In this project we are detecting whether person has wear a mask or not and to implement this project we have used following packages

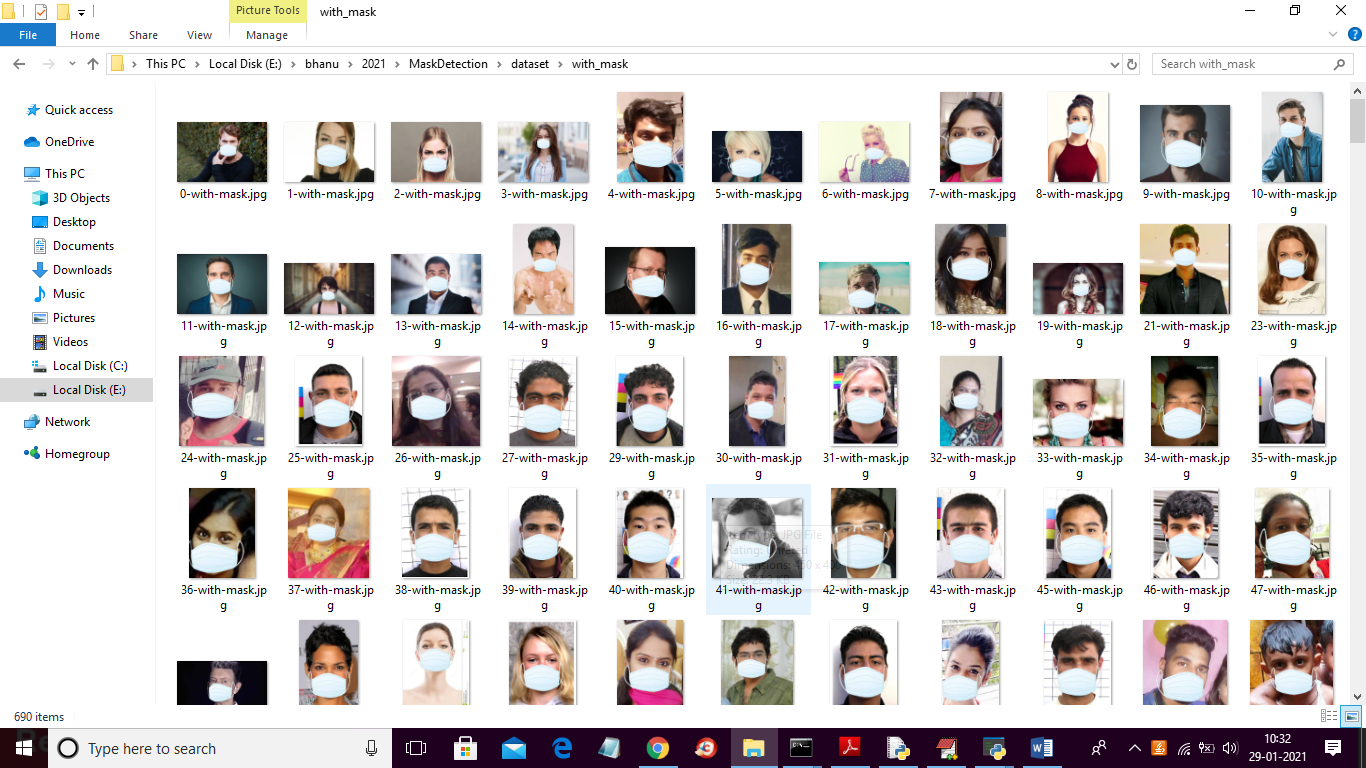
Opencv: using this package we can read and write images and can able to detect faces from images

Keras and tensorflow: tensorflow provide libraries to build CNN model and using keras we can define layers for that CNN model

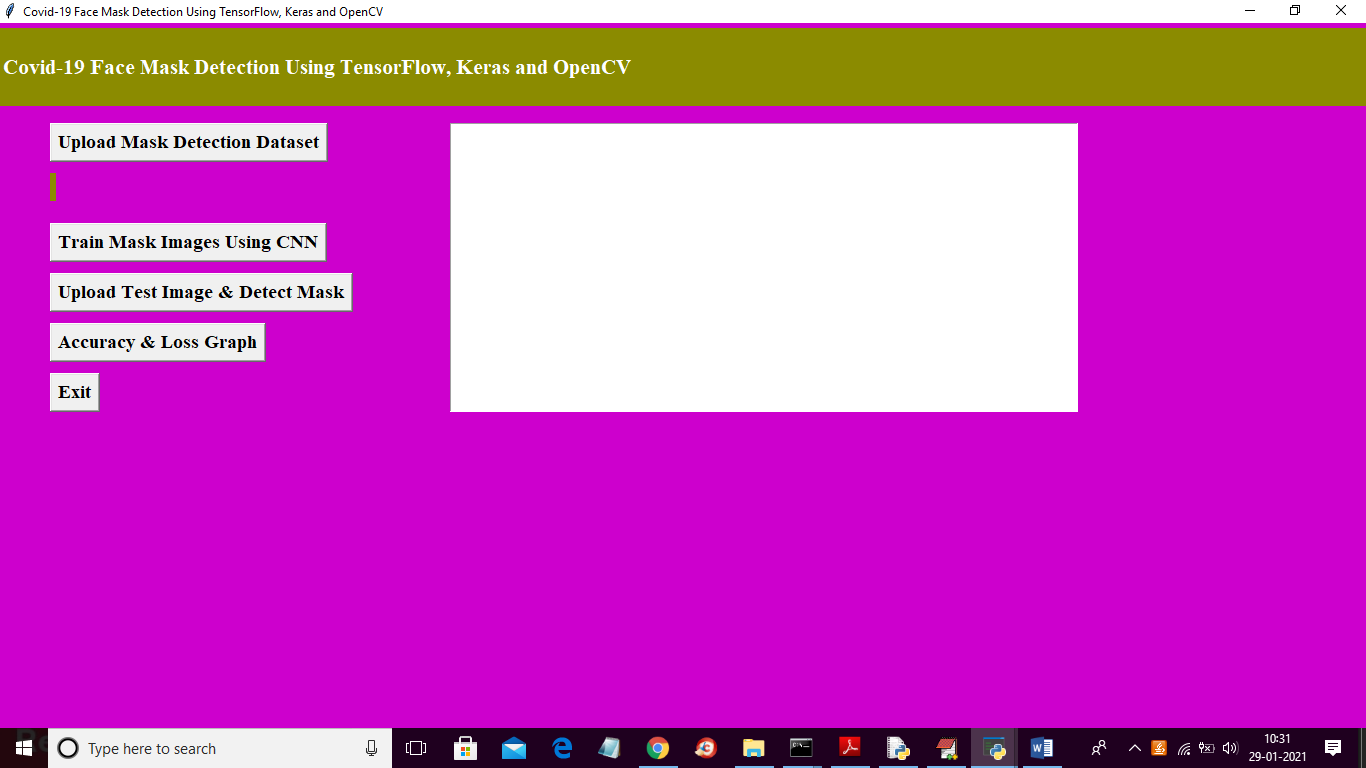
To implement this project we have used dataset from below link

<https://github.com/prajnasb/observations/tree/master/experiements/data>

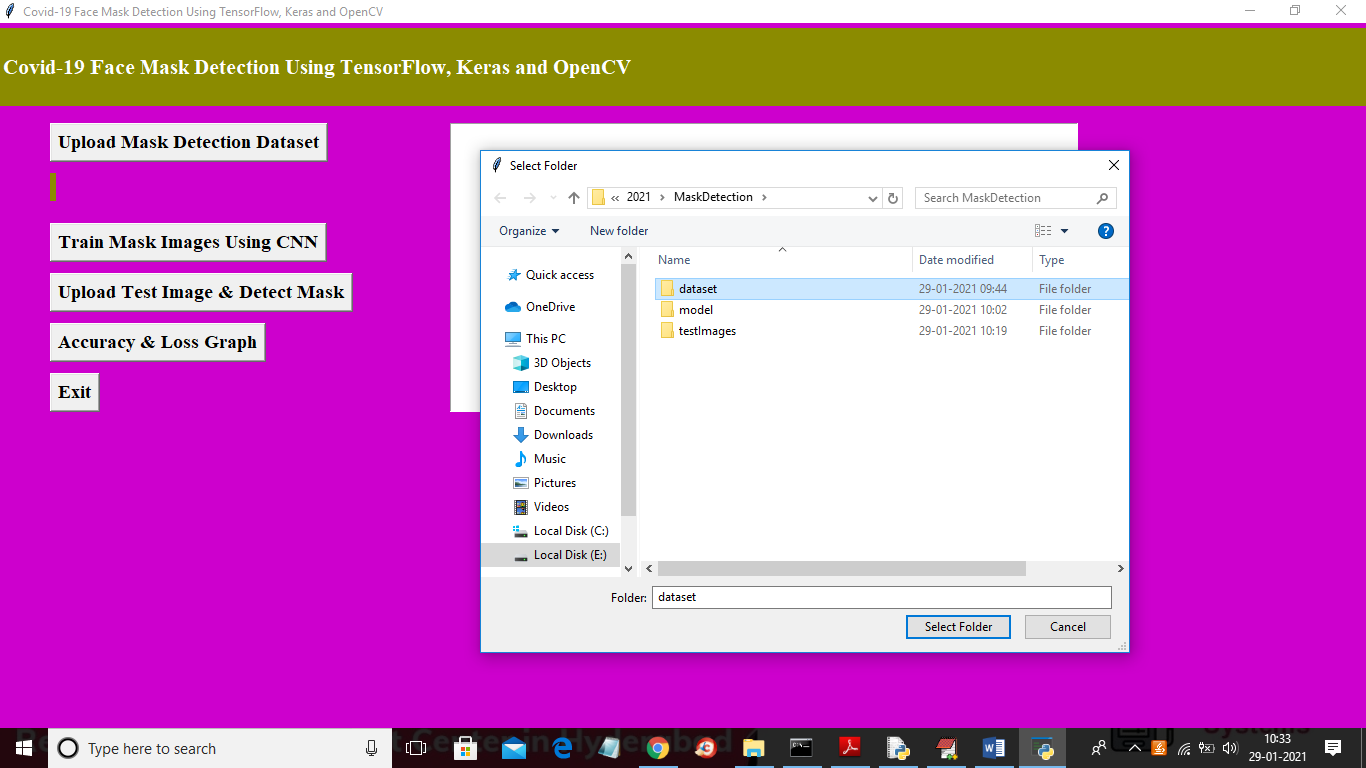
Below screen shots showing images from dataset used to trained CNN model



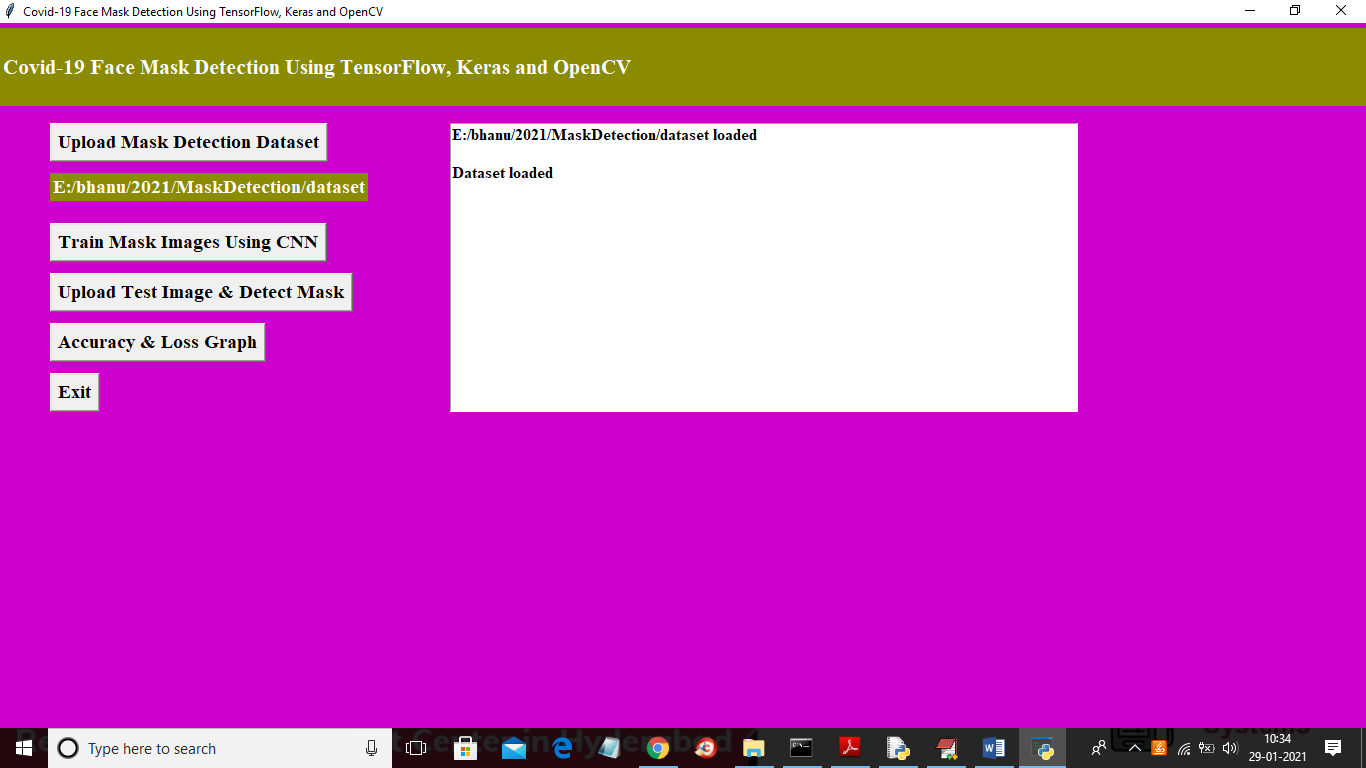
To run project double click on ‘run.bat’ file to get below screen



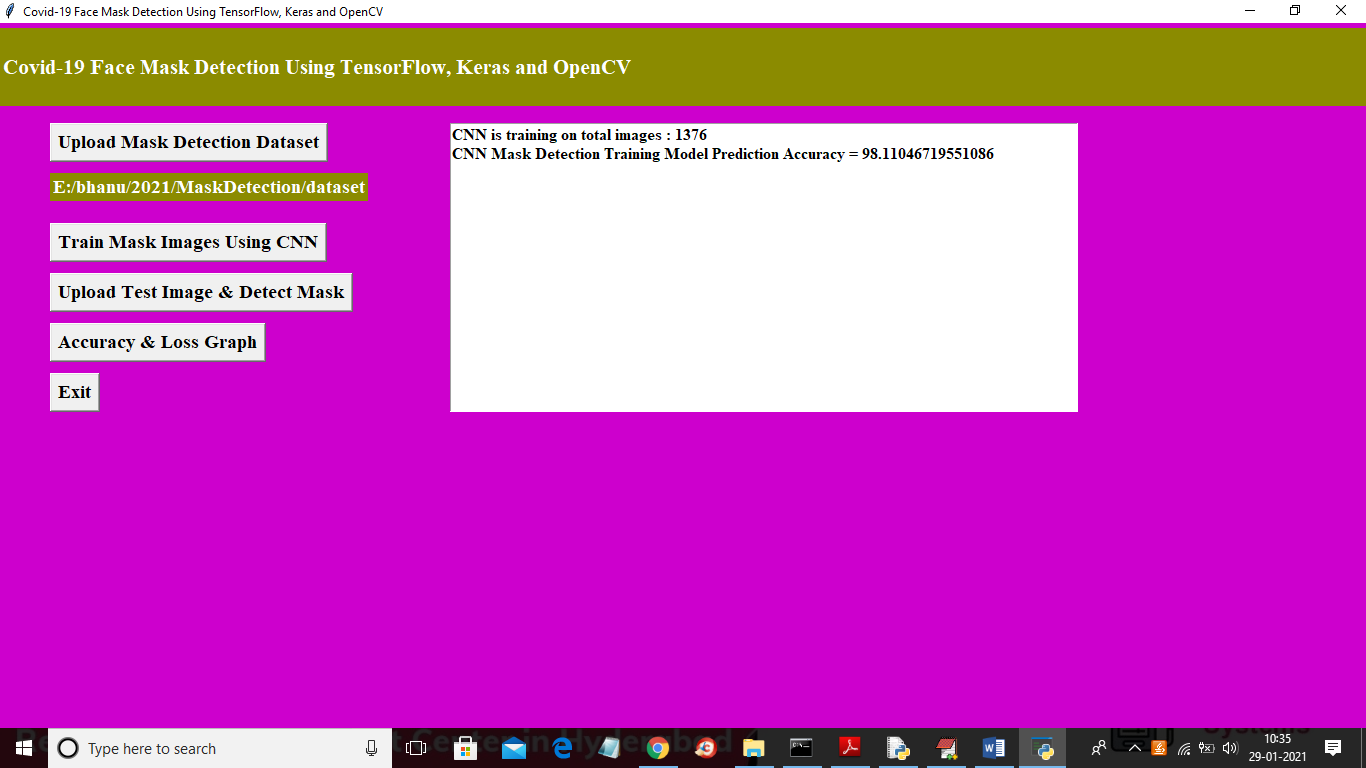
In above screen click on ‘Upload Mask Detection Dataset’ button to upload dataset and to get below screen



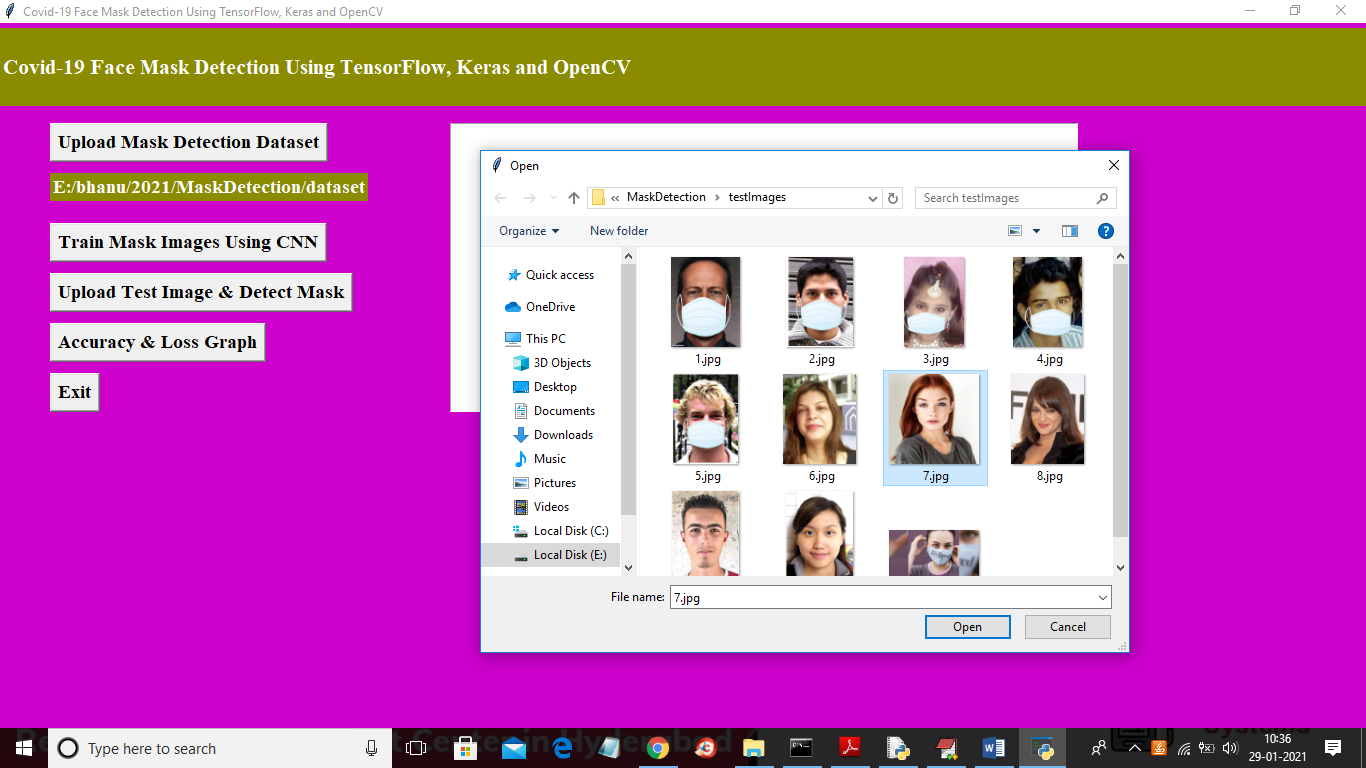
In above screen selecting and uploading ‘dataset’ folder and then click on ‘Select Folder’ button to load dataset and to get below screen



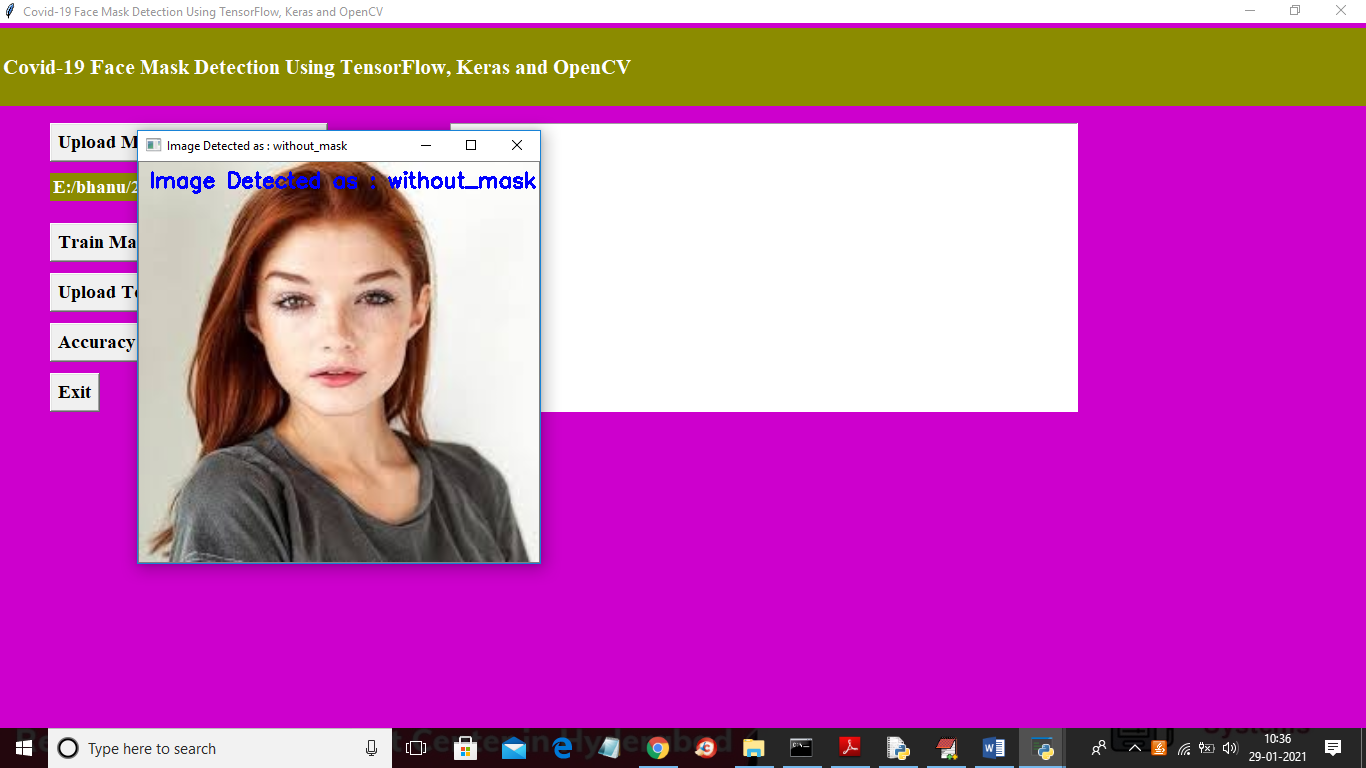
In above screen dataset loaded and now click on ‘Train Mask Images Using CNN’ button to train CNN with loaded dataset and while building CNN application will read images and then preprocess images and then build model



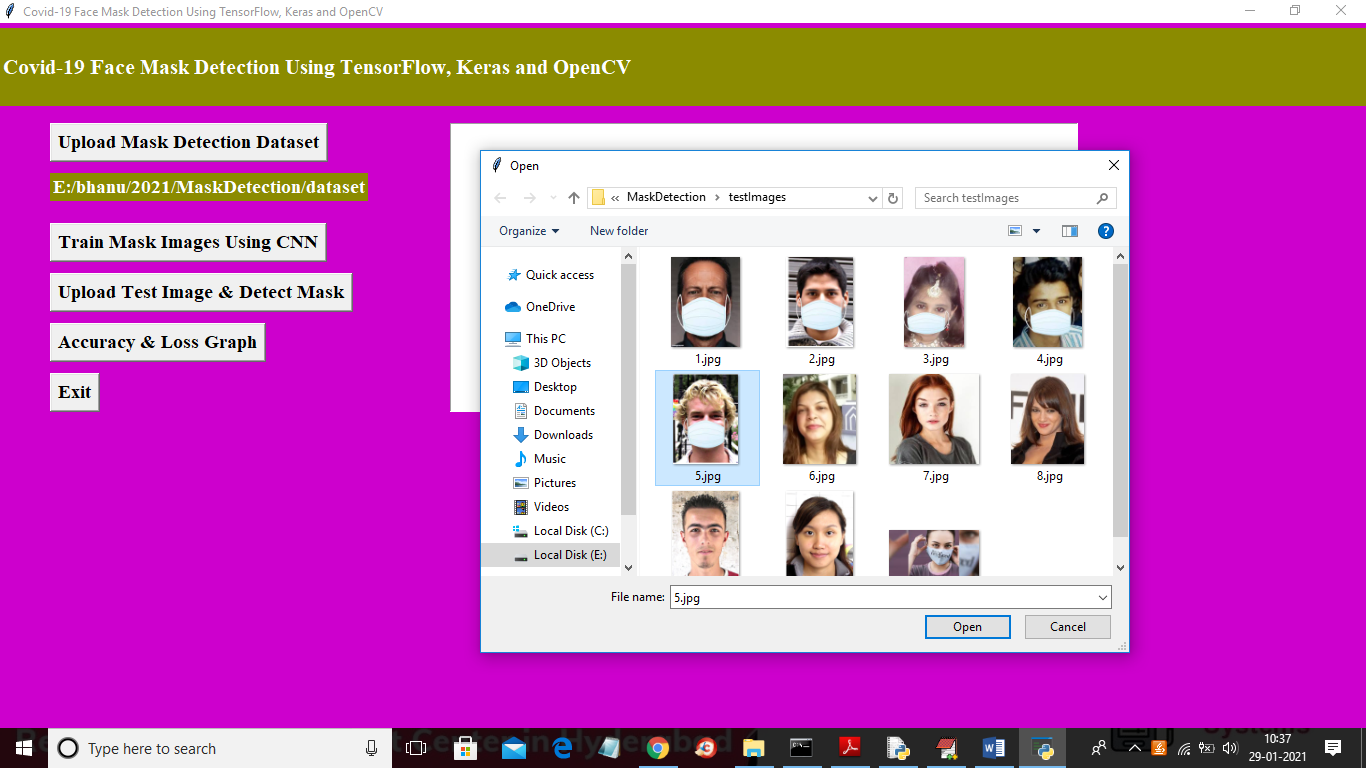
In above screen we can see CNN trained on 1376 images and its prediction accuracy is 98% ad now model is ready and now click on ‘Upload Test Images & Detect Mask’ button to upload test image and then application predict whether person wear mask or not

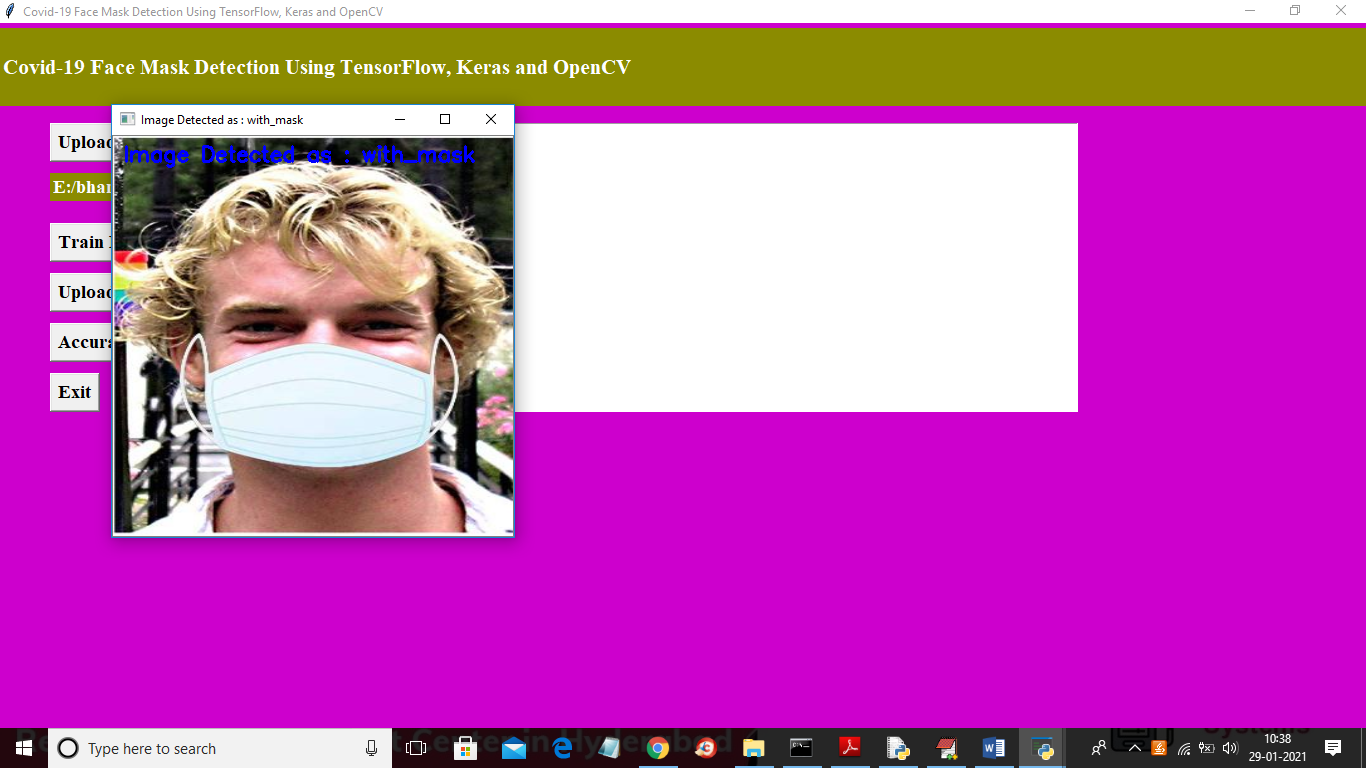


In above screen selecting and uploading 7.jpg file and then click on ‘Open’ button to get below screen

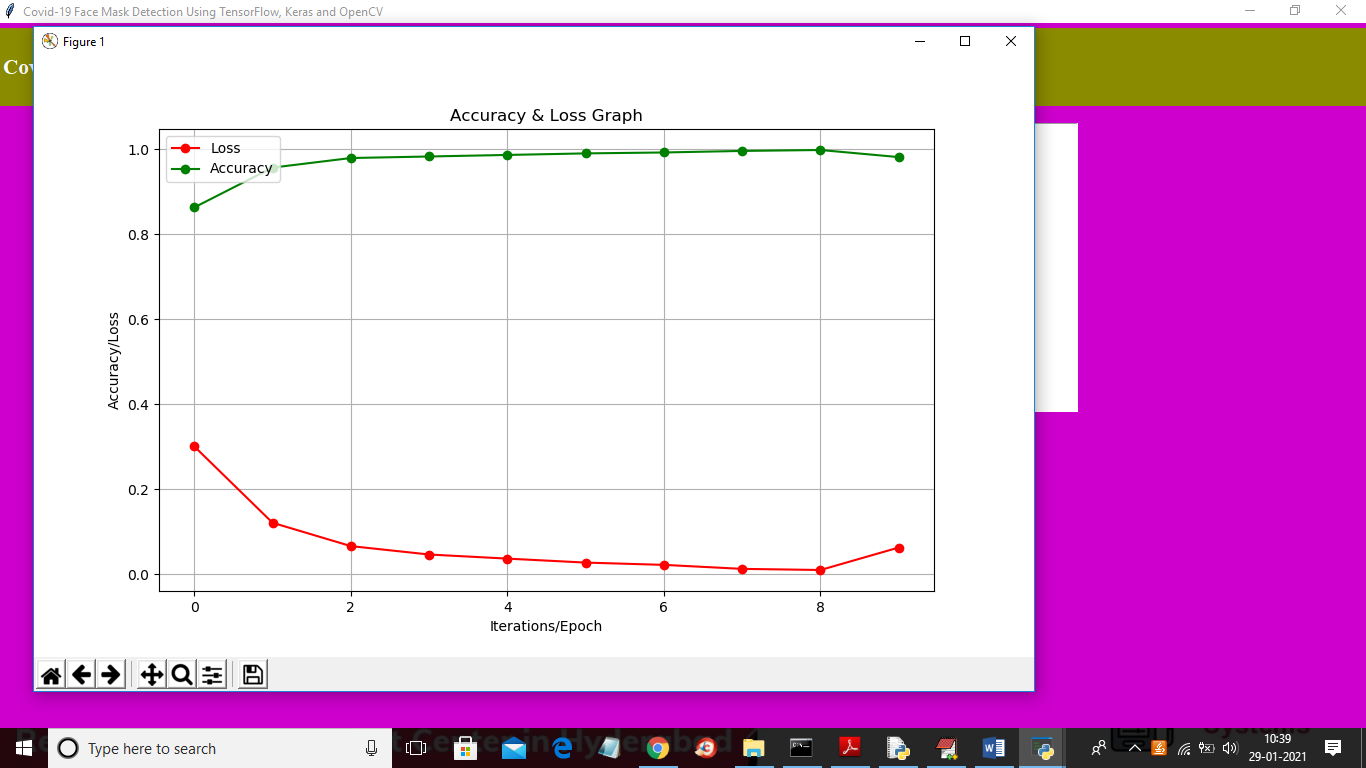


In above screen application saying image is without mask and you can see predicted result in image title bar also. Now test with other images

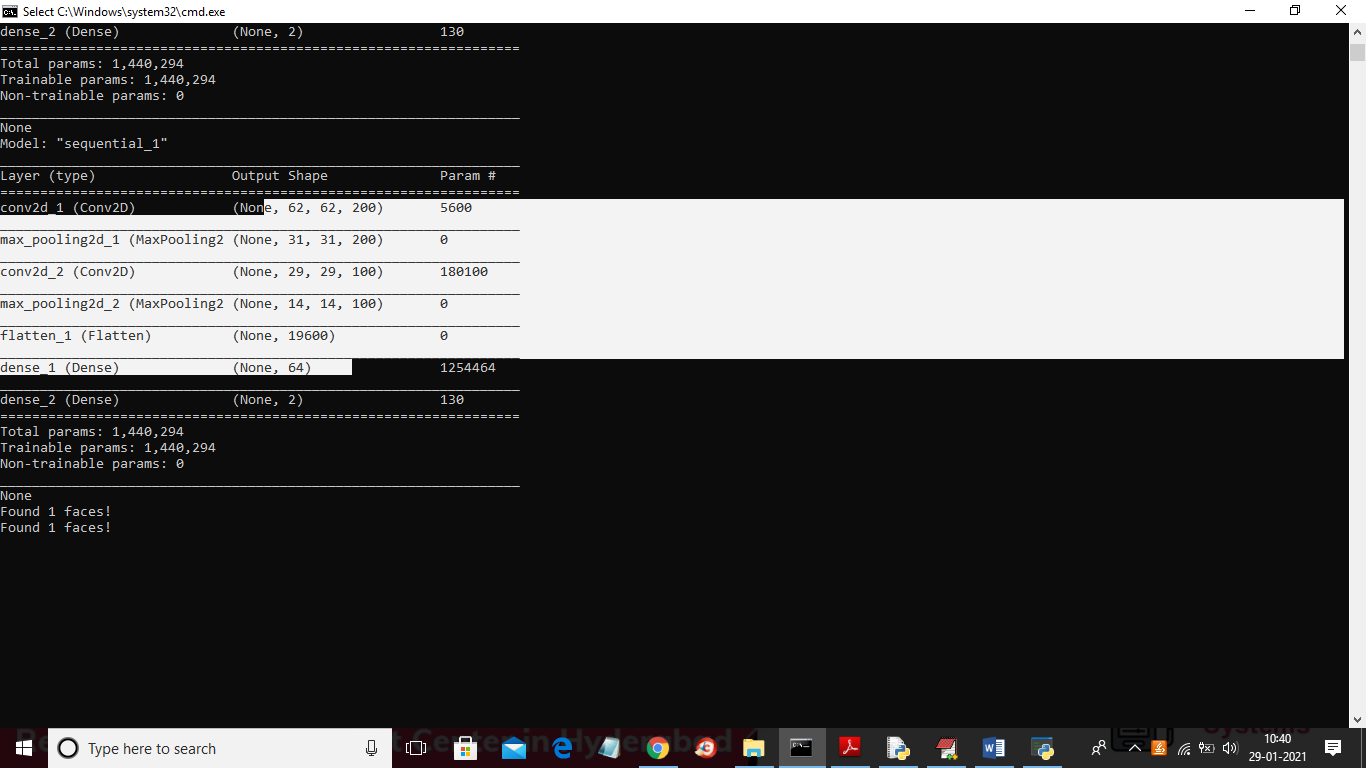




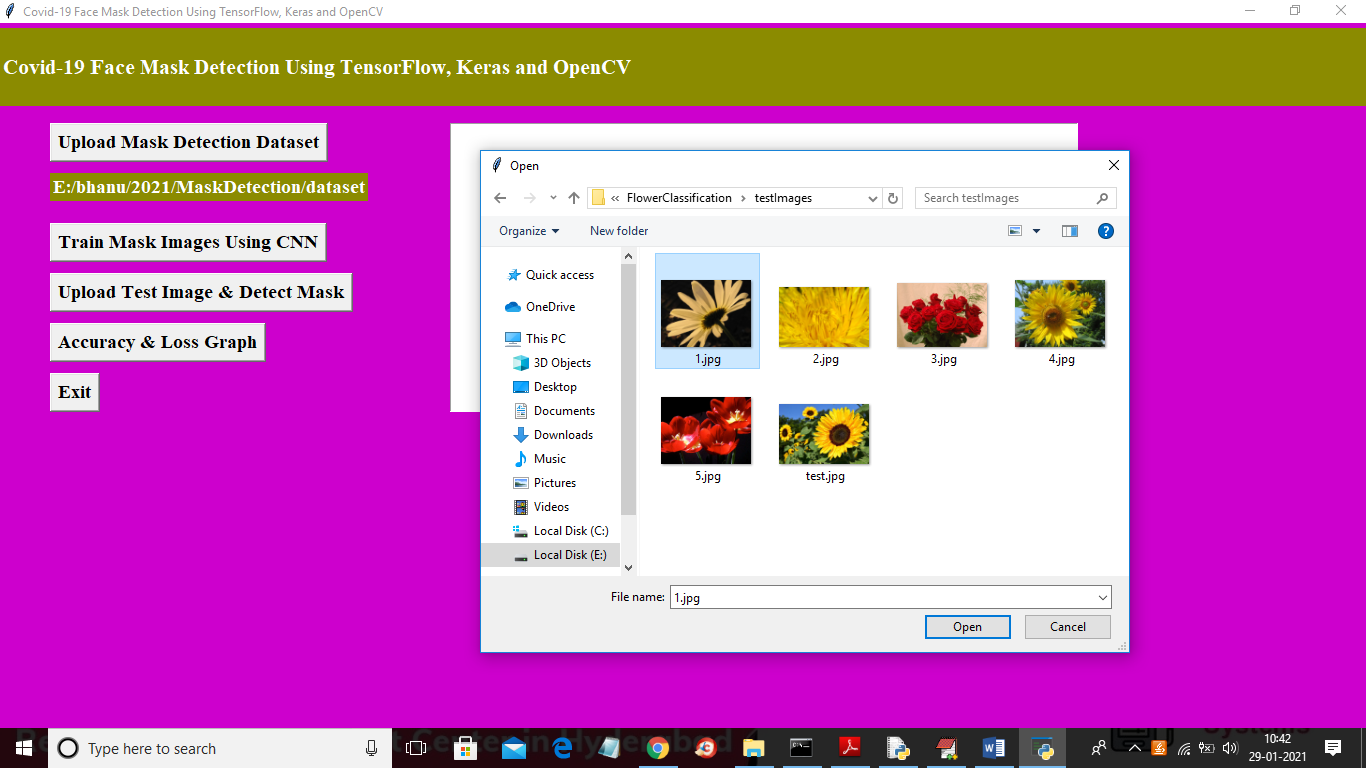
Similarly you can upload any image and application will perform detection. Now click on ‘Accuracy and Loss Graph’ button to get below graph



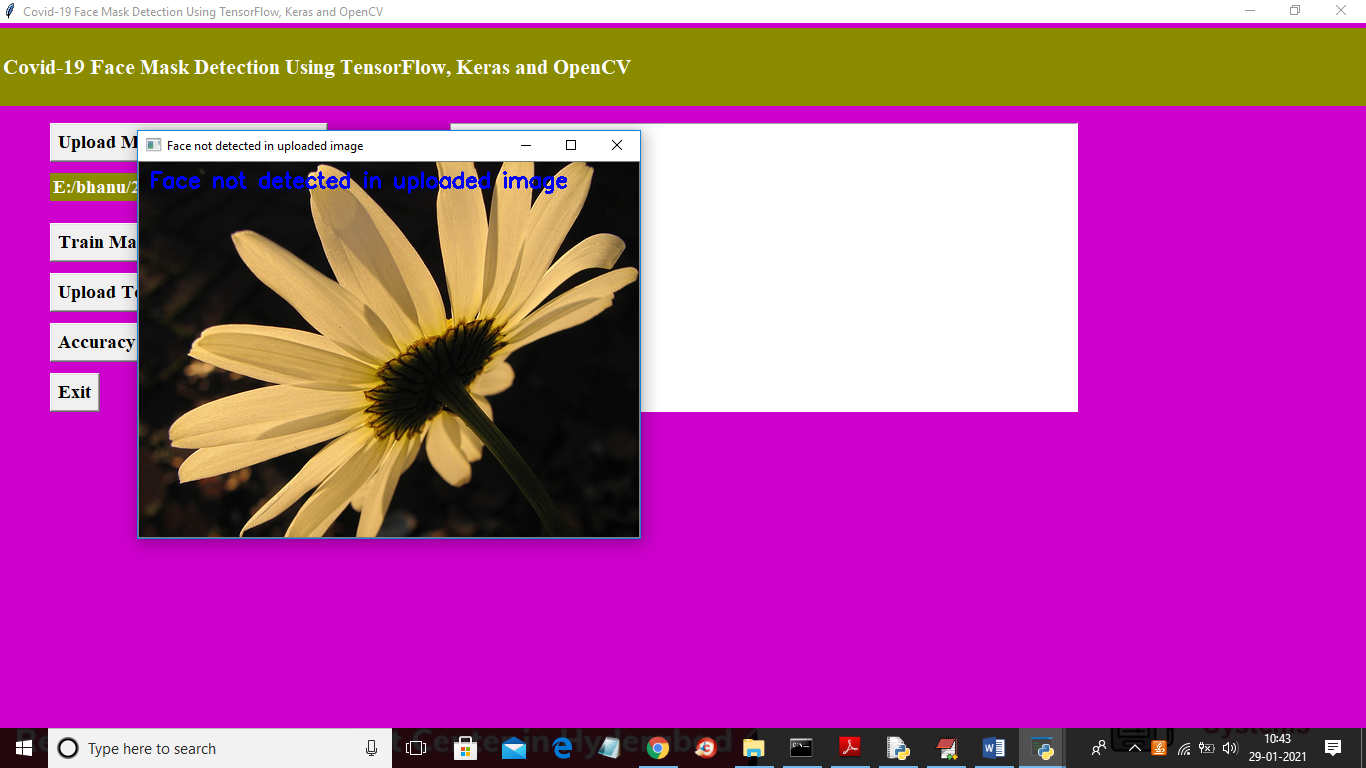
In above screen red refers to loss and green line refers to accuracy and x-axis represents EPOCH and y-axis represents accuracy and loss value. In above graph with each increasing epoch accuracy get better and loss get decrease. Below screen showing CNN model with 200, 100 layers for input and 64 for output as given in your paper algorithm



In above screen multiple layers created where 62 X 62 is the image size and 200 is the filters for first layer. In below screen we are showing output with out face



In above screen selecting and uploading 1.jpg and below is the prediction result



In above screen application saying face not detected in uploaded image.

Packages

Pip install opencv-python==4.1.1.26

Pip install keras==2.3.1

Pip install tensorflow==1.14.0

Pip install pickle5==0.0.11

Pip install matplotlib==3.1.1

Pip install numpy==1.19.2