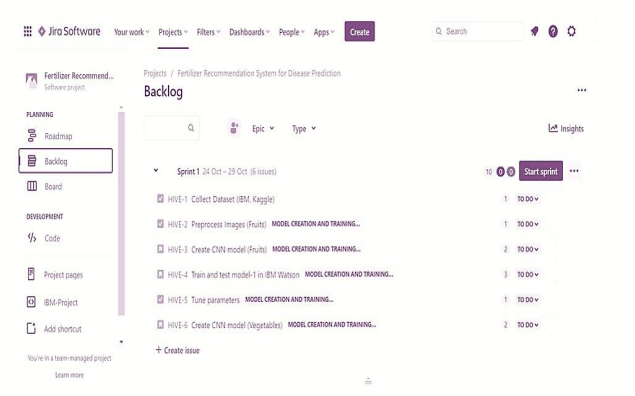
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** |
| Sprint-1 | User input | USN-1 | As a user, I can input the particular URL in the required field and waiting for validation. | 2 |
| Sprint-1 | Feature extraction | USN-1 | Here system can extract feature using  heuristic and visual similarity approach | 1 |
| Sprint-1 | Prediction | USN-1 | Here the Model will predict the URL websites using Machine Learning algorithms | 2 |
| Sprint-1 | Classifier | USN-1 | Here it will send all the model output to classifier in order to produce final result | 2 |
| Sprint-1 | Announcement | USN-1 | Displays whether website is a legal  site or a phishing site. | 1 |
| Sprint-2 | Bugs | USN-2 | As a user, I can report bugs in the application | 1 |
| Sprint-2 | Feedback | USN-3 | As a user, I can send feedback about the application and opinions for improvement | 1 |
| Sprint-3 | Tips | USN-4 | Here cyber security tips are provided for the Customers/Users | 1 |

Delivery Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total Story Points** | **Duration** | **Sprint End Date (Planned)** | **Story Points Completed (as on Planned End Date)** | **Sprint Release Date (Actual)** |
| Sprint-1 | 20 | 6 Days | 29 Oct 2022 | 20 | 29 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 05 Nov 2022 | 20 | 05 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 12 Nov 2022 | 20 | 12 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 19 Nov 2022 | 20 | 19 Nov 2022 |

**6.3 Report from JIRA**





**7.CODING AND SOLUTIONING**

**7.1 Feature 1**

import os, types

import pandas as pd

from botocore.client import Config

import ibm\_boto3

def \_\_iter\_\_(self): return 0

# @hidden\_cell

# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.

# You might want to remove those credentials before you share the notebook.

cos\_client = ibm\_boto3.client(service\_name='s3',

ibm\_api\_key\_id='kd5eBcEmb7JJzWSXkYlPmUw2UY5ZVWDZENSIy9s3bnzg',

ibm\_auth\_endpoint="https://iam.cloud.ibm.com/oidc/token",

config=Config(signature\_version='oauth'),

endpoint\_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'fertilizerrecommendation-donotdelete-pr-let4go6j2nmgdu'

object\_key = 'PlantDoc-Dataset.zip'

streaming\_body\_2 = cos\_client.get\_object(Bucket=bucket, Key=object\_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.

# Please read the documentation of ibm\_boto3 and pandas to learn more about the possibilities to load the data.

# ibm\_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/

# pandas documentation: http://pandas.pydata.org/

# In[10]:

from io import BytesIO

import zipfile

unzip = zipfile.ZipFile(BytesIO(streaming\_body\_2.read()), "r")

file\_paths = unzip.namelist()

for path in file\_paths:

unzip.extract(path)

# In[11]:

pwd

# In[12]:

import os

filenames = os.listdir('/home/wsuser/work/PlantDoc-Dataset/train')

# In[13]:

x\_train=train\_datagen.flow\_from\_directory("/home/wsuser/work/PlantDoc-Dataset/train",target\_size=(128,128),class\_mode='categorical',batch\_size=24)

# In[14]:

x\_test=test\_datagen.flow\_from\_directory(r"/home/wsuser/work/PlantDoc-Dataset/test",target\_size=(128,128),

class\_mode='categorical',batch\_size=24)

# In[15]:

x\_train.class\_indices

# # CNN

# # In[16]:

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten

# In[17]:

model=Sequential()

# In[18]:

model.add(Convolution2D(32,(3,3),input\_shape=(128,128,3),activation='relu'))

# In[19]:

model.add(MaxPooling2D(pool\_size=(2,2)))

# In[20]:

model.add(Flatten())

# In[21]:

model.summary()

# In[22]:

32\*(3\*3\*3+1)

# # Hidden Layers

# In[23]:

model.add(Dense(300,activation='relu'))

model.add(Dense(150,activation='relu'))

# # Output Layer

# In[24]:

model.add(Dense(6,activation='softmax'))

# In[25]:

model.compile(loss='categorical\_crossentropy',optimizer='adam',metrics=['accuracy'])

# In[26]:

len(x\_train)

# In[27]:

1238/24

# In[29]:

model.fit(x\_train,steps\_per\_epoch=len(x\_train),validation\_data=x\_test,validation\_steps=len(x\_test),epochs=10)

# # Saving Model

# In[ ]:

ls

# In[ ]:

model.save('fruit.h5')

# In[ ]:

get\_ipython().system('tar -zcvf Train-model\_new.tgz fruit.h5')

# In[ ]:

ls -1

# # IBM Cloud Deployment Model

# In[ ]:

get\_ipython().system('pip install watson-machine-learning-client --upgrade')

The code handles the crop recommendation model that extracts 30 different features of the dataset and the statistical data to the machine learning model for predicting the legitimacy of the website.

**7.2 Feature 2**

# In[ ]:

client.set.default\_space(space\_uid)

# In[ ]:

client.software\_specifications.list()

# In[ ]:

software\_space\_uid = client.software\_specifications.get\_uid\_by\_name("tensorflow\_rt22.1-py3.9")

software\_spec\_uid

# In[ ]:

ls

# In[ ]:

model\_details = client.repository.store\_model(model= 'Train-model\_new.tgz',

meta\_props={

client.repository.ModelMetaNames.NAME:"CNN",

client.repository.ModelMetaNames.TYPE:"tensorflow\_2.7",

client.repository.ModelMetaNames.SOFTWARE\_SPEC\_UID:software\_space\_uid}

)

# In[ ]:

model\_id = client.repository.get\_model\_id(model\_details)

# In[ ]:

model\_id

# In[ ]:

ls

# # Test The Model

# In[ ]:

import numpy as np

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing import image

# In[ ]:

model=load\_model('fruit.h5')

# In[ ]:

img=image.load\_img(r"C:\Users\Sree Ram\Desktop\ibm\Dataset Plant Disease\fruit-dataset\fruit-dataset\test\Apple\_\_\_healthy\0adc1c5b-8958-47c0-a152-f28078c214f1\_\_\_RS\_HL 7825.JPG")

# In[ ]:

img

# In[ ]:

img=image.load\_img(r"C:\Users\Sree Ram\Desktop\ibm\Dataset Plant Disease\fruit-dataset\fruit-dataset\test\Apple\_\_\_healthy\0adc1c5b-8958-47c0-a152-f28078c214f1\_\_\_RS\_HL 7825.JPG",target\_size=(128,128))

img

# In[ ]:

x=image.img\_to\_array(img)

# In[ ]:

x

# In[ ]:x=np.expand\_dims(x,axis=0)

# In[ ]:

x

# In[ ]:

y=np.argmax(model.predict(x),axis=1)

# In[ ]:

x\_train.class\_indices

# In[ ]:

index=['Apple\_\_\_Black\_rot','Apple\_\_\_healthy','Corn\_(maize)\_\_\_Northern\_Leaf\_Blight','Corn\_(maize)\_\_\_healthy','Peach\_\_\_Bacterial\_spot','Peach\_\_\_healthy']

# In[ ]:

index[y[0]]

# In[ ]:

img=image.load\_img(r"C:\Users\Sree Ram\Desktop\ibm\Dataset Plant Disease\fruit-dataset\fruit-dataset\test\Peach\_\_\_healthy\0a2ed402-5d23-4e8d-bc98-b264aea9c3fb\_\_\_Rutg.\_HL 2471.JPG",target\_size=(128,128))

x=image.img\_to\_array(img)

x=np.expand\_dims(x,axis=0)

y=np.argmax(model.predict(x),axis=1)

index=['Apple\_\_\_Black\_rot','Apple\_\_\_healthy','Corn\_(maize)\_\_\_Northern\_Leaf\_Blight','Corn\_(maize)\_\_\_healthy','Peach\_\_\_Bacterial\_spot','Peach\_\_\_healthy']

index[y[0]]

# import os

# from tensorflow.keras.models import load\_model

# from tensorflow.keras.preprocessing import image

# from flask import Flask,render\_template,request

# app=Flask(\_\_name\_\_)

#

# model=load\_model("fruit.h5")

#

# @app.route('/')

# def index():

# return render\_template("index.html")

#

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

# def upload():

# if request.method=='POST':

# f=request.files['image']

# basepath=os.path.dirname('\_\_file\_\_')

# filepath=os.path.join(basepath,'uploads',f.filename)

# f.save(filepath)

# img=image.load\_img(filepath,target\_size=(128,128))

# x=image.img\_to\_array(img)

# x=np.expand\_dims(x,axis=0)

# pred=np.argmax(model.predict(x),axis=1)

# index=['Apple\_\_\_Black\_rot','Apple\_\_\_healthy','Corn\_(maize)\_\_\_Northern\_Leaf\_Blight','Corn\_(maize)\_\_\_healthy','Peach\_\_\_Bacterial\_spot','Peach\_\_\_healthy']

# text="The Classified Fruit disease is : " +str(index[pred[0]])

# return text

# if \_\_name\_\_=='\_\_main\_\_':

# app.run(debug=False)

# In[ ]:

The code handles the plant disease prediction model that extracts 30 different features of the dataset and the statistical data to the machine learning model for predicting the legitimacy of the website

**8.TESTING**

**8.1TestCases**

# 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 | 7 | 0 | 0 | 7 |
| Client Application | 51 | 0 | 0 | 51 |
| Security | 2 | 0 | 0 | 2 |
| Outsource Shipping | 3 | 0 | 0 | 3 |
| Exception Reporting | 9 | 0 | 0 | 9 |
| Final Report Output | 4 | 0 | 0 | 4 |
| Version Control | 2 | 0 | 0 | 2 |

**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 project at the time of the release to User Acceptance Testing (UAT).

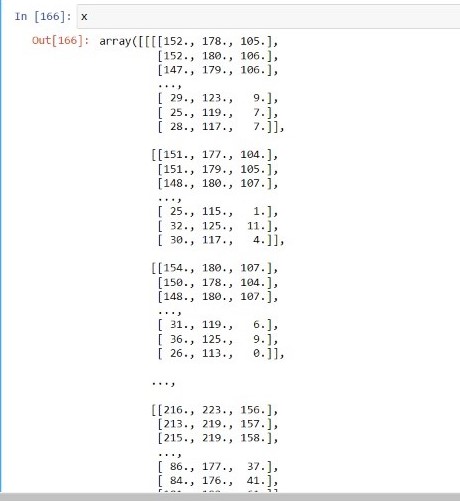
Defect Analysis

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 | 10 | 4 | 2 | 3 | 20 |
| Duplicate | 1 | 0 | 3 | 0 | 4 |
| External | 2 | 3 | 0 | 1 | 6 |
| Fixed | 11 | 2 | 4 | 20 | 37 |
| Not Reproduced | 0 | 0 | 1 | 0 | 1 |
| Skipped | 0 | 0 | 1 | 1 | 2 |
| Won't Fix | 0 | 5 | 2 | 1 | 8 |
| Totals | 24 | 14 | 13 | 26 | 77 |

**9.RESULTS**

**9.1 Performance Matrix**



**10.ADVANTAGES AND DISADVANTAGES**

**ADVANTAGES**

**●** The proposed model here produces very high accuracy ofclassification.

● Very large datasets can also be trained and tested.

● Images of very high can be resized within the proposed itself.

**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**

While our application runs very smoothly, we have several directions in which we can improve our application. Firstly, for crop recommendation and fertilizer recommendation, we can provide the availability of the same on the popular shopping websites, and possibly allow users to buy the crops and fertilizers directly from our application.

(a) Leaf Image Input

(b) LIME Output

Fig. 10: An example of LIME explanation method on AppleCedarRust2 leaf

Another improvement that can be done with fertilizer recommendation is that we want to be able to find data on various brands and items available based on the N,P,K values.

Currently, we only provide six kinds of recommendations, but in future, we want to be able to use complex machine learning systems to provide fifiner recommendations.

**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 vidoe 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**

fruit.ipynb (due to limited page size the code vegetable.ipynb uploaded in github)

#!/usr/bin/env python

# coding: utf-8

# In[1]:

pwd

# In[2]:

cd E:/IBM\_MY\_COURSE/Project/Dataset Plant Disease/fruit-dataset/fruit-dataset

# # Apply ImageDataGenerator functionality to Train and Test set

# # Preprocessing

# In[3]:

from keras.preprocessing.image import ImageDataGenerator

train\_datagen =

ImageDataGenerator(rescale=1./255,shear\_range=0.2,zoom\_range=0.2,horizontal\_fli

p=True)

test\_datagen = ImageDataGenerator(rescale=1)

# In[4]:

pwd

# In[5]:

x\_train = train\_datagen.flow\_from\_directory('E:/IBM\_MY\_COURSE/Project/Dataset

Plant Disease/fruit-dataset/fruit-

dataset/train',target\_size=(128,128),batch\_size=32,class\_mode='categorical')

**#** In[6]:

x\_test=test\_datagen.flow\_from\_directory('E:/IBM\_MY\_COURSE/Project/Dataset Plant

Disease/fruit-dataset/fruit-dataset/test',target\_size=(128,128),

batch\_size=32,class\_mode='categorical')

# # Import the models

# In[7]:

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense,Convolution2D,MaxPool2D,Flatten

# # Initializing the models

# In[8]:

model=Sequential()

# # Add CNN Layers

# In[9]:

model.add(Convolution2D(32,(3,3),input\_shape=(128,128,3),activation='relu'))

# In[10]:

x\_train.class\_indices

# # Add Pooling layer

# In[11]:

model.add(MaxPool2D(pool\_size=(2,2)))

# # Add Flatten layer

# In[12]:

model.add(Flatten())

# # Add Dense Layer

# In[21]:

model.add(Dense(40, kernel\_initializer='uniform',activation='relu'))

model.add(Dense(20, kernel\_initializer='random\_uniform',activation='relu'))

# # Add Output Layer

# In[24]:

model.add(Dense(6,activation='softmax', kernel\_initializer='random\_uniform'))

# # Compile the model

# In[25]:

model.compile(loss='categorical\_crossentropy',optimizer='adam',metrics=['accuracy'

])

# In[26]:

len(x\_train)

# In[27]:

5384/32

# # Fit the Model

# In[28]:

model.fit\_generator(x\_train,steps\_per\_epoch=168,validation\_data=x\_test,validation\_st

eps=52,epochs=3)

# # Save the Model

# In[29]:

model.save("fruit.h5")

# In[30]:

ls

# # Test the Model

# In[32]:

from keras.preprocessing import image

from tensorflow.keras.preprocessing.image import img\_to\_array

from tensorflow.keras.models import load\_model

import numpy as np

# In[33]:

model = load\_model("fruit.h5")

# # Test Apple\_Healthy Class images

# In[37]:

img = image.load\_img('E:/IBM\_MY\_COURSE/Project/Dataset Plant Disease/fruit-

dataset/fruit-dataset/test/Apple\_\_\_healthy/00fca0da-2db3-481b-b98a-

9b67bb7b105c\_\_\_RS\_HL 7708.JPG',target\_size=(128,128))

# In[39]:

x=image.img\_to\_array(img)

x=np.expand\_dims(x,axis=0)

# In[40]:

pred = model.predict\_classes(x)

# In[41]:

pred

# In[45]:

index

=['Apple\_\_Black\_rot','Apple\_healthy','Corn(maize)\_\_Northern\_Leaf\_Blight','Corn(

maize)\_\_healthy','Peach\_Bacterial\_spot','Peach\_\_healthy']

# In[46]:

print('the given image belogs to=',index[pred[0]])

# # Test Apple Black Rot class images

# In[54]:

img = image.load\_img('E:/IBM\_MY\_COURSE/Project/Dataset Plant Disease/fruit-

dataset/fruit-dataset/test/Apple\_\_\_Black\_rot/0f3d45f4-e121-42cd-a5b6-

be2f866a0574\_\_\_JR\_FrgE.S 2870.JPG',target\_size=(128,128))

# In[55]:

x=image.img\_to\_array(img)

x=np.expand\_dims(x,axis=0)

pred = model.predict\_classes(x)

print('the given image belogs to=',index[pred[0]])

# # Test Corn Northern leaf Blight class images

# In[56]:

img = image.load\_img('E:/IBM\_MY\_COURSE/Project/Dataset Plant Disease/fruit-

dataset/fruit-dataset/test/Corn\_(maize)\_\_\_Northern\_Leaf\_Blight/00a14441-7a62-

4034-bc40-b196aeab2785\_\_\_RS\_NLB 3932.JPG',target\_size=(128,128))

# In[57]:

x=image.img\_to\_array(img)

x=np.expand\_dims(x,axis=0)

pred = model.predict\_classes(x)

print('the given image belogs to=',index[pred[0]])

# # Test Corn Healthy class images

# In[58]:

img = image.load\_img('E:/IBM\_MY\_COURSE/Project/Dataset Plant Disease/fruit-

dataset/fruit-dataset/test/Corn\_(maize)\_\_\_healthy/0a68ef5a-027c-41ae-b227-

159dae77d3dd\_\_\_R.S\_HL 7969 copy.jpg',target\_size=(128,128))

# In[59]:

x=image.img\_to\_array(img)

x=np.expand\_dims(x,axis=0)

pred = model.predict\_classes(x)

print('the given image belogs to=',index[pred[0]])

# # Test Peach Bacterial spot class images

# In[60]:

img = image.load\_img('E:/IBM\_MY\_COURSE/Project/Dataset Plant Disease/fruit-

dataset/fruit-dataset/test/Peach\_\_\_Bacterial\_spot/00ddc106-692e-4c67-b2e8-

569c924caf49\_\_\_Rutg.\_Bact.S 1228.JPG',target\_size=(128,128))

# In[61]:

x=image.img\_to\_array(img)

x=np.expand\_dims(x,axis=0)

pred = model.predict\_classes(x)

print('the given image belogs to=',index[pred[0]])

# # Test Peach Healthy class images

# In[62]:

img = image.load\_img('E:/IBM\_MY\_COURSE/Project/Dataset Plant Disease/fruit-

dataset/fruit-dataset/test/Peach\_\_\_healthy/1a07ce54-f4fd-41cf-b088-

144f6bf71859\_\_\_Rutg.\_HL 3543.JPG',target\_size=(128,128))

# In[63]:

x=image.img\_to\_array(img)

x=np.expand\_dims(x,axis=0)

pred = model.predict\_classes(x)

print('the given image belogs to=',index[pred[0]])

**GitHub and Project Demo Link**

https://drive.google.com/file/d/1jIBBomc2kV85pqjzYT6kvd6OY\_qAiq69/view?usp=share\_link

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |
| --- |
|  |

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |
| --- |
|  |

|  |  |
| --- | --- |
| |  | | --- | |  | |

### 