Novelties in Intelligent Digital Systems
C. Frasson et al. (Eds.)

doi:10.3233/FAIA210102

Real-Time Face Mask Detector Using Convolutional Neural Networks Amidst COVID-19 Pandemic

Efstratios KONTELLIS[1], Christos TROUSSAS, Akrivi KROUSKA, Cleo SGOUROPOULOU
Department of Informatics and Computer Engineering
University of West Attica
Greece

Abstract. The COVID-19 pandemic provoked many changes in our everyday life. For instance, wearing protective face masks has become a new norm and is an essential measure, having been imposed by countries worldwide. As such, during these times, people must wear masks to enter buildings. In view of this compelling need, the objective of this paper is to create a real-time face mask detector that uses image recognition technology to identify: (i) if it can detect a human face in a video stream and (ii) if the human face, which was detected, was wearing an object that it looked like a face mask and if it was properly worn. Our face mask detection model is using OpenCV Deep Neural Network (DNN), TensorFlow and MobileNetV2 architecture as an image classifier and after training, achieved 99.64% of accuracy.

Keywords. Virus Protection, COVID-19, Face Detection, Deep Learning, Convolutional Neural Network, Computer Vision, MobileNetV2

1. Introduction

The COVID-19 pandemic has forced many countries around the world to initiate new rules for face mask-wearing and social distancing. Also, governments have started working on new strategies [1] to stop the spreading of the virus by forcing people to wear face masks. The goal of wearing face masks is to lessen the transmission and spreading rate. As COVID-19 is transmitted [2] through air drops and close contact, more and more people are concerned about their health and public health is considered as top priority for governments.

According to World Health Organization (WHO) directions [3], the wearing of face masks appears as a solution for limiting the spread of COVID-19. That is the reason why face mask detection has become a trending application [4] due to the pandemic, to monitor that people are following this basic safety principle. Artificial Intelligence (AI) [5] based on Machine Learning and Deep Learning can help humanity fight against COVID-19 in many ways. For example, AI can be used to track and to predict how the disease will spread over time and space. Face mask detection means to identify whether a person is wearing a mask or not. There are two steps that should be implemented to achieve this. The first step is to detect human faces in a video stream and the second step

[1] Corresponding Author; E-mail: stratis96@outlook.com.

is to identify if the human face that was detected was wearing an object that it looked like a face mask and was properly worn. Face detection is one of the most popular applications of object detection and can be used in many areas like security, face emotions, biometrics, law enforcement and more.

In view of the above, the main objective of the research is to detect whether someone is wearing his face mask correctly to protect himself and the public health from COVID-19 spreading. To perform this task, a large dataset of masked and unmasked faces is necessary for training deep learning models towards detecting people wearing masks and those not wearing masks. However, at the moment, there are not large datasets available of masked faces and because of that, we applied a method called "Data Augmentation" in our training data. Our face mask detection model is using OpenCV Deep Neural Network (DNN), TensorFlow and MobileNetV2 [6] architecture as an image classifier. MobileNetV2 consists of Convolutional Neural Networks [7-11], which are ideal for image classification. The model is trained for 15 epochs which maintains a trade-off between accuracy and chances of overfitting. Finally, thanks to our accurate dataset, our trained model reaches 99,64% of accuracy.

2. Related Work

Object detection from an image is probably the deepest aspect of computer vision due to widely used in many cases such as, face recognition, neurobiology, robotic navigation etc. There has been supervised and unsupervised based learn in the field of computer vision to outfit the work of object detection in an image. This section conducts the recent academic papers for applying representative works related to object detection based on deep learning for the medical face mask. Most of face mask detection methods focus on face construction and face detection based on traditional machine learning techniques. In this paper, our focus is on detecting and finding the human who is wearing a face mask or not, to help in lessening the spreading of COVID-19.

In face detection method, a face is detected from an image that has several attributes in it. According to [12], research into face detection requires expression recognition, face tracking and pose estimation. Given a solidary 2D image, the challenge is to identify the face from the picture. Face detection is a difficult errand because faces change in color, size, shape, etc. and they are not immutable. In [13], the authors present occlusive face detection coming with two major challenges: (1) unavailability of sizably voluminous datasets containing both masked and unmasked faces, and (2) exclusion of facial expression in the covered area. According to the work reported in [14], convolutional neural networks (CNNs) in computer vision comes with a strict constraint regarding the size of the input image. The prevalent practice reconfigures the images before fitting them into the network to surmount the inhibition.

In [15], the authors presented a system for face mask detection that attains accuracy of 95.77%. They used TensorFlow, OpenCV and Data augmentation to reach this result. Another research [16] presented a DPM-based face mask detector using around 30,000 faces divided into masked and unmasked. This approach achieved an exceptional accuracy of 97.14 %. In [17], the authors presented a system for detecting the presence of a compulsory medical mask in public places. The proposed system achieved 97.1% accuracy.

Analyzing the related literature and towards advancing the literature, this paper presents a face mask detection model which has been developed using deep neural network modules from OpenCV and TensorFlow, containing a Single Shot Multibox Detector object detection model. Fine-tuned MobileNetV2 classifier has been used, which has been an improvement over MobileNetV1 architecture classifier. In conclusion, MobileNetV2 architecture (Table 1) is made up of 17, 3 x 3 convolutional layers in a row accompanied by a 1 x 1 convolution, an average layer of max pooling and a layer of classification. Also, thanks to the accurate dataset of masked and unmasked faces, our proposed method attains accuracy up to 99.64%.

Table 1. MobileNetV2 architecture

Input	Operator	t	c	n	s
224^2 x 3	conv2d	-	32	1	2
112^2 x 32	bottleneck	1	16	1	1
112^2 x 16	bottleneck	6	24	2	2
56^2 x 24	bottleneck	6	32	3	2
28^2 x 32	bottleneck	6	64	4	2
14^2 x 64	bottleneck	6	96	3	1
14^2 x 96	bottleneck	6	160	3	2
7^2 x 160	bottleneck	6	320	1	1
7^2 x 320	conv2d 1x1	-	1280	1	1
7^2 x 1280	avgpool 7x7	-	-	1	-
1 x 1 x 1280	conv2d 1x1	-	k	-	

where t: expansion factor, c: number of output channels, n: repeating number, s: stride. 3×3 kernels are used for spatial convolution.

3. Proposed Method

To predict whether a person has worn a face mask correctly, there are two phases as mentioned above. Firstly, an accurate face detection model is required to detect faces in real-time. Details about this model have been discussed above in Section 3.4. Secondly, a proper database can help us to train our model to reach high accuracy. Furthermore, to raise the accuracy of mask detection without being too resource-heavy, our classifier uses a pre-trained model MobileNetV2 to recognize whether a person is wearing a mask or not. Devices like smartphones and the Raspberry Pi are benefited from this approach. The methodology used in this paper is depicted in Figure 1.

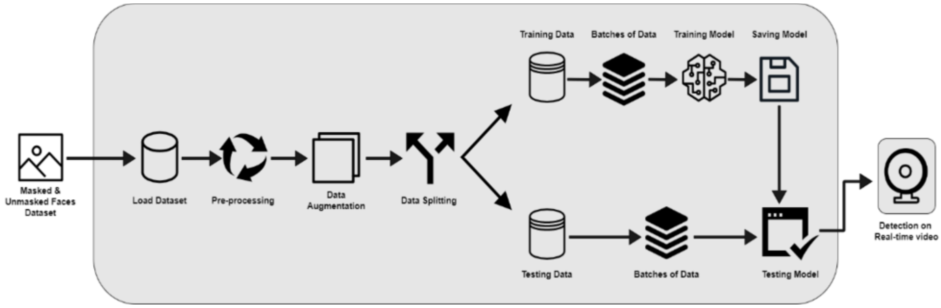

Figure 1. Flow Diagram of our model.

3.1. Dataset used

Until today, there are only a few images with masked people available for the detection of face masks [18]. Some of them are artificially created, which do not represent the real world accurately, while others are blurred with low resolution. By choosing the right dataset that contains the right images require a little effort [19]. The dataset used for training the model consists of 4000 total images out of which 2000 are of masked faces having the label "with_mask" and 2000 are of unmasked faces having the label "without_mask" [20]. All the images are actual images extracted from Kaggle datasets and cover diverse races i.e., Asian, Southern Europeans etc. As a result, the distribution between the two categories is visualized in Figure 2.

In our approach, we have dedicated 80% of the dataset as the training data and the remaining 20% as the testing data, which makes the split ratio as 0.8:0.2 of train to test set.

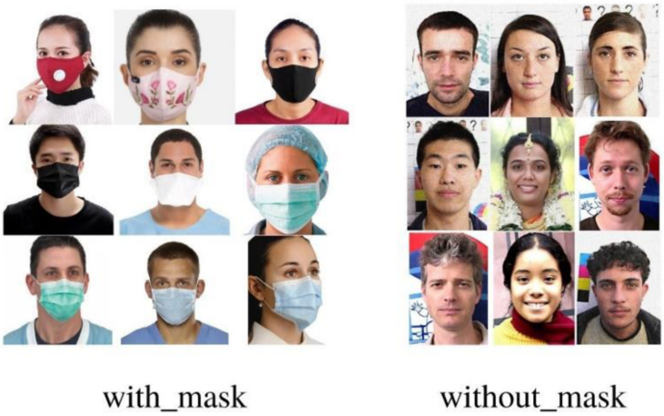

Figure 2. Samples of our dataset.

3.2. Pre-processing

There are four steps in the pre-processing phase which are resizing image size, converting the image into an array, pre-processing input using MobileNetV2 and the last is performing hot encoding on labels. But, before that, we should evaluate our dataset. Since a good dataset dictates the accuracy of the trained model, it is a vital part to manually remove corrupt images and repetitions. Cleaning, identifying, and correcting errors in a dataset removes adverse effects from any predictive model.

This phase explains the procedure of pre-processing the data and then training on data. The resizing image is a critical step in computer vision due to effectiveness of training models. The smaller the image size is, the better the model will perform. In this research, all the images are resized into 224x224 pixels.

The next step is to process all the images from the dataset into a NumPy array for faster calculation. After this, the process of data augmentation is applied to increase the training data as we have been discussed above in Section 3.3. Additionally, the images are used to pre-process input using MobileNetV2.

The last step in this phase is performing hot encoding on labels because a lot of machine learning algorithms cannot operate on data labeling directly. They require all the input and output labels to be numeric.

3.3. Data augmentation

For the training of our model, an enormous quantity of data is required to perform training effectively. Due to the non-availability of an adequate amount of data, the method of data augmentation [21] is used to solve this issue because neural networks are only as good as the data, we feed them. In this technique, methods like flipping, rotation, scaling, cropping, and translate the picture are used for generating numerous versions of a similar picture.

3.4. Face detection using OpenCV

Before detecting whether a person is wearing his face mask or not, we have to use a model called "Caffemodel" [22] to detect human faces in real-time using OpenCV. This model is a deep learning framework developed as an even faster and efficient alternative as compared to other object detection methods, i.e., Haar Cascades, and is created and managed by Berkeley AI Research (BAIR) and community contributors. Wearing a facial mask means that an object that it looks like a mask, is hiding some facial characteristics, i.e., nose, mouth etc. This means that some algorithms and pre-trained models may not be able to detect human faces with facial masks. For this reason, because the Caffemodel relies on deep learning neural networks, it makes it an ideal choice for masked or unmasked face detection. To implement 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 and the location of their bounding boxes. These video frame outputs are then used as input for the face mask classifier. Using this approach to detect masked or unmasked faces allows for real-time detection without much resource usage and it can also detect human faces in different orientations i.e., top, bottom, right and left with good accuracy.

3.5. Classification of images using MobileNetV2

MobileNetV2 is a Deep Neural Network based on Convolutional Neural Network architecture that has been deployed for the classification problem. Also, it aims to run very efficiently on mobile devices and it can be used as a basic image classifier or as a feature extractor. In our case, pretrained weights of ImageNet were loaded from TensorFlow and then, the base layers of our network are frozen to avoid impairment of already learned features. Subsequently new trainable layers are added and these are trained on collected dataset to classify features of a masked face from an unmasked face. Then, the model is fine-tuned, and then the weights are saved. By using pre-trained models helps to avoid needless computational costs and helps in taking advantage of already biased weights without losing any learned features (Figure 3).

4. Training

After the pre-processing phase, the model needs to be trained using our dataset and then to be tested against a different dataset. To do this, the data is split into two batches, which 80% of them is training data and the rest 20% is testing data. Each batch is containing both of masked and unmasked faces. The final step before training the model, is to set the hyperparameters. In machine learning, a hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters are derived via training. Initially, we trained with different values of hyperparameters by changing one and keeping the others constant and noted down the results in each case. By this way, we could select the hyperparameters that produced better performance though evaluation metrics. We have chosen the hyperparameters as follows: initial learning rate is taken as 0.0001, batch size is taken to be 32 and number of epochs as 15. The model is trained for 15 epochs which maintains a trade-off between accuracy and chances of overfitting.

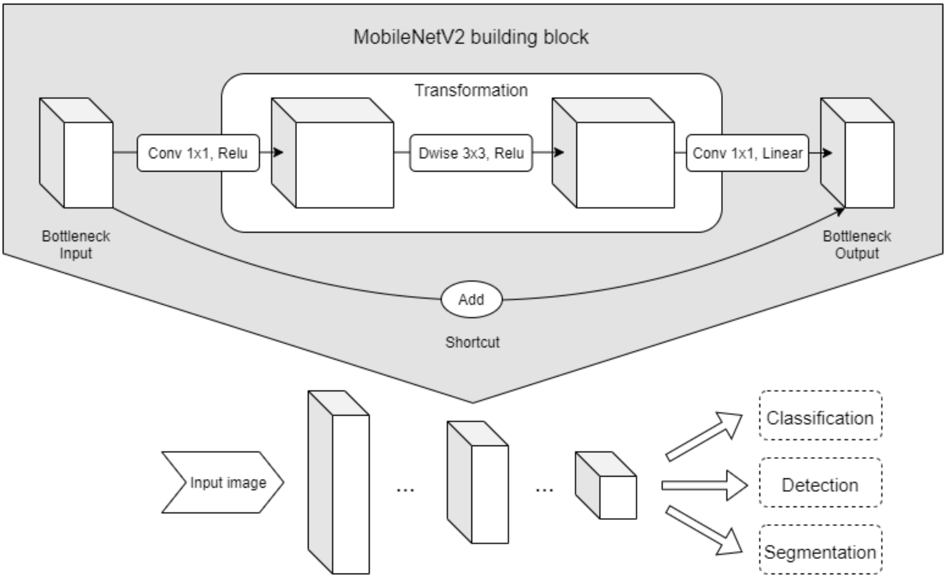

Figure 3. MobileNetV2 Convolutional Blocks.

5. Experimental Results

Our proposed model has simulated using Python and TensorFlow. The metrics selected for evaluation of our model are explained below.

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

$$\text{Precision} = \frac{\text{Tp}}{(\text{Tp} + \text{Fp})}$$

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

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

where Tp is the count of True positive samples, Tn is the count of True negative samples, Fp is the count of False positive samples and Fn is the count of False negative samples. Also, “Precision” is the number of correct predictions over how many occurrences of that class were in the test dataset and “Recall” is the actual true positives over how many times the classifier predicted that class. The “f1 score” can be interpreted as a weighted average of the precision and recall, where an f1 score reaches its best value at 1 and worst score at 0.

Corresponding to our dataset, the method attains accuracy up to 99,64% (Figure 4).

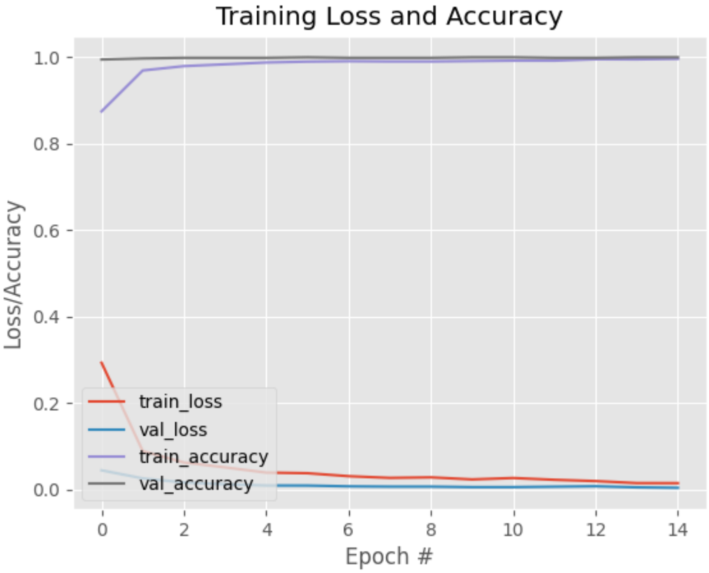

Figure 4. Training Results.

Table 2. Classification report

	Precision	Recall	f1 score	Support
with_mask	1.00	1.00	1.00	388
without_mask	1.00	1.00	1.00	414
accuracy			1.00	802
Macro average	1.00	1.00	1.00	802
Weighted average	1.00	1.00	1.00	802

Our real-time face mask detection model follows the procedure below. First of all, the video stream is scanning each frame to detect human faces. If a face is detected, it proceeds to the next process. From detected frames containing human faces, reprocessing will be carried out including resizing the image size, converting into an array and pre-processing input using MobileNetV2. The next step is predicting input data from the trained model. Afterward, the video frame will be labeled that the person is wearing or a mask or not. A testing example is depicted in Figure 5.

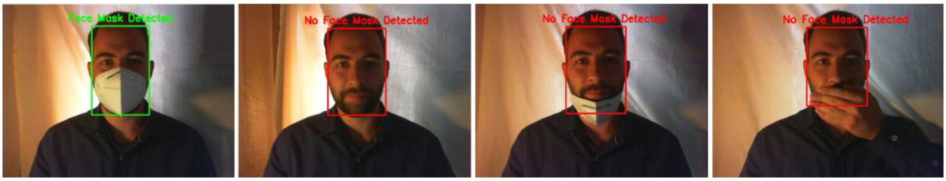

Figure 5. Real-time face mask detection results.

6. Conclusion

Due to the outbreak of COVID-19 pandemic, all the countries worldwide have imposed people to wear a face mask as a compulsory measure to prevent the spread of coronavirus. Manual observation of the face mask in crowded places is a critical task. To address this compelling need, this paper presents a real-time face mask detector. In this approach, we trained our model using MobileNetV2 with masked and unmasked faces. Furthermore, thanks to our accurate dataset, our model understands whether a face mask is properly worn. Later, we also applied this model into a real-time video to check our model's fps performance and accuracy. As future work, different types of face masks should be taken into consideration to expand the types of detecting objects and a hybrid model using deep learning techniques will be created to further investigate its accuracy in face masks detection.

References

[1] M. Loey, G. Manogaran, M. H. N. Taha, N. E. M. Khalifa, Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection, *Sustainable Cities and Society* **65** (2021).

[2] I. B. Venkateswarlu, J. Kakarla and S. Prakash, Face mask detection using MobileNet and Global Pooling Block, 2020 IEEE 4th Conference on Information & Communication Technology (2020), 1-5.

[3] Y. Chen et al., Face Mask Assistant: Detection of Face Mask Service Stage Based on Mobile Phone, *IEEE Sensors Journal* **21** (2021), 11084-11093.

[4] S. A. Sanjaya and S. Adi Rakhmawan, Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic, *2020 IEEE International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy* (2020), 1-5.

[5] M. Loey, G. Manogaran, M. H. N. Taha, N. E. M. Khalifa, A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic, *Measurement* **167** (2021).

[6] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, MobileNetV2: Inverted Residuals and Linear Bottlenecks, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018) 4510-4520.

[7] F. Giannakas, C. Troussas, I. Voyiatzis, C. Sgouropoulou, A deep learning classification framework for early prediction of team-based academic performance, *Applied Soft Computing* **106** (2021).

[8] C. Troussas, F. Giannakas, C. Sgouropoulou, I. Voyiatzis, Collaborative activities recommendation based on students' collaborative learning styles using ANN and WSM, *Interactive Learning Environments* (2020). DOI: 10.1080/10494820.2020.1761835.

[9] C. Troussas, A. Krouska, M. Virvou, A multilayer inference engine for individualized tutoring model: adapting learning material and its granularity, Neural Computing and Applications (2021). https://doi.org/10.1007/s00521-021-05740-1.

[10] A. Krouska, C. Troussas, M. Virvou, *Deep Learning for Twitter Sentiment Analysis: The Effect of Pre-trained Word Embedding*, Tsihrintzis G., Jain L. (eds) Machine Learning Paradigms, Learning and Analytics in Intelligent Systems 18, Springer, Cham, 2020. https://doi.org/10.1007/978-3-030-49724-8_5.

[11] C. Troussas, C. Sgouropoulou, *Innovative Trends in Personalized Software Engineering and Information Systems - The Case of Intelligent and Adaptive E-learning Systems",* Frontiers in Artificial Intelligence and Applications, IOS Press, 324, Netherlands, 2020.

[12] D. Meena, R. Sharan, An approach to face detection and recognition, *2016 IEEE International Conference on Recent Advances and Innovations in Engineering* (2016), 1-6.

[13] S. Ge, J. Li, Q. Ye, Z. Luo, Detecting Masked Faces in the Wild with LLE-CNNs, *2017 IEEE Conference on Computer Vision and Pattern Recognition* (2017), 426-434.

[14] S. Ghosh, N. Das, M. Nasipuri, Reshaping inputs for convolutional neural network: Some common and uncommon methods, *Pattern Recognition* **93** (2019).

[15] A. Das, M. Wasif Ansari, R. Basak, Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV, *2020 IEEE 17th India Council International Conference* (2020), 1-5.

[16] K. Zhang, Z. Zhang, Z. Li, Y. Qiao, Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks, *IEEE Signal Processing Letters* **23** (2016), 1499-1503.

[17] A. Oumina, N. El Makhfi and M. Hamdi, Control The COVID-19 Pandemic: Face Mask Detection Using Transfer Learning, *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science* (2020), 1-5.

[18] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, J. Hemanth, SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2, *Sustainable Cities and Society* **66** (2021).

[19] M. Jiang, X. Fan, RetinaMask: A Face Mask detector, Hong Kong, 2020. https://arxiv.org/abs/2005.03950.

[20] M. R. Bhuiyan, S. A. Khushbu, M. S. Islam, A Deep Learning Based Assistive System to Classify COVID-19 Face Mask for Human Safety with YOLOv3, *2020 11th IEEE International Conference on Computing, Communication and Networking Technologies* (2020), 1-5.

[21] N. C. Ristea, R. T. Ionescu, Are you wearing a mask? Improving mask detection from speech using augmentation by cycle-consistent GANs, Romania, 2020. https://arxiv.org/abs/2006.10147.

[22] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding, UC Berkeley EECS, 2014. https://arxiv.org/abs/1408.5093.