MACHINE LEARNING PROJECT

Identifying if a person is wearing a mask (N-95)

INTRODUCTION

The coronavirus is a respiratory virus which spreads primarily through droplets generated when an infected person coughs or sneezes, or through droplets of saliva or discharge from the nose. Using the available data, the minimum time to clinical recovery for mild cases is approximately 2 weeks and is 3-6 weeks for patients with severe or critical disease. you only need to wear a mask(N-95) if you are taking care of a person with Corona Virus infection. Wear a mask if you are coughing or sneezing. Masks are effective only when you use it with frequent hand-cleaning with soap and water. If you wear a mask, then you must know how to use it and dispose of it properly. A mask shall be used for a period of 2 to 3 days approximately.

Due to the distribution of duplicate face mask it becomes our responsibility as an engineer to devise a method that shall identify whether a person is wearing the correct face-mask(N-95).

What is a N-95 mask?

An N-95 respirator mask is a respiratory protective device with high filtration efficiency to airborne particles. To provide the requisite air seal to the wearer, such masks are designed to achieve a very close facial fit.

N-95 MASK DETECTION is an act of determining whether the person is wearing the particular mask which he should wear in order to protect himself from the disease. The successful detection of face-mask could yield significant result by reducing the spread of the disease on a large scale.

The **Chief goal** of this project is to detect the face-mask by using the available machine Learning techniques and concepts, the hope is that by understanding the query we can yield a significant result and help the world during this current pandemic.

ABSTRACT:

The aim of this mini project is to examine a number of images and predict accurately who is wearing a mask and who is not wearing a mask in an image. We do this by applying supervised learning method of image processing by preparing and training a dataset.

Proposed solution:

Implement a python script to train a face-mask detector model on the prepared dataset using keras and TensorFlow. Now with this trained face-mask detector we will be implementing an additional python script to detect the face-mask in an image.

In detail it can be given as two steps.

- Training - Here we are loading our face mask detector dataset from disk, and then train a model (using Keras/TensorFlow) on this dataset, and again add the face mask detector to disk.

- Implementation – When the mask detector is trained, we can then load our mask detector, performing face detection, and then identifying whether the face has mask or not.

Some of the steps that need to be done for dataset collection are:

1. Pre-processing and Cleaning

- Recovering the missing data and removing the redundant data. This step involves creating new features from existing ones.

2. Feature Extraction

- This step involves searching the possible feature subset. We then pick the subset that is optimal or near optimal with respect to some objective function. This is done to avoid problems of overfitting or underfitting the dataset.

3. Data Normalization

- Data is needed to be normalized for better accuracy by ensuring that all features are not given excessive low weightage.

DATASET:

The dataset that we are going to use here are images of people wearing their masks and not wearing their mask. We preserve some pictures and consider them as our validation dataset (the images that are not used for training).

Condition:

The dataset that we are preparing should be balanced. i.e the range of images of people with mask and people without mask should be negligible. When the difference in range is large then the model would be biased and will not generate the required result (will provide with incorrect result).

IMPORTANT LIBRARIES AND PACKAGES:

- Keras- open source python library used for evaluating and training my model. (supports only high-level API).
- Tensor flow- for performing fast numerical computations. This also used to create models directly or by using libraries. (support both high level and low-level API).

- NumPy – NumPy array is used to store the pixel value of an image in a multi-dimensional array. These arrays can be used for face detection in image processing.
- Matplotlib (by choice) - for creating interactive figures that can be easily read by outsiders.
- Argparse – used for parsing the command line arguments .it is used for involving command line arguments to the program.
- Scikit – learn: Provides various algorithm like regression, clustering and classification.
- Imutils – for specifying the path where our input dataset are placed.

CODE IMPLEMENTATION:

Before writing the python script we should install the necessary libraries. These libraries can be installed by using the Anaconda command prompt. Once the necessary libraries have been installed, we shall move onto our next step.

For running our python script, we have to launch the Spyder IDE using the Anaconda navigator. Once everything is set, we shall begin our work.

WHAT IS SPYDER ?

Spyder is an open source cross-platform **integrated development environment** (**IDE**) for scientific programming in the Python language. This IDE can be used to execute our python script.

We will be having two python scripts one for training the neural network model and another one detecting if a person is wearing a mask or not.

1. TRAINING THE MODEL WITH OUR DATASET.

Here we will be developing our neural network classifier.

STEPS THAT ARE FOLLOWED:

- Get the dataset, label it and pre-process (resize, scatter, colour change) the image.
- Performing one hot encoding on the pre-processed data (converting the label to zero and one).
- Training the neural network classifier. (embedding the fully connected layer).

- Saving the trained model to our disk.

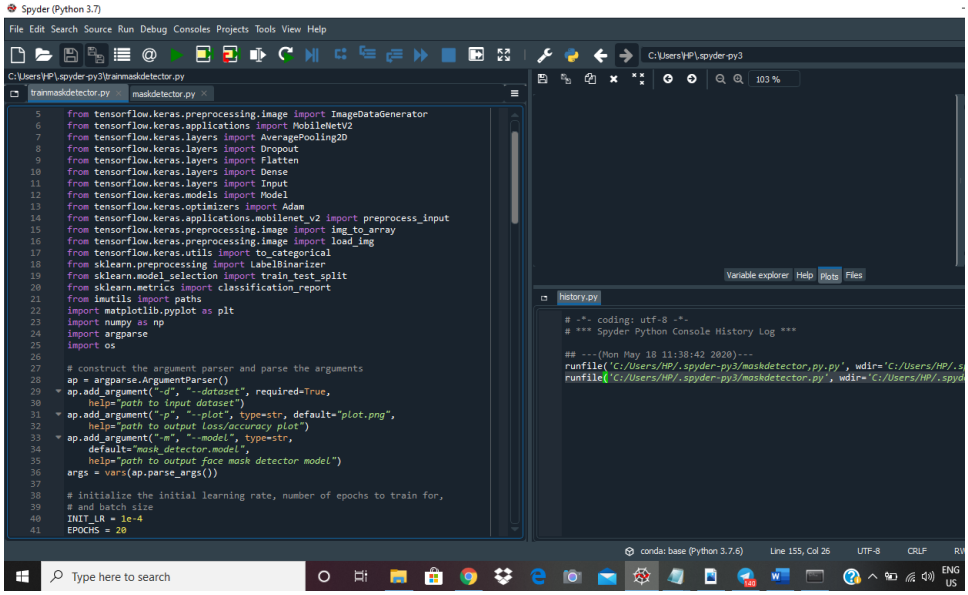

Fig 1.1

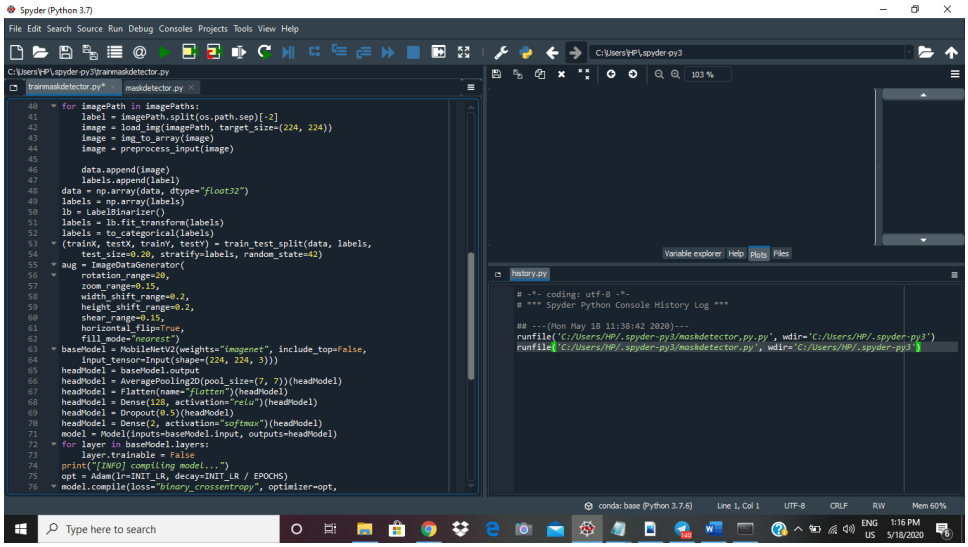

Fig 1.2

- Using the mobilenetv2 architecture (already trained model) to create our own model and saving the model using the model object.
- Finally performing data augmentation (it stores/gives our image as 5 different variants).

2.PREDICTING (CLASSIFYING THE IMAGE) USING THE TRAINED CLASSIFIER:

Here we will be giving our image (validation set) to the face detector model which will give us the facial features and the model that we have trained will be applied over the extracted facial landmarks. It performs mask detection in static images.

SOME OF THE STEPS FOLLOWED ARE:

- Load the saved models.
- Pre-process the input image and give to face detector model.
- The face features are stored in an object named Blob which is given to the face detector model which predicts the result for us.

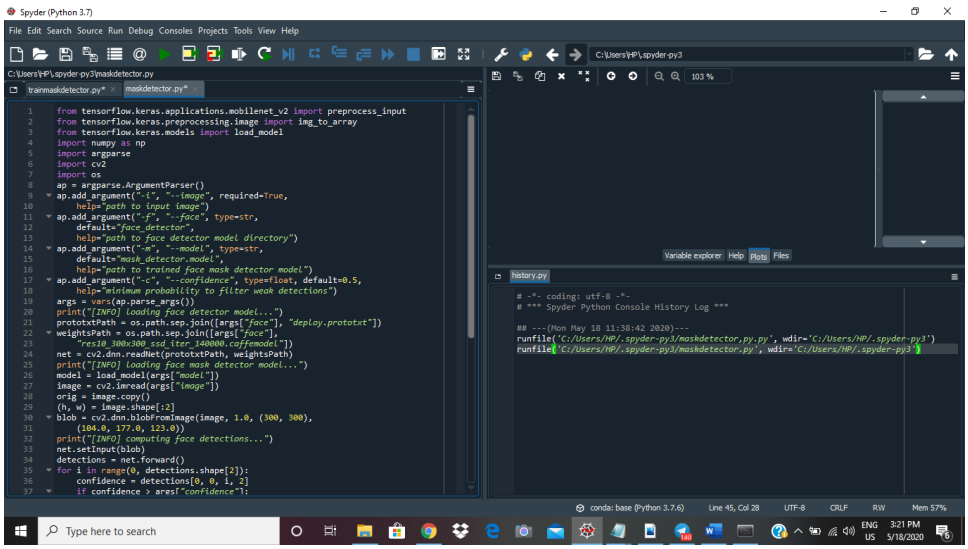

Fig 2.1

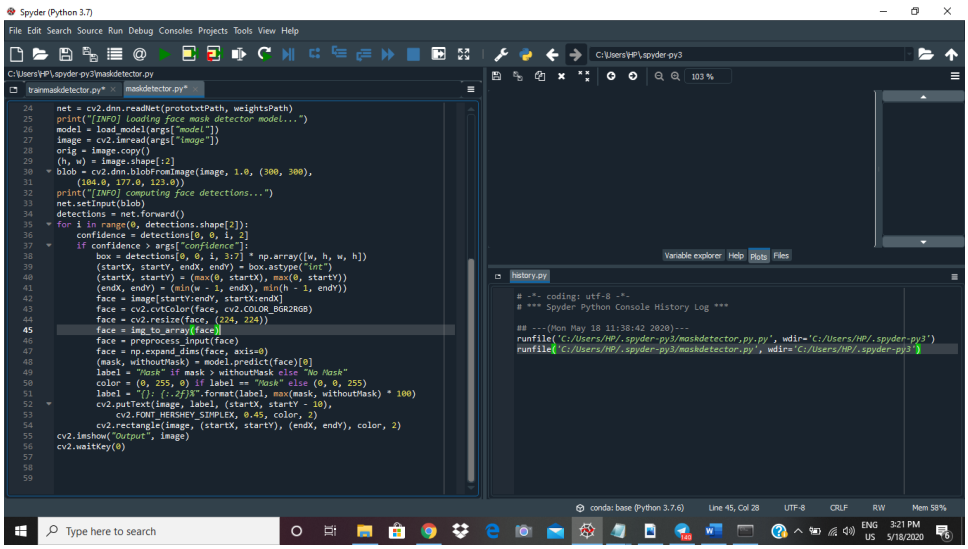

Fig 2.2

- Here we will be loading an extra model (face detector model) that will be useful for face detection. This model is an already trained model. we will be loading this model from the disk using the OS library (for specifying the path). The input images are resized to the same exact size that is used in the dataset with the help of imutils library.

CONCLUSION AND FUTURE WORK:

Detecting the mask is pretty difficult. As we saw in the documentation many different aspects (variables) need to be considered in order to make a good prediction.Some of the reasons that this model may behave differently (inaccurately) will be due to the vast difference in the number of images with mask and without mask. When the difference is large the model may become biased and will not predict the results accurately. If we could detect the particular type of mask accurately then it would be very beneficial. That however is impossible with this program we could detect the masked face in an image but cannot find whether the person is wearing that particular type of mask.