Neural Computing and Applications (2023) 35:15261–15271
https://doi.org/10.1007/s00521-023-08556-3

ORIGINAL ARTICLE

A new YOLO-based method for social distancing from real-time videos

Mehmet Şirin Gündüz[1] · Gültekin Işık[1]

Received: 26 October 2022 / Accepted: 29 March 2023 / Published online: 7 April 2023

Abstract

The coronavirus disease (COVID-19) is primarily disseminated through physical contact. As a precaution, it is recommended that indoor spaces have a limited number of people and at least one meter apart. This study proposes a real-time method for monitoring physical distancing compliance in indoor spaces using computer vision and deep learning techniques. The proposed method utilizes YOLO (You Only Look Once), a popular convolutional neural network-based object detection model, pre-trained on the Microsoft COCO (Common Objects in Context) dataset to detect persons and estimate their physical distance in real time. The effectiveness of the proposed method was assessed using metrics including accuracy rate, frame per second (FPS), and mean average precision (mAP). The results show that the YOLO v3 model had the most remarkable accuracy (87.07%) and mAP (89.91%). On the other hand, the highest fps rate of up to 18.71 was achieved by the YOLO v5s model. The results demonstrate the potential of the proposed method for effectively monitoring physical distancing compliance in indoor spaces, providing valuable insights for future use in other public health scenarios.

Keywords Real-time video processing · Person detection · Social distance estimation · Deep learning · YOLO

1 Introduction

The new coronavirus disease (COVID-19), which first appeared in 2019, spread very rapidly. It was named a pandemic by the World Health Organization (WHO) because it caused deaths in many countries [1]. COVID-19 disease is transmitted by breathing of droplets spread by coughing, sneezing, and speaking [2]. Various preventive measures have been taken to prevent transmission through droplets. One of these measures is to keep the distance of at least one meter between people [3]. This physical distancing is also called social distancing. The issue of the effectiveness of social distancing in the transmission of COVID-19 disease was studied by Prem et al. [4]. Accordingly, it has been revealed that the infection can spread to large areas in the absence of social distancing.

It is difficult to manually monitor whether social distancing has been violated in crowded, indoor, or outdoor spaces. Therefore, computers can be used to monitor crowded areas with cameras and monitor the correct application of the social distancing rule. For this, (1) the people in the video recording should be detected, (2) the social distance between the detected people should be calculated, and (3) accordingly, those who follow the one-meter social distancing rule and those who do not should be marked with different colors. In this way, people are ensured to comply with the social distancing rule. In a place like a shopping mall, only people in the payment queue can be expected to follow the social distancing rule. Therefore, certain areas such as the payment queue should be marked on the video, and only the people here should be taken into account. The present study introduces a novel technique for detecting social distancing infractions among individuals captured in a designated area of a video frame. Accordingly, instead of the entire area seen in the video frame, only a selected region was handled, and other regions were excluded from the scope.

Object recognition methods are used for detecting and tracking the social distancing between people from the image. In recent years, deep learning techniques for object recognition have become popular. YOLO (You Only Look Once) [5], SSD (Single Shot Detector) [6], and R-CNN (Region-based Convolutional Neural Network) [7] are the most widely used deep learning-based object detection

✉ Gültekin Işık
gultekin.isik@igdir.edu.tr

[1] Department of Computer Engineering, Igdir University, Igdir, Turkey

methods. Deep learning has performed well in recent years, not only for object detection, but also for object classification and other types of detection. Deep learning has proved to be efficacious across various data types and modalities [8–14].

The task of object recognition is to identify and categorize objects within an image or video. This process typically involves two key steps: first, creating a bounding box to isolate the object of interest, and then based on the characteristics of the confined area, determining the class or type of the object. YOLO models for object detection have been giving very successful results in recent years. YOLO is one of the single-stage object detection methods and uses deep learning-based convolutional neural networks. Redmon et al. [5] developed the original YOLO model.

In recent years, there has been a surge of research aimed at examining social distancing measures due to their increased relevance and significance. Deep learning-based YOLO models have been widely employed for detecting individuals and computing the spatial separation between them. Magoo et al. [15] presented a framework for a deep learning and video surveillance application that operates on a bird's eye view perspective. Accordingly, the YOLO v3 [16] was used to determine the places in the image, which they called the key feature points. Punn et al. [17] used the YOLO v3 algorithm with the Deepsort [18] method to detect individuals and computing the spatial separation between them. Their research showcased the effectiveness of the Deepsort algorithm and YOLO v3 in comparison with other object detection techniques.

Open source datasets such as Inria [19], Shangaitech [20], and Caltech [21] have been developed for detecting people from the image and crowd counting. For example, people were counted with the YOLO v3 model using these datasets [22]. The evaluation of the Inria dataset demonstrated a high accuracy rate of 96.1%, while a comparatively lower accuracy of 87.3% was achieved on the Shanghaitech dataset.

In a study [24] using YOLO v4 [23], social distancing was monitored by detecting people with motionless cameras in a low-light environment. The person detection performance of YOLO v3 was investigated using images taken from above [25]. Accordingly, with the YOLO v3 model, people from the top view were detected and counted with an accuracy of up to 95%. Rezaei et al. [26] developed a hybrid method for social distance monitoring by combining classical methods with YOLO v4-based deep neural networks. More recently, [27] used real-time object detection models such as YOLO and SSD for the sustainable development of a smart city. In [28], it is developed a social distancing detection system using YOLOv4-tiny for COVID-19.

The YOLO v5 model [29] was developed in 2020. This model was also used in crowd detection. For example, Purwar & Verma [30] have used YOLO v5 for counting applications in places where the crowd is not dense and obtained successful results. Ren et al. [31] developed an improved variant of YOLO, known as YOLO-PC, which was used to create a people counting system. The use of this novel algorithm resulted in a high level of accuracy.

The following are seen as the main contributions of our proposed approach:

- This work provides a new method for evaluating social distancing violations of people at a defined location on the video frame. Accordingly, instead of the entire area seen in the video frame, only a determined region was handled, and other regions were excluded from the scope. Other studies are generally aimed at detecting all the people seen in the video image. In this respect, a study similar to the approach here has not been found in the literature.
- The developed method adjusts the distance between people based on the camera resolution. The people closest to the camera and the people farthest away do not significantly differ from one another because the average height and width of the individuals are calculated using the developed algorithm.
- People were detected in this study using YOLO models (v3, v4, and v5s) in real-time surveillance videos, and to estimate physical distance, and the performances of these models were analyzed. The test results revealed that the developed social distancing detection method reached high accuracy (acc) and mAP (0.5), and high speed (fps). The YOLO v3 model achieved the highest performance with 87.07% of accuracy and 89.91% of mAP (0.5). The YOLO v5s model, on the other hand, achieved the highest fps rate with 18.71.

The bounding boxes specify the height and width of the people. In cases where the video resolution is poor, people are not clearly visible, or the postures of the people are different in some video frames, the dimensions of the bounding boxes may not accurately correspond to the true height and width measurements. This could be the developed method's major challenge.

2 Materials and methods

This study focuses on assessing the degree to which individuals adhere to social distancing guidelines in densely populated environments, utilizing surveillance cameras to monitor such spaces. YOLO models were implemented to identify people in crowded areas and measure the distance between them.

2.1 YOLO models

YOLO performs object detection quickly and with high accuracy, as it passes the entire image through the neural network at once. It performs object detection operations in real time. The YOLO model is a type of single-stage detector based on a convolutional neural network (CNN). SSD [6] and YOLO models are examples of single-stage detectors. YOLO uses a single convolutional neural network to split an image into grids, with each grid making predictions on bounding boxes and confidence scores. The object class is then determined based on the highest confidence score [32].

YOLO v3 incorporates a new network for extracting features that is connected to the network that was previously used in YOLO v2, known as Darknet-19 [16]. Compared to other versions, YOLO v3 has a more successful classifier network and is constructed using the pretrained Darknet-53 and Imagenet [33]. Compared to YOLO v2, YOLO v3 is slower due to its 53 convolutional layers and 5 residual blocks with multiple residual units. It also utilizes a hybrid approach that merges ResNet components [34].

YOLO v4, an upgraded iteration of YOLO v3, was introduced by Bochkovskiy et al. [23]. The feature extraction model named CSPDarknet53 (Cross Stage Partial Darknet) in YOLO v4 has been shown to give better results [23]. In general, the backbone, neck, and dense prediction are the three distinct building blocks of the YOLO v4 design. Figure 1 shows the YOLO v4 architecture.

The backbone, neck, and head are the three parts of the YOLO v4 architecture. The backbone, which is CSPDarkNet53 [36], extracts features and improves the detection system's accuracy [37]. An intermediate neck layer is included between the backbone and the head to enhance the accuracy of the object classification. The head predicts the class and position of objects, calculates object size, and coordinates of bounding boxes. These details are shown in Fig. 2.

The task of feature extraction from the input image or frame is carried out by the backbone CSPDarknet53 network. The existing layer is split into two layers by the backbone, called DenseNet and CSPDenseNet, which are then combined through a hierarchy. In the YOLOv4 architecture, one of the outputs from the first layer is fed into a dense block, as depicted in Fig. 3a, while the other is directly linked to the next transition layer, as shown in Fig. 3b. The dense block consists of a convolutional layer, Rectified Linear Unit (ReLU), and batch normalization. The feature maps from all earlier levels are used in each layer of the dense block as input to identify the intricate image characteristics.

Jocher et al. [29] developed YOLO v5, which is also partitioned into sections. The backbone section extracts features from the input images, the neck section combines these features to generate a feature map, and the output section identifies objects from the feature maps [39]. The YOLO v5 architecture has two types of partial networks called CSP, which ties the front and rear layers together to reduce model size and increase inference speed [40]. By changing the depth and width of the YOLO model, it is possible to create four models with various parameters, namely YOLO *v5s*, *v5m*, *v5l*, and *v5x*.

In this study, the YOLO v3, v4, and v5s models were employed. The number of layers and other characteristics of the models used [29] are given in Table 1. In general, the detection performances of the models are compared with the average precision (AP) metric. This metric means that the model predicts the object correctly with more than 50% probability. The table uses the *B* to represent the number of boxes utilized for each detection and *C* to represent the number of classes.

2.2 Dataset

The study used the MS COCO dataset [41], which is a collection of 1.5 million images used for image captioning, object detection, and segmentation tasks. There are 80 object categories in the dataset, including the *person* category, which is the focus of this study. Therefore, only images labeled as "person" were used.

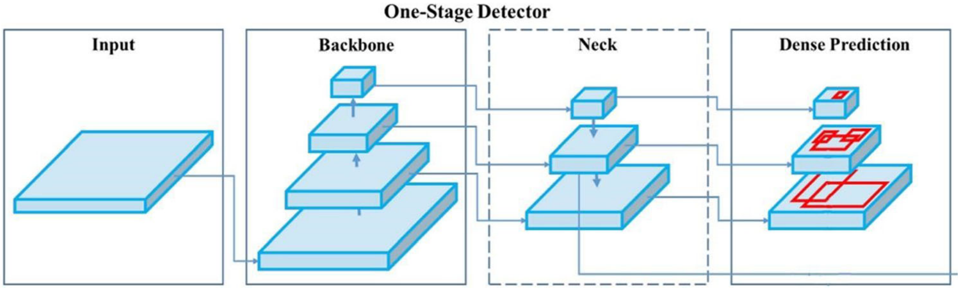


Fig. 1 Architecture of YOLO v4 [23]

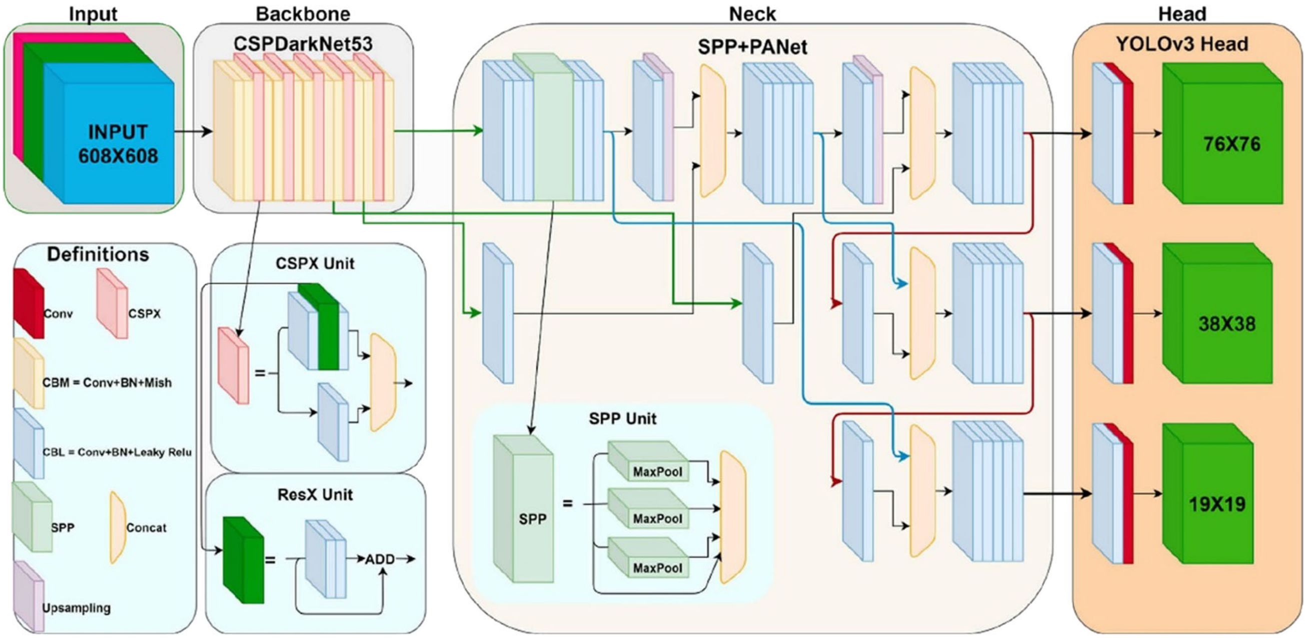
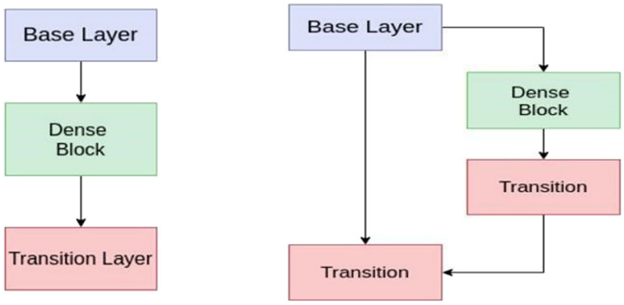


Fig. 2 Main components of YOLO v4 [35]



(a) DenseNet (b) CSPDenseNet

Fig. 3 Comparison of the structures [38] of **a** Standard DenseNet and **b** CSPDenseNet

3 Estimation of social distancing

It is a difficult process to ensure that people comply with the social distancing rule, especially in crowded and closed areas. There is a need to establish an automatic system to detect people who do not comply with this rule. In this study, an algorithm has been developed to detect social distancing violations. To determine whether social distancing has been violated, the distance between people needs to be measured. The Euclidean distance formula (Eq. 1) was used to determine the distance between objects, similar in [15]. Instead of Euclidean distance, Manhattan, Mahalanobis, or Chebyshev distances can also be used. In Eq. 1, x and y represent the coordinates of the two objects. If the estimated distance between two persons is less than a certain threshold value, a distancing violation is detected. The color of people in this situation is marked as red (Eq. 3). The dimensions of the bounding boxes are influenced by the height and shoulder width of each individual. The study utilizes YOLO to detect individuals and extract their bounding boxes, and then calculates the average pixel values for height and width from these boxes. Then, a threshold value is calculated for the corresponding frame by subtracting the shoulder width from the height

Table 1 Characteristics of the YOLO (v3, v4 and v5s) models [29].

	YOLO v3	YOLO v4	YOLO v5s
# of Layers	107	162	224
Weight (MB)	236	245	28.8
AP@0.50	58.2	48.7	56.8
Backbone	DarkNet53	CSPDarkNet53	CSPDarkNet53
Neck	FPN	SPP and PANet	PANet
Head	B x (5 + C)	B x (5 + C)	B x (5 + C)
Neural network type	CNN	CNN	CNN

FPN: Feature pyramid network, *SPP*: Spatial pyramid pooling, *PANet*: Path aggregation network value according to Eq. 2. This value, found in pixels, is assumed as the threshold value between all detected people (pairs) in the corresponding frame of the video. After calculating the threshold value for the frame, the distance between the detected people should be calculated in pairs. To calculate the distance using the YOLO models, at least two people should be detected and enclosed in the bounding box in the video frame. Figure 4 displays the center points of detected persons' bounding boxes. The Euclidean formula in Eq. 1 was utilized to calculate the pixel distance between the center points of these pairs.

$$E = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{1}$$

$$\text{Threshold}\,(T) = \text{height} - \text{width} \tag{2}$$

$$\text{Bounding box} = \begin{cases} \text{Red } if\ E < T \\ \text{Green } if\ E > T \end{cases} \tag{3}$$

The calculated distance and the threshold value of the corresponding frame were then compared. Threshold value and distance calculation processes were carried out on all frames of the video. In Eq. 2, the average threshold value calculation for the corresponding video frame is given. It examined the anthropological characteristics of people in Turkey and found the average height and shoulder width to be 1.65 m and 0.4 m, respectively [42]. When Eq. 2 is applied to these two values, a value of 1.25 m is obtained (Fig. 4). Therefore, it can be thought that the threshold values calculated in pixels in video frames actually correspond to 1.25 m. The reason for using the corresponding threshold value of 1.25 m here is to guarantee a social distancing of at least 1 m between people. In Fig. 5a, different stepping positions of people increased the width of the bounding boxes. The average height and width of both individuals were 50 and 22 pixels, respectively. Accordingly, the threshold value of the people in the image is calculated as 28 pixels (Eq. 2). In Fig. 5b, on the other hand, since people are in a standing position, their bounding boxes are more ideal in terms of aspect ratio. The average height and width of both individuals were found to be 48 and 14 pixels, respectively. The threshold value of the people in this image was also calculated as 34 pixels (Eq. 2). By taking the average of both images in Fig. 5, the threshold value is 31 pixels; thus, the margin of error is reduced. This situation is also reflected in the results in Table 5, and an average of 32 pixels threshold value was calculated at YOLO v3 800 × 600 pixels resolution. This 32-pixel threshold actually corresponds to a distance of 1.25 m (Fig. 4). In order to see the steps followed in the developed method more clearly, a flow chart is given in Fig. 6.

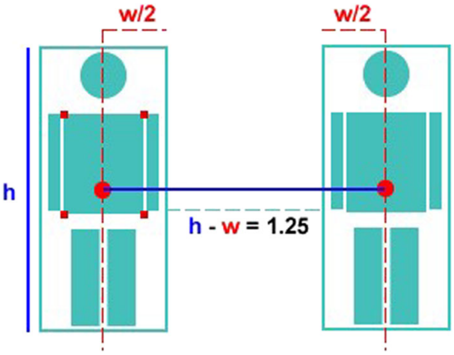


Fig. 4 Center points of the bounding boxes were used to compute the distance between two individuals

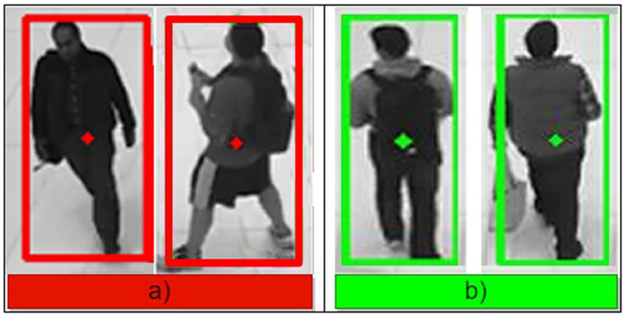


Fig. 5 Height and width values of person pairs (YOLO v3–800 × 600 pixels resolution)

3.1 Advantages of the proposed algorithm

The method developed in this study adjusts the distance between individuals based on the camera resolution, resulting in consistent results across different camera resolutions. Additionally, individuals closer to the camera are larger in height and width, but the developed algorithm

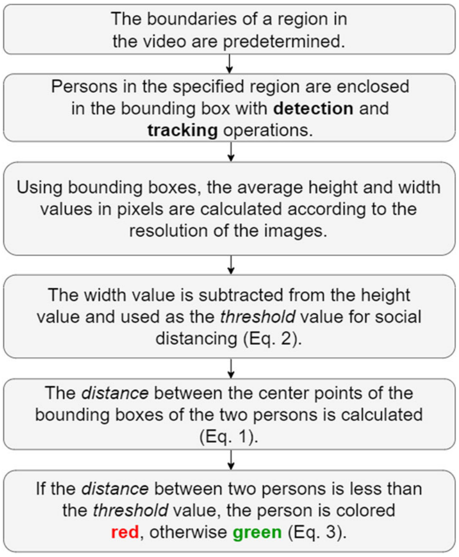

The boundaries of a region in the video are predetermined.
Persons in the specified region are enclosed in the bounding box with **detection** and **tracking** operations.
Using bounding boxes, the average height and width values in pixels are calculated according to the resolution of the images.
The width value is subtracted from the height value and used as the *threshold* value for social distancing (Eq. 2).
The *distance* between the center points of the bounding boxes of the two persons is calculated (Eq. 1).
If the *distance* between two persons is less than the *threshold* value, the person is colored **red**, otherwise **green** (Eq. 3).

Fig. 6 Flowchart to estimate the distance and threshold values

calculates the average height and width of individuals, resulting in no significant differences between those closest to the camera and those farthest away.

3.2 Disadvantages

The size of people in a video is calculated utilizing the bounding box's height and width. However, the accuracy of the size estimation may be affected by factors such as low video resolution, unclear images, and variations in people's postures. In such cases, the calculated height and width may not reflect the actual physical size of the people (Fig. 5a). This situation negatively affects the distance between people in some frames of the video. In such cases, the distance between people may be less than 1 m. In normal video images, the distance between people is sometimes 1 and sometimes more than 1.25 m (Fig. 5b). Since all frames of the video are processed and an average value is calculated here, it is estimated that the distance between people is approximately 1 m. In this way, the margin of error is minimized.

4 Experimental study

The study utilized the Python programming language and OpenCV library. The Deepsort algorithm was used to track people.

The object classification in the Deepsort algorithm is performed using a CNN model. The Deepsort algorithm assigns appearance feature vectors to each object detected by the neural network, allowing it to associate objects across frames. This approach avoids the need to process the entire video at once and instead considers both previous and current frames. The neural network is trained iteratively until satisfactory results are obtained.

In order to be able to track the objects, the object should be detected first. Object detection is a process that involves locating objects through the use of bounding boxes, while also determining their location, type, and class. In this study, YOLO learns to detect the class label *person* by examining the traits of individuals. After learning, YOLO separates the image into bounding boxes and identifies the person if they are present within the grid. If there are multiple bounding boxes for a single person, YOLO reduces the number to one by utilizing the Non-maximum Suppression (NMS) method [43]. In this way, people in the image are detected by bounding boxes.

In this study, the value calculated by subtracting the width from the height of the bounding boxes belonging to the people detected in a certain area in each frame of the video is used as the threshold value between the people. When the distance between two people is less than the threshold value, the people are colored red; otherwise, they are colored green. The images in Fig. 7 are samples taken from the real-time experiments. Appropriate colorings in the predetermined region can be seen from the samples. The images in Fig. 7 were taken from 800 × 600 and 1024 × 768 resolution videos using the YOLO v5s model.

To ensure consistent and comparable results, all parameters other than the YOLO models were kept constant throughout this study. For example, all experiments were performed on videos with the same resolution. Using the videos given in Table 2, people were detected in an area with predetermined borders and colored according to the social distancing rule. Table 2 summarizes the video testing details, including the number of frames, duration, and resolution.

4.1 Experiment environment

Deep learning applications usually require powerful GPUs, which can be expensive. YOLO and similar models typically operate at a slow speed of 1–3 FPS when running on a CPU. To overcome this limitation, GPUs are often used. In this study, the computer used had an Intel Xeon processor with a 2.20 GHz frequency and 6 MB cache, an NVIDIA Tesla K80 graphics processor, and 16 GB memory. Additionally, CUDA and cuDNN were installed to enable the models to run on the GPU.

4.2 Evaluation metrics

Various measures are employed in this research to evaluate the effectiveness of the proposed approach in detecting individuals in videos and estimating the distance between them in real time. These include:

4.2.1 Accuracy rate

The accuracy metric measures the closeness of the estimated results to the actual ground truth, and a high value is desirable. The formula for calculating the accuracy rate is provided in Eq. 4, which involves the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values.

$$\text{Accuracy Rate} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4}$$

4.2.2 Mean average precision (mAP)

This is a widely used evaluation metric for object detection systems that measures a model's performance by averaging the average precision (AP) for each class over several classes. The mAP provides a single numerical value for

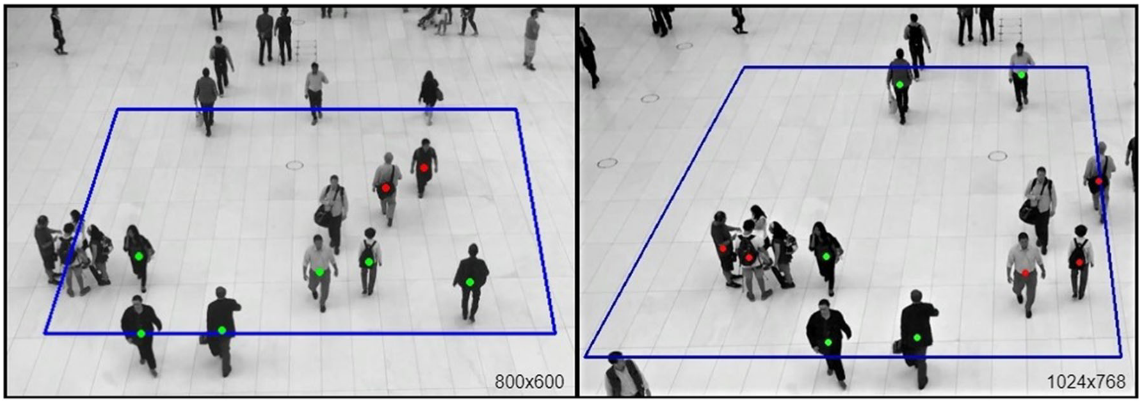


Fig. 7 Coloring of the detected people in a certain area according to the social distancing rule

Table 2 Properties of the videos [44] used in the test phase

Test Data	# of Frames	Duration (sec)	Resolution
Video-1	341	13.64	800 × 600–1024 × 768
Video-2	375	15.00	800 × 600–1024 × 768
Video-3	359	14.36	800 × 600–1024 × 768

assessing the performance of a model. The formula for calculating mAP is given as follows:

$$mAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \quad (5)$$

where Q denotes the total number of queries and q represents the average precision of a query. In this study, as there is only one class, which is "person," the number of classes is considered as 1 in the calculation of the metric.

The metric mAP (0.5) is calculated by considering a confidence threshold of 0.5 for Intersection over Union (IoU) when computing the average precision (AP). A prediction is considered correct if the IoU is greater than or equal to 0.5.

4.2.3 Frames per second (FPS)

FPS is a common metric in real-time applications, measuring the number of images which can be handled in a second using a trained deep learning algorithm. Real-time applications require high FPS for quick processing.

5 Results and discussions

The dimensions of the input image for the YOLO v3 and v4 models are 416 × 416 pixels. In the YOLO v5s model, the input image is 640 × 640 pixels. The YOLO v5s model, which is written using PyTorch framework, is not compatible with OpenCV. To run this model, it needs to be converted to the ONNX (Open Neural Network Exchange) format, which is supported by OpenCV.

The YOLO models' bounding box sizes were considered during the calculation of social distancing between pairs of individuals. According to Eq. 2, the average threshold values obtained in pixels for 800 × 600 and 1024 × 768 resolutions are shown in Table 3. Images with a resolution of 1024 × 768 pixels are larger than images with a resolution of 800 × 600 pixels by 63.84%. The bounding boxes' height and breadth are measured in pixels and increase along with the video's resolution. The threshold value between pairs at 800 × 600 pixels was calculated as 32, 41 and 34 pixels for the YOLO v3, v4 and v5 model, respectively. Likewise, the threshold values of 59, 63 and 57 pixels were calculated for these models at 1024 × 768 pixels, respectively. These threshold values are used as the mandatory distance between people in the indicated YOLO models. While the YOLO v4 model produced the bounding boxes larger than they should be in both resolution types, the v3 and v5s models produced values close to each other.

The characteristics of the YOLO models and the average performance results over the 3 videos used for testing are given in Table 4. Accordingly, videos were read at an average of 9.51 FPS with YOLO v3 and 18.71 FPS with YOLO v5s. Despite having the lowest FPS value, YOLO v3 achieved the highest accuracy rate of 87.07%.

On average, the YOLO v3 algorithm took 35.14 s to process the test videos, indicating that it may have a performance issue in terms of speed. To eliminate this problem, the YOLO v5s model has been developed. The YOLO v5s read the test videos in an average of 17.48 s, reaching 18.71 FPS. The YOLO v5s algorithm predicted the people in the video with an average accuracy of 73.70%. The YOLO v5s model has undergone changes in its pre-trained weight size, which has been reduced to 28.8 MB, and the number of layers has been increased to 224. According to

Table 3 Average threshold values calculated in pixels for social distancing in YOLO models

	800 × 600 resolution (in pixels)	1024 × 768 resolution in pixels)
YOLO v3	32	55
YOLO v4	41	63
YOLO v5s	34	57

Table 4 Comparison of the YOLO models for person detection. (CIoU: Complete IoU)

Model	Loss F	Activation F	Accuracy (%)	Video processing time (S)	FPS	mAP(0.5) (%)
YOLO v3	CIoU	ReLu	87.07	35.14	9.51	89.91
YOLO v4	CIoU	Leaky ReLu	63.79	33.42	10.49	63.54
YOLO v5s	CIoU	Leaky ReLu	73.70	17.48	18.71	78.88

the mAP(0.5) metric, it is seen that the YOLO v3 model achieved 89.91%, giving better results than other models. Additionally, according to Table 4, YOLO v4 performed worse than other models in all metrics. In terms of mAP, FPS, and accuracy metrics, YOLO v3 and v5s models are superior to one another. Depending on the application, a tradeoff should be made between the two models. V3 should be used if high accuracy and mAP are desired, while v5s should be used to achieve close to real-time (i.e., high speed) performance.

To implement the social distancing rule, accurate identification of people in the video is necessary. To achieve this, the study automatically counted the total number of people in the specific regions of the videos listed in Table 2 and evaluated the detection metrics of the people, as presented in Table 5. Here, the following procedure was followed to automatically count people: 10 experiments were conducted by changing the IoU and confidence threshold values for YOLO in the range of 0.25–0.75, and the results of these experiments were averaged. The result obtained here gives the average number of people. Afterward, the consistency of the results was observed by visual inspection. Some people could not be detected by YOLO models even when the values of IoU and the confidence threshold were set to greater than 0.5.

The number of people in the predetermined area is the average number of people in the 3 videos, and the results are shown in Table 5. Accordingly, when the resolution increases, it is seen that the models give more successful results in person detection. While the person detection success of the YOLO v5s model at 800 × 600 pixels resolution was 70.68%, this score increased to 76.72% at 1024 × 768 pixels resolution. At 800 × 600 pixels resolution, the YOLO v5s model could not detect 34 persons. When the resolution increased, it was the YOLO v5s model that provided the highest increase. YOLO v3 was the most successful model for person detection in both resolutions. Since the FPS value of the YOLO v5s model is high, it can be thought that while reading the videos frame by frame, it cannot fully perform the person detection due to the fast transition in some frames. It is more likely that the difference in accuracy between YOLO v3 and v5s is due to differences in the architecture of the models or the data used to test them.

RetinaNet [45] and SSD [6] models were also evaluated for comparison in addition to the YOLO models used in this study. RetinaNet and SSD are single-stage detectors that are trained on the MS COCO dataset like YOLO. Therefore, comparisons can be made between them.

Figure 8 compares the performance of YOLO models with state-of-the-art models such as SSD and RetinaNet on the test videos in Table 2. Accordingly, in terms of mAP metric, 89.52% and 90.31% values were obtained with SSD and RetinaNet, respectively. These values are very

Table 5 Social distancing compliance status of people detected by YOLO models at different resolutions

	800 × 600 pixels			1024 × 768 pixels		
	YOLOv3	YOLOv4	YOLOv5s	YOLOv3	YOLOv4	YOLOv5s
# of people following social distancing (green)	59	51	57	58	55	60
# of violators of social distancing (red)	41	21	25	44	21	29
# of undetected persons	16	44	34	14	40	27
Person detection and social distancing performance (%)	86.21	62.06	70.68	87.93	65.52	76.72

close to the mAP of YOLO v3. In addition, in terms of FPS metric, again YOLO v5s had the highest value by far. SSD and RetinaNet have FPS of 7.45 and 6.08, respectively.

In this study, social distancing estimation was made using YOLO object detection models from real-time videos. However, there are other approaches to obtaining more precise results. For example, social distancing measurement in indoor spaces was made using a thermal sensor array in [46]. In addition, social distancing can be measured closer to reality with Internet-of-Things tools in [47, 48].

6 Conclusion

The COVID-19 pandemic has introduced new rules that were not present before. The social distancing rule is one of the new rules that has emerged. According to this rule, there must be a distance of at least 1 m between people in indoor spaces. Visually detecting social distancing violations is not scalable. Therefore, the detection of social distancing violations should be automated with the help of cameras that monitor indoor areas. The study proposes a Python-based method for detecting violations of social distancing using YOLO models and the OpenCV library. The method is based on the size of bounding boxes generated by the YOLO models. The size threshold is set based on the average height and shoulder width of people in real life. The calculated threshold value is in pixels and corresponds to a real-life distance of 1.25 m. In addition, the distance between person pairs is also calculated in pixels according to the Euclidean distance. By comparing these two values, it was decided whether people comply with social distancing.

The proposed method is based on the size of the bounding boxes obtained from the detected people in the video frames. The method adjusts the threshold value, which represents the mandatory distance between people, based on the resolution of the video. This minimizes the margin of error for social distancing violation detection. Based on test videos, the YOLO v5s showed higher frames per second (FPS) than others. Both the YOLO v3 and v5s models are superior to one another in terms of mAP, FPS, and accuracy metrics. A trade-off between the two models should be made based on the application. If high accuracy and mAP are desired, v3 should be used, while v5s should be used to achieve performance that is nearly real time.

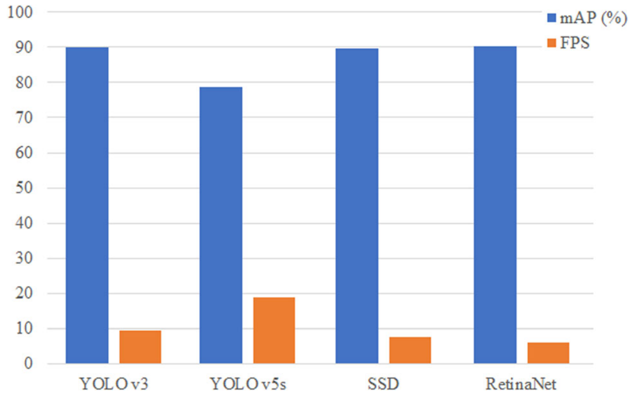


Fig. 8 Comparison of well-performing YOLO models in this study with SSD and RetinaNet

In real-time videos, analysis processes are slower as parameters such as camera features, computer performance, and Internet speed are also taken into account. Results were found to be better in images with higher resolutions. For quick scanning, this study focused only on the "person" category instead of scanning all objects in the MS COCO dataset. This approach can make models more efficient and targeted for specific needs. However, for real-time object detection, the YOLO v3 and v4 models may perform poorly on low-equipped systems, and it is suggested to run them on more powerful hardware. The YOLO v5s model has a small weight size and a high number of layers, yet it showed satisfactory results in terms of accuracy, video reading time, and FPS metrics. This improvement in performance is due to the enhanced layer structure. As a result, both low- and high-performance devices can be used with the YOLO v5s model.

YOLO is a popular object detection model. This is because it is fast, efficient, and easy to use. Additionally, it has been widely adopted in a variety of real-world applications and has been the basis of many other object detection models.

The proposed method for automated detection of social distancing violations has practical implications for public health officials and policymakers, as it can be an effective tool to monitor and enforce compliance. Additionally, businesses and organizations that are required to ensure social distancing in their premises, such as supermarkets, schools, and offices, can benefit from this method. The real-time detection of social distancing violations can also be useful for preventing the spread of other infectious diseases that require physical distancing, such as the flu. Moreover, the method can be extended and applied to other scenarios, such as monitoring crowd density in public spaces or detecting anomalies in traffic patterns.

However, as computer vision technologies become more accessible, there is a need to address ethical considerations and guidelines for their deployment. It is crucial to ensure that the use of automated surveillance systems does not result in privacy violations or discrimination based on race, gender, or other factors. This study can contribute to the ongoing discussion on how to balance the benefits and risks of using automated surveillance systems.

In future research, efforts will be made to improve the accuracy of measuring the distance between people and objects, similar to the method used in this study. This will allow for more realistic distance estimates to be made. The effectiveness of various types of object detection models, including single-stage models like YOLO, SSD, RetinaNet, and two-stage models such as R-CNN, Fast R-CNN, and Faster R-CNN, will be evaluated and compared to determine their suitability for addressing the problem at hand. It is also considered to use newer versions of the YOLO model.

Data availability The datasets analyzed during the current study are available in the [Social-Distancing] repository, github.com/msgndz/Social-Distancing.

Declarations

Conflict of interest The authors of this study declare that they did not receive any support or funding from any organization for the submitted work. They also declare that they have no financial interests that could have influenced the outcome of this study. Additionally, the authors declare that they have no conflicts of interest that are relevant to the content of this article.

References

1. WHO. (2020). WHO Director-General's opening remarks at the media briefing on COVID-19, https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19—11-march-2020
2. WHO. (2019). Coronavirus disease (COVID-19): How is it transmitted? https://www.who.int/emergencies/diseases/ novel-coronavirus-2019/question-and-answers-hub/q-a-detail/ coronavirus-disease-covid-19-how-is-it-transmitted
3. WHO. (2020). Overview of public health and social measures in the context of COVID-19: interim guidance, 18 May 2020 (No. WHO/2019-nCoV/PHSM_Overview/ 2020.1). World Health Organization.
4. Prem K, Liu Y, Russell TW, Kucharski AJ, Eggo RM, Davies N, Klepac P (2020) The effect of control strategies to reduce social mixing on outcomes of the COVID-19 epidemic in Wuhan China: a modelling study. Lancet Public Health 5(5):e261–e270
5. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
6. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: Single shot multibox detector. In European conference on computer vision Springer, Cham, pp 21–37
7. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 1440–1448.
8. Akhtar N, Mian A. (2018) Threat of Adversarial Attacks On Deep Learning İn Computer Vision: A Survey. Içinde IEEE Access (C. 6, ss. 14410–14430). Institute of electrical and electronics engineers Inc.
9. Young T, Hazarika D, Poria S, Cambria E (2018). Recent Trends İn Deep Learning Based Natural Language Processing [Review Article]. Içinde IEEE Computational Intelligence Magazine (C. 13, Sayı 3, ss. 55–75). Institute of Electrical and Electronics Engineers Inc.
10. Chen WY, Liu YC, Kira Z, Wang YCF, Huang JB. (2019) A closer look at few-shot classification. arXiv preprint arXiv:1904.04232.
11. Purwins H, Li B, Virtanen T, Schlüter J, Chang SY, Sainath T (2019) Deep learning for audio signal processing. IEEE J Selected Topics Signal Process 13(2):206–219
12. Işık G, Artuner H (2020) Turkish dialect recognition using acoustic and phonotactic features in deep learning architectures. Int J Inform Technol 13:207–216
13. Mayer C, Nelson M (2020) Phonotactic learning with neural language models. Proceed Soc Comput Linguist 3(1):149–159
14. Bayat S, Işık G (2022) Recognition of aras bird species from their voices with deep learning methods. J Inst Sci Technol 12(3):1250–1263
15. Magoo R, Singh H, Jindal N et al (2021) Deep learning-based bird eye view social distancing monitoring using surveillance video for curbing the COVID-19 spread. Neural Comput Appl 33(22):15807–15814
16. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. arXiv preprint arXiv: 1804.02767
17. Punn NS, Sonbhadra SK, Agarwal S, Rai G (2020) Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques. arXiv preprint arXiv:2005.01385.
18. Wojke N, Bewley A, Paulus D (2017) Simple online and realtime tracking with a deep association metric. In: 2017 IEEE International Conference on Image Processing (ICIP).IEEE. pp. 3645–3649
19. Dalal N, Triggs B (2005). Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05).IEEE. Vol. 1. pp. 886–893
20. Zhang Y, Zhou D, Chen S, Gao S, Ma Y (2016) Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 589–597
21. Dollar P, Wojek C, Schiele B, Perona P (2011) Pedestrian detection: An evaluation of the state of the art. IEEE Trans Pattern Anal Mach Intell 34(4):743–761
22. Menon A, Omman B, Asha S (2021) Pedestrian counting using Yolo V3. ın: 2021 International conference on ınnovative trends in ınformation technology (ICITIIT). IEEE. pp. 1–9
23. Bochkovskiy A, Wang CY, Liao HYM (2020) Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv: 2004.10934.
24. Rahim A, Maqbool A, Rana T (2021) Monitoring social distancing under various low light conditions with deep learning and a single motionless time of flight camera. PLoS ONE 16(2):e0247440
25. Ahmad M, Ahmed I, Adnan A (2019) Overhead view person detection using YOLO. In: 2019 IEEE 10th annual ubiquitous computing, electronics & mobile communication conference (UEMCON). IEEE pp. 0627–0633
26. Rezaei M, Azarmi M (2020) Deepsocial: Social distancing monitoring and infection risk assessment in covid-19 pandemic. Appl Sci 10(21):7514
27. Shorfuzzaman M, Hossain MS, Alhamid MF (2021) Towards the sustainable development of smart cities through mass video surveillance: A response to the COVID-19 pandemic. Sustain Cities Soc 64:102582
28. Saponara S, Elhanashi A, Zheng Q (2022) Developing a real-time social distancing detection system based on YOLOv4-tiny and bird-eye view for COVID-19. J Real-Time Image Proc 19(3):551–563

29. Jocher G, Changyu L, Hogan A, Yu L, Rai P, Sullivan T (2020) ultralytics/yolov5: Initial Release (v1.0). Zenodo. https://doi.org/10.5281/zenodo.3908560
30. Purwar RK, Verma, S (2022) Analytical study of YOLO and ıts various versions in crowd counting. ın: Intelligent data communication technologies and ınternet of things. Springer, Singapore. pp. 975–989
31. Ren P, Fang W, Djahel S (2017) A novel YOLO-Based real-time people counting approach", In: 2017 international smart cities conference (ISC2). IEEE. p. 1-2.
32. Shinde S, Kothari A, Gupta V (2018) YOLO based human action recognition and localization makale. Procedia Comput Sci 133:831–838
33. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) ImageNet: A large-scale hierarchical ımage database. IEEE computer vision and pattern recognition (CVPR)
34. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778
35. Pacal I, Karaboga D (2021) A robust real-time deep learning based automatic polyp detection system. Comput Biol Med 134:104519
36. Jiang Z, Zhao L, Li S, Jia Y (2020) Real-time object detection method based on improved YOLOv4-tiny. arXiv preprint arXiv:2011.04244.
37. Yu J, Zhang W (2021) Face mask wearing detection algorithm based on improved YOLO-v4. Sensors 21(9):3263
38. Wang CY, Liao HYM, Wu YH, Chen PY, Hsieh JW, Yeh IH (2020) CSPNet: A new backbone that can enhance learning capability of CNN. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 390–391
39. Mastromichalakis S (2020). ALReLU: A different approach on Leaky ReLU activation function to improve Neural Networks Performance. arXiv:2012.07564 .
40. Zhou F, Zhao H, Nie Z (2021) Safety Helmet Detection Based on YOLOv5, In: 2021 IEEE ınternational conference on power electronics, computer applications (ICPECA), pp. 6–11.
41. Lin TY, Maire M, Belongie S, Bourdev L, Girshick R, Hays J, Perona P, Ramanan D, Zitnick CL, Dollár P. (2015) Microsoft COCO: common objects in context. computer vision and pattern recognition, pp 2–8.
42. Ali İ, Arslan N (2009) Estimated anthropometric measurements of Turkish adults and effects of age and geographical regions. Int J Ind Ergon 39(5):860–865. https://doi.org/10.1016/j.ergon.2009.02.007
43. Hosang J, Benenson R, Schiele B (2017) Learning non-maximum suppression. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4507–4515.
44. People Passing By-Free HD Stock Footage - People Walking Mall - YouTube, [online] Available: https://www.youtube.com/watch?v=L3-XaUzaJ HU.
45. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017). Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988.
46. Naser A, Lotfi A, Zhong J (2021) Towards human distance estimation using a thermal sensor array. Neural Comput Appl 15:1–11
47. Friji H, Khanfor A, Ghazzai H, Massoud Y (2022) An end-to-end smart IoT-driven navigation for social distancing enforcement. IEEE Access 10:76824–76841
48. Giuliano R, Innocenti E, Mazzenga F, Vegni AM, Vizzarri A (2021) IMPERSONAL: an IoT-aided computer vision framework for social distancing for health safety. IEEE Int Things J 9(10):7261–7272

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.