Face Mask Detection Using Machine Learning

Varshitha Gudibandi, Amey Gawankar, *Electronics and Communications Engineering, B.Tech, VIT Vellore*

***Abstract*— COVID-19, the virus that's spreading faster beyond imagination and crores of people are infected with this virus. The COVID-19 virus can be spread through contact with infected persons and also from public surfaces. In this case, the traditional biometric frameworks which depend on passwords or fingerprints are not any longer safe as many people touch them. Face acknowledgment are more secure with no compelling reason to contact any gadget. Researches have proved that wearing a mask to cover nose and mouth and social distancing will diminish extensively the transmission of the infection. Taking this into thought, we wanted to build up a code which identifies if the individual is wearing a cover or no with no association with the gadget. We achieved it using Machine Learning. To do so we utilized Open CV to access webcam, TensorFlow with Keras library and MobileNet to build and train the model.**

***Index Terms*—COVID-19, Machine Learning, OpenCV, Face Mask, MobileNet**

I. Introduction

IN recent days, the situation report of world health organization (WHO) presented that coronavirus disease 2019 (COVID-19) has globally infected over 38,394,169 people including over 1,089,047 deaths, as reported on 15th October 2020. In addition to that there are many severe respiratory diseases COPD, Pneumonia, Pulmonary enema, Pulmonary fibrosis, Asthma, Pneumothorax, Pulmonary embolism, Pulmonary arterial hypertension, severe acute respiratory syndrome (SARS) etc, have affected a large number of people in the past few years and also about 3 million die from it each year, making it the leading cause of death and disability all over the world. Therefore, more and more people are concerned about their health, and public health is considered as the top priority for governments. Effectively, the WHO (World Health Organization) has shown that the surgical face masks (N95 as well as Fabric masks) could reduce the spread of coronavirus. At the moment, WHO recommends that people should wear face masks if they are showing signs of respiratory symptoms, or they are taking care of the people with the respiratory symptoms.

Also adding to this point, the main reason that we are making this paper is that, many public product and service providers insist customers to use the services only if they wear masks. Therefore, the detection of face mask on an individual is a crucial task. Hence, we have developed a model.

Face mask detection is to detect whether a person/individual is wearing a mask or not and what is the orientation/location of the mask on the face in real-time atmosphere.

II. Related Work

Since Covid-19 pandemic is developed in recent times, not many people have published about it. Considering the few that were published as well as few papers that are related to the topics, we could summarise the following.

RetinaMask: A Face Mask Detector (Mingjie Jiang, Xinqi Fan, Hong Yan) [1] is a mask detection system which is based on object detection for face detection mechanism and the CNN to filter the face with mask and face without mask images. The dataset used consists some faces are masked by hands or other objects rather than physical masks, which brings advantages to the dataset to improves variants of Face Mask Dataset. Some samples are shown in including faces with masks, faces without masks, faces with and without masks in one image and confusing images without masks. The accuracy of the whole setup for identification without mask is 81.3% and the accuracy for with mask is 82.8%.

Identifying Facemask-wearing Condition Using Image SuperResolution with Classification Network to Prevent COVID-19 (Bosheng Qin, Dongxiao Li) [2] proposes a system will detect if there is a face mask present of the person's face, partially present or not present at all. The development of facemask-wearing condition identification network is challenging for several reasons. The limitation in dataset is one main challenge. The facemask-wearing condition dataset is generally small, and the image quality is not well enough compared with general face recognition dataset. It does not show the most accurate output when different colored masks are used. Besides, the various performances of wearing facemask incorrectly largely increase the difficulty of identification. To overcome these challenges, the SRCNet was introduced, which utilized SR network and transfer learning before classification. The SR network solved low quality image problem, and transfer learning solved the challenge of small dataset with various wearing facemask incorrectly, which get performance improved considerably. Facemask-wearing condition was detected with 98.70% accuracy, which indicated that SRCNet has great potential to support the automatic facemask-wearing condition identification.

Efficient Masked Face Recognition Method during the COVID-19 Pandemic (Walid Hariri) [3] There are two different tasks namely: face mask recognition and masked face recognition.The first one checks whether the person is wearing a mask or no. This can be applied in public places where the mask is compulsory. Masked face recognition, in the other hand, aims to recognize a face with mask basing on the eyes and the forehead regions. In this paper we handle the second task using deep learning based method.

The images of this dataset are already cropped around the face. Therefore, to apply it to real time we need to include image pre-processing which involves cropping the face from the background.

A SNoW-Based Face Detector (Ming-Hsuan Yang Dan Roth Narendra Ahuja) [4] **The** SNoW learning architecture is a sparse network of linear functions over a pre-defined or incrementally learned feature space and is specifically tailored for learning in the presence of a very large number of features. A wide range of face images in different poses, with different expressions and under different lighting conditions are used as a training set to capture the variations of human faces. The dataset with images which are rotated and with different intensities are used, but still it is claimed that the presented method is still not able to detect rotated faces. The dataset doesn't contain images where the parts of face is covered which becomes a limitation.

Face Detection Based on Facial Features(Lihong Zhao; Xiaolin Sun; Xinhe Xu) [5] provides algorithm for face recognition of colour pictures. Eye-mouth triangle for all possible combinations of the two eye candidates and one mouth candidate is formed. A face score is computed for each verified eye-mouth triangle based on it's eyes/mouth maps and the triangle with highest score that exceeds a threshold is retained.For constructing the triangle 2 eyes and a mouth is considered. The drawback is that a face with one eye covered (still a face) will not be detected and the triangle is not formed. In the same way if a face with the mouth covered (shill a face) will not be detected. This limits the detection of face in all cases possible. Similar limitation in paper [6][7]

III. PROPOSED WORK

A. Dataset Creation

The better the dataset the more accuracy is obtained. The dataset we choose decides the extent to which our model is trained. The reason of creation of a new dataset is to include the same images with different angles and creating with mask dataset from without mask dataset which will increase the accuracy. In this step, images of mask will be placed on the faces without mask and create a new set of images of faces with mask. There are some key features on a person's face that is recognized by the software and these features like the nose bridge and the Chin is recognized after the face detection & the mask image is placed on the nose and the chin area and is rotated according to the face. In this process, it also rotates the image in possible angles and creates images to increase the detection accuracy so that the model detects even when the image is not in perfect angle.

B. Training the Model

This step involves two processes - data pre-processing and training the model. In data pre-processing, we create arrays of images categorized as with mask and without mask to build the model as for training the model we can give only arrays. These images are joined with their respective paths which are also stored as an array. In training the model, we use MobileNetV2 to perform neural networks as it is faster and simpler convolutional neural network. While training, the training loss and accuracy for each epoch we define is calculated and plotted to know how well and better the model is being trained. The given epoch is 20 and learning rate is 1e-4(lesser the learning rate more the accuracy).
The output after giving it to MobileNet is saved as basemodel which is used to create our headmodel. The headmodel undergoes maxpooling and and fully connected layer is formed.The final output is mask and no mask.

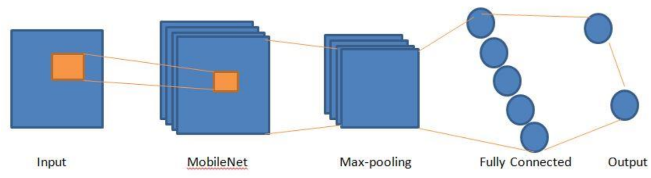

Fig. 1.1 Convolution Neural Network Architechture using MobileNet

C. Real Time Detection using Webcam

To do so, we used Open CV to access the webcam. It captures every frame and every frame is considered as an image. This frame is converted to gray scale image to make it possible for detection of all coloured masks. After capturing the image we need to detect the face and then the mask. We used Facenet and Masknet to do so respectively. We load the trained model to it. Facenet detects the face and a box is displayed around the detected face. Masknet detects if there is a mask or no and is displayed above the box as "MASK" and "NO MASK" along with accuracy. Masknet detects the objects according to the model we give to it. Here, our given model is to detect the face mask. Therefore, it detects the mask The box for masked face appears in GREEN and no mask face appears in RED.

Fig. 1.2 Facenet Architecture

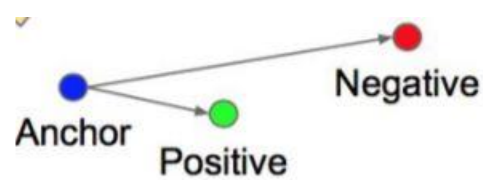

Fig. 1.3 Triplet

FaceNet learns a mapping from face images to a compact Euclidean Space where distances directly correspond to a measure of face similarity as shown in Fig 1.2. Batch is the target image that we get from webcam. Training is done using triplets: one image of a face ('anchor'), another image of that same face ('positive exemplar'), and an image of a different face ('negative exemplar').This is the basic architecture as shown in Fig 1.3.

IV. IMPLEMENTATION OF PROPOSED SYSTEM

We have used Jupyter Notebook to write and run the codes. We have chosen this as in jupyter notebook we can divide the code and run it part by part and re run the desired cells with the code.
First step is to run the code for creating the dataset. A folder named dataset is created which consists of a subfolder named without_mask. As we run the code, a new folder named with_mask will be created as a subfolder inside dataset folder. Now the whole dataset folder is our desired dataset for

training the model.

Second step is to create an array of images along with its path which are also in the form of arrays. The images used here are the images from our dataset.

Third step is to train the model with our dataset. Then we give the input image to MobileNetV2 to perform neural network which distinguishes the face into two outputs that is mask and no mask. We chose MobileNet over others assuming it would be faster than others.

The next step is to access the webcam of the system in which we run the code and capture the frames in front of camera. We give our model to Masknet. Facenet detects the face and Masknet detects the face.

A final output that is a box around the detected face along with mask or no ask and its accuracy is displayed.

V. Results and Analysis

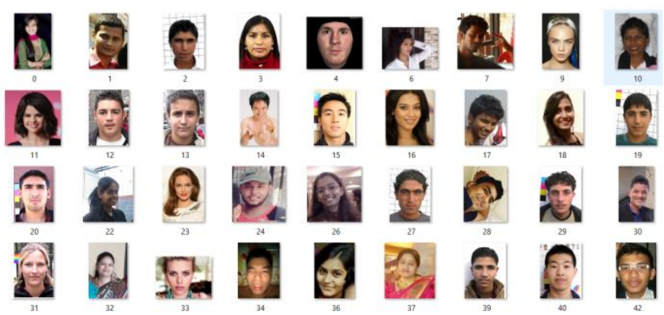

Fig. 2.1 Set of images without masks

The images shown in Fig 2.1 are the dataset that is used to create with mask images.

Fig. 2.2 images three different colour masks

The masks in Fig.2.2 are used to place it on without mask faces.

Fig. 2.3 Set of images with black masks

Fig. 2.4 Set of images with blue masks

Fig. 2.5 Set of images with white masks

Fig. 2.6 Set of images which are rotated to certain angle.

The images in Fig 2.3, Fig 2.4, Fig 2.5 are the images after the three coloured masks are placed on the without mask faces. The image in Fig 2.6 shows that the images are also rotated to certain angle and all angle images are stored. The images are also shifted right and left accordingly and stored in with_mask folder.

```
[INFO] training head...
Epoch 1/20
66/66 [==============================] - 120s 2s/step - loss: 0.2586 - accuracy: 0.8959 - val_loss: 0.0495 - val_accuracy: 0.99
25
Epoch 2/20
66/66 [==============================] - 100s 2s/step - loss: 0.0576 - accuracy: 0.9829 - val_loss: 0.0206 - val_accuracy: 0.99
81
Epoch 3/20
66/66 [==============================] - 98s 1s/step - loss: 0.0307 - accuracy: 0.9929 - val_loss: 0.0156 - val_accuracy: 0.998
1
Epoch 4/20
66/66 [==============================] - 100s 2s/step - loss: 0.0228 - accuracy: 0.9933 - val_loss: 0.0151 - val_accuracy: 0.99
81
Epoch 5/20
66/66 [==============================] - 98s 1s/step - loss: 0.0180 - accuracy: 0.9948 - val_loss: 0.0119 - val_accuracy: 0.998
1
Epoch 6/20
66/66 [==============================] - 100s 2s/step - loss: 0.0147 - accuracy: 0.9967 - val_loss: 0.0097 - val_accuracy: 0.99
81
Epoch 7/20
66/66 [==============================] - 99s 2s/step - loss: 0.0177 - accuracy: 0.9929 - val_loss: 0.0108 - val_accuracy: 0.996
3
Epoch 8/20
66/66 [==============================] - 102s 2s/step - loss: 0.0117 - accuracy: 0.9971 - val_loss: 0.0155 - val_accuracy: 0.99
44
Epoch 9/20
66/66 [==============================] - 105s 2s/step - loss: 0.0075 - accuracy: 0.9981 - val_loss: 0.0114 - val_accuracy: 0.99
63
```

Fig. 2.7 loss and accuracy of each epoch

The loss and accuracy is calculated for every epoch. There are 20 epoches with 66 as batch size.

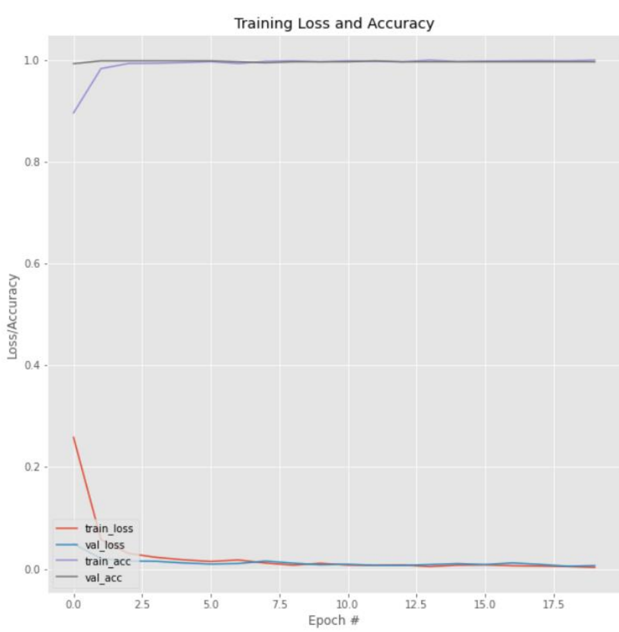

Fig. 2.8 Plot of calculated loss and accuracy.

The loss function is basically the sum of errors made for each example in training or validation sets. Here, loss has eventually decreased to almost 0 which implies the model is trained good. Accuracy is the measure of how accurate the model's prediction is compared to the true data. Here, from the plot we can say that the accuracy has increased to alomost 1 which implies the model is highly accurate.

We have tested for two different masks. A light coloured mask and a dark coloured mask to show that the colour of mask doesn't affect the detection as we have converted the frames into grayscale. All the different angles and possibilities were tested to mask sure that the model detects fask mask with a greatest accuracy possible. The following are the output images.

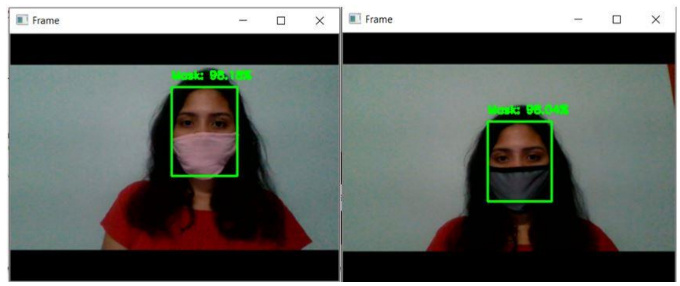

Fig. 2.9 Final detecting of face mask

In Fig 2.9 The face mask is effectively detected and a box appears around the image with a text saying Mask (which tells that the person is wearing mask) and an accuracy with which it is detected. We are successful in obtaining a accuracy of around 98%.

Model	Face		Mask	
	Precision	Recall	Precision	Recall
baseline [8]	89.6%	85.3%	91.9%	88.6%
RetinaFaceMask+MobileNet	83.0%	95.6%	82.3%	89.1%
RetinaFaceMask+ResNet	91.9%	96.3%	93.4%	94.5%

Fig. 2.10 Result obtained in paper [1] Retinamask: Face Mask Detector

We have obtained a higher accuracy than obtained in paper [2] as shown in Fig 2.10. Indeed, our accuracy is even higher than the accuracy obtained with ResNet which is another method to apply convolution.

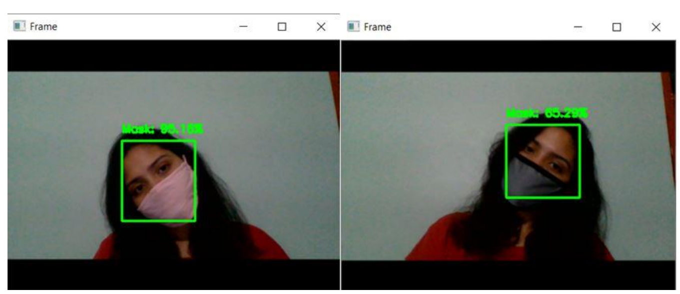

Fig. 2.11 Final detecting of face mask when face is rotated

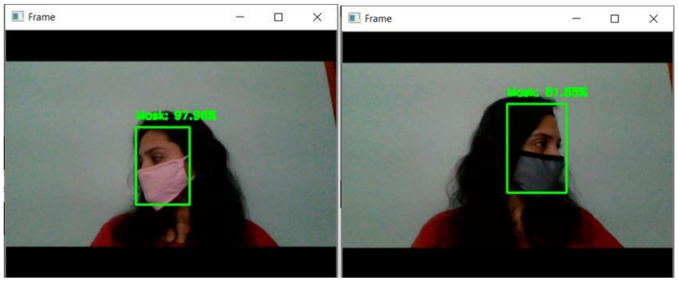

Fig. 2.12 Final detecting of face mask when face is turned

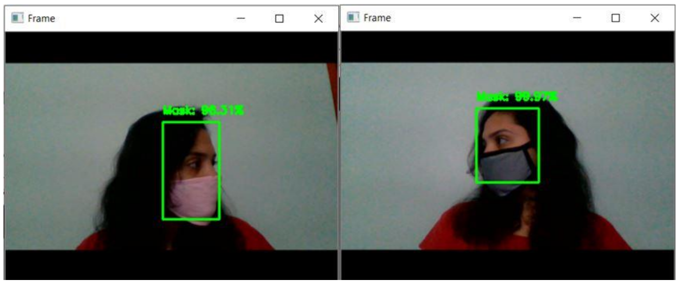

Fig. 2.13 Final detecting of face mask when face is turned

The face mask is detected even at a tilted or ratated angle with an accuracy of 65-95% as shown in Fig 2.11 overcoming the limitation in paper [4].

The face is detected even when turned left and right where the region of interest reduces with equally high accuracy as shown in Fig 2.12 and Fig 2.13.

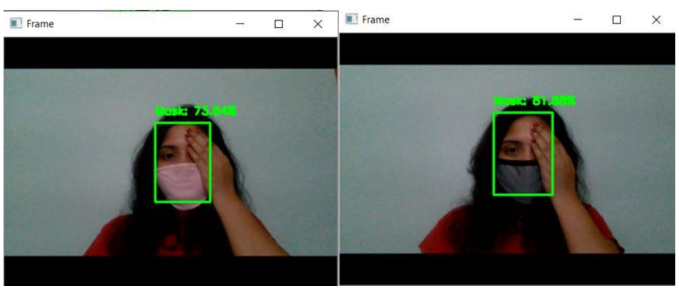

Fig. 2.14 Final detecting of face mask when a feature is covered.

Even if a feature of face is covered it's still a face and must be detected. Fig 2.14 shows the output image where our model detects the face and mask even when one of the feature is covered which overcomes the limitation of not being able to detect face when a feature is covered in paper[5][6][7]

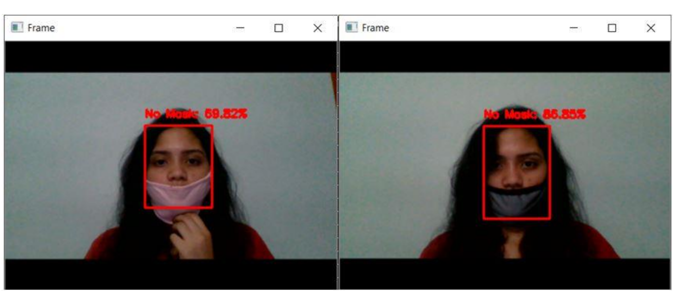

Fig. 2.15 Final detecting of face mask when face is turned

The ideal way of wearing a mask is covering both mouth and nose appropriately. If the nose is not covered it's the wrong way to wear a mask and will be detected as no mask as shown in Fig 2.15

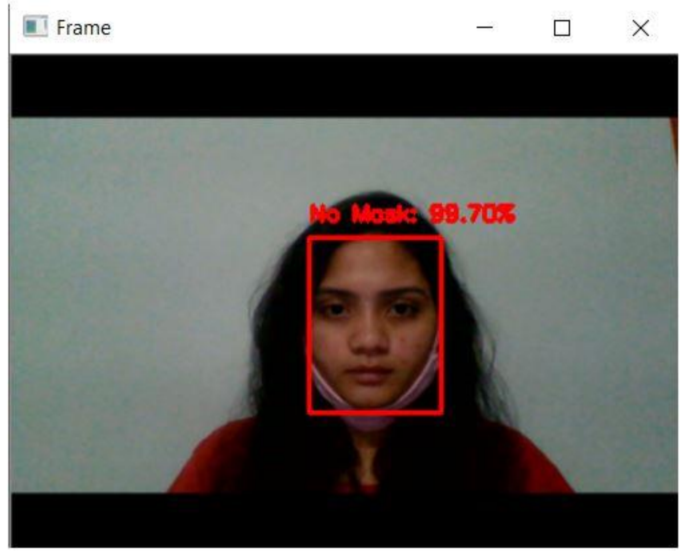

Fig. 2.16 Final detecting of face without mask.

When the mask is not worn itwill detect as no mask as shown in Fig 2.16.

VI. Conclution

In this paper, we have proposed a face mask detector, which can possibly contribute to public healthcare. The architecture of this detector consists of MobileNet as the backbone of the project. In order to extract more robust features, we utilize transfer learning to adopt weights from a similar task face detection, which is trained on a very large dataset. Furthermore, we have proposed a Key Feature Extraction module to focus on the face (which are the chin and the nose bridge) and mask features.

The proposed algorithm contained three main steps: Face Detection and Crop, Data Set Creation, and CNN. The accuracy of the system increases overtime with the increasing number of trained data sets. Our findings indicate that the proposed that MobileNet could achieve high accuracy in facemask-identification, which is meaningful for the prevention of epidemic diseases involving COVID-19 in the public.

Acknowledgment

We have taken efforts in the project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them. We are highly indebted to **Prof. SURESH CHAVHAN** for their guidance and constant supervision as well as providing necessary information regarding the project & also for their support in completing the project. We would like to express our gratitude towards the members of our team for their kind co-operation and collaboration which helped us in completion of this project. Our thanks and appreciations also go to VIT, Vellore in developing the project and to the people who have willingly helped us out with their abilities.

REFERENCES

[1] https://arxiv.org/abs/2005.03950
[2] https://www.researchsquare.com/article/rs-28668/latest.pdf
[3] https://www.researchsquare.com/article/rs-39289/latest.pdf
[4] http://papers.nips.cc/paper/1747-a-snow-based-face-detector.pdf
[5] https://ieeexplore.ieee.org/abstract/document/4129189/
[6] https://ieeexplore.ieee.org/abstract/document/1000242/
[7] https://ieeexplore.ieee.org/abstract/document/4359325/
[8] https://ieeexplore.ieee.org/abstract/document/8888092/
[9] https://link.springer.com/chapter/10.1007/978-3-540-73007-1_85
[10] https://www.academia.edu/download/34901524/48455-aug14.pdf