Real Time Face Mask Detector

TEAM NAME: N00BS

Janit Lodha (Team Leader)
Indian Institute of Information Technology, Allahabad

Arpit Agarwal
Indian Institute of Information Technology, Allahabad

Tamoghno Bhattacharya
Indian Institute of Information Technology, Allahabad

Mridul Mittal
Indian Institute of Information Technology, Allahabad

IDEA/APPROACH

In the present scenario due to Covid-19, there are no efficient face mask detection applications which are now in high demand for transportation means, densely populated areas, residential districts, large-scale manufacturers and other enterprises to ensure safety. Also, the absence of large datasets of 'with_mask' images has made this task more cumbersome and challenging.

In the COVID-19 crisis wearing masks is absolutely necessary for public health and controlling the spread of the pandemic. So, what if we could have a system that could monitor and supervise people complying with the safety measures like wearing masks.

We tried to build a model which would be able to identify which person has not worn a mask in a group of people who are wearing them. We also try to deploy our model so that it could be used by those people too who don't have much knowledge about deep learning.

We built our own CNN model which gives an accuracy of around 98%. Also, the use of Transfer Learning has been done in order to reduce the training time and further enhance accuracy.

DATASET

We have obtained the dataset from this link (we will still be looking for a better data set):

https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset

TECHNOLOGY STACK:

Numpy, pandas, matplotlib, seaborn, keras, sci-kit learn
Spyder & Jupyter Notebook(Anaconda Environment), Kaggle Notebook & Google Colab.

USE CASES:

- **Retail**: Retailers need to monitor their premises to control the current occupancy and wearing of masks.
- **Public Transport** : Wearing face masks in public transport will be mandatory in many parts of the world. Public transport organizations can use the software to automate the checking process with very little resources needed.
- **Airports, Hotels, Hospitals** : Our face mask detector can be very effectively used at airports mainly for entrance flow management and monitoring. The software can be added to any access gate or entrance to make sure that entry is allowed to any person only if he/she follows the safety rules.
- **Corporate Buildings** : With many office buildings opening up and employees coming back to work, face mask detection can be used to maintain a safe environment for everyone. The mask detection system can also be combined with facial recognition which uses employee database information to match an individual at the office entrance to only mark him/her present for that day if they are wearing masks.

SOCIAL IMPACT ANALYSIS WITH COVID19

This model can be deployed on data provided by CCTV's all over the world and help the government keep a track of people with no mask as in the current covid situation wearing a mask is a necessity for both self and others safety. Using this will be a lot of help for monitoring people not wearing masks.

As malls and offices are reopening after the lockdown mask have become a compulsory part and strict fines are there for the violation. It is not possible to physically check masks all day thus this model can be used in mall entrances and office entrances and to allow entry only if masks are detected.

Attendance in school can be checked at the gates using this model. Places like metro stations and offices where a ticket or id is required we can also use this as an authentication criteria and only allow it if a mask is identified.

MODEL IMPLEMENTATION

STEP-1 (DATA PRE-PROCESSING)

The real challenge in coming up with any Neural Network or Machine Learning model is pre-processing the data and generating it into a ready-to-use form for feeding it to the model.

The thought process for the same is stated below.

We download the dataset from the link stated above which is already segregated into training, validation and testing set. For our current use we will not use validation set rather combine validation and test set as one and test on them. Our final motive is to build a model having such accuracy that it could be used and applied in the CCTVs and other detectors.

- ❏ Training Set will be used to train the neural network for recognizing the difference between a masked and a non masked person.
- ❏ Validation Set would be used to check the accuracy and the fitness of the model.

We will then create our training and validation data generator using ImageDataGenerator and then applying Data Augmentation on the training data to get a variety of samples from a limited data set. Most of the time while dealing with image classification, the dataset is insufficient as per the industrial and real-life usage as the problem can still remain unsolved when there is bad lighting or many other similar problems.

ImageDataGenerator helps us to create augmented data like zooming images and resizing and colors of images in the training set. It is used for making better training data.

From the data generator created in the previous step, we will try to create the training and test Set for finally feeding it to the model by specifying the uniform input image_size and batch_size.

STEP-2 (MODELS TRAINED)

First let's discuss our initial model we have developed. We have taken our pre-processed data and done the following -

- ❏ Created 4 Convolutional layers followed by Max Pooling layers each. Kernel sizes are fixed to 3*3 for the Conv2D layers and 2*2 for the Max Pool layers respectively.
- ❏ Applied a Dropout layer for regularization at the end.
- ❏ Applied flattening and used 3 fully connected layers to make the final prediction.

We have used relu activations wherever required (except for the final layer which requires a sigmoid activation). The optimizer used is Adam, the loss function is the standard binary

cross-entropy, and the accuracy metric for validation. The model has been trained for 10 epochs, with a batch size of 128 and default Adam learning rate of 0.01 to reach an accuracy of 98.84% on the training set and 98.79% on the test set.
We have also tried a transfer learning procedure using the Mobilenetv2 architecture, to perform classification on our dataset and achieved similar levels of accuracy.

STEP-3 (DETECTION USING OpenCV)

We import the already saved model for detecting faces.

To detect if we are wearing a face mask using our PC's webcam for this, we need to implement face detection. In this, we use the Haar Feature-based Cascade Classifiers for detecting the features of the face.

HAAR CASCADE MODEL:

Haar Cascade is a machine learning object detection algorithm used to identify objects in an image or video and based on the concept of features proposed in the paper Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001.

The algorithm has four stages:

❏ Haar Feature Selection
❏ Creating Integral Images
❏ Adaboost Training
❏ Cascading Classifiers

Initially we will need to train the classifier by providing a lot of positive and negative examples and finally extract the best features. The Haar-like features are used as they are digital images features used for objects recognition, but even that way that's a lot of calculations, so they introduced the Cascade Classifiers, based on the fact that most of the area of a face image is not actually face area so it would be inefficient to apply all there features in a non-face area. Cascade Classifiers are grouped by stages and applied one by one so non-face images are detected in early stages with no need of extra processing.

STEP-4 (Detecting Faces with and without Masks)

We need to label the two probabilities (0 for with_mask and 1 for without_mask).
After that, we need to set the bounding rectangle color using RGB values.
We use the OpenCV library to run an infinite loop to use our web camera in which we detect the face using the Cascade Classifier.
The code webcam = cv2.VideoCapture(0) denotes the usage of webcam. In order to check for a

video or an image we just have to change the argument to their paths. The model will predict the possibility of each of the two classes ([without_mask, with_mask]). Based on which probability is higher, the label will be chosen and displayed around our faces.

FUTURE SCOPE OF THE MODEL

We propose to further develop our model to work on real-time video input so that it can detect multiple faces and whether or not masks are being worn. This can be done using algorithms such as YOLO and non-max-suppression. We can use anchor box-based architecture to determine multiple faces at once and run the classifier over them to predict whether each face is wearing a mask or not.

MODEL DEPLOYMENT

A deep learning & computer vision model can only be useful to the society when that model's insights routinely become available to the users for which it was built.

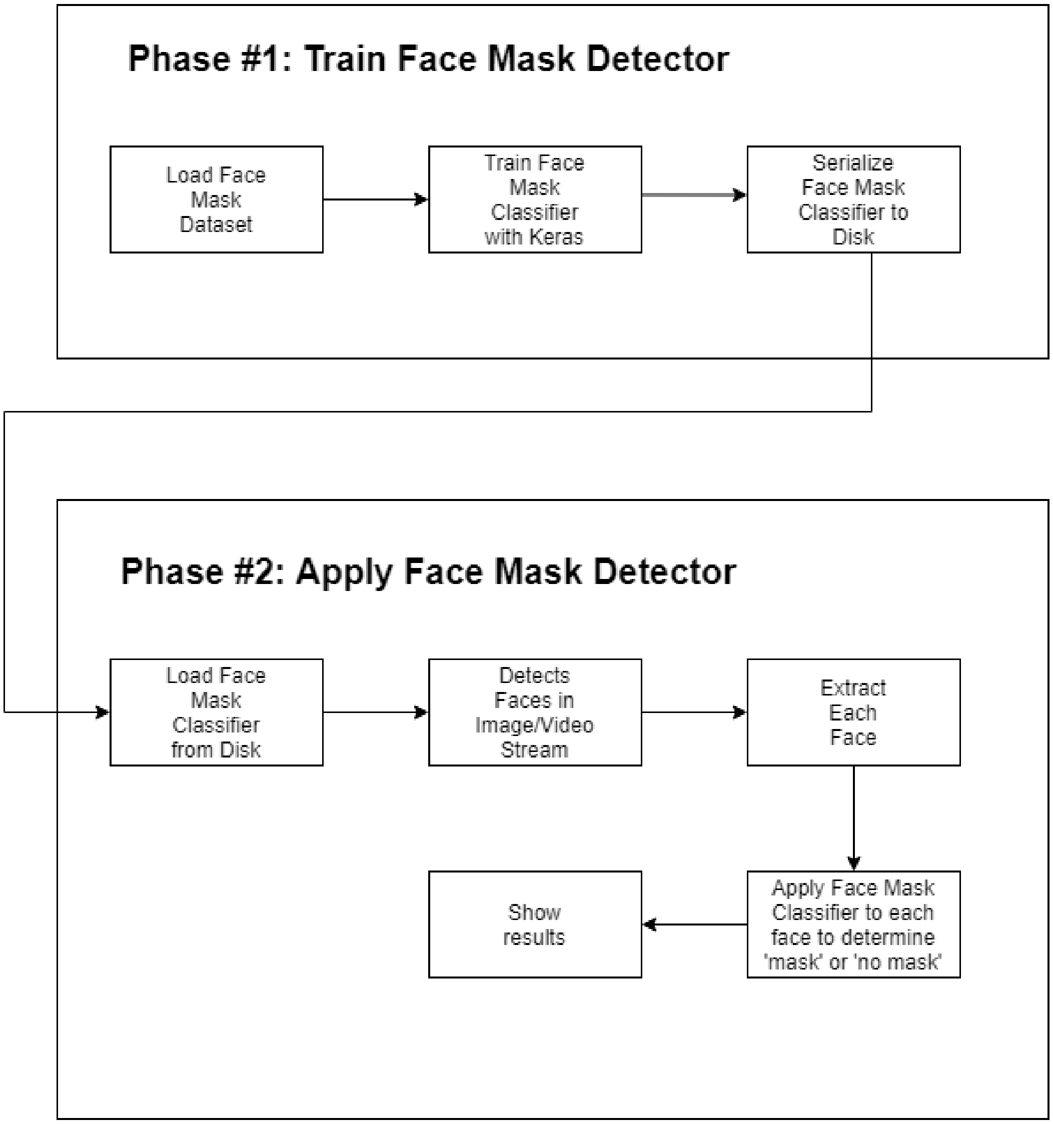
It is not possible for everyone to be able to run a python file. Any model needs to be readily available only then it is considered useful. This limits the reach of various machine learning and deep learning models. This is where open source frameworks like Streamlit and Tensorflow lite come handy as they provide an effortless and an efficient way to deploy our models in websites or apps thus increasing the usability and convenience of the model for common usage purposes very easily.

StreamLit :
Streamlit is an open-source app framework for Machine Learning and Data Science. In our project with the help of Streamlit we can conveniently deploy a web app of our model which will enable us to increase the reach of our model.In this web app user can upload either a image or video from his device or an URL and our model will classify the people with mask and no mask

Tensorflow Lite :
TensorFlow Lite is an open source deep learning framework for on-device inference. It can easily deploy ML models on both ios and android devices. Deploying a model in an app using Tensorflow lite can help a lot as phones have way more accessibility than laptops and PC's and moreover now using a phone camera one can easily identify mask and no mask people from the crowd.

BLOCK DIAGRAM

This block diagram represents the complete process that we propose to do in our defined model.


Phase #1: Train Face Mask Detector
Load Face Mask Dataset
Train Face Mask Classifier with Keras
Serialize Face Mask Classifier to Disk
Phase #2: Apply Face Mask Detector
Load Face Mask Classifier from Disk
Detects Faces in Image/Video Stream
Extract Each Face
Apply Face Mask Classifier to each face to determine 'mask' or 'no mask'
Show results