EEE-2004-Measurement and Instrumentation
Project Review-1

Early Identification of Disease in Tomato Leaf using Machine Learning(Deep Learning)

To early Identify the type of disease in tomato plant leaf using Deep Learning

TEAM MEMBERS:-

Bishal Saha(20BEE0298)

Lakshit Ahuja(20BEI0018)

Deepanshu Sharma(20BEE0300)

FACULTY GUIDE:-

Dr P.Arulmozhivarman,Professor ,SELECT

CURRENT STATUS AND MOTIVATION

Plant disease: a threat to global food security

India is an agrarian country, and a major part of its economy depends on the agricultural sector. The share of agriculture in the Indian Gross Domestic Product (GDP) and total exports are 16% and 10%, respectively (Himani, 2014). About 75% population of India depends on the agricultural sector either directly or indirectly (Himani, 2014). Therefore, disease-free good quality crop production is essential for the growth of the country's economy.

But It is also a hard reality that because of poor level of early identification of type of diseases in plants a high percentage of food plant got wasted which directly or indirectly affect our Food System.

Motivation

Plants are susceptive to various diseases in their growing phases. Early detection of diseases in plants is one of the most challenging problems in agriculture. If the diseases are not identified in the early stages, then they may adversely affect the total yield, resulting in a decrease in the farmers' profits. To overcome this problem, we thought we can apply modern intelligence like Machine Learning,Deep Learning,AI in order to early identify the diseases.

As if we can identify the diseases early we can also use proper pesticides,etc which helps to protect the plant,crops etc

Problem Statement

Early Identification of Disease in Tomato Leaf using Machine Learning(Deep Learning)

Objective

- To train Deep Learning Algorithm in order to early identify the type of disease in Tomato leaf.
- To help in agriculture field so that the type of diseases in plant leaf could be identified and proper treatment could be taken early.
- To increase the profit of farmers and hence reduce the shortage of tomato crisis.
- To make a website where user can easily test the condition of Tomato plant by draging drop of leaf.
- To built a Android App so that our farmers can install it and manully check the condition of leaf early with high accuracy.

Methodology

1. Data Collection

We will use Kaggle website for collection of dataset.This dataset contain 7 Type of diseases in tomato leaf including 1 healthy leaf.All are images of size (256,256) RGB (colourful) format.This dataset is also verified by ATLIQ Agriculure**.**

In order to test our app in real time will be collecting tomato leaf from **vit Agri Farm(healthy and Early Blight)**and test in 2nd Android App.Conditions of leafs is guided by a senior in Agriculture Branch of VIT.

Types of Diseases in Dataset:-

1.Early Blight,1000

2.Late Blight,1909

3.Mosaic Virus,373

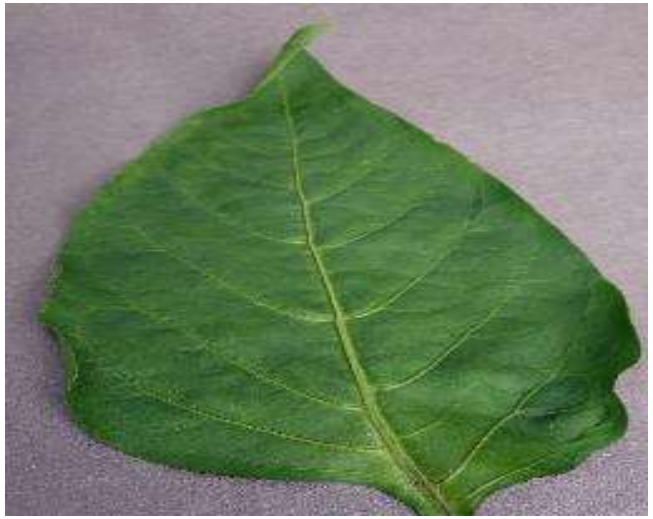
4.Yellow leaf-curl virus 3209

5.Leaf Mold 952

6. Two spotted spider mite 1676

7. Bacteria spot 2127

8.Healthy,1591



2.Data cleaning,Preprocessing,analyzing,Data augmentation(zooming+Rotation+increasing data)

3.Resizing the images,Converting (256*256*3) between (0-1) as a numpy array for processing the images.

4.Spliting dataset into Training,Test and validation categories,(80+10+10).

5.Using CNN for model building as CNN is good for image classification and identification.

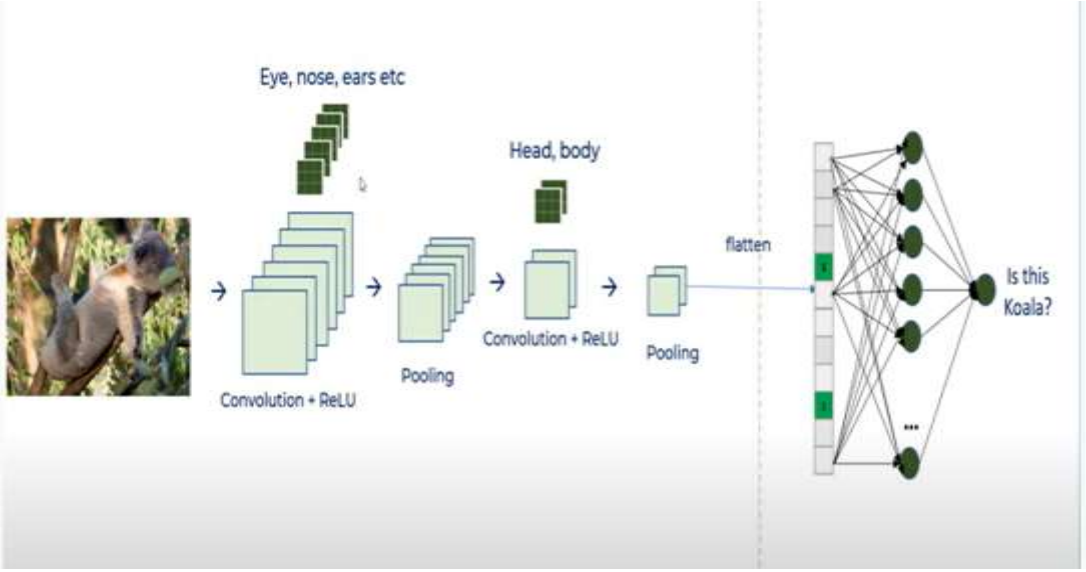
6.Testing the test set and saving the model in .tflite and .h5 format.

- After Building the model we will save it as..tflite format.Then we will write backend server in Flask ,use the trained model,and then integrate with Frontend(using HTML CSS) .We will make make a drag drop type website where users can quickly get the contidition of leaf.

- We will also use Android studio as a app making Software for making our mobile app.we may use Google cloud Platform for making our app or we can use fast api server.

ALGORITHM

- **Convolution Neural Network Algorithm**



1.CNN is a advance ANN,Feature of CNN is that it can detect pattern because it has hidden layers called convolutional layer.

2.Each convolution layer is paired with filter matrix(which are only responsible for detectiojn features).These filter matrix convolve throughout the image pixels and store the dot product in convolution layer

3.To bring non linearity in our layer we use relu,which makes negative number to 0.

4.In order to reduce the convolve layer size we use Max pooling which reduces the convolve layer image feature size

Shifted 9 at different position

1	1	1	-1	-1
1	-1	1	-1	-1
1	1	1	-1	-1
-1	-1	1	-1	-1
-1	-1	1	-1	-1
-1	1	-1	-1	-1
1	-1	-1	-1	-1

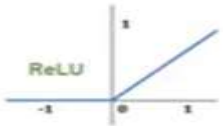
*

Loopy pattern filter

1	1	1
1	-1	1
1	1	1

→

1	-0.11	-0.11
0.11	-0.33	0.33
0.33	-0.33	-0.33
-0.11	-0.55	-0.33
-0.55	-0.33	-0.55

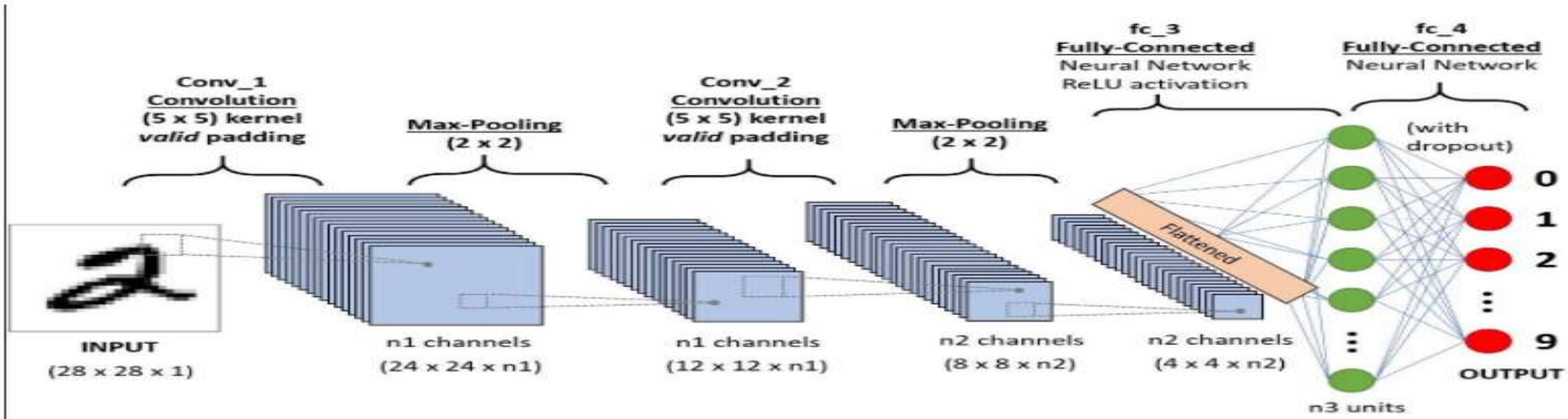


→

1	0	0
0.11	0	0.33
0.33	0	0
0	0	0
0	0	0

Max pooling →

1	0.33
0.33	0.33
0.33	0
0	0



Design Implementation-Hardware and Tools Details

SOFTWARES

1.Jupyter Notebook

2.Tensorflow

3.Pycharm IDE

4.Python Compiler

5.Android Studio

6.Flask

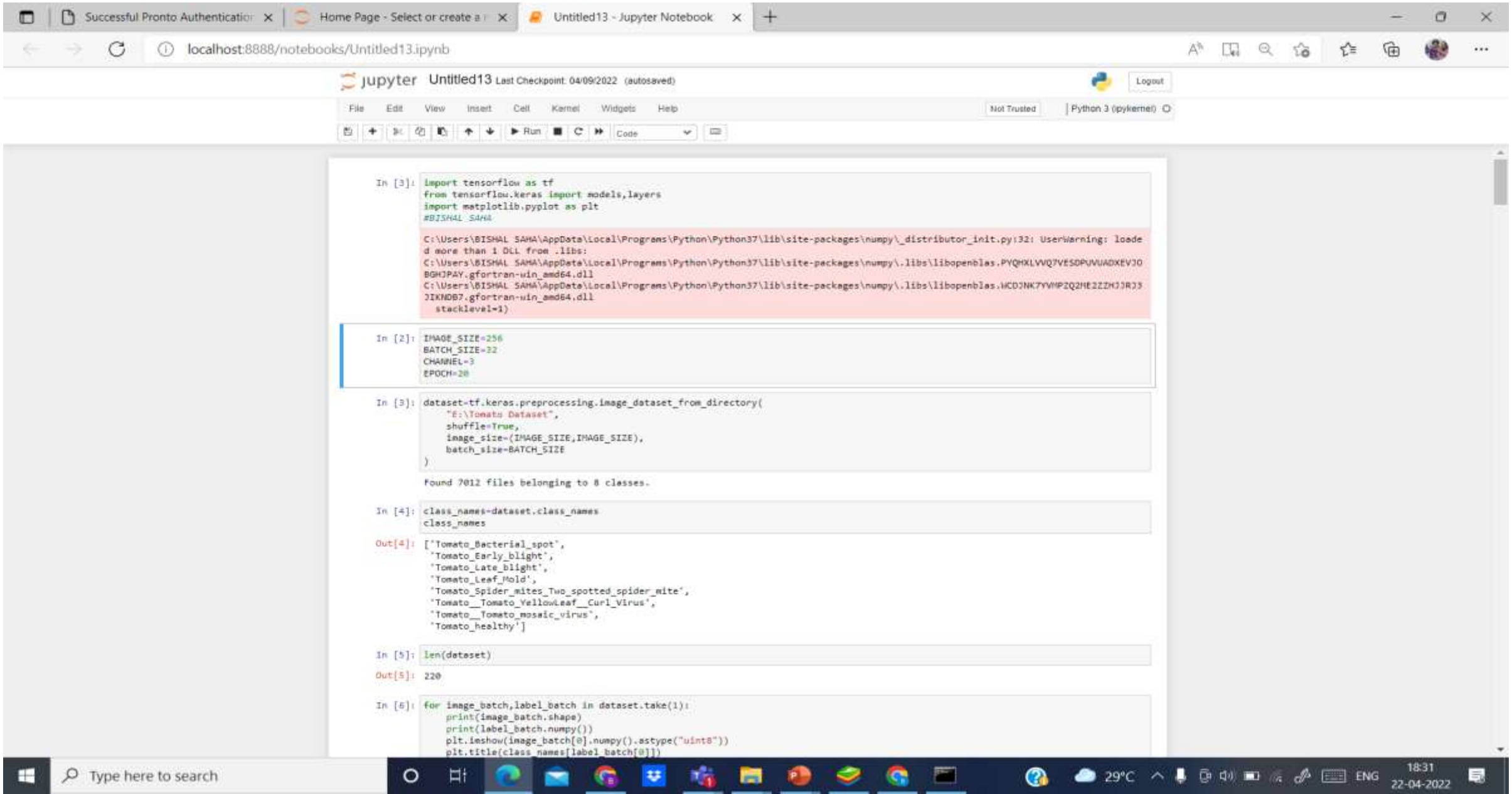
HARDWARES

1.Android Mobile Phone

RESULTS ACHIEVED(till date)

- We successfuly code the deep learning model in python using Tensorflow and Keras to detect tomato leaf disease.
- We deployed our deep learning model in djnago website using flask as backened and (CSS+HTML) as frontend
- We also built 2 Android App (one for test and gallery images and other for real time images) in Android Studio using Java programming Language for the same.

- [Untitled13 - Jupyter Notebook]



Successful Pronto Authentication | Home Page - Select or create a r | Untitled13 - Jupyter Notebook

localhost:8888/notebooks/Untitled13.ipynb

jupyter Untitled13 Last Checkpoint: 04/09/2022 (autosaved) Logout

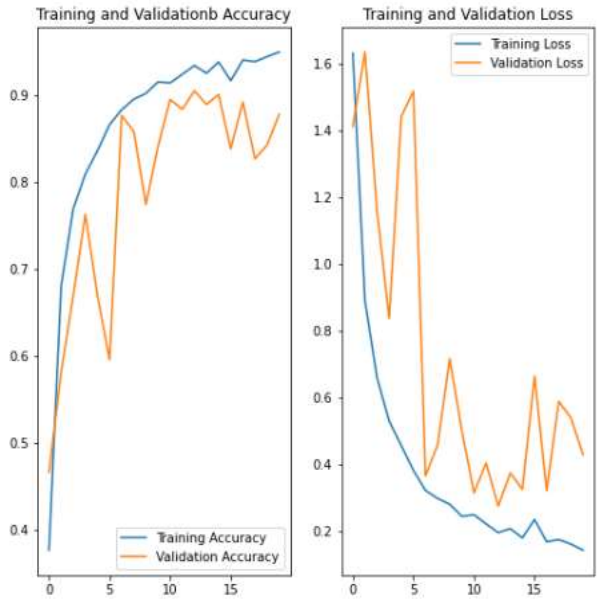
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Code

In [27]:
```
plt.figure(figsize=(8,8))
plt.subplot(1,2,1)
plt.plot(range(EPOCH),acc,label='Training Accuracy')
plt.plot(range(EPOCH),val_acc,label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validationb Accuracy')

plt.subplot(1,2,2)
plt.plot(range(EPOCH),loss,label='Training Loss')
plt.plot(range(EPOCH),val_loss,label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
```

Out[27]: Text(0.5, 1.0, 'Training and Validation Loss')



In [28]:
```
import numpy as np
for images_batch,labels_batch in test_ds.take(1):
    first_image=images_batch[0].numpy().astype('uint8')
    first_label=labels_batch[0].numpy()

    print("First image to Predict")
    plt.imshow(first_image)
```

Type here to search 29°C ENG 18:32 22-04-2022

Successful Pronto Authentication | Home Page - Select or create a r | Untitled13 - Jupyter Notebook

localhost:8888/notebooks/Untitled13.ipynb

jupyter Untitled13 Last Checkpoint: 04/09/2022 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Code

```
Epoch 17/20
176/176 [==============================] - 618s 4s/step - loss: 0.1680 - accuracy: 0.9402 - val_loss: 0.3213 - val_accuracy: 0.
8920
Epoch 18/20
176/176 [==============================] - 1762s 10s/step - loss: 0.1743 - accuracy: 0.9384 - val_loss: 0.5886 - val_accuracy:
0.8267
Epoch 19/20
176/176 [==============================] - 1784s 10s/step - loss: 0.1611 - accuracy: 0.9441 - val_loss: 0.5394 - val_accuracy:
0.8423
Epoch 20/20
176/176 [==============================] - 1705s 10s/step - loss: 0.1429 - accuracy: 0.9495 - val_loss: 0.4299 - val_accuracy:
0.8778
```

In [21]:
```
scores=model.evaluate(test_ds)
```

```
22/22 [==============================] - 44s 1s/step - loss: 0.4085 - accuracy: 0.8864
```

In [22]:
```
scores=model.evaluate(test_ds)
```

```
22/22 [==============================] - 44s 1s/step - loss: 0.4099 - accuracy: 0.8821
```

In [23]:
```
scores=model.evaluate(test_ds)
```

```
22/22 [==============================] - 37s 1s/step - loss: 0.3902 - accuracy: 0.8906
```

In [24]:
```
history.params
```

Out[24]:
```
{'verbose': 1, 'epochs': 20, 'steps': 176}
```

In [25]:
```
history.history.keys()
```

Out[25]:
```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

In [26]:
```
acc=history.history['accuracy']
val_acc=history.history['val_accuracy']
loss=history.history['loss']
val_loss=history.history['val_loss']
```

In [27]:
```
plt.figure(figsize=(8,8))
plt.subplot(1,2,1)
plt.plot(range(EPOCH),acc,label='Training Accuracy')
plt.plot(range(EPOCH),val_acc,label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validationb Accuracy')

plt.subplot(1,2,2)
plt.plot(range(EPOCH),loss,label='Training Loss')
plt.plot(range(EPOCH),val_loss,label='Validation Loss')
plt.legend(loc='upper right')
```

Type here to search 29°C ENG 18:33 22-04-2022

Successful Pronto Authentication | Home Page - Select or create a n | Untitled13 - Jupyter Notebook

localhost:8888/notebooks/Untitled13.ipynb

jupyter Untitled13 Last Checkpoint: 04/09/2022 (autosaved)

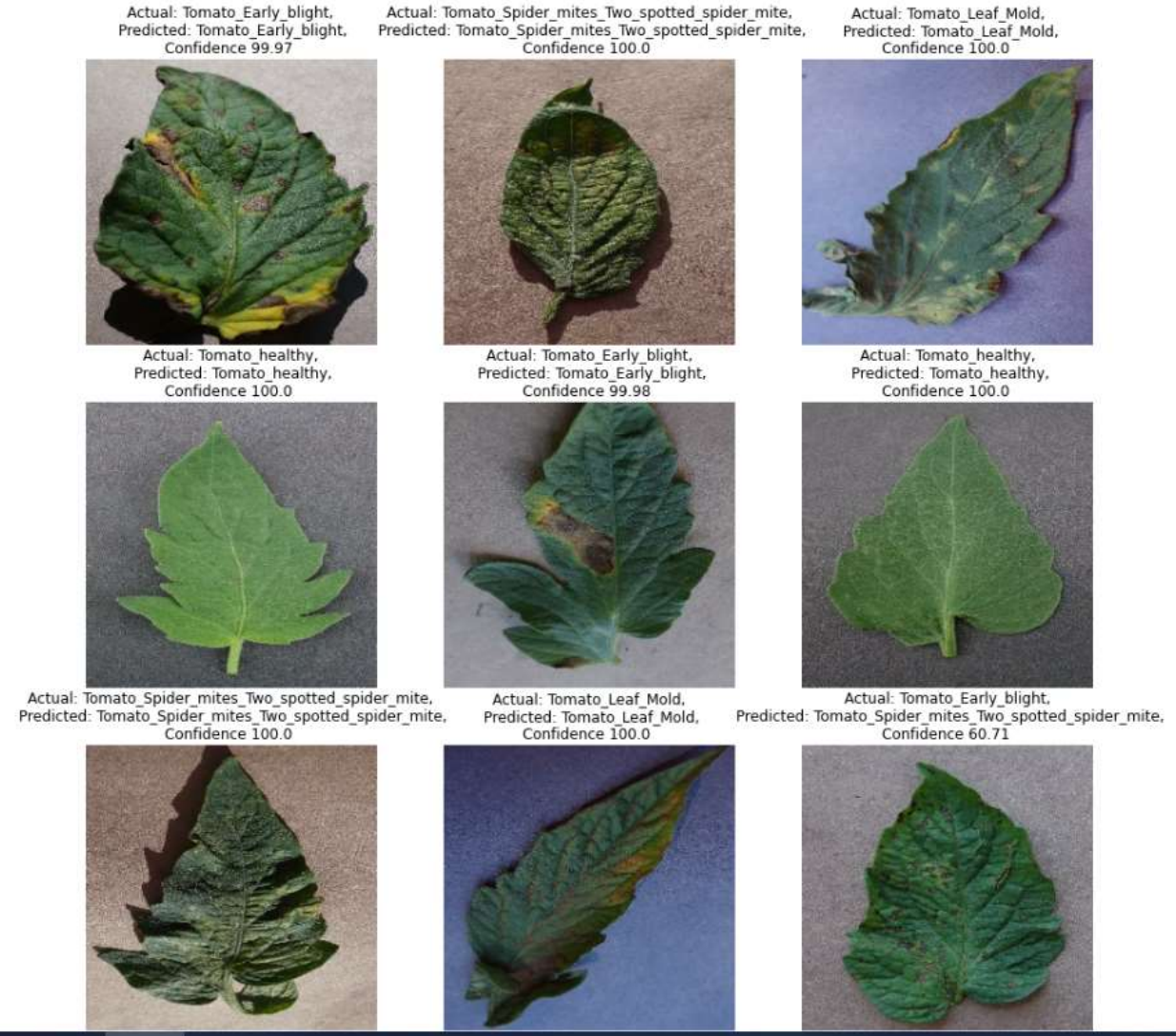
Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Code

```
plt.axis("off")
```



Type here to search

29°C ENG 18:32 22-04-2022

File Edit View Navigate Code Refactor Run Tools VCS Window Help Tomato Leaf Diseases - main.py

Tomato Leaf Diseases > main.py

main

Project

Tomato Leaf Diseases D:\pythonProject\Tomato Leaf Diseas
- static
- templates
  - about.html
  - bootstrap.html
  - index.html
  - upload.html
- main.py
- model2Tomato

External Libraries

Scratches and Consoles

main.py

```
from flask import Flask, render_template
import os
import numpy as np
from io import BytesIO
from PIL import Image
import tensorflow as tf
from flask import Flask, flash, request, redirect, url_for
from werkzeug.utils import secure_filename
UPLOAD_FOLDER = 'E:\Potato Extras\Flask'
ALLOWED_EXTENSIONS = {'txt', 'pdf', 'png', 'jpg', 'jpeg', 'gif'}
#Bishal Saha
app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER



@app.route("/upload")
def hellos():

    return render_template('upload.html')

def allowed_file(filename):
    return '.' in filename and \
           filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS



def read_file_as_image(data) -> np.ndarray:
    image = np.array(Image.open(BytesIO(data)))
    return image
```

22 1

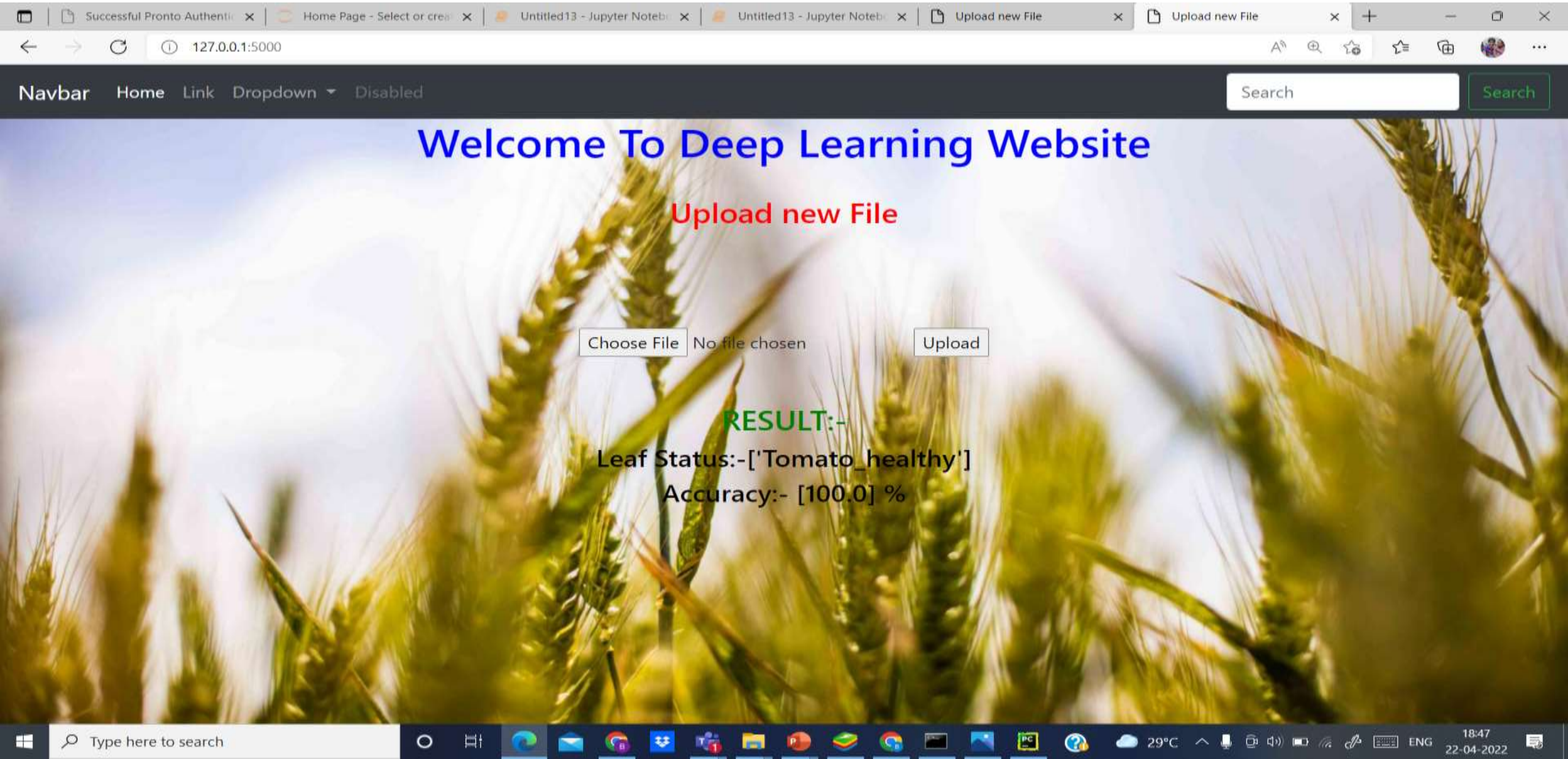
allowed_file()

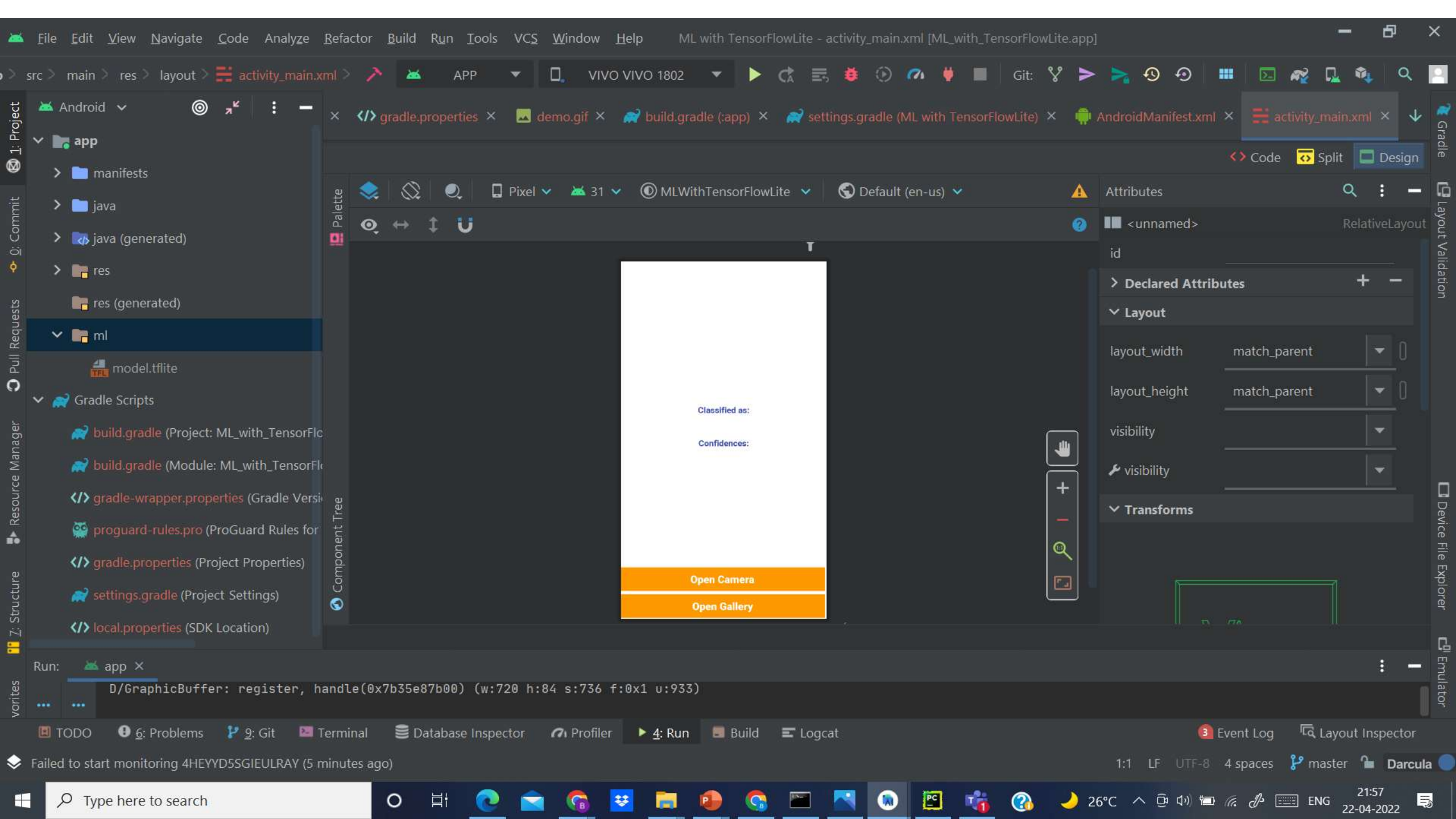
TODO Problems Terminal Python Packages Python Console Event Log

Cleaning up skeletons... 24:68 CRLF UTF-8 4 spaces Python 3.7

Type here to search 29°C ENG 18:38 22-04-2022

Website :-http://127.0.0.1:5000/




File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ML with TensorFlowLite - activity_main.xml [ML_with_TensorFlowLite.app]
src > main > res > layout > activity_main.xml
APP
VIVO VIVO 1802
Git:
Android
app
manifests
java
java (generated)
res
res (generated)
ml
model.tflite
Gradle Scripts
build.gradle (Project: ML_with_TensorFlc
build.gradle (Module: ML_with_TensorFl
gradle-wrapper.properties (Gradle Versi
proguard-rules.pro (ProGuard Rules for
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
gradle.properties
demo.gif
build.gradle (:app)
settings.gradle (ML with TensorFlowLite)
AndroidManifest.xml
activity_main.xml
Code
Split
Design
Palette
Component Tree
Pixel
31
MLWithTensorFlowLite
Default (en-us)
Classified as:
Confidences:
Open Camera
Open Gallery
Attributes
<unnamed>
RelativeLayout
id
Declared Attributes
Layout
layout_width
match_parent
layout_height
match_parent
visibility
visibility
Transforms
Run:
app
D/GraphicBuffer: register, handle(0x7b35e87b00) (w:720 h:84 s:736 f:0x1 u:933)
TODO
6: Problems
9: Git
Terminal
Database Inspector
Profiler
4: Run
Build
Logcat
Event Log
Layout Inspector
Failed to start monitoring 4HEYYD5SGIEULRAY (5 minutes ago)
1:1
LF
UTF-8
4 spaces
master
Darcula
Type here to search
26°C
ENG
21:57
22-04-2022

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help ML with TensorFlowLite - MainActivity.java [ML_with_TensorFlowLite.app]

onCreate > anonymous OnClickListener > onClick > APP VIVO VIVO 1802 Git:

Android

app
- manifests
- java
- java (generated)
- res
- res (generated)
- ml
  - model.tflite

Gradle Scripts
- build.gradle (Project: ML_with_TensorFlc
- build.gradle (Module: ML_with_TensorFl
- gradle-wrapper.properties (Gradle Versi
- proguard-rules.pro (ProGuard Rules for
- gradle.properties (Project Properties)
- settings.gradle (Project Settings)
- local.properties (SDK Location)

MainActivity.java | build.gradle (ML with TensorFlowLite) | gradlew | gradle.properties | demo.gif | build.gradle (:app) | settings.gradle

```
1   /.../
5
6   package app.ij.mlwithtensorflowlite;
7
8   import ...
32  //Bishal
33  public class MainActivity extends AppCompatActivity {
34
35      TextView result, confidence;
36      ImageView imageView;
37      Button camera,gallery;
38      int imageSize = 256;
39
40      @Override
41      protected void onCreate(Bundle savedInstanceState) {
42          super.onCreate(savedInstanceState);
43          setContentView(R.layout.activity_main);
44
45          result = findViewById(R.id.result);
46          confidence = findViewById(R.id.confidence);
47          imageView = findViewById(R.id.imageView);
48          camera = findViewById(R.id.button1);
49          gallery = findViewById(R.id.button2);
50
51          camera.setOnClickListener(new View.OnClickListener() {
52              @Override
```

Run: app

```
D/GraphicBuffer: register, handle(0x7b35e87b00) (w:720 h:84 s:736 f:0x1 u:933)
```

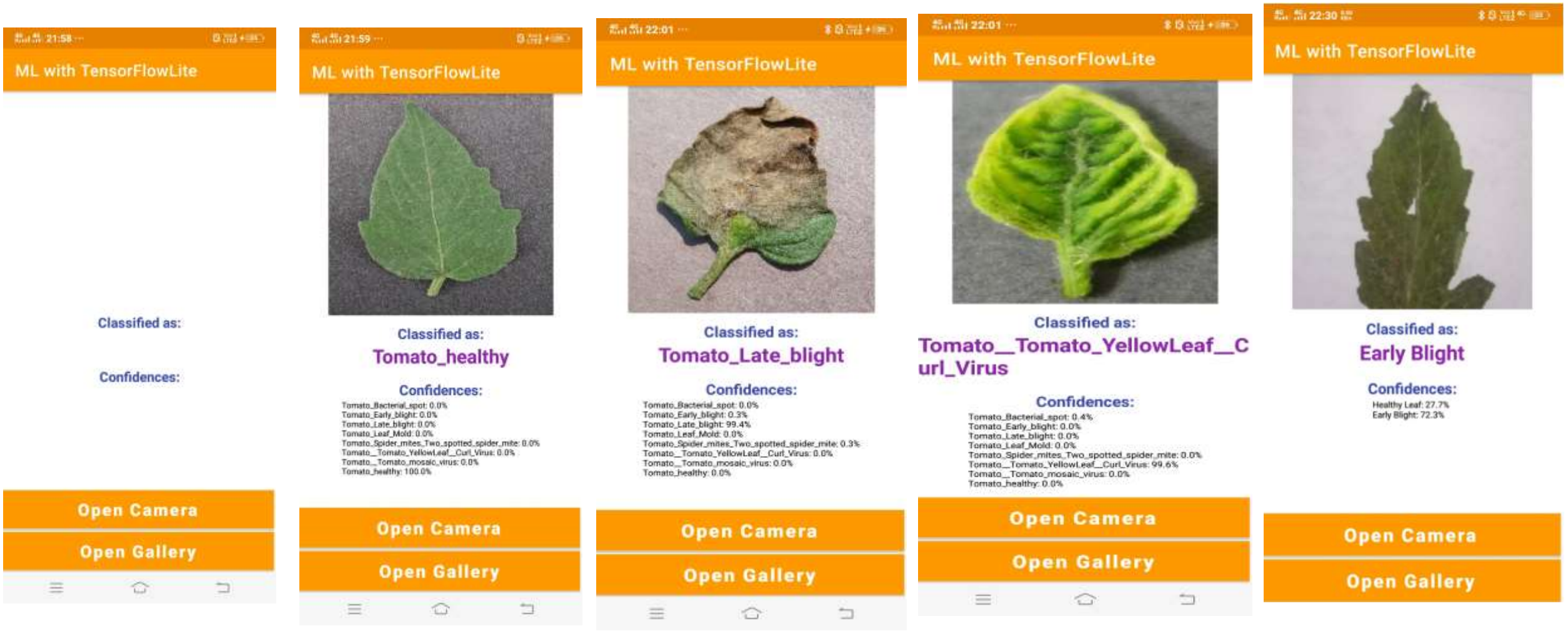
TODO | 6: Problems | 9: Git | Terminal | Database Inspector | Profiler | 4: Run | Build | Logcat | Event Log | Layout Inspector

Failed to start monitoring 4HEYYD5SGIEULRAY (5 minutes ago)

52:22 LF UTF-8 4 spaces master Darcula

Type here to search 26°C ENG 21:57 22-04-2022

Android App


ML with TensorFlowLite
Classified as:
Confidences:
Open Camera
Open Gallery
ML with TensorFlowLite
Classified as:
Tomato_healthy
Confidences:
Tomato_Bacterial_spot: 0.0%
Tomato_Early_blight: 0.0%
Tomato_Late_blight: 0.0%
Tomato_Leaf_Mold: 0.0%
Tomato_Spider_mites_Two_spotted_spider_mite: 0.0%
Tomato__Tomato_YellowLeaf__Curl_Virus: 0.0%
Tomato__Tomato_mosaic_virus: 0.0%
Tomato_healthy: 100.0%
Open Camera
Open Gallery
ML with TensorFlowLite
Classified as:
Tomato_Late_blight
Confidences:
Tomato_Bacterial_spot: 0.0%
Tomato_Early_blight: 0.3%
Tomato_Late_blight: 99.4%
Tomato_Leaf_Mold: 0.0%
Tomato_Spider_mites_Two_spotted_spider_mite: 0.3%
Tomato__Tomato_YellowLeaf__Curl_Virus: 0.0%
Tomato__Tomato_mosaic_virus: 0.0%
Tomato_healthy: 0.0%
Open Camera
Open Gallery
ML with TensorFlowLite
Classified as:
Tomato__Tomato_YellowLeaf__Curl_Virus
Confidences:
Tomato_Bacterial_spot: 0.4%
Tomato_Early_blight: 0.0%
Tomato_Late_blight: 0.0%
Tomato_Leaf_Mold: 0.0%
Tomato_Spider_mites_Two_spotted_spider_mite: 0.0%
Tomato__Tomato_YellowLeaf__Curl_Virus: 99.6%
Tomato__Tomato_mosaic_virus: 0.0%
Tomato_healthy: 0.0%
Open Camera
Open Gallery
ML with TensorFlowLite
Classified as:
Early Blight
Confidences:
Healthy Leaf: 27.7%
Early Blight: 72.3%
Open Camera
Open Gallery

Lessons Learned- SCOPE FOR IMPROVEMENT

- We can make it more accurate by training the real time totamo leaf from Agriculture Field
- We can use more advanced Deep learning Algorithm like RNN etc to overcome the isuues of brightness,background and distance of images.
- Using GPU and data augmentation we may increase the accuracy rate by increasing the number of Epochs.

Standard Adopted-IEEE Standard

IEEE P2841™ – Framework and Process for Deep Learning Evaluation

This document defines best practices for developing and implementing deep learning algorithms and defines a framework and criteria for evaluating algorithm reliability and quality of the resulting software systems.

Time Line(Jan to May 2022)

Jan 2022 - Topic Selection

Feb 2022- Dataset collection,Deep Learning Coding using CNN in Jupyter Notebook

March 2022- Making Website(backend+Frontend)

April 2022- Android app using android studio,Deploying in Google cloud Platform

May 2022- Wrapping up ,Testing,and optimizing our project

Roles and Responsiblity

Team Members	Contribution
Bishal 20BEE0298	Dataset collection,Deep Learning Coding,Data analysis Preprocessing,Website(Backend+HTML),Android(Backend using JAVA).
Lakshit 20BEI0018	Research, Model Testing, Android(1st App Design),CSS,Bootstrap,Final Report
Deepanshu 20BEE0300	Deep Learning Coding,Writing part,website(bootstrap),Data collection,Android(2nd App Design)

Reference

- Ping kuang, Wei-na cao and Qiao wu, “Preview on Structures and Algorithms of Deep Learning”, 11th International Computer Conference on Wavelet Actiev Media Technology and Information Processing (ICCWAMTIP), IEEE, 2014.

- Zhengwei Huang, Min Dong, qirong Mao and Yongzhao Zhan, “Speech Recognition using CNN”, IEEE/ACM Transactions on Audio, Speech and Language Processing, pp. 1533-1545, Volume 22, Issue 10, 2014, http://dx.doi.org/10.1145/2647868.2654984.

- Christian szegedy, Wei liu, Yangqing Jia et al., “Going Deeper with Convolutions”,Conference on Computer Vision and Pattern Recognition (CPVR) , IEEE explorer, Boston, MA, USA, 2015.

- Mahmoud M. Abu Ghosh ; Ashraf Y. Maghari, “A Comparative Study on Handwriting Digit Recognition Using Neural Networks”, IEEE, 2017.

- Yehya Abouelnaga , Ola S. Ali , Hager Rady , Mohamed Moustafa, “ CIFAR-10: KNN-based ensemble of classifiers”, IEEE, March 2017.