Social Distance Measurement and Face Mask Detection using Deep Learning Models

Md Naimul Islam Suvon[1], Md Mahabub Alam[1] and Riasat Khan[1]

[1] North South University, Dhaka, Bangladesh
mahabub.alam01@northsouth.edu, naimul.suvon@northsouth.edu, riasat.khan@northsouth.edu

Abstract. The coronavirus disease 2019 has caused a worldwide catastrophe with its destructive spreading and causing death of more than 2.47 million people around the globe. In the current circumstance, most of the countries are trying to implement social distancing, wearing masks, extensive testing, and contact tracing strategies to curb the virus outbreaks. Maintaining adequate social or physical distance is believed to be a sufficient precautionary measure (standard) against the spread of the pandemic infection. This research paper has two different contributions of social distance measurement and face mask detection using various deep learning approaches. In the first section, we have monitored the social distance where we have detected people by examining a video feed with SSD-MobileNet and Faster R-CNN ResNet50 deep learning algorithms. Next, the image is converted into an overhead view to measure the specific distance among people to ensure safe physical distancing. In the second section, we have detected the face masks used by the people by implementing MobileNetV2 convolutional neural network architecture. Hence, we have used computer vision to find the region of interest of a face, and finally, we have found that the mask is in the face or not. Both of our social distance measurement and face mask detection systems offer high accuracy. As for the social distance monitoring, the accuracy greatly depends on the people detection, and the execution time is 30 ms and 89 ms for SSD-MobileNet and Faster R-CNN ResNet50, respectively. For the face mask detection, we obtained 99% accuracy, and it is checked in real-time so that we can prove that our model is not overfitting and it performs well outside our dataset in real-time camera.

Keywords: bird view, computer vision, distance matrix, Euclidean distance, Faster R-CNN ResNet50, pairwise distance, people detection, SSD-MobileNet.

1 Introduction

The novel coronavirus outbreak started in Wuhan, China, from December 2019, and then it consequently influenced numerous nations worldwide. The World Health Organization (WHO) declared it a pandemic sickness as the infection spread through 114 nations, caused more than 111 million dynamic cases till February 2021. The infection fundamentally spreads in those individuals, who are in close contact with one another (inside 6 feet), which is known as person-to-person transmission. Someone can transmit it without having any symptoms, and they are referred to as the asymptomatic carriers. Hence, it is important to wear a mask over the nose and mouth and maintain 6 feet distance from others, specially outside the home, regardless of whether individuals do not have any indications or not.

This paper has implemented a robust system that will automatically monitor the people maintaining adequate physical and social distance and wearing facemask outside their home. We will detect humans from video sequences with SSD-MobileNet and Faster R-CNN ResNet50 deep learning algorithms in the first proposed system. Next, the video frame will be converted into an overhead or bird view to estimate the distance between people accurately. It offers comprehensive coverage of the total view of the image and eliminates the effect of different barriers. Finally, an approximation of adequate physical separation to the image pixels is established to check the possible violation of the social distance. We will train a custom dataset model containing mask and no mask data for the facemask detection system. Then we will detect the region of interest (ROI) of an input image and detect facemask in every ROI using the custom model.

We have arranged the paper in the following sections: first, the related works are discussed in Section 2. Section 3 contains the system methodology of our work in detail. Section 4 includes assessing the individual and the total system's performance. Finally, Chapter 5 concludes the research work with a short description of future works.

2 Related Work

In the last few years, there are many works on image processing, computer vision, machine and deep learning, which have been implemented on the object detection and distance measurement tasks very successfully. In [1], G. Deore *et al.* used the face detection approach for security purposes. The authors used the Viola-Jones algorithm for facial part detection and the pinhole camera model to measure the distance from the camera. A projection histogram is used in their project for eye line detection. They achieved higher accuracy in measuring the distance from the camera and eye line detection. On the other hand, they were not able to increase facial part detection and eye detection accuracy. The accuracy of these two tasks was less than 50%.

S. V. Tathe and S. P. Narote detected humans from moving cameras using several algorithms [2]. They utilized projection function and pixel count methods to detect the eye region, a shift algorithm, and the Kalman filter technique for tracking humans. This work utilizes a hierarchical Gaussian process to detect humans, which helps them detect people more accurately than it would be possible from one frame. Finally, the authors used the Markov model to extend their work into people tracking in video sequences. This approach covered a small number of frames at once, and it also helped to possibly longer real-time tracking by reducing the detection time.

In [3], E. N. Kajabad and S. V. Ivanov proposed a system to detect distance between people using the video frames from a static camera. This paper uses the RCNN algorithm [11] to detect moving objects (people), which applies a sequential learning data approach from the video frame. The image is represented by pixels, and a pixel consists of numbers, mostly between 0 to 255. The work used a background subtraction rule (known as Gaussian Mixture algorithms), optical flow methodology (heat map color technique to analyze every frame), and median filter to eliminate noise throughout the tracking and detect people in their system.

Y. C. Hou *et al.* proposed a profound CNN model on object detection in [4]. They moderated the computational unpredictability issues by figuring the location with a solitary regression issue. The authors used the You only look once (YOLO) model [5] to obtain the high-speed performance. They trained the YOLO model with the COCO dataset. To report the YOLO model's result, they used box coordinates, object confidence, and people object class for people detection. For camera view, the ROI of a picture centers around the person on foot strolling road was changed, which is discussed briefly in this paper [6]. Overall, they have shown a well-structured way of measuring distance between people in an image as the distance between pixels and real life is not the same.

3 System Methodology

In this paper, a social distance detection system has been implemented first to measure the distance between people in various outdoor and indoor environments. The proposed social distance detection is divided into three sections.

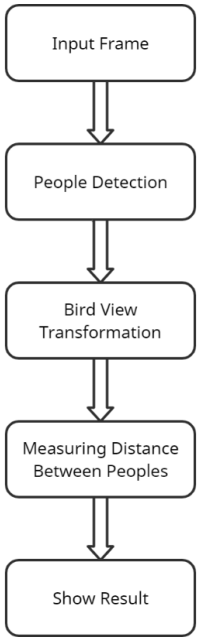


Fig. 1. Proposed social distancing detection process.

As shown in Fig. 1, the proposed system will take first the video images as an input frame from the camera; then, it will detect all the people in that frame. Next, it will convert the image into a bird view image, which will give us the top view or overhead

view of the image. We will then use this image to measure the distances between the people and demonstrate the images' results by showing an alert if the threshold distance is violated. The top or overhead view approach of the image leads to a better approximation of the distance by overcoming the occlusion effect.

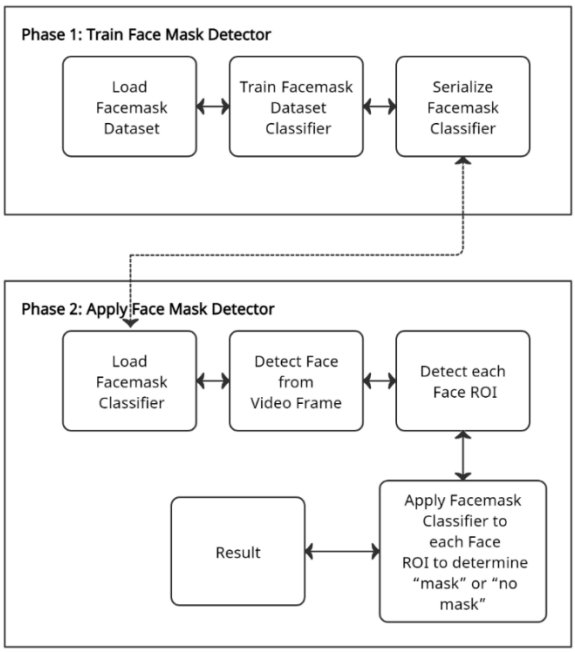


Fig. 2. Proposed face mask detection system.

In Fig. 2, we have partitioned our face mask detection system into two parts, where in the first phase, we will focus on loading the dataset and train our dataset to make a model and then serialize the face mask detector. Finally, during the second stage, we can move into stacking our indicator model, perform face discovery and ordering each face as either "Mask" and "No Mask". In the subsequent sections, the social distance measurement and face mask detection processes have been discussed in detail.

3.1 Social Distancing Detection

Model Selection. For people detection, first of all, we have to select an effective dataset and then a perfect object detection model has to be applied. There are already many pre-trained models that can detect people with high accuracy. We have used the model trained by the COCO (Common objects in context) dataset [7]. This dataset has 120,000 images with a total of 88,000 labeled objects in these images. All the models of COCO are trained to detect 90 different kinds of objects. From these, one of the objects is people's class, which is needed for our system. Many trained models have already been

used this dataset using different CNN architectures, e.g., SSD-MobileNet [8], Inception_v3 [9], Resnet50, Fasten RCNN, and more. These models exhibit different performances depending on the speed of the model. We have done some tests on some of these models to evaluate our model's quality, especially its frames per second. As we will do real-time video analysis, finally, we have chosen the SSD-MobileNet_v1_COCO model for the people detection, which has a faster execution speed [10].

People Detection. To use our SSD-MobileNet model to detect people, we load it into a TensorFlow graph and define the output in our case. After that, we have passed all our input frame images to obtain the desired outcome. We specified some of the parameters during this process, including the type of input required for the model and the specific output we want from our model. In our case, we need the outputs of bounding box coordinates, the confidence of each prediction and prediction object class. In the COCO model, the required people class is one from the 90 object classes. After detecting, we have filtered out the weak prediction results by selecting a threshold of 0.75.

Bird View Transformation and Measuring Distancing. For transforming an image into a bird view, we have first taken the 4 points on the original image that we are going to convert. The topics have to form a rectangle with at least two opposite sides being parallel. It has been done to maintain the same proportion when the transformation takes place. For the people detection, we have found a location of the bounding box for each person. Next, from that bounding box, we have to calculate the lowest point, which is the bottom-center point of that box, and it will help us find the ground point of a person. Next, the transformation matrix is used to get each person's real GPS coordinates to each of these points.
After finding each person's ground point, the distance between each pair of points is computed. We have used equation (1), which is the Euclidian distance, to measure the distance between the points.

$$d\,(p,q) \;=\; (\,(p_1 - q_1)^2 \;+\; (p_2 - q_2)^2 \;+\; (p_3 - q_3)^2\,)^{\frac{1}{2}} \qquad (1)$$

We have chosen 120 pixels as our social distance threshold, which is approximately equal to 2 feet.

3.2 Face Mask Detection

Dataset and Model Selection. We have used the open-source dataset of P. Bhandary [12] for face mask detection. There are in total 1,376 images, and among them, 690 and 686 images are with and without the mask, respectively. We have taken almost 1:1 ratio for both of the classes. Our aim is to make a custom model to detect whether a person is wearing a mask or not. We have chosen MobileNetV2 convolutional neural network architecture for the face mask detection model, which is a highly efficient architecture, especially for implementing embedded devices (Raspberry Pi, Google Coral, NVIDIA Jetson Nano) with minimum computational capacity.

Data Preparation and Training. Before training, we have preprocessed the images by resizing them into 224×224 pixels, converting them to array format, and scaling the pixel intensities in the input image to the range $[-1, 1]$ by using the convenience function. Then we have appended the preprocessed data and associated label to labels and data lists, respectively. Next, we have done a one-hot encoding to our class labels, where each element of our array consists of an array in which only one index is "hot". We have divided our face mask detection data into 80:20 for training and testing, respectively. We have applied an on-the-fly mutation to our data at the time of our training to improve our generalization, known as data augmentation, where random zoom, rotation shear, shift, and flip parameters are initialized. After preparing the dataset, next, we must fine-tune the MobileNetV2 architecture. To fine-tune the MobileNetV2, we have followed three steps. First, we have loaded the MobileNetV2 model with pre-trained ImageNet weights by leaving off the head of network. Next, we have constructed a new fully connected head and append it to the base in place of the old head. Finally, we freeze the base layer of the network where the weights of this base layer will not be updated during the process of backpropagation.

Implementation of Face mask detector with OpenCV. Finally, after training our face mask detector model, we will start the second phase of our system, which loads the model and detects faces and the region of interest (ROI). We have first taken the input image and preprocess it using OpenCV's 'blobFromImage' function of the deep neural network module. We have resized the image into 300×300 pixels and performed a subtraction. After this, we detect the faces from the images to find their precise location. Next, we have extracted all the face ROIs. We have then computed the bounding box values for each face and compare them to see if the boxes fall within the boundaries of the images. Finally, we obtained a prediction of "Mask" and "No mask" to each of those faces. We also attain a percentage of detection for each face to know the accuracy of each face mask detection.

4 Performance Evaluation

In this section, we have evaluated the performance of the people detection and social distance measurement and face mask recognition systems. The social distance measurement work's performance has been assessed in two steps. First, we have evaluated the performance of human detection and then, we have shown the performance of the overall system.

4.1 Performance of People Detection

For the people detection, we have used a pre-trained COCO model. As we know, for object detection, we cannot measure the accuracy metrics like the classification problem. There exist 12 different accuracy metrics for object detection; the mAP (mean average precision) is one of them. mAP is the average measurement area under the precision-recall (PR) curve. It indicates the detection model's accuracy and compares

the detected bounding box with the ground-truth box. The higher value of the mAP score indicates the better accuracy of the detection system. For our implemented SSD-MobileNet_v1_COCO model for people detection, we obtained an average mAP of 21% with a 30 ms execution time, which is fast compared to the other detection models. We choose this model because we have to analyze it in real-time, and we have a plan to implement our proposed system in an embedded system, so our frames should execute very fast. We have also tried the Faster R-CNN ResNet50 (COCO) model, which has better accuracy of 30% mAP, but the execution time is 89 ms, which is slow for the real-time analysis. In Table 1, we have shown the performance metrics of the pre-trained COCO models with their execution time and mAP.

Table 1. Model Accuracy Metrics for various human detection algorithms.

Accuracy Metrics	SSD-MobileNet	Faster R-CNN ResNet50
mAP	21%	30%
Execution time	30 ms	89 ms

4.2 Performance of Overall Social Distance Measurement System

After detecting the people, the image has been transformed into a top or bird view. Finally, the distance between the people will be measured using that bird view image, and we have found the overall result.

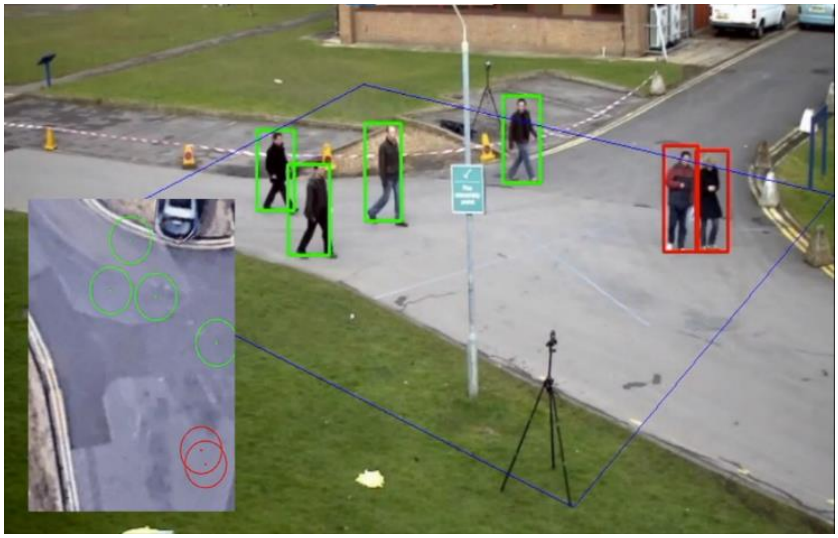


Fig. 3. Social distancing detection evaluation on video with the bird view transformation by the SSD-MobileNet algorithm.

In Fig. 3, we have shown an image by implementing our proposed SSD-MobileNet algorithm. We can see the bird view on a small screen (lower left corner of the image), which is the four corner points that we take from the original image and transform it into a bird view with a ground point for each person. From Fig. 3, we can observe that when the person is too close to each other (below the predefined threshold of 120 pixels), the bounding box around the individual will be indicated with a red box, and the circle of the bird view image turns to red.

4.3 Performance of Face mask Detection

We have used 34 epochs during the training and validate our model with 276 (20%) sample images for the face mask detection model. Fig. 4 demonstrates the training and testing accuracy and loss curves of the face mask detection model. It illustrates the high accuracy of the detection system with the validation loss lower than the training loss. We have achieved an accuracy of approximately 99% in our test dataset.

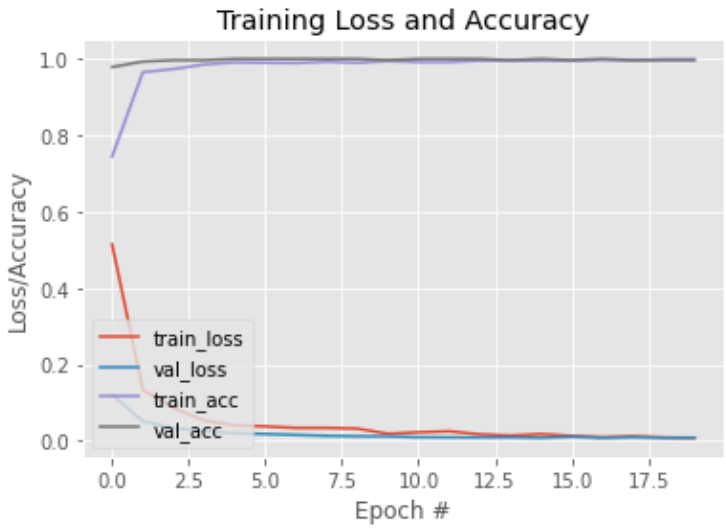


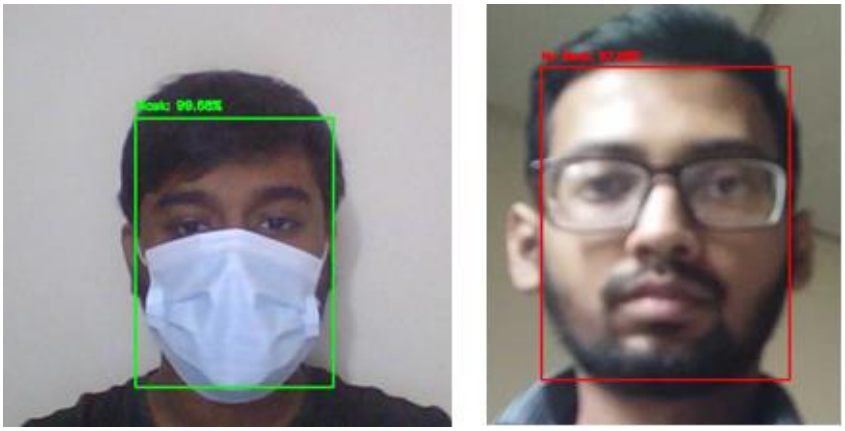
Fig. 4. Training loss vs. accuracy of the face mask detection model.

Table 2 demonstrates different performance metrics (precision, recall, f1-score, and support) of the face mask detection system. Finally, we have implemented our face mask recognition model in a real environment by using a mobile IP camera, which is connected to our detection program, to see if it is working well.

Table 2. Model Accuracy Metrics for face mask Detection.

	Without Masks	With Masks	Macro avg	Weighted avg	Accuracy

Precision	0.99	1.00	0.99	0.99	
Recall	1.00	0.99	0.99	0.99	
F1-Score	0.99	0.99	0.99	0.99	0.99
Support	138	138	276	276	276



(a) (b)

Fig. 5. Real-time face mask detection with (a) mask and (b) without mask.

Fig. 5 (a) shows that our model can detect a face with the mask accurately. The images are outside our training and testing dataset in real-time video frames, and it is detecting the face mask continuously. In Fig. 5 (b), we have also demonstrated that our model can also detect a face without mask in real-time with high accuracy. The frame was a little bit blurry, but still, it could detect the face without mask.

5 Conclusion

In this paper, we have implemented two safety systems for the current and future COVID-19 situation, i.e., physical distance measurement and face mask detection. We have detected people from video frames with the SSD-MobileNet and Faster R-CNN ResNet50 deep learning algorithms in the first system. Next, we measured the distance between them by transforming the image into an overhead or bird view approach. An approximation of social distance into pixel is considered, and a threshold value is established to detect its violation. This system's accuracy greatly depends on people detection, where we have used a pre-trained people detection model that has above 95% accuracy, with a fast execution time. The second preventive system is implemented to

detect face masks by implementing MobileNetV2 convolutional neural network architecture. Our model of mask detection has 99% accuracy, and our system can also recognize faces in real-time.

In the future, we can efficiently run our proposed deep learning-based models in an embedded system, e.g., NVIDIA Jetson Nano, and make a wireless face mask detector and social distance detector device. Moreover, we can also make a parallel implementation of these two systems to work into a single device that will reduce the cost of the total system. The human detection system's accuracy can be further increased by adding an overhead dataset with the pre-trained model.

References

1. Deore, G., Bodhula, R., Udpikar, V., More, V.: Study of masked face detection approach in video analytics. Conference on Advances in Signal Processing (CASP), pp. 196–200, Pune, India (2016).
2. Tathe, S. V., Narote, S. P.: Realtime human detection and tracking. Annual IEEE India Conference (INDICON), pp. 1–5, Mumbai, India (2013).
3. Kajabad, E. N., Ivanov, S. V.: People detection and finding attractive areas by the use of movement detection analysis and deep learning approach. Procedia Computer Science, vol. 156, pp. 327–337 (2019).
4. Hou, Y. C., Baharuddin, M. Z., Yussof, S., Dzulkifly, S.: Social distancing detection with deep learning model. 8th International Conference on Information Technology and Multimedia (ICIMU), pp. 334–338, Selangor, Malaysia (2020).
5. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788, Las Vegas, United States (2016).
6. Ahamad, A. H., Zaini, N., Latip, M. F. A: Person detection for social distancing and safety violation alert based on segmented ROI. 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), pp. 113–118, Penang, Malaysia (2020).
7. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L.: Microsoft COCO: Common objects in context. European conference on computer vision, pp. 740–755, Springer, Cham (2014).
8. Chiu, Y. C., Tsai, C. Y, Ruan, M. D, Shen, G. Y., Lee, T. T.: Mobilenet-SSDv2: an improved object detection model for embedded systems. 2020 International Conference on System Science and Engineering (ICSSE), pp. 1–5, Kagawa, Japan (2020).
9. Documentation of Inception_v3. https://www.tensorflow.org/api_docs/python/tf/keras/applications/inception_v3 Accessed 22 Feb 2021.
10. TensorFlow Model Garden. https://github.com/tensorflow/models Accessed 22 Feb 2021.
11. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Region-based convolutional networks for accurate object detection and segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (1), pp. 142–158 (2016).
12. Bhandary, P.: Face mask dataset, GitHub repository. https://github.com/prajnasb/observations/tree/master/experiements/data (2020).