Sustainable Cities and Society 66 (2021) 102692

Contents lists available at ScienceDirect

Sustainable Cities and Society

journal homepage: www.elsevier.com/locate/scs

SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2

Preeti Nagrath [a], Rachna Jain [a], Agam Madan [a], Rohan Arora [a], Piyush Kataria [a], Jude Hemanth [b,*]

[a] Dept. of Computer Science and Engineering, Bharati Vidyapeeth's College of Engineering, Delhi, India
[b] Department of ECE, Karunya Institute of Technology and Sciences, Coimbatore, India

ARTICLE INFO

Keywords:
Bottleneck
Convolutional Neural Network
Data augmentation
Fine tuning
MobileNetV2

ABSTRACT

Face mask detection had seen significant progress in the domains of Image processing and Computer vision, since the rise of the Covid-19 pandemic. Many face detection models have been created using several algorithms and techniques. The proposed approach in this paper uses deep learning, TensorFlow, Keras, and OpenCV to detect face masks. This model can be used for safety purposes since it is very resource efficient to deploy. The SSDMNV2 approach uses Single Shot Multibox Detector as a face detector and MobilenetV2 architecture as a framework for the classifier, which is very lightweight and can even be used in embedded devices (like NVIDIA Jetson Nano, Raspberry pi) to perform real-time mask detection. The technique deployed in this paper gives us an accuracy score of 0.9264 and an F1 score of 0.93. The dataset provided in this paper, was collected from various sources, can be used by other researchers for further advanced models such as those of face recognition, facial landmarks, and facial part detection process.

1. Introduction

COVID-19 pandemic has had a lasting impact in many countries worldwide since December 2019. It originated in Wuhan, China. The World Health Organization (WHO) as on March 11, 2020, declared it as a deadly diseases that gained its roots across the globe and severely affected 114 countries. Every medical professionals, healthcare organizations, medical practitioners and researchers are in search for a proper vaccines (Megahed & Ghoneim, 2020) and medicines to overcome this deadly disease, however no breakthrough has been reported till date. The virus spreads through air channel when an infected person sneezes or communicate with the other person, the water droplets from their nose or mouth disseminate through the air and affect other peoples in the vicinity (Kumar et al., 2020). Face Mask detection has become a trending application due to the Covid-19 pandemic, which demands a person to wear face masks, keep social distancing, and use hand sanitizers to wash their hands. While other problems of social distancing and sanitization have been addressed until now, the issue of face mask detection has not yet been adequately addressed. Wearing a mask during this pandemic is a critical preventive measure (Rahmani & Mirmahaleh, 2020) and is most vital step in times when social distancing is hard to maintain. Wearing a mask is essential, particularly for those people who are at a greater risk of severe illness from COVID-19 diseases. It is found that the spread of COVID-19 is mainly among people who are in immediate contact with one another (nearly about 6 feet), it can be spread by people who do not have symptoms and are unaware of the fact that they are infected (Ge et al., 2020). So Centers for Disease Control and Prevention (CDC)[1] recommended all people 2 years of age and older to wear a mask in public areas especially when other social distancing (Sun & Zhai, 2020) measures are difficult to maintain. Hence by reducing the risk of transmission of this deadly virus from an infected person to a healthy, the virus' spread and disease severity can be reduced to a great extent.

Face Mask detection has turned up to be an astonishing problem in the domain of image processing and computer vision. Face detection has various use cases ranging from face recognition to capturing facial motions, where the lattercalls for the face to be revealed withvery high precision. Due to the rapid advancement in the domain ofmachine

* Corresponding author.
E-mail address: judehemanth@karunya.edu (J. Hemanth).

[1] https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/cloth-face-cover-guidance.html#:~:text=Masks%20are%20a%20critical%20preventive,and%20disinfecting%20frequently%20touched%20surfaces.

https://doi.org/10.1016/j.scs.2020.102692
Received 8 August 2020; Received in revised form 25 December 2020; Accepted 26 December 2020
Available online 31 December 2020

learning algorithms, the jeopardies of face mask detection technology seem to be well addressed yet. This technology is more relevant today because it is used to detect faces not only in static images and videos but also in real-time inspection and supervision. With the advancements of convolution neural networks (Lawrence, Giles, Tsoi, & Back, 1997) and deep learning (Ahmed, Ahmad, Rodrigues, Jeon, & Din, 2020), very high accuracy in image classification and object detection can be achieved.Probably because of the sudden emergence of the COVID-19 pandemic, at present, there are various facial recognition technology applied to people wearing masks. HanvonTechnology Wang et al. (2020) reported that the accuracy of masked face recognition is about 85 %. An accuracy of over 90 % was obtained from Minivision Technology (Wang et al., 2020). The face-eye-based multi-granularity model (Wang et al., 2020) achieves 95 % recognition accuracy. In Li, Wang, Li, and Fei (2020), the authors used the YOLOv3 algorithm for face Mask detection. This method achieved 93.9 % accuracy. The accuracies achieved were on artificial dataset which was not the case in this paper which uses both real and artificial images.

A model named as SSDMNV2 has been proposed in this paper for face mask detection using OpenCV Deep Neural Network (DNN) (Velasco--Montero, Fernández-Berni, Carmona-Galán, & Rodríguez-Vázquez, 2018), TensorFlow (Abadi et al., 2016), Keras, and MobileNetV2 architecture (Nguyen, 2020) which is used as an image classifier. SSDMNV2 performs competently in differentiating images having frontal faces with masks from images having frontal faces without masks.To impede the COVID-19 transmission the proposed model can be integrated with surveillance cameras so that it can be used for the detection of people who are not wearing face masks. This paper also keeps complete attention towards the removal of various inaccurate predictions mainly in cases of real world datasets that occurred in different other proposed models (Li et al., 2020; Wang et al., 2020).

Detection of face masks is an extremely challenging task for the present proposed models of face detectors (Chen, Hua, Wen, & Sun, 2016; Li, Sun, Wu, & Wang, 2016; Ranjan, Patel, & Chellappa, 2017; Zhang, Zhang, Li, & Qiao, 2016; Zhu, Zheng, Luu, & Savvides, 2017). This is because faces with masks have varied accommodations, various degrees of obstructions, and diversified mask types. They are used to facilitate self-focusing (Huang, Ai, Li, & Lao, 2007), the interaction between humans and computers (Jun, Choi, & Kim, 2012), and managing image database (Ge, Li, Ye, & Luo, 2017). Even after having such extraordinary and exceptional results in the existing face detectors, there is still high rising scrutiny in the development of more advanced face detectors as for existing models, event analysis and video surveillance is always a challenging job. Several reasons were found for the poor achievement of existing face mask detection model as compared to the normal ones, two of them were First due to lack of suitable datasets with properly masked faces and facial recognition. Secondly, the presence of masks on the face brings a certain kind of noise, which further deteriorates the detection process. These issues have been studied in some existing research papers such as (Ghiasi & Fowlkes, 2014; Opitz, Waltner, Poier, Possegger, & Bischof, 2016; Yang, Luo, Loy, & Tang, 2015) still, there is an excellent challenge for a vast dataset so that an efficient face mask detection model can be easily developed.

The main contributions of the paper are as follows:

i.) AGitHub repository is made available, which contains a self-made Dataset of masked faces, including datasets taken from online resources. This dataset could be used for developing new face mask detectors and performing several applications. This has been explained in Section 3.1.

ii.) OpenCV DNNs have been used for face mask detection, which allows for real-time detection without much resourceusage. It can also detect faces in different orientations and can also detect occluded faces with good accuracy.The proposed SSDMNV2 model outperforms various previous models.

iii.) Several provocations that were faced during the development of this model have been considered in this paper; this may help the reader to develop more improved face mask detectors.

The rest of the work discussed in the paper is structured as follows. The upcoming Section2 discusses the similar work proposed and completed in the realm of face mask detection. In Section 3, the dataset used, the methodology adopted and technology used to build this Face mask detection model is discussed. In Section 4, experimental results have been shown along with a comparison table. Section 5 shows the conclusion.

2. Related work

In the recent past, various researchers and analysts mainly focused on gray-scale face image (Ojala, Pietikainen, & Maenpaa, 2002). While some were completely built on pattern identification models, possessing initial information of the face model while others were using AdaBoost (Kim, Park, Woo, Jeong, & Min, 2011), which was an excellent classifier for training purposes. Then came the Viola-Jones Detector, which provided a breakthrough in face detection technology, and real-time face detection got possible. It faced various problems like the orientation and brightness of the face, making it hard to intercept. So basically, it failed to work in dull and dim light.Thus, researchers started searching for a new alternative model that could easily detect faces as well as masks on the face.

In the past, many datasets for face detection were developed to form an impression of face mask detection models. Earlier datasets consisted of images fetched in supervised surroundings, while recent datasets are constructed by taking online images like WiderFace (Yang, Luo, Loy, & Tang, 2016), IJB-A (Klare et al., 2015a), MALF (Yang, Yan, Lei, & Li, 2015), and CelebA (Klare et al., 2015b). Annotations are provided for present faces in these datasets as compared to earlier ones. Large datasets are much more needed for making better training and testing data and perform real-world applications in a much simpler way. This calls for various deep learning algorithms which can read faces and mask straight from the data provided by the user.

Face Mask detection models have many variations. These can be divided into several categories. In Boosting-based classification, boosted cascades with easy haar features were embraced using the Viola-Jones face detector (Jones, Viola, & Jones, 2001), which was discussed above in this section. Then a Multiview face mask detector was made motivated by the Viola-Jones detector model. In addition to this, a face mask detector model was made using decision trees algorithms. Face mask detectors in this category were very effective in detecting face masks.

In Deformable Part Model-based classification, the structure and orientations of several different faces are modelled using DPM. In 2006 Ramanan proposed a Random forest tree model in face mask detection, which accurately guesses face structures and facial poses. Zhang, Zhang, Li, and Qiao (2016), one of the renowned researchers made a DPM-based face mask detector using around 30, 000 faces divided into masks and without masks category. His work achieved an exceptional accuracy of 97.14 %. Further models of face mask detectors were made by Chen, Ren, Wei, Cao, and Sun (2014). Typically, DPM-based face mask detection models can achieve majestic precisions, but it may be tolerant from the very towering cost of computation due to the use of DPM.

In Convolutional Neural Network-based classification, face detector models learn directly from the user's data and then apply several deep learning algorithms on it (Ren, He, Girshick, & Sun, 2015). In the year 2007, Li, Lin, Shen, Brandt, and Hua (2015) came up with Cascade CNN. In Yang, Yan et al. (2015), Yang et al. came up with the idea of features aggregation of faces in the face detection model. In further research works, Ojala et al. (2002) upgraded the AlexNet architecture for fine-tuning the image dataset. For uninhibited circumstances, Zhu et al.

(2017) propose a Contextual Multi-Scale Region-based Convolutional Neural Network (CMS-RCNN), which brought a significant impact on the face detection models. To minimize the error on the substitute layers of CNN layers and dealing with the biased obstructions generated in the mask detection models, Opitz et al. (2016) prepared a grid loss layer. As technology advanced, further CNN-based 3D models started coming up; one was proposed by Li et al. (2015). It was a learning structure for face mask detection models. Several other works were done in the sphere of pose recognition, gender estimation, localization of landmarks, etc.

TheFace mask detection model named SSDMNV2has been developed using deep neural network modules from OpenCV and TensorFlow, which contains a Single Shot Multibox Detector object detection model. Typical classification architectures like ResNet-10 which is used as a backbone architecture for this model and image classification and fine-tuned MobileNetV2 classifier has been used, MobileNetV2 classifier has been an improvement over MobileNetV1 architecture classifier as it consisted of 3×3 convolutional layer as the initial layer, which is followed by 13 times the previous building blocks. In contrast, MobileNetV2 architecture is made up of 17, 3×3 convolutional layers in a row accompanied by a 1×1 convolution, an average layer of max pooling, and a layer of classification. The residual connection is a new addition in the MobileNetV2 classifier.

3. Proposed methodology

To predict whether a person has worn a mask correctly, the initial stage would be to train the model using a proper dataset. Details about the Dataset have been discussed above in Section 3.1. After training the classifier, an accurate face detection model is required to detect faces, so that the SSDMNV2 model can classify whether the person is wearing a mask or not. The task in this paper is to raise the accuracy of mask detection without being too resource-heavy. For doing this task, the DNN module was used from OpenCV, which contains a 'Single Shot Multibox Detector' (SSD) (Liu et al., 2016) object detection model with ResNet-10 (Anisimov & Khanova, 2017) as its backbone architecture. This approach helps in detecting faces in real-time, even on embedded devices like Raspberry Pi. The following classifier uses a pre-trained model MobileNetV2 (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018) to predict whether the person is wearing a mask or not. The approach used in this paper is depicted in the flow diagram in Fig. 1.

3.1. Dataset used

There are only a few datasets available for the detection of face masks. Most of them are either artificially created, which doesn't represent the real world accurately, or the dataset is full of noise and wrong labels. So, choosing the right dataset which would work best for the SSDMNV2 model required a little effort. The dataset used in for training the model in a given approach was a combination of various open-source datasets and pictures, which included data from Kaggle's Medical Mask Dataset by Mikolaj Witkowski and Prajna Bhandary dataset available at PyImageSearch. Data were also collected using the dataset provided by the masked face recognition dataset and application (Wang et al., 2020). The Kaggle dataset contains pictures of people wearing medical masks and XML files containing their descriptions and masks. This dataset had a total of 678 images. The other Artificial mask dataset was taken from 'Prajna Bhandary' from PyImageSearch. The dataset includes 1,376 images separated into two classes with wearing masks, 690 pictures, and without wearing a mask, 686 pictures.

The artificial dataset created by Prajna Bhandary took standard images of faces and applied facial landmarks. Facial landmarks allowed to locate facial features of a person like eyes, eyebrows, nose, mouth, and jawline. This used an artificial way to create a dataset by including a mask on a non-masked person image. Still, those images were not again used in the artificial generation process. The use of non-face mask samples involved the risk of the model becoming heavily biased. It was a risk to use such dataset images from various other sources. So they have included a dataset which would consist of images of people with masks and unmasked, which compensated for the error correction.

In the end, a dataset which included 5521 images was obtained, having the label "*with_mask*" and "*without_mask*" also contained 5521 images to make a balanced dataset. The distribution between the two classes is visualized in Fig. 2. The created dataset can also be used for detecting assailants who cover their faces while performing unlawful deeds. The dataset has been made available at https://github.com/TheSSJ2612/Real-Time-Medical-Mask-Detection/releases/download/v0.1/Dataset.zip.

3.2. Pre-processing

The Dataset from Masked face recognition and application contained a lot of noise, and a lot of repetitions were present in the images of this dataset. Since a good dataset dictates the accuracy of the model trained on it, so the data from the above-specified datasets were taken. They were then processed, and also all the repetitions were removed manually. The data cleaning was done manually to remove the corrupt images which were found in the said dataset. Finding these images was a vital part. As it is well known, the corrupt image was a tricky task, but due to valid efforts, we divided the work and cleaned the data set with mask images and without mask images. Cleaning, identifying, and correcting errors in a dataset removes adverse effects from any predictive model.

This part explains the procedure of pre-processing the data and then training on data. First, we define a function name sorted alphanumerically to sort the list in lexicographical order. A function pre-processing is defined, which takes the folder to the dataset as input, then loads all the files from the folder and resizes the images according to the SSDMNV2 model. Then the list is sorted using sorted alphanumerically, and then the images are converted into tensors. Then the list is converted to NumPy array for faster calculation. After this, the process of data augmentation is applied to increase the accuracy after training the

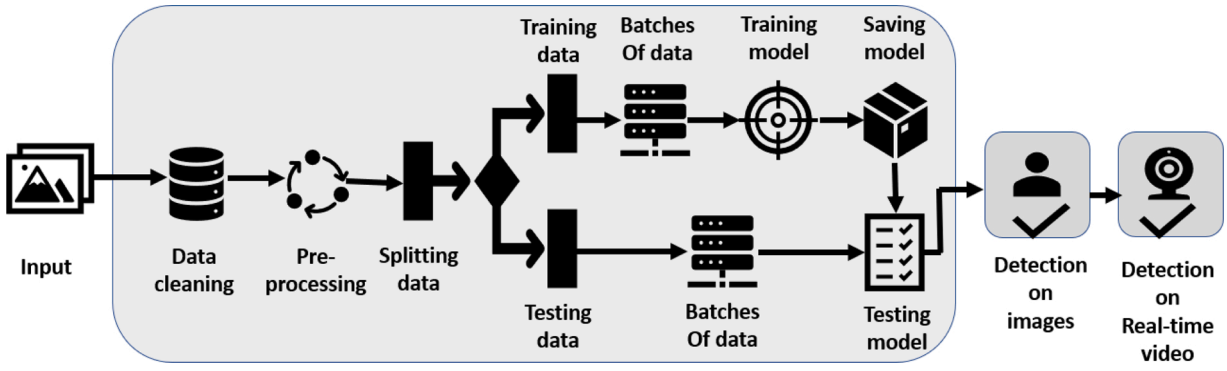

Fig. 1. Flow Diagram of the SSDMNV2 model.

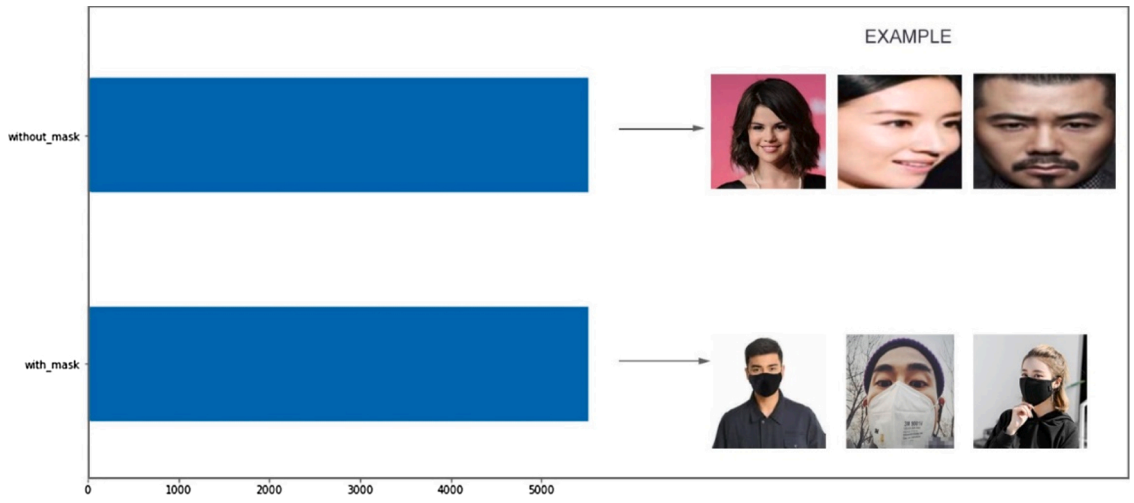

Fig. 2. Dataset Visualization.

model.

3.3. Data augmentation

For the training of SSDMNV2 model, an enormous quantity of data is necessary to perform training effectively since due to the non-availability of an adequate amount of data for training the proposed model. The method of data augmentation is used to solve this issue. In this technique, methods like rotation, zooming, shifting, shearing, and flipping the picture are used for generating numerous versions of a similar picture. In the proposed model, image augmentation is used for the data augmentation process. A function image data generation is created for image augmentation, which returns test and train batches of data.

3.4. Face detection using OPEN-CV DNN

This model is included in the GitHub repository of OpenCV, starting from the latestversion. It is established on the 'Single Shot Multi-box Detector' (SSD) and uses 'ResNet-10' architecture as the base-model. The Single Shot Multi-box Detector is similar to YOLO technique which takes only one shot to detect multiple objects present in an image using Multibox. It is significantly faster in speed and high-accuracy object detection algorithm. The images from which the model is trained have not been disclosed. Two versions of the model are made available by OpenCV:

i.) CaffeImplementation (Jia et al., 2014) (Floating point 16 version)
ii.) Original TensorFlow Implementation (8-bit quantized version)

Caffe is a deep learning framework developed as a more faster, powerful and efficient alternative as compared to other object detection methods and is created and managed by Berkeley AI Research (BAIR) and community contributors.We have used the Caffemodel for our implementation for SSDMNV2 model to detect faces for the detection of facial masks. For this, the Caffemodel and prototxt files were loaded using *cv2.dnn.readNet ("path/to/prototxtfile", "path/to/caffemodel-weights")*. After applying the face detection model, we get the number of faces detected, the location of their bounding boxes, and the confidence score in those predictions. These outputs are then used as input for the face mask classifier. Using this approach to detect faces allows for real-time detection without much resource usage. It can also detect faces in different orientations, i.e., left, right, top, and bottom, with good accuracy. It is also able to detect the face mask of different sizes, i.e., big or small.

3.5. Classification of images using MobileNetV2

MobileNetV2 is a Deep Neural Network that has been deployed for the classification problem. Pretrained weights of ImageNet were loaded from TensorFlow. Then the base layers are frozen to avoid impairment of already learned features. Then new trainable layers are added, and these layers are trained on the collected dataset so that it can determine the features to classify a face wearing a mask from a face not wearing a mask. Then the model is fine-tuned, and then the weights are saved. Using pre-trained models helps avoid unnecessary computational costs andhelps in taking advantage of already biased weights without losing already learned features. This plan of action is shown in Fig. 3.

3.5.1. Building blocks of MobileNetV2

MobileNetV2 is a deep learning model based on Convolutional Neural Network, which uses the following layers and functions. The structure of MobileNetV2 has been shown in Fig. 4.

- ***Convolutional Layer***

This layer is the fundamental block of the Convolutional Neural Network. The term convolution implies a mathematical combination of two functions to get the third function. It works on a sliding window mechanism, which helps in extracting features from an image. This helps in generation feature maps. The convolution oftwo functional matrices, one being the input image matrix A and the other being convolutional kernel B give us the output C as:

$$C(T) = (A * B)(x) = \int_{-\infty}^{\infty} A(T) \times B(T - x)\, dT \quad (1)$$

Fig. 5 shows the convolution operation between the two matrices.

- ***Pooling Layer***

Applying the pooling operations can make calculations go faster by allowing a reduction in the size of the input matrix without losing many features. Different kind of pooling operations can be applied out of which some are explained below:

i.) Max Pooling: It takes the maximum value present in the selected region where the kernel is currently at as the value for the output of matrix value for that cell.
ii.) Average Pooling: It takes the average of all the values that are currently in the region where the kernel is at and takes this value

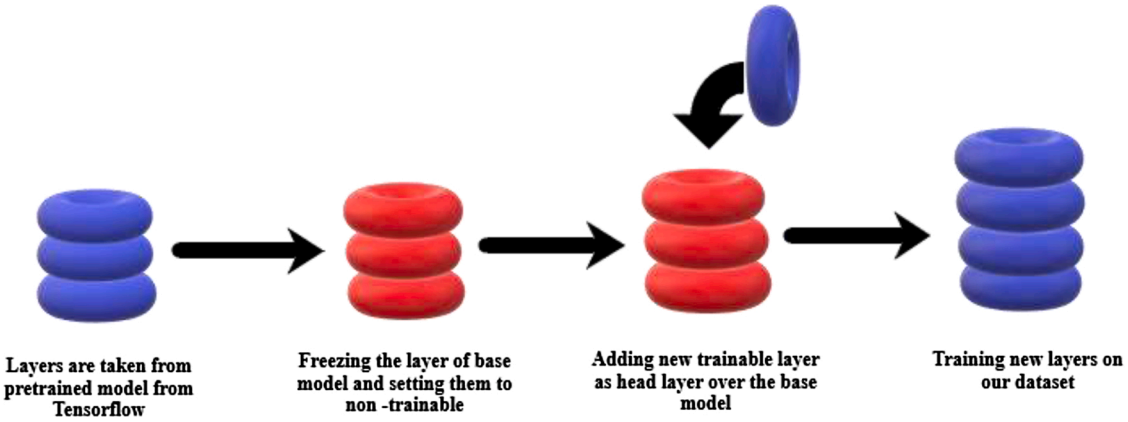

Fig. 3. Pipeline of using Pretrained Model.

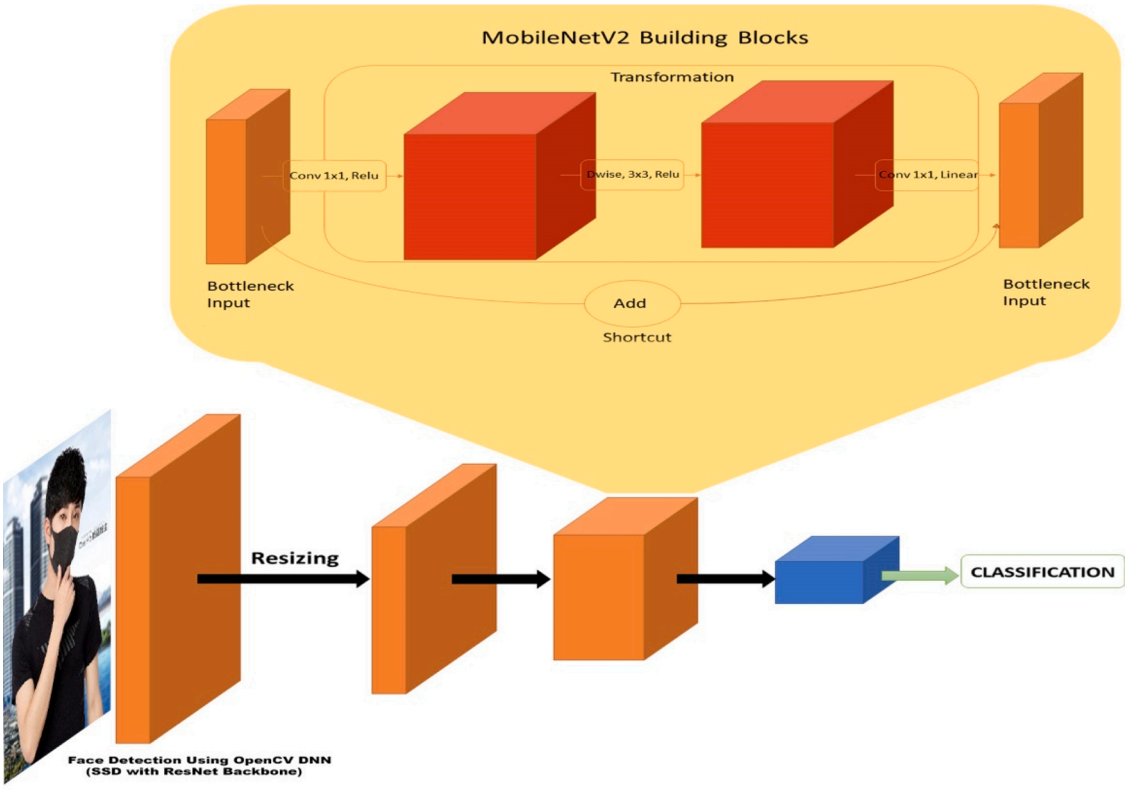

Fig. 4. Architecture of MobileNetV2.

as the output for the matrix value of that cell. Fig. 6 shows the average-pooling operation.

- ***Dropout Layer***

This helps reduce the overfitting, which may occur while training by dropping random biased neurons from the model. These neurons can be a part of hidden layers as well as visible layers. The likelihood for a neuron to be dropped can be changed by changing the dropout ratio.

- ***Non-Linear Layer***

These layers usually follow the convolutionallayers. Most commonly used non-linear functions include different kinds of Rectified Linear Unit (ReLU) (Hahnloser, Sarpeshkar, Mahowald, Douglas, & Seung, 2000), i. e., Leaky ReLU (Zhang, Zou, & Shi, 2017), Noisy ReLU (Gulcehre, Moczulski, Denil, & Bengio, 2016), Exponential ReLU (Opschoor, Schwab, & Zech, 2019), etc., sigmoid function as well as tanh functions. Different kinds of Non-Linear Functions and their equations are shown below.

$$Sigmoid: \ \sigma(x) = \frac{1}{1+e^{-x}} \tag{2}$$

$$Leaky\ Relu: \ f(x) = \max(0.1x,\ x) \tag{3}$$

$$Tanh: \ f(x) = \tanh(x) \tag{4}$$

$$Maxout: \ f(x) = \max\left(w_1^T x + \ b_1\ ,\ w_2^T x + b_2\right) \tag{5}$$

$$Relu: \ f(x) = \max(0, x) \tag{6}$$

$$ELU: \ f(x) = \begin{cases} x\ x \geq 0 \\ \alpha(e^x - 1)\ x < 0 \end{cases} \tag{7}$$

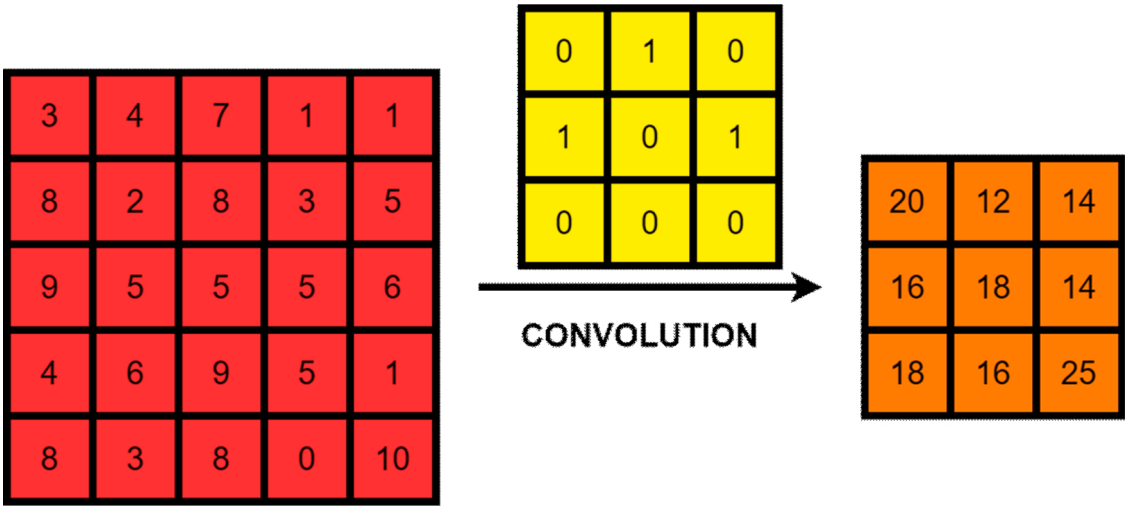

Fig. 5. Convolutional Operation.

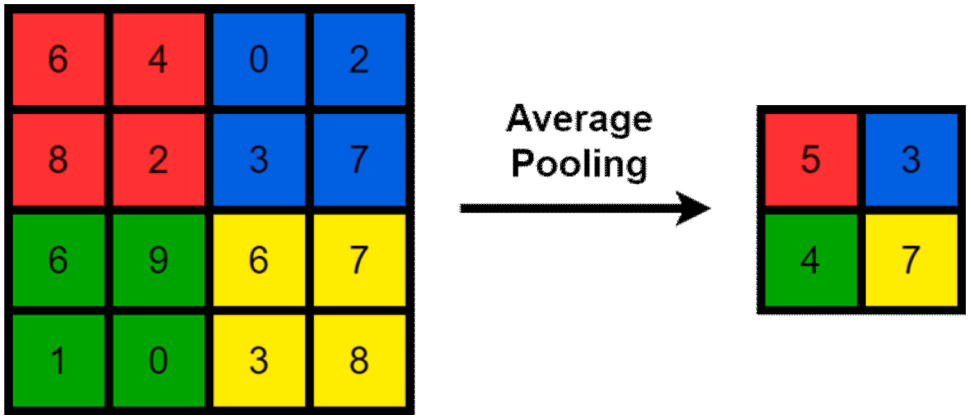

Fig. 6. Average-Pooling Operation.

- ***Fully-Connected Layer***

These layers are appendedin the model and have full connections to activations layers. These layers help in classifying the given images in multi-class or binary classification. SoftMax is an example of activation functions used in these layers, and it provides the result of predicted output classes in terms of probability.

- ***Linear Bottlenecks***

As multiple matrix multiplications cannot be diminished to a single numerical operation, non-linear activation functions like ReLU6 are applied in the neural networks, so several discrepancies can be easily removed. Through this a multilayer neural network can be built. Since the ReLU activation function abandons the values which are less than 0. The dimensions of the network increase by escalating the number of channels to challenge the loss of information.

For a reversed residual block, layers of the blocks are compressed, and the contrary procedure is done as done above. This is done at the point where the skip connections are linked; this could affect the execution of the network. To deal with this, the concept of the linear bottleneck was introduced in which before adding the block to initial activation, the last convolution of the left-over block is given a linear output.

3.6. Algorithms explaining the complete pipeline

The proposed SSDMNV2 methodology has been clearly explained using the two algorithms as shown below. First the images were pre-processed and were trained on the whole dataset. Second the model trained in the first part was used to detect the face mask with the appropriate accuracy. In Algorithm 1 shown below, images along with their pixel values were taken as an input, resized and normalized. Data augmentation technique was then applied on the images in order to get more accuracy. Data was then splitted into training and testing batches and MobilenetV2 model was applied on it. Adam optimizer was used to compile the whole model. In Algorithm 2 the model trained in previous part was then deployed on both classification on static images and on real time webcam. If the faces are detected using SSD, a bounding box showing the face of the person wearing a mask is shown in the output.

ALGORITHM 1: Pre-processing and Training on Dataset

INPUT: Images along with their pixels values
OUTPUT: Trained Model

STEP 1: Load Images and their pixel values.
STEP 2: Process the images, i.e., resizing, normalization, and conversion to a 1D array.
STEP 3: Load the Filenames and their respective labels.
STEP 4: Perform Data augmentation and then split data into training and testing batches.
STEP 5: Load MobilenetV2 model from Keras. Train it on training batches and compile it using Adam optimizer.
STEP6: Save the model for future use.

ALGORITHM 2: Deployment of Face Mask Detector

INPUT: Choice of deployment and Files(optional).
OUTPUT: Images classified into the mask and no mask or Classification in Real-time.

STEP 1: Load saved classifier from disk. Also, load face detector from OpenCV.
STEP 2: If the choice is classification on image:
Load Image(s)
STEP 2.1: Apply face detection model to Detect faces in an image
STEP 2.2: If Faces are detected:
Crop face to bounding box coordinates from face detection model
Get predictions from the face classifier model.
Show predictions and save resultant image.
Else:
Show no output
STEP 3: If the choice is classification in real-time:
Load real-time feed from OpenCV
Read the feed frame by frame.
STEP 3.1: Apply face detection model to Detect faces in Frames read in real-time
STEP 3.2: If Faces are detected:
Crop face to bounding box coordinates from face detection model
Get predictions from the face classifier model.
Show output in a real-time feed
Else:
Show normal feed
STEP 4: End stream when q is pressed

4. Experimental results

All the experimental trials have been conducted on a laptop equipped by an Intel i7-8750H processor (4.1 GHz), 16 GB of RAM with 1050ti max-Q with 4 GB of VRAM. The Jupyter Notebook software equipped with Python 3.8 kernel was selected in this research for the development and implementation of the different experimental trails.

The metrics selected for evaluation of SSDMNV2 model is explained below.

$$Accuracy = \frac{Tp + Tn}{(Tp + Fp + Fn + Tn)} \tag{8}$$

$$Precision = \frac{Tp}{(Tp + Fn)} \tag{9}$$

$$Recall = \frac{Tp}{(Tp + Fn)} \tag{10}$$

$$f1\ score = 2 * \frac{Recall * Precision}{(Recall + Precision)} \tag{11}$$

Where Tp = *True positive*,
Tn = *True negative*,
Fp = *False positive*,
Fn = *False negative*

In above formulas, True positive values refer to images which were labelled true and after prediction by model gave true result. Likewise, for True negative refers to images which were labelled true but after prediction resulted in false result. False positive refers to images which were labelled false and after prediction resulted in false hence false positives. False negative refers to images which were labelled false and

Fig. 7. Training accuracy curve (Without data augmentation).

Fig. 8. Training loss curve (Without data augmentation).

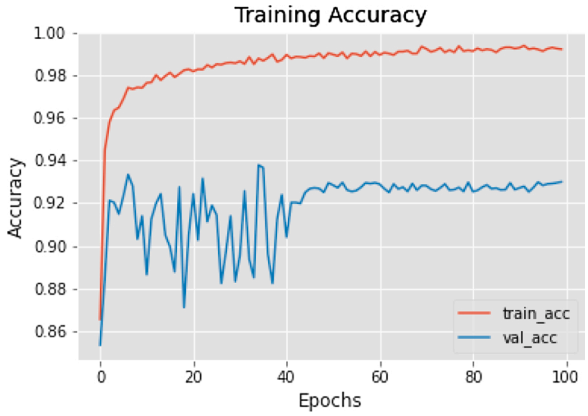

Fig. 9. Training accuracy curve.

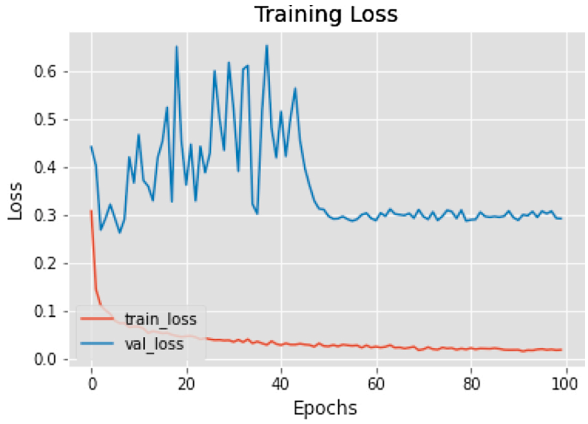

Fig. 10. Training loss curve on the train.

after prediction resulted in true hence false negatives. The accuracy was a good measure to start with, because the classes were balanced. Precision gave the measure of positive predicted values. Recall gave the ability to a classifier to find all positive samples and f1 score gave the measure of test accuracy. These evaluation metrics were chosen because of their ability to give best results in balanced dataset.

4.1. Pre-processing results

This part explains the results of pre-processing the data and then training on data. First, a function named sorted_ alphanumerically is used to sort the list in lexicographical order. A function pre-processing is defined, which takes the folder to the dataset as input, then loads all the files from the folder and resizes the images according to the model. Then after the list is sorted, the images are converted into tensors. Then the list is converted to NumPy array for faster calculation. After applying pre-processing, the accuracy of our model increased significantly. After this, the process of data augmentation is applied to increase the accuracy after training the model.

4.2. Data augmentation results

The results of SSDMNV2 model before data augmentation are shown in Figs. 7 and 8. The model without data augmentation showed significant decrease in its growth as when trained on 100 epochs after data augmentation. In Fig. 7 depicting training accuracy without data augmentation, the model struggled to learn features till 60 epochs in validation accuracy and became stable afterwards. After 100 epochs the training accuracy came out to be 87.51 % as compared to 92.64 % training accuracy when data augmentation is applied. In Figure 8depicting training loss on train set without data augmentation shows the training loss of the model till 10 epochs andstarted with approximately at 0.5 loss on validation loss maximum validation loss at a peak of 1.6 even at 80 epochs. Figs. 9 and 10 shows improvement after applying data augmentation which could not be achieved before data augmentation.

4.3. Evaluation of SSDMNV2 model

To solve the binary classification, a deep learning model problem is used in this paper. Keras is used to make a classification model in this paper, which is an advanced-level artificial neural networks API. Datasets for the (training, and testing) are split up to (80 % for training, 20 % for testing phase).The evaluation metrics used in this paper are accuracy, the area under the Receiver Operating Characteristics (ROC) curve, classification report, confusion matrix, and comparison of models. The plots are based on model accuracy; the pyplot command style function makes matplotlib work like MATLAB. The accuracy gives us the level of correct prediction of the masked person identified by the machine by the proposed model. In Fig. 9 depicting training accuracy with data augmentation, the model struggled to learn features till 42 epochs in validation accuracy and became stable afterwards. After 100 epochs the

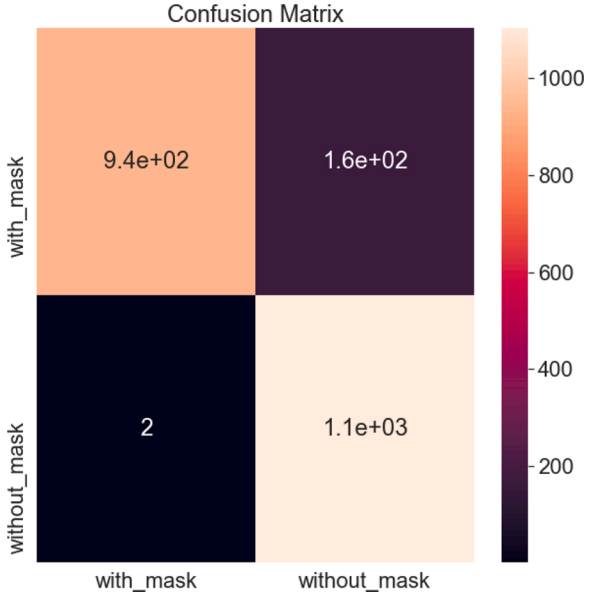

Fig. 11. Confusion matrix.

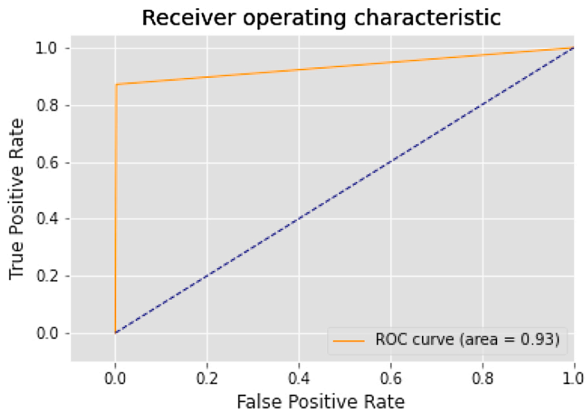

Fig. 12. Roc curve.

Table 1
Classification Report.

	Precision	recall	F1 Score	support
with mask	1.00	0.85	0.93	1104
without mask	0.87	1.00	0.93	1105
accuracy			0.93	2209
Macro average	0.94	0.93	0.93	2209
Weighted average	0.94	0.93	0.93	2209

training accuracy came out to be 92.64 %. The average accuracy of our model is '93 %' for predicting if a person is wearing a mask or not on a validation dataset, as shown in Fig. 9. The red curve shows the training accuracy, which is nearly equal to 99 %, whereas the blue line represents the accuracy of the validation dataset. The training loss curve corresponding to training andvalidation is shown in Fig. 10. Here, the red line shows the loss in training dataset less than 0.1, whereas the blue line depicts training loss on the validation dataset.

The confusion matrix shown in Fig. 11 depicts a form to compare the labels, model prediction, and actual labels it was supposed to predict. It is showing where the model is getting confused. The confusion matrix is plotted with the help of heatmap showing a 2D matrix data in graphical format. It has successfully identified 941 true positives, 163 false negatives, 2 false-positive, and 1103 true negative out of 2209 images used for validation.The Roc curve compares a model's 'true positive rate' (tpr) from a model's with 'false positive rate' (fpr), as shown in Fig. 12. The model's roc accuracy score is close to the ideal roc curve but not the same. The roc accuracy score showing 93 % predicts the correctness of predicting the model values.

The classification report shown in Table 1 explains the level of f1 score, recall, precision, and accuracy of SSDMNV2 model.

4.4. Visualization of results

Fig. 13 shows the predictions on some images. These are the predictions made on 2 test images by SSDMNV2 model using MobileNetV2. The rectangular green box depicts the correct way of wearing a mask with an accuracy score on the top left, while the red rectangular box represents the incorrect way of wearing a mask. The model learns from the pattern of train dataset and labels and then makes predictions.

4.5. Comparison of models

The SSDMNV2 model was also compared with different pre-existing models by training them on the same dataset, and the results from them have been shown in this paper. For this purpose, LeNet-5 (LeCun, Bottou, Bengio, & Haffner, 1998), AlexNet (Alom et al., 2018), VGG-16 (Simonyan & Zisserman, 2014), and ResNet-50 (He, Zhang, Ren, & Sun, 2016) were chosen. Table 2 shows the comparison of Accuracy of various models with the SSDMNV2 model. In Table 3, F1-score of selected models have been analysed. Lastly the average performance in terms of FPS (Frames per seconds) processed, of the selected models are examined and listed in Table 4. These tables also show the improvement of the proposed work over other models. For calculating FPS mean of 5 runs was taken after running it in real time for 60 s. LeNet-5 is a very simple and classic neural network. It contains 7 layers and is very less demanding computationally although the its scores are the worst in the selected pretrained models, it manages to outperform most models in the performance metric due to its simple architecture. AlexNet is capable of performing greatly and gets decent scores in our testing but these performances require the depth of model, making AlexNet computationally heavy. Hence it could only provide an average FPS of 6.31 on our machine. VGG-16 won the first and second price for object localisation and image classification task in 2014 ILSVRC challenge. Despite getting good results, this model is also very heavy computationally and only provides a meagre 2.76 FPS in real time making it unsuitable for use in embedded devices. ResNet-50 is a deep residual network and is 50 layers deep. While this network provides good accuracy and F1-score, they are not easy to deploy in real-time conditions because they are computationally heavy and provide an average FPS of 2.89 in the testing.

In the end, MobileNetV2 was chosen to be our model for the proposed approach since it outperforms other models in F1-score and Average performance. This makes the proposed approach easy to deploy in real-time even on embedded devices which is not possible with heavy models and to do real-time detection using these models requires good

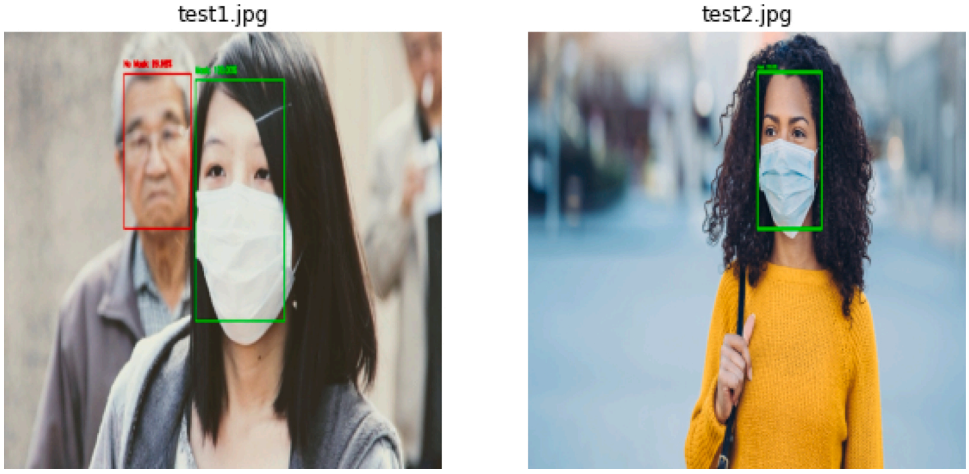

Fig. 13. Predictions on test images.

Table 2
Comparison of accuracy between different models.

Architectures Used	Year	Accuracy (%)	Percentage Improvement
LeNet – 5	1998	84.6	+ 9.37 %
AlexNet	2012	89.2	+ 3.73 %
SSDMNV2 (Proposed Method)	**2020**	**92.64**	**+ 0 %**

Table 3
Comparison of F1 Score between different models.

Architectures Used	Year	F1 Score	Percentage Improvement
LeNet – 5	1998	0.85	+ 9.41 %
AlexNet	2012	0.88	+ 5.68 %
VGG -16	2014	0.92	+ 1.09 %
ResNet – 50	2016	0.91	+ 2.2 %
SSDMNV2 (Proposed Method)	**2020**	**0.93**	**+ 0 %**

Table 4
Comparison of Performance between different models using FPS parameter.

Architectures Used	Year	Average Performance (FPS)	Percentage Improvement
LeNet – 5	1998	14.55	+ 7.97 %
AlexNet	2012	6.31	+ 148.97 %
VGG -16	2014	2.76	+ 469.2 %
ResNet – 50	2016	2.89	+ 443.6 %
SSDMNV2 (Proposed Method)	**2020**	**15.71**	**+ 0 %**

computational power which might make it difficult to play in real life

5. Conclusion

In the proposed face mask detection model named SSDMNV2, both the training and development of the image dataset, which was divided into categories of people having masks and people not having masks have been done successfully. The technique of OpenCV deep neural networks used in this model generated fruitful results. Classification of images was done accurately using the MobilenetV2 image classifier, which is one of the uniqueness of the proposed approach.

Many existing researches faced problematic results, while some were able to generate better accuracy with their dataset. The problem of various wrong predictions have been successfully removed from the model as the dataset used was collected from various other sources and images used in the dataset was cleaned manually to increase the accuracy of the results. Real-world applications are a much more challenging issue for the upcoming future. The SSDMNV2 model should hopefully help the concerned authorities in this great pandemic situation which had largely gained roots in most of the world; other researchers can use the dataset provided in this paper for further advanced models such as those of face recognition, facial landmarks, and facial part detection process.

Declaration of Competing Interest

The authors report no declarations of interest.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., … Ghemawat, S. (2016). *Tensorflow: Large-scale machine learning on heterogeneous distributed systems*. arXiv preprint arXiv:1603.04467.

Ahmed, I., Ahmad, M., Rodrigues, J. J., Jeon, G., & Din, S. (2020). A deep learning-based social distance monitoring framework for COVID-19. *Sustainable Cities and Society*, Article 102571.

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., … Asari, V. K. (2018). *The history began from alexnet: A comprehensive survey on deep learning approaches*. arXiv preprint arXiv:1803.01164.

Anisimov, D., & Khanova, T. (2017). Towards lightweight convolutional neural networks for object detection. August. In *2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS)* (pp. 1–8).

Chen, D., Ren, S., Wei, Y., Cao, X., & Sun, J. (2014). Joint cascade face detection and alignment. September. *European conference on computer vision* (pp. 109–122). Cham: Springer.

Chen, D., Hua, G., Wen, F., & Sun, J. (2016). Supervised transformer network for efficient face detection. October. *European conference on computer vision* (pp. 122–138). Cham: Springer.

Ge, S., Li, J., Ye, Q., & Luo, Z. (2017). Detecting masked faces in the wild with lle-cnns. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2682–2690.

Ge, X. Y., Pu, Y., Liao, C. H., Huang, W. F., Zeng, Q., Zhou, H., … Chen, H. L. (2020). Evaluation of the exposure risk of SARS-CoV-2 in different hospital environment. *Sustainable Cities and Society, 61*, Article 102413.

Ghiasi, G., & Fowlkes, C. C. (2014). Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2385–2392.

Gulcehre, C., Moczulski, M., Denil, M., & Bengio, Y. (2016). Noisy activation functions. June. *International conference on machine learning* (pp. 3059–3068).

Hahnloser, R. H., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., & Seung, H. S. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature, 405*(6789), 947–951.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Huang, C., Ai, H., Li, Y., & Lao, S. (2007). High-performance rotation invariant multiview face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(4), 671–686.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., … Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. November *Proceedings of the 22nd ACM international conference on multimedia*, 675–678.

Jones, P., Viola, P., & Jones, M. (2001). *Rapid object detection using a boosted cascade of simple features*. Charles Rich: University of Rochester.

Jun, B., Choi, I., & Kim, D. (2012). Local transform features and hybridization for accurate face and human detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(6), 1423–1436.

Kim, T. H., Park, D. C., Woo, D. M., Jeong, T., & Min, S. Y. (2011). Multi-class classifier-based adaboost algorithm. October. *International conference on intelligent science and intelligent data engineering* (pp. 122–127). Berlin, Heidelberg: Springer.

Klare, B. F., Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., … Jain, A. K. (2015a). Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1931–1939.

Klare, B. F., Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., … Jain, A. K. (2015b). Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1931–1939.

Kumar, P., Hama, S., Omidvarborna, H., Sharma, A., Sahani, J., Abhijith, K. V., … Tiwari, A. (2020). Temporary reduction in fine particulate matter due to 'anthropogenic emissions switch-off' during COVID-19 lockdown in Indian cities. *Sustainable Cities and Society, 62*, Article 102382.

Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks, 8*(1), 98–113.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324.

Li, H., Lin, Z., Shen, X., Brandt, J., & Hua, G. (2015). A convolutional neural network cascade for face detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5325–5334.

Li, Y., Sun, B., Wu, T., & Wang, Y. (2016). Face detection with end-to-end integration of a convnet and a 3d model. October. *European conference on computer vision* (pp. 420–436). Cham: Springer.

Li, C., Wang, R., Li, J., & Fei, L. (2020). Face detection based on YOLOv3. *Recent trends in intelligent computing, communication and devices* (pp. 277–284). Singapore: Springer.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., … Berg, A. C. (2016). Ssd: Single shot multibox detector. October. *European conference on computer vision* (pp. 21–37). Cham: Springer.

Megahed, N. A., & Ghoneim, E. M. (2020). Antivirus-built environment: Lessons learned from Covid-19 pandemic. *Sustainable Cities and Society, 61*, Article 102350.

Nguyen, h. (2020). Fast object detection framework based on mobilenetv2 architecture and enhanced feature pyramid. *Journal of Theoretical and Applied Information Technology, 98*(05).

Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(7), 971–987.

Opitz, M., Waltner, G., Poier, G., Possegger, H., & Bischof, H. (2016). Grid loss: Detecting occluded faces. October. *European conference on computer vision* (pp. 386–402). Cham: Springer.

Opschoor, J. A., Schwab, C., & Zech, J. (2019). Exponential ReLU DNN expression of holomorphic maps in high dimension. *SAM research report, 2019*.

Rahmani, A. M., & Mirmahaleh, S. Y. H. (2020). Coronavirus disease (COVID-19) prevention and treatment methods and effective parameters: A systematic literature review. *Sustainable Cities and Society*, Article 102568.

Ranjan, R., Patel, V. M., & Chellappa, R. (2017). Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 41*(1), 121–135.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* (pp. 91–99).

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556.

Sun, C., & Zhai, Z. (2020). The efficacy of social distance and ventilation effectiveness in preventing COVID-19 transmission. *Sustainable Cities and Society, 62*, Article 102390.

Velasco-Montero, D., Fernández-Berni, J., Carmona-Galán, R., & Rodríguez-Vázquez, Á. (2018). Performance analysis of real-time DNN inference on Raspberry Pi. May. In *Real-Time Image and Video Processing 2018* (Vol. 10670, p. 106700F). International Society for Optics and Photonics.

Wang, Z., Wang, G., Huang, B., Xiong, Z., Hong, Q., Wu, H., … Chen, H. (2020). *Masked face recognition dataset and application*. arXiv preprint arXiv:2003.09093.

Yang, S., Luo, P., Loy, C. C., & Tang, X. (2016). Wider face: A face detection benchmark. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5525–5533.

Yang, S., Luo, P., Loy, C. C., & Tang, X. (2015). From facial parts responses to face detection: A deep learning approach. *Proceedings of the IEEE International Conference on Computer Vision*, 3676–3684.

Yang, B., Yan, J., Lei, Z., & Li, S. Z. (2015). Fine-grained evaluation on face detection in the wild. May. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)* (Vol. 1, pp. 1–7).

Zhang, X., Zou, Y., & Shi, W. (2017). Dilated convolution neural network with LeakyReLU for environmental sound classification. August. In *2017 22nd International Conference on Digital Signal Processing (DSP)* (pp. 1–5).

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016a). Joint face detection and alignment using multi-task cascaded convolutional networks. *IEEE Signal Processing Letters, 23*(10), 1499–1503.

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016b). Joint face detection and alignment using multi-task cascaded convolutional networks. *IEEE Signal Processing Letters, 23*(10), 1499–1503.

Zhu, C., Zheng, Y., Luu, K., & Savvides, M. (2017). Cms-rcnn: Contextual multi-scale region-based cnn for unconstrained face detection. *Deep learning for biometrics* (pp. 57–79). Cham: Springer.