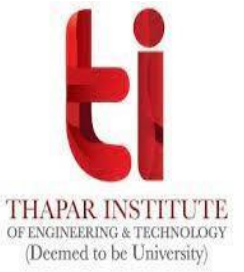
A REAL TIME FACE MASK DETECTION SYSTEM USING CONVLATIONAL NEURAL NETWORK



Submitted By:
Aryaman Kalia
(102003099)
Bharat Dudeja
(102003118)

Submitted To:
Mr. Niyaz Ahmed Wani

July 2022 – December 2022

INDEX

Abstract

In current times, after the rapid expansion and spread of the COVID-19 outbreak globally, people have experienced severe disruption to their daily lives. One idea to manage the outbreak is to enforce people wear a face mask in public places. Therefore, automated and efficient face detection methods are essential for such enforcement. In this project, a face mask detection model for real time videos has been presented which classifies the video frame as “with mask” and “without mask”. The model is trained and evaluated using the data-set from github. The gathered data-set comprises approximately about 4,000 pictures and attained a performance accuracy rate of 98%. The proposed model is computationally efficient and precise. This work can be utilized as a digitized scanning tool in schools, hospitals, banks, and airports, and many other public or commercial locations.

1 Introduction

As the COVID-19 (Coronavirus) pandemic continues to spread, most of the world's population has suffered as a result. COVID-19 is a respiratory disorder that results in severe cases of pneumonia in affected individuals . The disease is acquired via direct con-tact with an infected person, as well as through salivation beads, respiratory droplets, or nasal droplets released when the infected individual coughs, sneezes, or breathes out the virus into an airspace . Globally, thousands of individuals die from the COVID-19 virus daily.

An effective and efficient computer vision strategy intends to develop a real-time application that monitors individuals publicly, whether they are wearing face masks or not. The first stage to identify the existence of a mask on the face is identifying the face. This divides the whole procedure into two parts: face detection and mask detection on the face. Computer vision and image processing have an extraordinary impact on the detection of the face mask. Face detection has a range of case applications, from face recognition to facial movements, where the latter is required to show the face with extremely high accuracy. As machine learning algorithms progress rapidly, the threats posed by face mask detection technology still seem effectively handled. This innovation is becoming increasingly important as it is used to recognize faces in real-time video feeds.

However, for the currently proposed models of face mask detection, face detection alone is a very tough task . In creating more improved facial detectors, following the remarkable results of current face detectors, the analysis of events and video surveillance is always challenging.

Recent years have seen the rise of big data as well as a dramatic rise in the capabilities of computers that use parallel computing. Target detection has become an important research area in computer vision and is also extensively utilized in the real world.With the introduction of convolution neural networks, significant advances have already been madein the field of image classification -

The major contribution of the proposed work is given below:

1. Created a face-mask detector implemented in three phases to assist in precisely detectingthe presence of a mask in real-time using images and video streams.
2. The dataset is made available on GitHub that consists of masked and unmasked images. This data set can be used to create new face mask detectors and use them in a variety of applications.
3. The proposed model has used some popular deep learning methods to develop the classifier and gather photos of an individual wearing a mask and distinguish between face masks and non-face mask classes. This work is implemented in Python along with **Open-CV** and **Keras**. It requires less memory and computational time in comparison toother models
4. The study offers future research recommendations based on the findings for developing reliable and effective AI models that can recognize faces in real-life situations.

Dataset

The dataset used in this research was collected in various picture formats such as JPEG, PNG, and others. Figure 1 exhibits the sample of the dataset. It was a mixture of different open-source datasets and images, including Kaggle's dataset for Face Mask Detection by Omkar Gaurav. As a result, there were different varieties of images with variations in size and resolution. All the photos were from open-source resources, out of which some resemble real-world scenarios, and others were artificially created to put a mask on the face.

Omkar Gaurav's dataset gathered essential pictures of faces and applied facial landmarks to find the individual's facial characteristics in the image. Major Facial landmarks include the eyes, nose, chin, lips, and eyebrows. This intelligently creates a dataset by forming a mask on a non-masked image.

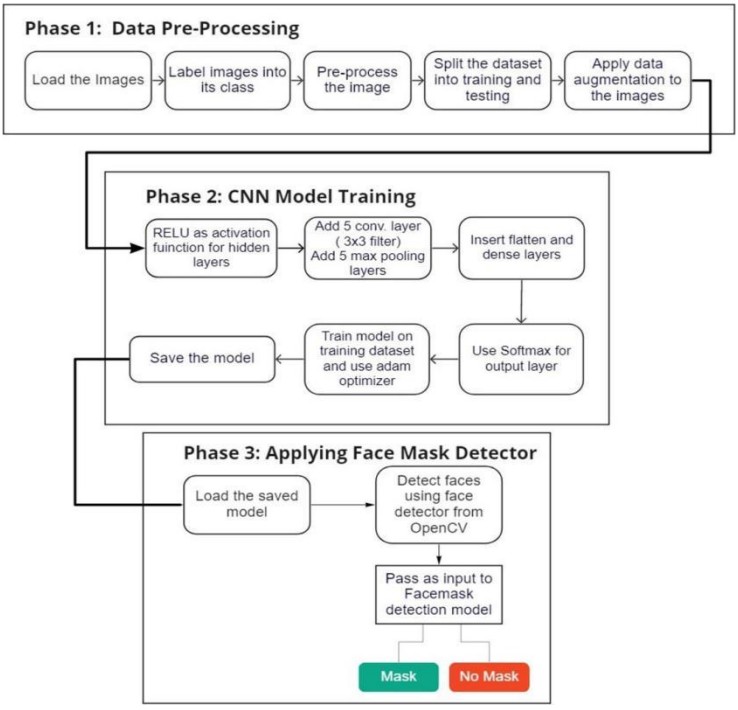
Finally, the dataset was divided into two classes or labels. These were **'with_mask'** and **'without_mask'**, and together, the images were curated, aggregating to around **4000 images.**The data set can be downloaded from Github.

Proposed methodology

In order to predict whether a person has put on a mask, the model requires learning from a well-curated dataset, as discussed later in this section. The model uses Convolution Neural Network layers (CNN) as its backbone architecture to create different layers. Along with this, libraries such as OpenCV and Keras are also used. The proposed model is designed in three phases: Data pre-processing, CNN model training and Applying face mask detector as described in Fig. 2.

1.1 Data pre-processing

The accuracy of a model is dependent on the quality of the data-set. The initial data cleaning is done to eliminate the faulty pictures discovered in the data-set. The images are resized into a fixed size of **96 x 96**, which helps to reduce the load on the machine while training and to provide optimum results. The images are then labeled as being with or without masks. The array of images are then transformed to a NumPy array for quicker computation. Along with that, the *preprocess input* function from the MobileNetV2 is also used. Following that, the data augmentation technique is utilized to increase the quantity of training data-set and also improve it's quality. A function *ImageDataGenerator* is used with appropriate values of rotation, zoom, horizontal or vertical flip, **to generate numerous versions of the same picture.** The training samples has been increased to elude over-fitting. It enhances generalization and robustness of the trained model. The whole dataset is then divided into training data and test data in a ratio of 8:2 by randomly selecting images from the dataset. The *stratify* parameter is used to keep the same proportion of data as in the original dataset in both the training and testing datasets.



1.2 CNN model training

The model architecture adopted for the research is described in Table 1. The main components of the architecture are 2D convolutional layers (conv2D), pooling layer, activationfunctions and fully-connected layers. The proposed model comprises of a total of *5 Conv2D*layers with padding 'same' and stride of 1.

Pooling layers decrease the size of the feature map. Thus, the number of trainable parameters is reduced, resulting in rapid calculations without losing essential features. Two major kinds of pooling operations can be carried out: max pooling and average pooling. Max pooling implies making the most significant value present in the specific location where the kernel resides. On the other hand, average pooling computes the mean of every value in thatregion.

Activation functions are the nodes that are placed at the end or among neuronal networks (layers). They decide whether or not the neuron fires. Choice of activation function at hidden layers as well as at output layer is important as it controls the quality of model learning. The ReLU activation function is primarily used for hidden layers; whereas, Softmax is used for the output layer and calculates probability distribution from a real number vector. The latter is the preferred choice for multi-class classification problems. Regarding ReLU, it offers better performance and widespread depth learning compared to the function of sigmoid and tanh

After all Convolutional layers have been implemented, the FC layers are applied. These layers help to classify pictures in both the multi class and binary categories. In these layers, the softmax activation function is the choice of preference to produce probabilistic results.

$$ReLU : f(x) = max(0, x) \tag{2}$$

$$Softmax : (x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{3}$$

Table I Model Summary

Layer(type)	Output Shape	Structure
Conv2D	96 x 96	Filters = 16, Filter Size = 3 x 3, Stride = 1
MaxPooling	48 x 48	Filters = 16, Filter Size = 2 x 2, Stride = 1
Conv2D	48 x 48	Filters = 32, Filter Size = 3 x 3, Stride = 1
MaxPooling	24 x 24	Filters = 32, Filter Size = 2 x 2, Stride = 1
Conv2D	24 x 24	Filters = 64, Filter Size = 3 x 3, Stride = 1
MaxPooling	12 x 12	Filters = 64, Filter Size = 2 x 2, Stride = 1
Conv2D	12 x 12	Filters = 128, Filter Size = 3 x 3, Stride = 1
MaxPooling	6 x 6	Filters = 128, Filter Size = 2 x 2, Stride = 1
Conv2D	6 x 6	Filters = 256, Filter Size = 3 x 3, Stride = 1
MaxPooling	3 x 3	Filters = 256, Filter Size = 2 x 2, Stride = 1
Classification Layer	–	Fully-connected, Softmax
Total params: 2,818,658		
Trainable params: 2,818,658		
Non-trainable params: 0		

1.2.1 Training of face mask detection model

The classification of a supervised learning CNN model is done after its training to classify the trained images to their respective classes by learning important visual patterns. Tensor-Flow and Keras are the primary building blocks for the proposed model. In this study, 80% of the dataset contributes to the training set and the rest to the testing set.The input image is pre-processed and augmented using the steps described above. There is a total of *5 Conv2D* layers with ReLu activation functions with a 3 x 3 filter and 5 Max-Pooling Layers with a filter size of 2 x 2. Flatten and Dense are used as the fully connected layers. The output layer uses softmax as its activation function. This results in 2,818,658 trainable parameters in this Convolutional Neural Network (see Table 1). It calculates them on different parameters such as Accuracy, Loss, Validation Accuracy, and Validation Loss. From Table 1 and Table 2. Binary crossentropy is used for calculation of the classification loss for the model. For a classification problem, it yields a value between 0 and 1 (probability value).

1.3 Applying face detection model

1.3.1 Face detection using open-CV

The face detection model used is pre-trained on 300x300 image size with 140,000 iterations. The DNN face detector uses a ResNet-10 architecture built upon the Single Shot Detector (SSD) framework as a base model. A Single Shot Detector applies a single shot to identify numerous objects in a picture using a multi-box. Therefore, it has a much quicker and more accurate object detecting system.

Another deep learning framework is Caffe, which is created and faster, more powerful, and more effective alternative to existing object detection techniques.To use the model, Caffe model files are required that can be downloaded from the OpenCV GitHub repository. The *deploy.prototxt* file describes the network architecture and *res10 300x300 ssd iter 140000.caffemodel* is the model which has weights ofthe layers. The *cv2.dnn.readNet* function takes two parameters ("path/to/prototxtfile", "path/to/caffemodelweights"). We receive the number of identified faces after using the face detection model, which is then provided as input to the face mask detection model. Faces can be detected using this face detection model in both static pictures and real-time video streams. Overall, this model is rapid and accurate and has minimal resource usage.

1.3.2 Application of face mask model

Now that the model is trained, it can be implemented for any image to detect the presence of mask. The given image is first fed to the face detection model to detect all faces within the image. Then, these faces are passed as an input to the CNN-based face mask detection model. The model would extract hidden patterns/features from the image and thus classify the images as either "Mask" or "No Mask". Figure 5 describes the complete procedure.

Table 3 Training Model onAdam Optimizer

Epochs	loss	accuracy	val_loss	val_accuracy
10	0.2938	0.8612	0.2520	0.9050
20	0.2172	0.9125	0.2506	0.9100
30	0.1524	0.9375	0.1783	0.9300
40	0.1688	0.9350	0.2488	0.9300
50	0.1223	0.9600	0.2259	0.9450
60	0.1204	0.9575	0.3382	0.9150
70	0.0760	0.9800	0.2224	0.9400
80	0.0478	0.9825	0.2671	0.9450
90	0.0541	0.9825	0.2407	0.9500
100	0.0288	0.9937	0.2102	0.9500

Table 4 Hyper Parameters used in Training

Parameter	Detail
Learning Rate	0.0005
Epochs	100
Batch Size	32
Optimizer	Adam
Loss Function	binary crossentropy

2 Results and analysis

The final results were achieved after multiple experiments using various hyper-parameter values such as learning-rate, epoch size, and batch size. Table 4 depictsthe hyper-parameters used.

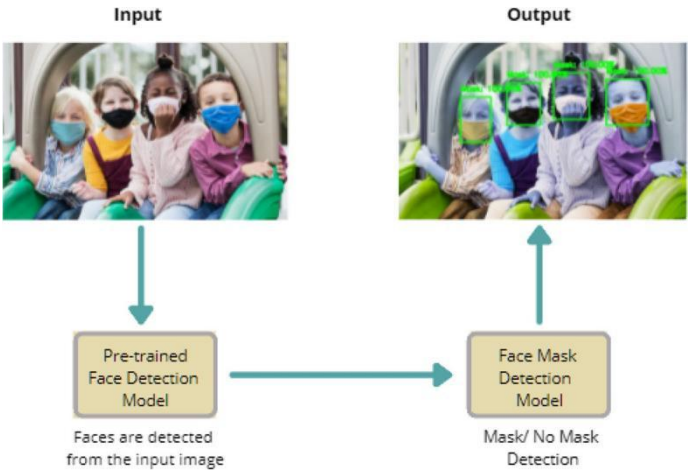


Fig. 5 Diagram showing implemented Face Mask Model

The accuracy of the masked individual identified by the developed model provides a good standard of prediction. In Fig. 6 above, which shows training accuracy, the model hadstruggled to acquire features until it reached 40 epochs, The accuracy was around 98% after 100 epochs. Loss in the training dataset, which is smaller than 0.1, and the loss in the validation dataset which is also less than 0.2 after 100 epochs.

Model Testing

The model was tested on various diverse images, and some of them are exhibited below in the Fig. 10. The green rectangular box demonstrates a person correctly wearing the mask along with the accuracy score at the top, whereas the rectangular red box displays that the individual is without any mask. In summary, the model learns from the training dataset in order to label and then predict.

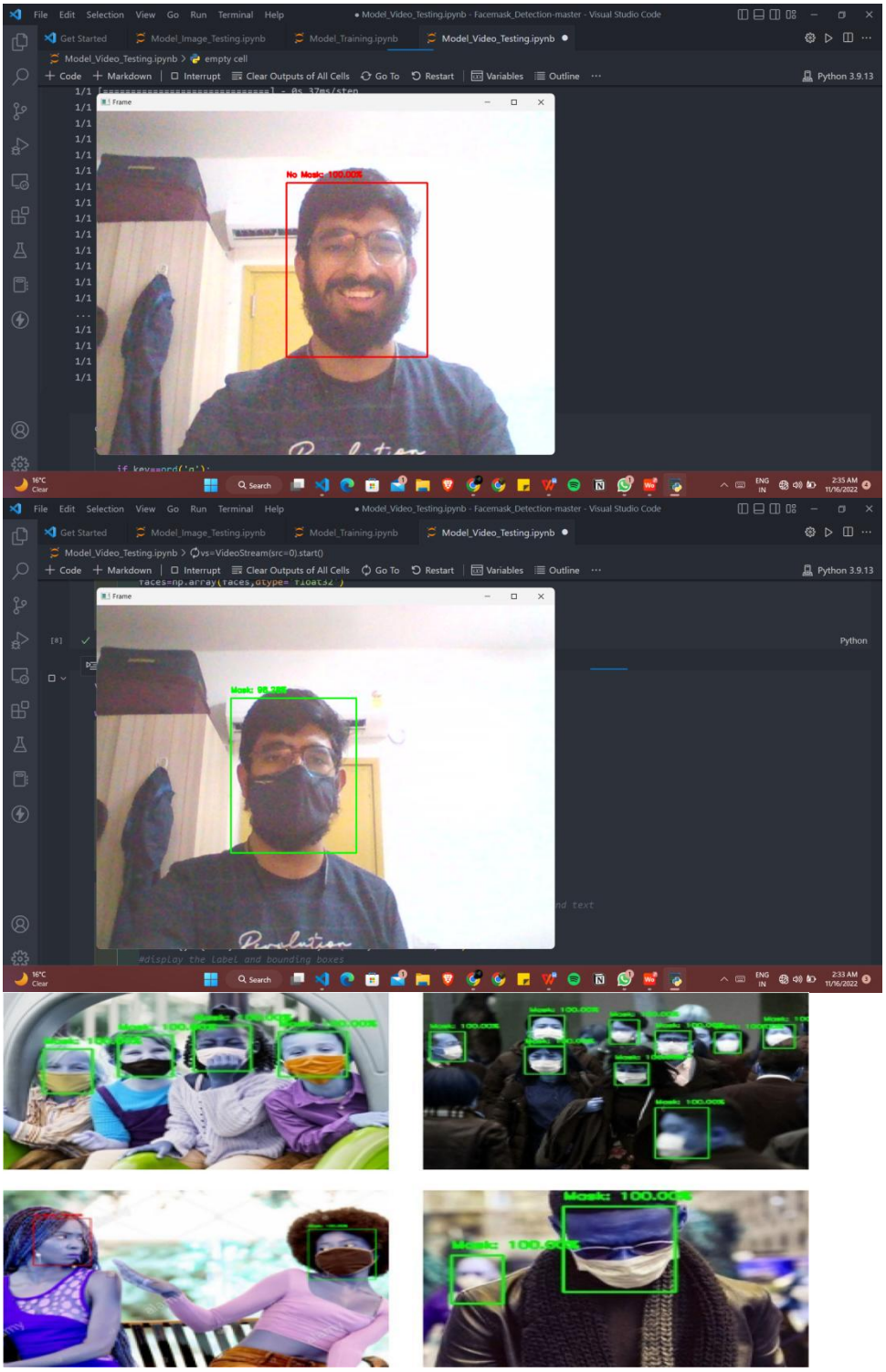


Fig. 10 Predictions on Test Image