Human Face Detection with and without masks using Deep-learning

Gajulamandyam Deva Kumar[1], Pavan Kumar Kotapally[2]

[12]UG Student, Dept of ECE, SRM Institute of Science and Technology, Kattankulathur, Chennai, India

---***---

Abstract - *The worldwide health system is being caused havoc by the COVID-19 (Coronavirus) pandemic. According to WHO, wearing a mask in public places is one of the most important barriers or protections for reducing virus transmission. Wearing a facemask will reduce the possibility of virus transmission. Thus, we need physical monitoring to ensure that everyone is wearing a facemask, which is unsafe in this situation. Our key goal is to use deep learning and determine whether or not the person in the picture or video stream is wearing a mask. As a result, therefore, with the help of this project, a person in charge of monitoring individuals will be able to control and give instructions from a distant place. This paper discusses the detection of human face with and without masks by using various libraries of python such as Keras, TensorFlow, open-cv and by training the dataset using MobileNetV2 architecture. It has one residual block with stride 1 and a stride 2 downsizing block with a ReLU6 as an activation function for the hidden layers and SoftMax for the output layer. We have finetuned Mobile-Net by adding a Fully Connected (FC) layer to the base model and achieved a testing accuracy of 99.28%. Using real-time videos, the trained model is tested in a variety of scenarios and is adapted and modified accurately. This study will help in preventing virus transmission and reducing contact with the virus.*

Key Words: WHO, Convolutional Neural Network, Deep Learning, ReLU

1.INTRODUCTION

This document is template. We ask that authors follow some simple guidelines. In essence, we ask you to make your paper look exactly like this document. The easiest way to do this is simply to download the template, and replace(copy-paste) the content with your own material. Number the reference items consecutively in square brackets (e.g. [1]). However, the authors name can be used along with the reference number in the running text. The order of reference in the running text should match with the list of references at the end of the paper. Owing to the global outbreak of the coronavirus pandemic, wearing a facemask in public is now the most important thing. According to the World Health Organization, having a mask reduces the virus's dissemination and can also prevent it from spreading. Officially, 170M cases are registered around the world, with 26.2 million of those cases being registered in India. Covid-19 [1] can only be transmitted by near contacts. As of today, there is no air transmission. As a result, wearing a mask while in close proximity can help to reduce infection transmission. Artificial Intelligence (AI) models built on deep learning will assist us in a variety of ways in combating this. Face detection is a computational technology that uses artificial intelligence (AI) to locate and identify faces in images. Face detection technology[2-4]is used in a variety of fields to include real-time monitoring and tracking of individuals, like defense, law enforcement, biometrics, personal safety and entertainment. Face detection has evolved from basic computer vision approaches to developments in deep learning and associated technology, resulting in ongoing efficiency enhancements. It now serves as a crucial first step in a variety of important applications as mentioned above. Using this technology, we can track wide numbers of individuals with and without masks.

Therefore, various researchers in the past have proposed a lot of experiments and deep-learning models to fight this situation by leveraging technology. But the collection of the dataset and the model's accuracy have been major issues. We can accomplish this job in a variety of ways, but the first step in all of them is to detect the face in all circumstances.

HaarCascade classifier [5] is one of the popular methods for detecting the face. Incorporating this algorithm to detect the presence of facemask by training the model with a training size of 0.9 and achieved an accuracy of 94.58%. But the Haar Cascade classifier is known to perform well only in good lighting conditions as it localizes facial landmarks within the frame.

Further SSDM V2 [6] was proposed for facemask detection using deep neural networks. This is built on the 'Single Shot Multibox Detector' (SSD) which uses the 'ResNet-10' architecture as its core. But due to increased complexity of its architecture, it takes longer time for facemask detection and makes it difficult for real time application. With this the accuracy obtained was 92.64%.

The Mobile Net V2 [7] architecture is introduced for facemask detection which can outperform all the Simulations but they could achieve only an accuracy of 90.52% with a train size of 0.75. The detection of faces with different colored masks was also not discussed and due to the low accuracy, the false positives and true negatives are high which leads to more errors in prediction.

In this work, a facemask detector is built using transfer learning with Mobile Net V2 architecture as it is highly efficient and can be easily deployed/integrated in embedded devices [8] such as NVIDIA Jetson Nano, Google Coral etc. The imported Mobile Net V2 layers are loaded with pre-trained ImageNet weights. We have finetuned Mobile Net architecture by adding additional layers to improve the overall performance of model and replacing head layer of Mobile Net with FC layer. The parameters of the neural network have been tuned so that the weights of Mobile Net layers won't be updated during back-propagation whereas the additional layers' weights will be changed as per gradient descent. Training our dataset with this revamped architecture, generated a robust and accurate model.

The following is a breakdown of the paper's structure. In section 2, the overall design of the model is described along with the Architecture. The model's performance is examined in Section 3. The results and comparison have been discussed in Section 4, and the conclusion is addressed in Section 5.

2 Architecture of Proposed Model

The face mask detection phase flow is depicted in fig.1 The work is divided into two phases, which cover the whole process from data collection to implementation. Images of masked and unmasked faces are used as data, which are converted to NumPy arrays for training. The training is carried out with the help of CNN and the Mobile Net V2 layers. The proposed model's architecture is divided into two stages, as shown.

1. Stage 1: Training
2. Stage 2: Applying the face mask detector

The method is divided into four steps, as seen in fig Stage 1. The masked and unmasked images are collected first, followed by pixel scaling, resizing, and transformation of image to NumPy arrays in data preprocessing. Finally, the dataset is separated into training and testing data, with training data being 80% and testing data including 20% of the photos. The training data is then transferred for further training, and the model is tested with the experimental data. The Model with the highest accuracy is saved as a h5 file after training.

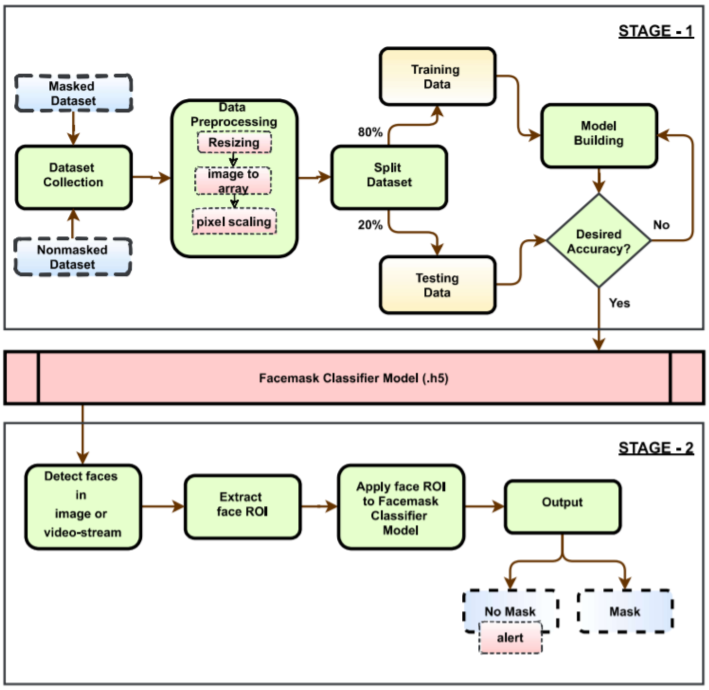


Fig-1: Architecture of the Model

2.1 Dataset

There are only a few datasets available for identifying face masks. The majority of them are either digitally generated and do not correctly reflect the natural world, or there's a lot of noise in the dataset. As a result, it took some time and work to find the right dataset for the model. A mix of open-source datasets and images were used to train the model in each process, including data from Mikolaj Witkowski's Medical Mask Dataset on Kaggle and the Prajna Bhandary dataset on PyImageSearch. This diversification in datasets enabled our model to perform well in all scenarios. We have a total of 3856 images after combining all of the datasets available. These are split into training and testing datasets as 80% and 20% respectively with 1546 training images and 394 testing images for No Mask. Similarly, total mask pictures are 1916, with 1520 for training and 396 for testing.

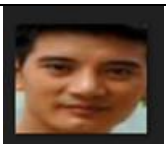
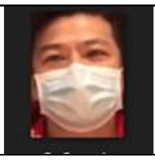
Classes	Sample Image	Total Images	Training Images	Testing Images
No Mask		1940	1546	394
Mask		1916	1520	396

Table-1: Distribution of dataset for training, testing along with sample image

2.2 Data Preprocessing

All the images within the dataset are pre-processed as per Mobile Net architecture input requirements. First step is to resize images to 224x224 pixels. Second step is the conversion of the image to array format. Final step is scaling the pixel intensities of the image. These are done by *using the preprocess input ()* function *from the mobilenet_v2 module.*

2.3 Model Building

At this stage, the dataset is split for training and testing as 80% and 20% respectively. Now we fine tune the Mobile Net model by loading it with pre-trained ImageNet weights which is a de facto standard for image classification. It helps the model to converge in less epochs. The head of the model is removed and replaced with a new Fully Connected (FC) layer with ReLU and SoftMax activation functions. During back-propagation, only the weights of the head layer are updated and not that of the base layer. Now we build our model with training data, Adam Optimizer, Binary Cross Entropy loss function since this is a binary classification problem.

2.3.1 Mobile Net V2

MobileNetV2 is a convolutional neural network that is designed to function on mobile devices. There are other models, however MobileNetV2 has a lower calculation capacity and consumes fewer computing resources. This makes it a great match for deploying it quickly on mobile devices, installed frameworks, and PCs with poor computational productivity or no GPU, all without compromising performance. MobileNetV2, which is primarily focused on real-time computer vision, makes use of the Open-Source Computer Vision Library. In Fig 2 The layers of Mobile Net V2 are shown following with a Fig 3 that shows the working in strides.

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Fig-2: Layers of Mobile Net V2

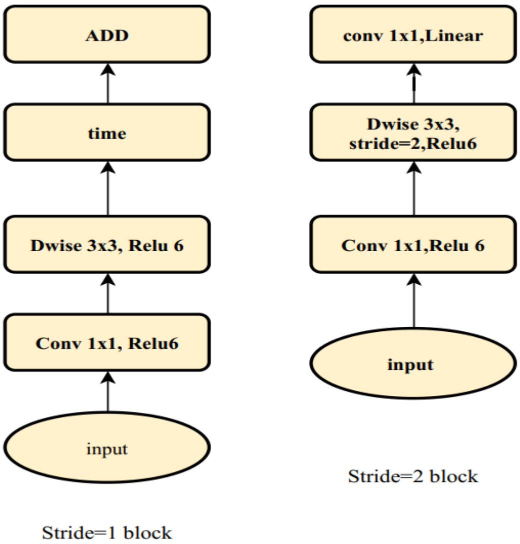


Fig-3: Strides of Mobile Net V2

2.4 Saving the Model

After the optimum accuracy is achieved, the model is saved in HDF5 format. The HDF5 file format is famous for its hierarchical system, versatility, and the ability to store arbitrary metadata with each object. The datasets in HDF can be either fixed or flexible sized. As a result, appending data to a huge dataset without needing to generate a new copy is easy. Furthermore, since HDF5 is a structured format with libraries for almost any language, it's simple to share your on-disk data between, say, R, C, Python, MATLAB and Fortran.

2.5 Face ROI extraction

After the model is saved, it is loaded and each frame of a video is sent into the model for classification of presence of facemask which we have trained earlier. The image is pre-processed like the training images are done and then face detection is performed on it to localize all the faces in the image. Later, we calculate bounding box values for each face if the confidence of prediction is more than threshold confidence and make sure that the box falls within the boundary of the face as shown in the figure below. We extract this face ROI by NumPy slicing and this ROI is sent for model prediction and labelled as 'Mask' or 'No Mask' in an appropriate color rectangle box around the image and displayed on screen.

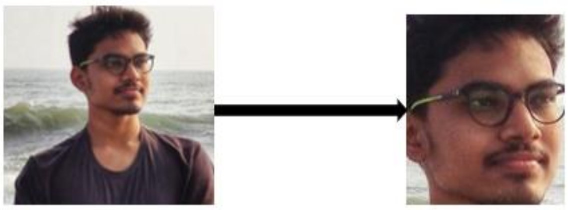


Fig-4: Face ROI Extraction

3 Performance analysis

With a batch size of 34 and 10 epochs we trained our model with a total of 3900 photos (from both classes) as stated in the proposed method. We achieved a training accuracy of 99.56 percent and a validation accuracy of 99.28 percent after training. This is a good-fit model since both are almost identical.

```
Epoch 1/10
34/34 [==============================] - 26s 673ms/step - loss: 0.8823 - accuracy: 0.5642 - val_loss: 0.2508 - val_accuracy: 0.9674
Epoch 2/10
34/34 [==============================] - 21s 627ms/step - loss: 0.2632 - accuracy: 0.9510 - val_loss: 0.1137 - val_accuracy: 0.9855
Epoch 3/10
34/34 [==============================] - 26s 754ms/step - loss: 0.1267 - accuracy: 0.9828 - val_loss: 0.0725 - val_accuracy: 0.9855
Epoch 4/10
34/34 [==============================] - 21s 620ms/step - loss: 0.0917 - accuracy: 0.9853 - val_loss: 0.0528 - val_accuracy: 0.9891
Epoch 5/10
34/34 [==============================] - 21s 612ms/step - loss: 0.0618 - accuracy: 0.9913 - val_loss: 0.0434 - val_accuracy: 0.9891
Epoch 6/10
34/34 [==============================] - 21s 613ms/step - loss: 0.0621 - accuracy: 0.9824 - val_loss: 0.0356 - val_accuracy: 0.9928
Epoch 7/10
34/34 [==============================] - 21s 620ms/step - loss: 0.0493 - accuracy: 0.9870 - val_loss: 0.0311 - val_accuracy: 0.9891
Epoch 8/10
34/34 [==============================] - 21s 625ms/step - loss: 0.0356 - accuracy: 0.9922 - val_loss: 0.0277 - val_accuracy: 0.9891
Epoch 9/10
34/34 [==============================] - 21s 617ms/step - loss: 0.0372 - accuracy: 0.9921 - val_loss: 0.0252 - val_accuracy: 0.9928
Epoch 10/10
34/34 [==============================] - 21s 623ms/step - loss: 0.0265 - accuracy: 0.9956 - val_loss: 0.0227 - val_accuracy: 0.9928
```

Fig-5 Model Training

In the below figures, we can see the graph plotted between accuracy and no. of epochs using matplotlib library. As can be seen in fig.6, a graph has been plotted between training and testing accuracy, and it is apparent that the accuracy increases as the number of epochs increases. And in fig.7 graph has been plotted between training and validation loss and it is apparent that the loss is decreasing with increase in epochs.

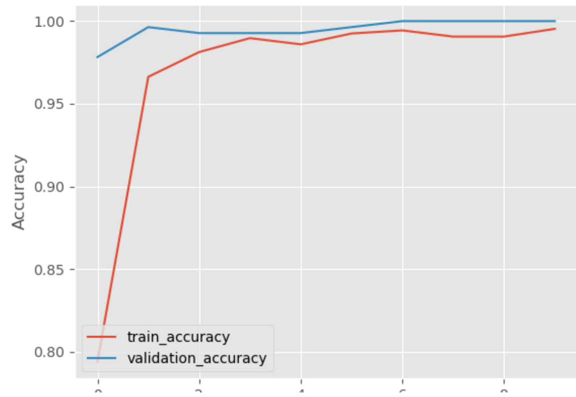


Fig-6: Accuracy Vs Epochs

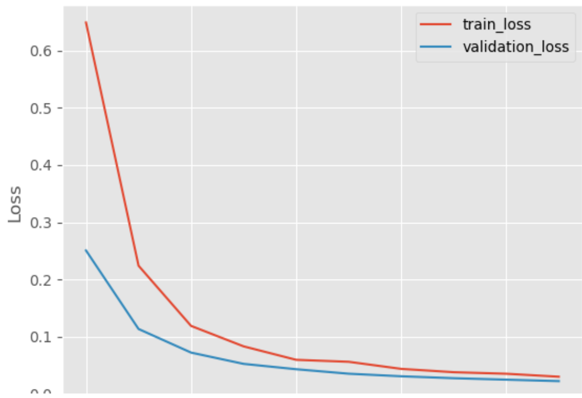


Fig-7: Loss vs Epochs

4 Results and Discussion

On an Anaconda platform with a 16GB ram computer, the suggested face mask detection algorithm was tested. The results are shown below.

The red rectangle box in the fig indicates that there is no mask or that it is being worn improperly, while the green rectangular box shows that the individual in the photo is wearing the mask and is safe. The probability of categorization is also represented by the number followed by prediction.

4.1 Good Lighting Conditions

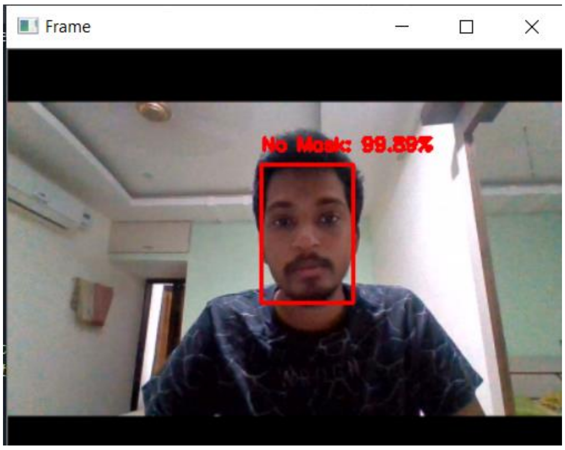


Fig -8(a): No Mask

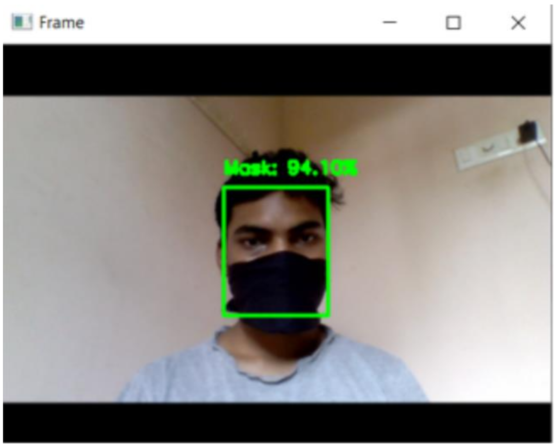


Fig-8(b): Mask

4.2 Low Lighting Conditions

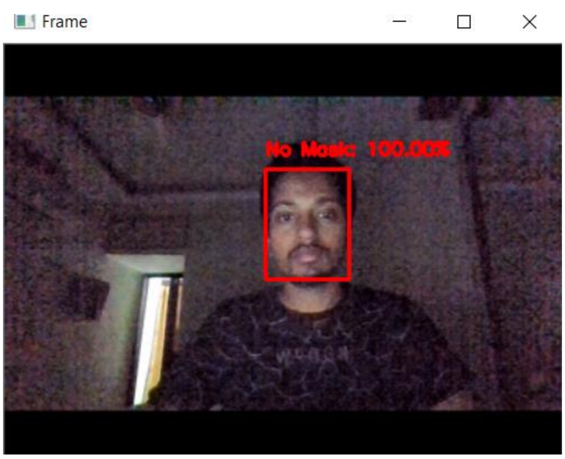


Fig-9(a): No Mask

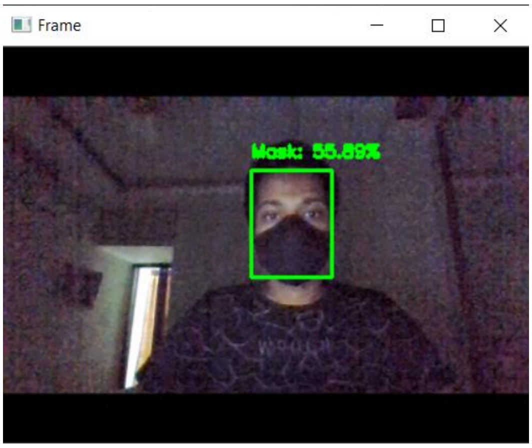


Fig-9(b): Mask

4.3 Multiple Faces in a Frame

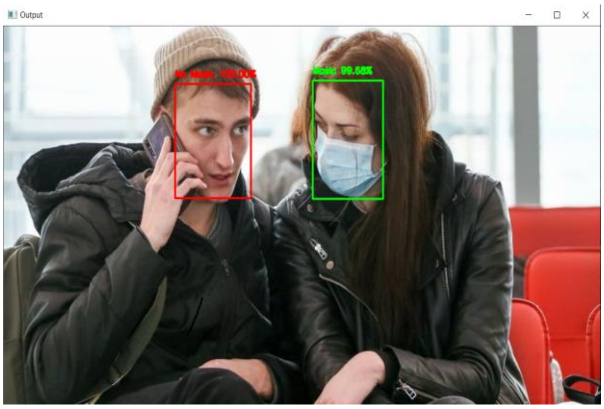


Fig-10(a)

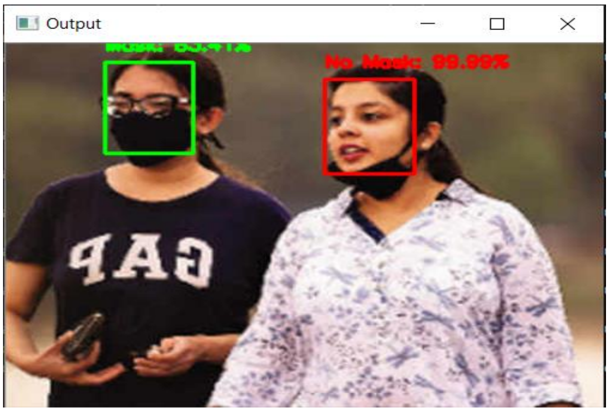


Fig-10(b)

The model architecture, train size and the accuracy of other prominent models are compared in the below Table 2 as these are considered as a tool to indicate the performance of the model.

From the Table-2, it is observed that the proposed model outperformed the existing models due to the fine-tuning of Mobile Net architecture. This gave us the validation accuracy of 99.28% with a train size of 0.8.

Model Architecture	Year	Train Size	Accuracy
Mobile Net and SSDMNV2	2020	0.80	92.64%
CNN using Mobile Net V2	2020	0.75	90.52%
CNN using Haar-Cascade	2021	0.90	94.58%
Mobile Net V2 (Proposed Method)	2021	0.80	99.28%

Table-2: Model Comparison

5 Conclusion

The paper presents a method for a smart world that can help minimize the transmission of coronavirus by alerting authorities when someone is not wearing a facial mask. This model that has Mobile Net V2 as its backbone has shown that there is an improvement in the performance compared to the SSDMNV2 and Haar-Cascade models. This model has 99.28% accuracy and has been evaluated in a variety of environments, making it extremely promising for future deployment in public areas. Furthermore, integration of this model with temperature sensors can be done for auto temperature detection for entry.

6 References

[1] R. Kumari et al., "Analysis and predictions of spread, recovery, and death caused by COVID-19 in India," in Big Data Mining and Analytics, vol. 4, no. 2, pp. 65-75, June 2021,doi:10.26599/BDMA.2020.9020013.

[2] G. Singh and A. K. Goel, "Face Detection and Recognition System using Digital Image Processing," 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 2020, pp. 348-352, doi:10.1109/ICIMIA48430.2020.9074838.

[3] J. Chatrath, P. Gupta, P. Ahuja, A. Goel and S. M. Arora, "Real time human face detection and tracking," 2014 International Conference on Signal Processing and Integrated Networks (SPIN), 2014, pp. 705-710, doi: 10.1109/SPIN.2014.6777046.

[4] R. Qi, R.-S. Jia, Q.-C. Mao, H.-M. Sun, and L.-Q. Zuo,"Face detection method based on cascaded convolutional networks," IEEE Access, vol. 7, pp. 110740-110748, 2019.

[5]. A. Das, M. Wasif Ansari, and R. Basak, "Covid-19 face mask detection using tensorflow, keras and opencv," in 2020 IEEE 17th India Council International Conference (INDICON),pp.1-5,2020.

[6]. P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth, "Ssdm V2 : A real time dnn-based face mask detection system using single shot multibox detector and mobilenetv2," Sustainable cities and society, vol.66p.102692,2021.

[7]. S. A. Sanjaya and S. Adi Rakhmawan, "Face mask detection using mobilenetv2 in the era of covid-19 pandemic," in 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), pp. 1-5, 2020.

[8]. Zhao, Hong, Xi-Jun Liang, and Peng Yang. "Research on face recognition based on embedded system." Mathematical Problems in Engineering 2013 (2013).

[9]. T. Singhal, "A review of coronavirus disease-2019 (covid-19)," The indian journal of pediatrics, vol. 87, no. 4, pp.281-286,2020.

[10]. M. A. Iqbal, Z. Wang, Z. A. Ali, and S. Riaz, "Automatic fish species classification using deep convolutional neural networks," Wireless Personal Communications, pp. 1{11, 2019.

[11]. H. Kaur and N. Jindal, "Deep convolutional neural network for graphics forgery detection in video," Wireless PersonalCommunications,pp.1-19,2020.

[12]. M. M. Rahman, M. M. H. Manik, M. M. Islam, S. Mahmud, and J.-H. Kim," An automated system to limit covid-19 using facial mask detection in smart city network," in 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), pp. 1{5, 2020.

[14]. S. Sen and K. Sawant, "Face mask detection for covid 19 pandemic using pytorch in deep learning," in IOP Conference Series: Materials Science and Engineering, vol. 1070,p.012061,IOPPublishing,2021.

[15]. H. Zhang, A. Jolfaei, and M. Alazab, "A face emotion recognition method using convolutional neural network and image edge computing," IEEE Access, vol. 7, pp. 159081-159089, 2019.