Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.030638
Article

Tech Science Press

Face Mask and Social Distance Monitoring via Computer Vision and Deployable System Architecture

Meherab Mamun Ratul, Kazi Ayesha Rahman, Javeria Fazal, Naimur Rahman Abanto and Riasat Khan*

Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh
*Corresponding Author: Riasat Khan. Email: riasat.khan@northsouth.edu.
Received: 30 March 2022; Accepted: 08 June 2022

Abstract: The coronavirus (COVID-19) is a lethal virus causing a rapidly infectious disease throughout the globe. Spreading awareness, taking preventive measures, imposing strict restrictions on public gatherings, wearing facial masks, and maintaining safe social distancing have become crucial factors in keeping the virus at bay. Even though the world has spent a whole year preventing and curing the disease caused by the COVID-19 virus, the statistics show that the virus can cause an outbreak at any time on a large scale if thorough preventive measures are not maintained accordingly. To fight the spread of this virus, technologically developed systems have become very useful. However, the implementation of an automatic, robust, continuous, and lightweight monitoring system that can be efficiently deployed on an embedded device still has not become prevalent in the mass community. This paper aims to develop an automatic system to simultaneously detect social distance and face mask violation in real-time that has been deployed in an embedded system. A modified version of a convolutional neural network, the ResNet50 model, has been utilized to identify masked faces in people. You Only Look Once (YOLOv3) approach is applied for object detection and the DeepSORT technique is used to measure the social distance. The efficiency of the proposed model is tested on real-time video sequences taken from a video streaming source from an embedded system, Jetson Nano edge computing device, and smartphones, Android and iOS applications. Empirical results show that the implemented model can efficiently detect facial masks and social distance violations with acceptable accuracy and precision scores.

Keywords: Artificial intelligence; COVID-19; deep learning technique; face mask detection; social distance monitor; you only look once

1 Introduction

COVID-19, known as coronavirus or SARS-CoV-2, originated from animals in Wuhan, China, at the end of 2019 [1]. This virus causes respiratory illness and can affect multiple organ systems in the body [2]. According to the latest statistics, COVID-19 has claimed a staggering number of more than 6.22 million lives worldwide in the span of more than two years. The coronavirus has become one of the

deadliest epidemics of the 21st century. By taking the form of a pandemic, this deadly virus continuously infects new people every day. Apart from inevitable deaths, the coronavirus is causing long-term health complications to all of its victims, which total up to 140 million people [3]. From causing respiratory illness to severe heart and kidney failure, the world has realized the importance of preventing the spread of this destructive disease within a few months of its discovery and outbreak. Various institutions and vaccine providers are working round the clock to create a preventive solution to the pandemic [4]. However, until a vaccine is globally administered, the coronavirus continues to pose the risk of claiming thousands of lives every day. Therefore, we cannot solely depend on the innovation of a vaccine. As the disease transmits rapidly, it cannot be controlled without proper safety and monitoring protocols [5].

Coronavirus spreads through person-to-person contact through respiratory droplets [6]. People without symptoms or those who do not have symptoms yet can transmit the virus efficiently. From reports by the World Health Organization, maintaining social distance, wearing face masks, and avoiding crowded places are the most effective and simple methods to reduce health risks amongst the general masses and continue everyday activities with little to no obstruction [7]. Even though the prevention of the coronavirus is a relatively simple process, many people all around the globe are still not following the safety guidelines [8]. Therefore, this issue calls for the need for a systematized and automated monitoring of the general people to ensure preventive and protective measures against the disease.

In this research, an automatic system has been developed to monitor everyday pandemic preventative measures using artificial intelligence and computer vision techniques. This paper will employ two detection models that observe and detect within the local feeds from surveillance cameras, smartphones, and embedded system video streams to detect facial masks and social distance between the people in public areas effectively. It will also attempt to create a safer environment for people to carry on their daily duties in light of the pandemic by evaluating the system performance and detection on remote platforms, spreading its deployment further using different video streaming devices, including embedded systems, Jetson Nano with webcam and smartphone. The proposed system is expected to be highly efficient for spaces where large crowds gather, as it can be operated using a simple smartphone. The major contributions of this manuscript are as follows:

- An automatic COVID-19 prevention system has been developed to detect face masks and monitor social distance. A modified version of a convolutional neural network (CNN), the ResNet50 model, has been used to identify masked faces in this research. YOLOv3 and DeepSORT approaches are applied to monitor the social distance between people.
- NVIDIA Jetson Nano embedded system has been used as a medium to collect video streams from various CCTV surveillance footage.
- We have also utilized smartphones (both Android and iOS devices) to obtain real-time video streams. Smartphone applications have been developed employing Xcode and Swift environments.
- To the best of our knowledge, an integrated face mask detection and social distance monitoring system has been designed for the first time in this paper using Jetson Nano embedded device and smartphone applications.

The paper is arranged accordingly: Section 2 summarizes some of the related works on automatic face mask recognition and social distance monitoring. Section 3 explains the proposed system implementation, including the dataset, utilized software and hardware components. Next, the real-time results of the implemented system are demonstrated in Section 4. Lastly, Section 5 wraps up the paper with some suggestions for future implementations.

2 Related Work

Traditional prevention of COVID-19 by manual inspection of face mask detection and physical distance measurement is highly inefficient and needs extensive human labor. Therefore, computer vision and artificial intelligence-based automatic face mask detection and physical distance measurement have been thoroughly studied in recent years [9]. Some of these studies involve only face mask detection [10]; some of them include social distance measurement [11] and some with mixed results. Several works in this context of automatic coronavirus prevention have been discussed briefly in the subsequent paragraphs.

Many works have studied the detection of masked and unmasked faces for COVID-19 prevention, security reasons, identifying individual persons, and tracking criminals. For instance, in a recent work [12], the authors have proposed an automated technique to detect masked faces by utilizing four different steps. These steps are estimating the person's distance from the camera, eye line detection, facial part recognition, and eye detection. They have outlined the benchmarks in every step where commonly known algorithms have been used for human and face detection. Analog Devices, Inc.'s Cross Core Embedded Studio (CCES) and HOGSVM were utilized to determine the distance from the camera step and identify individuals. The distance from the camera step was found using the ADSP BF609 dual-core processor, with the highest accuracy of 90 percent. On the other hand, eyeline, facial part, and eye detection give an accuracy of 69.8 percent, 46.6 percent, and 40 percent, respectively. There are some limitations in face detection where the face cannot be appropriately detected if the person is masked. Also, false rate detection is the highest in eye detection and eye line detection. So, the accuracy percentage is relatively lower than expected for these two cases. In [13], the authors harnessed the power of TensorFlow, Keras, OpenCV, and Scikit-Learn libraries and various machine learning packages to detect face masks. It can identify faces from other objects and then determine whether the face has a mask on or not. When provided with a surveillance feed, it can recognize both face and face mask even in motion. The proposed model has been tested on a couple of datasets, where it achieved 95.77 percent and 94.5 percent accuracies, respectively, on two open-source datasets. X. Fan and his colleagues implemented a deep learning-based lightweight face mask detection framework utilizing MobileNet CNN architecture in [14]. A Gaussian heat map regression is added to increase the feature learning task of the proposed model. Next, the performance of the implemented network is assessed on two public datasets, viz. AIZOO and Moxa3K. The authors reported improvements in mAP scores by 1.7% and 10.5% of the proposed CNN architecture compared to the YOLOv3 model.

Many authors employed computer vision-based deep learning techniques to monitor social distance and measure the physical spacing between people. Authors in [15] presented a well-planned framework for monitoring social distance through object detection and deep learning models. In this paper, the authors initially accomplish calibration using bird's eye view, where all the pedestrians are assumed to be on the same plane road. Consequently, the distance is estimated between each person concerning the bird's eye view. Next, the YOLOv3 object detection model is used for person detection. Lastly, after the detection, it draws bounding boxes on people to distinguish the individuals violating the social distancing protocols. The dataset is accumulated from Oxford Town Center, containing CCTV footage of 2,200 pedestrians. The result section has observed that in terms of frames per second (FPS), Single Shot MultiBox Detector (SSD) is performing well compared to YOLOv3. However, in the case of mean average precision (mAP), YOLOv3 outperformed SSD. In [16], I. Ahmed implemented a deep learning-based social distance monitoring technique that uses the pre-trained YOLOv3 object detection model and an extra-trained transfer learning technique to improve the model's accuracy. The distance between the people is detected using bounding box detection information. Next, alarms are generated if people violate the minimum distance threshold. Finally, the proposed centroid technique achieved a tracking accuracy of 95%. The accuracy of the detection model is 92%, and with transfer learning, accuracy increases to 95%. In [17], A. Rahim, A. Maqbool and T. Rana presented a well-organized framework for monitoring social distance

in low-light environments through the YOLOv4 model and a single motionless ToF camera. The YOLOv4 model evaluated by COCO detection metrics is trained on the ExDARK dataset with 12 distinct classes of objects. Also, a custom dataset is used for social distance supervision, which is obtained from the market of Rawalpindi, Pakistan. Finally, the results show that the YOLOv4 model achieves the highest accuracy in the low-light environment with a mAP coefficient of 0.9784.

Recently, face mask detection and social distance monitoring have been done simultaneously in some works. As an illustration, in [18], K. Bhambani et al. implemented an efficient system that focuses on three particular objects, i.e., masked and unmasked faces and Euclidean distance between people. The authors used the MAFA dataset and locally linear embedding convolutional neural networks to detect face masks. The MAFA test set achieved an accuracy of 76.4 percent while identifying face masks. Then the authors implemented YOLO object detection and a DeepSORT object tracking modality to track people in the video stream with pixel limitations between 90 to 170. The dataset used in this paper is the renowned COCO dataset that contains a staggering 7,959 images consisting of WIDER Face and MAFA datasets. About 6,120 images were set aside for training from the dataset, whereas 1,839 images were used for validation. Next, a bounding box is created to identify and label people according to the height and width of the image. Finally, the authors have also created their own dataset for detecting social distance and the error of the proposed system increases when the subjects move far from the camera. In [19], the authors implemented an automated system for face mask detection and social distance measurement to address the COVID-19 pandemic. The authors used Faster R-CNN (with 97 percent accuracy) and deep learning techniques for detecting masked faces. They also used YOLOv2 model to estimate the physical separation between two people. Interestingly, even if people wore glasses and scarves or had beard faces, the proposed system's accuracy using Faster CNN is 93.4 percent. As for social distance, 3 meters of distance was set by the authors, and it was detected among pedestrians successfully.

From the above literature reviews, we can conclude that significant works have been done on automatic COVID-19 detection. Artificial intelligence and neural network techniques have been successfully applied in many studies for automatic face mask identification and social distance monitoring. However, most of the works do not consider implementing this automated detection process in an embedded device. Therefore, a simultaneous face mask detection and social distance monitoring system has been proposed in this article employing Jetson Nano edge device and smartphone applications.

3 Proposed System

In the following paragraphs, the required software and hardware components and system architecture of the proposed face mask detection and social distance monitoring network has been discussed. The main objective of the proposed system is to analyze frames from a video stream or clip from any recording source or real-time live stream, then implement detection algorithms to detect whether the COVID-19 precautionary measures are violated or not. It will also implement the optimized model on a remote desktop PC environment with a real-time video feed from an embedded system (Jetson Nano) with a webcam and a smartphone for easy distribution and low-cost architecture integration within a pre-built infrastructure.

The violation measurements of this work depend on a specific set of rules:

a) Are people properly wearing a facial mask or not? This condition is detected by initially using facial features recognition and then executing the facial mask detector on the obtained features.

b) Are people maintaining safe social distances (6 feet minimum from one person to another)? The system follows this step by calculating the distance between the bounding box of objects (single person).

3.1 Software Tools

We employed the following tools and technologies in our system for detecting precautionary measure violations of face mask and social distance, which have been discussed in Tab. 1.

Table 1: Used software tools in this work

Software tools	Functions
Ubuntu 20.10 Groovy Gorilla	Main operating system
JupyterLab text editor	Coding purposes
Python 3	Core programming language
Jetpack software development kit	Supports a newer version of CUDA, cuDNN and Tensor RT for better optimization
Android Studio version 2020.3.1	Captures the video feed onto the remote PC *via* the Android smartphone device
Xcode version 13	Takes the video feed onto the remote PC *via* the iOS smartphone device

3.2 Hardware Components

We employed the following hardware components in our system for detecting precautionary measure violations of face mask and social distance, which have been discussed in Tab. 2.

Table 2: Utilized hardware components in this work

Hardware components	Functions
Desktop PC	In this paper, the experimental hardware platform used as the sole processing unit of the proposed system has an Intel Core i5 8th generation processor, 16GB memory and GTX 1060 graphics card. The CCTV footage is collected through a CCTV camera system and various online CCTV video clips. Some of the footage is taken from a webcam using Jetson Nano and from multiple smartphones possessing two operating systems, *i.e.*, Android and iOS smartphones.
Jetson Nano with webcam	Jetson Nano developer kit of 4GB RAM, with a Fantech 1080P 2MP web camera, is used as a medium for taking real-time video streams for the detection models.
Smartphones	As the size and usability of smartphones for taking video streams are convenient and effective, we have used an Android smartphone (Samsung Galaxy A51) and an iOS smartphone (Apple iPhone 10) to collect real-time video feed. This implementation will enable us to deploy the proposed system detection methods on a broader scale in the future.

3.3 System Architecture

The proposed system's primary function is to monitor people who violate standard physical distances and facial masks protocols using video footage from CCTV cameras. A deep learning approach, YOLOv3 and DBSCAN clustering have been utilized for monitoring social distance. To identify people without a face mask, ResNet50, a convolutional neural network, is employed. Blurring effects and augmented masked faces are also applied to train the proposed model to identify real-life faces instantly. The designed detection system works as a sequence of different tasks, such as person detection, face

identification, face mask classifier, and, finally, clustering detection. Eventually, the proposed network employs violations based on the face mask and proximity clustering recognitions on the detected persons. The working sequences of the proposed social distance monitoring and facial mask recognition system have been demonstrated in Fig. 1.

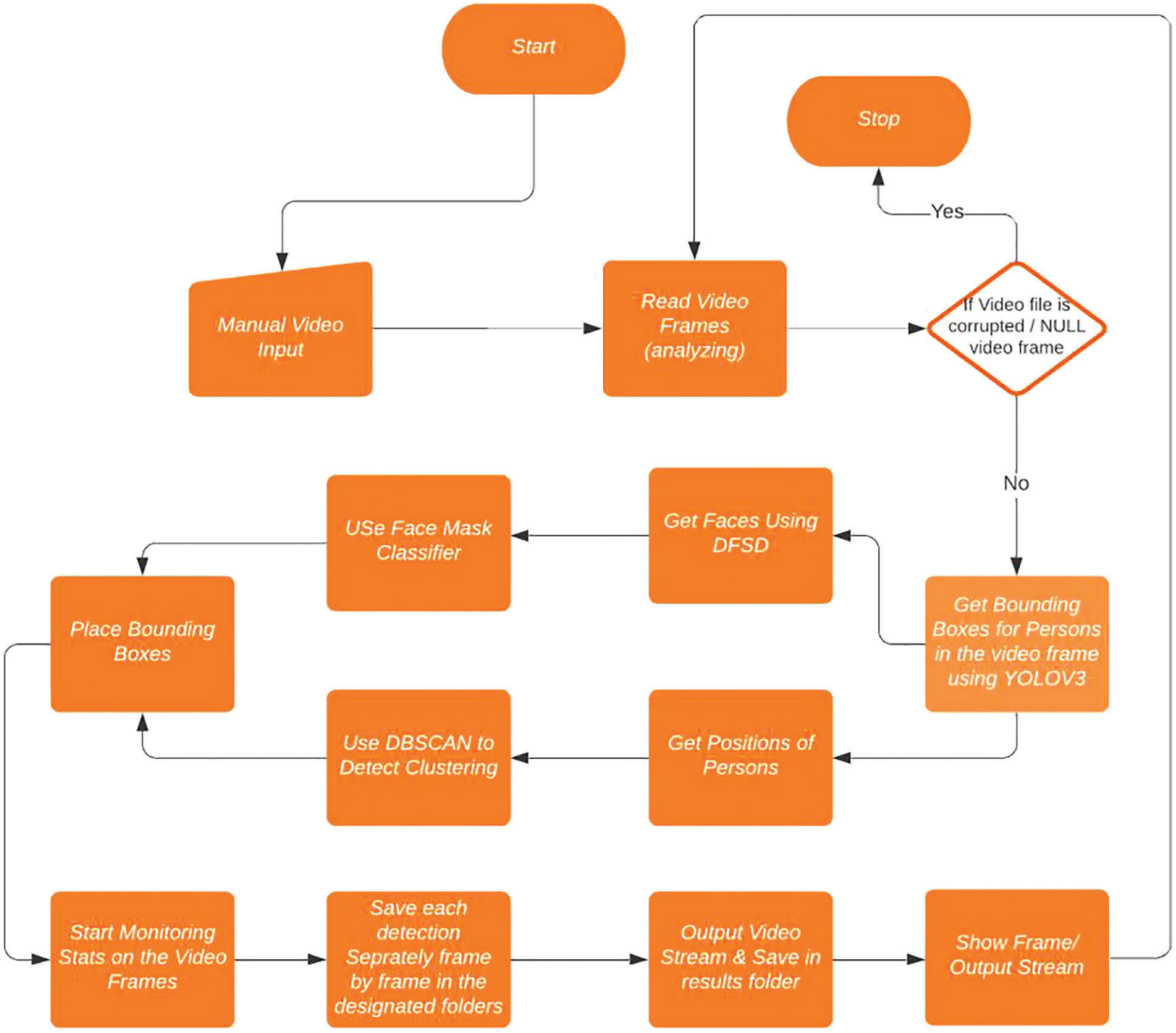

Figure 1: Working sequences of the proposed COVID-19 prevention system

3.3.1 Working Procedure of the Social Distance Monitoring Model

There are various deep learning techniques for automatic object (people) detection, e.g., region-based convolutional neural networks (RCNN), SSD, YOLO, etc. These models offer diverse accuracies concerning mAP scores and frames per second speed, which has been demonstrated in Fig. 2 [20]. High inference speed in tandem with acceptable accuracy needs to be considered for real-time object detection of embedded devices. In this work, the deep learning model, YOLO, has been used for object detection because of its moderate mAP scores and high frame rate, as consequently, it will be executed in real-time edge devices. The specific YOLOv3 framework has been employed in this work because of its outstanding balance between accuracy and detection speed.

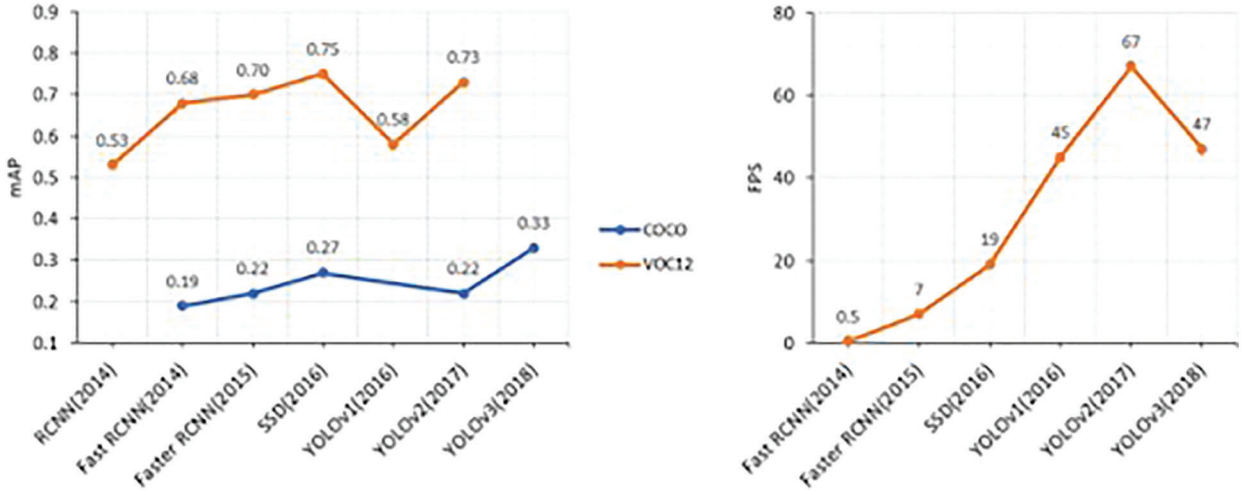

Figure 2: Performance comparison of various object detection models on MS-COCO dataset

YOLO-You Only Look Once is a deep learning-based object detection model that is fast and can detect multiple classes within a dataset [21]. This algorithm uses CNN for detection. Over the years, the YOLO algorithm has been undergoing various optimizations and its latest beta version is YOLOv5. However, YOLOv3 is well known for stable optimization, and consequently, this model has been used for the proposed social distance violation detection. The YOLOv3 network used for physical distance monitoring has been illustrated in Fig. 3. In this paper, the monitoring of social distances with people recognition and tracking are performed with YOLOv3 and DeepSORT approaches.

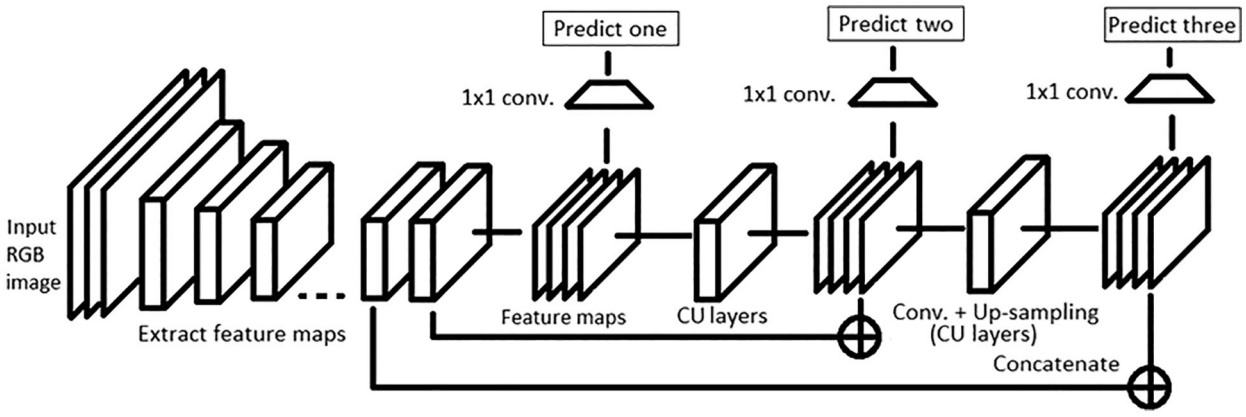

Figure 3: Schematic representation of the proposed YOLOv3 architecture

The social distance measurement requires the usage of the COCO dataset's person key-point localization, where the key points are defined as the value points of a person from uncontrolled images. It targets the localizations of all the points and describes the points as a combination of a detected person. Consequently, it recognizes a person from an image or stream of images known as a video stream. The key-point evaluation metrics used by the COCO database for object detection are average precision (AP) and average recall (AR) and their alternatives. These metrics measure the correlation between ground truth objects and detected objects. For each object, the ground truth key-points are denoted such as [x_1, y_1, v_1, ..., x_k, y_k, v_k], where x, y are the key point locations and v denotes a visibility flag defined as

$v = 0$: unlabeled, $v = 1$: labeled but not visible, and $v = 2$: labeled and visible. The object key-points similarity (OKS) is expressed as:

$$OKS = \frac{\sum i\ [\ e^{\frac{-d_i^2}{2s^2k_i^2}}\ .\delta(v_i > 0)]}{\sum i[\delta(v_i > 0)]} \tag{1}$$

According to (1), the object keypoints similarity detection algorithm is a method for identifying objects based on the accuracy of their key points, which has been used in this work. Here, s indicates object scale, and k_i means a per-keypoint constant that controls falloff. For each keypoint, this generates a keypoint similarity that ranges between 0 and 1 [22]. In (1), Euclidean distance (d_i) measures the spacing between the corresponding ground truth and detected keypoints. On the other hand, v_i indicates the visibility flags of the ground truth (this is separate from the predicted v_i value from the detector). We then pass the d_i value through an unnormalized Gaussian with standard deviation sk_i in order to calculate the OKS function.

Now, going through the video, the model focuses on every frame, identifying, detecting, and labeling (with bounding boxes) objects, which are then stored. The model identifies people by detecting faces. The faces are checked to determine if they are masked or unmasked. By finding faces, the proposed system can identify the presence of people in a single frame and calculate their number. The proposed framework places bounding boxes on people, placing red/green depending on whether they are wearing masks or not. It also places bounding boxes on each person.

$$S_i^{(t)} = \left\{ x_p : ||x_p - \mu_i^{(t)}||^2 \leq ||x_p - \mu_j^{(t)}||^2 \ \forall j, 1 \leq j \leq k \right\} \tag{2}$$

In (2), the K-means clustering algorithm is demonstrated, which is a method for vector quantization. Here, S expresses the objective function, which is calculated from the distance of the points. In (2), the total number of clusters is denoted by k and μ illustrates the centroid for the corresponding cluster. The algorithm classifies people based on similar data points. It checks whether the bounding boxes of one person overlap the bounding box of someone in his cluster and measures their distance using Euclidean distancing [23]. This function expressed in (2) is carried out multiple times till the model can identify whether people have crossed the threshold value for social distancing or not. The working sequences of the proposed social distance monitoring system are:

Algorithm 1: Algorithm of the Proposed Social Distance Monitoring Model

Input: key point locations [x_1, y_1, ..., x_k, y_k]; Visibility flag (v), ($v = 0$): not labeled, ($v = 1$): labeled but not visible, ($v = 2$): labeled and visible, visibility flags of ground truth (v_i), Euclidean distance (d_i), standard deviation (s_i), per-keypoint constant that controls falloff (k_i).

Initialize: for $i = 1$ to k **do**

Compute: OKS using (1)

Apply K-means clustering algorithm for vector quantization using (2)

Enable forward pass and bounding boxes

Employ non-maximum suppression

Output: Detect persons from an image or video stream, Label bounding boxes, Identify whether people have violated social distances or not.

Tab. 3 demonstrates the hyperparameters to train the social distance monitoring framework. In this work, non-maximum suppression (NMS) with a threshold value of 0.30 is chosen for the bounding box of people detection.

Table 3: Parameters used in the social distancing detection model

Parameter	Corresponding value
Confidence Threshold	Confidence Threshold = 0.50, NMS IoU Threshold = 0.30
Distance Threshold (Safe distance in pixel units)	150 Pixel, best suited for our analyzed captured videos
Object Detection Frame Range with YOLOv3	blobFromImage = (0.00392, (416, 416), (0, 0, 0))
Stored Detected Objects Confidences	(0.50, 0.40)
Facemask Classifier Dropout	(0.50)

3.3.2 Working Procedure of the Face Mask Detection Model

The well-known face identification network, dual shot face detector (DSFD), is utilized to detect faces in this work. This improved version of this face detection model with feature learning improvement exhibited better accuracy than a single shot detector [24]. It can detect faces in many orientations with enhanced anchor matching and improved data augmentation techniques, rendering it better than other pre-trained classifiers that work to identify faces. We used a modified version of a convolutional neural network, the ResNet50 model, to identify masked faces in this research [25]. It comprises ImageNet, AveragePooling2D, and dense (with dropout) layers, followed by a sigmoid or softmax classifier. The used DSFD framework uses a feature enhance module where, the module is used on top of a feedforward ResNet architecture as well as two loss layers: first shot PAL for the original features and second shot PAL for the enchanted features to generate the enhanced features.

We artificially put face masks on random face images to enhance the performance of the mask classifier through the deep learning process. In these images we obtained through a deep neural network, some points are found by manipulating facial landmarks, namely the nose bridge and chin. Then face masks are placed to create the auto-generated image. If we encounter blurred faces on video frames, DSFD will mark them correctly. This blurriness may occur because of rapid movements, incorrect camera settings, low light conditions, or grainy footage. As a result, we will need to apply a blurring effect to a random portion of the training data. On a kernel of size, three sorts of effects have been used, i.e., Motion Blur (mimics rapid movement), Average Blur (describes out of focus), and Gaussian Blur (produces random noise).

Finally, ImageDataGenerator function of Keras has been utilized to perform on-the-fly augmentation. The training data is automatically upgraded after every epoch. Additionally, traditional augmentations are applied, such as rotation, horizontal flip, and brightness shift. It is worth mentioning that blurring augmentations are added with particularly associated probabilities during the training stage. Tab. 4 depicts the parameters used in the facemask classifier of the pre-trained ResNet50 model.

Table 4: Parameters used in the facemask classifier on top of pre-trained ResNet50 model

Confidence threshold	Confidence threshold = 0.50, NMS IoU threshold = 0.30
Motion Blur Kernel Range	Motion Blur Kernel Range = (4,8), Average Blur Kernel Range = (3,7), Gaussian Blur Kernel Range = (3,8)
ResNet50 Base Network Input Shape	(224, 224, 3)
Facemask Classifier AveragePooling2D Pool Size	(7, 7)
Facemask Classifier Dense Layer (activation)	(128, activation ReLU) (1, activation_sigmoid)
Facemask Classifier Dropout	(0.50)

3.3.3 Working Procedure of Using Jetson Nano with Webcam for Capturing the Video Stream

The main objective of this research is to provide a detection system that can help with the control and monitoring of the COVID-19 pandemic. The proposed face mask detection and physical distance monitoring system also need to be flexible and easily deployable to achieve this goal. The Jetson Nano embedded system allows us to quickly deploy a remote architecture capable of receiving real-time video feed from various remote locations. As it is an AI-capable device, it provides room for future improvements and video feed optimizations. This paper uses a webcam with a Jetson Nano computing device to receive the captured video feed and analyze the video stream for people detection, face mask identification, and social distance violation feedback. This implementation would require us to use the webcam through Jetson Nano and capture the video while simultaneously sending the captured video stream onto the server, which has been depicted in Fig. 4. Then the remote host computer will receive and execute the detection models and generate relevant results. We are using a desktop personal computer as the processing unit, so we did not require cloud-based resources. We devised a localhost server where the real-time stream was sent, and then the same server would be used on the PC and Jupyter notebook to process and show the results on the monitor.

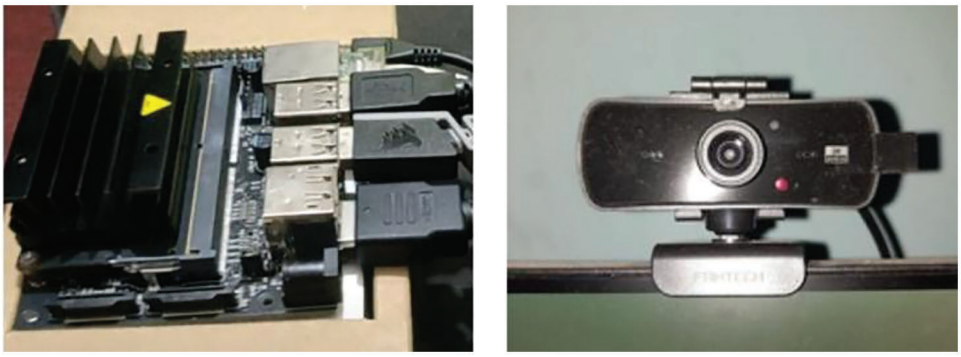

Figure 4: Jetson Nano Setup with webcam

3.3.4 Working Procedure of Smartphone Devices for the Capturing of Video Stream

As technologies improve further each day, smartphones have become widely used devices for all of our everyday life. These devices are capable of adequate processing power and are essential as a portable medium for many on-device real-time video transfers. To ensure the proposed system architecture

provides all the scopes of development, we have also tested Android and iOS smartphones to send captured video streams to the remote desktop computer. Fig. 5 shows the working procedure of the designed Android application.

a) Android Device: To enable an Android smartphone device for taking and transferring video feed directly to the remote PC, we have devised an Android application. The app is configured to record and simultaneously send the video to the connected server that is set up in the remote local network on the PC. The app can also be configured to record video at different resolutions and 30 and 60 frames per second, allowing the user to use the app conveniently at times of poor network connection and faster processing.

b) iOS Device: The exact process has been used for an iOS device (Apple iPhone 10). A separate iOS platform application has been developed to enable the device to stream the video feed to the remote server. The app has been made using Xcode and Swift for the compatibility of iOS devices. Users can configure the app to record video at different resolutions and 30 and 60 frames per second.

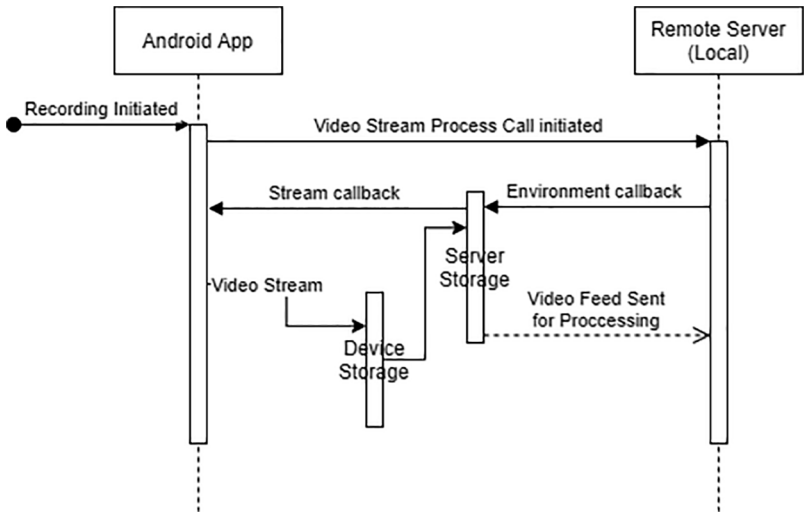

Figure 5: Working procedure of the video streaming application

4 Results and Discussion

This section presents real-time results for the proposed face mask detection and social distance monitoring system. In this paper, two datasets have been used, one combined dataset for the face mask identification model and the COCO (Common Objects in Context) dataset for the social distance violation detection model. For the face mask detection model, the combined open-source dataset contains a total of more than 1,000 pictures. All these pictures are then divided into four different classes, (1) Human face with a facial mask, (2) Human face without a facial mask, (3) Facial mask worn correctly, and (4) Facial mask worn incorrectly.

The COCO dataset is used for the social distance violation model, which is extensively used for large-scale object detection, segmentation, and captioning assignments. COCO dataset consists of more than 300 thousand images where 200 thousand images are labeled. It contains 80 object categories and 250 thousand people with critical points that allow any people detection model to detect a person in a captured frame with high precision [26].

4.1 Results of Face Mask Classifier System

In this work, a modified version of a convolutional neural network, the ResNet50 model, has been employed to identify masked faces. The accuracy and loss vs. epochs of the face mask detection model have been shown in Fig. 6. According to this figure, it can be observed that the proposed network achieved 78.50% validation accuracy. The reason behind the low accuracy is that the face mask dataset was relatively small, with 1,000 images that included both masked and unmasked images.

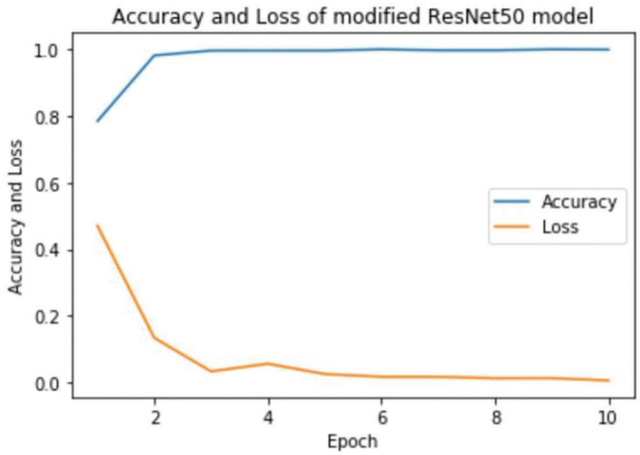

Figure 6: Validation accuracy and loss *vs*. epochs of the modified ResNet50 model

The confusion matrix in Fig. 7 signifies the total results of the proposed face mask detection model, including true positive, false positive, true negative, and false negative values. The matrix below shows that our model correctly predicted 517 masked people and 268 unmasked people. On the other hand, it incorrectly detected 132 people as masked even though they were unmasked. Similarly, it recognized 83 people as unmasked, even though they were masked.

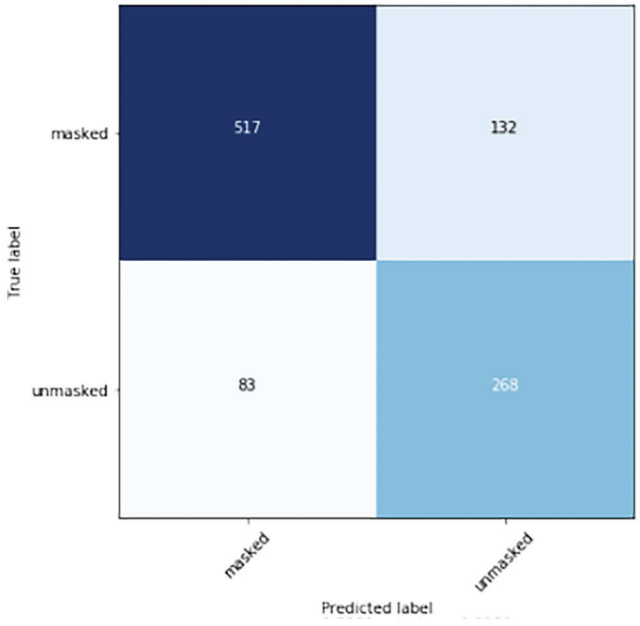

Figure 7: Confusion matrix of the face mask detection model

Precision calculates the ratio between true positive and total (both true and false positive) values. In this case, the positive values are 649 and the accurate positive value is 517. Calculating their ratio, the face mask detection model achieves a precision of 79.66%. We consider this a high value, predicting an impressive number of correctly masked faces. The accuracy of a model determines its effectiveness at predicting correct values compared to total predictions. As we have used a pre-trained model, the obtained accuracy is 78.50%, which is a moderate performance. As expected, the accuracy will improve if a dataset with a higher number of face mask images is used. Finally, from the matrix, it can be noticed that the false positive and false negative values are significantly low compared to the total dataset, respectively 132 and 83. Therefore the F1 score is 82.79%, symbolizing that the proposed model performs relatively well with a low number of false/incorrect predictions.

4.2 Evaluation of Social Distance Monitoring Using YOLOv3

In this paper, the deep learning model, YOLOv3 and DeepSORT technique have been used to monitor social distances with person detection and tracking. In the following paragraphs, the precision of the YOLOv3 detection method in detecting people in the video stream and how this technique has differentiated in terms of accuracy and class recognition compared with various other models are discussed.

For object detection, YOLO predicts the type and location of an RGB image by only looking at one picture at a time. The algorithm considers the detection assignment a regression task instead of a classification one. Next, it assigns the extracted image sectors according to their predicted classes probabilities and binds them to the anchor boxes.

The losses per iteration are measured by applying RPN localization loss, RPN objective loss, and classification with localization loss for the YOLOv3 model. Then the total loss and the overall result with mAP (mean average precision) are determined. The equation for AP (average precision) is given by:

$$AP = \sum_{n} (R_n - R_{n-1}) P_n \tag{3}$$

In (3), the average precision algorithm is expressed, where it is calculated by taking the mean over all classes and/or overall IoU thresholds. Here, denotes the class numbers, and indicate recall and precision, respectively.

Next, we captured ten random frames from the surveillance camera footage to identify the evaluation metrics of the proposed social distance detection model with the YOLOv3 approach. The confusion matrix in Fig. 8 shows that the total number of correctly detected distances is 110, falsely detected distance is 2, and the false-negative value is 27. This measurement helps us ascertain that the implemented model exhibits 79% accuracy, which is significantly high. Moreover, the precision score is 98%, which means that out of the predicted positive values, 98% values were correct. The mean average precision (mAP) of the YOLOv3- based object detection model, computed by (3), is 95%. The recall coefficient of 80% indicates that out of 137 true values, 27 cases were misidentified as false. Finally, the F1 score is 88%, which denotes that the YOLOv3 model performs well.

Finally, we can successfully detect social distance and facial masks simultaneously among the people in frames on the testing videos. Simulation test results of various output frames for the proposed system have been demonstrated in Fig. 9. We can observe that the model detects persons from the video frames and detects whether they are maintaining social distances. By utilizing the K-mean clustering, the implemented model can easily detect clusters of people and mark them with corresponding bounding boxes. The system places green and red bounding boxes depending on whether they maintain the health protocols. The proposed network counts different cases depending on the social distance and face mask detection, i.e., masked, unmasked, safe, unsafe, and unknown.

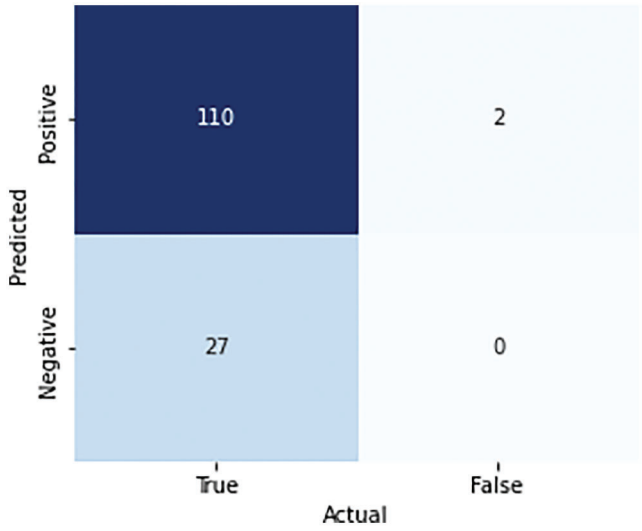

Figure 8: Confusion matrix of the social distance measurement model using YOLOv3

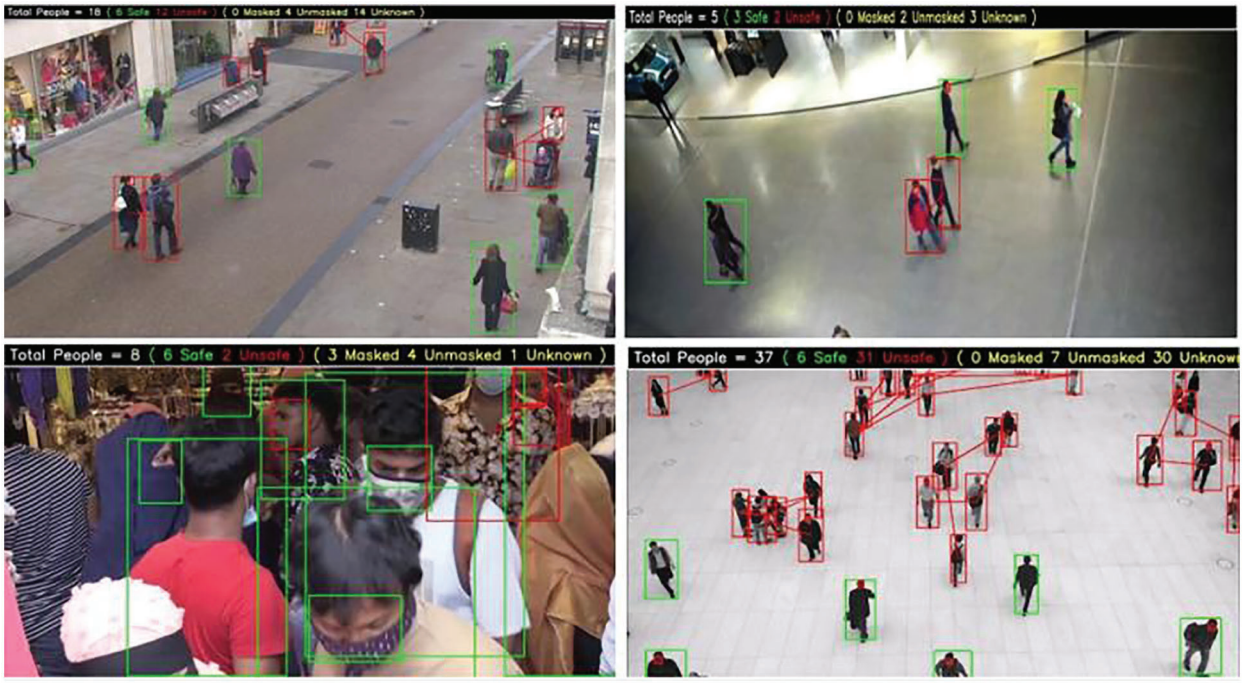

Figure 9: Simulation test results of the proposed social distance and face mask monitoring system

Fig. 10 illustrates an output frame of the proposed system when the processed video is obtained from the Jetson Nano device with a webcam setup.

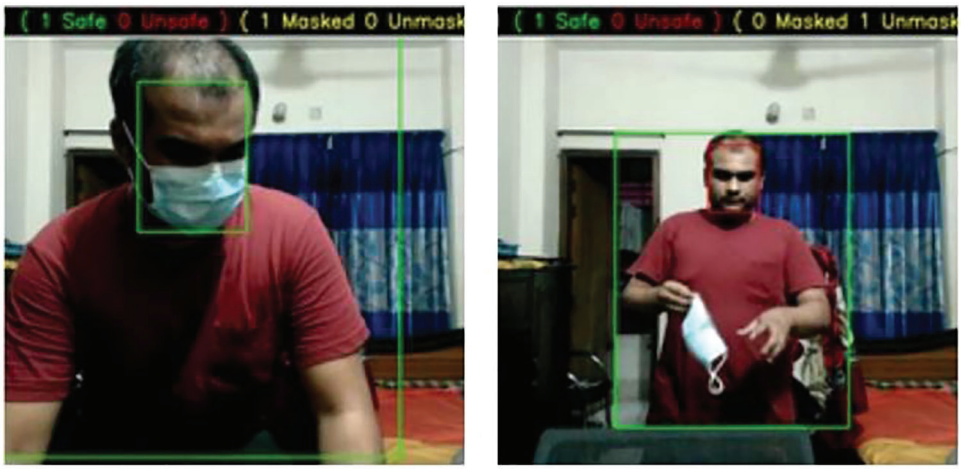

Figure 10: Processed video feed from Jetson Nano with webcam setup

Lastly, the efficiency of the proposed pandemic prevention system is tested for the video sequence captured by the designed application of a smartphone. The qualitative result of this setup is demonstrated in Fig. 11.

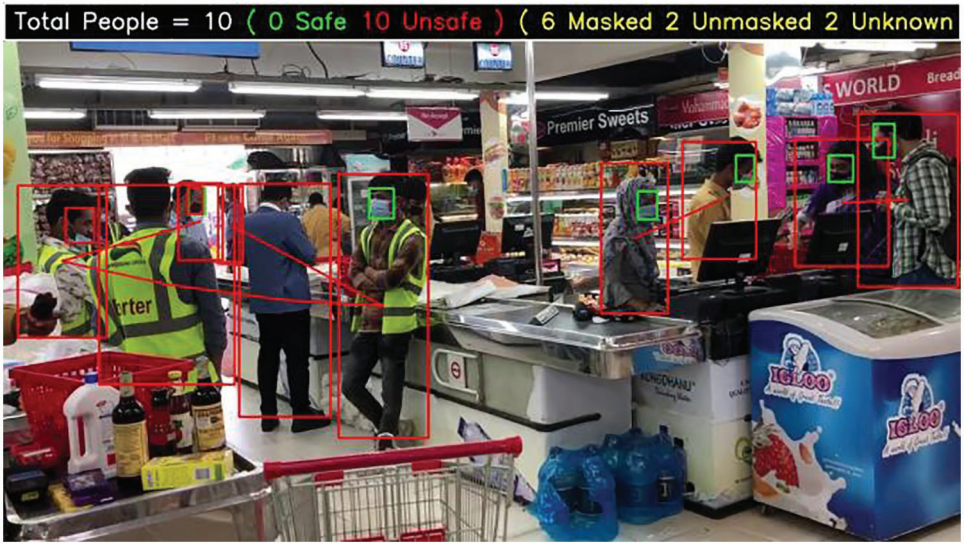

Figure 11: Processed video feed frame, which is captured using the designed application of a smartphone

Tab. 5 demonstrates the comparison of the proposed framework with other similar works. According to Tab. 5, this work exclusively implemented an integrated approach to detect face masks and monitor social distance simultaneously. This simultaneous implementation constrained us to use low-quality real-time video sequences, which lowered the model's performance. Additionally, this work captured instantaneous video frames by utilizing Jetson Nano with a webcam and specialized smartphone applications.

Table 5: Comparison of the proposed framework with other similar works

Reference	Primary features	Applied techniques	Performance metrics for mask detection	Performance metrics for social distance
[12]	Detect face masks	HOGSVM	Accuracy: 69.8%	NA
[13]	Detect face mask	TensorFlow, Keras, OpenCV, Scikit-learn	Accuracy: 95%	NA
[15]	Social distance monitor	YOLOv3	NA	mAP: 91%
[16]	Monitor social distance	YOLOv3, extra-trained with transfer learning	NA	Accuracy: 92%
[17]	Monitor social distance under low light conditions	YOLOv4, ToF camera	NA	mAP: 97.84%
[18]	Face mask and social distance detection in real-time	CNN and YOLOv3	Accuracy: 76.4%	mAP: 94.75%
[19]	Real-time face mask and social distance detection	Faster R-CNN, YOLOv2	Accuracy: 93.4%	NA
Proposed Work	Simultaneous face mask and social distance detection	YOLOv3, DBSCAN clustering, ResNet50, Faster CNN	Accuracy: 82.79%	Accuracy: 79%

5 Conclusions

The main objective of this paper is to develop an automatic system that can monitor the COVID-19 precautionary measures by identifying face masks and measuring physical separation simultaneously. The face mask detection model is implemented on the modified ResNet50 technique. The proposed social distance measuring model employs the deep learning based YOLOv3 object detection and DeepSORT techniques. The performance of the proposed pandemic prevention system is validated on real-time video feeds captured by Jetson Nano embedded tool and webcam and customized smartphone applications for both Android and iOS devices. The implemented object detection model creates bounding boxes, and consequently, red boxes indicate the facial masks and social distance infarction. The performance of the object detection model can be improved by using more advanced artificial intelligence and deep learning techniques, such as multi-layer feedforward BP neural framework, transformer-based and anchor-free modalities [27]. The accuracy of the face mask classifier can be increased by incorporating real-time CCTV footage datasets and hyperparameter tuning. Future improvements may involve using a more lightweight model incorporating a transfer learning framework for the face mask detection algorithm [28]. This detection system can assist healthcare facilities, malls, education centers, and public gathering sites in identifying the preventive measure violation instances and imposing improved safety protocols to restrict the spread of coronavirus.

Funding Statement: The authors would like to thank North South University, Bangladesh, for procuring the Jetson Nano developer kit under the CTRG research grant 2021.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. G. Andersen, A. Rambaut, W. I. Lipkin, E. C. Holmes and R. F. Garry, "The proximal origin of SARS-CoV-2," *Nature Medicine*, vol. 26, no. 4, pp. 1–3, 2020.

[2] T. Singhal, "A review of coronavirus disease-2019 (COVID-19)," *The Indian Journal of Pediatrics*, vol. 87, no. 4, pp. 1–6, 2020.

[3] D. Groff, A. Sun, A. E. Ssentongo, D. M. Ba, N. Parsons *et al.,* "Short-term and long-term rates of postacute sequelae of SARS-CoV-2 infection: A systematic review," *JAMA Network Open*, vol. 4, no. 10, pp. 1–17, 2021.

[4] Q. Wu, M. Dudley, X. Chen, X. Bai, K. Dong *et al.,* "Evaluation of the safety profile of COVID-19 vaccines: A rapid review," *BMC Medicine*, vol. 19, pp. 1407–1416, 2021.

[5] M. Lotfi, M. Hamblin and N. Rezaei, "COVID-19: Transmission, prevention, and potential therapeutic opportunities," *Clinica Chimica Acta*, vol. 508, no. 10223, pp. 254–266, 2020.

[6] M. Shereen, S. Khan, A. Kazmi, N. Bashir and R. Siddique, "COVID-19 infection: Origin, transmission, and characteristics of human coronaviruses," *Journal of Advanced Research*, vol. 24, no. 9393, pp. 91–98, 2020.

[7] H. A. H. Mahmoud, A. H. Alharbi and N. S. Alghamdi, "A framework for mask-wearing recognition in complex scenes for different face sizes," *Intelligent Automation & Soft Computing*, vol. 32, no. 2, pp. 1153–1165, 2021.

[8] S. Anwar, M. Nasrullah and M. J. Hosen, "COVID-19 and Bangladesh: Challenges and how to address them," *Frontiers in Public Health*, vol. 8, pp. 1–8, 2020.

[9] A. Nowrin, S. Afroz, M. S. Rahman, I. Mahmud and Y. -Z. Cho, "Comprehensive review on facemask detection techniques in the context of COVID-19," *IEEE Access*, vol. 9, pp. 106839–106864, 2021.

[10] M. Koklu, I. Cinar and Y. Taspinar, "CNN-based bi-directional and directional long-short term memory network for determination of face mask," *Biomedical Signal Processing and Control*, vol. 71, no. 8, pp. 1–13, 2022.

[11] M. Ansari and D. Singh, "Monitoring social distancing through human detection for preventing/reducing COVID spread," *International Journal of Information Technology*, vol. 13, no. 3, pp. 1255–1264, 2021.

[12] G. Deore, R. Bodhula, V. Udpikar and V. More, "Study of masked face detection approach in video analytics," in *Conf. on Advances in Signal Processing (CASP)*, Pune, India, pp. 196–200, 2016.

[13] A. Das, M. W. Ansari and R. Basak, "COVID-19 face mask detection using Tensorflow, Keras and OpenCV," in *IEEE India Council Int. Conf. (INDICON)*, Delhi, India, pp. 1–5, 2020.

[14] X. Fan, M. Jiang and H. Yan, "A deep learning based light-weight face mask detector with residual context attention and Gaussian heatmap to fight against COVID-19," *IEEE Access*, vol. 9, pp. 96964–96974, 2021.

[15] R. Magoo, H. Singh, N. Jindal, N. Hooda and P. Rana, "Deep learning based bird eye view social distancing monitoring using surveillance video for curbing the COVID-19 spread," *Neural Computing and Applications*, vol. 33, no. 22, pp. 1–8, 2021.

[16] I. Ahmed, M. Ahmad, J. Rodrigues, G. Jeon and S. Din, "A deep learning based social distance monitoring framework for COVID-19," *Sustainable Cities and Society*, vol. 65, pp. 1–12, 2020.

[17] A. Rahim, A. Maqbool and T. Rana, "Monitoring social distancing under various low light conditions with deep learning and a single motionless time of flight camera," *PLoS One*, vol. 16, no. 2, pp. 1–19, 2021.

[18] K. Bhambani, T. Jain and K. A. Sultanpure, "Real-time face mask and social distancing violation detection system using YOLO," in *Bangalore Humanitarian Technology Conf. (B-HTC)*, Bangalore, India, pp. 1–6, 2020.

[19] S. Meivel, K. Devi, S. Maheswari and J. Menaka, "Real time data analysis of face mask detection and social distance measurement using MATLAB," in *Materials Today: Proceedings*, pp. 1–7, 2021. https://www.sciencedirect.com/science/article/pii/S2214785320407606?via%3Dihub.

[20] S. Srivast, A. Divekar, C. Anilkumar, I. Naik, V. Kulkarni *et al.,* "Comparative analysis of deep learning image detection algorithms," *Journal of Big Data*, vol. 8, pp. 1–27, 2020.

[21] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Nevada, USA, pp. 779–788, 2016.

[22] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards realtime object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett (eds.), Vol. 28, Curran Associates, Inc, USA, pp. 91–99, 2015.

[23] A. A. Bushra and G. Yi, "Comparative analysis review of pioneering DBSCAN and successive density-based clustering algorithms," *IEEE Access*, vol. 9, pp. 87 918–87 935, 2021.

[24] J. Li, Y. Wang, C. Wang, Y. Tai, J. Qian *et al.,* "DSFD: Dual shot face detector," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Nevada, USA, pp. 5055–5064, 2016.

[25] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Nevada, USA, pp. 770–778, 2016.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona *et al.,* "Common objects in context," in *Computer Vision–ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (eds.), Cham: Springer International Publishing, pp. 740–755, 2014.

[27] X. R. Zhang, X. Sun, W. Sun, T. Xu and P. P. Wang, "Deformation expression of soft tissue based on BP neural network," *Intelligent Automation & Soft Computing*, vol. 32, no. 2, pp. 1041–1053, 2022.

[28] X. R. Zhang, J. Zhou, W. Sun and S. K. Jha, "A lightweight CNN based on transfer learning for COVID-19 diagnosis," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 1123–1137, 2022.