Fertilizers Recommendation System For Disease Prediction

1. Introduction:

1.1 Overview:

Agriculture is the most important sector in today's life. Most plants are affected by a wide variety of bacterial and fungal diseases. Diseases on plants placed a major constraint on the production and a major threat to food security. Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality. In recent years, the number of diseases on plants and the degree of harm caused has increased due to the variation in pathogen varieties, changes in cultivation methods, and inadequate plant protection techniques.

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

Deep learningtechniques are used to identifythe diseases and suggest the precautions that can be taken for those diseases.

1.2 Purpose:

To Detectand recognize the plant diseasesand to **recommendfertilizer**,it is necessaryto provide symptoms in identifying the disease at its earliest. Hence the authors proposed and implemented new **fertilizers Recommendation System** for crop **disease prediction**.

2.LITERATURE SURVEY:

2.1 Existing problem:

Adequate mineralnutrition is centralto crop production.

However, it can also exert considerable Influence on disease development.Fertilizer application can increase or decrease development of diseases caused by different pathogens, and the mechanisms responsible are complex, including effects of nutrients on plant growth, plant resistance mechanisms and direct effects on the pathogen. The effects of mineral nutritionon plant diseaseand the mechanisms responsible for those effectshave been dealt with comprehensively elsewhere.In India, around 40% of land is kept and grown using reliable irrigation technologies, while the rest relies on the monsoon environment for water. Irrigation decreases reliance on the monsoon, increases food security, and boosts agricultural production.

Most research articles use humidity, moisture, and temperaturesensors near the plant's root, with an external device handling all of the data provided by the sensors and transmitting it directly to an external display or an Android application. The application was created to measure the approximate values of temperature, humidity,and moisture sensorsthatwere programmed into a microcontroller to manage the amount of water.

2.2 Proposed solution:

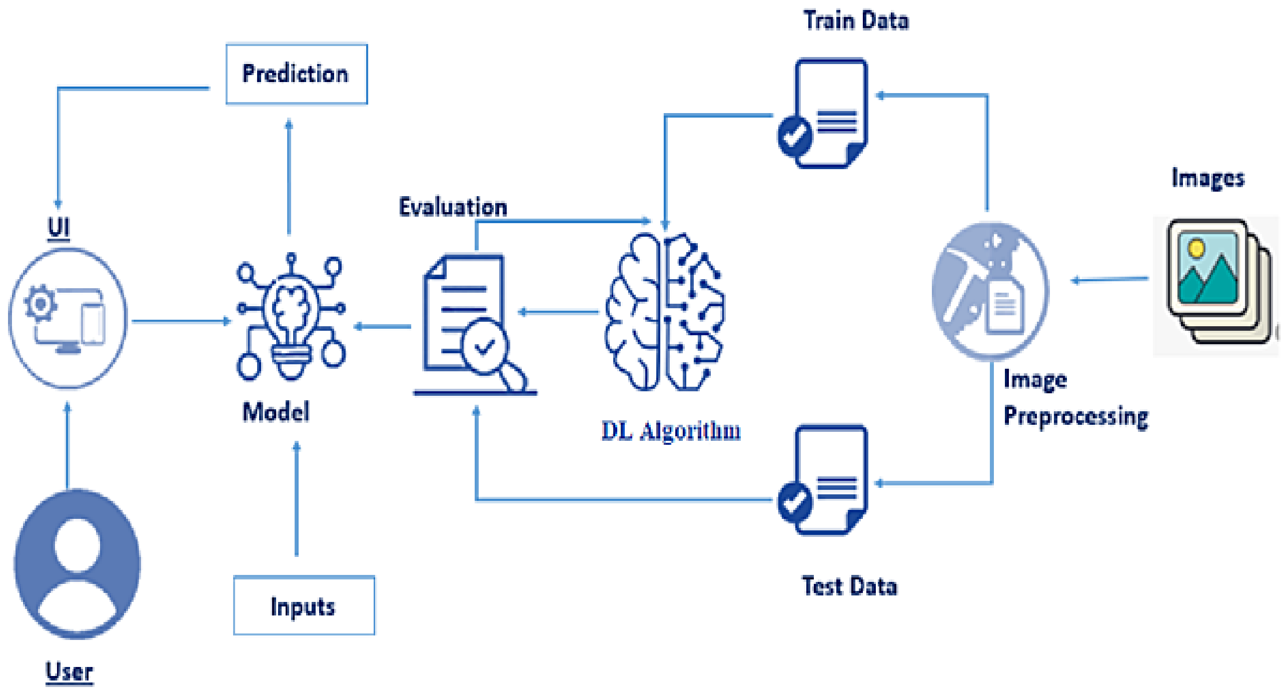
A Web Application built where :

1. Farmers interact with the portal build
2.Interacts with theuser interface to upload imagesof diseased leaf
3.Our model builtanalyses the Diseaseand suggests the farmer withfertilizers are to be used

3. THEORITICAL ANALYSIS:

3.1 Block diagram:

- A web Application is built where
- Farmers interact with the portal build
- Interacts with the user interface to upload images of diseased leaf
- Our model built analyses the Disease and suggests the farmer with fertilizers are to be used
- To accomplish the above task you must complete the below activities and tasks
- Download the dataset.
- Classify the dataset into train and test sets.
- Add the neural network layers.
- Load the trained images and fit the model.
- Test the model.
- Save the model and its dependencies.
- Build a Web application using a flask that integrates with the model built.



3.2 **Hardware / Software designing:**

To completethis project you should have the following software and packages.

Softwares:

- **> Anaconda Navigator**
- **> spyder**
- **> visual studio code**
- **> Jupiter notebook**
- **> IBM watson studio**

Packages:

- **> Tensorflow**
- **> Keras**

> Flask
> numpy
> Pandas

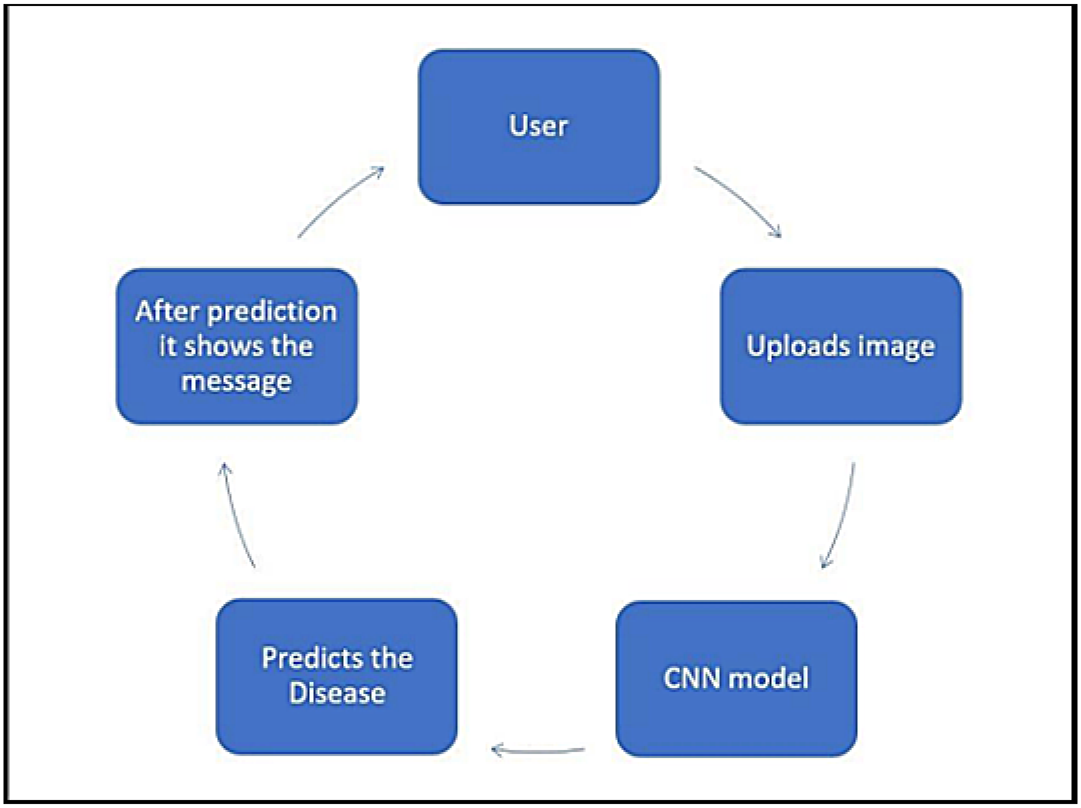
By using the above listed softwares and packages ,we build this application to take the input (image) from the farmer and detects whether theplant is infected or not. Here we use Deep learning techniques and give the out put to the user (Farmer).

4. EXPERIMENTAL INVESTIGATIONS :

Agriculture is the most important sector in today'slife. Most plantsareaffected by a wide variety of bacterial and fungal diseases. Diseases on plants placed a major constraint on the production and a major threat to food security. Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality. In recent years, the number of diseases on plants and the degree of harm caused has increased due to the variation in pathogen varieties, changes in cultivation methods,and inadequate plant protection techniques.

An automated system is introduced to identify different diseases onplants by checking the symptoms shown on the leaves of the plant. Deeplearning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases. Analysis or the investigation made while workingon the solution.

5. FLOWCHART :


User
Uploads image
CNN model
Predicts the Disease
After prediction it shows the message

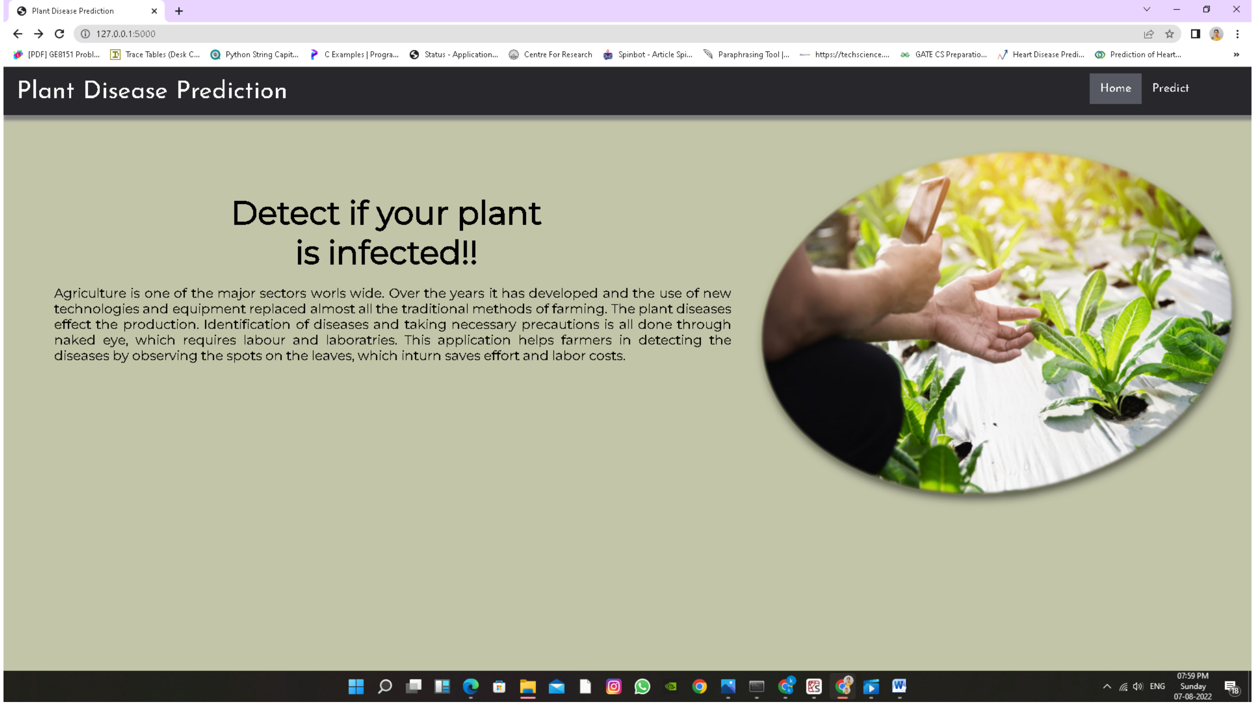
To accomplish the above task you must complete the below activitiesand tasks :

1. Download the dataset.
2. Classify the dataset into train and test sets.
3. Add the neural network layers.
4. Load the trained imagesand fit the model.
5. Test the model.
6. Save the model and its dependencies.

Build a Web application using a flask that integrates with the modelbuilt.

6. RESULT:

HomePage:

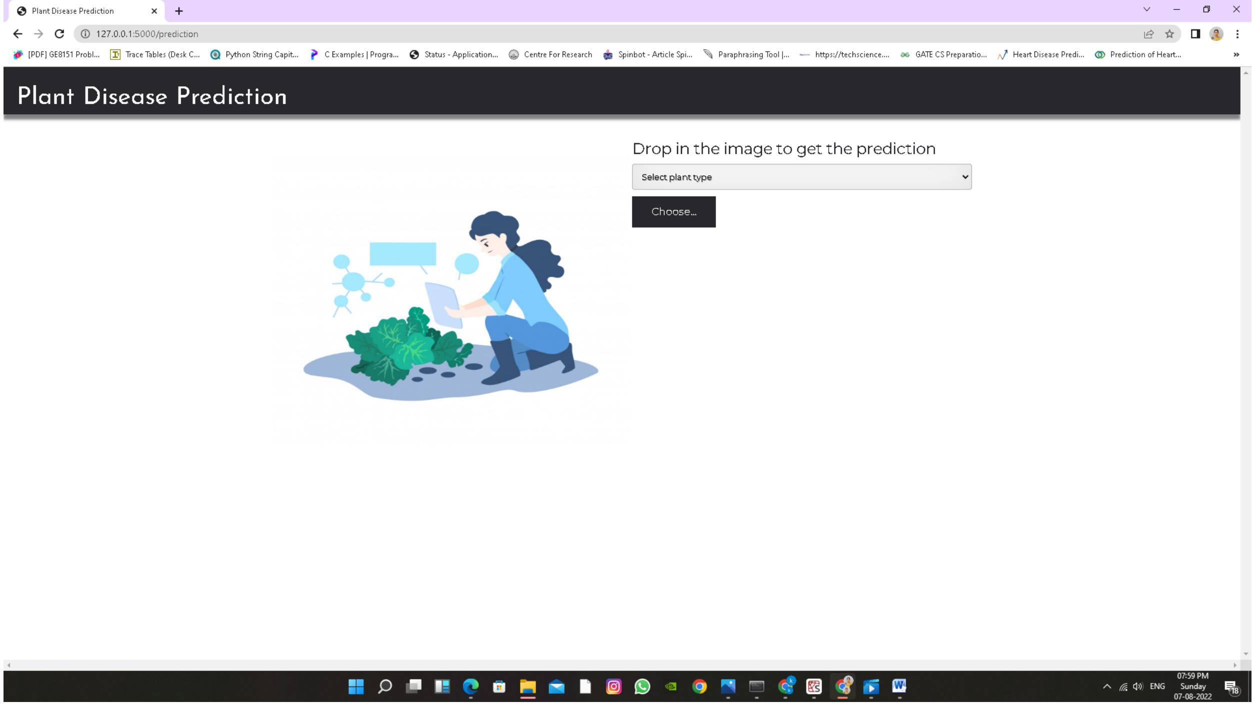


Plant Disease Prediction

Detect if your plant
is infected!!

Agriculture is one of the major sectors worls 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.

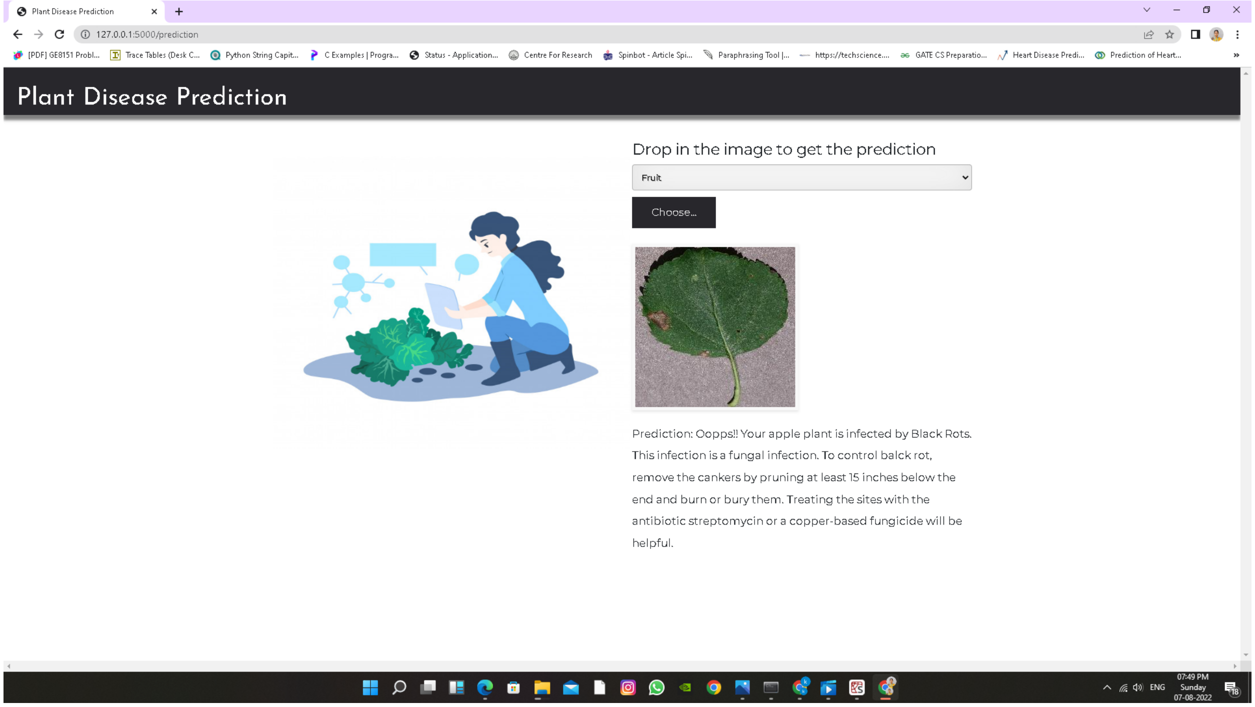
Prediction Page:



Plant Disease Prediction

Drop in the image to get the prediction

Result Page:



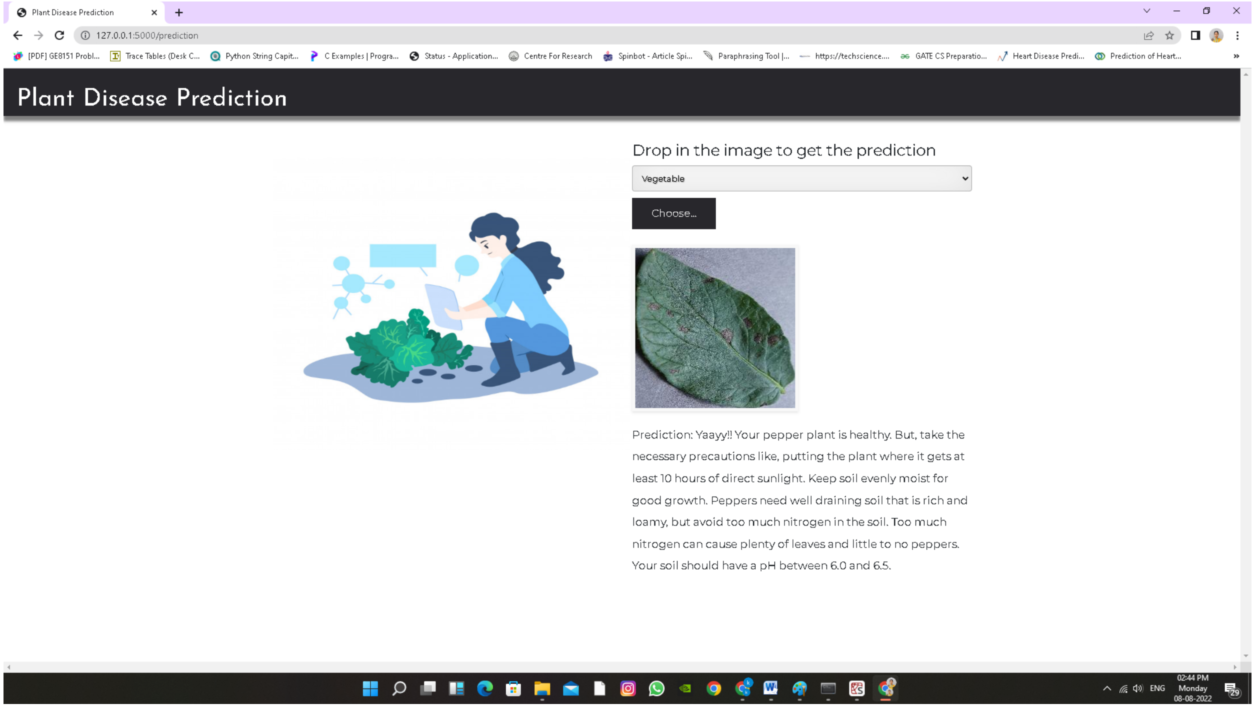
Plant Disease Prediction

Drop in the image to get the prediction

Fruit

Choose...

Prediction: Oopps!! Your apple plant is infected by Black Rots. This infection is a fungal infection. To control balck rot, remove the cankers by pruning at least 15 inches below the end and burn or bury them. Treating the sites with the antibiotic streptomycin or a copper-based fungicide will be helpful.



Plant Disease Prediction

Drop in the image to get the prediction

Vegetable

Choose...

Prediction: Yaayy!! Your pepper plant is healthy. But, take the necessary precautions like, putting the plant where it gets at least 10 hours of direct sunlight. Keep soil evenly moist for good growth. Peppers need well draining soil that is rich and loamy, but avoid too much nitrogen in the soil. Too much nitrogen can cause plenty of leaves and little to no peppers. Your soil should have a pH between 6.0 and 6.5.

7. **ADVANTAGES & DISADVANTAGES:**

ADVANTAGES:

The proposed model could predict the disease just from the image of a particular plant

1. Easy to use UI
2. Model has some good accuracy in detecting the plant just by taking theinput(leaf).

DISADVANTAGES:

1. Prediction is limited to few plants as we havent trained all the plants .

8. APPLICATIONS :

This webapplication can be used by farmers or users to check whether their plant is infectedor not and can also show the remedy so thatthe user can take necessary precautions.

These kind of web applications can be used in the agricultural sectoras well as for small house hold plants as well.

9. CONCLUSION :

Agriculture is the most important sector in today’slife. Most plantsareaffected by a wide variety of bacterial and fungal diseases. Diseases on plants placed a major constraint on the production and a major threat to food security. Hence, early and accurate identification of plant diseases is essential to ensure high quantityand best quality.

In recentyears, the numberof diseases on plants and the degree ofharm caused has increased due to the variation in pathogen varieties, changes in cultivation methods, and inadequate plant protection techniques. Usage of such applications could help the farmers to necessaryprecautions so that they dont face any loss as such.

10. FUTURE SCOPE :

As of now we have just build the web application which apparently takes the input as an image and then predictthe out in the near future wecan develop an application which computer vision and AI techniques to predict the infection once you keep the camera near the plant or leaf this could make our projecteven more usable.

11. BIBILOGRAPHY :

http://www.ijstr.org/final-print/nov2019/Fertilizers-Recommendation-SystemFor-Disease-Prediction-In-Tree-Leave.pdf

https://www.sciencedirect.com/science/article/pii/S0168169921004245

http://www.ijetajournal.org/volume-8/issue-2/IJETA-V8I2P1.pdf

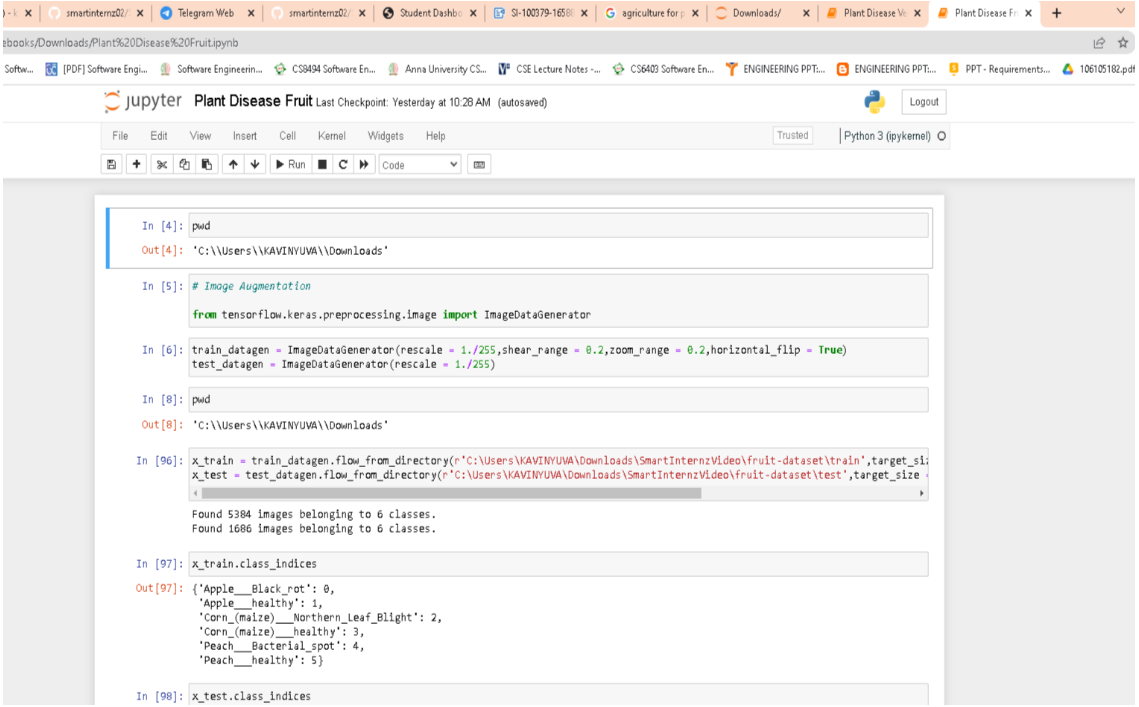
https://www.semanticscholar.org/paper/Fertilizers-Recommendation-SystemFor-Disease-In-Neela-Nithya/495379d3ef2b461fabd2de8d0605c164cb1e396f

https://ieeexplore.ieee.org/document/8878781

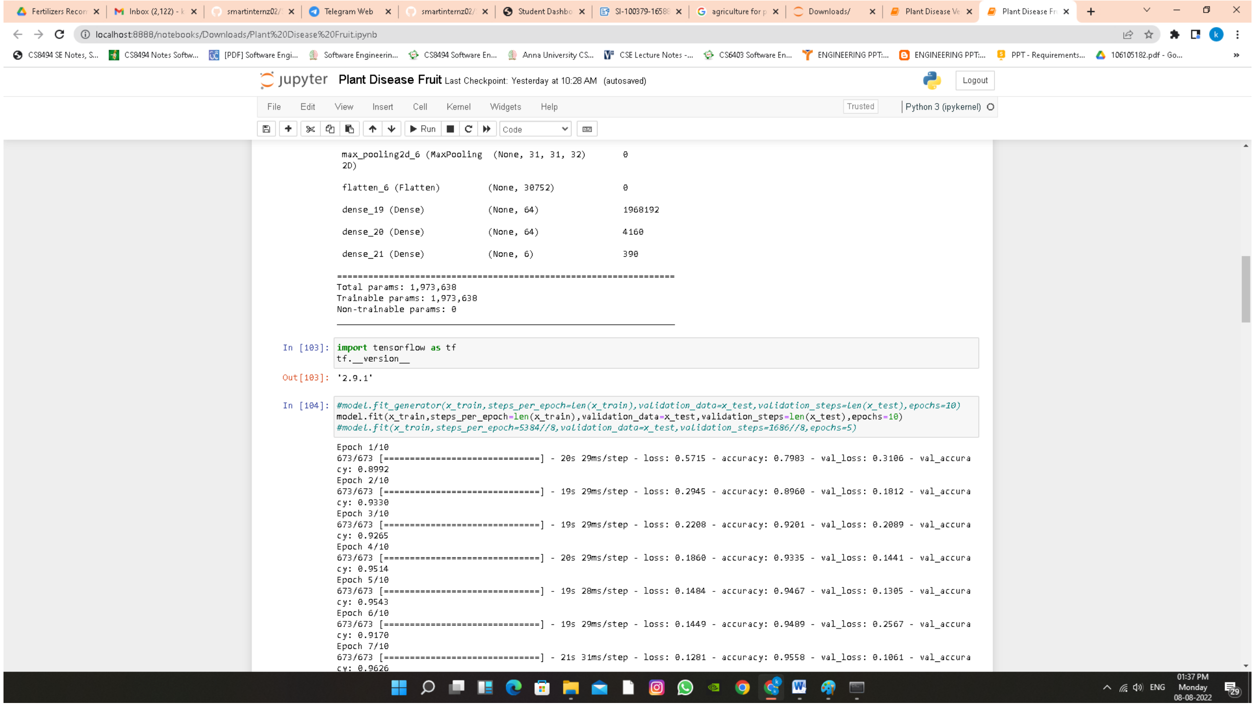
https://www.irjet.net/archives/V7/i10/IRJET-V7I1004.pdf

APPENDIX:

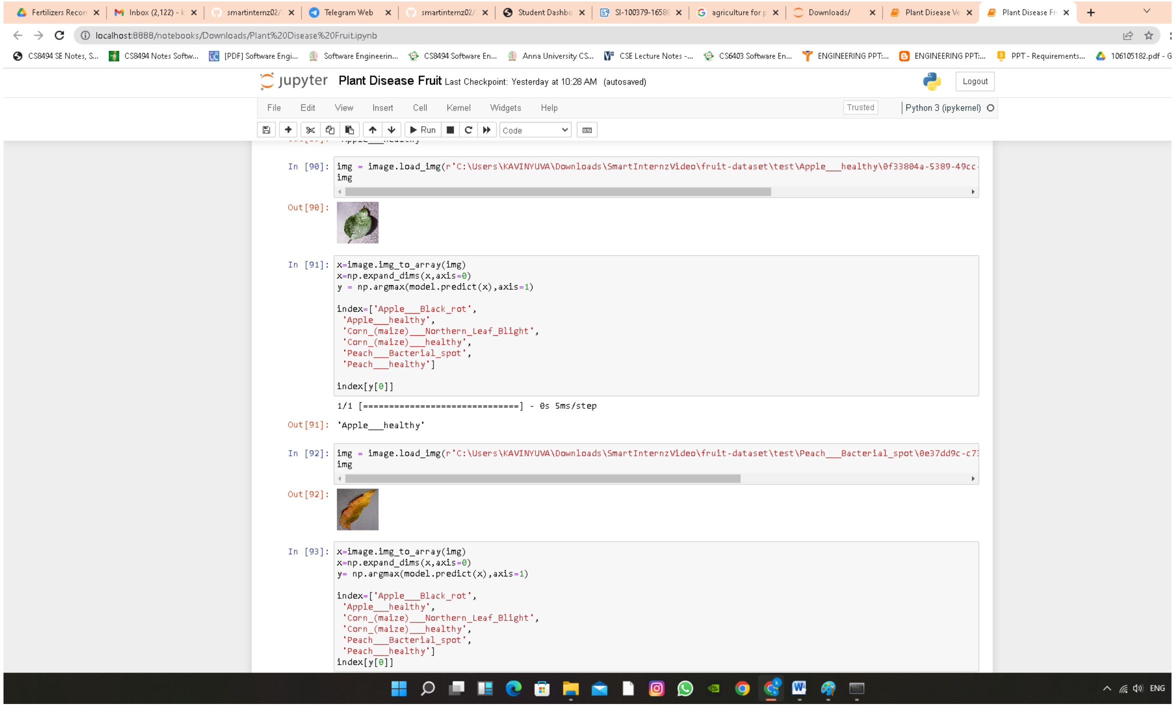
A.Source Code:


jupyter Plant Disease Fruit Last Checkpoint: Yesterday at 10:28 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [4]: pwd

Out[4]: 'C:\\Users\\KAVINYUVA\\Downloads'

In [5]: # Image Augmentation

        from tensorflow.keras.preprocessing.image import ImageDataGenerator

In [6]: train_datagen = ImageDataGenerator(rescale = 1./255,shear_range = 0.2,zoom_range = 0.2,horizontal_flip = True)
        test_datagen = ImageDataGenerator(rescale = 1./255)

In [8]: pwd

Out[8]: 'C:\\Users\\KAVINYUVA\\Downloads'

In [96]: x_train = train_datagen.flow_from_directory(r'C:\Users\KAVINYUVA\Downloads\SmartInternzVideo\fruit-dataset\train',target_si
         x_test = test_datagen.flow_from_directory(r'C:\Users\KAVINYUVA\Downloads\SmartInternzVideo\fruit-dataset\test',target_size

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

In [97]: x_train.class_indices

Out[97]: {'Apple___Black_rot': 0,
          'Apple___healthy': 1,
          'Corn_(maize)___Northern_Leaf_Blight': 2,
          'Corn_(maize)___healthy': 3,
          'Peach___Bacterial_spot': 4,
          'Peach___healthy': 5}

In [98]: x_test.class_indices
```


localhost:8888/notebooks/Downloads/Plant%20Disease%20Fruit.ipynb
jupyter Plant Disease Fruit Last Checkpoint: Yesterday at 10:28 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
01:37 PM
Monday
08-08-2022

```
 max_pooling2d_6 (MaxPooling  (None, 31, 31, 32)       0
 2D)

 flatten_6 (Flatten)         (None, 30752)             0

 dense_19 (Dense)            (None, 64)                1968192

 dense_20 (Dense)            (None, 64)                4160

 dense_21 (Dense)            (None, 6)                 390

=================================================================
Total params: 1,973,638
Trainable params: 1,973,638
Non-trainable params: 0
_________________________________________________________________

In [103]: import tensorflow as tf
          tf.__version__

Out[103]: '2.9.1'

In [104]: #model.fit_generator(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test),epochs=10)
          model.fit(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test),epochs=10)
          #model.fit(x_train,steps_per_epoch=5384//8,validation_data=x_test,validation_steps=1686//8,epochs=5)

          Epoch 1/10
          673/673 [==============================] - 20s 29ms/step - loss: 0.5715 - accuracy: 0.7983 - val_loss: 0.3106 - val_accura
          cy: 0.8992
          Epoch 2/10
          673/673 [==============================] - 19s 29ms/step - loss: 0.2945 - accuracy: 0.8960 - val_loss: 0.1812 - val_accura
          cy: 0.9330
          Epoch 3/10
          673/673 [==============================] - 19s 29ms/step - loss: 0.2208 - accuracy: 0.9201 - val_loss: 0.2089 - val_accura
          cy: 0.9265
          Epoch 4/10
          673/673 [==============================] - 20s 29ms/step - loss: 0.1860 - accuracy: 0.9335 - val_loss: 0.1441 - val_accura
          cy: 0.9514
          Epoch 5/10
          673/673 [==============================] - 19s 28ms/step - loss: 0.1484 - accuracy: 0.9467 - val_loss: 0.1305 - val_accura
          cy: 0.9543
          Epoch 6/10
          673/673 [==============================] - 19s 29ms/step - loss: 0.1449 - accuracy: 0.9489 - val_loss: 0.2567 - val_accura
          cy: 0.9170
          Epoch 7/10
          673/673 [==============================] - 21s 31ms/step - loss: 0.1281 - accuracy: 0.9558 - val_loss: 0.1061 - val_accura
          cy: 0.9626
```



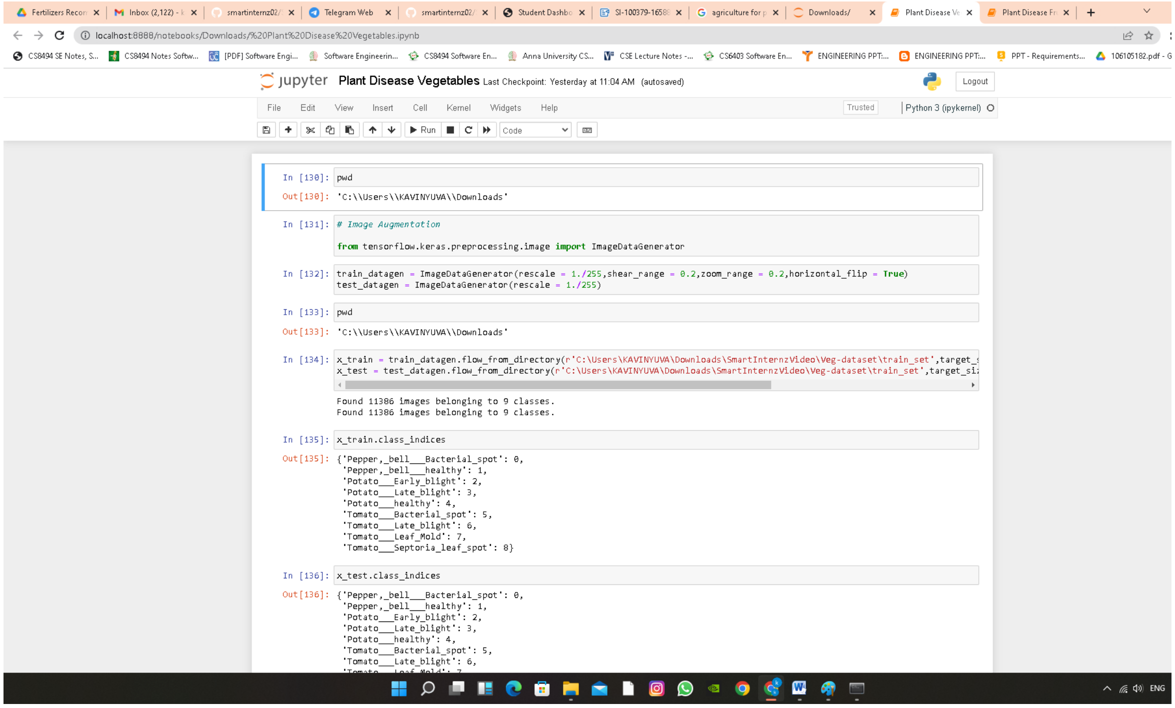
Plant Disease Fruit Last Checkpoint: Yesterday at 10:28 AM (autosaved)

```
In [90]: img = image.load_img(r'C:\Users\KAVINYUVA\Downloads\SmartInternzVideo\fruit-dataset\test\Apple___healthy\0f33804a-5389-49cc
         img

Out[90]:
```

```
In [91]: 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]]

         1/1 [==============================] - 0s 5ms/step

Out[91]: 'Apple___healthy'
```

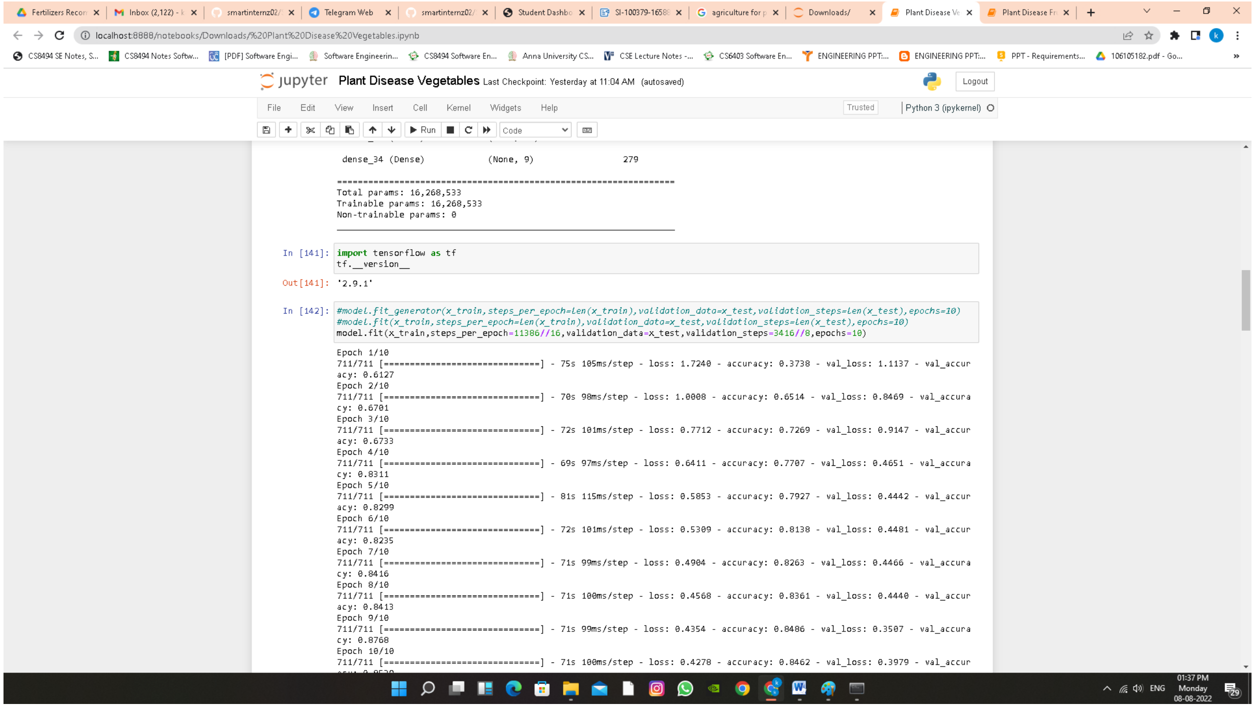
```
In [92]: img = image.load_img(r'C:\Users\KAVINYUVA\Downloads\SmartInternzVideo\fruit-dataset\test\Peach___Bacterial_spot\0e37dd9c-c7
         img

Out[92]:
```

```
In [93]: 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]]
```



Plant Disease Vegetables Last Checkpoint: Yesterday at 11:04 AM (autosaved)

```
In [130]: pwd

Out[130]: 'C:\\Users\\KAVINYUVA\\Downloads'

In [131]: # Image Augmentation

          from tensorflow.keras.preprocessing.image import ImageDataGenerator

In [132]: train_datagen = ImageDataGenerator(rescale = 1./255,shear_range = 0.2,zoom_range = 0.2,horizontal_flip = True)
          test_datagen = ImageDataGenerator(rescale = 1./255)

In [133]: pwd

Out[133]: 'C:\\Users\\KAVINYUVA\\Downloads'

In [134]: x_train = train_datagen.flow_from_directory(r'C:\Users\KAVINYUVA\Downloads\SmartInternzVideo\Veg-dataset\train_set',target_s
          x_test = test_datagen.flow_from_directory(r'C:\Users\KAVINYUVA\Downloads\SmartInternzVideo\Veg-dataset\train_set',target_si

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

In [135]: x_train.class_indices

Out[135]: {'Pepper,_bell___Bacterial_spot': 0,
           'Pepper,_bell___healthy': 1,
           'Potato___Early_blight': 2,
           'Potato___Late_blight': 3,
           'Potato___healthy': 4,
           'Tomato___Bacterial_spot': 5,
           'Tomato___Late_blight': 6,
           'Tomato___Leaf_Mold': 7,
           'Tomato___Septoria_leaf_spot': 8}

In [136]: x_test.class_indices

Out[136]: {'Pepper,_bell___Bacterial_spot': 0,
           'Pepper,_bell___healthy': 1,
           'Potato___Early_blight': 2,
           'Potato___Late_blight': 3,
           'Potato___healthy': 4,
           'Tomato___Bacterial_spot': 5,
           'Tomato___Late_blight': 6,
```

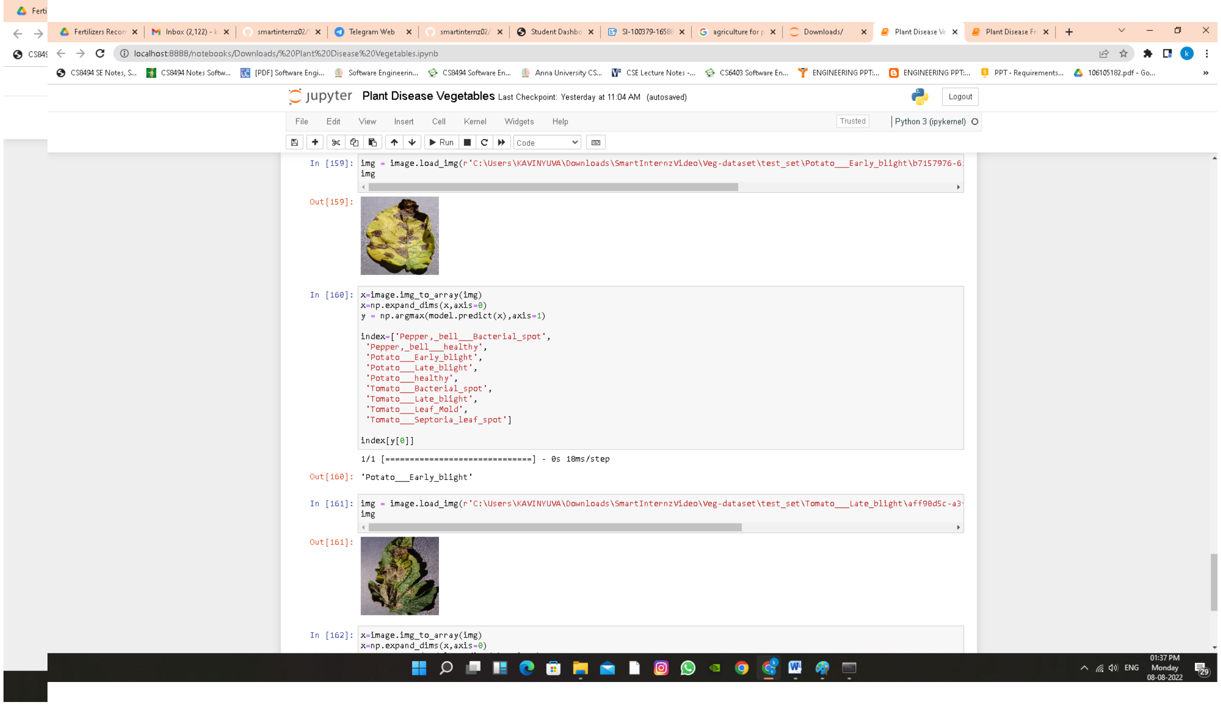

jupyter Plant Disease Vegetables Last Checkpoint: Yesterday at 11:04 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
localhost:8888/notebooks/Downloads/%20Plant%20Disease%20Vegetables.ipynb

```
 dense_34 (Dense)            (None, 9)                 279

=================================================================
Total params: 16,268,533
Trainable params: 16,268,533
Non-trainable params: 0
_________________________________________________________________
```

```
In [141]: import tensorflow as tf
          tf.__version__

Out[141]: '2.9.1'
```

```
In [142]: #model.fit_generator(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test),epochs=10)
          #model.fit(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test),epochs=10)
          model.fit(x_train,steps_per_epoch=11386//16,validation_data=x_test,validation_steps=3416//8,epochs=10)

Epoch 1/10
711/711 [==============================] - 75s 105ms/step - loss: 1.7240 - accuracy: 0.3738 - val_loss: 1.1137 - val_accur
acy: 0.6127
Epoch 2/10
711/711 [==============================] - 70s 98ms/step - loss: 1.0008 - accuracy: 0.6514 - val_loss: 0.8469 - val_accura
cy: 0.6701
Epoch 3/10
711/711 [==============================] - 72s 101ms/step - loss: 0.7712 - accuracy: 0.7269 - val_loss: 0.9147 - val_accur
acy: 0.6733
Epoch 4/10
711/711 [==============================] - 69s 97ms/step - loss: 0.6411 - accuracy: 0.7707 - val_loss: 0.4651 - val_accura
cy: 0.8311
Epoch 5/10
711/711 [==============================] - 81s 115ms/step - loss: 0.5853 - accuracy: 0.7927 - val_loss: 0.4442 - val_accur
acy: 0.8299
Epoch 6/10
711/711 [==============================] - 72s 101ms/step - loss: 0.5309 - accuracy: 0.8138 - val_loss: 0.4481 - val_accur
acy: 0.8235
Epoch 7/10
711/711 [==============================] - 71s 99ms/step - loss: 0.4904 - accuracy: 0.8263 - val_loss: 0.4466 - val_accura
cy: 0.8416
Epoch 8/10
711/711 [==============================] - 71s 100ms/step - loss: 0.4568 - accuracy: 0.8361 - val_loss: 0.4440 - val_accur
acy: 0.8413
Epoch 9/10
711/711 [==============================] - 71s 99ms/step - loss: 0.4354 - accuracy: 0.8486 - val_loss: 0.3507 - val_accura
cy: 0.8768
Epoch 10/10
711/711 [==============================] - 71s 100ms/step - loss: 0.4278 - accuracy: 0.8462 - val_loss: 0.3979 - val_accur
```

ENG
01:37 PM
Monday
08-08-2022

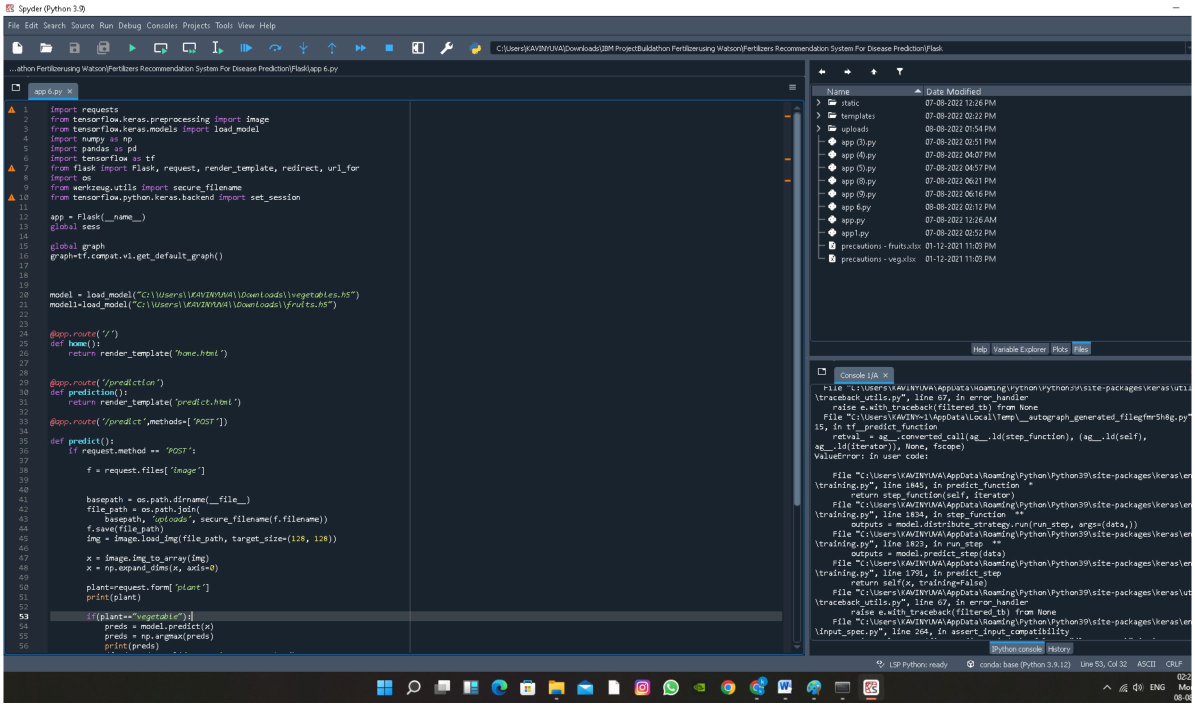

jupyter Plant Disease Vegetables Last Checkpoint: Yesterday at 11:04 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
localhost:8888/notebooks/Downloads/%20Plant%20Disease%20Vegetables.ipynb

```
In [159]: img = image.load_img(r'C:\Users\KAVINYUVA\Downloads\SmartInternzVideo\Veg-dataset\test_set\Potato___Early_blight\b7157976-6
          img

Out[159]:
```

```
In [160]: 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___Bacterial_spot',
           'Tomato___Late_blight',
           'Tomato___Leaf_Mold',
           'Tomato___Septoria_leaf_spot']

          index[y[0]]

          1/1 [==============================] - 0s 18ms/step

Out[160]: 'Potato___Early_blight'
```

```
In [161]: img = image.load_img(r'C:\Users\KAVINYUVA\Downloads\SmartInternzVideo\Veg-dataset\test_set\Tomato___Late_blight\aff90d5c-a3
          img

Out[161]:
```

```
In [162]: x=image.img_to_array(img)
          x=np.expand_dims(x,axis=0)
```



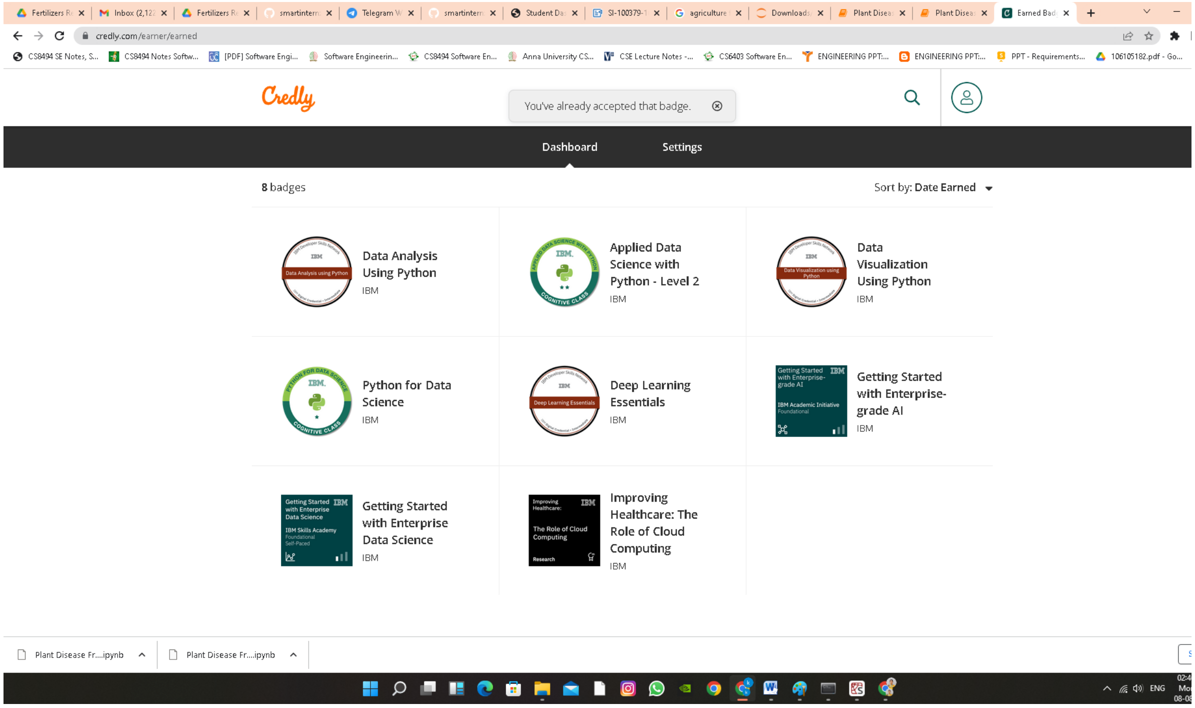
Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\KAVINYUVA\Downloads\IBM ProjectBuildathon Fertilizerusing Watson\Fertilizers Recommendation System For Disease Prediction\Flask
…athon Fertilizerusing Watson\Fertilizers Recommendation System For Disease Prediction\Flask\app 6.py
app 6.py

```python
import requests
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
import numpy as np
import pandas as pd
import tensorflow as tf
from flask import Flask, request, render_template, redirect, url_for
import os
from werkzeug.utils import secure_filename
from tensorflow.python.keras.backend import set_session

app = Flask(__name__)
global sess

global graph
graph=tf.compat.v1.get_default_graph()



model = load_model("C:\\Users\\KAVINYUVA\\Downloads\\vegetables.h5")
model1=load_model("C:\\Users\\KAVINYUVA\\Downloads\\fruits.h5")


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


@app.route('/prediction')
def prediction():
    return render_template('predict.html')

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

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

        f = request.files['image']


        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)
        img = image.load_img(file_path, target_size=(128, 128))

        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)

        plant=request.form['plant']
        print(plant)

        if(plant=="vegetable"):
            preds = model.predict(x)
            preds = np.argmax(preds)
            print(preds)
```

Name	Date Modified
static	07-08-2022 12:26 PM
templates	07-08-2022 02:22 PM
uploads	08-08-2022 01:54 PM
app (3).py	07-08-2022 02:51 PM
app (4).py	07-08-2022 04:07 PM
app (5).py	07-08-2022 04:57 PM
app (8).py	07-08-2022 06:21 PM
app (9).py	07-08-2022 06:16 PM
app 6.py	08-08-2022 02:12 PM
app.py	07-08-2022 12:26 AM
app1.py	07-08-2022 02:52 PM
precautions - fruits.xlsx	01-12-2021 11:03 PM
precautions - veg.xlsx	01-12-2021 11:03 PM

```
  File "C:\Users\KAVINYUVA\AppData\Roaming\Python\Python39\site-packages\keras\uti
\traceback_utils.py", line 67, in error_handler
    raise e.with_traceback(filtered_tb) from None
  File "C:\Users\KAVINY~1\AppData\Local\Temp\__autograph_generated_filegfmr5h8g.py"
15, in tf__predict_function
    retval_ = ag__.converted_call(ag__.ld(step_function), (ag__.ld(self),
ag__.ld(iterator)), None, fscope)
ValueError: in user code:

    File "C:\Users\KAVINYUVA\AppData\Roaming\Python\Python39\site-packages\keras\en
\training.py", line 1845, in predict_function  *
        return step_function(self, iterator)
    File "C:\Users\KAVINYUVA\AppData\Roaming\Python\Python39\site-packages\keras\en
\training.py", line 1834, in step_function  **
        outputs = model.distribute_strategy.run(run_step, args=(data,))
    File "C:\Users\KAVINYUVA\AppData\Roaming\Python\Python39\site-packages\keras\en
\training.py", line 1823, in run_step  **
        outputs = model.predict_step(data)
    File "C:\Users\KAVINYUVA\AppData\Roaming\Python\Python39\site-packages\keras\en
\training.py", line 1791, in predict_step
        return self(x, training=False)
    File "C:\Users\KAVINYUVA\AppData\Roaming\Python\Python39\site-packages\keras\ut
\traceback_utils.py", line 67, in error_handler
        raise e.with_traceback(filtered_tb) from None
    File "C:\Users\KAVINYUVA\AppData\Roaming\Python\Python39\site-packages\keras\en
\input_spec.py", line 264, in assert_input_compatibility
```

Certifications:



You've already accepted that badge.

Dashboard Settings

8 badges

Sort by: Date Earned

Data Analysis Using Python
IBM

Applied Data Science with Python - Level 2
IBM

Data Visualization Using Python
IBM

Python for Data Science
IBM

Deep Learning Essentials
IBM

Getting Started with Enterprise-grade AI
IBM

Getting Started with Enterprise Data Science
IBM

Improving Healthcare: The Role of Cloud Computing
IBM