PROJECT REPORT

COVID-19: Face Mask Detector with OpenCV, TensorFlow, and Deep Learning

INTRODUCTION

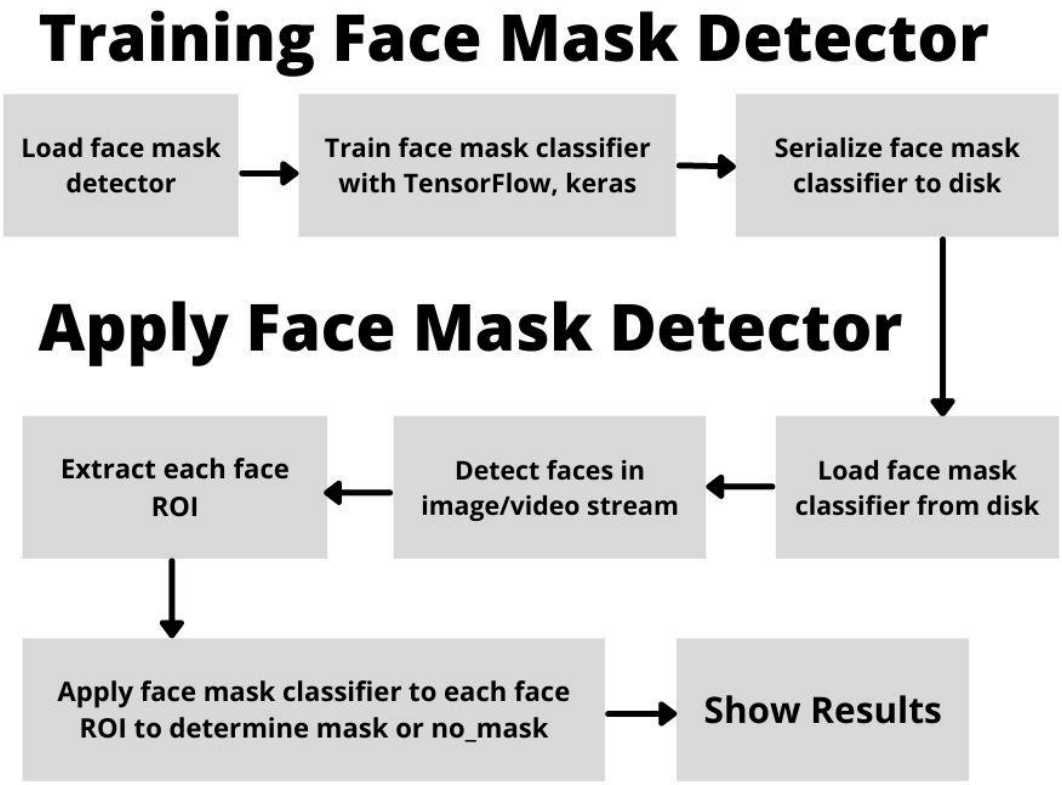
Some measures have to be taken to mitigate the spread of COVID-19. Even though most of them are following norms, some are not following them due to their irresponsibility. And wearing a mask frequently or for a long time can cause irritation that is caused because the mask traps dirt and oil in pores. So, some people tend to remove their masks in public places like shopping malls and stores, which may become the primary cause of spreading coronavirus. As it is challenging for the guard or workers or officers to detect them in public places, we would like to make a face mask detector which can make their task easier and stop the spread of covid.
Our goal is to train a custom deep learning model to detect whether a person is wearing a mask or not in both images and videos.

IMPLEMENTATION DETAILS

We have used TensorFlow as a deep learning library. In the first steps, we have imported various types of libraries we will use for training. Some standard libraries-TensorFlow for training our model,matplotlib for visualizing our model to see how our training accuracy and validation accuracy increase in both the training and validation phases. We are also using a famous machine learning library-scikit-learn(sklearn) for label_encoder and train_test_split, i.e., splitting the dataset into training and testing.
In the second step, we have loaded our dataset for training, which has 2 classes (one contains 500 images of faces with masks and 500 images of faces without masks). Before feeding the dataset to the model, we have done some primary label encoders-Image as feature and target as a class.
In the third step, we will initialize our hyperparameters that are batch_size, the number of epochs we will use, and the learning rate. Since our computer always deals with numeric data. It doesn't understand any string data, so we must change the labels(face_mask and without_face_mask) into a number using a label encoder. After that, we have partitioned our dataset into training and testing and performed data augmentation to overcome overfitting our model.

The fourth step is model building- in which we will use a pre_trained model -mobile_net. As we are using a pre_trained model, we will only train the head part instead of the whole body.
In the fifth step, we will perform our training. With epochs, we have taken as 20 and loss function we have used as binary cross-entropy.
We will evaluate our model in the sixth step to see how our model performs (accuracy and loss of model). We will change some hyperparameter tuning to improve our model accuracy and decrease our loss.
In the seventh step, we will save our model.
During face mask detection, we will use our trained model. Now we will check how our model is working and detect masks in people's faces. So for that, we will use OpenCV to load our image, preprocess our image, and then use our trained model that we have saved.

FLOW CHART



RESULT

We will load any image of a person into face_mask_images.ipynb file and run it. It shows whether the person is wearing a mask or not along with accuracy. We have tested in unseen images also, and our model predicted correctly.Similarly, if we run the face_mask_video.ipynb file, the webcam will pop up and show the results.
On average, our model accuracy is approx 90%.