SSDT: Distance Tracking Model Based on Deep Learning

Original Scientific Paper

Chhaya Gupta
Department of Computer Science & Applications
Maharshi Dayanand University, Rohtak
Haryana, India
chhayagupta.spm@gmail.com

Nasib Singh Gill
Department of Computer Science & Applications
Maharshi Dayanand University, Rohtak
Haryana, India
nasib.gill@mdurohtak.ac.in

Preeti Gulia
Department of Computer Science & Applications
Maharshi Dayanand University, Rohtak
Haryana, India
preeti.gulia81@gmail.com

Abstract – *Coronavirus disease (COVID-19) is an infectious disease caused by the SARS-CoV-2 virus and population vulnerability increased all over the world due to lack of effective remedial measures. Nowadays vaccines are available; but in India, only 18.8% population has been fully vaccinated till now. Therefore, social distancing is only precautionary norm to avoid the spreading of this deadly virus. The risk of virus spread can be avoided by adhering to this norm. The main objective of this work is to provide a framework for tracking social distancing violations among people. This paper proposes a deep learning platform-based Smart Social Distancing Tracker (SSDT) model which is trained on MOT (Multiple Object Tracking) datasets. The proposed model is a hybrid approach that is a combination of YOLOv4 as object detection model merged with MF-SORT, Kalman Filter and brute force feature matching technique to distinguish people from background and provide a bounding box around these. Further, the results are also compared with another model, namely, Faster- RCNN in terms of FPS (frames per second), mAP(mean Average Precision) and training time over the dataset. The results show that the proposed model provides better and more balanced results. The experiment has been carried out in challenging conditions including, occlusion and under lighting variations with mAP of 97% and a real-time speed of 24 fps. The datasets provide numerous classes and from all the classes of objects, only people class has been used for identifying people in a closet. The ultimate goal of the model is to provide a tracking solution that will be helpful for different authorities to redesigning the layout of public places and reducing the risk. This model is also helpful in computing the distance between two people in an image and the results confirm that the proposed model successfully distinguishes between individuals who walk too close or breach the social distancing norms.*

Keywords: *Deep Learning, COVID-19 Social Distancing, MF-SORT, Object Detection, People Detection, Bounding Box*

1. INTRODUCTION

COVID-19 was initiated from Wuhan, China and had affected many countries worldwide. WHO (World Health Organisation) had declared it as a pandemic [1]. Coronavirus is an infectious disease that causes acute respiratory syndrome. There are few more symptoms of this disease that are commonly found in people such as cold, cough, fever, trouble in breathing, body aches, loss of smell and taste. This virus spreads mostly when people come in contact with an infected person. This virus also spreads via air when an infected person sneezes or coughs. As per doctors and researchers, social distancing is an effective method for avoiding the spread of this deadly virus. As there is an increase in death rate and active cases, it is mandatory to follow social distancing to avoid coming in contact with infected persons. This hybrid model is helpful in monitoring whether people are following the norms of social distancing or not.

"Social Distancing" the word itself is an important effort aiming to reduce the transmission of the virus. The norm of social distancing aims to minimise the physical contact between people. According to WHO, people must maintain a minimum distance of 6 feet among each other[2]. Many studies have shown that social distancing is an important measure to control the widespread of this deadly disease [3]. Fig. 1 shows the reduced peak of the pandemic when people follow social distancing norms[4]. It also shows that social distancing is one of the best ways to minimize the physical contact that causes a reduction in an increased rate of infection [5]. A lot of medicinal organizations and pharmaceutical companies are trying to develop medicines and vaccines for COVID-19 but there is no magic happened to date which can be considered as a treatment for this disease. Therefore, people must follow precautionary measures to limit the spread of the virus.

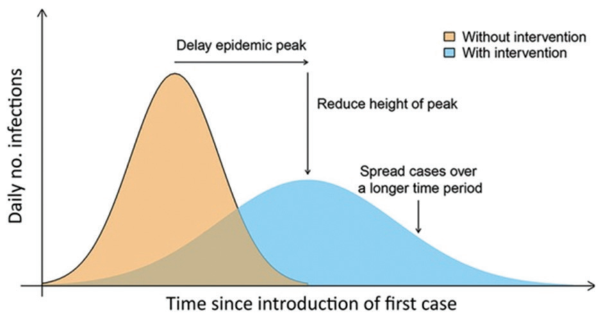

Fig. 1. Impact of Social Distancing on COVID-19 cases since the first case arrived

With an increasing number of cases and death counts, it is very important to maintain social distance and use masks to avoid the spread of coronavirus. The idea behind Social Distancing Tracker is to determine whether two people are following the social distancing norms or not. If two persons are following the social distancing norms then they are surrounded by a green bounding box and if they are too close to each other then they are surrounded by a red bounding box. This method works on the principle of measuring distances between people in pixels using different algorithms to compute the distance. The government has already specified the distance amongst two people i.e. 6 feet that need to be followed by everyone. The proposed hybrid model is developed using concepts of deep learning and image processing. It is compared with other object detection models used so far. The performance of object detection models has been compared to monitor the social distancing.

The rest of the paper is divided into the following sections: Section 2 provides a review of related literature in the field of study; section 3 provides knowledge on object detection tracking models and tracking techniques; section 4 presents the proposed SSDT model, section 5 discusses the implementation of SSDT model, section 6 presents results and discussions, and section 7 concludes the paper.

2. RELATED LITERATURE

In China, when Coronavirus arrived in December 2019, social distancing opted as a precautionary measure on January 23, 2020. Social distancing is an effective method to limit the transmission of the Covid-19 virus. The government has already specified the distancing rules to avoid the transmission of this deadly disease. A lot of work has been done by researchers to track the distance during this pandemic which is as follows:

Chhaya Gupta et al. [1] proposed a Deep learning-based CNN method for facemask detection. The idea behind their work is to determine whether a person is wearing a mask or not. If the person is not wearing a mask, then the person's face is surrounded by a red colour box and a label saying "without a mask" is also present.

Mahdi Rezaei et al. [4] proposed a hybrid computer vision and YOLOv4 based Deep Neural Network (DNN) for people detection in the crowd using common CCTV cameras. They have combined the model with the SORT tracking algorithm and inverse mapping technique to monitor social distancing. The proposed method was evaluated on the Oxford Town Centre dataset and the system provided 98% accuracy.

Dr S Syed Ameer Abbas et al. [6] founded a framework for human tracking and crowd management using Raspberry pi. They estimated the head tally and crowd is measured by contrasting the value and threshold value.

Prem K et al. [7] used synthetic location-specific contact patterns and produced matrices. They proposed an age-structured susceptible-exposed-infected-removed (SEIR) model for various social distancing measures. They found that social distancing measures were most effective in April and reduced the number of cases to 92%. They found that when restrictions were put on activities it helped in delaying epidemic peak.

Since coronavirus arrived, various countries take the help of technology to track movements of infected persons and tried to monitor their exposure to other people [8–10]. Many countries like India and South Korea used GPS tracking systems to track the movement of suspected persons among healthy people. The Indian government has also developed an Arogya Setu App to keep track of COVID-19 patients in surroundings [11]. The app uses Bluetooth and GPS tracking systems and helps healthy people to stay away from infected people. Many other departments have been using various other techniques like drones or surveillance cameras to identify gatherings and take required steps to disperse the crowd [12].

Ertem Zeynep et al. [13] proposed a decision analytical approach to estimate the effectiveness of social distancing methods. They proposed an age-structured compartmental simulation model to analyze computational results. The study shows that decision analytic tools help simulate different social distancing scenarios.

Su Jie et al. [14] proposed a visual social distancing (VSD) method for real-time social distancing measuring and analyzing the distance between pedestrians using CCTV videos in public areas. They proposed a multi-pedestrian tracking approach for Spatio-temporal trajectory. The method works on Euclidean distance between tracking objects but also considers discrete Frechet distance between trajectories. They worked on the MOT16 dataset.

Sreetama Das et al. [15] proposed a computer vision-based solution to encourage abidance with social distancing norms. The method provides options to choose between tool-based mode and automated mode of the camera. They have discussed various risk factors associated with social distancing violations.

Maria Fazio et al. [16] proposed an efficient and cost-effective indoor navigation system for moving people inside buildings. The method works on short-range wireless communication technology –IoT-based Bluetooth Low Energy (BLE) Beacons. The authors provide a new navigation system that recognizes user position according to information provided by beacons. This method reduces contagious risk and is very much helpful for the movement of people in smart cities during lockdowns. This method is really helpful for patients who want to travel from one ward to another. But this method is useful for people who have access to a smartphone which is a limitation.

Asif Hummam et al. [17] proposed a low confidence track filtering into simple online and Real-time tracking with a deep association metric (DeepSort) algorithm. The proposed method has shown improved results over the classic DeepSort algorithm with significant margins.

Narinder Singh Punn et al. [18] proposed a deep-learning-based framework using YOLOv3 merged with the DeepSort technique. The authors concluded that the proposed model showed better results when compared with different faster R-CNN models.

A lot of research has been done in the field for human detection and it is shown that human detection can be used in different applications. Despite the work done till now, there are a lot of limitations and hindrances that need to be investigated:

- Most of the work has been carried out with small datasets.
- The researchers have used pre-trained models via transfer learning for producing high accuracy results and it takes a lot of effort in changing the architecture of a pre-trained model and add something new to it.
- Most of the work has been done on available datasets and it is a task to work on real-time data.

To overcome the limitations, a hybrid approach using YOLOv4 merged with MF-SORT, Kalman Filter and Brute force feature matching technique has been proposed. The model in this paper does not work on real-time data as of now, but in future, it will be trained to work with real-time data.

3. OBJECT DETECTION TRACKING MODELS

Different object detection models have been used so far for detecting social distancing norms. The basic requirement for detecting objects in an image or any video is a bounding box that surrounds the object of interest. From the vast literature review, it has been observed that every object detection model uses bounding boxes for identifying multiple objects in an image [19]. The bounding box helps to identify each object independently and helps the object detection model to classify each object. The bounding boxes surround the images over different locations. The bounding boxes help the object detection model to predict the class of each bounding box and the object detection model helps to adjust the dimensions of the box to fit better. As there are many bounding boxes so one object can be associated with more than one bounding box and this problem is solved by evaluating the Intersection over Union (IoU) parameter [20] with the help of Non-Max Suppression (NMS) [21] helps in calculating ratios between overlapped regions and union of regions of various bounding boxes.

Every bounding box is assigned a label as positive or negative according to the associativity of the object of interest. For positive label 1 is used and for negative label 0 is used.

3.1. OBJECT DETECTION MODELS

Various object detection models have been used to detect social distancing norms which are as follows:

Faster R-CNN: This model was proposed by Ren et al. [22] from R-CNN and Fast R-CNN models and the architecture of this model is shown in Fig. 2. This model is composed of Region Proposal Network (RPN) module that is responsible for binary classification by which each object is classified. This model is an association of RPN and fast R-CNN models. Classification loss (Lcls) and regression loss (Lreg) for faster R-CNN are shown with the help of equations 1 and 2.

$$L_{cls}(p_i, p_i^*) = -p_i^* \log(\mathrm{p_i}) - (1 - p_i^*) \log(1 - p_i) \quad (1)$$

$$L_{reg}(t^u, v) = \sum_{x \epsilon x,y,w,h} L_1^{smooth} (t_i^u - v) \quad (2)$$

Where
tu = bounding box predicted corrections
pi= actual class
pi*= predicted class

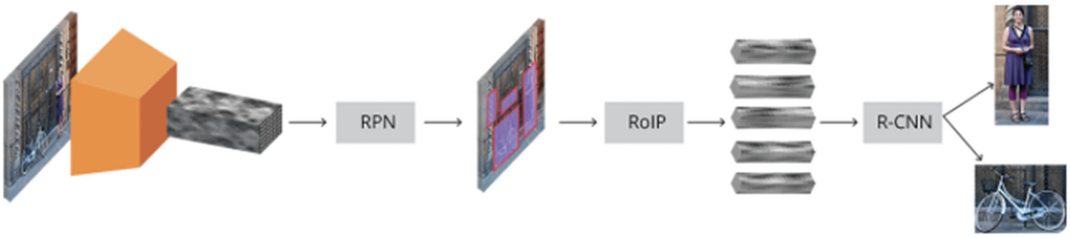

Fig. 2. The architecture of Faster R-CNN[22]

Single Shot Detector (SSD): This model is also used to detect people in videos. Faster R-CNN provides higher accuracy but has slow processing of frames and provides a low FPS rate. SSD improves the FPS rate and accuracy by using multiple scale features. It utilizes a feed-forward convolutional neural network that helps in associating objects with bounding boxes of fixed size and then the model follows the Non-Max Suppression (NMS) module to produce the final result. The complete model is categorized into three stages: in the first stage, a pre-trained network is used to extract feature maps, in the second stage multiple-scale feature is used and finally, the NMS module is used to provide the final result. The classification loss function and regression loss function for the SSD model is defined in equation 3 and 4. The architecture of this model is shown in Fig. 3[23].

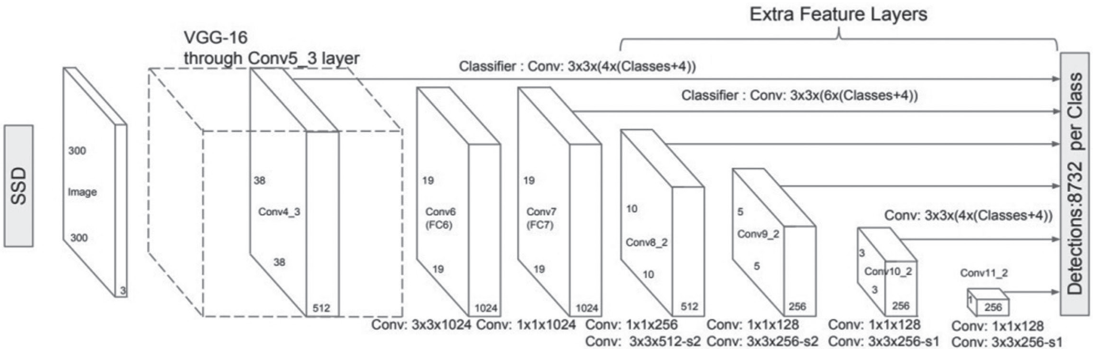

Fig. 3. The architecture of SSD[23]

$$L_{cls}(x,c) = -\sum_{i\in Pos}^{N} x_{ij}^{p}\log\left(\hat{c}_i^{p}\right) - \sum_{i\in Neg}\log(\hat{c}_i^{0}) \quad (3)$$

$$L_{reg}(x,l,g) = \sum_{i\in pos}^{N}\sum_{m\in c_x,c_y,w,h} x_{ij}^{k}\; smooth_{L_1}\left(l_i^{m} - \hat{g}_j^{m}\right) \quad (4)$$

Where

l = predicted box

g = ground box

xij = it matches ith bounding box to jth ground box

cx and cy = offsets for bounding box

N = number of default matched boxes

YOLO: You Only Look Once (YOLO) can predict the location and type of object by just looking at the image once. It solves the problem of object detection by assuming it as a regression task rather than a classification task. In this model, only one convolutional network is used that helps to predict the bounding boxes. YOLO has many versions available like YOLOv1 [23] which is inspired from GoogleNet (or inception model) [24], YOLOv2 [25] which improves the accuracy provided by YOLOv1 and YOLOv3 performs multi-label classification. Equation 9 shows the loss function for YOLO is the mean average precision loss function for YOLO and the basic architecture of YOLO is represented in Fig.4 [26].

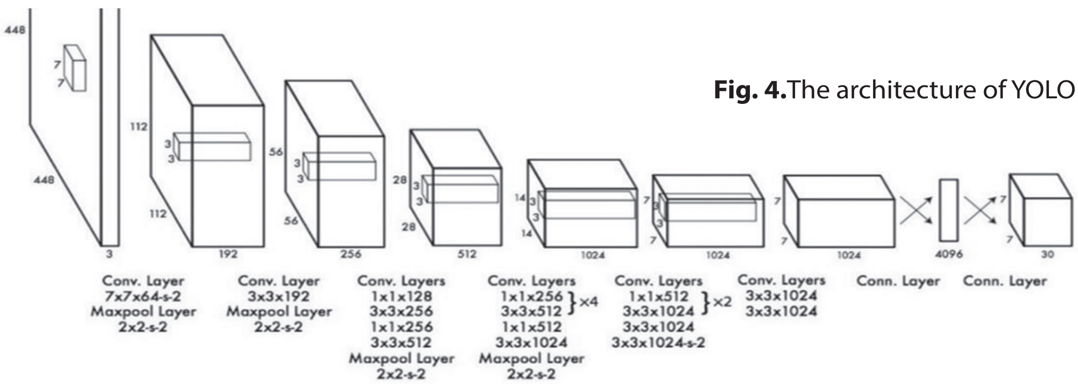

Fig. 4. The architecture of YOLO[26]

YOLOv4 outperforms the inference speed provided by other object detection models. It is a series of additions of computer vision methods. This method makes real-time object detection a priority. The ImageNet classification model forms the backbone for YOLOv4. This model also deploys the same bounding box detection steps as used by YOLOv3 and three steps of detection granularity. There are two threshold values taken in YOLO:

- IoU threshold: IoU is the ratio of the area of overlap of two images to the area of the union of two images. Equation 5 describes the IOU threshold value.

$$\text{IoU} = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (5)$$

Confidence threshold: This value refers to the box confidence score which means, the minimum confidence the model has on the detected object.The equations to calculate the confidence scores are shown below:

$$\text{Box Confidence score} = P_r(\text{object}).\text{IoU} \quad (6)$$

$$\text{Conditionalclass probability} = P_r(\text{class}_i|\text{object}) \quad (7)$$

$$\text{Class confidence score} = P_r(\text{class}_i).\ \text{IoU} \quad (8)$$

Where,

P_r(object) = probability that box has an object

IoU = Intersection over Union between the predicted box and ground truth

$P_r(\text{class}_i|\text{object})$ = probability that object belongs to class_i given an object is present.

$P_r(\text{class}_i)$ = probability that the object belongs to class_i.

If a box is detected then P(object) has value 1 otherwise 0. In this paper, the confidence score threshold is used.

$$\lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{i,j}^{obj} \left((t_x - \widehat{t_x}) + \left(t_y - \widehat{t_y}\right)^2 + (t_w - t_w)^2 + (t_h - t_h)^2 + \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{i,j}^{obj} \left(-\log\left(\sigma(t_0)\right) + \sum_{k=1}^{C} BCE\left(\widehat{y_k}, \sigma(s_k)\right)\right) + \lambda_{noobj} \sum_{i=0} \sum_{j=0}^{B} 1_{i,j}^{noobj} \left(-\log\left(1 - \sigma(t_0)\right)\right) \quad (9)$$

Where,

λ_{coord} = weighted coordinator error

s^2 = number of grids in an image

B = number of bounding boxes per grid

$1_{i,j}^{obj}$ = 1 indicates that the objectresides in the jth bounding box in grid i.

Fig. 5 shows successful object detection of various models like Faster R-CNN [22], Fast YOLO [27], Single Shot Detector (SSD) [28] and YOLO [29] validated on COCO and PASCAL-VOC datasets, compared on their speed and accuracy which is dependent on many factors like input sizes, resolution of images, batch sizes etc.

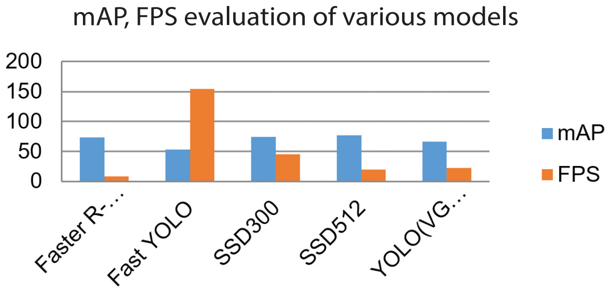

Fig. 5. Performance evaluation of various models

The models have been evaluated on the basis of mAP (mean Average Precision) [30] and FPS (Frames Per Second).

Table1 shows the performance evaluation of the above said models in terms of mAP, FPS, Batch size and Image resolution passed to models while testing them on PASCAL-VOC and COCO datasets.

Table 1. Performance of different Object Detection Models on various parameters.

Method	mAP	FPS	Batch Size	Input Resolution
Faster R-CNN	73.3	8	1	1000 X 600
Fast YOLO	52.8	154	1	448 X 448
SSD300	74.5	45	1	300 X 300
SSD512	77	20	1	512 X 512
YOLO(VGG16)	66.5	22	1	448 X 448

3.2. OBJECT DETECTION TRACKING TECHNIQUES

This section gives the idea about the basics of tracking. In a real-time environment, trackers need to be connected with a detector. Let us assume if the bounding box information for one ID in the frame is given then how to assign IDs in subsequent frames? This question is answered below:

Assigning ID based on Centroid: This is the simplest way to assign IDs. Centroids for each bounding box in one frame are calculated. After this, new centroids are calculated in the second frame (if any) and based on distance from previous centroids IDs are assigned. But this method fails when people come close to each other as this may switch IDs.

Particle-Filter: This method is being used widely in real-time object tracking. The main features of this filter are its simplicity and its flexibility. It is very easy to handle non-Gaussian and multimodality system models with this filter but it does not work well with gaussian systems, which is a drawback[31].

Kalman – Filter: This method is an improvement over the centroid method as this is based on the position and velocity of an object and thus helps in modelling track. It uses Gaussian scales to estimate the future position and velocity of an object. When a new reading is achieved, this method uses probability to assign a measurement to its prediction and hence updates itself accordingly. It uses very less space and is effectively fast and shows better results than the centroid method. This method also helps in reducing unwanted noise by inaccurate detections. The current position of the object can be predicted by its previous moment position[32].

In addition, with Kalman filter, the proposed model also uses the brute force feature matching technique.

4. PROPOSED MODEL

A lot of challenges faced during the categorization and detection of objects in any image can be solved with the support of advanced computer vision and deep learning models. Computer vision focuses on various challenging aspects like segmentation, tracking and detection of objects. Deep learning is an artificial intelligence method that is useful for data processing and objects detection. Neural networks come into the picture in the late 1940s [33].

The main purpose of neural networks is to solve learning problems [19]. According to the literature survey, a robust model has been developed that can detect whether people are maintaining social distancing or not with the help of OpenCV and YOLO object detectors. OpenCV is an open-source computer vision programming library that accelerates the utilization of machine perception within the commercial product. The paper proposes a deep learning-based hybrid model SSDT (Smart Social Distancing Tracker) that helps to deal with social distancing violations and can reduce the number of COVID-19 cases. The proposed model uses YOLOv4 merged with MF-SORT[34], Kalman filter and Brute Force feature matching technique for computing social distancing violations in a video. The bounding boxes are colour coded which means if two persons are not following social distancing norms then the red colour bounding box surrounds both of them otherwise the colour of the bounding box is green. The workflow of the proposed framework is shown in Fig. 6.

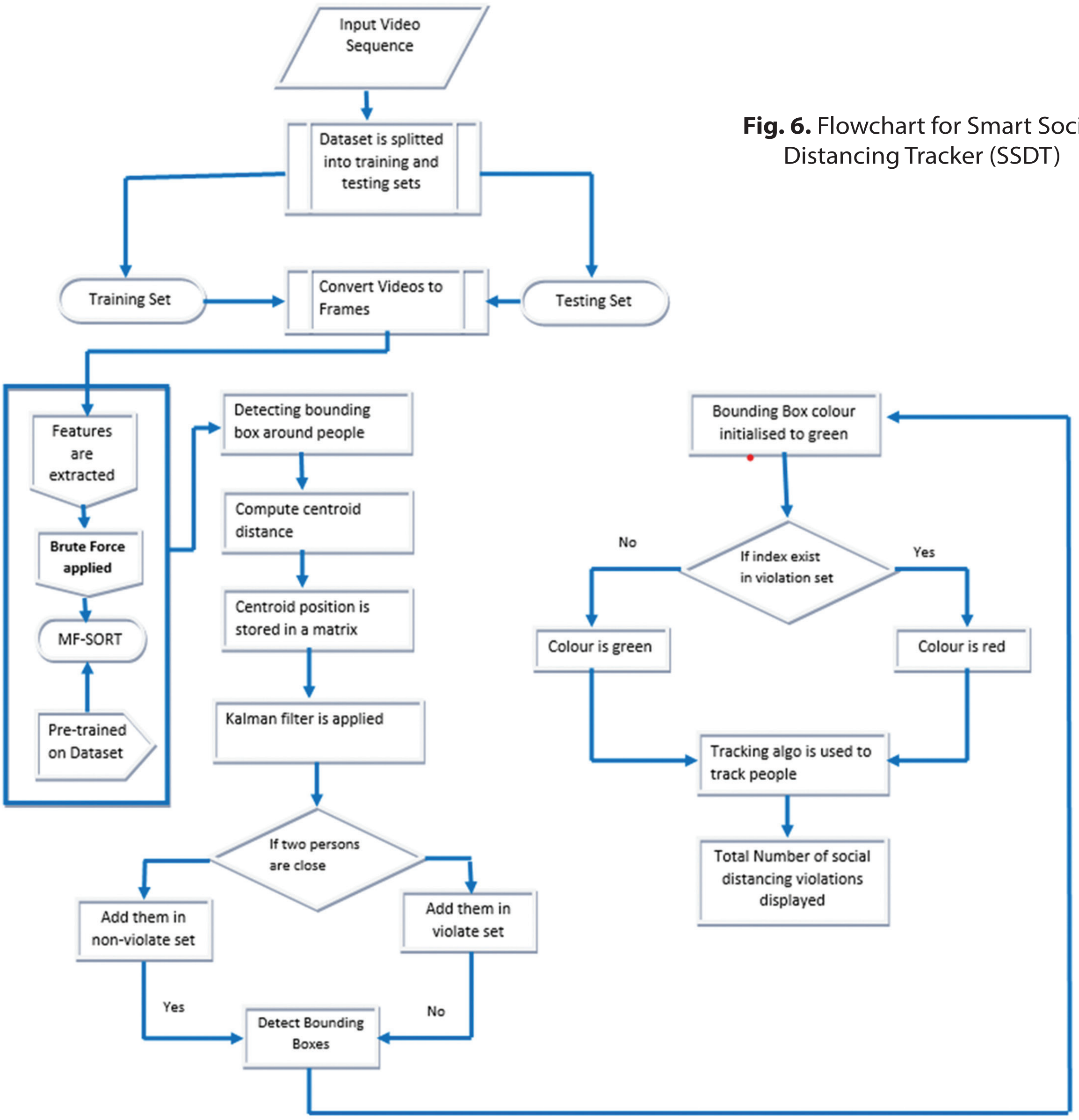

Fig. 6. Flowchart for Smart Social Distancing Tracker (SSDT)

5. IMPLEMENTATION OF SSDT MODEL

To develop the social distancing tracker, Computer vision and deep learning methods have been used. The hybrid model is based on YOLOv4 merged with MF-SORT (Simple Online and Real-time with motion features), Kalman-filter and brute force matching technique. The proposed model is implemented on MOT (Multiple Object Tracking) datasets[35] which is discussed as follows:

People Tracking – The first thing needed is a video and a tested model on it. The MOT dataset has been used for implementation which is a canonical dataset for computer vision people tracking. This dataset has many open-sourced clips having people's movements with different camera angles. Fig. 7 shows one of the frames from the clip.

Fig. 7. Sample frame from MOT dataset

Person tracking is the task of providing an ID to a person detected in every frame and carrying their ID forward. Once the person has left a frame the ID is not reused and if a new person enters the frame, he/she is assigned a new ID. Tracking is a tedious task as people can look similar. People get occluded behind another person or any other object and they are assigned a new ID when they re-emerged. Deep learning helps in multi-object tracking. In this paper, the Kalman-filter method is used to track people and provide IDs. The hybrid model in this paper uses Simple Online and Real-time with motion features (MF-SORT) technique with Kalman filter and brute force matching techniques. Kalman filter helps in predicting human position at t+1 time based on present time t and this helps in identifying people in case of occlusion as well hence reducing occlusion. Brute force feature matcher compares two sets of keypoint descriptors and generates output. The output is a list of matches found. This technique is used to match the features of one image with the features of the second image, it takes a descriptor of the first image and matches it with all descriptors of the second image, then it takes the second descriptor of the first image and matches it with all the descriptors of the second image and so on[36]. It takes Euclidean distance or hamming distance depending on the type of detector but in this researchwork, brute force matcher is making use of distance calculated as per the centroid distances.

The state of each person in a frame is stated as,

$$x = [a, b, c, d, a', b', c']^T \tag{10}$$

Where (a,b) are horizontal and vertical position (centroids) of bounding box a and c is area and d is the aspect ratio of bounding box a',b',c' is predicted values by Kalman filter.

If an identified person is associated with a new observation, the present bounding box will be updated with this newly observed value. This is calculated by the Kalman filter based on the velocity and acceleration of the object.

After the detection and tracking process, a detection matrix Dt is defined that includes the location of n detected human in any image grid as,

$$D_t = \{P^t_{(x_n, y_n)}\} \tag{11}$$

Tracking Social distancing – There are different tracks in different frames and distance is measured between all the tracks. A bounding box with an ID is known as a track. Hence, the distance between bounding boxes is compared using the Euclidean Distance formula. The following steps have been performed for every frame:

Step 1: Pixel distances are compared between each track.

Step 2: If distance < proximity value (6feet) then, two people are walking very close to each other, hence changing the value of safe to 1 in the data frame for both the bounding boxes. The variable "safe" is used as a visualization variable later.

Step 3: To count the total breaches for each ID, so anytime distance <proximity value, a list of tracks is maintained that have come too near to each other. Many calculations are repeated many times to measure the distance between tracks and hence to save time, results are stored in a single pass which cuts the run time to half. One of the frames with results is shown below in Fig. 8.

Fig. 8. A sample frame showing results of social distancing violations between people

If a minimum of 2 persons are detected in a frame, then the following steps are performed:

Step 1: Euclidean distance is calculated between both persons.

Step 2: Loop over the upper triangular distance matrix.

Step 3: To check whether the distance is violating the minimum value. If two persons are extremely close then they are added to the violation set.

Step 4: The present index is verified if it is present in the violation set, if it is present in the violation set then the colour of the bounding box changes to red.

Step 5: The bounding box of each person in a frame is drawn. Every person is colour coordinated and hence now individuals can be verified whether they are nearby or not.

Step 6: Finally, the total number of violations is shown.

Fig. 9 shows one of the sample frames identifying the total number of social distancing violations being marked.

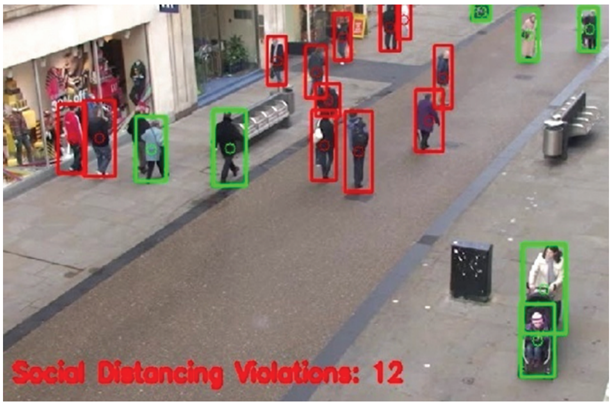

Fig. 9. A sample frame that shows the number of social distancing violations

6. RESULTS AND DISCUSSIONS

The proposed model has been trained on a video stream from the MOT dataset and various images. People are detected based on the distance between the pairs. The frames were also labelled as safe and unsafe and also the count of violations was made visible. The results are shown in Fig. 9. The red colour boxes around people indicate that they are not maintaining the social distancing and the green colour box indicates that the person is not violating any social distancing rule.

The is also compared with the Faster R-CNN model and YOLOv3 model[18]. Table-2 indicates the results of the models with the training time of each model, mean average precision (mAP) and frames per second (FPS) values. The result shows that Faster R-CNN achieves minimum loss when mAP is maximum but it also provides minimum FPS hence making it a non-suitable model for real-time object detection. The proposed hybrid model SSDT achieved much better results as compared to Faster RCNN in terms of mAP, FPS and training time. YOLOv3 and SSDT have similar training times but the big difference is in mean average precision values, hence concluding that SSDT is a better model over these two models. Fig. 10 shows the performance evaluation graphically.

Table 2. Performance comparison of SSDT and Faster R-CNN.

Model	Training Time (s)	mAP	FPS
Faster R-CNN	9765	0.964	3
YOLOv3	5659	0.846	23
SSDT	5645	0.976	24

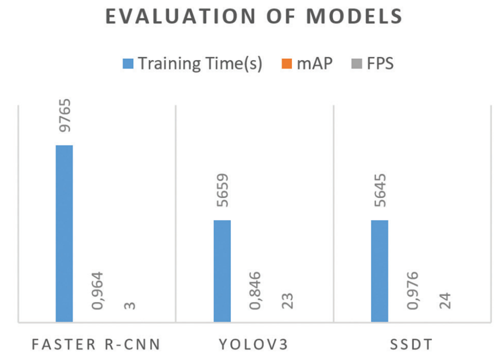

Fig. 10. Performance Evaluation of SSDT and Faster R-CNN

The proposed model is a hybrid social distancing tracker method but still, there are many challenging situations when it comes to implementing this type of model. Designing a system responsible for changes and flexible for all environments is a challenging task. This model can be deployed in many public areas to watch people and track social distancing violations. The proposed hybrid model works in challenging conditions such as occlusions, light conditions and hence it is an important method for different authorities to design public places with precautions as a measured thing to take care of in such difficult times.

7. CONCLUSION

The objective of this work is to identify a violation of social distancing norms for which a smart social distancing tracker (SSDT) has been proposed. If people are maintaining a safe distance between them then they will be marked with a green bounding box and if people are not following the social distancing laws then they will be marked with a red bounding box. The proposed model may be very helpful for cities, shops, restaurants, shopping malls etc. to assess public health risks. SSDT model is trained on MOT datasets and when compared with faster RCNN, and YOLOv3the SSDT model provides better and balanced results in terms of training time, mAP and FPS. The faster RCNN model provides minimum loss when it has maximum mAP but has a very less FPS rate and takes more training time as compared to SSDT hence making it not a suitable model for real-time object detection. Although YOLOv3 and SSDT have similar results, the main difference is in mean average precision loss, YOLOv3 has lower mAP compared to SSDT. As the proposed model is efficient in detecting social distancing violations and is also able to remove occlusions between people successfully still there are several unaddressed issues like - it is not able to remove all the challenges of object detection like primitive illusions, low light visions, etc. but in future, this model may be deployed with an IoT based model along with deep learning methods to resolve the challenges of real-time object detection and also, the model is trained on video clips from MOT datasets, but in future, it will be trained and tested on real-time data.

8. REFERENCES:

[1] C. Gupta, N. S. Gill, "Coronamask: A face mask detector for real-time data", International Journal of Advanced Trends in Computer Science and Engineering, Vol. 9, No. 4, 2020, pp. 5624–5630.

[2] N. El-Guebaly, "COVID-19 and social distancing", Canadian Journal of Addiction, Vol. 11, No. 2, 2020, pp. 4–6.

[3] F. Ahmed, N. Zviedrite, A. Uzicanin, "Effectiveness of workplace social distancing measures in reducing influenza transmission: A systematic review", BMC Public Health, Vol. 18, No. 1, 2018, pp. 1–13.

[4] M. Rezaei, M. Azarmi, "Deepsocial: Social distancing monitoring and infection risk assessment in covid-19 pandemic", Applied Sciences, Vol. 10, No. 21, 2020, pp. 1–29.

[5] J. Pannu, "Nonpharmaceutical measures for pandemic influenza in nonhealthcare settings-international travel-related measures", Emerging Infectious Diseases, Vol. 26, No. 9, 2020, pp. 2298–2299.

[6] S. Syed Ameer Abbas, M. Anitha, X. Vinitha Jaini, "Realization of multiple human head detection and direction movement using Raspberry Pi", Proceedings of the International Conference on Wireless Communications, Signal Processing and Networking, Chennai, India, 22-24 March 2017, pp. 1160–1164.

[7] K. Prem et al., "The effect of control strategies to reduce social mixing on outcomes of the COVID-19 epidemic in Wuhan, China: a modelling study", The Lancet Public Health, Vol. 5, No. 5, 2020, pp. e261–e270.

[8] S. K. Sonbhadra, S. Agarwal, P. Nagabhushan, "Target specific mining of COVID-19 scholarly articles using one-class approach", Chaos, Solitons and Fractals, Vol. 140, 2020, p. 110155.

[9] N. S. Punn, S. Agarwal, "Automated diagnosis of COVID-19 with limited posteroanterior chest X-ray images using fine-tuned deep neural networks", Applied Intelligence, Vol. 51, No. 5, 2021, pp. 2689–2702.

[10] N. S. Punn, S. K. Sonbhadra, S. Agarwal, "COVID-19 epidemic analysis using machine learning and deep learning algorithms", medRxiv, 2020, pp. 1–10.

[11] "Covid-19 contact tracing app Aarogya Setu has alerted 1.4 lakh users: Official", https://www.livemint.com/news/india/covid-19-contact-tracing-app-aarogya-setu-has-alerted-1-4-lakh-users-official-11589226902816.html (accessed: 2021)

[12] M. Robakowska et al. "The use of drones during mass events", Disaster and Emergency Medicine Journal, Vol. 2, No. 3, 2017, pp. 129–134.

[13] Z. Ertem, O. M. Araz, M. Cruz-Aponte, "A decision analytic approach for social distancing policies during early stages of COVID-19 pandemic", Decision Support Systems, 2021, p. 113630.

[14] J. Su, X. He, L. Qing, T. Niu, Y. Cheng, Y. Peng, "A novel social distancing analysis in urban public space: A new online spatio-temporal trajectory approach", Sustainable Cities and Society, Vol. 68, 2021, p. 102765.

[15] S. Das, A. Nag, D. Adhikary, R. J. Ram, S. K. Ojha, G. M. Hegde, "Computer Vision-based Social Distancing Surveillance Solution with Optional Automated Camera Calibration for Large Scale Deployment", arXiv:2104.10891v1, 2021.

[16] M. Fazio, A. Buzachis, A. Galletta, A. Celesti, M. Villari, "A proximity-based indoor navigation system tackling the COVID-19 social distancing measures", Proceedings of the IEEE Symposium on Computers and Communications, Rennes, France, 7-10 July 2020.

[17] A. H. Rais, R. Munir, "Vehicle Speed Estimation Using YOLO, Kalman Filter, and Frame Sampling",

Proceedings of the 8th International Conference on Advanced Informatics: Concepts, Theory and Applications, Bandung, Indonesia, 29-30 September 2021.

[18] Z. Q. Zhao, P. Zheng, S. T. Xu, X. Wu, "Object Detection with Deep Learning: A Review", IEEE Transactions on Neural Networks and Learning Systems, Vol. 30, No. 11, 2019, pp. 3212–3232.

[19] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15-20 June 2019, pp. 658–666.

[20] J. Hosang, C. V May, "Learning non-maximum suppression", arXiv:1705.02950v2, 2017.

[21] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, No. 6, 2017, pp. 1137–1149.

[22] X. Wang, X. Hua, F. Xiao, Y. Li, X. Hu, P. Sun, "Multi-object detection in traffic scenes based on improved SSD", Electronics, Vol. 7, No. 11, 2018.

[23] U. Handalage, L. Kuganandamurthy, "Real-Time Object Detection using YOLO: A review", 2021.

[24] P. Salavati, H. M. Mohammadi, "Obstacle detection using GoogleNet", Proceedings of the 8th International Conference on Computer and Knowledge Engineering, Mashhad, Iran, 25-26 October 2018 pp. 326–332.

[25] S. Gupta, T. U. Devi, "YOLOv2 based Real Time Object Detection", International Journal of Computer Science Trends and Technology, Vol. 8, No. 3, 2020, pp. 26–30.

[26] S. Shinde, A. Kothari, V. Gupta, "YOLO based Human Action Recognition and Localization", Procedia Computer Science, Vol. 133, 2018, pp. 831–838.

[27] M. J. Shaifee, B. Chywl, F. Li, A. Wong, "Fast YOLO: A Fast You Only Look Once System for Real-time Embedded Object Detection in Video", Journal of Computational Vision and Imaging Systems, Vol. 3, No. 1, 2017.

[28] W. Liu et al. "SSD Single Shot MultiBox Detector", arXiv:1512.02325v5, 2016.

[29] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", arXiv:1506.02640v5, 2016.

[30] P. H. B, V. Ferrari, "End-to-End Training of Object Class Detectors for Mean Average Precision", Proceedings of the 13th Asian Conference on Computer Vision, Taipei, Taiwan, 20-24 November 2016, pp. 198–213.

[31] H. Chu, K. Wang, X. Xing, "Target Tracking via Particle Filter and Convolutional Network", Journal of Electrical and Computer Engineering, Vol. 2018.

[32] C. Urrea, R. Agramonte, "Kalman Filter: Historical Overview and Review of Its Use in Robotics 60 Years after Its Creation", Journal of Sensors, Vol. 2021, No. 1, 2021.

[33] W. Pitts, W. S. McCulloch, "How we know universals the perception of auditory and visual forms", The Bulletin of Mathematical Biophysics, Vol. 9, No. 3, 1947, pp. 127–147.

[34] H. Fu, L. Wu, M. Jian, Y. Yang, X. Wang, "MF-SORT: Simple Online and Realtime Tracking with Motion Features", Proceedings of the International Conference on Image and Graphics, Beijing, China, 23-25 August 2019, pp. 157–168.

[35] Y. Zhang et al. "ByteTrack: Multi-Object Tracking by Associating Every Detection Box", arXiv:2110.06864v3, 2022.

[36] A. Jakubović, J. Velagić, "Image feature matching and object detection using brute-force matchers", Proceedings of the International Symposium ELMAR, Zadar, Croatia, 16-19 September 2018, pp. 83–86.