Grenze International Journal of Engineering and Technology, Jan Issue

Designing a Model to Monitor the Commitment of People to Wear a Face Mask, Even if an Image of their Real Face is Printed on it, using Deep Learning

Ansam Ghanom [1] and Dr. Mohamad-Bassam Kurdy [2]
[1]Department of Web Sciences, Syrian Virtual University, Damascus, Syria
Email: ansam_97187@svuonline.org
[2]Department of Artificial Intelligence, Syrian Virtual University, Damascus, Syria
ORCID: 0000-0002-3596-3320

***Abstract*—The novel coronavirus (COVID-19) has rapidly affected our daily lives, which led to stopping the wheel of life for millions of people around the world and disrupting trade and global movements.**
The most important measure to limit the spread of this virus is to wear a mask on the face, so this procedure has become a new normal. In the near future, many public service providers will require customers to wear masks properly to benefit from their services. Therefore, the discovery of the face mask has become a critical task to help the global and local community.
In this paper, we propose a method using deep learning techniques and models that detects the faces of several people in a image or video stream in real-time and then identifies the faces of people who wear a mask and people who do not wear a mask, it also discovers people who put a mask in the wrong way, that is, the mask is under the chin or covers the mouth and chin only, and the nose appears, and here it means its absence, and also discovers people who wear a mask with the real features of their faces hidden behind the mask, that is, they put a mask but appear as if they are not wearing a mask.

***Index Terms*—Covid-19, Computer Vision, Image Processing, Deep Learning, data augmentation, Transfer Learning, CNN(Convolution Neural Network), Back Propagation, Adam Optimizer, Face Detection and Face Mask Detection.**

I. INTRODUCTION

Novel Coronavirus (Covid-19) is a highly contagious epidemic disease caused by Severe Acute Respiratory Syndrome 2 (SARS-CoV-2) that originated in the Chinese city of Wuhan in late December 2019 and has since spread globally, common symptoms include fever, cough and shortness of breath, and COVID-19 infection can mostly be transmitted through respiratory droplets generated when people breathe, talk, cough, or sneeze. The World Health Organization (WHO) announced that Covid-19 became a pandemic on March 12, 2020, and since that time this epidemic has become a major threat to the life of the entire world, which has infected millions of people around the world and led to the death of many of them, and thus the world is facing a huge health crisis due to the rapid transmission of the virus. Therefore, a comprehensive ban was imposed on countries around the world to mitigate this spread, and this affected the local and global economy and led to a halt in the wheel of life, which made researchers work around the clock to find solutions and design strategies to control the epidemic and

Grenze ID: 01.GIJET.9.1.738
© Grenze Scientific Society, 2023

reduce its spread and its impact on human health and the economy. In light of the easing of the ban measures, many guidelines have been issued by the World Health Organization (WHO) to limit the spread of this virus. Accordingly, the most effective preventive measure is to wear a mask for personal protection in public places, but there are people who do not wear masks and it is difficult to monitor them manually. Therefore, when a certain person removes the mask in public, he is not only risking his life, it also risks the lives of others who may have been in contact with the person during the period when he was not wearing a mask. Currently, people who are not wearing a mask are manually and visually screened by guards at entry and exit points, and guards cannot be placed everywhere to check on those people who take off their masks and roam without restrictions once they are checked at the entrance gate, efforts have been made in automatically examining people who do not wear a mask with the help of computer vision and artificial intelligence techniques, and here deep learning models have been used because it has shown tremendous potential in many applications of object detection in images, and one of the most important applications of object detection is the detection of human faces. To monitor that people follow the basic safety principle by wearing masks, a strategy must be developed that is divided into two parts: detecting faces in each input image or in a live broadcast video taken by a camera set for this purpose, and discovering masks on those faces, and this includes discovering the mask if the part of the hidden face is imprinted on it. Behind the mask, and also detecting the mask if it covers the mouth or chin (the mask condition is placed in the wrong way), and an alarm is issued if a person is detected without wearing a mask.

II. Materials and Methods

A. Related Work

Object detection is one of the trending topics in the field of image processing and computer vision. Ranging from small scale personal applications to large scale industrial applications, object detection and recognition is employed in a wide range of industries. Some examples include image retrieval, security and intelligence, OCR, medical imaging and agricultural monitoring. The process of object detection mainly involves localizing the objects in images and classifying them (in case of multiple objects). Traditionally, researchers used pattern recognition to predict faces based on prior face models. A breakthrough face detection technology then was developed named as Viola Jones detector that was an optimized technique of using Haar Cascade [1], and HOG[2], and these algorithms are heavily based on Feature Engineering. However, it failed because it did not perform well on faces in dark areas and non-frontal faces. Since then, researchers are eager to develop new algorithms based on deep learning to improve the models of face detection . Deep learning allows us to learn features with end to end manner and removing the need to use prior knowledge for forming feature extractors. In the era of Deeplearning, it is possible to train Neural Networks that outperform these algorithms, and do not need any extra Feature Engineering. There are various methods of object detection based on deep learning which are divided into two categories: one stage and two stage object detectors. Two stage detectors use two neural networks to detect objects, for instance region-based convolutional neural networks (R-CNN) and faster R-CNN. The first neural network is used to generate region proposals and the second one refines these region proposals; performing a coarse-to-fine detection. This strategy results in high detection performance compromising on speed. The seminal work R-CNN is proposed by R. Girshick et al. [3]. R-CNN uses selective search to propose some candidate regions which may contain objects. After that, the proposals are fed into a CNN model to extract features, and a support vector machine (SVM) is used to recognize classes of objects. However, the second stage of R-CNN is computationally expensive since the network has to detect proposals on a one-by-one manner and uses a separate SVM for final classification. Fast R-CNN [4] solves this problem by introducing a region of interest (ROI) pooling layer to input all proposal regions at once. Faster RCNN [5] is the evolution of R-CNN and Fast R-CNN, and as the name implies its training and testing speed is greater than those of its predecessors. While R-CNN and Fast R-CNN use selective search algorithms limiting the detection speed, Faster R-CNN learns the proposed object regions itself using a region proposal network(RPN) .

On the other hand, a one stage detector utilizes only a single neural network for region proposals and for detection; some primary ones being SSD (Single Shot Detection) [6] and YOLO (You Only Look Once) [7]. To achieve this, the bounding boxes should be predefined. YOLO divides the image into several cells and then matches the bounding boxes to objects for each cell.

This, however, is not good for small sized objects. Thus, multi scale detection is introduced in SSD which can detect objects of varying sizes in an image. Later, in order to improve detection accuracy, [8] proposes Retina Network (RetinaNet) by combining an SSD and FPN (feature pyramid network) to increase detection accuracy and reduce class imbalance. One-stage detectors have higher speed but trades off the detection performance but then only are preferred over two-stage detectors. Like object detection, face detection adopts the same

architectures as one-stage and two-stage detectors, but in order to improve face detection accuracy, more face-like features are being added. However, there is occasional research focusing on face mask detection. Some already existing face mask detectors have been modeled using OpenCV, Pytorch Lightning, MobileNet, RetinaNet and Support Vector Machines. As the world began implementing precautionary measures against the Coronavirus, numerous implementations of Face Mask Detection systems came forth.

Ref[9] have performed facial recognition on masked and unmasked faces using Principal Component Analysis (PCA). However, the recognition accuracy drops to less than 70% when the recognized face is masked.

Ref[10]introduced a method to identify face mask wearing conditions. They divided the facemask wearing conditions into three categories: correct face mask wearing, incorrect face mask wearing, and no face mask wearing. Their system takes an image, detects and crops faces, and then uses

Ref[11] to perform image super-resolution and classify them. The work by [12] presented a method that detects the presence or absence of a medical mask. The primary objective of this approach was to trigger an alert only for medical staff who do not wear a surgical mask, by minimizing as many false positive face detections as possible, without missing any medical mask detections.

Ref[13] proposed a model that consists of two components. The first component performs uses ResNet50 [14] for feature extraction. The next component is a facemask classifier, based on an ensemble of classical Machine Learning algorithms. The authors evaluated their system and estimated that Deep Transfer Learning approaches would achieve better results since the building, comparing, and selecting the best model among a set of classical Machine Learning models is a time consuming process.

B. Proposed Methodology

a) Dataset: The dataset which we have used consists of 1580 total images out of which 889 are of masked faces and 691 are of unmasked faces. A data set is a collection of data. In Deep Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions. We need to split our dataset into two parts :

•Training Dataset: A dataset that we feed into our Deep learning algorithm to train our model.
•Testing Dataset: A dataset that we use to validate the accuracy of our model but is not used to train the model. It may be called the validation dataset.

The purpose of splitting data is to avoid overfitting which is paying attention to minor details/noise which is not necessary and only optimizes the training dataset accuracy. We need a model that performs well on a dataset that it has never seen (test data), which is called generalization. The training set is the actual subset of the dataset that we use to train the model. The model observes and learns from this data and then optimizes its parameters. The validation dataset is used to select hyperparameters (learning rate, regularization parameters). When the model is performing well enough on our validation dataset, we can stop learning using a training dataset. The test set is the remaining subset of data used to provide an unbiased evaluation of a final model fit on the training dataset. Data is split as per a split ratio which is highly dependent on the type of model we are building and the dataset itself. If our dataset and model are such that a lot of training is required, then we use a larger chunk of the data just for training which is our case. 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.

b) Architecture: Fig.1 represents our proposed system architecture. It consists of two major stages. The first stage of our architecture includes a Face Detection Model (CNN), which localizes multiple faces in images of varying sizes and detects faces . The detected faces extracted from this stage are then batched together and passed to the second stage of our architecture, which is a Face Mask Detection Model (CNN). The results from the second stage are decoded and the final output is the image with all the faces in the image correctly detected and classified as either masked or unmasked faces.

C) Face Detection Model: In order to detect the face in color image, we have used the OpenCV library. The latest OpenCV includes a Deep Neural Network (DNN) module, which comes with a pre-trained face detection convolutional neural network (CNN), which is depends on Single Shot MultiBox Detector (SSD). The new model enhances the face detection performance compared to the traditional models. Whenever a new test image is given, it is first resized into 300×300 , we use mean subtraction, normalization and converted from RGB into BGR and then sent into the pre-trained model which outputs the number of detected faces. Every face detected comes out with a level of confidence which is then compared with a threshold value to filter out the irrelevant detections. After we have the faces, we need to evaluate the bounding box around it and resized every detected face into 224×224 because Deep CNNs require a fixed-size input image. Therefore we need a fixed common size for all the images , and then send it to the face mask detection model to check if the face has a mask or not. The overall process flow diagram of the model is shown in Fig. 2 .

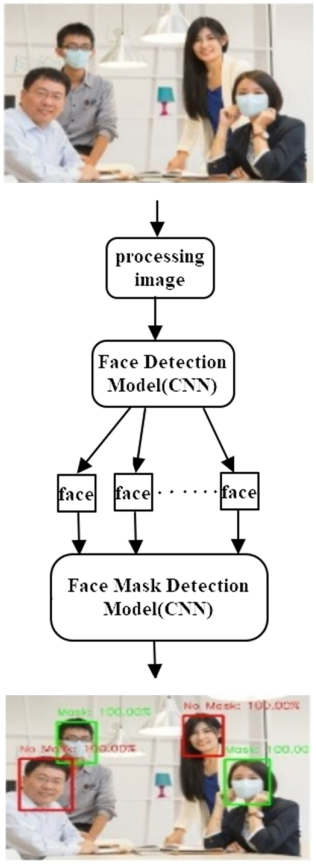

Fig. 1: system architecture

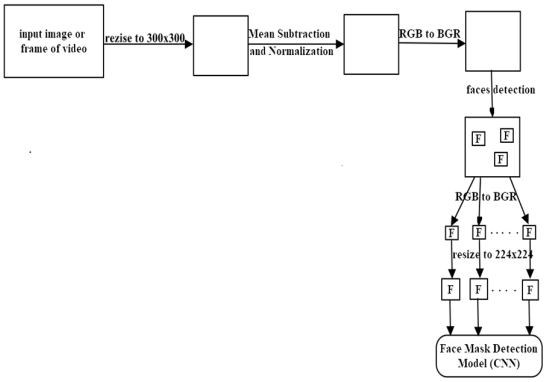

Fig. 2: process flow diagram

d) Face Mask Detection Model: The face mask detection model is trained by us using a dataset consisting of images with mask and without mask. We have used Keras along with Tensorflow to train our model. First part of the training includes storing all labels of the images in a Numpy array and the corresponding images are also reshaped (224, 244, 3) for the base model. Before inputting, new image data is generated based on old image data by making effects on images such as rotations up to 20 degrees, zooming in and out up to 15%, width or height shift up to 20%, up to 15 degrees shear angle in the counterclockwise direction, flip inputs horizontally and points outside the boundaries of the inputs are filled from the nearest available pixel of the input. This step is called Data Augmentation, and it is a very useful technique because it increases the data set of the images. This

step contributes to the process of generalization of the model, making it more efficient. For the image classification, it is now a common practice to use transfer learning which means using a model which has been pre-trained on millions of labels before and it has been tested that this method results in significant increase in accuracy. Obviously, the assumption here is that both the problems have sufficient similarity. It uses a well-structured and deep neural network that has been trained on a large amount of data set. Due to somewhat same nature of the problem, we can use the same weights which have the capability to extract features and later in the deep layers, convert those features to objects. The base model that we have used here is MobileNetV2 with the given 'ImageNet' weights. ImageNet is an image database that has been trained on hundreds of thousands of images hence it helps a lot in Image classification. For the base model, we truncate the head and use a series of our self defined layers. We used an average pooling layer to reduce the spatial dimensions of the output volume and Pool size is set to 7 x7, a flatten layer which transforms matrix of features into a vector that can be fed into a fully connected neural network classifier, a dense layer of 128 neurons with a ReLu activation function is added , a 50% dropout layer for optimization, The final layer (Dense) with two outputs for two categories uses the Softmax activation function. The overall process flow diagram of the model is shown in Fig. 3.

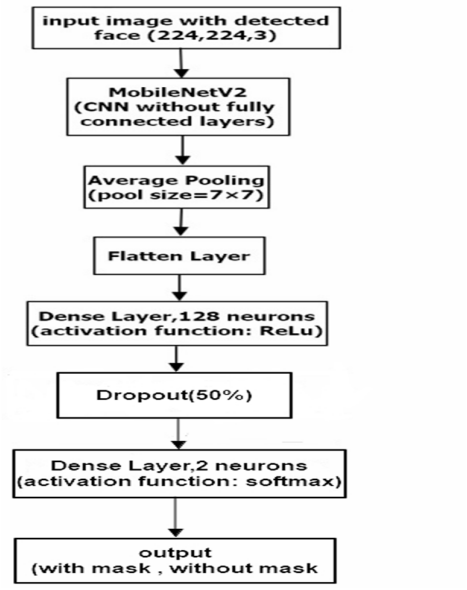

Fig. 3 : process flow diagram of the model

e) Training The Face Mask Detection Model: To train the built model, we rely on the Back Propagation method, that is, it adjusts the weights of the output layer on the test data based on the value of the loss function, which measures the difference between the actual output and the expected output of the model. As the number of epochs increases, its value decreases, On the other hand, the accuracy increases, and this is what we want, as we note in Fig. 4, which shows the training process, And here we use binary_crossentropy as a loss function. During the training process, the concept of Validation is used, which is a useful concept to check the efficiency of the model during the training process and prevent it from falling into the problem of overfitting. Therefore, the validation data is used to verify the efficiency of training the model and is used during the training phase, but is not used for the process of adjusting the weights.

f) Hyperparameters: A hyperparameter is a parameter or a variable we need to set before applying an algorithm into a dataset. These parameters express the "High Level" properties of the model such as its complexity or how fast it should learn. Hyperparameters are fixed before the actual training process begins. They can be divided into two categories: optimizer hyperparameters and model hyperparameters. Optimizer parameters help us to tune or optimize our model before the actual training process starts. Some common optimizer hyperparameters are as follows. Learning rate is a hyperparameter that controls how much we are adjusting the weights of our neural network with respect to the gradient. Mini-batch size is a hyperparameter that influences the resource requirements of the training and impacts training speed and number of iterations. Epochs are the

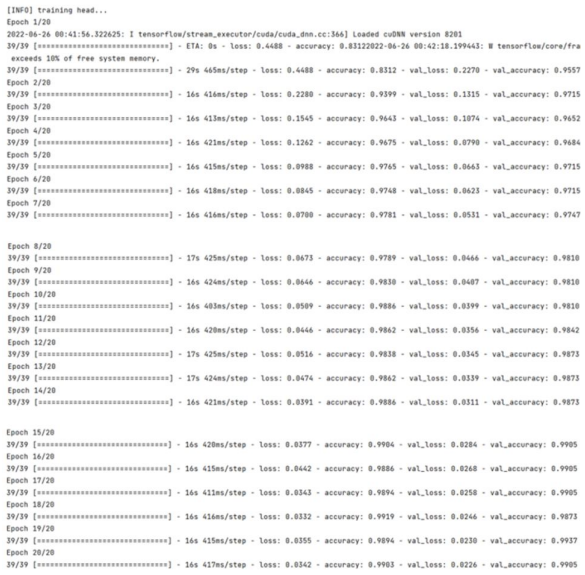

Fig. 4: training the model

hyperparameters that determine the frequency of running the model. One epoch is when an entire dataset is passed forward and backward through the neural network only once. Model hyperparameters are parameters that are more involved in the architecture or structure of the model. They help us to define our model complexity based on the different layers like the input layer, hidden layer, and output layer of a neural network. Initially, we trained with different values of hyperparameters by changing one and keeping the other constant and noted down the results in each case. We selected the hyperparameters that produced better performance through 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 20. In our case, the target size is also one of the hyperparameters which we kept (224, 224, 3) as it is default input shape of MobileNetV2.

g)Evaluation The Model: After the model training process is completed, the model is evaluated on the testing data, which are data that are not included in the model training process. The evaluation process consists of first looking at the classification report which gives us insight towards precision, recall and F1 score. The equations of these three metrics are as follows:

$$\text{Precision(P)}=\frac{True\ Positives}{True\ Positives+False\ Positives} \quad (1)$$

$$\text{Recall(R)}=\frac{True\ Positives}{True\ Positives+False\ Positives} \quad (2)$$

$$\text{F1-score}=2\times\frac{P\times P}{P+R} \quad (3)$$

Using these three metrics, we can conclude the efficiency and performance of the model. The Classifiction Report on the testing data during the process of training the model on the training data appears as follows, Fig.5.

```
[INFO] evaluating network...
2022-06-26 00:47:31.805471: W tensorflow/core/framework/cpu_allocator_impl.cc:80]
              precision    recall  f1-score   support

   with_mask       0.99      0.99      0.99       178
without_mask       0.99      0.99      0.99       138

    accuracy                           0.99       316
   macro avg       0.99      0.99      0.99       316
weighted avg       0.99      0.99      0.99       316
```

Fig.5 : Classifiction Report

Finally, the training curve and the validation curve for Loss/Accuracy are plotted in terms of Epochs, Fig. 6.

Fig. 6: the training curve and the validation curve for Loss/Accuracy

We note that the Accuracy rate of the training data (train_acc) starts with approximately 78% at the first epoch and increases with the number of epochs until it reaches approximately 99% , on the other hand, the average loss of training data (train_loss) starts with approximately 45% at the first epoch and decreases with increasing number of epochs to reach approximately 0.1% .
We note that the Accuracy rate of the validation data (val_acc) starts with about 95% at the first epoch and increases with increasing the number of epochs until it reaches approximately 99%, which is considered a very high percentage and therefore the efficiency of the model is high, on the other hand, the loss rate for the validation data (val_loss) starts with about 22% at the first epoch and decreases with the increase in the number of epochs to less than 0.1%, and there are simple signs indicating the emergence of overfitting because the value of the loss rate for the validation data is slightly less than the loss rate value for the training data, the lower the loss rate and the higher the accuracy rate, the better the model and neural network used.

III. RESULT

We implemented our model on images containing one and more faces. We also implemented it on videos and live video streams by removing and wearing masks one by one. Some screenshots of the results are shown below:

A. Test 1

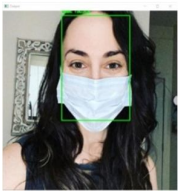

Fig. 7 : Test 1

Fig. 7 shows an image of a girl who puts a mask on her face correctly and covers the nose and mouth, when it was entered into the model that we built, the face was surrounded by a rectangle in green and given the classification mask and the percentage of detection, which means that it was discovered that she was wearing a mask at 100% .

B. Test 2

Fig. 8 : Test 2

Fig. 7 shows an image of a man who does not wear a mask on his face, when entered into the model that we built, the face was surrounded by a rectangle in red and given the classification no mask and the percentage of detection, which means that it was discovered that he is not wearing a mask 100% .

C. Test 3

Fig. 9 : Test 3

Fig. 9 shows an image of a young man putting a mask on his face, but in the wrong way, as it is placed on the chin only and does not cover the nose and mouth. When inserted into the model that we built, the face was surrounded by a rectangle in red and given the classification no mask and the percentage of detection, which means that it was discovered that he does not wear a mask with a percentage of 99.81 % .

C. Test 4

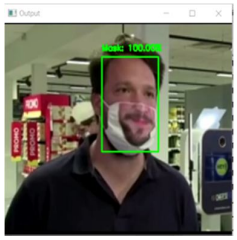

Fig. 10 : Test 4

Fig. 10 shows an image of a young man putting a muzzle on his face, but the real part of the face hidden behind the muzzle was printed on it, when it was entered into the model that we built, the face was surrounded by a rectangle in green and given the classification mask and the percentage of detection, which means that it was discovered that he was wearing a muzzle at 100% .

D. Test 5

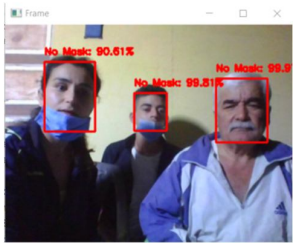

Fig. 11 : Test 5

Fig. 11 shows an image captured by a camera in real time, which is a frame from a live broadcast video of three people, one of whom is not wearing a mask, where the face is surrounded by a rectangle in red and given the classification no mask and the percentage of detection, which means that it was discovered that he is not wearing a mask by 99.99%, and the second is wearing a mask But in a wrong way, as it is placed on the chin and mouth only and does not cover the nose, as the face was surrounded by a rectangle in red and given the classification no mask and the percentage of detection, which means that it was discovered that he did not wear a mask with a percentage of 99.81%, and the third is wearing a mask but in a wrong way as it is placed on the chin only It does not cover the nose and mouth, and it was discovered that he does not wear a mask by 90.61% .

E. Test 6

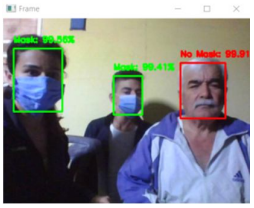

Fig. 12 : Test 6

Fig. 12 shows an image taken via a camera in real time, which is a frame from a live broadcast video of three people, one of whom is not wearing a mask, where the face is surrounded by a rectangle in red and given the classification no mask and the percentage of detection, which means that it was discovered that he is not wearing a mask by 99.91%, and the second is wearing a mask Correctly covering the nose and mouth, where the face was surrounded by a rectangle in green and given the classification mask and the percentage of detection, which means that it was discovered that he is wearing a mask with a percentage of 99.41%, and the third is wearing a mask correctly and covers the nose and mouth where the face was surrounded by a rectangle in green and giving the classification mask and the percentage for discovery, which means that it was discovered that he was wearing a muzzle, with a percentage of 99.56% .

IV. Future Work

More than fifty countries around the world have recently initiated wearing face masks compulsory. People have to cover their faces in public, supermarkets, public transports, offices, and stores. Retail companies often use software to count the number of people entering their stores. They may also like to measure impressions on digital displays and promotional screens. We are planning to improve our Face Mask Detection tool and release it as an open-source project. Our software can be equated to any existing USB, IP cameras, and CCTV cameras to detect people without a mask. This detection live video feed can be implemented in web and desktop applications so that the operator can see notice messages. Software operators can also get an image in case someone is not wearing a mask. Furthermore, an alarm system can also be implemented to sound a beep when someone without a mask enters the area. This software can also be connected to the entrance gates and only people wearing face masks can come in.

References

[1] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, vol. 1. IEEE, 2001, pp. I–I.

[2] Dalal, N., and Triggs, B., Histograms of oriented gradients for human detection, CVPR '05, 2005.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[4] R. Girshick, “Fast r-cnn,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

[5] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Advances in neural information processing systems, 2015, pp. 91–99.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in European conference on computer vision. Springer, 2016, pp. 21–37.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” 2017.

[9] Ejaz, M.S., Islam, M.R., Sifatullah, M., Sarker, A.,Implementation of principal component analysis on masked and non-masked face recognition, 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT),2019, pp. 1–5.

[10] Qin, B., and Li, D., Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19 (2020), Unpublished results, doi:10.21203/rs.3.rs-28668/v1.

[11] Dong, C., Loy, C.C., Tang, X., Accelerating the Super-Resolution Convolutional Neural Network, in Proceedings of European Conference on Computer Vision (ECCV), 2016.

[12] Nieto-Rodríguez, A., Mucientes, M., Brea, V.M., (2015), System for Medical Mask Detection in the Operating Room Through Facial Attributes. In: Paredes R., Cardoso J., Pardo X. (eds) Pattern Recognition and Image Analysis. IbPRIA 2015. Lecture Notes in Computer Science, vol 9117. Springer, Cham. https://doi.org/10.1007/978-3-319-19390-8_16.
[13] Loey, M., Manogaran, G., Taha, M. H. N., & Khalifa, N. E. M. (2021) (in press), A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic, Measurement: Journal of the International Measurement Confederation, 167. https://doi.org/10.1016/j.measurement.2020.108288.
[14] He, K., Zhang, X., Ren, S., and Sun, J., Deep residual learning for image recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.