Fertilizers Recommendation System for Disease Prediction

Submitted by
Team ID : PNT2022TMID09294

MANICK SRIRAM M	**REG : 310619104065**
RAJAVARMAN R	**REG : 310619104099**
RANJITH G	**REG : 310619104101**
SAMRISH R	**REG : 310619104114**

1 . INTRODUCTION :

Overview Two datasets—the fruit dataset and the vegetable dataset—are gathered for this research. Convolutional Neural Networks, a deep learning neural network, is used to train and test the datasets that has been collected (CNN). The fruit dataset is first trained, and then CNN is tested. There are 6 courses total, and each class is trained and tested. The vegetable dataset is then tested and trained.
Python is the language used to train and test datasets.

All of the Python code is initially created in the Jupyter notebook that comes with Anaconda Python, and it is then tested in the IBM cloud. Finally, Flask, a Python package, is used to construct a web-based framework. Along with their related files, two html files are created in the templates folder.

Purpose of this study is used to test samples of fruits and vegetables and find out which diseases they may have. Additionally, this project suggests fertilizer for certain ailments**.**

2 . LITERATURE SURVEY :

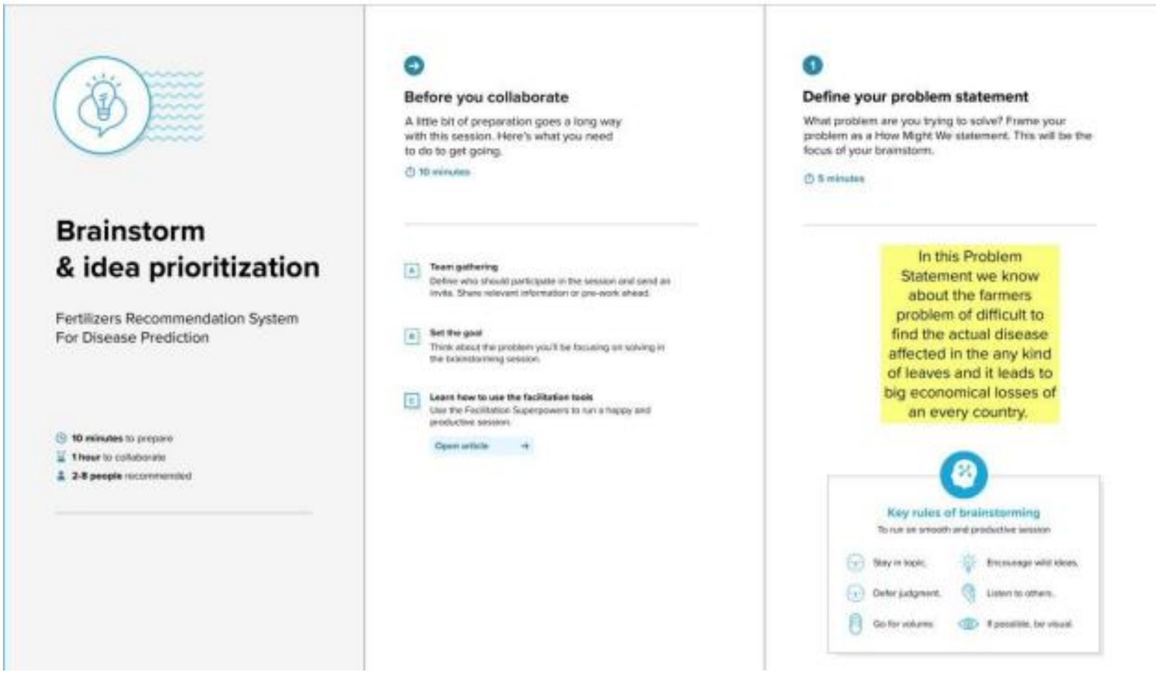
Existing issue our Team suggested a method for identifying leaf diseases and suggested fertilizer to treat them. However, the method's low number of train and test sets leads to subpar accuracy.
In order to recommend soil-based fertilizers for anticipated crop diseases, our team put up a straightforward prediction method. This approach offers less predictability and accuracy. An IOT-based system for leaf disease identification and fertilizer prescription that was proposed by us, which uses machine learning techniques and achieves accuracy levels of under 80%.

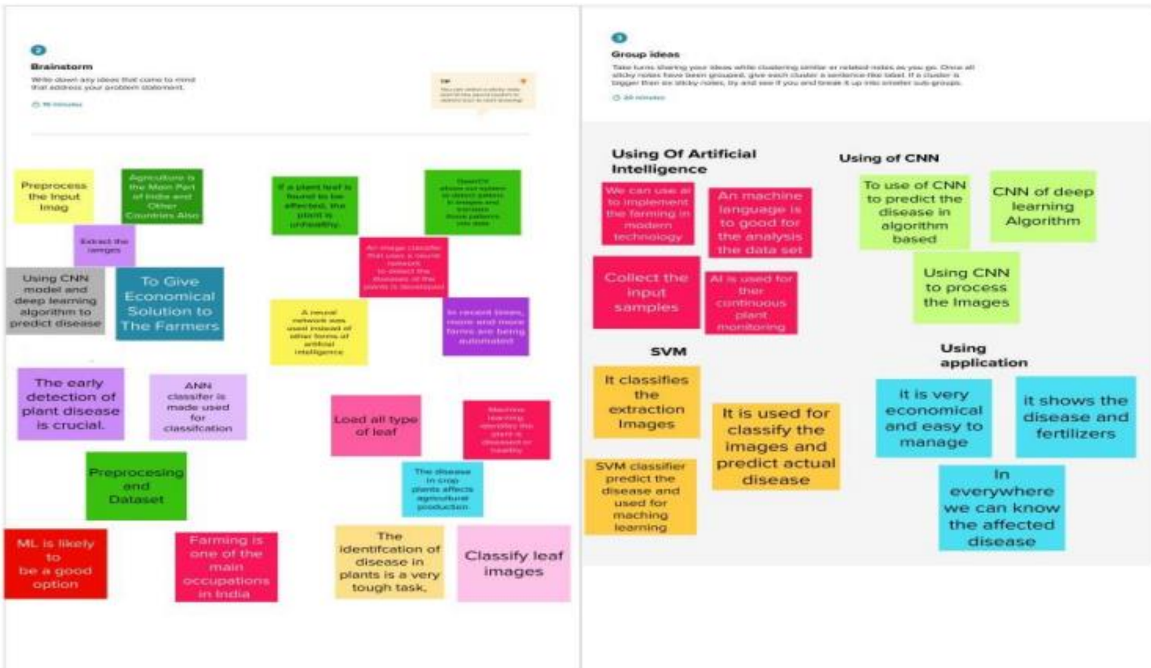
Proposed Remedy A deep learning-based neural network is implemented in this project's effort to train and test the datasets that were gathered. CNN, a deep learning-based neural network, provides classification accuracy rates of greater than 90%. By boosting the accuracy rate can be raised to 95% to 98% by adding more dense layers and changing hyperparameters like the number of epochs and batch size.

3 . IDEATION AND PROPOSED SYSTEM:

EMPATHY MAP:

IDEATION AND BRAINSTORMING:





PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Farmers' conventional methods of agricultural cultivation are ineffective. It does not make proper use of all available resources. Farmers are unable to detect crop diseases due to a lack of knowledge and old practices, which often result in soil nutrient deterioration and exhaustion. As a result, crop failure occurs. Growing only certain crops depletes the soil, and if the crops are harmed by illnesses, farmers are uninformed of how to recover such crops. Food needs cannot be met until and unless efficient resource management and use is implemented.
2.	Idea / Solution description	Efficient approach for controlling the overuse of insecticides and fertilizers in farming. Implementation of artificial intelligence for identification of pests and recommendation of insecticides using TPF-CNN.
3.	Novelty / Uniqueness	The proposed method uses SVM to classify tree leaves, identify the disease and suggest the fertilizer. The proposed method is compared with the existing CNN based leaf disease prediction. The proposed SVM technique gives a better result when compared to existing CNN.
4.	Social Impact / Customer Satisfaction	It also helps farmer to perform the activities like crop management including applications on yield prediction, disease detection, weed detection, crop quality, and growth prediction etc. This chapter describes the case study on “Crop Disease Detection and Yield prediction”. The study includes identification of crop Agriculture is the mainstay of a rising economy in India. condition, disease detection, prediction about specific crop and recommendation using machine learning algorithms. It gives an idea about how recommender system is used in agriculture for disease detection and prediction.

5.	Business Model (Revenue Model)	Being an extremely vital industry as it manufactures some of the most important raw materials required for crop production, it is not wrong to say that the success of the agricultural sector in India is largely dependent on the fertilizer industry. Fertilizers are extensively being used to improve per hectare production of crops that can be used for food and industrial applications. If you like the idea of making a profit by helping people work with the soil, you might enjoy being a part of the fertilizer industry.
6.	Scalability of the Solution	Fertilizers replace the nutrients that crops remove from the soil. Without the addition of fertilizers, crop yields and agricultural productivity would be significantly reduced. That's why mineral fertilizers are used to supplement the soil's nutrient stocks with minerals that can be quickly absorbed and used by crops.

PROBLEM SOLUTION FIT:

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Farmers are our customer

6. CUSTOMER CONSTRAINTS CC

Network connection, Minimum Spending time, Details for the previous soil test data.

5. AVAILABLE SOLUTIONS AS

- Easy to work done,
- Farmer Easily understand the application interface,
- Searching to get result is similar result showing former little difficult.
- Upload the details clearly.

Explore AS, differentiate

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

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

To find the solution to the farming queries, to getting the image of the plant or getting the farming land information to give the solution.

9. PROBLEM ROOT CAUSE RC

User no need to go any other places to using Mobile or Desktop browser to get solution to the agricultural queries.

7. BEHAVIOUR BE

- To cure the plant disease.
- To use Fertilizer to give the stable condition for the soil Nutarians.

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

Identify strong TR & EM

3. TRIGGERS TR

Anytime access, free of cost, work efficiency, find the disease.

4. EMOTIONS: BEFORE / AFTER EM

Farmers need to deal with many problems cope with climate change, soil Erosion and biodiversity loss. Farmers didn't know all the information about farming techniques and disease.

Get disease for plant to find the problem with solution to the disease Case.

10. YOUR SOLUTION SL

To get the image and also agricultural land information to recommend the Fertilizers .

Also give the disease prediction ideas .

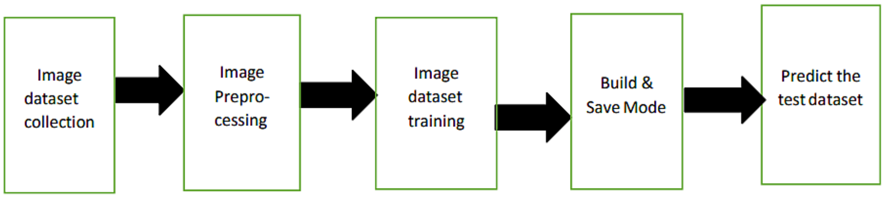
It fit the agricultural quires to improve the growth of the crops.

8. CHANNELS of BEHAVIOUR CH

Online

Identify strong TR & EM

4.REQUIREMENT ANALYSIS :



Block Diagram of the project

Above Diagram represents the project's overall block diagram. The collecting of picture datasets comes first, then enters image preparation. The training of picture datasets using various hyperparameter initializations is the third phase. After that, create the model and save it in ".h5 format". The final step involves applying the trained model to test new or existing datasets.

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
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Profile	Filling the profile page after logging in
FR-4	Uploading Data (Leaf)	Image of the leaves is to be uploaded
FR-5	Requesting solution	Uploaded image is compared with the pre-defined model and solution is generated.
FR-6	Fertilizer Recommendation	Based on the type of disease identified, suitable fertilizers are recommended.

Non-Functional Requirement :

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system allows the user to perform the task easily, efficiently and effectively.
NFR-2	**Security**	Information about the user and their data's are highly secured with the authorization technology
NFR-3	**Reliability**	The model deployed should be reliable and able to give accurate disease prediction and recommendation.
NFR-4	**Performance**	Response time and total processing time is fast.
NFR-5	**Availability**	The application should be available anytime and anywhere to all the registered users.
NFR-6	**Scalability**	Increase in the number of user does not affect the performance of the system.

Hardware / Software Designing :

Python is the language used to train and test the dataset. Python programming is done in a notebook tool called Jupyter, which also works with the IBM cloud.
Convolutional Neural Network is the neural network that was utilised to train and test the model (CNN).

The following Python libraries need to be imported before beginning the process in order to perform the aforementioned actions in Python :

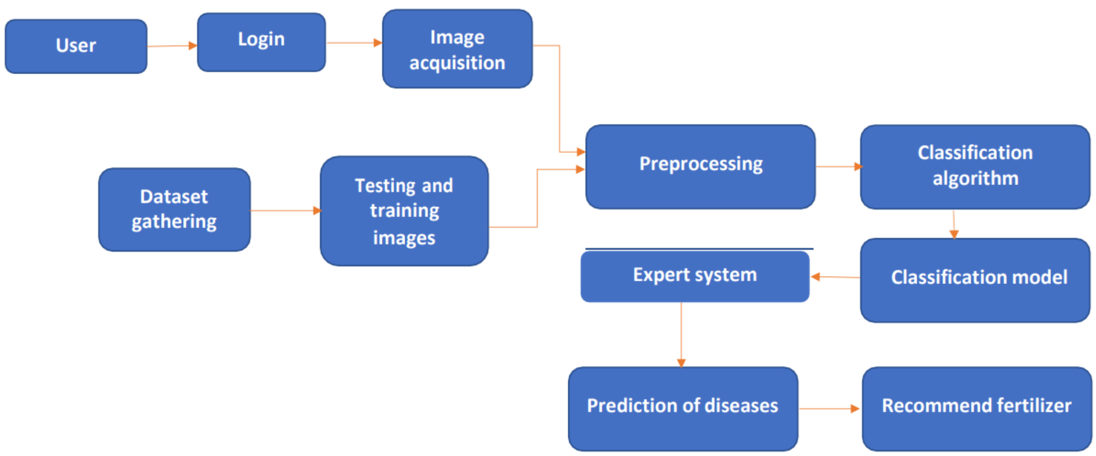
- NumPy,
- TensorFlow,
- Keras,
- Matplotlib (optional for data visualization).

The following activation functions used in the CNN training :

- RELU at the end of convolution layer and Max Pool layer,
- SoftMax at the end of output dense layer,
- For testing the dataset argmax is used, its an optional.

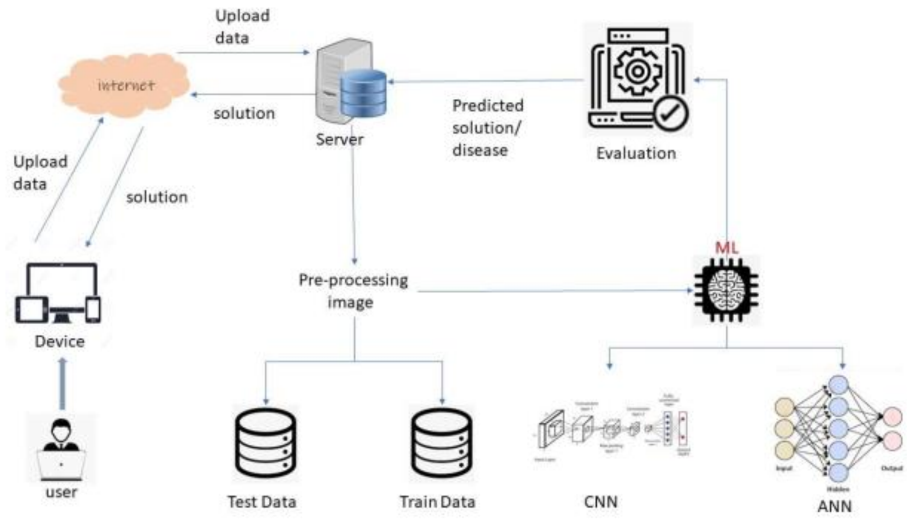
5 .PROJCT DESIGN:

Data Flow Diagram:

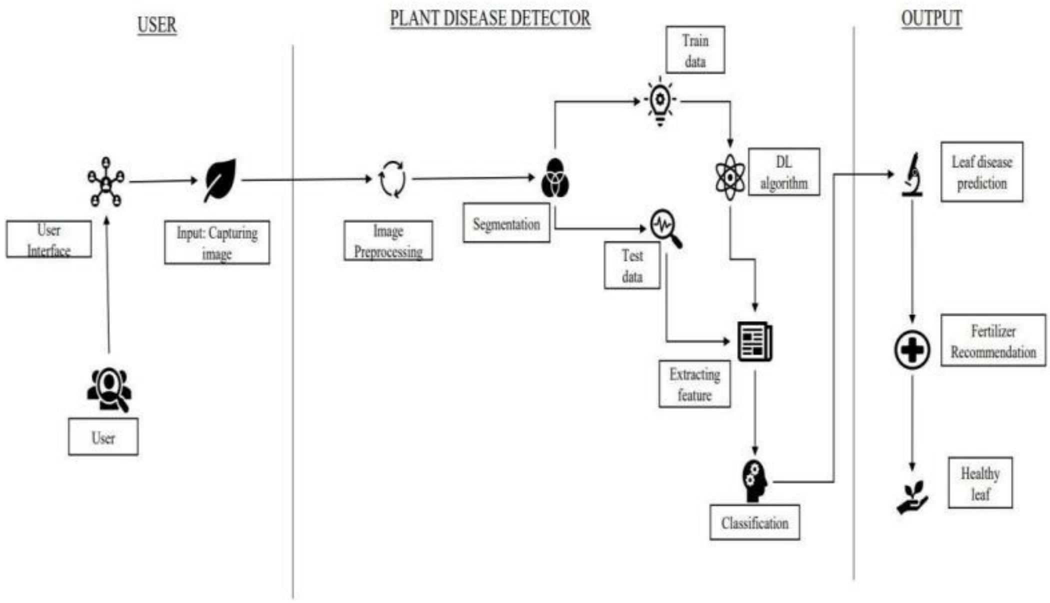


Solution and Technical Architecture:

Solution architecture:



Technical architecture:



User Stories:

User Stories

Use the below template to list all the user stories for the product.

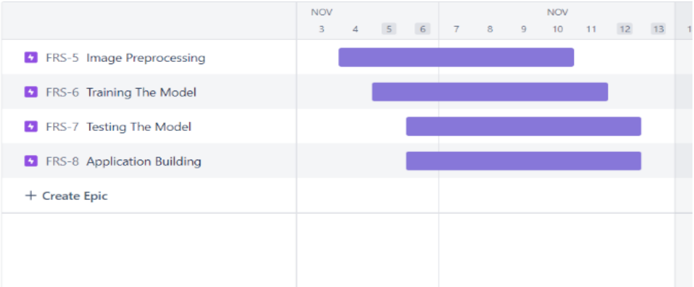
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-1	As a user, I can see my dashboard and go through the functions provided by the system.	I can access my dashboard	High	Sprint-1
Customer (Web user)	Registration		As a user, I can register for my account through web and login to my web page.			
Customer Care Executive	Login	USN-1	Make a call to the customer care executive and rectify the queries.	Help the user how to access the system.	High	Sprint-1
Administrator	User account control	USN-1	Responsible for carrying out the administration process.	Manage the total team	High	Sprint-1

6.PROJECT PLANNING AND SCHEDULING

Sprint Planning and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Collecting dataset for pre-processing	10	High	M.MANICK SRIRAM R.RAJAVARMAN G.RANJITH R.SAMRISH
Sprint-1		USN-2	Data pre-processing-Used to transform the data into useful format.	10	Medium	M.MANICK SRIRAM R.RAJAVARMAN G.RANJITH R.SAMRISH
Sprint-2	Model Building	USN-3	Model building for fruit and vegetable disease prediction	10	High	M.MANICK SRIRAM R.RAJAVARMAN G.RANJITH R.SAMRISH
Sprint-2		USN-4	Splitting the data into training and testing from the entire dataset.	10	Medium	M.MANICK SRIRAM R.RAJAVARMAN G.RANJITH R.SAMRISH
Sprint-3	Training and Testing	USN-5	Training the model and testing the performance of the model	20	Medium	M.MANICK SRIRAM R.RAJAVARMAN R.SAMRISH G.RANJITH
Sprint-4	Implementation of Web page	USN-6	Implementing the web page for collecting the data from user	10	High	M.MANICK SRIRAM R.RAJAVARMAN R.SAMRISH G.RANJITH
Sprint-4		USN-6	Deploying the model using IBM Cloud and IBM Watson Studio	10	Medium	M.MANICK SRIRAM R.RAJAVARMAN R.SAMRISH G.RANJITH

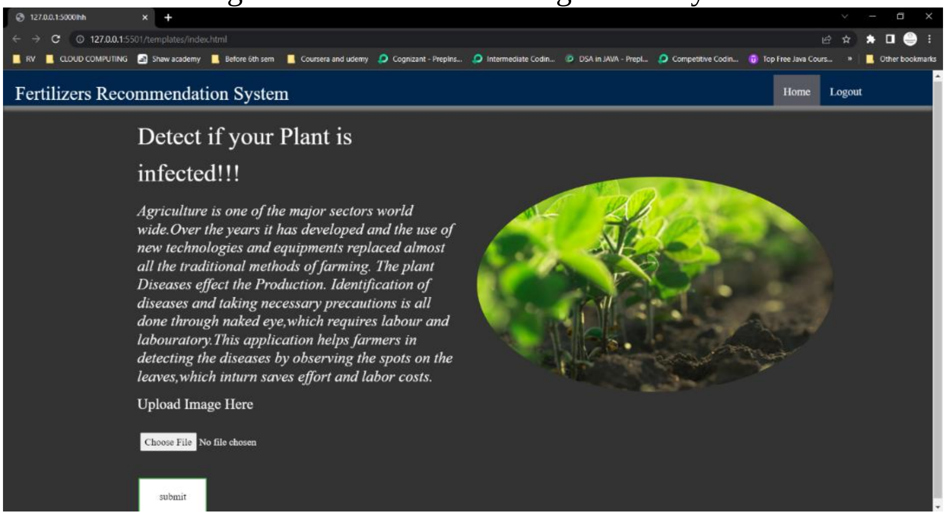
Results From JIRA :



7.CODING AND SOLUTIONING:

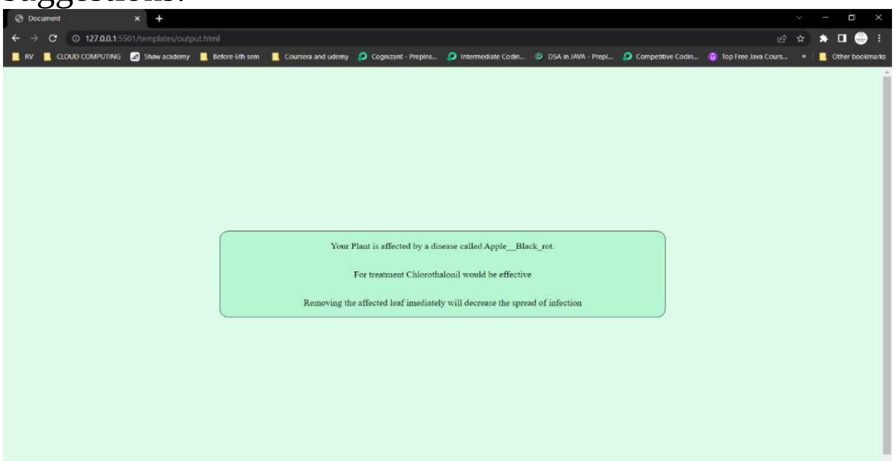
Feature 1 :

We create a website to analyse and identify the affected plants even a person with minimal knowledge in software and farming can easily access and understand.



Feature 2 :

Also in our project we are suggesting them a Remedy by providing fertilizer suggestions.



8.TESTING :
User Acceptance Testing:

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing.

2. Defect Analysis

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

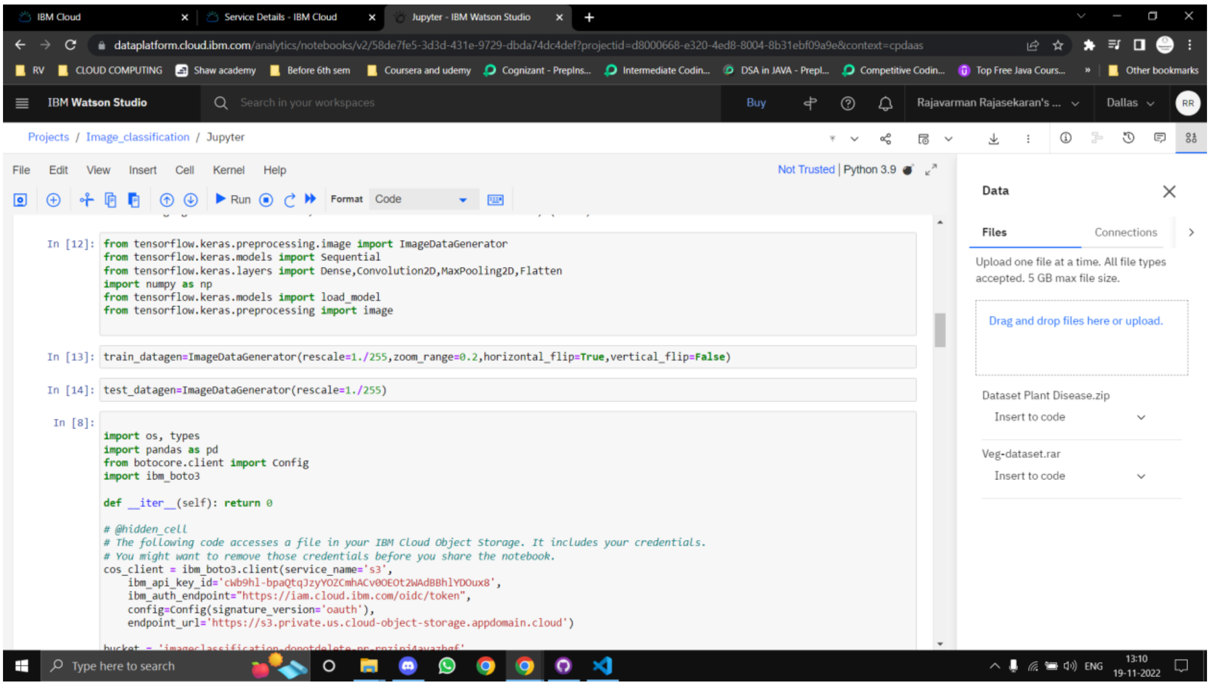
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	19
Duplicate	2	0	2	0	4
External	2	0	0	0	2
Fixed	10	3	2	1	16
Not Reproduced	0	1	0	0	1
Skipped	0	0	1	1	2
Won't Fix	0	0	0	0	0
Totals	24	8	7	5	44

3. Test Case Analysis

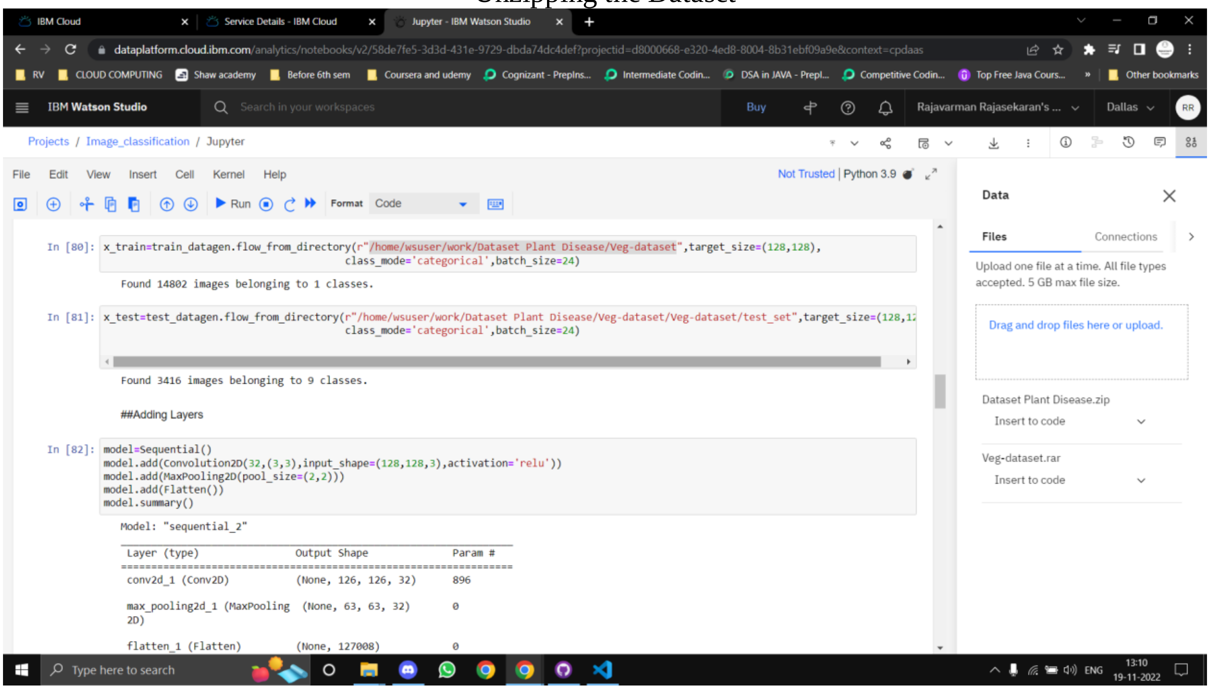
This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	43	0	0	43
Security	5	0	0	5
Outsource Shipping	4	0	0	4
Exception Reporting	10	0	0	10
Final Report Output	6	0	0	6
Version Control	2	0	0	2

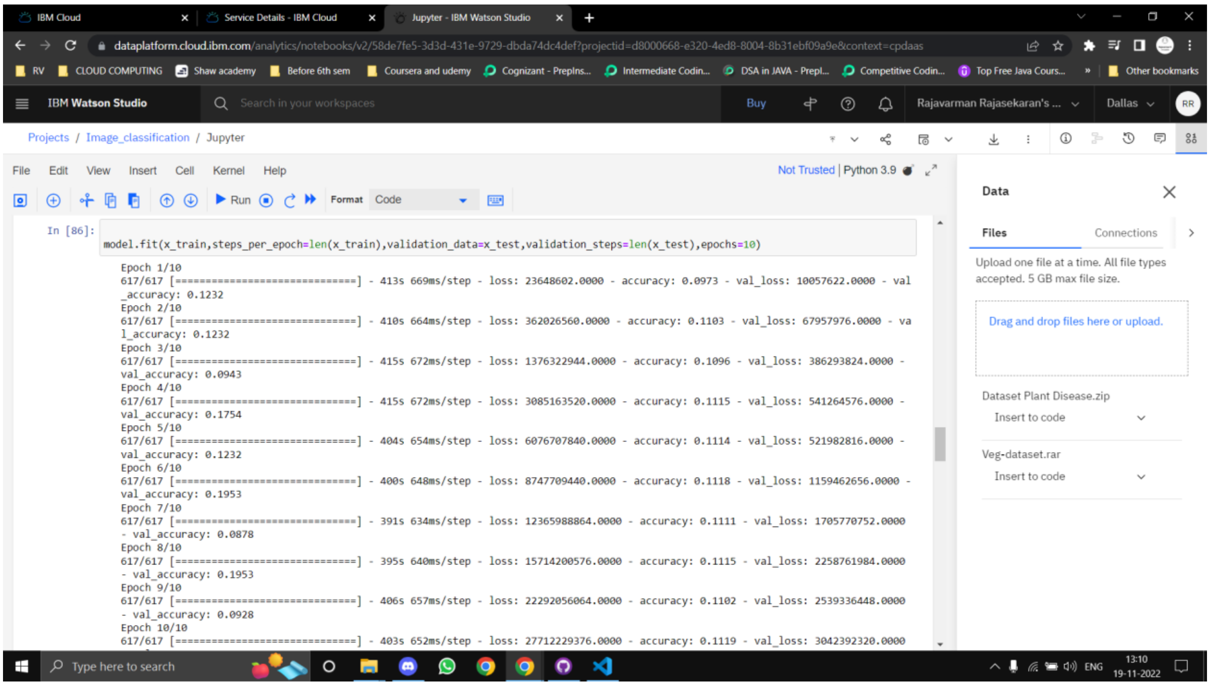
9.RESUTS :



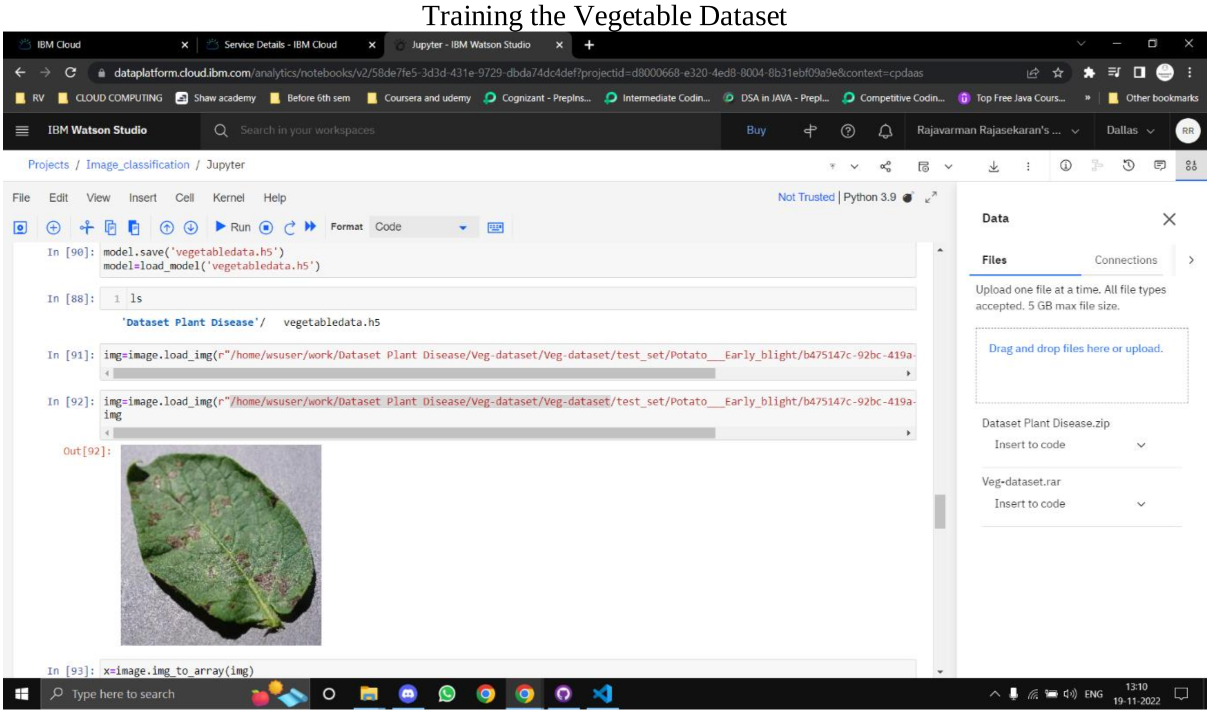
Unzipping the Dataset



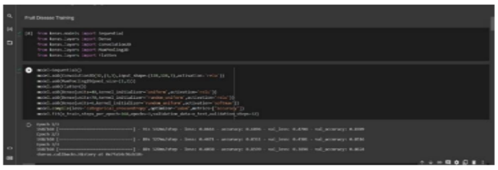
Training the Output Dataset

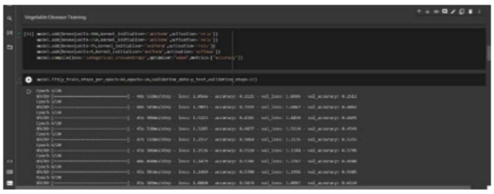


Training the Vegetable Dataset



Performance Metrics:

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	Fruit Summary  Vegetable Summary 
2.	Accuracy	Training Accuracy - Validation Accuracy -	Fruit Accuracy 

	Parameter	Values	Screenshot
2.	Accuracy	Training Accuracy - Validation Accuracy -	Vegetable Accuracy 

3.	Confidence Score (Only Yolo Projects)	Class Detected - Confidence Score -	Fruit Score  Vegetable Score 

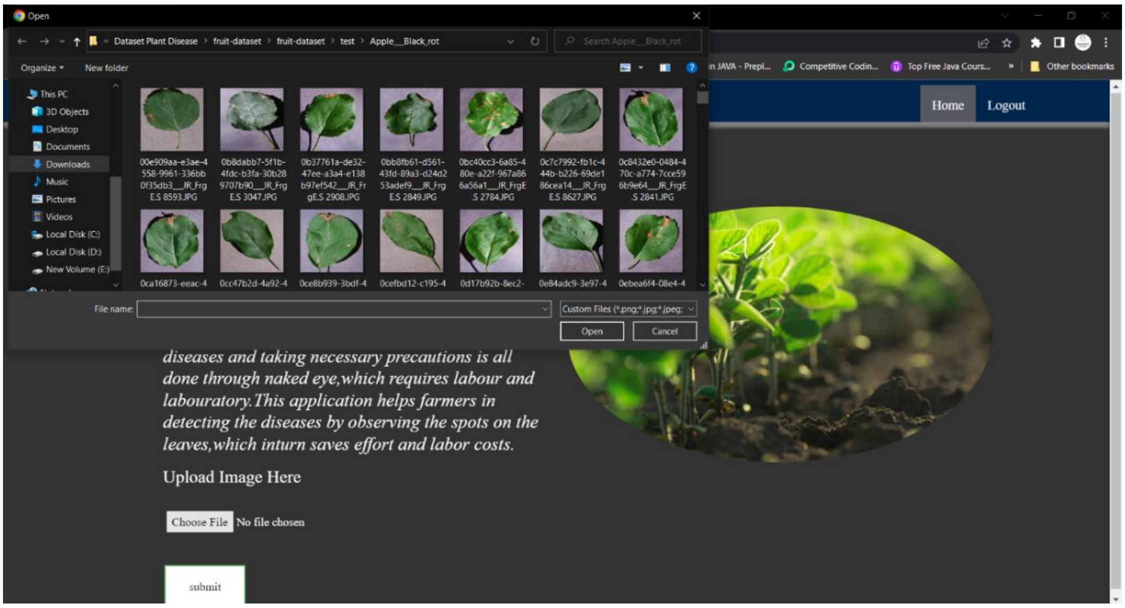
10 . ADVANTAGES :

- ➤ The model that is being suggested here achieves extremely high categorization accuracy.
- ➤ It is also possible to train and test very big datasets.
- ➤ Very high resolution images can be modified within the proposal itself.

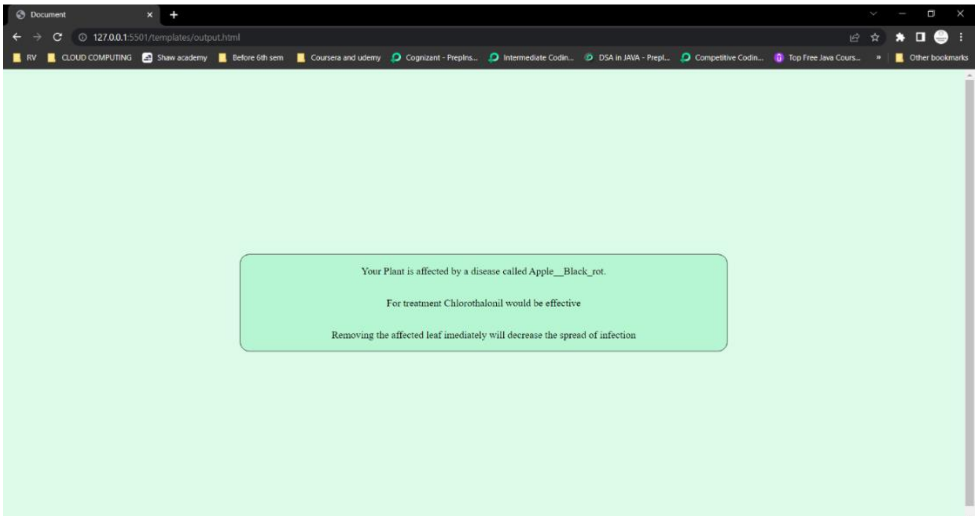
DIS - ADVANTAGES :

- ➤ The suggested model demands a significant amount of computational time for both training and testing.
- ➤ This project's neural network design is quite complicated.

11. CONCLUSION :



Uploading the Image



Output

The model proposed here involves classifying images from datasets of fruits and vegetables. Observations made during model testing and training include the following:

- ➤ The accuracy of classification increased by increasing the number of epochs.
- ➤ For different batch sizes, different classification accuracies are obtained.
- ➤ The accuracies are increased by increasing more convolution layers.
- ➤ The accuracy of classification also increased by varying dense layers.

- ➤ Different accuracies are obtained by varying the size of kernel used in the convolution layer output.
- ➤ Accuracies are different while varying the size of the train and test datasets.

12 . FUTURE SCOPE :

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

13.APPENDIX:

FRUIT MODEL TESTING:

ADD CNN LAYERS:

```
model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.save('fruit.h5')
model.summary()
```

ADD DENSE LAYERS :

```
model.add(Dense(300,kernel_initializer='uniform',activation='relu'))
model.add(Dense(150,kernel_initializer='random_uniform',activation='relu'))
model.add(Dense(9,kernel_initializer='random_uniform',activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
len(x_train)

len(x_test)

model.fit(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test),
epochs=10)
```

IMPORT THE LIBRARIES :

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
import numpy as np
from tensorflow.keras.models import load_model
```

INITIALIZING THE MODEL:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
from tensorflow.keras.models import load_model

train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_fli
p=True)

x_train=train_datagen.flow_from_directory(r"D:/Rajavarman/Documents/GitHub/IBM-Project-
37872-1660358144/Dataset Plant Disease/fruit-dataset/fruit-dataset/train",target_size=(128,128),
                                   class_mode='categorical',batch_size=24)


x_test=test_datagen.flow_from_directory(r'D:/Rajavarman/Documents/GitHub/IBM-Project-37872-
1660358144/Dataset Plant Disease/fruit-dataset/fruit-dataset/test',target_size=(128,128),
                                   class_mode='categorical',batch_size=24)
```

TEST AND SAVE THE MODEL:

```
model.save('fruitdata.h5')
 model=load_model('fruitdata.h5')

img=image.load_img(r"D:/Rajavarman/Documents/GitHub/IBM-Project-37872-1660358144/Dataset
Plant Disease/fruit-dataset/fruit-dataset/test/Apple___healthy/01b32971-5125-4982-98e2-
22daa9ae864a___RS_HL 7977.JPG")

x=image.img_to_array(img)

img=image.load_img(r"D:/Rajavarman/Documents/GitHub/IBM-Project-37872-1660358144/Dataset
Plant Disease/fruit-dataset/fruit-dataset/test/Apple___healthy/01b32971-5125-4982-98e2-
22daa9ae864a___RS_HL 7977.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]]
```

VEGETABLE MODEL TESTING:

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image



train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,v
ertical_flip=False)

test_datagen=ImageDataGenerator(rescale=1./255)

x_train=train_datagen.flow_from_directory(r"D:/Rajavarman/Documents/GitHub/IBM-Project-
37872-1660358144/Dataset Plant Disease/Veg-dataset",target_size=(128,128),
                                         class_mode='categorical',batch_size=24)

x_test=test_datagen.flow_from_directory(r"D:/Rajavarman/Documents/GitHub/IBM-Project-
37872-1660358144/Dataset Plant Disease/Veg-dataset/Veg-
dataset/test_set",target_size=(128,128),
                                         class_mode='categorical',batch_size=24)


model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.summary()

model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(9,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])



model.fit(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(
x_test),epochs=10)

model.save('vegetabledata.h5')
model=load_model('vegetabledata.h5')

img=image.load_img(r"D:/Rajavarman/Documents/GitHub/IBM-Project-37872-
1660358144/Dataset Plant Disease/Veg-dataset/Veg-
dataset/test_set/Potato___Early_blight/b475147c-92bc-419a-b2c3-
7d5aabbb79ec___RS_Early.B 7379.JPG")

img=image.load_img(r"D:/Rajavarman/Documents/GitHub/IBM-Project-37872-
1660358144/Dataset Plant Disease/Veg-dataset/Veg-
```

```
dataset/test_set/Potato___Early_blight/b475147c-92bc-419a-b2c3-
7d5aabbb79ec___RS_Early.B 7379.JPG")

x=image.img_to_array(img)
img=image.load_img(r"D:/Rajavarman/Documents/GitHub/IBM-Project-37872-
1660358144/Dataset Plant Disease/Veg-dataset/Veg-
dataset/test_set/Potato___Early_blight/b475147c-92bc-419a-b2c3-
7d5aabbb79ec___RS_Early.B 7379.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=['Pepper,_bell___Bacterial_spot','Pepper,_bell_
__healthy','Potato___Early_blight','Potato___Late_blight','Potato___healthy','Tomato___Bac
terial_spot','Tomato___Late_blight','Tomato___Leaf_Mold','Tomato___Septoria_leaf_spot']
index[y[0]]
```

Drive link:

https://drive.google.com/file/d/1m4y0HoSF3wz9CBrR-3O2sU0SlC2urIUT/view?usp=sharing

Github:

https://github.com/IBM-EPBL/IBM-Project-37872-1660358144/tree/8abff428a0680696dc6e759e10e5076d26602141