AI-enabled Face Mask Detection Tool

[1.]Sai Suresh M V, [2]Vaishali S, [3] Subashini R, [4] Akshara K
UG Scholars

Department of Computer Science and Engineering
Velammal Engineering College, Chennai

saisuresh1999@gmail.com

***Abstract*—COVID-19 pandemic has affected people worldwide. The count of deceased and affected individuals is getting higher day-by-day. Governments are imposing many measures to limit the spread and to protect mankind. Wearing face masks is one of the measures that could possibly control the spread. If this is the case, every person must be checked before moving into public interaction. In this paper, we propose a model that could detect whether people are wearing a mask using image augmentation and convolutional neural networks. A pre-trained MobileNetV2 model is used to extract feature maps from the dataset. The model is trained with images that are augmented using in-place/on-the-fly data augmentation to improve the performance of CNN. Haar cascade classifier is used to identify the frontal faces in the frame, then the model predicts the faces with and without a mask and outputs the same. The model achieves an accuracy of around 95%.**

Keywords— COVID-19, MobileNetV2, CNN

I. INTRODUCTION

COVID-19 is an infectious disease caused by a coronavirus which mainly causes respiratory illness [1]. Over 17.3 million people have been tested positive and over 6.7 lakhs have died worldwide because of this virus as of July 31, 2020. Several symptoms disclose whether a person is affected by COVID-19 or not. The most common symptoms are dry cough, fever, and tiredness [2]. Besides, there is also another side of COVID-19 that does not show any symptoms but the test results positive. One of the assertive reasons for the spread of coronavirus is droplets from the respiratory aerosols which are expelled from infected patients when they speak, cough, or sneeze. An uninfected person, when exposed to these droplets, may get infected. Certain precautions reduce the risk of getting affected by this virus which includes washing our hands regularly, maintaining social distance up to 6 feet in public places, wearing a face mask and avoid touching of parts like eyes, nose, and mouth. One who doesn't wear a mask either contributes more to the spread of COVID-19 or becomes an easy target for the virus. If at least 75 percent of the population follow the trend of wearing a mask, then it will create a big impact in managing the current pandemic situation. The extensive use of a face mask along with social distancing could significantly reduce the R value of the virus. The World Health Organization is encouraging governments to promote people to wear a mask in public areas. If a person is wearing a mask the chances of transmission of the virus are 3% in comparison to 17% on not wearing a mask [3]. Wearing a mask is compulsory for those people who work in healthcare, those who are affected by COVID-19, and those who take care of COVID-19 infected patients. Also, masks have to be worn by every individual when they head towards grocery shops or to the places where it seems to be a crowd. Mask has to be worn properly such that it should cover both nose and the mouth area to protect from the virus droplets expelled from the infected persons. People show negligence in wearing masks even though the government is stressing them to wear masks and incorporate it as a part of their lifestyle. In some countries, wearing a mask is made compulsory as per the advice of national authorities. A face mask helps in high filtration by resisting fluid penetration. In situations like this, the best practice is to wear a mask and to avoid direct contact. This, in turn, reduces the transmissibility and helps in substantially reducing the death rate and the economic impact. In India, the lockdown is being lifted in phases, this increases the responsibility of public centres which may act as a hub in increasing the transmission and alleviating the situation. Precautions like physical distancing and wearing masks should be imposed strictly, on public centres. Manual checking would be useful in some ways, but in the current pandemic situation, there is insecurity among people which makes the manual way of checking not so reliable. This method may also act as a central point of spreading the virus from one to many. A system is required to constantly monitor the violation of precautions and prompt people to correct it accordingly. This system could be deployed in public centres and all the other crowd prone areas. Hence there is a need for a face mask detector that detects whether a person is wearing a mask or not. A model is developed that could efficiently differentiate between a person wearing a mask and not wearing a mask and to prompt them to do so by denying access to the place or conveying the information to concerned personnel. This face mask detector can be used in many places such as malls, temples, supermarkets, an educational institution which identifies the human face and looks for whether a person is wearing a face mask or not. This may help to identify those who are not wearing the mask and warn them for the same. Many kinds of research to contain the spread of disease, finding an efficient way to test using technology is being carried out worldwide but research related to the face mask detection is not discussed exhaustively. The face mask detection system could be of great use in future when the common public starts to return to their day-to-day life living in the new normal. This paper provides the implementation of Face Mask Detection

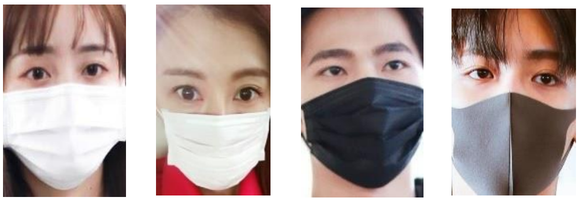

Fig. 1. With Mask.

Tool using Neural Networks, Keras, TensorFlow and OpenCV. With further improvements, these types of models could be integrated with CCTV or other types of cameras to detect and identify people without masks. With the prevailing worldwide situation due to COVID-19 pandemic, these types of systems would be very supportive of various kinds of institutions around the world.

II. DATASET DESCRIPTION

The dataset is obtained from GitHub [4] which is available as open-source. This dataset is prepared by web scraping where the classes are with a mask and without a mask. Mainly, the dataset consists of 3833 images in which 1915 images are with the mask and 1918 images are without the mask. Few images are depicted in Fig. 1 and Fig. 2 indicating with a mask and without a mask.

III. DATA PREPROCESSING

The entire data preprocessing is taken in Google Colab, where the dataset is uploaded to the google drive and it has been mounted to the Google Colab. Initially, the required python packages are imported in which some of them are tensorflow.keras.preprocessing.image, NumPy. The main python package is NumPy because it is developed in a way to reduce the time complexity for high-level mathematical manipulations and also tensorflow.keras.preprocessing.image plays a major role in image preprocessing. The image is loaded in the PIL format because for the obvious reason where there are around 3800 images, so PIL format is a faster way to load an image than the traditional way of loading an image. The parameters for loading an image is the path, grayscale which by default is false, the colour mode which by default is RGB, target size which is (224,224) and the interpolation which by default is nearest neighbour technique. The image size is set to (224,224) because the base model has the weights pre-trained on ImageNet. So, the image shape has to be (224,224,3) since the usage of MobileNetV2. Thereafter, the conversion of a PIL Image instance to a NumPy array is performed. The function of the image array returns a 3D NumPy array where the data type by default is float32. Since the MobileNetV2 is used for the construction of the base model, the pixel value should be in between [-1, 1], so the usual way of dividing each pixel by 255 doesn't work out. Therefore, the function preprocess_input which is available in the package of mobilenetv2 has been used to fit the image to the required format of the model which in this case, is MobileNetV2 as the base model. Hence, the

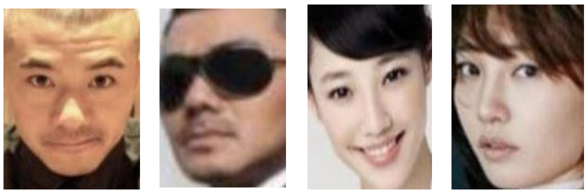

Fig. 2. Without Mask.

image preprocessing is completed whereas the labels need to be in an organized manner. Considering label for “NO MASK” as 0 and label for "MASK ON" as 1, the list of labels undergoes one-hot encoding. One hot encoding lets the representation of categorical data to be more significant. Finally, the data is preprocessed and ready for usage in the model.

IV. IMAGE AUGMENTATION

The simple understanding of image augmentation is the transformation of an original image into its various forms without changing its label. The example of an augmented image into its various forms is shown in Fig. 3. The image augmentation is commonly misunderstood because of the occurrence of the word "augment ", which means increase or add, but in the context of AI, in image augmentation, the old data will not be used for the training itself. The Image Data Generator is a class in Keras that does Image Augmentation. The image augmentation works in a way where accepting a batch of images is used for training and taking this batch and applying a set of random transformations to each image in the batch (including shearing, resizing, arbitrary rotation, etc.) [5]. Finally, replacing the original batch with the new randomly transformed batch and used for training the Convolutional Neural Network on this randomly transformed batch (i.e., the original data itself is not used for training). The Image Data Generator will return only the randomly transformed data rather than the original class of data. There are three types of data augmentation which are dataset generation and data expansion through data augmentation which is used less, in-place/on-the-fly data augmentation which is used the most and finally, combining dataset generation and in-place augmentation [5]. The second type, which is the in-place/on-the-fly data augmentation is used in this context. The name “in-place” and “on-the-fly” data augmentation is because the augmentation is held during training time (i.e. ahead of time/before training, the augmentation doesn't take place). There are several parameters in the Image Dataset Generator function, some of them are rotation range, horizontal flip, vertical flip, shear range, zoom range, width shift range, height shift range, etc. These parameters can be fine-tuned based on the sensibility of the data required. The main role of the Image Augmentation is to ensure that the CNN perceives the “new” images that it has never “seen” before at each epoch. Also, Image Augmentation is to enhance the generalizability of the model.

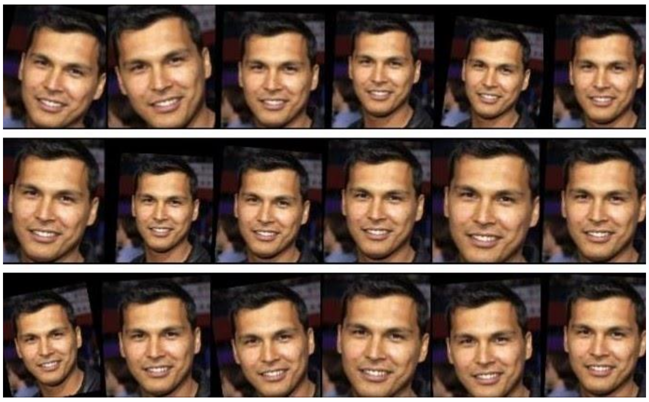

Fig. 3. Augmented Image into its various form obtained from [6].

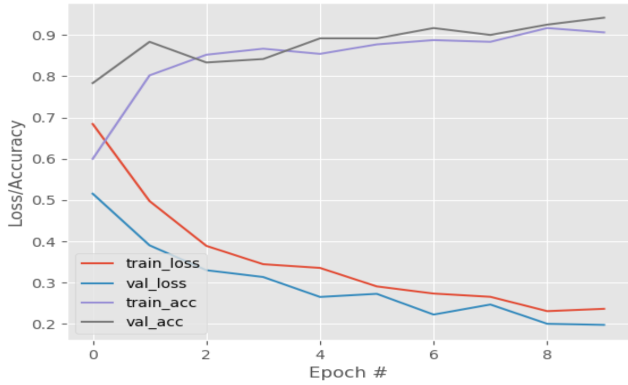

Fig. 4. Training Loss and Accuracy.

V. CONSTRUCTION AND WORKING MODEL

A. BASE MODEL

A pre-trained model is a preserved network that was earlier trained on a huge dataset, typically on a large-scale image-classification task and it is used as the pre-trained model as it is or uses transfer learning to customize this model to a given task [7] . The foreknowledge behind having a pre-trained model for image classification is that if a model is trained with enormous and general enough dataset, this model will adequately serve as a nonexclusive model of the seen world [8]. The advantage of these learned feature maps enables the training of a comprehensive model on a huge dataset without having to start from scratch.

In this context, the base model is MobileNetV2 which was developed by Google, pre-trained on the ImageNet dataset with 1.4M images and 1000 classes of web images [9]. There are 2 types of blocks present in MobileNetV2, one is a residual block with a stride of 1 and another one is a block with a stride of 2 for downsizing. The top layers (i.e. the output layers) are set manually to solve the problem of this context which is obviously, with the mask or without the mask. The remaining layers should be set as non-trainable because, the weights used when the model is trained on ImageNet should not be disrupted.

B. HEAD OF THE MODEL

Head of the model will be placed on top of the base model. The output layer of the whole model is considered as head of this model, where it consists of several layers. Initially, 4D tensor as input in which the 4 entities are batch size, rows, columns and channels. In this context, the batch size is assigned as 32 and the rows, columns, channels are represented as (224,224,3) respectively which is a usual occurrence. The average pooling which calculates the average value for each patch on the feature map. Generally, Max Pooling is to down sample the feature maps. This is done to decrease the resolution of the pre-processed image but also to enhance the receptive fields of the following layers because it is a form of down sampling that throws away all the information about the spread with average-pooling. The output from this layer is a 4D tensor which includes batch size, pooled rows, pooled columns and channels. Thereafter, a flatten layer which will flatten the 4D tensor into a unidimensional tensor. The unidimensional tensor which is a single vector consisting of each pooled feature maps. The Flatten function flattens all the feature maps into a single column [10]. Dense adds the fully connected layer to the neural network [10]. For the dense layer, the activation function used is ReLU (rectified linear activation unit). Since the MobileNetV2 is used, the pixel values are transformed such that the value lies between [-1,1], but the value of each pixel is regularly in between [0,1] therefore, the ReLU function acts as an activation function which has a property of showing X if X>0 then as X else as 0. This dense layer is fully connected so the dropout layer is used after the dense layer. The dropout rate is set as 0.5 which is used to prevent the model from becoming overfit. Finally, the output layer is a dense layer of 2 neurons and the activation function is SoftMax. Even, the sigmoid function can be used as an activation function since the context deals with binary classification. In the whole process of construction of the head of the model, the hyperparameters are decided on the basis of the theoretical and default approach.

C. TRAINING OF MODEL

The hyperparameters like Learning rate, where the amount of the weights that are updated during training are indicated to be the step size or the “learning rate” which is set to 0.0001. In training a neural network, the learning rate is a configurable hyperparameter that has a small positive value, often in the limit between 0.0 and 1.0 [11]. Also, the batch size is 32 which is a power of 2. The batch size is kept in power of 2 because the mapping of the virtual processor onto the physical processor takes place comfortably. The compilation of the model is performed using a custom Adam optimizer, where the decay rate is given as,

$$Decay\ rate = \frac{\text{Initial Learning Rate}}{\text{Number of Epochs}}$$

Thereafter, the loss function used here is binary cross-entropy because the context deals with binary classification and the evaluation metric used is accuracy because the task is classification. Finally, the model is ready to get fit, where the function of image augmentation is called, to perform in-place/on-the-fly data augmentation while training. Therefore, Fig. 4 depicts the plot of training loss and accuracy. The training loss is around 0.2 and accuracy is around 95% which is a better result at the initial stage. The model is also saved in h5 format because this model is to be loaded while detecting the face with or without the mask.

VI. DETECTING THE FACE WITH/WITHOUT MASK

The idea is to use Cascade Classifier to detect the faces in every frame of the video source which can be either a live stream or recorded source. Therefore, the Haar cascade for frontal face is used to detect the faces in each frame and for each face apply prediction and finally display the output by putting a rectangle around the face and a text above the rectangle which is either "MASK ON" or "NO MASK" with green and red coloured box respectively. After performing the prediction of all faces on the frame, resize the frame back to original size and write it to the output source. Some of the frames of the output source are shown in Fig. 5 and Fig. 6. In Fig. 5 it is noticeable that a man is not detected because of the cap. Since the facial features are already covered by a mask and as an add on the cap is there which acts as a hindrance to the cascade classifier.

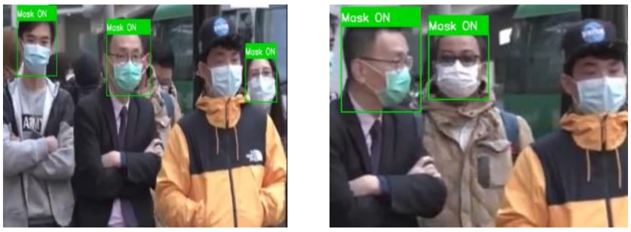

Fig. 5. Few frames of output depicting the people with the mask.

Fig. 6. Few frames of output depicting a man without the mask.

VII. CONCLUSION

The current pandemic demands a crucial and timely response. Each and every safety precaution are being followed. Wearing a mask is one among such precautions, monitoring is necessary to ensure that at crowded areas. The model developed could be used to monitor the general public. Computer vision is used to develop a robust system that will classify objects and react accordingly. The future works are focussed on training with tricky images that may confuse the model thereby enhancing its accuracy.

VIII. REFERENCES

[1] "What is COVID-19" - Website link- https://www.who.int/health-topics/coronavirus#tab=tab_1

[2] "Common symtoms of COVID-19"-Website link – https://www.healthline.com/health-news/flu-allergies-coronavirus-different-symptoms

[3] https://www.livescience.com/face-masks-eye-protection-covid-19-prevention.html

[4] https://github.com/balajisrinivas/Face-Mask-Detection/tree/master/dataset

[5] "Data Augmenatation "- Website link - https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/

[6] "Demystifying Face Recognition V: Data Augmentation" - Website link - https://melgor.github.io/blcv.github.io/static/2018/02/27/demystifying-face-recognition-v-data-augmentation/index.html#group-7

[7] "Transfer Learning in keras with Computer Vision Model "– Weblink – https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/

[8] "Image classification" - Website link - https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751

[9] "A practical Guide To Implement Transfer Learning : MobileNet V2 In TensorFlow" - Website link - https://analyticsindiamag.com/a-practical-guide-to-implement-transfer-learning-in-tensorflow/

[10] "Convolutional Neural Network : An Intro Tutorial " - Website link - https://heartbeat.fritz.ai/a-beginners-guide-to-convolutional-neural-networks-cnn-cf26c5ee17ed

[11] "How to Configure the Learning Rate When Training Deep Learning Neural Networks" - Website link – https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks/