

Project Development Phase Delivery of Sprint 3

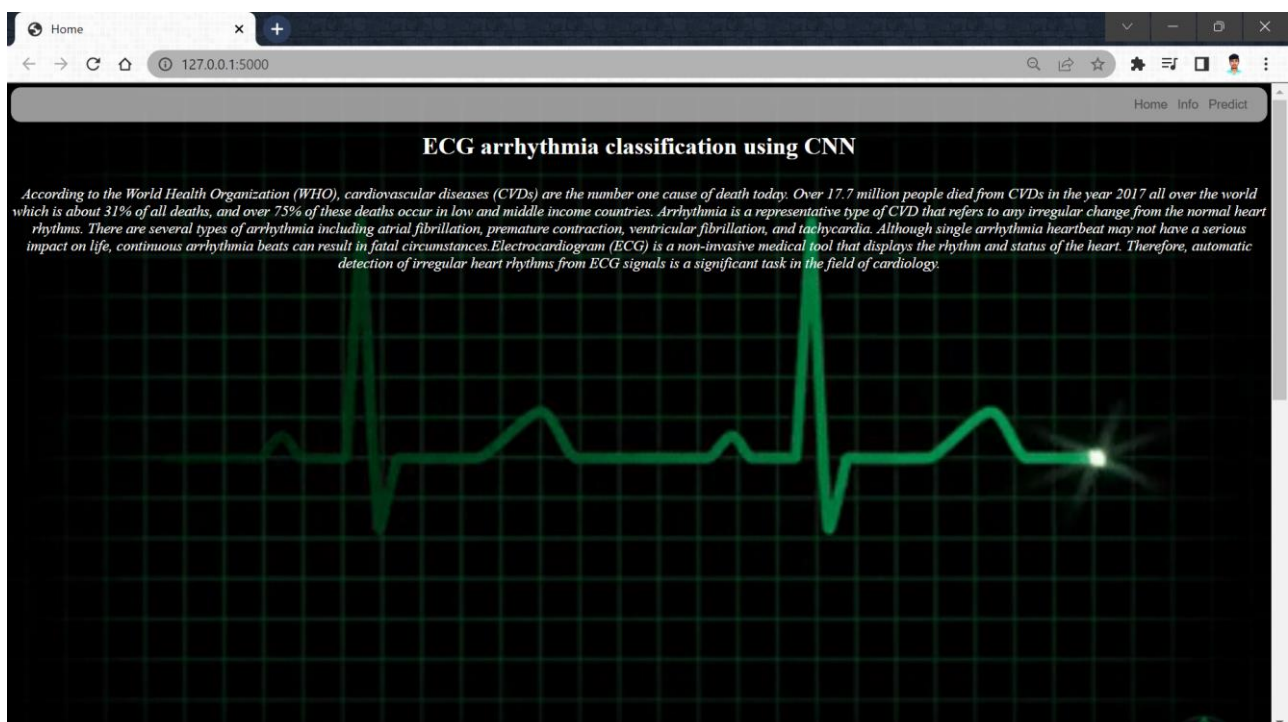
Date	November 11
Team ID	PNT2022TMID10109
Project Name	Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation

Local Deployment:

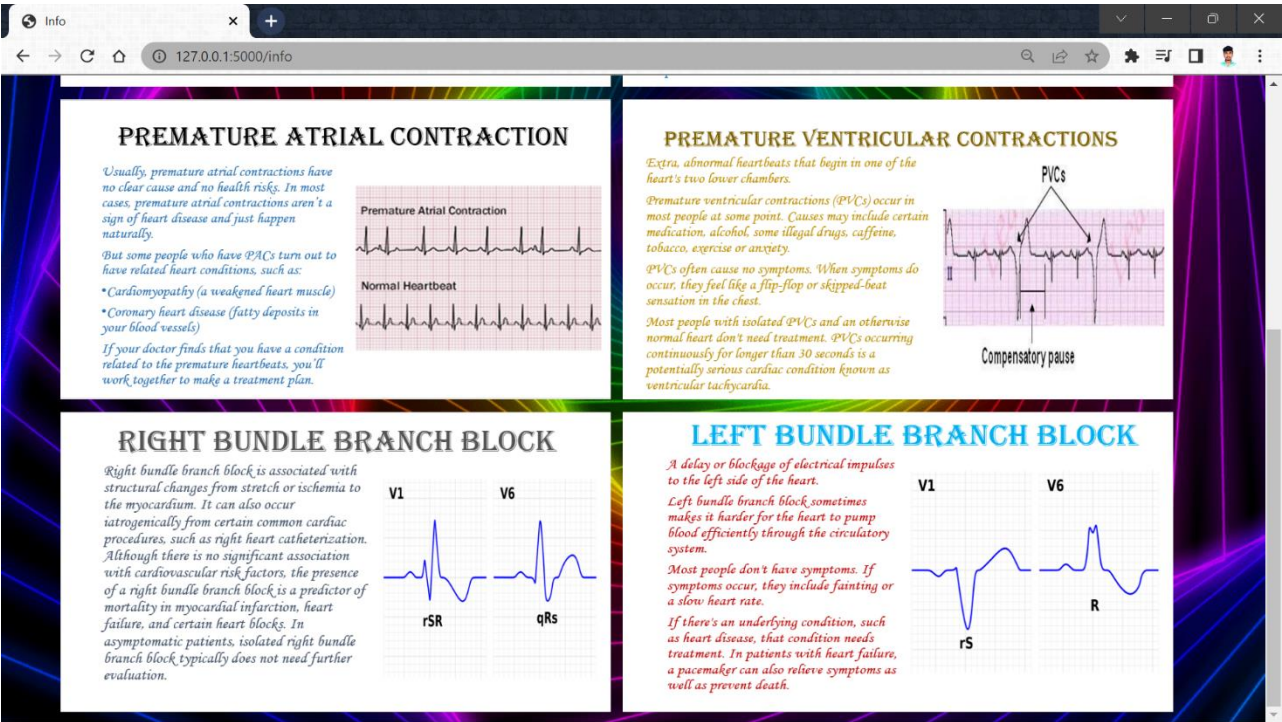
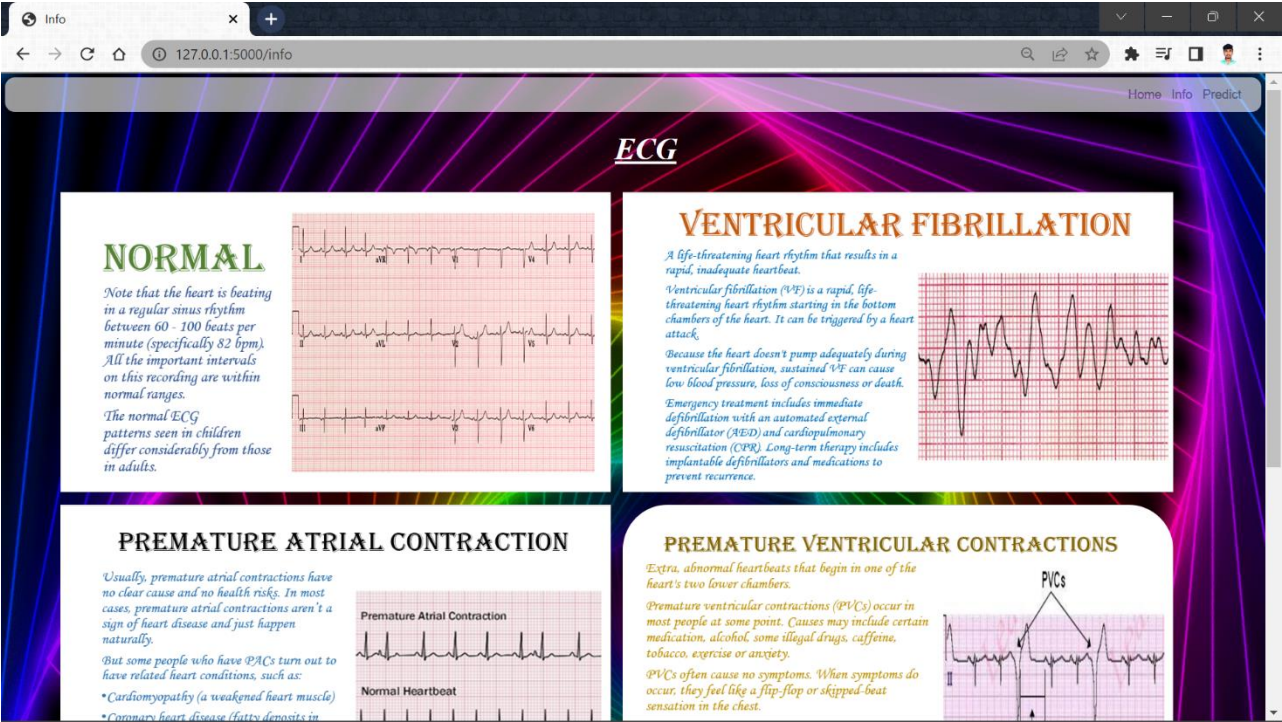
```
C:\Windows\System32\cmd.exe - python app.py
Microsoft Windows [Version 10.0.22000.1908]
(c) Microsoft Corporation. All rights reserved.

D:\Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation>python app.py
2022-11-09 18:53:43.075379: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2022-11-09 18:53:43.075930: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
check
D:\Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation>model\CAUDL.h5
2022-11-09 18:55:26.667248: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dlerror: nvcuda.dll not found
2022-11-09 18:55:26.680214: W tensorflow/stream_executor/cuda/cuda_driver.cc:763] failed call to cuInit: UNKNOWN ERROR (303)
2022-11-09 18:55:26.704461: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: DESKTOP-4VCEC33
2022-11-09 18:55:26.704904: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: DESKTOP-4VCEC33
2022-11-09 18:55:26.778397: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

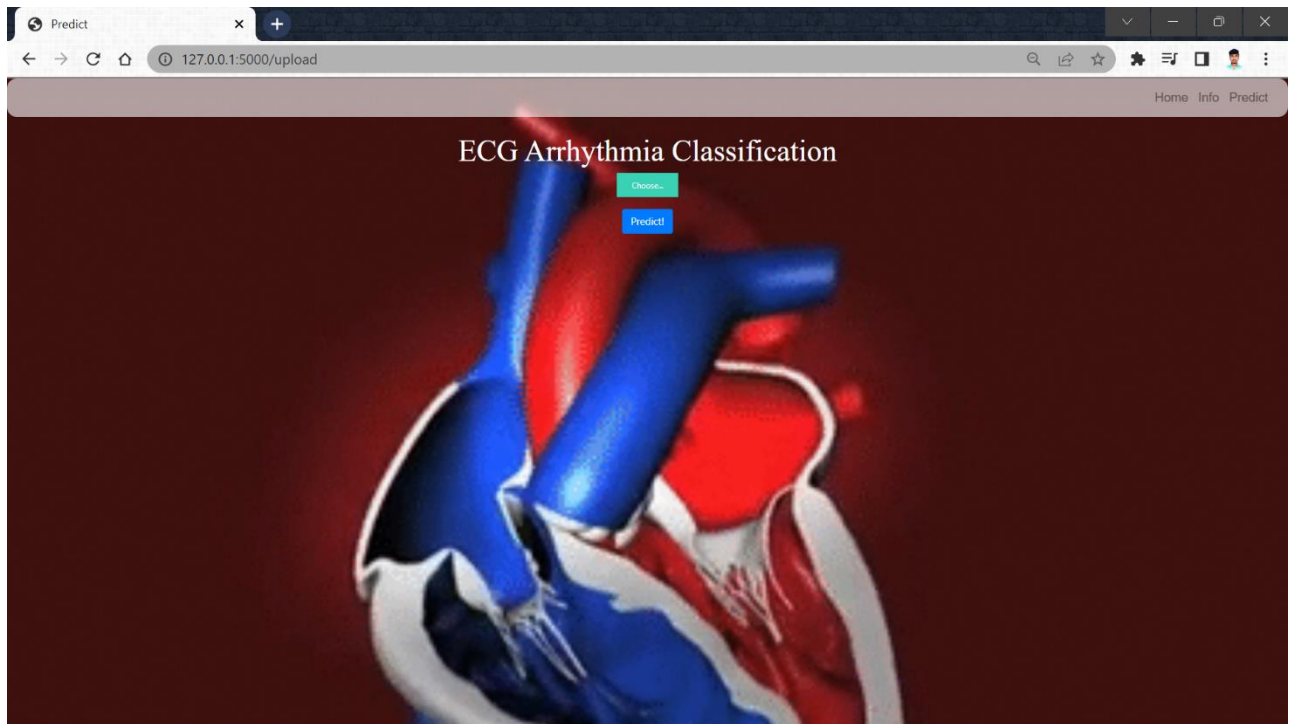
(Running App ! – Flask)



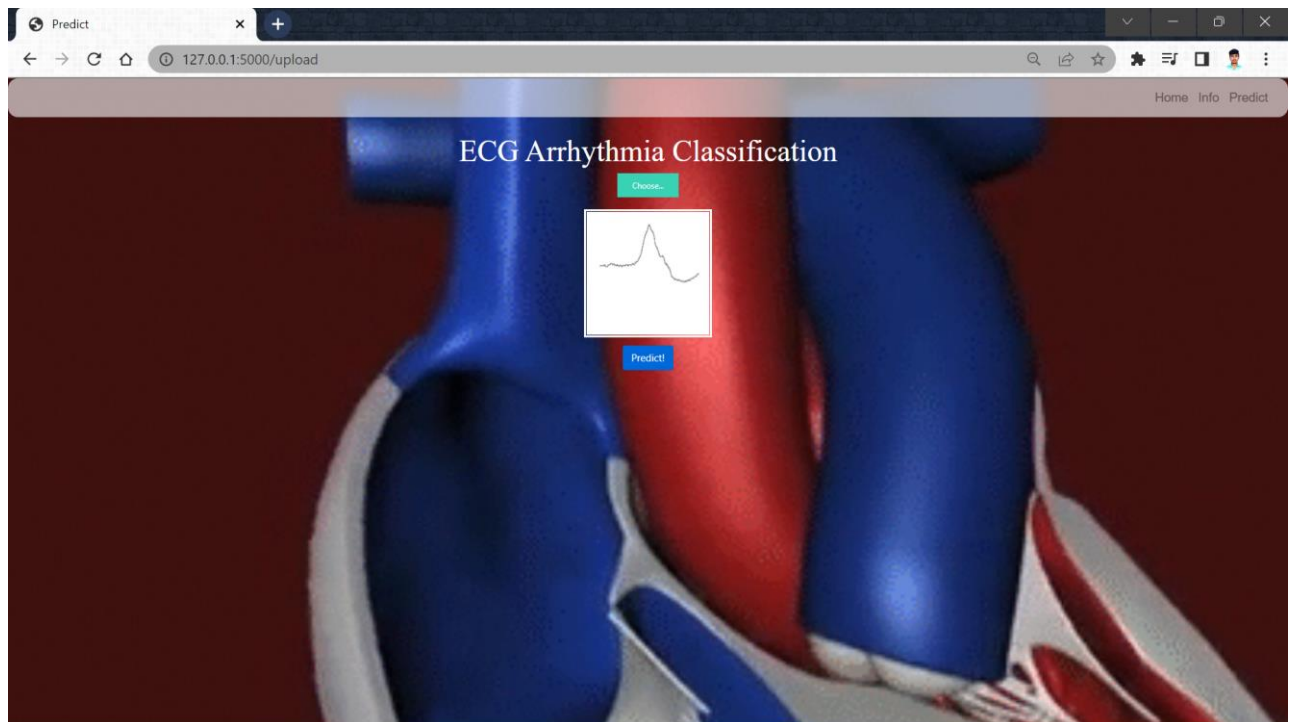
(UI Application Opens
Successfully..)



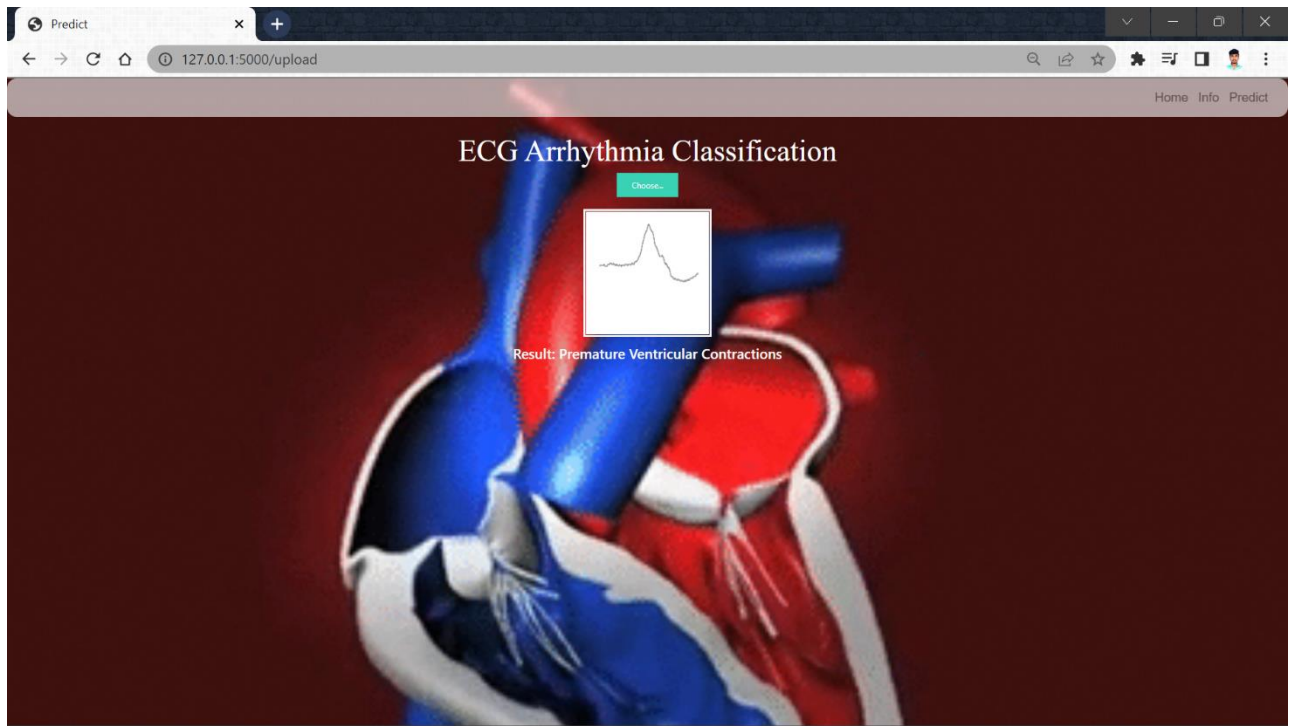
Information about the type of arrhythmia



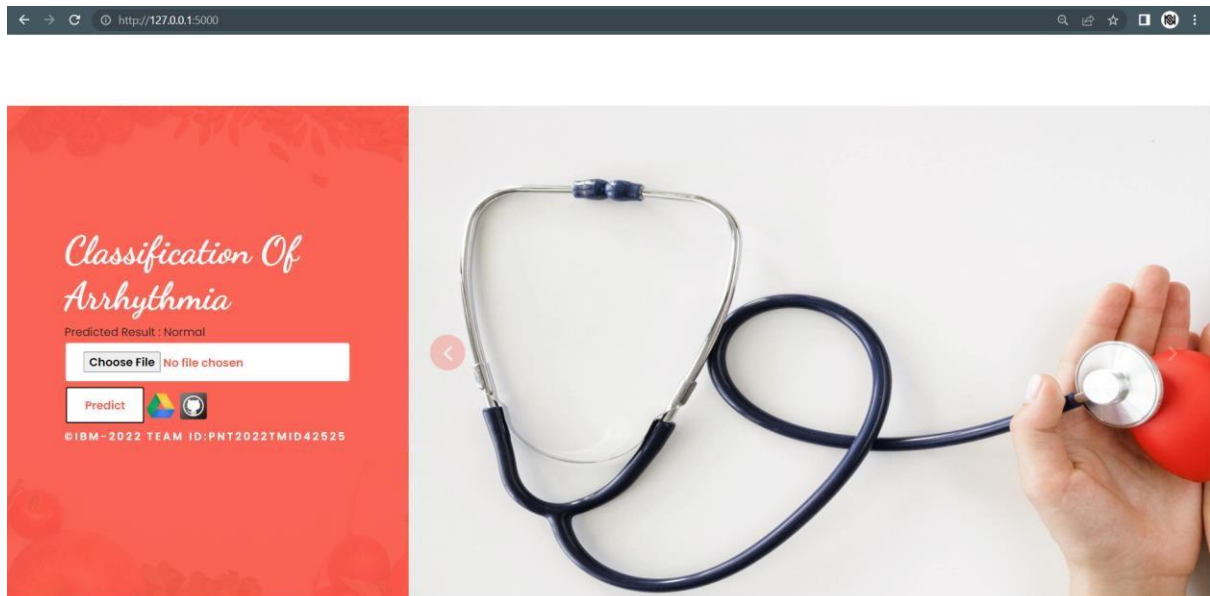
(The Image Is Uploaded – Click Predict Button To Predict)



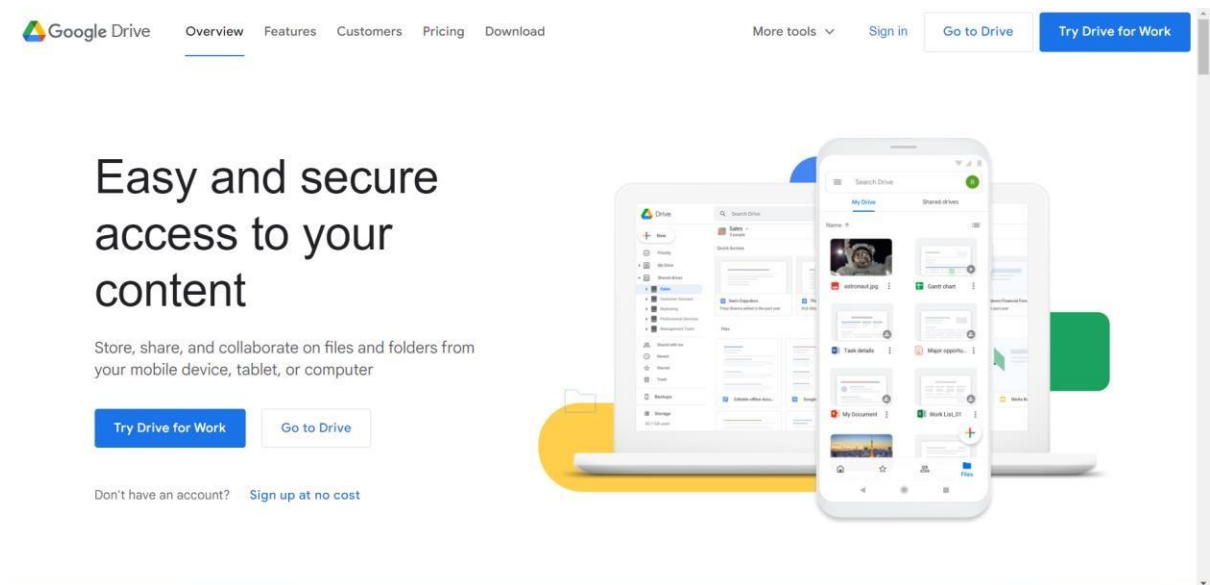
(The Uploaded Image Is Stored In The process !!)



The Output is final verify



(Through “Predicted Result” The Image Classification Is Visible..)



(Click Google Drive Button For Sharing Purposes)

(Click Choose File To Upload Images)