**Mask Detector**

**Run application:** First install the dependent libraries using requirement.txt file using command “pip install -r requirements.txt”

To run the application cd the path to application directory via cmd. Run command “python MaskDetection.py “

To quit the application press in the key “q”

**DataSet:** The dataset contains images of people with mask and without mask that has been taken from internet sources (opensource image libraries, Kaggle and google images) there are about 1915 images of each type. The model to perform mask detection was trained on this data.

**Approach**: The dataset I have doesn’t have bound box data. It just has the label for binary classification, so best approach should be to use 2 different models 1 for classification of mask or no mask and 2nd model to detect face. For face detection I have used pretrained SSD model. For classification task I have used MobileNetV2 with additional top layer for classification. As I am using pre trained MobileNetV2, while training I have freeze the base model which is MobileNetV2 and trained the top layer.

**Training the model:** The first step in model training is to initialize the number of epochs(20) to train for, batch size (64) and learning rate=1e-4 . Once this is done, we can move ahead and grab the list of images in our dataset directory and then initialize the list of data and class images. Then we need to perform one-hot encoding on the labels and then construct the training image generator for data augmentation. Once this is done, we load the MobileNetV2 network, ensuring the head FC layer sets are left off and construct the head of the model that will be placed on top of the base model. Loop over the base model and freeze them so they will be updated during the first training process. Then we place the head FC model on top of the base model this is the model we will train on. For optimizer, I have used adam and for loss cross entropy is used.

**Apply mask detector over images** : For this step grab the dimensions of the frame and then construct a blob from it. OpenCV dnn – deep neural network module contains a function that can be used for preprocessing images called blob .

So, in the first step, just grab the frame and construct a blob from it and then pass it through the network and obtain the face detections. We then pass the blob through the network and obtain the mask classification. Once we’re done with that, the next step is to initialize the list of faces, their corresponding locations and the list of predictions from the face mask network. Using outputs I have generated bound box coordinates for the face detection and with label suggesting mask or no mask.