Deep Learning Assisted Tool for Face Mask Detection

Karthik Prasad[1], Madhava Raj B[2], Gokulnath S[3], Sanjan Govind K[4]

[1,2,3,4]*Student, Electronics and Communication Engineering, Vivekananda College of Engineering and Technology, Puttur, Karnataka, India*

---***---

Abstract - *COVID-19 pandemic has already caused the huge havoc around the world with its recurring multiple waves. Many have lost their loved ones. Great number of people lost their jobs. World economy is still struggling to recover the losses caused by the pandemic. The pandemic has made a significant riffles in the lives of people around the globe. The World Health Organisation(WHO) has issued the several preventive measures to avoid the transmission of virus like measure masks in public gathering, maintain social distance, vaccination etc [3]. Checking whether the public is following the safety guidelines is a tedious job for the authorities concerned. Our team has worked over a model, which can easily identify whether a person is wearing mask or not in public gatherings like cinema theatres, malls, govt offices etc. where the CCTV is functional. The system can prevent the transmission of virus. This model has been developed using the various machine learning frameworks like Tensorflow and Keras. We have used MobileNetV2 architecture in this model.*

Keywords__ Tensorflow, Covid-19, MobileNetV2, Deep Learning, Face Mask, CNN

1. INTRODUCTION

Corona virus(COVID-19) is an infectious illness caused by the SARS-CoV-2 virus. The first case was initially discovered in Wuhan(China) by the end of 2019, later spreading throughout the global became pandemic. As of now (3rd January 2022) 5.4 million people have succumbed to Covid-19 [1]. Currently the world is in the danger of another wave of virus. In spite of high vaccination rates, new cases are in surge across Europe. WHO has recently warned, if strict restrictions were not imposed as soon as possible, may cause 0.5 millions deaths by the end of February [2]. It also issued several guidelines to check the faster spreading of the virus such as wearing face mask, maintaining social distance, regular health checkups, vaccination etc [3]. In this project we have worked over a system, which can easily encounter whether the person is wearing face mask or not in the public places like cinema halls , shops etc where the CCTV is functional. The system can help to prevent the rapid transmission of the viruses. Face recognition is one of the most promising field of computer vision. Recognizing the face and verifying whether the person is wearing a face mask or not automatically is the key principle of our project.

2. LITERATURE SURVEY

There are methods in the literature that are applied on the face mask detection model. [4] combining image super-resolution and classification network (SRCNet) developed a method for identification face mask�wearing condition ,which determined three categories of classification problem on the basis of unconstrained 2-Dimensional facial images. [5] this paper proposes a different approach for face recognition with deep neural network. In this method, extracted features of face are given as input to the deep neural network instead of raw pixel data. This helps in better performance on Yale Face dataset by reducing the complexity and improved accuracy. [6] in this paper, using deep learning and image processing techniques , proposes a method to detect human faces and classify them. This model uses MobileNetV2 [7] architecture and for optimization ADAM [8] optimizer is used. Proposed model is discussed in section-III, Methodology in section-IV, Results in section-V and finally paper is concluded in section-VI.

3. PROPOSED MODEL

The primary focus of the proposed model is to identify whether a person is wearing a mask or not, based on the visuals of image/video stream.

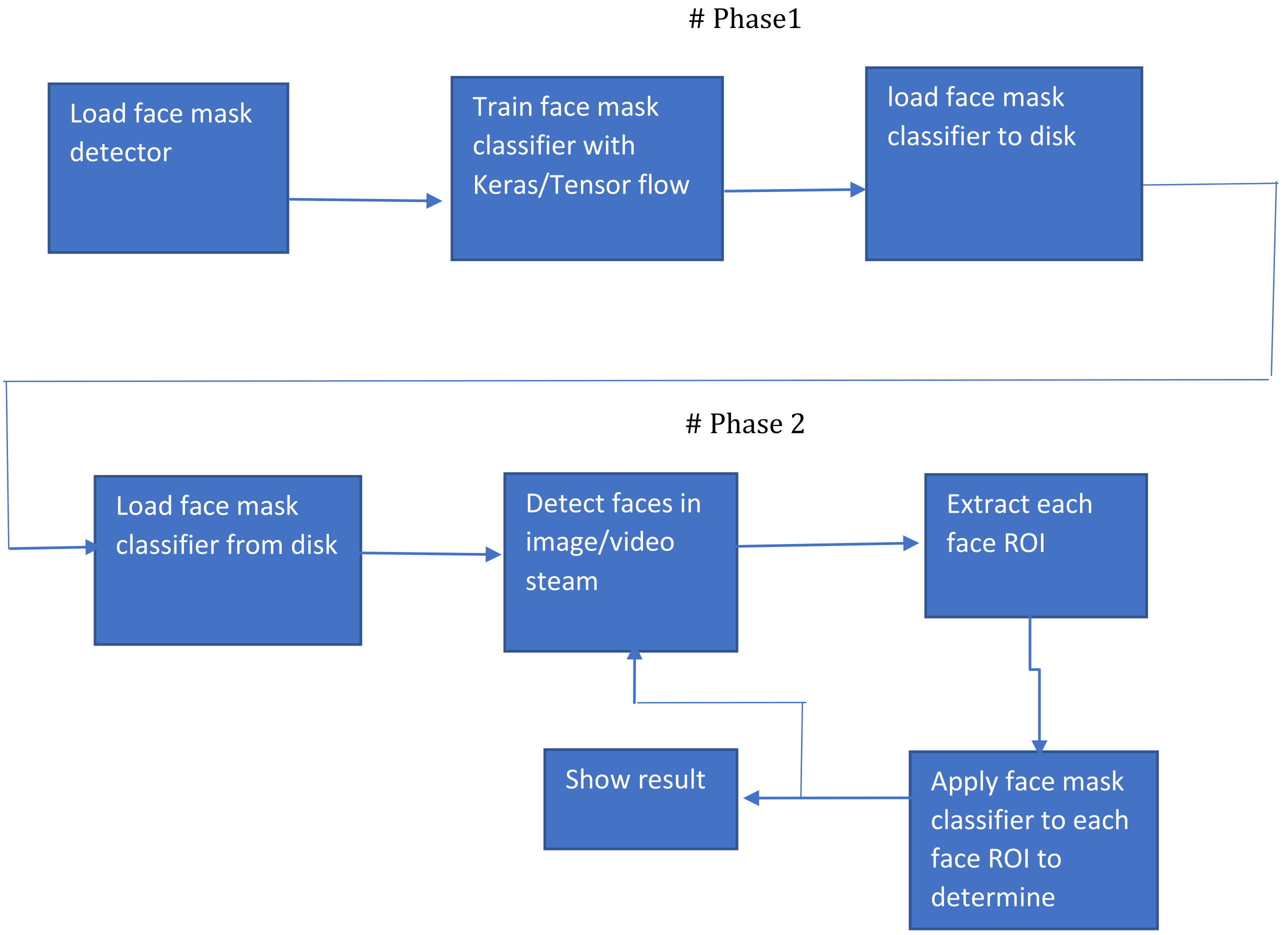
# Phase1
Load face mask detector
Train face mask classifier with Keras/Tensor flow
load face mask classifier to disk
# Phase 2
Load face mask classifier from disk
Detect faces in image/video steam
Extract each face ROI
Apply face mask classifier to each face ROI to determine
Show result

Fig. 1. Block diagram of the proposed model

In first phase, we will gather the with and without the mask images. This dataset consists of 1800+ images belonging either of the two classes. In second phase, we will train the classifier for face mask detection, and in the final phase, we will use the trained model to distinguish each face as with mask or without a mask.

In the second phase, we load face mask classifier from the disk. We have to add dataset to detect the face. The we have to turn On the camera to detect the face whether they are wearing the mask or not.

Apply trained face mask detector model to each face ROI(Region of Interest) to determine “mask” or “no mask” after this we get the result whether the person is wearing the mask or not.

To detect whether a person is wearing or not, TensorFlow and Keras frameworks are used. Here in this current proposed system, initially training of the model is done using the datasets which are collected from Kaggle. For training purpose TensorFlow and Keras are used together. Once the training of the model is complete we load the trained model, here by real time videostream faces are classified accordingly. In this proposed system we used MobileNetV2 architecture since it is very much helpful in training the model with huge dataset and maintaining the quality of images which we used to train. After loading of images, since we cannot directly trained model using raw images, they are converted into arrays,. After that using MobileNet images are processed and appended into the datalist. The most significant part of this current system are face recognition and mask detection. MobileNetV2 and OpenCV are explicitly used for this purpose. Once the ROI i.e. face has been detected by the system, a red or green rectangular box will be placed surrounding the face, depending on face with or without mask. Around 1800+ images of people with and without masks are collected from Kaggle for this purpose. By using these images we train model which will classifies in 2 categories namely, “with mask” and “without mask”. When a person appears in front of the camera, initially the system will recognize the face and later it will detect whether a person is wearing mask or not. It will also display the correctness of the weared mask with rectangular box around the face. The monitoring will done continuously with the help of videostream and real time data of analysis can be obtained. If the person is found without mask, his picture will immediately sent to the concerned authority so that they can further action. Since world is in the verge of another wave of corona virus, the proposed model can be considered dire necessity.

4. METHODOLOGY

Methodology of proposed model will be discussed in detail in this section.

Dataset: Our proposed model uses dataset containing cropped images of face with variable rotation and different orientation. This makes our data more diverse. Images were collected from images dataset offered by Kaggle, a open-source online community of data scientists and machine learning practitioners and few within the public domain. The dataset folder contains two subfolders namely With mask and Without mask each containing the images of respective types. Each sub folder contains 3600+ images. Accordingly they are labelled and trained in our model. We use MobileNet and OpenCV for automated real time mask detection.

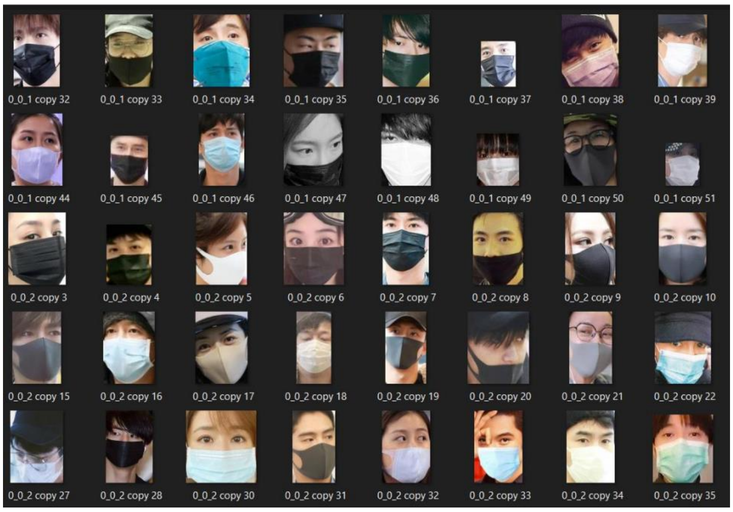

Fig. 2. Image dataset (with mask)

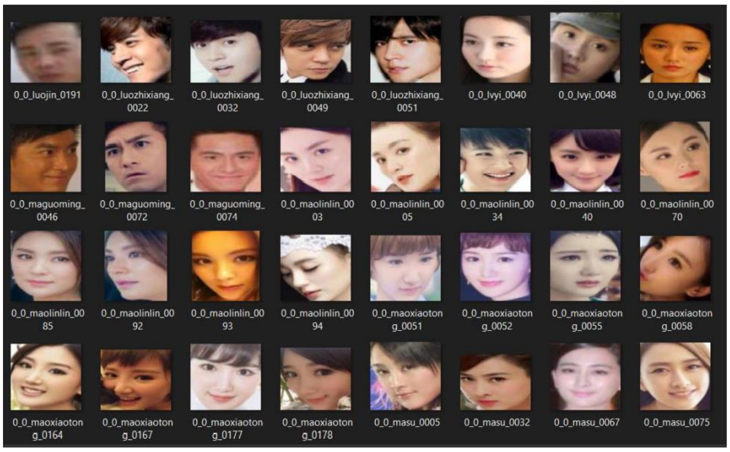

Fig. 3. Image dataset (without mask)

4.1 Steps involved in face mask detection:

STEP 1: Pre-processing of Data:

a)Data Pre-processing: In Data preprocessing step, data will be processed into desired format. It involves transformation of data from a given format to as required format by the model. This processed data well fit into the model and will be compatible with different entities. With the help of Numpy and OpenCV libraries the proposed model perfoms operations on image and video.

b) Data Visualization: It can be defined as the process of transforming abstract datas to easily interpretable representations. It gives the user real-time trends, new insights about the data. It is useful in determining pattern in a given dataset. In the proposed model, we first initialize two empty lists namely 'data' and 'labels' which store the array representations of images and its corresponding label i.e. whether the image is masked or non-masked image. Two sub-folders in the dataset folder is iterated by a for loop, each sub-folder at a time. Initially the images are labelled as same as that of the name of the folder they are present. After labelling the dataset through for loop, the 'labels' list in the form of ['with mask', 'without mask',….] matching the images. Since we cannot train the model with categorical data, it needs to be encoded as 0's and 1's. We perform One-hot encoding on the 'labels' in order to transform it into binary data using the command below.

c) RGB to Grayscale conversion : RGB images contains many features those were may not be useful for the training algorithm and they take long time to process. In order to extract the features that were unique, we need to the image from RGB to grayscale. It will reduced the complexity of the code, difficulty of visualization. It will increase the speed of training the model.

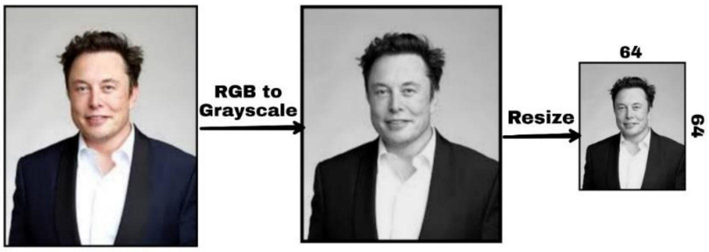

Fig. 4. RGB to Gray Scale conversion of a image of size 100x100

d) Image Reshaping: Majority of the CNNs requires their iputs to be fined-tuned. Size of input images must be identical to the the 3D-feature tensor. It is required to reconfigure the inputs before we provided them as inputs to the model. They are compressed to the pixel range [0, 1]. After that they are transformed to 4D-arrays. Last NN will give two possible outputs i.e. where the input image is masked or unmasked in terms of binary values.

STEP 2: Splitting the Dataset

Here we split the dataset into training set and testing test. In the training set, there will be images that we used to train the model and using the testing set we evaluate our CNN model. In out model we set test size = 0.2, which will split 80% and 20% data to training set and testing set respectively.

STEP 3: Model - Building and Training

In our proposed model, we load the pre-processed dataset and train the model using algorithm on labelled dataset. Here, loaded images are resized and are transformed into numpy based array. In our proposed model, the dataset is loaded and for the purpose of training labelled images are used. At this stage, images from the dataset are resized and convered into numpy based array. In this model, we use MobileNet to construct the lightweight deep CNNs. Using TensorFlow model is trained. MobileNet architecture is used here which can be considered as the mainstay and tensorflow framework is used to train the model. We keep initial learning rate ,INIT_LR = 1e-4, EPOCHS = 20, batch size BS = 32. In this model, for the purpose of face detection webcam is used. If the model recognizes the face, it will place red square box and accuracy around the face.

a) Building the model: CNN is playing a significant role in modern day computer vision applications. We used Sequential CNN in our model. Sequential CNN is used in the current method. Output of the first convolution layer is fed into the activation function called Rectified Linear Unit (ReLU) [9] followed by the MaxPooling layer. The inputs that are fed into first convolution layers is convolved with 180 filters of kernel size 3 X 3. Along with first we need to provide the information of shape of the expected input because the model must have the knowledge of shape of the input will be receiving. Then we use "relu" activation functin for Conv2D class. For the Conv2D layer, ReLU activation function is used. This optimization parameter has all the properties of linear models that can easily be optimized with different gradient-descent types. It performs well compared to other types of activation functions. Then MaxPooling layer is used to decrease the dimensions of result from the preceding operation. the output of previous operation is reduced by MaxPooling layer. Here MaxPooling layer of Pool size 7 x 7 is used. The resulting output of pooling layer is again convolved with Convolution layer with filter size of 128 and kernel size of 3 X 3. The resulting output is again fed into ReLU activation function followed by MaxPooling layer. This resulting matrix of features is flattened using Flatten layer and this flattened matrix features in the form vector is fed as input parameters to the fully connected neural network. In order to reduced the overfitting a Dropout layer is used. In the final dense layer, softmax activation function is used which classifies the face in the input image as 'with' mask or 'without' mask.

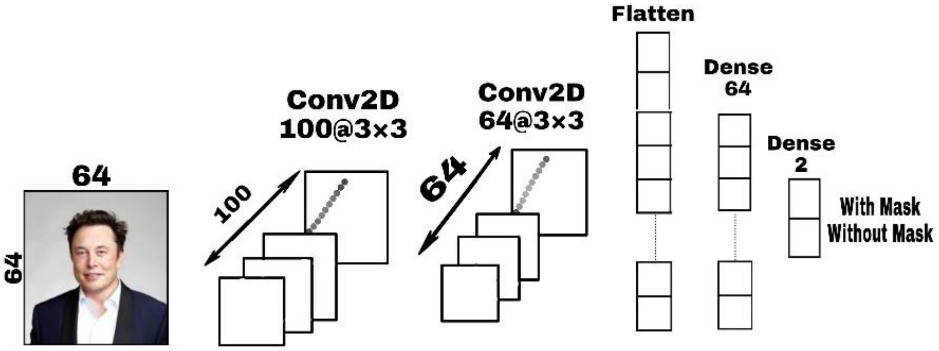

Fig. 5. Convolutional Neural Network architecture

b)Training the model: On according to the compilation method, the learning methodology needs to be configured according to the requirement. Optimizer used is "adam", which uses Adam algorithm, a stochastic gradient descent method.. Adam is considered to be the best among adaptive optimizers. Categorical crossentropy is used as loss function here. We set the metrics to "accuracy" since problem is a classification problem. For better accuracy MobileNetV2 can be used.

STEP 4: Labeling the Information:

Once the model is built, results will be labelled with binary labels as per the binary probabilities, i.e. ["0" for "without mask", "1" for "with mask"]. Using RGB values color of the rectangle that bounds also be changed. ["GREEN" for "with mask" , "RED" for "without mask"].

STEP 5: Importing the trained model:

By using our Laptop's webcam, real-time face mask is detected. Prior to that, we have to implement the dace detection model in our program, used for real-time input.

STEP 6: Detection of Faces with and without Mask:

In this final step, using the CV library, webcam will open for infinite loop till we manually exit the loop. Here facial recognition will done by cascade classifier. The code webcam=cv2.videocapture(0) is used to turn on the primary camera of the PC. We can use multiple cameras by varying the argument to the above code. The model will estimate the probability of possibility of each class. Class with higher probability will be choosen and respective tags will be placed around the face.

5. RESULTS AND DISCUSSIONS

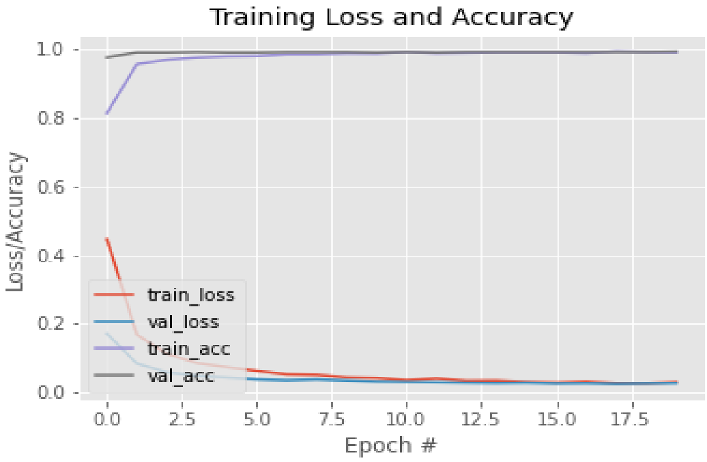

Fig. 6. Training Model accuracy and loss curves

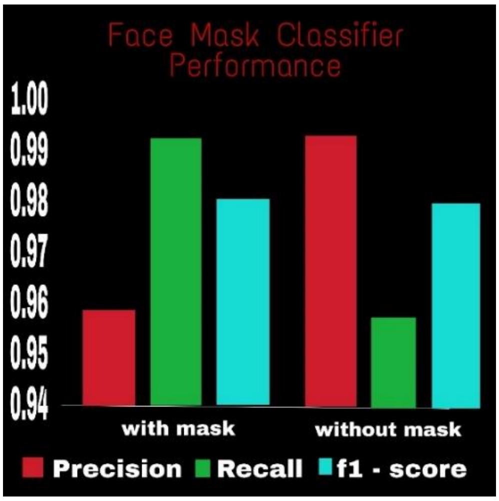

Fig. 7. Performance Metrics Histogram graph

By integrating every component of our model, we get the accurate mask detection system. MobileNetV2 classifier is utilized in the current model. The developed model is capable of detecting the facemask in videostream even with many faces and different orientation.

5.1 Face mask detect from real time Videostream:

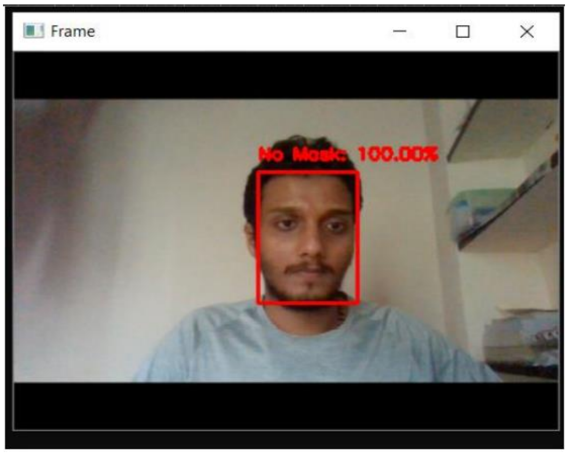

Fig. 8. Output when no mask

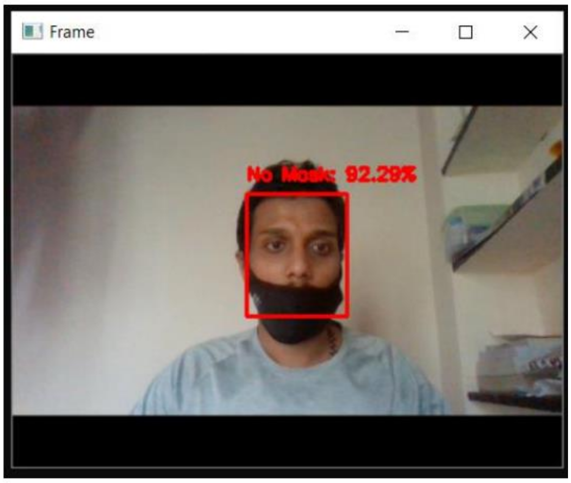

Fig. 9. Output when mask weared improperly

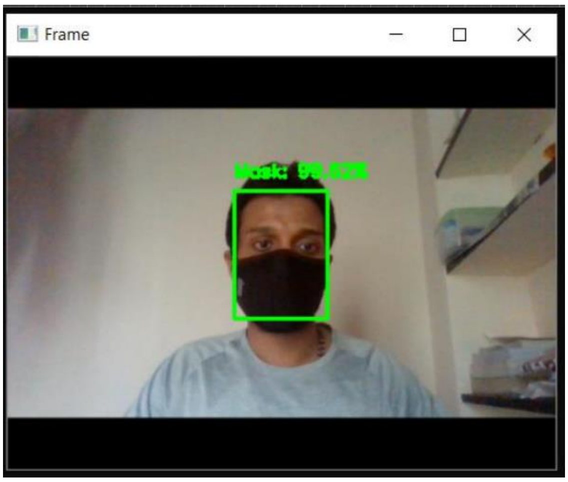

Fig. 10. Output when Mask weared properly

Table 1. Comparison of current model with other models

Author	Method	Classifier	Result (Accurecy)
1. Ejaz et al. [10]	Viola-Jones algorithm Principle component ananlysis(PCA)	Nearest neighbor (NN) classifier	Masked face 72% Non-masked face 95%.
2. Arjya Das et al. [11]	Sequential Convolutional Neural Network	cascade classifier and CNN	95.77%
3. Park et al [12]	Recursive PCA	PCA	90%
4. Nieto-Rodr´ıguez et al [13]	Viola-Jones face detector	Cascade classifier	Recal>95% False positive<5

Table 2. Performance Metrics of current Face Mask Classifier:

	precision	recall	f1-score	support
With_mask	0.99	1.00	0.99	383
Without_mask	1.00	0.99	0.99	384
accuracy			0.99	767
Macro avg	0.99	0.99	0.99	767
Weighted avg	0.99	0.99	0.99	767

6. CONCLUSION

The current model automatically detects whether a person appearing in front of the camera is wearing face mask or not without human intervention. Results of process can be used by the concerned authorities to take legal action against the culprits. This proposed model uses Computer Vision and machine learning algorithms to ensure public safety against COVID-19 virus. The proposed face mask detection model is trained using MobileNetV2 architecture with the help of Python libraries such as OpenCV, Tensor Flow, Keras. The proposed model can be implemented in various places like commercial offices, schools-colleges, airports, railway stations, shopping malls, etc. Good accuracy has been achieved and there is greater scope for real time implementation and optimization.

REFERENCES

[1] World Health Organization (2022), WHO Coronavirus (COVID-19) Dashboard,

Available: https://covid19.who.int/ [Accessed 03.01.2022]

[2] INSIDE NE, "WHO says Europe, Central Asia could see 5 lakhs COVID deaths by Feb 2022", Available: https://www.insidene.com/who-says-europe-central-asia-could-see-5-lakhs-covid-deaths-by-feb-2022/amp/ [Accessed: 03.01.2022]

[3] WHO, "Advice for the public: Coronavirus disease (COVID-19)", Available: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public/ [Accessed: 03.01.2022].

[4] Qin, Bosheng, and Dongxiao Li. "Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19." Sensors 20, no. 18 (2020): 5236.

[5]Priya Gupta, Nidhi Saxena, Meetika Sharma, and Jagriti Tripathi. "Deep neural network for human face recognition." International Journal of Engineering and Manufacturing (IJEM) 8, no. 1 (2018): 63-71.

[6] Bhadani, Amrit & Sinha, Anurag. (2020). "A FACEMASK DETECTOR USING MACHINE LEARNING and PROCESSING TECHNIQUE" 13.

[7] Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. "Mobilenetv2: Inverted residuals and linear bottlenecks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510-4520. 2018.

[8] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

[9] Nair, Vinod, and Geoffrey E. Hinton. "Rectified linear units improve restricted boltzmann machines." In Icml. 2010.

[10] M.S. Ejaz, M.R. Islam, M. Sifatullah, A. Sarker , "Implementation of principal component analysis on masked and non-masked face recognition", 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT) (2019).

[11] A. Das, M. Wasif Ansari and R. Basak, "Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV," 2020 IEEE 17th India Council International Conference (INDICON), 2020, pp. 1-5, doi:10.1109/INDICON49873.2020.9342585.

[12] Jeong-Seon Park, You Hwa Oh, Sang Chul Ahn and Seong-Whan Lee, "Glasses removal from facial image using recursive error compensation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 5, pp. 805-811, May 2005, doi: 10.1109/TPAMI.2005.103.

[13] Nieto-Rodríguez, Adrian, Manuel Mucientes and Víctor Manuel Brea. "System for Medical Mask Detection in the Operating Room Through Facial Attributes." *IbPRIA* (2015).