Fertilizers Recommendation System for Disease Prediction

PROJECT REPORT

Submitted by

Team ID:PNT2022TMID15357

RISHIKA.R (Team leader)

RAJALA TEJASWI

RESHMA.P

POONKUZHALI.M.C

In partial fulfilment for the award of the degree Of

BACHELOR OF ENGINEERING in COMPUTER SCIENCE AND ENGINEERING

R.M.D ENGINEERING COLLEGE,

KAVARAIPETTAI,TIRUVALLUR.

CONTENT

1. INTRODUCTION

1.1. Project Overview

In this project, two datasets name fruit dataset and vegetable dataset are collected. The collected datasets are trained and tested with deep learning neural network named Convolutional Neural Networks (CNN). First, the fruit dataset is trained and then tested with CNN. It has 6 classes and all the classes are trained and tested. Second, the vegetable dataset is trained and tested. The software used for training and testing of datasets is Python. All the Python codes are first written in Jupyter notebook supplied along with Anaconda Python and then the codes are tested in IBM cloud. Finally, a web-based framework is designed with help Flask a Python library. There are 2 html files are created in templates folder along with their associated files in static folder. The Python program 'app.py' used to interface with these two webpages is written in Spyder-Anaconda python and tested.

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.

2. LITERATURE SURVEY

2.1. Existing problem

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

2.2. References

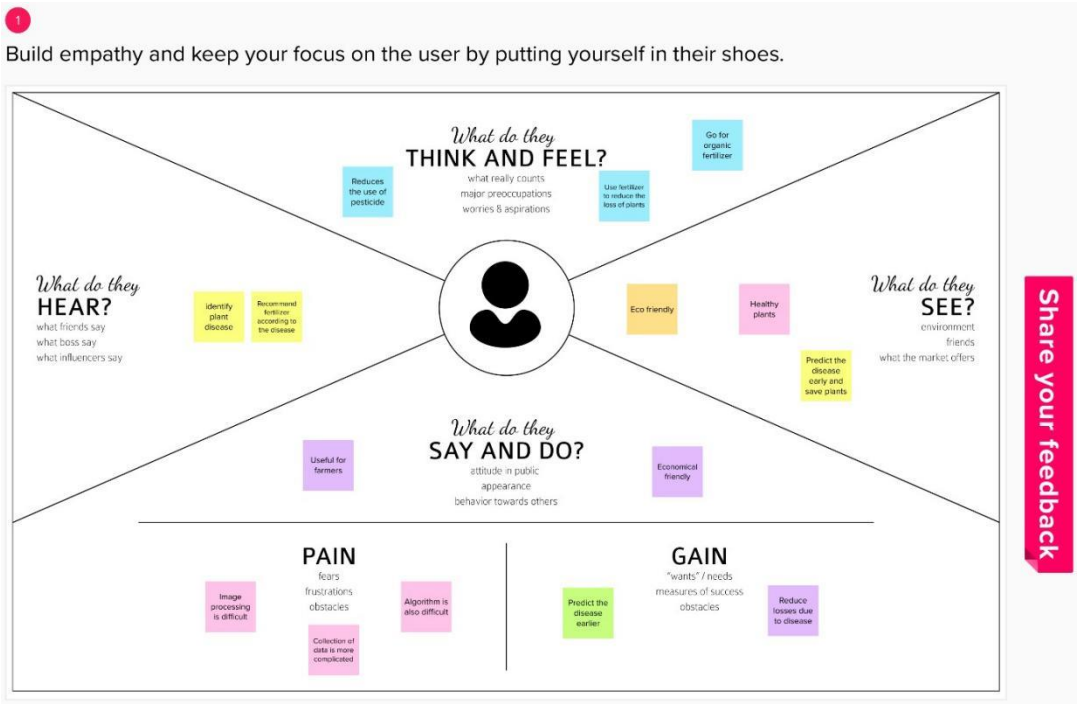
- Leaf Disease Detection and Fertilizer Suggestion
- Plant Disease Detection and Classification using CNN Model with Optimized Activation Function
- Crop leaf disease detection using machine learning algorithm

2.3. Problem Statement Definition

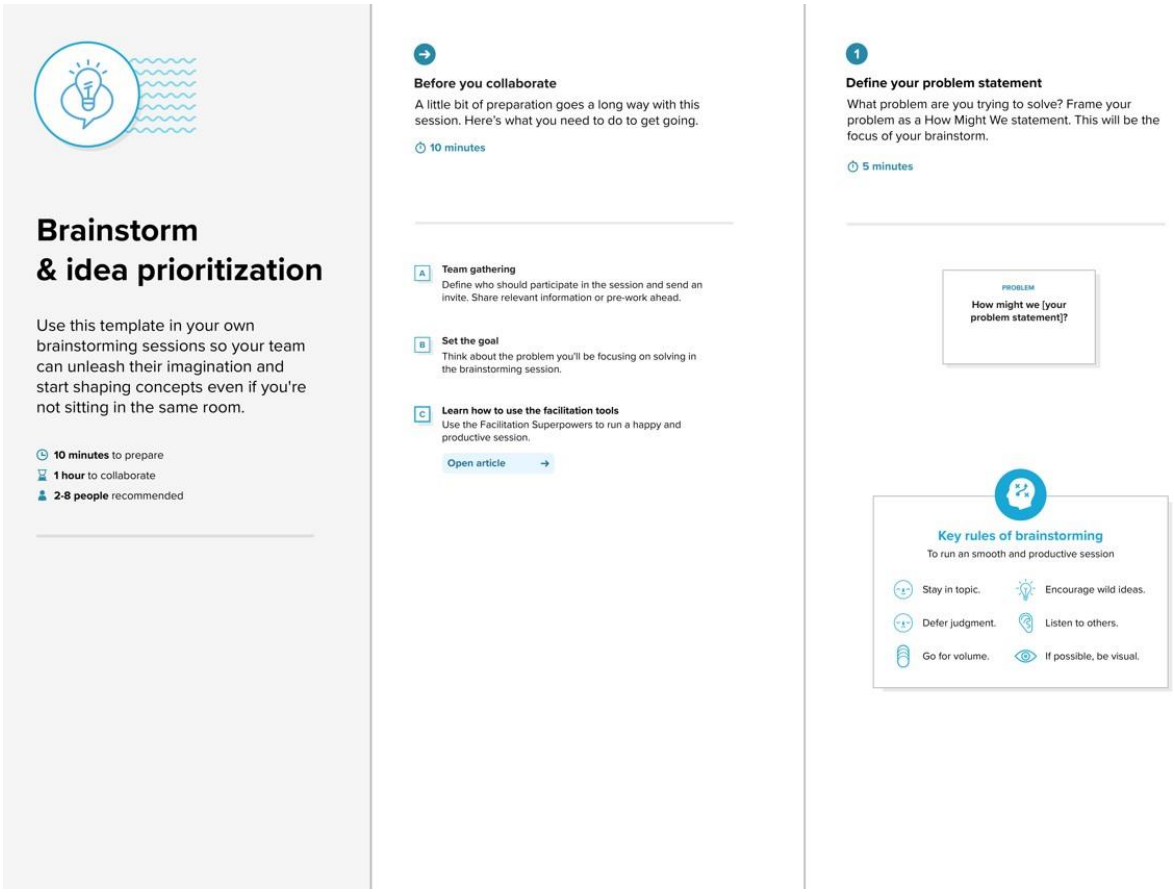
Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Farmers	Cultivate healthy crops	It is not possible	It is affected by disease and insufficient fertilizer	Frustrated and Economically weak
PS-2	Local people	Buy healthy and organic vegetables at low cost	The price of food products are high and the quality is low	Crops are affected by disease	Frustrated

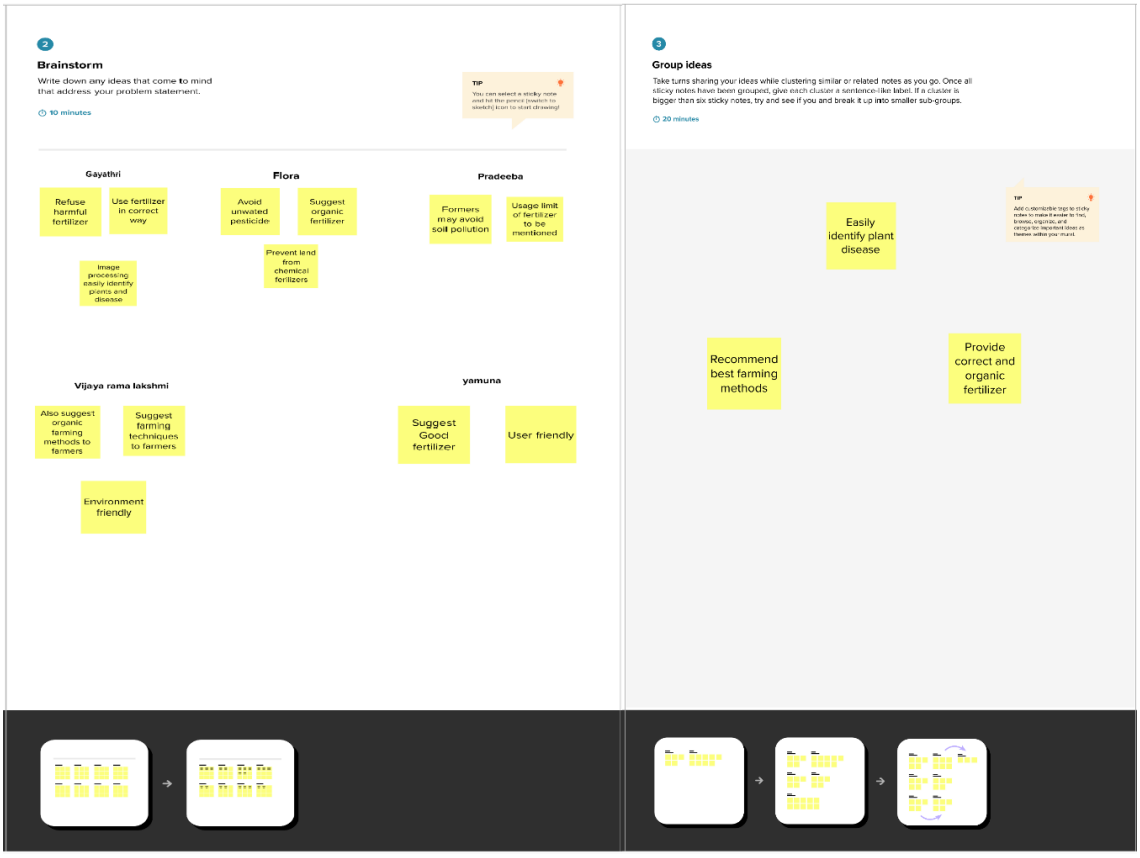
3. IDEATION & PROPOSED SOLUTION

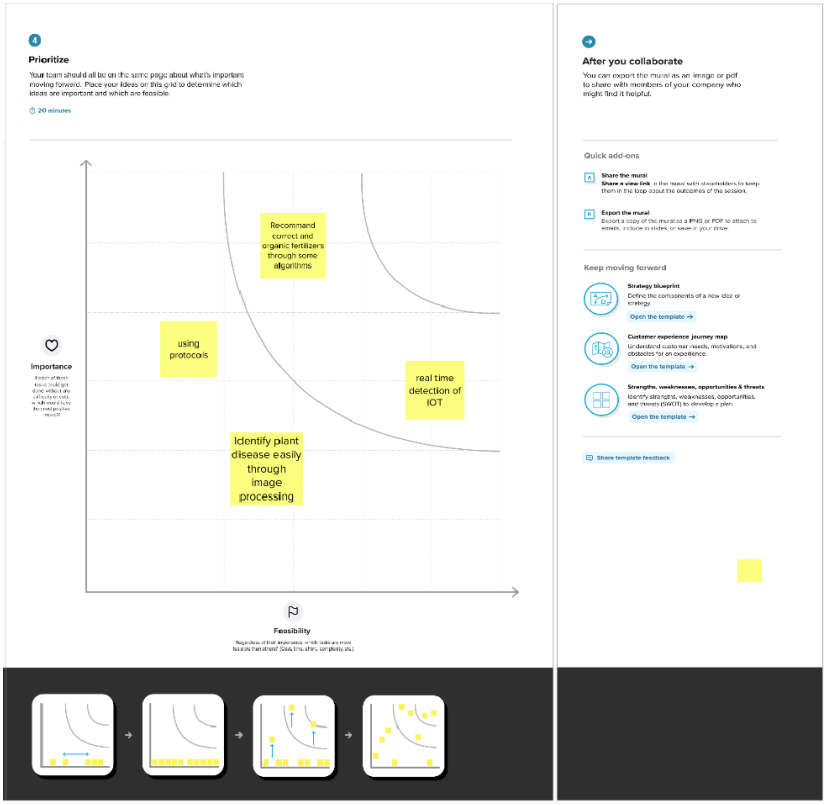
3.1. Empathy Map Canvas



3.2. Ideation & Brainstorming







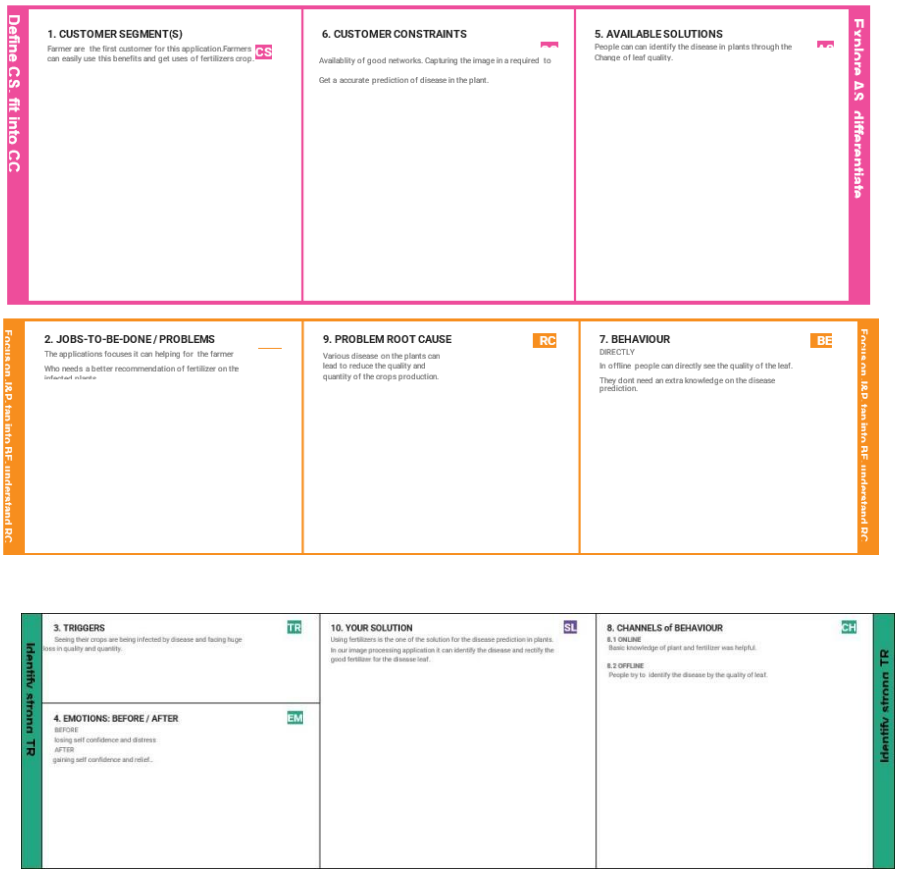
3.3. Proposed Solution

In this project work, a deep learning based neural network is used to train the collected datasets and test the same. The deep learning based neural network is CNN which gives more than 90% classification accuracies. By increasing the more number of dense layers and by modifying hyperparameters such as number of epochs, batch size, the accuracy rate can be increased to 95% to 98%.

3.4. Problem Solution Fit

Project Title: Fertilizers Recommendation System for Disease Prediction.
Team ID: PNT2022TMID51209

Project Design Phase-I - Solution Fit Template



4. REQUIREMENT ANALYSIS

4.1. Functional requirement

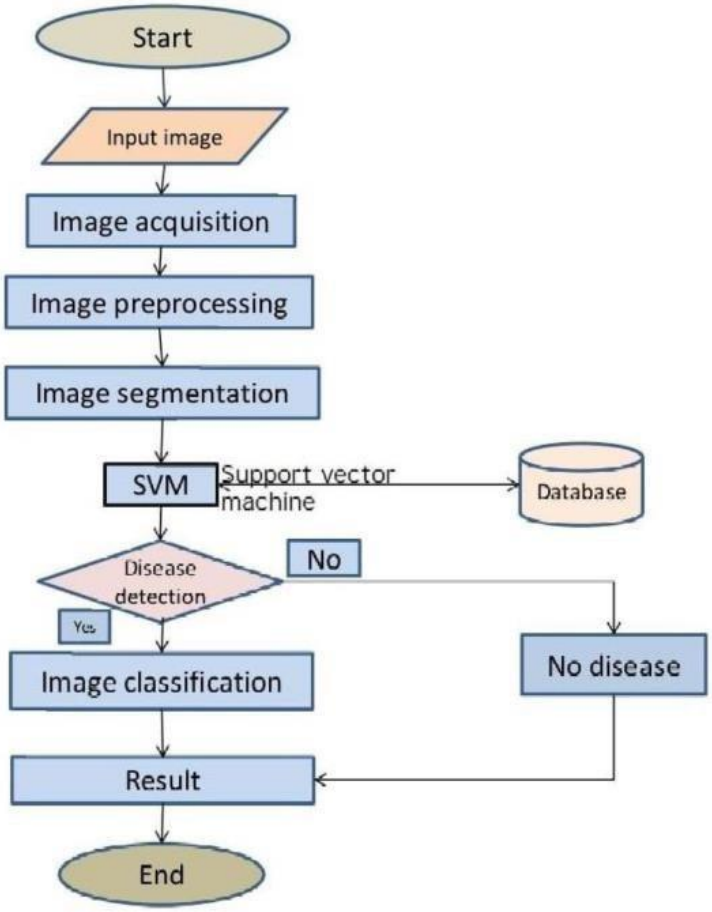
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Capturing image	Capture the image of the leaf and check 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 leaf.
Fr-6	Image description	Suggesting the best fertilizer for the disease.

4.2. Non -Functional Requirements

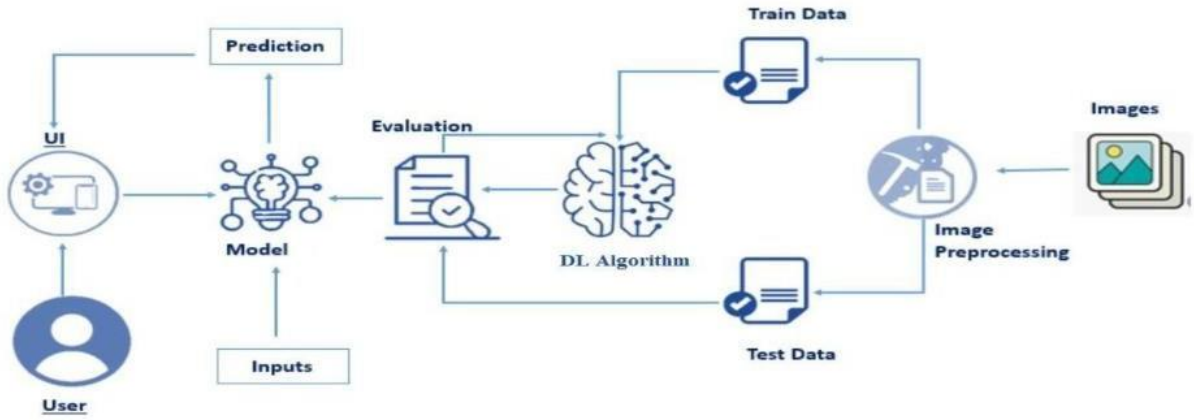
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Datasets of all the leaf is used to detecting the disease that present in the leaf.
NFR-2	**Security**	The information belongs to the user and leaf are secured highly.
NFR-3	**Reliability**	The leaf quality is important for the predicting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

5. Project Design

5.1. Data Flow Diagrams



5.2. Solution & Technical Architecture



5.3. User Stories

Functional Requirement (Epic)	User Story Number	User Story / Task
Data collection	USN-1	Collect and create the data set related to the objective
Image processing	USN-2	Process the images
Model Building for fruit disease prediction	USN-3	Import libraries
Model Building for fruit disease prediction	USN-4	Initializing the model
Model Building for fruit disease prediction	USN-5	Adding layers
Model Building for fruit disease prediction	USN-6	Train and save the model for fruits
Model Building for vegetable disease prediction	USN-7	Train and save the model for vegetable
Test both model	USN-8	Testing the built model
Application building	USN-9	Build python code
Application building	USN-10	Build HTML code
Application building	USN-11	Run the code

Train the model on IBM	USN-12	Register cloud account
Train the model on IBM	USN-13	Train the model on IBM

6. Project Planning & Scheduling

6.1. Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data collection	USN-1	Collect and create the data set related to the objective	10	High	RISHIKA.R RAJALA TEJASWI RESHMA.P POONKUZHALI.M.C
Sprint-1	Image processing	USN-2	Process the images	10	High	RISHIKA.R RAJALA TEJASWI RESHMA.P POONKUZHALI.M.C
Sprint-2	Model Building for fruit disease predicti on	USN-3	Import libraries	2	Low	RISHIKA.R RAJALA TEJASWI RESHMA.P POONKUZHALI.M.C
Sprint-2	Model Building for fruit disease predicti on	USN-4	Initializing the model	2	Low	RISHIKA.R RAJALA TEJASWI RESHMA.P POONKUZHALI.M.C
Sprint-2	Model Building for fruit disease predicti on	USN-5	Adding layers	2	Low	RISHIKA.R RAJALA TEJASWI RESHMA.P POONKUZHALI.M.C
Sprint-2	Model Building for fruit disease predicti on	USN-6	Train and save the model for fruits	7	High	RISHIKA.R RAJALA TEJASWI RESHMA.P POONKUZHALI.M.C
Sprint-2	Model Building for vegetab le disease predicti on	USN-7	Train and save the model for vegetable	7	High	RISHIKA.R RAJALA TEJASWI RESHMA.P POONKUZHALI.M.C

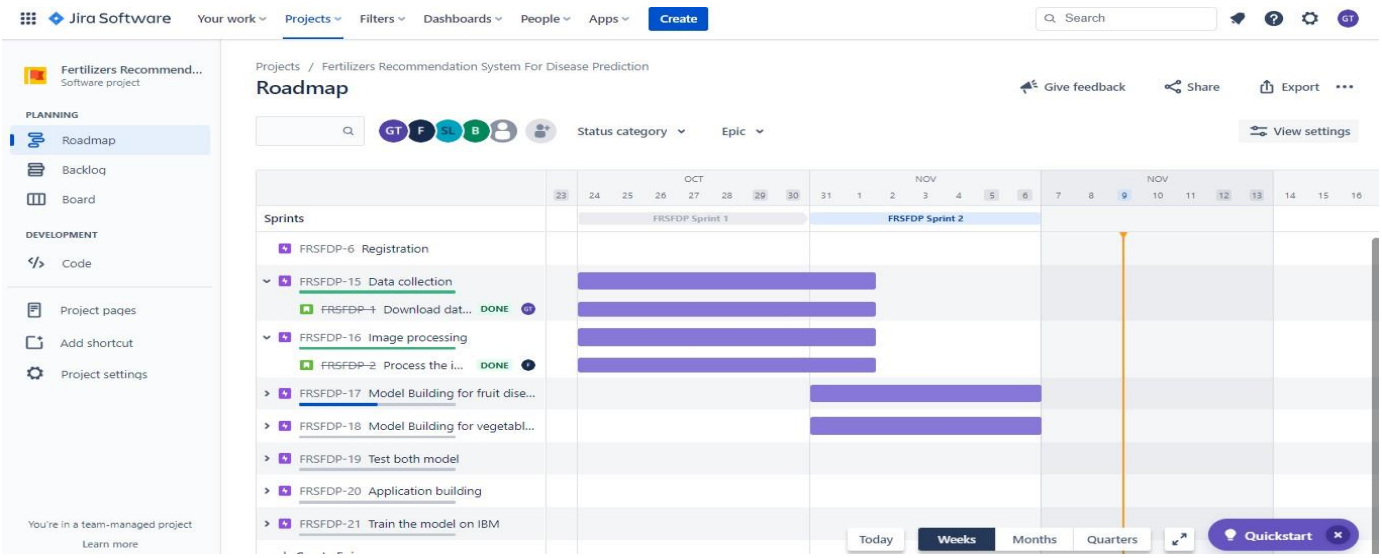
Sprint-3	Test both model	USN-8	Testing the built model	5	Medium	RISHIKA.R RAJALA TEJASWI RESHMA.P POONKUZHALI.M.C
Sprint-4	Application building	USN-9	Build python code	5	Medium	RISHIKA.R RAJALA TEJASWI RESHMA.P POONKUZHALI.M.C
Sprint-4	Application building	USN-10	Build HTML code	5	Medium	RISHIKA.R RAJALA TEJASWI RESHMA.P POONKUZHALI.M.C
Sprint-4	Application building	USN-11	Run the code	10	High	RISHIKA.R RAJALA TEJASWI RESHMA.P POONKUZHALI.M.C
Sprint-3	Train the model on IBM	USN-12	Register cloud account	5	Medium	RISHIKA.R RAJALA TEJASWI RESHMA.P POONKUZHALI.M.C
Sprint-3	Train the model on IBM	USN-13	Train the model on IBM	10	High	RISHIKA.R RAJALA TEJASWI RESHMA.P POONKUZHALI.M.C

6.2. Sprint Delivery Schedule

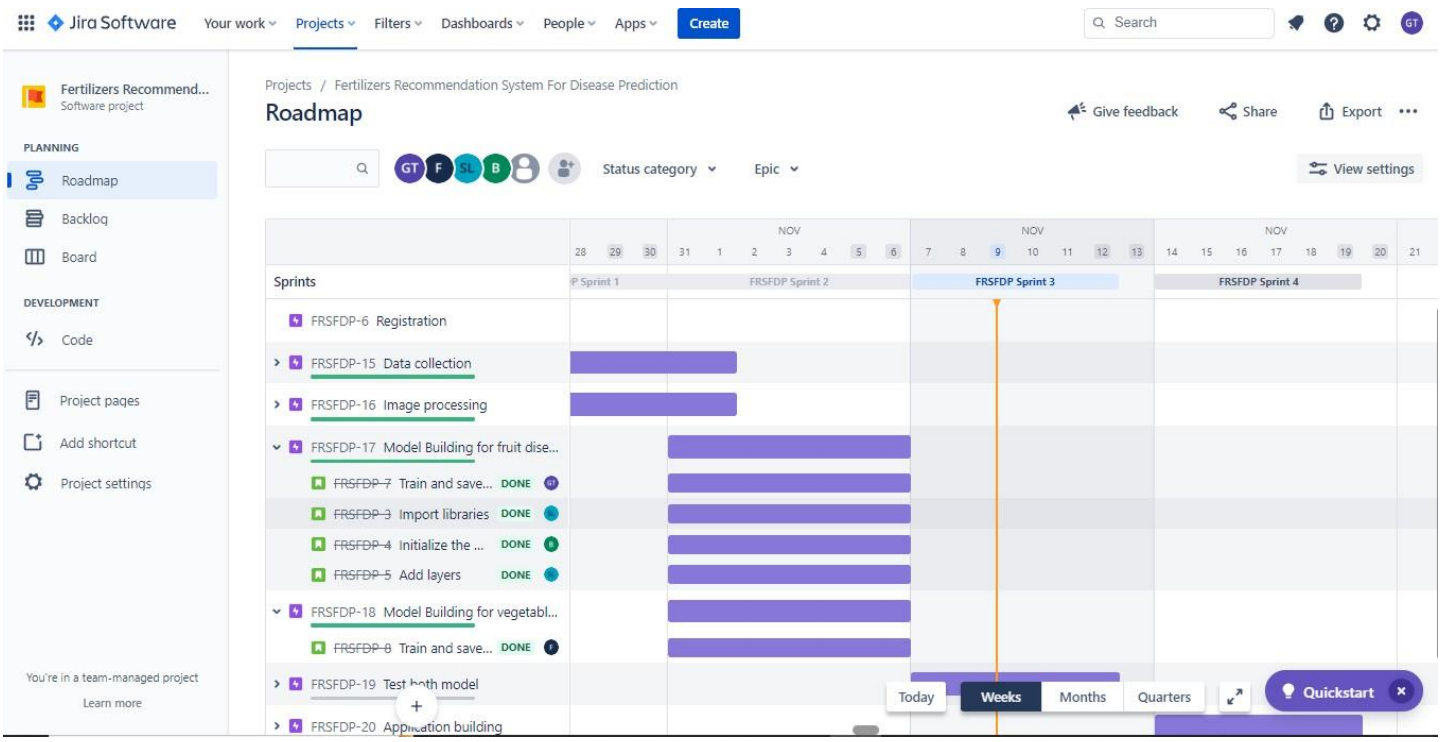
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 ov 2022

6.3. Reports From JIRA

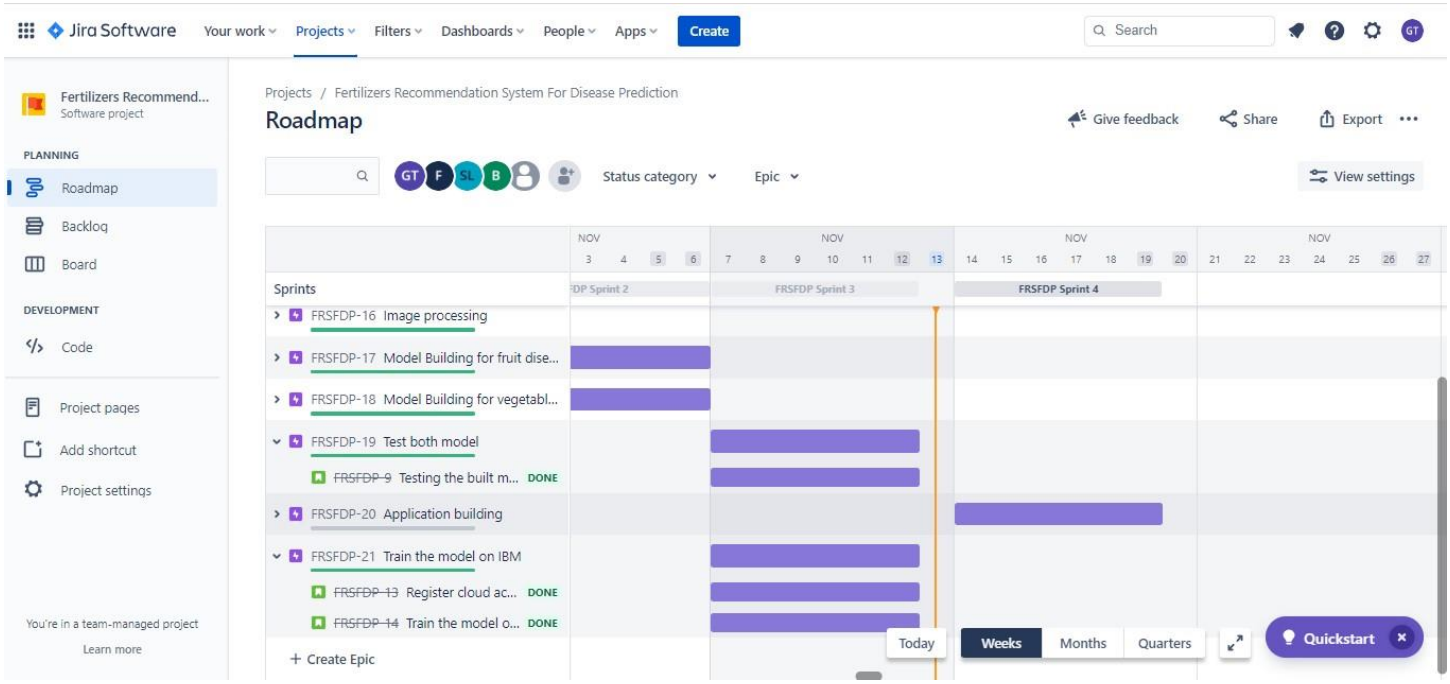
- **SPRINT – 1**



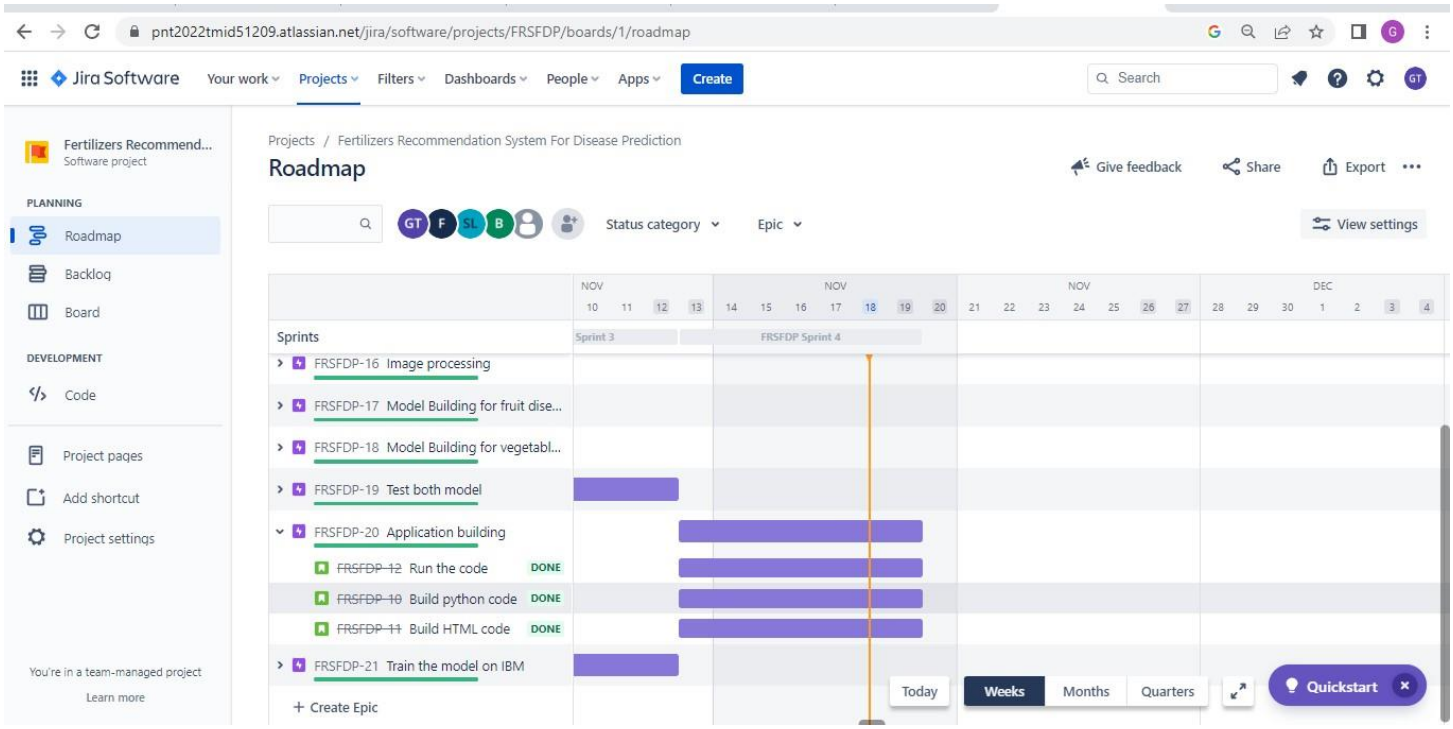
- **SPRINT – 2**



- **SPRINT – 3**



- **SPRINT – 4**



7. Coding & Solutioning

7.1. Feature 1 (HTML Code)

Index page:

```
<!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>
                   
           
           
           
                 &nb
sp;                 &
nbsp;           

                 &nb
```

```
sp;         
    <li id="abc"> plant Disease Prediction           
           
           
                 &nb
sp;                 &
nbsp;                 
;                 &nb
sp;                 &
nbsp;                 
;                 &nb
sp;                 &
nbsp;                 
;                 &nb
sp;    <a href="firstpage.html" id="abc"> home</a>
                <a href="predict.html" 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 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.</p>
        <img src="img.jpg" height="300" width="300">
    </div>
    </div>
    </div>
</body>
</html>
```

Prediction page:

```
<!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="img1.jpg">
        <div class="card">
        <form>
            <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="vagitable" 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>
```

7.2. Feature 2 (Python code)

```
import os
from flask import Flask, redirect, render_template, request
from PIL import Image
import torchvision.transforms.functional as TF
import CNN
import numpy as np
import torch
import pandas as pd
import torch.nn as nn


disease_info = pd.read_csv('disease_info.csv' , encoding='cp1252')
supplement_info = pd.read_csv('supplement_info.csv',encoding='cp1252')

model = CNN.CNN(39)
model = nn.DataParallel(model)
model.load_state_dict(torch.load(r"../Model/model.pth", map_location=torch.device("cpu")))
model.eval()


def prediction(image_path):
    image = Image.open(image_path)
    image = image.resize((224, 224))
    input_data = TF.to_tensor(image)
    input_data = input_data.view((-1, 3, 224, 224))
    output = model(input_data)
    output = output.detach().numpy()
    index = np.argmax(output)
```

```
 return index
app = Flask(_name_)

@app.route('/')
def home_page():
    return render_template('home.html')

@app.route('/index')
def ai_engine_page():
    return render_template('index.html')

@app.route('/mobile-device')
def mobile_device_detected_page():
    return render_template('mobile-device.html')

@app.route('/submit',  methods=['GET',  'POST'])
def submit():
    if request.method  == 'POST':
        image = request.files['image']
        filename = image.filename
        file_path = os.path.join('static/uploads',  filename)
        image.save(file_path)

        print(file_path)
        pred = prediction(file_path)
        title = disease_info['disease_name'][pred]
        description =disease_info['description'][pred]
        prevent = disease_info['Possible  Steps'][pred]
        image_url = disease_info['image_url'][pred]
        supplement_name  = supplement_info['supplement  name'][pred]
        supplement_image_url  = supplement_info['supplement  image'][pred]
        supplement_buy_link  = supplement_info['buy  link'][pred]
        return render_template('submit.html'  , title = title , desc = description , prevent = prevent ,
                           image_url = image_url , pred = pred ,sname = supplement_name  , simage =
supplement_image_url  , buy_link = supplement_buy_link)

@app.route('/market',  methods=['GET',  'POST'])
def market():
    return render_template('market.html',  supplement_image  = list(supplement_info['supplement  image']),
                       supplement_name  = list(supplement_info['supplement  name']), disease =
list(disease_info['disease_name']),  buy = list(supplement_info['buy  link']))

if _name_  == '_main_':
    app.run(debug=True)
```

8. Testing

8.1. Test Cases

				Date	18-Nov-22								
				Team ID	PNT2022TMID15357								
				Project Name	Project - Fertilizers Recommendation System for Disease Prediction								
				Maximum Marks	4 marks								
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Pre-Requisite**	**Steps To Execute**	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**Comments**	**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	-	RISHIKA R
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	-	RAJALA TEJASWI
PredictPage_TC_003	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	-	RESHMA.P ,POONKUZHALI.M.C
PredictPage_TC_004	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	-	POONKUZHALI.M .C
PredictPage_TC_005	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	-	RISHIKA.R, RESHMA.P,
PredictPage_TC_006	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	-	RAJALATEJASWI ,RISHIKA.R
PredictPage_TC_007	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	-	POONKUZHAILM .C RESHMA.P

8.2. User Acceptance Testing

- **Defect Analysis**

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**

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	1	0	0	1
Client Application	1	0	0	1

9. Results

9.1. Performance Metrics

Model Summary

Total params: 5,084,552

Trainable params: 5,084,552

Non-trainable params: 0

```
In [41]: model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_5 (Conv2D)           (None, 126, 126, 32)      896

 max_pooling2d_2 (MaxPooling  (None, 63, 63, 32)       0
 2D)

 flatten_2 (Flatten)         (None, 127008)            0

 dense_6 (Dense)             (None, 40)                5080360

 dense_7 (Dense)             (None, 70)                2870

 dense_8 (Dense)             (None, 6)                 426

=================================================================
Total params: 5,084,552
Trainable params: 5,084,552
Non-trainable params: 0
_________________________________________________________________
```

Accuracy

Training Accuracy – 96.55

Validation Accuracy – 97.45

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

```
C:\Users\Sree Ram\AppData\Local\Temp\ipykernel_13228\1582812018.py:1: UserWarning: `Model.fit_generator` is deprecated and will
be removed in a future version. Please use `Model.fit`, which supports generators.
  model.fit_generator(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test),epochs=10)
Epoch 1/10
225/225 [==============================] - 96s 425ms/step - loss: 1.1095 - accuracy: 0.7829 - val_loss: 0.3157 - val_accuracy:
0.8861
Epoch 2/10
225/225 [==============================] - 88s 393ms/step - loss: 0.2825 - accuracy: 0.9042 - val_loss: 0.3015 - val_accuracy:
0.9075
Epoch 3/10
225/225 [==============================] - 85s 375ms/step - loss: 0.2032 - accuracy: 0.9303 - val_loss: 0.2203 - val_accuracy:
0.9288
Epoch 4/10
225/225 [==============================] - 84s 374ms/step - loss: 0.1576 - accuracy: 0.9463 - val_loss: 0.2424 - val_accuracy:
0.9164
Epoch 5/10
225/225 [==============================] - 84s 372ms/step - loss: 0.1719 - accuracy: 0.9389 - val_loss: 0.1330 - val_accuracy:
0.9632
Epoch 6/10
225/225 [==============================] - 85s 376ms/step - loss: 0.1240 - accuracy: 0.9580 - val_loss: 0.1340 - val_accuracy:
0.9573
Epoch 7/10
225/225 [==============================] - 87s 388ms/step - loss: 0.1235 - accuracy: 0.9591 - val_loss: 0.1638 - val_accuracy:
0.9478
Epoch 8/10
225/225 [==============================] - 83s 371ms/step - loss: 0.1012 - accuracy: 0.9643 - val_loss: 0.1468 - val_accuracy:
0.9561
Epoch 9/10
225/225 [==============================] - 83s 367ms/step - loss: 0.0967 - accuracy: 0.9655 - val_loss: 0.1412 - val_accuracy:
0.9531
Epoch 10/10
225/225 [==============================] - 83s 369ms/step - loss: 0.0954 - accuracy: 0.9655 - val_loss: 0.0905 - val_accuracy:
0.9745
```

10. ADVANTAGES & DISADVANTAGES

List of advantages

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

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

List of disadvantages

- For training and testing, the proposed model requires very high computational time.
- The neural network architecture used in this project work has high complexity.

11. CONCLUSION

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

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

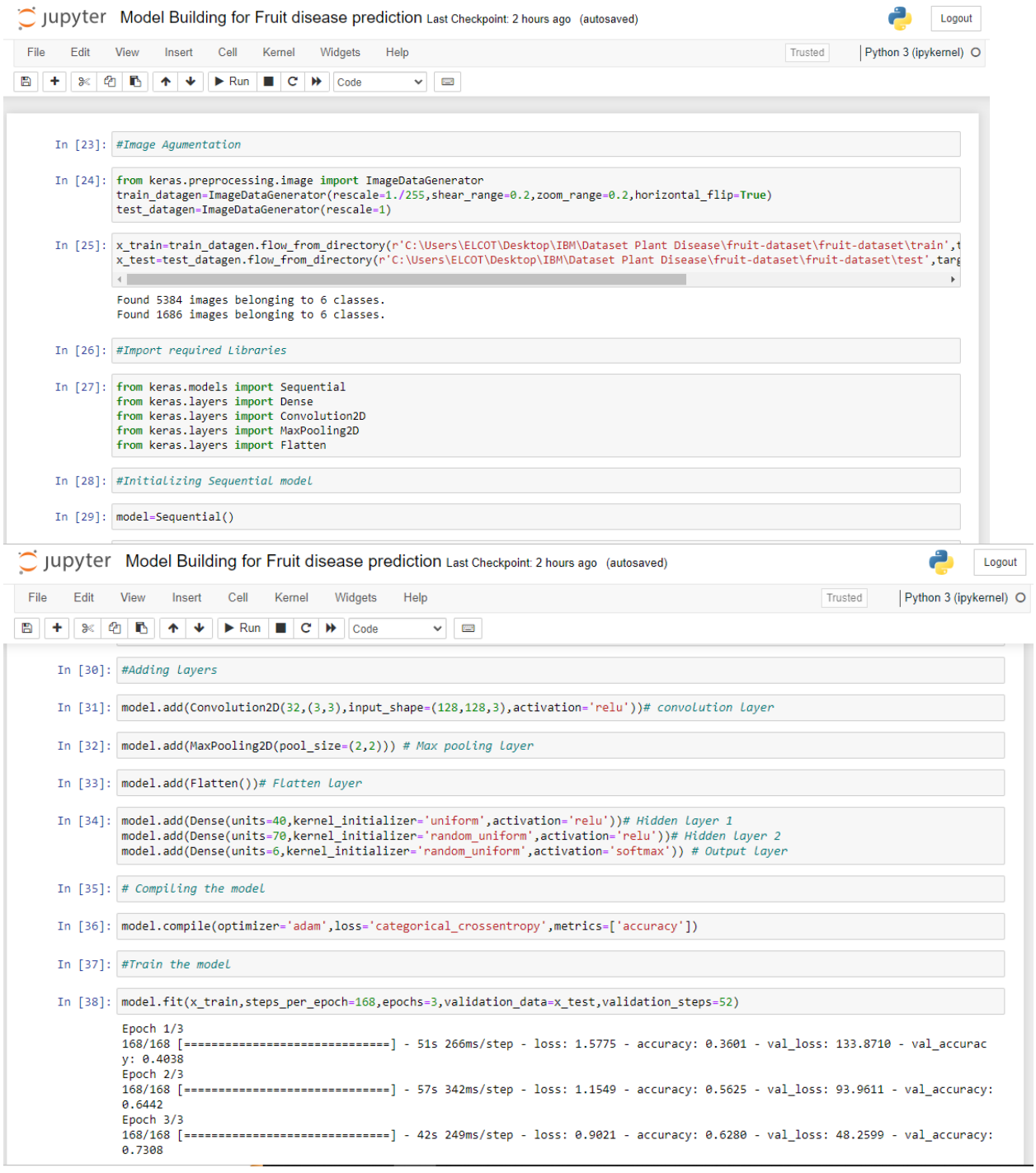
12. FUTURE SCOPE

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

13. Appendix

13.1. Source Code

Model Building For Fruit Disease Prediction



jupyter Model Building for Fruit disease prediction Last Checkpoint: 2 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

```
In [23]: #Image Agumentation

In [24]: from keras.preprocessing.image import ImageDataGenerator
         train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
         test_datagen=ImageDataGenerator(rescale=1)

In [25]: x_train=train_datagen.flow_from_directory(r'C:\Users\ELCOT\Desktop\IBM\Dataset Plant Disease\fruit-dataset\fruit-dataset\train',t
         x_test=test_datagen.flow_from_directory(r'C:\Users\ELCOT\Desktop\IBM\Dataset Plant Disease\fruit-dataset\fruit-dataset\test',targ

         Found 5384 images belonging to 6 classes.
         Found 1686 images belonging to 6 classes.

In [26]: #Import required libraries

In [27]: 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

In [28]: #Initializing Sequential model

In [29]: model=Sequential()
```

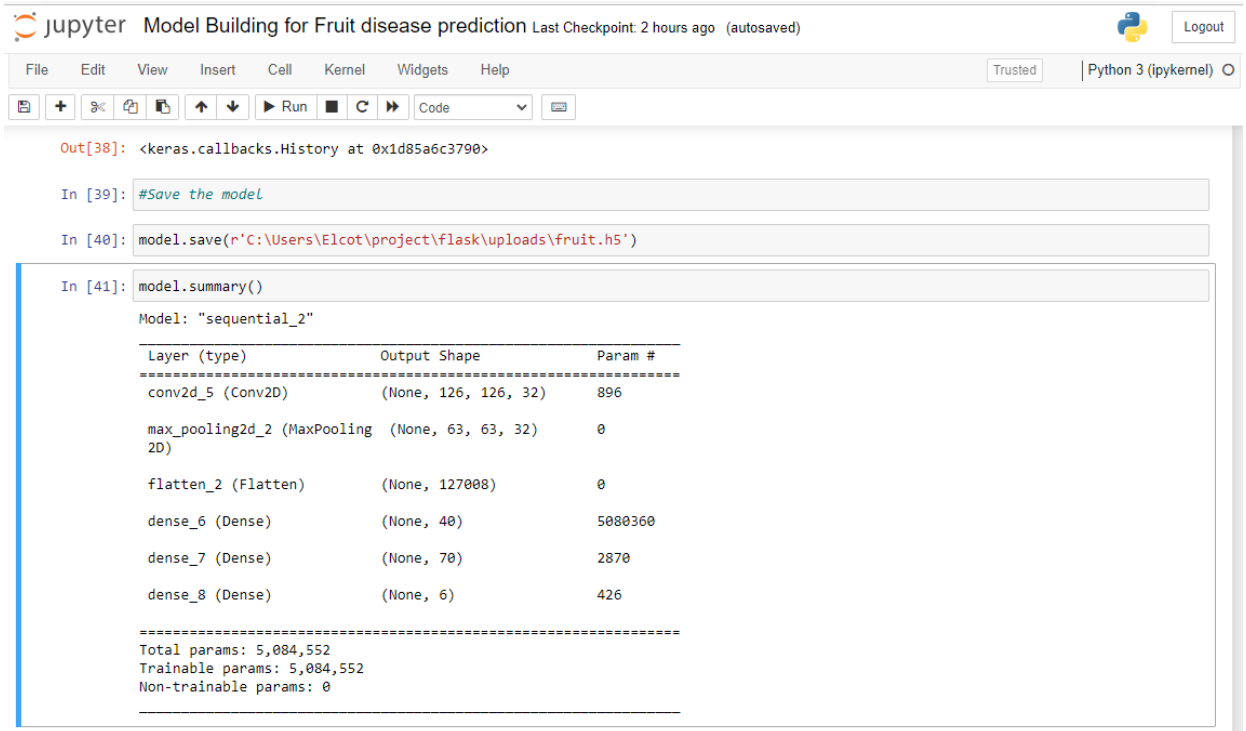

jupyter Model Building for Fruit disease prediction Last Checkpoint: 2 hours ago (autosaved)

Logout

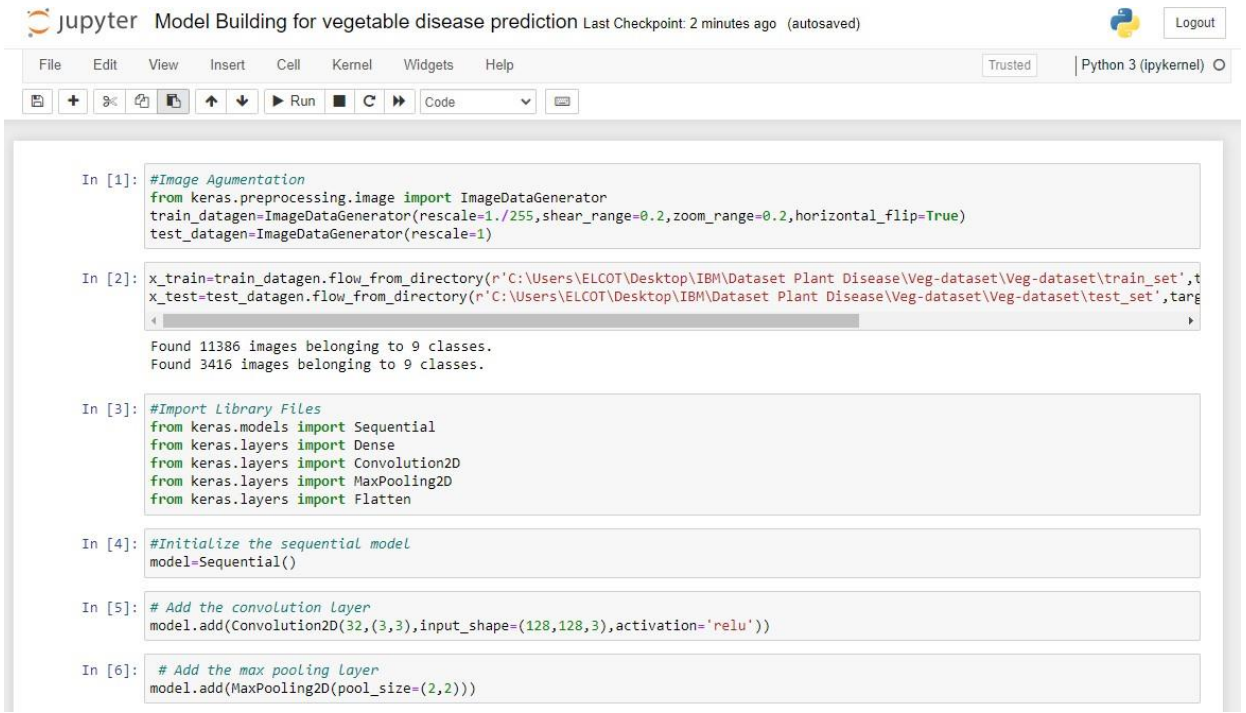
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

```
In [30]: #Adding Layers

In [31]: model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))# convolution layer

In [32]: model.add(MaxPooling2D(pool_size=(2,2))) # Max pooling layer

In [33]: model.add(Flatten())# Flatten layer

In [34]: model.add(Dense(units=40,kernel_initializer='uniform',activation='relu'))# Hidden layer 1
         model.add(Dense(units=70,kernel_initializer='random_uniform',activation='relu'))# Hidden layer 2
         model.add(Dense(units=6,kernel_initializer='random_uniform',activation='softmax')) # Output layer

In [35]: # Compiling the model

In [36]: model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])

In [37]: #Train the model

In [38]: model.fit(x_train,steps_per_epoch=168,epochs=3,validation_data=x_test,validation_steps=52)

         Epoch 1/3
         168/168 [==============================] - 51s 266ms/step - loss: 1.5775 - accuracy: 0.3601 - val_loss: 133.8710 - val_accurac
         y: 0.4038
         Epoch 2/3
         168/168 [==============================] - 57s 342ms/step - loss: 1.1549 - accuracy: 0.5625 - val_loss: 93.9611 - val_accuracy:
         0.6442
         Epoch 3/3
         168/168 [==============================] - 42s 249ms/step - loss: 0.9021 - accuracy: 0.6280 - val_loss: 48.2599 - val_accuracy:
         0.7308
```



Model Building for Vegetable disease Prediction



jupyter Model Building for vegetable disease prediction Last Checkpoint: 3 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [7]:
```
# Add the Flatten Layer
model.add(Flatten())
```

In [8]:
```
# Add the Hidden Layer 1
model.add(Dense(units=300,kernel_initializer='uniform',activation='relu'))
```

In [9]:
```
# Add the Hidden Layer 2
model.add(Dense(units=150,kernel_initializer='uniform',activation='relu'))
```

In [10]:
```
# Add the Hidden Layer 3
model.add(Dense(units=75,kernel_initializer='uniform',activation='relu'))
```

In [11]:
```
# Add the Output Layer
model.add(Dense(units=9,kernel_initializer='uniform',activation='softmax'))
```

In [12]:
```
# Compiling the model
model.compile(loss='categorical_crossentropy',optimizer="adam",metrics=["accuracy"])
```

In [13]:
```
# Train model
model.fit(x_train,steps_per_epoch=89,epochs=20,validation_data=x_test,validation_steps=27)
```

```
Epoch 1/20
89/89 [==============================] - 66s 690ms/step - loss: 2.1881 - accuracy: 0.1685 - val_loss: 194.3618 - val_accuracy:
0.1296
Epoch 2/20
89/89 [==============================] - 59s 663ms/step - loss: 2.1182 - accuracy: 0.1236 - val_loss: 109.5233 - val_accuracy:
0.1296
Epoch 3/20
89/89 [==============================] - 67s 755ms/step - loss: 2.0954 - accuracy: 0.1180 - val_loss: 80.7805 - val_accuracy:
0.1296
```

jupyter Model Building for vegetable disease prediction Last Checkpoint: 4 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

```
Epoch 4/20
89/89 [==============================] - 73s 817ms/step - loss: 2.1350 - accuracy: 0.1854 - val_loss: 164.9028 - val_accuracy:
0.1111
Epoch 5/20
89/89 [==============================] - 69s 776ms/step - loss: 2.1272 - accuracy: 0.1798 - val_loss: 74.0293 - val_accuracy:
0.1296
Epoch 6/20
89/89 [==============================] - 65s 729ms/step - loss: 2.0987 - accuracy: 0.2079 - val_loss: 94.8363 - val_accuracy:
0.2037
Epoch 7/20
89/89 [==============================] - 57s 643ms/step - loss: 2.0832 - accuracy: 0.2079 - val_loss: 32.7240 - val_accuracy:
0.0926
Epoch 8/20
89/89 [==============================] - 56s 628ms/step - loss: 2.0565 - accuracy: 0.2584 - val_loss: 134.1384 - val_accuracy:
0.2963
Epoch 9/20
89/89 [==============================] - 56s 628ms/step - loss: 2.1103 - accuracy: 0.1517 - val_loss: 61.7429 - val_accuracy:
0.1481
Epoch 10/20
89/89 [==============================] - 55s 622ms/step - loss: 2.0565 - accuracy: 0.2416 - val_loss: 121.7573 - val_accuracy:
0.1481
Epoch 11/20
89/89 [==============================] - 53s 598ms/step - loss: 2.1168 - accuracy: 0.1854 - val_loss: 30.3666 - val_accuracy:
0.1667
Epoch 12/20
89/89 [==============================] - 57s 635ms/step - loss: 2.0713 - accuracy: 0.2079 - val_loss: 56.4360 - val_accuracy:
0.2778
Epoch 13/20
89/89 [==============================] - 59s 665ms/step - loss: 1.9673 - accuracy: 0.2416 - val_loss: 66.7629 - val_accuracy:
0.3333
Epoch 14/20
89/89 [==============================] - 55s 612ms/step - loss: 1.9854 - accuracy: 0.3202 - val_loss: 95.1238 - val_accuracy:
0.3148
```

jupyter Model Building for vegetable disease prediction Last Checkpoint: 5 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

```
Epoch 15/20
89/89 [==============================] - 55s 614ms/step - loss: 1.7150 - accuracy: 0.3933 - val_loss: 148.8870 - val_accuracy:
0.2037
Epoch 16/20
89/89 [==============================] - 56s 624ms/step - loss: 1.6605 - accuracy: 0.3708 - val_loss: 292.1395 - val_accuracy:
0.2222
Epoch 17/20
89/89 [==============================] - 55s 611ms/step - loss: 1.8242 - accuracy: 0.3427 - val_loss: 156.9938 - val_accuracy:
0.3704
Epoch 18/20
89/89 [==============================] - 56s 628ms/step - loss: 1.7093 - accuracy: 0.3876 - val_loss: 161.5939 - val_accuracy:
0.2778
Epoch 19/20
89/89 [==============================] - 66s 745ms/step - loss: 1.6356 - accuracy: 0.3933 - val_loss: 143.6697 - val_accuracy:
0.3704
Epoch 20/20
89/89 [==============================] - 64s 714ms/step - loss: 1.8686 - accuracy: 0.3427 - val_loss: 70.6791 - val_accuracy:
0.4259
```

Out[13]: <keras.callbacks.History at 0x259831a6be0>

In [16]:
```
# Save model
model.save(r'C:\Users\Elcot\project\flask\uploads\vegetable.h5')
```

jupyter Model Building for vegetable disease prediction Last Checkpoint: 6 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

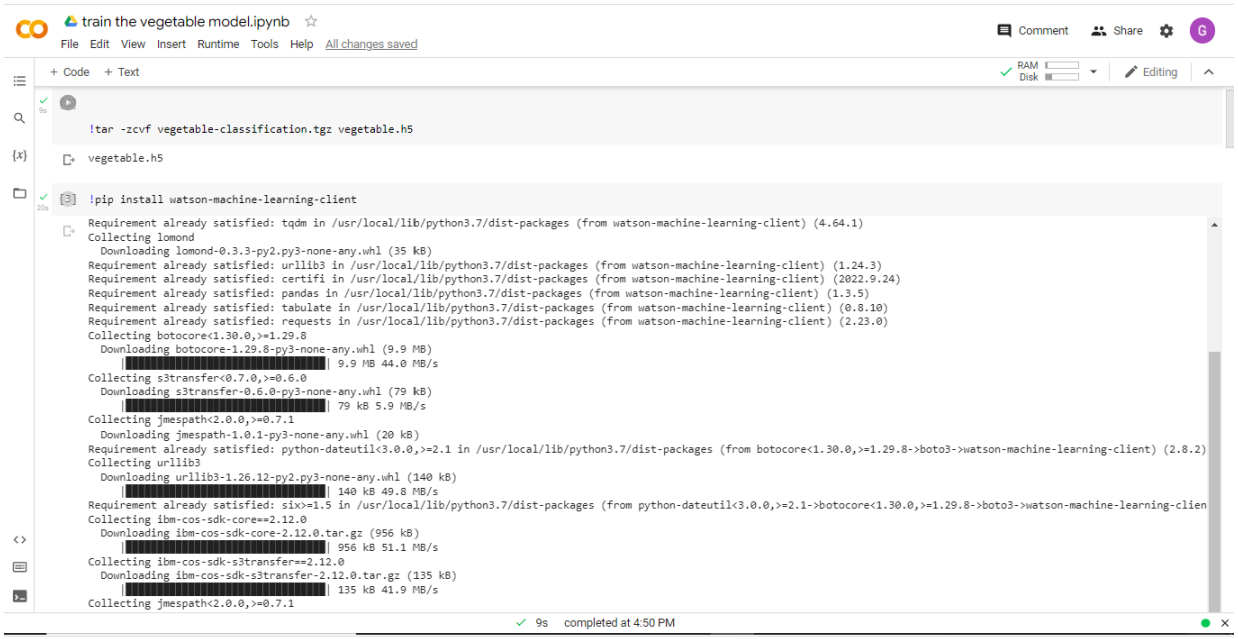
Run Code

In [16]:
```
# Save model
model.save(r'C:\Users\Elcot\project\flask\uploads\vegetable.h5')
```

In [17]:
```
model.summary()
```

```
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

 dense (Dense)               (None, 300)               38102700

 dense_1 (Dense)             (None, 150)               45150

 dense_2 (Dense)             (None, 75)                11325

 dense_3 (Dense)             (None, 9)                 684

=================================================================
Total params: 38,160,755
Trainable params: 38,160,755
Non-trainable params: 0
_________________________________________________________________
```

Train the Vegetable model on IBM



train the vegetable model.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share G

+ Code + Text

RAM Disk Editing

```
!pip install ibm_watson_machine_learning
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting ibm_watson_machine_learning
  Downloading ibm_watson_machine_learning-1.0.257-py3-none-any.whl (1.8 MB)
     |████████████████████████████████| 1.8 MB 9.4 MB/s
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (1.3.5)
Requirement already satisfied: lomond in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (0.3.3)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (1.26.12)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (2.28.1)
Collecting ibm-cos-sdk==2.7.*
  Downloading ibm-cos-sdk-2.7.0.tar.gz (51 kB)
     |████████████████████████████████| 51 kB 708 kB/s
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (21.3)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (4.13.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (2022.9.24)
Requirement already satisfied: tabulate in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (0.8.10)
Collecting ibm-cos-sdk-core==2.7.0
  Downloading ibm-cos-sdk-core-2.7.0.tar.gz (824 kB)
     |████████████████████████████████| 824 kB 52.1 MB/s
Collecting ibm-cos-sdk-s3transfer==2.7.0
  Downloading ibm-cos-sdk-s3transfer-2.7.0.tar.gz (133 kB)
     |████████████████████████████████| 133 kB 51.2 MB/s
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /usr/local/lib/python3.7/dist-packages (from ibm-cos-sdk==2.7.*->ibm_watson_machine_learning) (0.10.0)
Collecting docutils<0.16,>=0.10
  Downloading docutils-0.15.2-py3-none-any.whl (547 kB)
     |████████████████████████████████| 547 kB 49.4 MB/s
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/python3.7/dist-packages (from ibm-cos-sdk-core==2.7.0->ibm-cos-sdk==2.7.*->ibm_watson_machine_learning
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (2022.6)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (1.21.6)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.7.0->ibm-cos-sdk==2.7.*->ibm_watson_machin
Requirement already satisfied: charset-normalizer<3,>=2 in /usr/local/lib/python3.7/dist-packages (from requests->ibm_watson_machine_learning) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->ibm_watson_machine_learning) (2.10)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->ibm_watson_machine_learning) (3.10.0)
Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->ibm_watson_machine_learning) (4.1.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->ibm_watson_machine_learning) (3.0.9)
```

9s completed at 4:50 PM

train the vegetable model.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share G

+ Code + Text

RAM Disk Editing

```
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->ibm_watson_machine_learning) (3.0.9)
Building wheels for collected packages: ibm-cos-sdk, ibm-cos-sdk-core, ibm-cos-sdk-s3transfer
  Building wheel for ibm-cos-sdk (setup.py) ... done
  Created wheel for ibm-cos-sdk: filename=ibm_cos_sdk-2.7.0-py2.py3-none-any.whl size=72563 sha256=cd40613ff61488741dd9bde985e8c9561c1a2ae2743ed568bbdb200a974f8715
  Stored in directory: /root/.cache/pip/wheels/47/22/bf/e1154ff0f5de93cc477acd0ca69abfbb8b799c5b28a66b44c2
  Building wheel for ibm-cos-sdk-core (setup.py) ... done
  Created wheel for ibm-cos-sdk-core: filename=ibm_cos_sdk_core-2.7.0-py2.py3-none-any.whl size=501013 sha256=f44c9b5718819e77836f7a0873bec7ea10d05f84b04b8e83813378e42f2accfc
  Stored in directory: /root/.cache/pip/wheels/6c/a2/e4/c16d02f809a3ea998e17cfd02c13369281f3d232aaf5902c19
  Building wheel for ibm-cos-sdk-s3transfer (setup.py) ... done
  Created wheel for ibm-cos-sdk-s3transfer: filename=ibm_cos_sdk_s3transfer-2.7.0-py2.py3-none-any.whl size=88622 sha256=8328123a45bd1abb82b1298fbf27c0d4e3ccc067064611a3c0449fab16
  Stored in directory: /root/.cache/pip/wheels/5f/b7/14/fbe02bc1ef1af890650c7e51743d1c83890852e598d164b9da
Successfully built ibm-cos-sdk ibm-cos-sdk-core ibm-cos-sdk-s3transfer
Installing collected packages: docutils, ibm-cos-sdk-core, ibm-cos-sdk-s3transfer, ibm-cos-sdk, ibm-watson-machine-learning
  Attempting uninstall: docutils
    Found existing installation: docutils 0.17.1
    Uninstalling docutils-0.17.1:
      Successfully uninstalled docutils-0.17.1
  Attempting uninstall: ibm-cos-sdk-core
    Found existing installation: ibm-cos-sdk-core 2.12.0
    Uninstalling ibm-cos-sdk-core-2.12.0:
      Successfully uninstalled ibm-cos-sdk-core-2.12.0
  Attempting uninstall: ibm-cos-sdk-s3transfer
    Found existing installation: ibm-cos-sdk-s3transfer 2.12.0
    Uninstalling ibm-cos-sdk-s3transfer-2.12.0:
      Successfully uninstalled ibm-cos-sdk-s3transfer-2.12.0
  Attempting uninstall: ibm-cos-sdk
    Found existing installation: ibm-cos-sdk 2.12.0
    Uninstalling ibm-cos-sdk-2.12.0:
      Successfully uninstalled ibm-cos-sdk-2.12.0
Successfully installed docutils-0.15.2 ibm-cos-sdk-2.7.0 ibm-cos-sdk-core-2.7.0 ibm-cos-sdk-s3transfer-2.7.0 ibm-watson-machine-learning-1.0.257
```

train the vegetable model.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share G

+ Code + Text

RAM Disk Editing

```
[5] from ibm_watson_machine_learning import APIClient
    wml_credentials ={
        "url":"https://eu-gb.ml.cloud.ibm.com",
        "apikey":"wHviGg2JmwaTLSiOX_1iRsFEiTms97bQQIt3fFv1WGXC"
    }
    client = APIClient(wml_credentials)
```

```
Python 3.7 and 3.8 frameworks are deprecated and will be removed in a future release. Use Python 3.9 framework instead.
```

```
[6] client
```

```
<ibm_watson_machine_learning.client.APIClient at 0x7f335d3cc750>
```

```
client.spaces.get_details()
```

```
{'resources': [{'entity': {'compute': [{'crn': 'crn:v1:bluemix:public:pm-20:eu-gb:a/d2a512c7590048cfaf7f28cd26438f49:f78a826e-3c47-40ec-9ced-ab40e6d544d3::',
      'guid': 'f78a826e-3c47-40ec-9ced-ab40e6d544d3',
      'name': 'Watson Machine Learning-ak',
      'type': 'machine_learning'}],
    'description': '',
    'name': 'v1_Deploy',
    'scope': {'bss_account_id': 'd2a512c7590048cfaf7f28cd26438f49'},
    'stage': {'production': False},
    'status': {'state': 'active'},
    'storage': {'properties': {'bucket_name': 'a90b4771-730e-49cc-b5e5-1758dc2fb7c1',
      'bucket_region': 'eu-gb-standard',
      'credentials': {'admin': {'access_key_id': 'e827a1d463054462be58c11d21f4a342',
        'api_key': 'hIknNxBOWj461q3pDFSLZ-NveCEz7cnqSm8c0lQa4LG3',
        'secret_access_key': 'e730510472c7fed75868de0d665b713b701909a93da86a73',
        'service_id': 'ServiceId-a30f82e7-316e-46bd-b229-331030406a23'},
       'editor': {'access_key_id': '004f08db143c4550b25166a8f5271771',
        'api_key': '9Ysax1Y9-XojQaS6GwvJZKf64HyFtwcxf67ncD-jrXqu',
        'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/d2a512c7590048cfaf7f28cd26438f49:0e43b750-15aa-4634-9c02-9954d7f0be49::',
        'secret_access_key': '7cbb5e54037e5876c10f85a86b0acd6e6fba4b2caa4338f7',
```

9s completed at 4:50 PM

train the vegetable model.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share G

+ Code + Text

RAM Disk Editing


```
 'stage': {'production': False},
 'status': {'state': 'active'},
 'storage': {'properties': {'bucket_name': 'a90b4771-730e-49cc-b5e5-1758dc2fb7c1',
   'bucket_region': 'eu-gb-standard',
   'credentials': {'admin': {'access_key_id': 'e827a1d463054462be58c11d21f4a342',
     'api_key': 'hIknNxBOWj46lq3pDFSlZ-NveCEz7cnqSm8c0lQa4LG3',
     'secret_access_key': 'e730510472c7fed75868de0d665b713b701909a93da86a73',
     'service_id': 'ServiceId-a30f82e7-316e-46bd-b229-331030406a23'},
    'editor': {'access_key_id': '004f08db143c4550b25166a8f5271771',
     'api_key': '9Ysax1Y9-XojQaS6GwvJIKf64HyFtwcxf67ncD-jrXqu',
     'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/d2a512c7590048cfaf7f28cd26438f49:0e43b750-15aa-4634-9c02-9954d7f0be49::',
     'secret_access_key': '7cbb5e54037e5876c10f85a86b0acd6e6fba4b2caa4338f7',
     'service_id': 'ServiceId-68abd7ca-11d3-4e3e-b8dd-da61d9c93eae'},
    'viewer': {'access_key_id': '08ec08d406014223b26317ff7d844345',
     'api_key': 'amCM5-GBgCOFE6cmRFIl18Pz5kIVU8SFiTPFL284UmPP',
     'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/d2a512c7590048cfaf7f28cd26438f49:0e43b750-15aa-4634-9c02-9954d7f0be49::',
     'secret_access_key': 'e7c659f9ebf2790368703f119b8a0f28082cf7256833fd2d',
     'service_id': 'ServiceId-ea9f16f2-4120-4ae0-b248-f5bc5605c424'}},
   'endpoint_url': 'https://s3.eu-gb.cloud-object-storage.appdomain.cloud',
   'guid': '0e43b750-15aa-4634-9c02-9954d7f0be49',
   'resource_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/d2a512c7590048cfaf7f28cd26438f49:0e43b750-15aa-4634-9c02-9954d7f0be49::'},
  'type': 'bmcos_object_storage'}},
 'metadata': {'created_at': '2022-11-13T10:24:33.382Z',
  'creator_id': 'IBMid-6640043H3B',
  'id': 'c77baaef-a816-44c5-95eb-9d2bf178dc6f',
  'updated_at': '2022-11-13T10:24:47.901Z',
  'url': '/v2/spaces/c77baaef-a816-44c5-95eb-9d2bf178dc6f'}}]}
```


```
[8] client.spaces.list()
```

```
Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
------------------------------------  ---------  ------------------------
ID                                    NAME       CREATED
c77baaef-a816-44c5-95eb-9d2bf178dc6f  v1_Deploy  2022-11-13T10:24:33.382Z
------------------------------------  ---------  ------------------------
```

✓ 9s completed at 4:50 PM

train the vegetable model.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share G

+ Code + Text

RAM Disk Editing


```
[9] space_uid = "c77baaef-a816-44c5-95eb-9d2bf178dc6f"
    space_uid
```

```
'c77baaef-a816-44c5-95eb-9d2bf178dc6f'
```

```
[10] client.set.default_space(space_uid)
```

```
'SUCCESS'
```

```
client.software_specifications.list()
```

```
------------------------------  ------------------------------------  ----
NAME                            ASSET_ID                              TYPE
default_py3.6                   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6         09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12      09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9       0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6           0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                      0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod    1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6               10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl       111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10         125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9              12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6         154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                    1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6          1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
kernel-spark3.3-r3.6            1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt   1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6            1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2                 20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod    217c16f6-178f-56bf-824a-b19f20564c49  base
```

✓ 9s completed at 4:50 PM

train the vegetable model.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share G

+ Code + Text

RAM Disk Editing

```
runtime-22.1-py3.9              12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6         154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                    1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6          1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
kernel-spark3.3-r3.6            1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt   1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6            1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2                 20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod    217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda         26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                        295addb5-9ef9-547e-9bf4-92ae3563e720  base
autoai-ts_3.8-py3.8             2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
tensorflow_1.15-py3.6           2b73a275-7cbf-420b-a912-eae7f436e0bc  base
kernel-spark3.3-py3.9           2b7961e2-e3b1-5a8c-a491-482c8368839a  base
pytorch_1.2-py3.6               2c8ef57d-2687-4b7d-acce-01f94976dac1  base
spark-mllib_2.3                 2e51f700-bca0-4b0d-88dc-5c6791338875  base
pytorch-onnx_1.1-py3.6-edt      32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37            36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                 390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10         396b2e83-0953-5b86-9a55-7ce1628a406f  base
xgboost_0.82-py3.6              39e31acd-5f30-41dc-ae44-60233c80306e  base
pytorch-onnx_1.2-py3.6-edt      40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
pytorch-onnx_rt22.2-py3.10      40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38                 41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9          4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                  42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
pmml-3.0_4.3                    493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
spark-mllib_2.4-r_3.6           49403dff-92e9-4c87-a3d7-a42d0021c095  base
xgboost_0.90-py3.6              4ff8d6c2-1343-4c18-85e1-689c965304d3  base
pytorch-onnx_1.1-py3.6          50f95b2a-bc16-43bb-bc94-b0bed208c60b  base
autoai-ts_3.9-py3.8             52c57136-80fa-572e-8728-a5e7cbb42cde  base
spark-mllib_2.4-scala_2.11      55a70f99-7320-4be5-9fb9-9edb5a443af5  base
spark-mllib_3.0                 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
autoai-obm_2.0                  5c2e37fa-80b8-5e77-840f-d912469614ee  base
spss-modeler_18.1               5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
cuda-py3.8                      5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
autoai-kb_3.1-py3.7             632d4b22-10aa-5180-88f0-f52dfb6444d7  base
```

9s completed at 4:50 PM

train the vegetable model.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share G

+ Code + Text

RAM Disk Editing

[11]
```
autoai-obm_2.0                5c2e37fa-80b8-5e77-840f-d912469614ee  base
spss-modeler_18.1             5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
cuda-py3.8                    5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
autoai-kb_3.1-py3.7           632d4b22-10aa-5180-88f0-f52dfb6444d7  base
pytorch-onnx_1.7-py3.8        634d3cdc-b562-5bf9-a2d4-ea90a478456b  base
----------------------------  ------------------------------------  ----
Note: Only first 50 records were displayed. To display more use 'limit' parameter.
```

[12]
```
software_space_uid = client.software_specifications.get_uid_by_name("tensorflow_rt22.1-py3.9")
software_space_uid
```

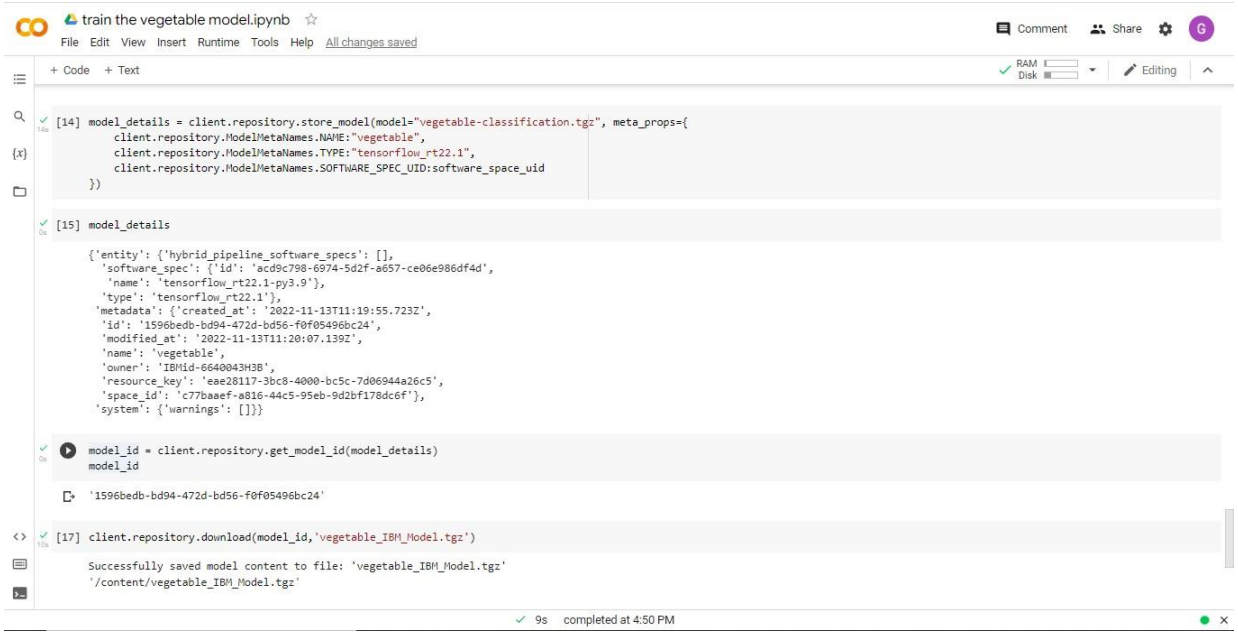
```
'acd9c798-6974-5d2f-a657-ce06e986df4d'
```

[14]
```
model_details = client.repository.store_model(model="vegetable-classification.tgz", meta_props={
    client.repository.ModelMetaNames.NAME:"vegetable",
    client.repository.ModelMetaNames.TYPE:"tensorflow_rt22.1",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
})
```

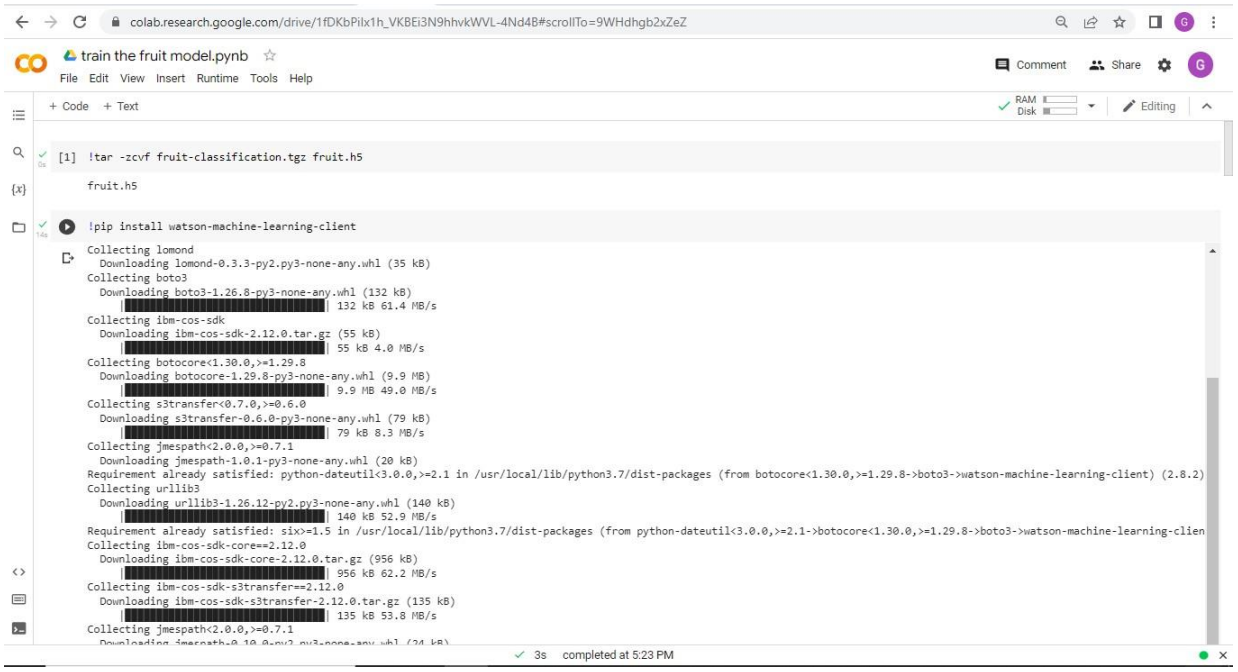
[15]
```
model_details
```

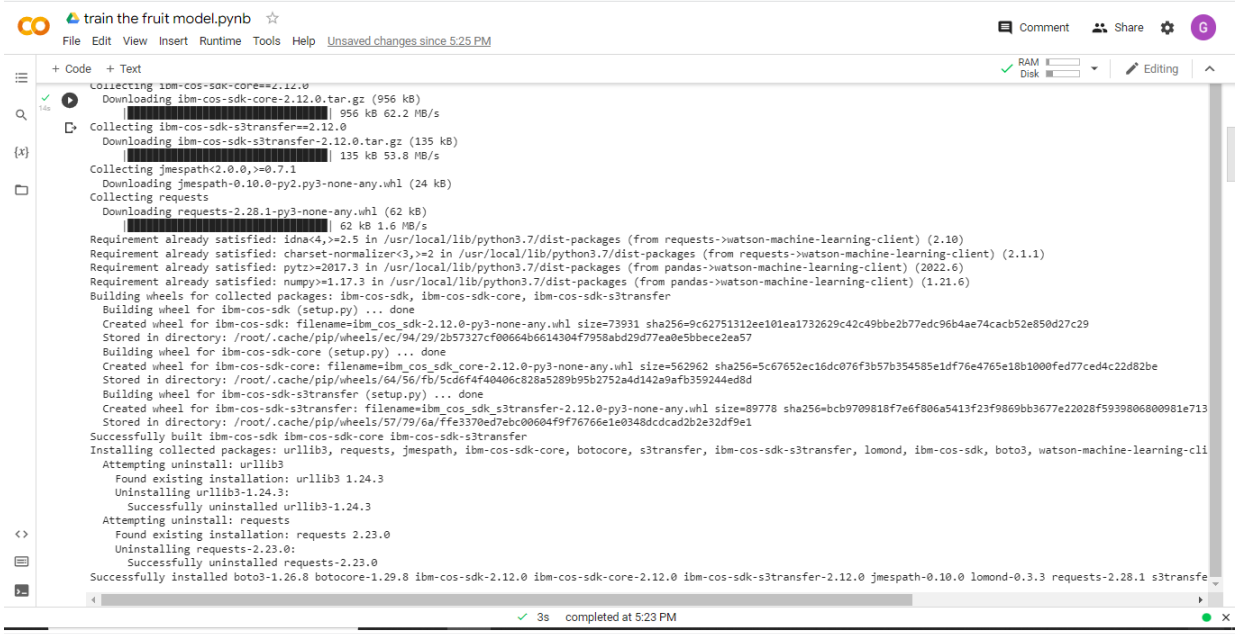

```
{'entity': {'hybrid_pipeline_software_specs': [],
  'software_spec': {'id': 'acd9c798-6974-5d2f-a657-ce06e986df4d',
   'name': 'tensorflow_rt22.1-py3.9'},
  'type': 'tensorflow_rt22.1'},
 'metadata': {'created_at': '2022-11-13T11:19:55.723Z',
  'id': '1596bedb-bd94-472d-bd56-f0f05496bc24',
  'modified_at': '2022-11-13T11:20:07.139Z',
  'name': 'vegetable',
  'owner': 'IBMid-6640043H3B',
  'resource_key': 'eae28117-3bc8-4000-bc5c-7d06944a26c5',
  'space_id': 'c77baaef-a816-44c5-95eb-9d2bf178dc6f'},
 'system': {'warnings': []}}
```


9s completed at 4:50 PM



TRAIN THE FRUIT MODEL ON IBM




train the fruit model.pynb
File Edit View Insert Runtime Tools Help Unsaved changes since 5:25 PM
Comment
Share
+ Code + Text
RAM
Disk
Editing
Collecting ibm-cos-sdk-core==2.12.0
Downloading ibm-cos-sdk-core-2.12.0.tar.gz (956 kB)
| 956 kB 62.2 MB/s
Collecting ibm-cos-sdk-s3transfer==2.12.0
Downloading ibm-cos-sdk-s3transfer-2.12.0.tar.gz (135 kB)
| 135 kB 53.8 MB/s
Collecting jmespath<2.0.0,>=0.7.1
Downloading jmespath-0.10.0-py2.py3-none-any.whl (24 kB)
Collecting requests
Downloading requests-2.28.1-py3-none-any.whl (62 kB)
| 62 kB 1.6 MB/s
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->watson-machine-learning-client) (2.10)
Requirement already satisfied: charset-normalizer<3,>=2 in /usr/local/lib/python3.7/dist-packages (from requests->watson-machine-learning-client) (2.1.1)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas->watson-machine-learning-client) (2022.6)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packages (from pandas->watson-machine-learning-client) (1.21.6)
Building wheels for collected packages: ibm-cos-sdk, ibm-cos-sdk-core, ibm-cos-sdk-s3transfer
Building wheel for ibm-cos-sdk (setup.py) ... done
Created wheel for ibm-cos-sdk: filename=ibm_cos_sdk-2.12.0-py3-none-any.whl size=73931 sha256=9c62751312ee101ea1732629c42c49bbe2b77edc96b4ae74cacb52e850d27c29
Stored in directory: /root/.cache/pip/wheels/ec/94/29/2b57327cf00664b6614304f7958abd29d77ea0e5bbece2ea57
Building wheel for ibm-cos-sdk-core (setup.py) ... done
Created wheel for ibm-cos-sdk-core: filename=ibm_cos_sdk_core-2.12.0-py3-none-any.whl size=562962 sha256=5c67652ec16dc076f3b57b354585e1df76e4765e18b1000fed77ced4c22d82be
Stored in directory: /root/.cache/pip/wheels/64/56/fb/5cd6f4f40406c828a5289b95b2752a4d142a9afb359244ed8d
Building wheel for ibm-cos-sdk-s3transfer (setup.py) ... done
Created wheel for ibm-cos-sdk-s3transfer: filename=ibm_cos_sdk_s3transfer-2.12.0-py3-none-any.whl size=89778 sha256=bcb9709818f7e6f806a5413f23f9869bb3677e22028f5939806800981e713
Stored in directory: /root/.cache/pip/wheels/57/79/6a/ffe3370ed7ebc00604f9f76766e1e0348dcdcad2b2e32df9e1
Successfully built ibm-cos-sdk ibm-cos-sdk-core ibm-cos-sdk-s3transfer
Installing collected packages: urllib3, requests, jmespath, ibm-cos-sdk-core, botocore, s3transfer, ibm-cos-sdk-s3transfer, lomond, ibm-cos-sdk, boto3, watson-machine-learning-cli
Attempting uninstall: urllib3
Found existing installation: urllib3 1.24.3
Uninstalling urllib3-1.24.3:
Successfully uninstalled urllib3-1.24.3
Attempting uninstall: requests
Found existing installation: requests 2.23.0
Uninstalling requests-2.23.0:
Successfully uninstalled requests-2.23.0
Successfully installed boto3-1.26.8 botocore-1.29.8 ibm-cos-sdk-2.12.0 ibm-cos-sdk-core-2.12.0 ibm-cos-sdk-s3transfer-2.12.0 jmespath-0.10.0 lomond-0.3.3 requests-2.28.1 s3transfe
3s completed at 5:23 PM


train the fruit model.pynb
File Edit View Insert Runtime Tools Help Unsaved changes since 5:25 PM
Comment
Share
+ Code + Text
RAM
Disk
Editing
!pip install ibm_watson_machine_learning
| 51 kB 622 kB/s
Requirement already satisfied: lomond in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (0.3.3)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (21.3)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (2.28.1)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (4.13.0)
Requirement already satisfied: tabulate in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (0.8.10)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (1.3.5)
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (2022.9.24)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (from ibm_watson_machine_learning) (1.26.12)
Collecting ibm-cos-sdk-core==2.7.0
Downloading ibm-cos-sdk-core-2.7.0.tar.gz (824 kB)
| 824 kB 19.0 MB/s
Collecting ibm-cos-sdk-s3transfer==2.7.0
Downloading ibm-cos-sdk-s3transfer-2.7.0.tar.gz (133 kB)
| 133 kB 61.8 MB/s
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /usr/local/lib/python3.7/dist-packages (from ibm-cos-sdk==2.7.*->ibm_watson_machine_learning) (0.10.0)
Collecting docutils<0.16,>=0.10
Downloading docutils-0.15.2-py3-none-any.whl (547 kB)
| 547 kB 62.9 MB/s
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/python3.7/dist-packages (from ibm-cos-sdk-core==2.7.0->ibm-cos-sdk==2.7.*->ibm_watson_machine_learning
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (1.21.6)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (2022.6)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.7.0->ibm-cos-sdk==2.7.*->ibm_watson_machin
Requirement already satisfied: charset-normalizer<3,>=2 in /usr/local/lib/python3.7/dist-packages (from requests->ibm_watson_machine_learning) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->ibm_watson_machine_learning) (2.10)
Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->ibm_watson_machine_learning) (4.1.1)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->ibm_watson_machine_learning) (3.10.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->ibm_watson_machine_learning) (3.0.9)
Building wheels for collected packages: ibm-cos-sdk, ibm-cos-sdk-core, ibm-cos-sdk-s3transfer
Building wheel for ibm-cos-sdk (setup.py) ... done
Created wheel for ibm-cos-sdk: filename=ibm_cos_sdk-2.7.0-py2.py3-none-any.whl size=72563 sha256=fa8c58715e9ddfd8350537fcfbf2acb45d7a6b69eae3a282aeb718abe8e56577
Stored in directory: /root/.cache/pip/wheels/47/22/bf/e1154ff0f5de93cc477acd0ca69abfbb8b799c5b28a66b44c2
Building wheel for ibm-cos-sdk-core (setup.py) ... done
Created wheel for ibm-cos-sdk-core: filename=ibm_cos_sdk_core-2.7.0-py2.py3-none-any.whl size=501013 sha256=3fc3c0f924895c68b671b029e8383b69c72ca3a5c5c132c3127a0d208bff9d7b
Stored in directory: /root/.cache/pip/wheels/6c/a2/e4/c16d02f809a3ea998e17cfd02c13369281f3d232aaf5902c19
3s completed at 5:23 PM

train the fruit model.pynb

File Edit View Insert Runtime Tools Help Saving...

Comment Share G

+ Code + Text

RAM Disk Editing

```
[18] from ibm_watson_machine_learning import APIClient
     wml_credentials ={
         "url":"https://eu-gb.ml.cloud.ibm.com",
         "apikey":"AQ_xcScv7Q5JrYnfPbfimSyAc6YWz5fpFMuU-V8mdFlP"
     }
     client = APIClient(wml_credentials)
```

```
Python 3.7 and 3.8 frameworks are deprecated and will be removed in a future release. Use Python 3.9 framework instead.
```

```
[19] client
```

```
<ibm_watson_machine_learning.client.APIClient at 0x7fb57312f350>
```

```
client.spaces.get_details()
```

```
        'bucket_region': 'eu-gb-standard',
        'credentials': {'admin': {'access_key_id': 'e827a1d463054462be58c11d21f4a342',
          'api_key': 'hIknNxBOWj461q3pDFSLZ-NveCEz7cnqSm8c0lQa4LG3',
          'secret_access_key': 'e730510472c7fed75868de0d665b713b701909a93da86a73',
          'service_id': 'ServiceId-a30f82e7-316e-46bd-b229-331030406a23'},
         'editor': {'access_key_id': '004f08db143c4550b25166a8f5271771',
          'api_key': '9Ysax1Y9-XojQaS6GwvJZKf64HyFtwcxf67ncD-jrXqu',
          'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/d2a512c7590048cfaf7f28cd26438f49:0e43b750-15aa-4634-9c02-9954d7f0be49::',
          'secret_access_key': '7cbb5e54037e5876c10f85a86b0acd6e6fba4b2caa4338f7',
          'service_id': 'ServiceId-68abd7ca-11d3-4e3e-b8dd-da61d9c93eae'},
         'viewer': {'access_key_id': '08ec08d406014223b26317ff7d844345',
          'api_key': 'amCM5-GBgCOFE6cmRFIl18Pz5kIVU8SFiTPFL284UmPP',
          'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/d2a512c7590048cfaf7f28cd26438f49:0e43b750-15aa-4634-9c02-9954d7f0be49::',
          'secret_access_key': 'e7c659f9ebf2790368703f119b8a0f28082cf7256833fd2d',
          'service_id': 'ServiceId-ea9f16f2-4120-4ae0-b248-f5bc5605c424'}},
        'endpoint_url': 'https://s3.eu-gb.cloud-object-storage.appdomain.cloud',
        'guid': '0e43b750-15aa-4634-9c02-9954d7f0be49',
        'resource_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/d2a512c7590048cfaf7f28cd26438f49:0e43b750-15aa-4634-9c02-9954d7f0be49::'},
       'type': 'bmcos_object_storage'}},
```

3s completed at 5:23 PM

train the fruit model.pynb

File Edit View Insert Runtime Tools Help Saving...

Comment Share G

+ Code + Text

RAM Disk Editing

```
       'url': '/v2/spaces/c77baaef-a816-44c5-95eb-9d2bf178dc6f'}},
 {'entity': {'compute': [{'crn': 'crn:v1:bluemix:public:pm-20:eu-gb:a/d2a512c7590048cfaf7f28cd26438f49:f78a826e-3c47-40ec-9ced-ab40e6d544d3::',
       'guid': 'f78a826e-3c47-40ec-9ced-ab40e6d544d3',
       'name': 'Watson Machine Learning-ak',
       'type': 'machine_learning'}],
     'description': '',
     'name': 'f1',
     'scope': {'bss_account_id': 'd2a512c7590048cfaf7f28cd26438f49'},
     'stage': {'production': False},
     'status': {'state': 'active'},
     'storage': {'properties': {'bucket_name': 'f4356370-41d7-4406-9b54-9eaddfb02d12',
       'bucket_region': 'eu-gb-standard',
       'credentials': {'admin': {'access_key_id': '0345751f31684f7d988cf9bd2d9a633c',
         'api_key': 'fIncMb_KsJjqr-XfOuAO30EfqYvsz-c_Qgu4yJrpVcGQ',
         'secret_access_key': '0bb6e5186f2aee7525776c876cfd74f9dcec8c50f0dea71c',
         'service_id': 'ServiceId-560fac55-10b4-494d-983d-15821b101c02'},
        'editor': {'access_key_id': 'f016e464721b4db2a0fbaa1879de673a',
         'api_key': 'H8OTzyHWiovMdg6d0cnT9NYZnQENRbjbQqpWbaUYA9KF',
         'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/d2a512c7590048cfaf7f28cd26438f49:0e43b750-15aa-4634-9c02-9954d7f0be49::',
         'secret_access_key': '921110b4f02600f7b655f740b7de3af0b5620bd3f88b2108',
         'service_id': 'ServiceId-c6333f38-a775-4ffe-8461-f9451f8ce259'},
        'viewer': {'access_key_id': '2ae1637c88ce48bfa3e80cdd8b161a4d',
         'api_key': 'U85JPKfPKai2FO8yFpBh0FwJnZC4uBF_vTGzORLnjKJU',
         'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/d2a512c7590048cfaf7f28cd26438f49:0e43b750-15aa-4634-9c02-9954d7f0be49::',
         'secret_access_key': '348290e05e1c239dbc638225fd69349dee370472db875bb8',
         'service_id': 'ServiceId-16fbfab2-dfd6-4c87-9219-03d0470377db'}},
       'endpoint_url': 'https://s3.eu-gb.cloud-object-storage.appdomain.cloud',
       'guid': '0e43b750-15aa-4634-9c02-9954d7f0be49',
       'resource_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/d2a512c7590048cfaf7f28cd26438f49:0e43b750-15aa-4634-9c02-9954d7f0be49::'},
      'type': 'bmcos_object_storage'}},
    'metadata': {'created_at': '2022-11-13T11:47:24.278Z',
     'creator_id': 'IBMid-6640043H3B',
     'id': '4ad0fa56-67ce-4b1a-b30f-e8035c8149b8',
     'updated_at': '2022-11-13T11:47:37.954Z',
     'url': '/v2/spaces/4ad0fa56-67ce-4b1a-b30f-e8035c8149b8'}}]}
```

```
[21] client.spaces.list()
```

3s completed at 5:23 PM

train the fruit model.pynb

File Edit View Insert Runtime Tools Help Saving...

Comment Share G

+ Code + Text

RAM Disk Editing

```
[21] client.spaces.list()

Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
------------------------------------  ---------  -------------------------
ID                                    NAME       CREATED
4ad0fa56-67ce-4b1a-b30f-e8035c8149b8  f1         2022-11-13T11:47:24.278Z
c77baaef-a816-44c5-95eb-9d2bf178dc6f  v1_Deploy  2022-11-13T10:24:33.382Z
------------------------------------  ---------  -------------------------

[22] space_uid = "4ad0fa56-67ce-4b1a-b30f-e8035c8149b8"
     space_uid

     '4ad0fa56-67ce-4b1a-b30f-e8035c8149b8'

[23] client.set.default_space(space_uid)

     'SUCCESS'

[24] client.software_specifications.list()

     ------------------------------  ------------------------------------  ----
     NAME                            ASSET_ID                              TYPE
     default_py3.6                   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
     kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a  base
     pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d288  base
     scikit-learn_0.20-py3.6         09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
     spark-mllib_3.0-scala_2.12      09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
     pytorch-onnx_rt22.1-py3.9       0b848dd4-e681-5599-be41-b5f6fccc6471  base
     ai-function_0.1-py3.6           0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
     shiny-r3.6                      0e6e79df-875e-4f24-8ae9-62dcc2148306  base
     tensorflow_2.4-py3.7-horovod    1092590a-307d-563d-9b62-4eb7d64b3f22  base
     pytorch_1.1-py3.6               10ac12d6-6b30-4ccd-8392-3e922c096a92  base
     tensorflow_1.15-py3.6-ddl       111e41b3-de2d-5422-a4d6-bf776828c4b7  base
```

3s completed at 5:23 PM

train the fruit model.pynb

File Edit View Insert Runtime Tools Help Saving...

Comment Share G

+ Code + Text

RAM Disk Editing

```
[24] kernel-spark3.3-r3.6            1c9e5454-f216-59dd-a20e-474a5cdf5988  base
     pytorch-onnx_rt22.1-py3.9-edt   1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
     tensorflow_2.1-py3.6            1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
     spark-mllib_3.2                 20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
     tensorflow_2.4-py3.8-horovod    217c16f6-178f-56bf-824a-b19f20564c49  base
     runtime-22.1-py3.9-cuda         26215f05-08c3-5a41-a1b0-da66306ce658  base
     do_py3.8                        295addb5-9ef9-547e-9bf4-92ae3563e720  base
     autoai-ts_3.8-py3.8             2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
     tensorflow_1.15-py3.6           2b73a275-7cbf-420b-a912-eae7f436e0bc  base
     kernel-spark3.3-py3.9           2b7961e2-e3b1-5a8c-a491-482c8368839a  base
     pytorch_1.2-py3.6               2c8ef57d-2687-4b7d-acce-01f94976dac1  base
     spark-mllib_2.3                 2e51f700-bca0-4b0d-88dc-5c6791338875  base
     pytorch-onnx_1.1-py3.6-edt      32983cea-3f32-4400-8965-dde874a8d67e  base
     spark-mllib_3.0-py37            36507ebe-8770-55ba-ab2a-eafe787600e9  base
     spark-mllib_2.4                 390d21f8-e58b-4fac-9c55-d7ceda621326  base
     autoai-ts_rt22.2-py3.10         396b2e83-0953-5b86-9a55-7ce1628a406f  base
     xgboost_0.82-py3.6              39e31acd-5f30-41dc-ae44-60233c80306e  base
     pytorch-onnx_1.2-py3.6-edt      40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
     pytorch-onnx_rt22.2-py3.10      40e73f55-783a-5535-b3fa-0c8b94291431  base
     default_r36py38                 41c247d3-45f8-5a71-b065-8580229facf0  base
     autoai-ts_rt22.1-py3.9          4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
     autoai-obm_3.0                  42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
     pmml-3.0_4.3                    493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
     spark-mllib_2.4-r_3.6           49403dff-92e9-4c87-a3d7-a42d0021c095  base
     xgboost_0.90-py3.6              4ff8d6c2-1343-4c18-85e1-689c965304d3  base
     pytorch-onnx_1.1-py3.6          50f95b2a-bc16-43bb-bc94-b0bed208c60b  base
     autoai-ts_3.9-py3.8             52c57136-80fa-572e-8728-a5e7cbb42cde  base
     spark-mllib_2.4-scala_2.11      55a70f99-7320-4be5-9fb9-9edb5a443af5  base
     spark-mllib_3.0                 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
     autoai-obm_2.0                  5c2e37fa-80b8-5e77-840f-d912469614ee  base
     spss-modeler_18.1               5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
     cuda-py3.8                      5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
     autoai-kb_3.1-py3.7             632d4b22-10aa-5180-88f0-f52dfb6444d7  base
     pytorch-onnx_1.7-py3.8          634d3cdc-b562-5bf9-a2d4-ea90a478456b  base
     ------------------------------  ------------------------------------  ----
     Note: Only first 50 records were displayed. To display more use 'limit' parameter.
```

3s completed at 5:23 PM

train the fruit model.pynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share G

+ Code + Text

RAM Disk Editing

```
[25] software_space_uid = client.software_specifications.get_uid_by_name("tensorflow_rt22.1-py3.9")
     software_space_uid
```

```
'acd9c798-6974-5d2f-a657-ce06e986df4d'
```

```
model_details = client.repository.store_model(model="fruit-classification.tgz", meta_props={
    client.repository.ModelMetaNames.NAME:"fruit",
    client.repository.ModelMetaNames.TYPE:"tensorflow_rt22.1",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
})
```

```
[27] model_details
```

```
{'entity': {'hybrid_pipeline_software_specs': [],
  'software_spec': {'id': 'acd9c798-6974-5d2f-a657-ce06e986df4d',
   'name': 'tensorflow_rt22.1-py3.9'},
  'type': 'tensorflow_rt22.1'},
 'metadata': {'created_at': '2022-11-13T11:52:22.919Z',
  'id': '7656b11c-b4d6-4c77-83aa-bce9052215df',
  'modified_at': '2022-11-13T11:52:27.272Z',
  'name': 'fruit',
  'owner': 'IBMid-6640043H3B',
  'resource_key': 'a1e6d912-974d-477f-8809-2ab9871afe44',
  'space_id': '4ad0fa56-67ce-4b1a-b30f-e8035c8149b8'},
 'system': {'warnings': []}}
```

```
[28] model_id = client.repository.get_model_id(model_details)
     model_id
```

3s completed at 5:23 PM

fruit_IBM_Model.tgz

Show all

train the fruit model.pynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share G

+ Code + Text

RAM Disk Editing

```
[26] })
```

```
model_details
```

```
{'entity': {'hybrid_pipeline_software_specs': [],
  'software_spec': {'id': 'acd9c798-6974-5d2f-a657-ce06e986df4d',
   'name': 'tensorflow_rt22.1-py3.9'},
  'type': 'tensorflow_rt22.1'},
 'metadata': {'created_at': '2022-11-13T11:52:22.919Z',
  'id': '7656b11c-b4d6-4c77-83aa-bce9052215df',
  'modified_at': '2022-11-13T11:52:27.272Z',
  'name': 'fruit',
  'owner': 'IBMid-6640043H3B',
  'resource_key': 'a1e6d912-974d-477f-8809-2ab9871afe44',
  'space_id': '4ad0fa56-67ce-4b1a-b30f-e8035c8149b8'},
 'system': {'warnings': []}}
```

```
[28] model_id = client.repository.get_model_id(model_details)
     model_id
```

```
'7656b11c-b4d6-4c77-83aa-bce9052215df'
```

```
[30] client.repository.download(model_id,'fruit_IBM_Model.tgz')
```

```
Successfully saved model content to file: 'fruit_IBM_Model.tgz'
'/content/fruit_IBM_Model.tgz'
```

3s completed at 5:23 PM

Test both the models

Test the Fruit disease prediction model

jupyter fruit model building and testing Last Checkpoint: 30 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [22]:
```
# Testing the model
```

In [23]:
```
from tensorflow.keras.preprocessing import image
import numpy as np
```

In [29]:
```
img=image.load_img(r'C:\Users\Elcot\project\flask\sample_images\Healthy corn.jpg',grayscale=False,target_size=(128,128))
img
```

Out[29]:

In [30]:
```
x = image.img_to_array(img) # Converting image into array
x = np.expand_dims(x,axis=0) # expanding Dimensions
pred = np.argmax(model.predict(x)) # Predicting the higher probablity index
op = ['Apple___Black_rot','Apple___healthy','Corn_(maize)___healthy','Corn_(maize)___Northern_Leaf_Blight',''] # Creating list
op[pred] # List indexing with output
```

```
1/1 [==============================] - 0s 63ms/step
```

Out[30]:
```
'Corn_(maize)___Northern_Leaf_Blight'
```

In [31]:
```
img=image.load_img(r'C:\Users\ELCOT\project\flask\Sample_images\apple.jpg',grayscale=False,target_size=(128,128))
img
```

jupyter fruit model building and testing Last Checkpoint: 31 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

Out[31]:

In [32]:
```
x = image.img_to_array(img) # Converting image into array
x = np.expand_dims(x,axis=0) # expanding Dimensions
pred = np.argmax(model.predict(x)) # Predicting the higher probablity index
op = ['Apple___Black_rot','Apple___healthy','Corn_(maize)___healthy','Corn_(maize)___Northern_Leaf_Blight',''] # Creating list
op[pred] # List indexing with output
```

```
1/1 [==============================] - 0s 128ms/step
```

Out[32]:
```
'Apple___healthy'
```

In [34]:
```
x_train.class_indices
```

Out[34]:
```
{'Apple___Black_rot': 0,
 'Apple___healthy': 1,
 'Corn_(maize)___Northern_Leaf_Blight': 2,
 'Corn_(maize)___healthy': 3,
 'Peach___Bacterial_spot': 4,
 'Peach___healthy': 5}
```

Test the Vegetable disease prediction model

jupyter Test the vegetable model Last Checkpoint: 25 minutes ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [5]:
```
from keras.preprocessing import image
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
import numpy as nps
```

In [8]:
```
model=load_model(r'C:\Users\Elcot\project\flask\uploads\vegetable.h5')
```

In [9]:
```
img=image.load_img(r'C:\Users\Elcot\project\flask\sample_images\iqptredf.jpg',grayscale=False,target_size=(128,128))
img
```

Out[9]:

In [11]:
```
x = image.img_to_array(img) # Converting image into array
x = nps.expand_dims(x,axis=0) # expanding Dimensions
pred = nps.argmax(model.predict(x)) # Predicting the higher probablity index
op = ['Pepper,_bell___Bacterial_spot','Pepper,_bell___healthy','Potato___Early_blight','Potato___healthy','Potato___Late_blight',
op[pred] # List indexing with output
```

```
1/1 [==============================] - 104s 104s/step
```

Out[11]: 'Pepper,_bell___Bacterial_spot'

13.2. GitHub & Project Demo Link

Github link

https://github.com/IBM-EPBL/IBM-Project-31930-1660206641

Demo video link

https://drive.google.com/drive/folders/1A7e4mbQFkz9VIA42gmnwnS_ulpI2kt8L