Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 1157 – 1170

doi:10.3906/elk-2008-66

Subjective analysis of social distance monitoring using YOLO v3 architecture and crowd tracking system

Muhammed Murat ÖZBEK[1], **Mustafa SYED**, **İlkay ÖKSÜZ**[2,3]

[1] Aeronautical Engineering Department, Faculty of Aeronautics and Astronautics, İstanbul Technical University, İstanbul, Turkey

[2] Computer Engineering Department, Faculty of Computer and Informatics Engineering, İstabul Technical University, İstanbul, Turkey

[3] School of Biomedical Engineering & Imaging Sciences, King's College London, London, UK

Received: 16.08.2020 • **Accepted/Published Online:** 12.10.2020 • **Final Version:** 30.03.2021

Abstract: The lethal infection, World Health Organization (WHO) reported coronavirus (COVID-19) as a pandemic. Lack of proper vaccine, low levels of immunity against COVID-19 has led to vulnerability of the human beings. Due to lack of efficient vaccine treatment, the only options left to fight against this pandemic are lockdown and social distance. This work offers an autonomous monitoring system on social distancing using deep learning techniques. The proposed architecture tracks the humans on roads and calculates their distance between each other. This surveillance detects the furore violation of social distance utilizing CCTV cameras. The proposed framework uses YOLO v3 object-detection model built on COCO dataset and used to classify human class among 79 classes. The bounding box's dimensions and centroid coordinates are computed in the two-dimensional feature space from the pairwise vectorized L2 norm and a threshold is fixed for computing the distance maintained between each other. We illustrate the superior performance of our framework checked against other state of the art methods regarding inference speed, mean average precision and loss defined from the localization.

Key words: COVID-19, social distancing, YOLO v3, COCO dataset, inference, video surveillance, edge devices, object detection

1. Introduction

COVID-19 also known as Corona virus disease initially was reported in the state of Wuhan in China during late December 2019. As on May 2020, it is believed to be spread across 180 nations, with around 7.5 million active cases and more than 1.1 million deaths. World Health Organization (WHO) has announced this as a pandemic.[1] Scientists, researchers, doctors round the globe are intensively working on the vaccine but none of them have succeeded completely till date. The super spreader reported through human-to-human transmission has led to the maintain social distancing by the governments.[2] With social distancing being in order, the transmission across global has slowed down. Figure 1 describes the outcomes on maintaining appropriate social distancing.[3]

[1]WHO (2020). Who announces COVID-19 outbreak a pandemic [online]. Website http://www.euro.who.int/5en/health-topics/health-emergencies/coronavirus-covid-19/news/news/2020/3/6who-announces-covid-19-outbreak-a-pandemic [accessed 21 September 2020].

[2]Centers for Disease Control and Prevention (2020).Social distancing [online]. Website www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/social-distancing.html [accessed 19 September 2020].

[3]The Washington Post (2020). Why outbreaks like coronavirus spread exponentially, and how to "flatten the curve" [online]. Website www.washingtonpost.com/graphics/2020/world/corona-simulator/ [accessed 23 September 2020].

As shown on the graph, it clearly indicates curbing the virus transmission by maintaining social distancing is the best practice than any other means.

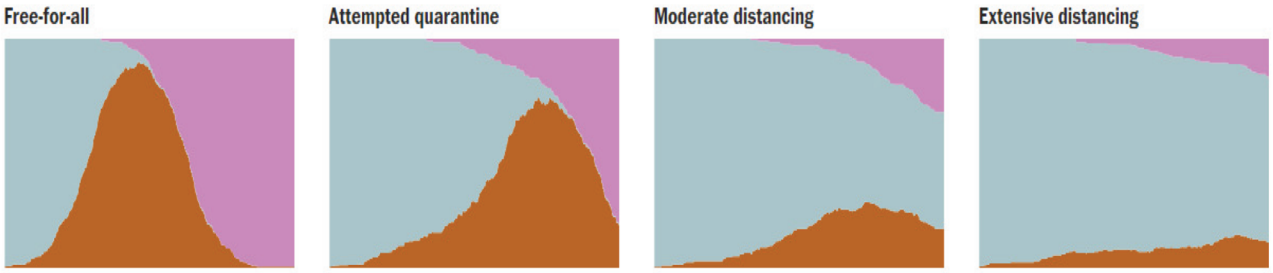

Figure 1. Graphical comparison of outcomes on with and without social distancing. Brown curve represents the possible active cases. It is seen that the peak point of the distribution is at a lower point where there is social distance. In places where there is no social distance, the peak of distribution is very high.

Social distancing has a vital role in preventing the virus because the people having very mild symptoms or no symptoms at all shall fortuitously carry the virus with them and without social distancing, it can be a super spreading in the community. As per the WHO guidelines, to avoid such devastating catastrophes, the best way is to either stay at home completely or to maintain at least 3 feet physical distance with others.[4] Due to dense population and large number of cases rising on a daily basis, leading to shortage of medical essentials, definitely the most reliable way to prevent the spreading of contagious virus is social distance. When (COVID-19) appeared in Wuhan, China December of 2019, social distancing was selected as an unprecedented way on January 2020.[5]

However, after the social distancing being put in place, a varied assemblage of people is noticed flouting the norms on streets. Due to impossibility of security forces being present round the clock everywhere, the surveillance has been a vital tool to ensure social distancing. As economy is being deprived due to the lockdown[6], the governments are partially lifting up the restrictions to restore a minimum economy for survival. These new norms are crucial in the battle against corona virus and there is need for active surveillance. Since, the security staff cannot be present everywhere and merely impossible to monitor every citizen, computer vision helps in broader way by monitoring everyone within no time and alerting the security. This not only helps the security in restoring dispersion of crowd but also results in curbing super spreading of the virus. In this process, we design a framework, which can aid in automatically detecting violations of social distancing measures.

2. Related works

We provide an analysis of works that is related for object detection. We review methodologies while featuring the distinctions with our proposed method.

Deep Learning has shown great success on challenges and tasks for example speech recognition [1], machine translation, medical diagnosis, etc. [2]. The majority of the tasks are focused on object detection, classification, tracking, recognition, and segmentation. Object-detection is yet a really difficult issue for the

[4]WHO (2020). Corona virus disease (COVID-19) advice for the public [online]. Website www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public [accessed 18 August 2020].

[5]BBC (2020). China coronavirus: lockdown measures rise acress Hubei province [online]. Website www.bbc.com/news/world-asia-china-51217455 [accessed 20 September 2020].

[6]BBC (2020). Coronavirus: a visual guide to the economic impact [online]. Website www.bbc.com/news/business-51706225 [accessed 17 September 2020].

vision of computer field. It mainly needs to address two key issues: determining the objects coordinates and predicting their class. Recently convolutional neural networks (CNN) became the state of the art in object detection task. These methods are split to two groups: one stage methods, that maintain coordinate detection also class prediction in one step, and two stage methods, that initially identify areas of the photos that object can be existing, afterward put these area to an classifier of images [3].

Two examples for the most mainstream ways on the 1 stage group are You Only Look Once (YOLO) [4] and single shot detector (SSD) [5] architectures. The outputs of the two architectures are the bounding boxes on every discovered thing, with the class name and the confidence score of it, however not to the all of the photo. These are put to a lot of routinely divided locations and to various scales and aspect-ratio of rectangles named boxes.

2.1. Single shot detector

The method of single shot detector (SSD)[5] utilized in the detection of objects for detecting humans in live videos of surveillance camera. R-CNN method generates zone suggestions for specifying things, have a more good accuracy, yet slower processing of frame rate per second (FPS). SSD for live processing make the FPS a lot better utilizing a lot of scales and default-boxes in one process. This obeys the rule of feed forward convolution, that makes a score based on the presence of standard size bounding boxes and the presence of object class samples in the boxes. Afterwords, the final fixations are produced with the nonmaximum suppression (NMS) step. It is a two step process: an architecture with three main parts, applying folding filters to extract property maps and detect objects. The first part is a basic pretrained network. Then on the 2nd step, the property of multiscale utilized such that the convolution filters sequence is spurt on the network that is basic. Eventually in the final section is a unit of suppression that is nonmaximum to eliminate boxes overlapping also every box gets single object.

Localization loss can be defined as the failure to match of discrepancy of the ground truth labels and the boundary box predicted class labels. Identifications get penalized from matches that is positive by the single shot detector. As the identifications from the matches that is positive are meant to be more close for the ground truth, which is the least error rate, negative matches are neglected to avoid dilution of the predictions. Among the g ground truth box and the l predicted box there is a localization lost which is classified as the smooth L1 lost with cy,cx as the standard d bounding box offset of h for height and w for width.

$$\lambda_{loc}(x,l,g) = \sum_{i\in Pos}^{N} \sum_{m\in cx,cy,w,h} x_{ij}^{k} smooth_{L1}(l_i^m - \hat{g}_j^m) \tag{1}$$

$$\hat{g}_j^{cx} = \frac{(g_j^{cx} - d_i^{cx})}{d_i^w} \quad , \quad \hat{g}_j^{cy} = \frac{(g_j^{cy} - d_i^{cy})}{d_i^h} \tag{2}$$

$$\hat{g}_j^{w} = \log(\frac{g_j^{w}}{d_i^w}) \quad , \quad \hat{g}_j^{h} = \log(\frac{g_j^{h}}{d_i^h}) \tag{3}$$

$$x_{ij}^{p} = \begin{cases} 1 & \text{if } 0.5 < \text{loU among j the ground true box and i the standard box in the p class} \\ 0 & \text{or else} \end{cases} \tag{4}$$

The confidence loss is utilized in predicting a group for each detected object. For each positive match identification, as maintained by the score of confidence of the class and the lost get penalized. For negative match identification, as maintained by the score of confidence of "0" the corresponding class. "0" class classifies no thing got identified and returns a void and the lost get penalized. It is measured as softmax lost over a lot of groups confidence c (class score).

$$\lambda_{conf}(x,c) = -\sum_{i\in Pos}^{N} x_{ij}^{p}\log(\hat{c}_i^p) - \sum_{i\in Neg}\log(\hat{c}_i^0) \qquad \text{where} \qquad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \tag{5}$$

the amount of standard boxes that is matched is defined by N. Function of loss which is final get computes such:

$$\lambda(x,c,l,g) = \frac{1}{N}(\lambda_{conf}(x,c) + \alpha\lambda_{coord}(x,l,g)) \tag{6}$$

the amount of match that is positive is defined by N also the localization loss weight is defined by .

2.2. Faster RCNN

In the 2-stage group, faster RCNN (region proposal CNN) [6] reported improved results over many benchmark architectures which depend on 2 networks that is independent, i.e. one network for classification and one network for region-proposal. Faster RCNN reported its benchmark outcomes with regard to precision, particularly in the localisation to object. The quality is judged utilizing the precision metric for mean-average, to quantify the performance of identification as a function of the recall, that is the ratio of things that are identified on these benchmark approaches.

Faster RCNN is improved and achieved from previous RCNN [7] and fast RCNN [8] through a selective outsourced bidding approach search (SS).[7] Ren et al. suggested regional proposal network (RPN) using CNN models, which used VGGNet [9], ResNet [10] and many others similar networks. More fast RCNN architecture which the RPN module groups the objects and assigns categories for each detected object using the region of interest (RoI) creating a pool of projected attribute maps [6].

Faster RCNN consists of 2 modules: detector of fast RCNN and RPN. Multitasking lost function consists of classification lost (L_{cls}) also regression loss of the bounding box (L_{reg}) identified in this following equation:

$$L_{cls}(p_i,p_i^*) = p_i^*\log(p_i) - (1-p_i^*)\log(1-p_i) \qquad , \qquad L_{reg}(t^u,v) = \sum_{x\in x,y,w,h} L_1^{smooth}(t_i^u - v) \tag{7}$$

$$L_1^{smooth}(q) = \begin{cases} 0.5q^2, & if \quad |q| < 1 \\ \text{q-0.5}, & \text{otherwise} \end{cases} \tag{8}$$

The main drawback of these networks is its computation complexity caused due to usage of two independent networks. Conversely, with the architectures of 1-stage, a defined amount and size of bounding boxes in each layer make the classification, and afterwords via regression adjust the objects that is detected localization. On the other hand, 2 independent networks predict slow on a very embedded computation platforms due to

[7] Open Images Dataset V6 + Extensions [online]. Website https://storage.googleapis.com/openimages/web/index.html [accessed 23 September 2020].

the heavy computation burden due to its individuality, because two networks consume much RAM when compared to 1-stage architectures. In 1-stage architectures of detection, data get classified on bounding boxes at defined layers that predetermined amount and size, afterwords via regression. Usually the architectures got the same performance and more fast than the others, the 2-stage ones [5]. Two stage architectures consume decent amount of computational power to extract features on more quantity of dataset. Given that social distancing application depends on spatio-temporal data which complies of videos, the flaws such as not accurate identification for tiny objects and not precise localisation are not more important than the live performance. However, trade-offs such as getting more bounding boxes and resolution of inputs is adapted to overcome such flaws in detection. Depending on the analysis we mentioned beforehand, we decide to utilize the state-of-the-art 1-stage object identification method YOLO v3.

To estimate the distance from one person to another, there are different sensors: infrared ray, laser ultrasonic etc. [11]. These sensors are commonly utilized in civil and defence fields. Yet the utilization of multisensors get the system to a more complex level, also increases the cost, as it is planed to utilize a cost effective camera with a monocular for detection of object, which looks to be reliable commercially to approximate the interval and detection of objects using one sensor utilizing a camera with a monocular. On condition that the data sets utilized for object identification had the interval between object-to-object as its main, idea of distance can be straightly extracted via applying a regression output to the CNN, however no such data set exists. One solution is to use pixel based distance measure. This method takes coordinates of bounding boxes and measure distances between another bounding boxes. To plot the distance and for measuring the distance between each object, Euclidean distance is used. Accuracy of this method however depends on the camera angle. For the best results camera should be in top-view position.

3. Methods

In this section, we first introduce the object detection pipeline. As illustrated in Figure 2, a semantic network is built on top of the classic convolutional neural network. Afterwards, details are provided of the neural network architectures utilized for for social distance measurement. Finally, implementation details and lost function are introduced.

3.1. Network architecture

We build our framework on the YOLO v3-416 model. Every bounding boxes object that is detected is the architecture output, its score of confidence and group which are a collection of systemically spaced coordinates and for different aspect ratio and scales of rectangles named boxes.

YOLO way could make the prediction of its kind and position of an object just by seeing the image one time. YOLO takes into account object identification issue as regression job in place of classification to allocate anchor class probability boxes. A single convolution predicts the network and the bounding boxes for each class along with their class possibilities.YOLO got 4 main versions: v1, v2, v3 also v4. v1 of YOLO [12] is inspired by the designed GoogleNet (startup network) for classification of object in the photograph. Twenty four folded layers and two connected in full way layers consists the network. YOLO in place of modules for inception utilized via GoogleNet v1 is straightforwardly a layer for reduction and it follows layers that is convolutional. Then, YOLO v2 [13] significantly improved the accuracy and became more fast. YOLO v2 utilizes Darknet-19 for the main network. Classification of objects utilized an output softmax layer, nineteen convolution layers, and five maximum pooling layers. YOLO v2 performed significantly better than its last version (YOLO v1).

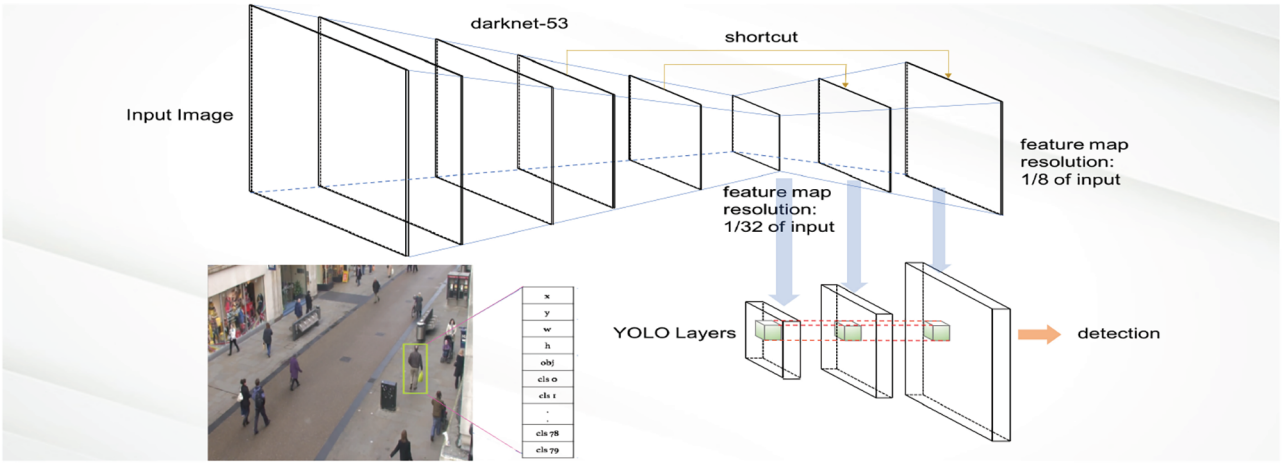

Figure 2. Structure detail of YOLO v3, that uses three scale predictions and Darknet-53 as the backbone network. This graph shows the way YOLO v3 works. YOLO is a very fast and used worldwide model. We can extract the bounding boxes and get the coordinates of each object on the image.

Enhancement in AP, classification of objects score and FPS. As opposed, YOLO v3 makes multiple label classifications using logistics classifiers rather than utilizing softmax. Unlike Darknet-19, Darknet-53 now that is consisted from blocks (short links) in the company of top sampling layers to joining also for the depth of network. YOLO v3 produces 3 predicts to every spatial position in various scales in one photo and remove the issue of not detecting small objects efficiently. Each forecast is calculated and tracked. Figure 2 is a schematic description of the YOLO v3 architecture, we utilize.

For a better performance of precision (mAP), a rich dataset is required. The model was trained on COCO dataset with a total sixty four thousand one hundred and fifteen photo of people which cast 5.6 mAP enhancement on Pascal VOC data set[14] in the company of two thousand photo of people. The amount of the classes does not notably effect the velocity of the inference. The model of 1 class reported a 0.4 FPS more fast from the model of 80 classes. However, the quantity of parameters directly impacts the inference time. The fully connected layers amount parameters are directly effected by the classes suppression on the network, which although is a small part of the whole model [3].

3.2. Loss function

We have utilize a loss function for YOLO v3 as a combination of loss of localization (bounding box regression), loss of confidence for score of classification and entropy of cross, interpreted as follows:

$$\lambda_{cord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}[(x_i-\hat{x_i})^2+(y_i-\hat{y_i})^2+(w_i-\hat{w_i})^2+(h_i-\hat{h_i})^2]+$$

$$\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}(-\log(\sigma(t_o))+\sum_{k=1}^{C}BCE(\hat{y}_k,\sigma(s_k)))+\lambda_{noobj}+\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{noobj}(-\log(1-\sigma(t_o))) \quad (9)$$

The error of coordinate weight is indicated as λ_{cord}, the photos amount of grids is indicated as S^2, finally for every grid the amount of bounding boxes that is generated is indicated as B. The confined object in the grid i and j^{th} bounding box is described as $1_{i,j}^{obj} = 1$, else gets the value of 0.

3.3. Implementation details

With having Darknet-53 as backbone and the YOLO v3 model is trained using Pascal Titan X GPU using dataset acquired from COCO dataset [15]. Oxford town centre dataset[8] is used for evaluation. The footage in the Oxford Town Centre is used for simulating the distance between each identified person. In case of faster RCNN, the images from the dataset and evaluating photos are sized changed to pixels of P in the edge which is shorter that has six hundred and one thousand and twenty four to low and high quality, during the time that on YOLO v3 and SDD, the photos get scaled to the stable dimension p × p within p valued four hundred and sixteen. In the time of training, the models performance is repeatedly watched utilizing the mAP accompanied by human detection overall loss, localization, and classification.

Figure 3 is a zoomed in sample taken from the dataset. In the first stage, we apply our detection model to this dataset to extract bounding boxes for each person in a frame. In Figure 4 we can see the detection and classifying output of the YOLO v3. Normally it operates on bound boxes, but we illustrate only the center of the bounding box for simple visualization. To illustrate the center points, we take the coordinates of the center of the bounding box. After this stage we are measuring the Euclidean distance. To measure the Euclidean distance, we need to know the difference distances of these centers. To measure the difference of centers, we are extracting the point coordinates of each center. This operation needs to be done for all frame sequences. Then, we are taking absolute value of output number. The main outputs from these frames are its x and y coordinates. In the final stage we measure the Euclidean distance using x and y coordinates. Euclidean distance is formulated by:

$$E = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{10}$$

From the Figure 5 we can calculate distance, for instance;

$$\delta(x_{23}) = |x_2 - x_3| \quad , \quad \delta(y_{23}) = |y_2 - y_3| \tag{11}$$

$$E = \sqrt{\delta(x_{23})^2 + \delta(y_{23})^2} \tag{12}$$

$$x_{ij}^{p} = \begin{cases} \text{Draw red line, if E} < \text{Threshold} \\ \text{Do not draw red line, if E} > \text{Threshold} \end{cases} \tag{13}$$

Using this formulation, we can measure the distance between detected humans in the frame sequences as illustrated in Figure 5. After measuring the euclidean distance, we should keep a threshold for human-to-human distance. This number is dependent on the camera angle and how far the camera is from the ground. The angle of camera has a significant effect on the outcomes and we have utilized the best camera angle position as the top-view.

[8]MegaPixels Project (2020). Oxford Town Centre [online]. Website https://exposing.ai/oxford_town_centre/ [accessed 19 September 2020].

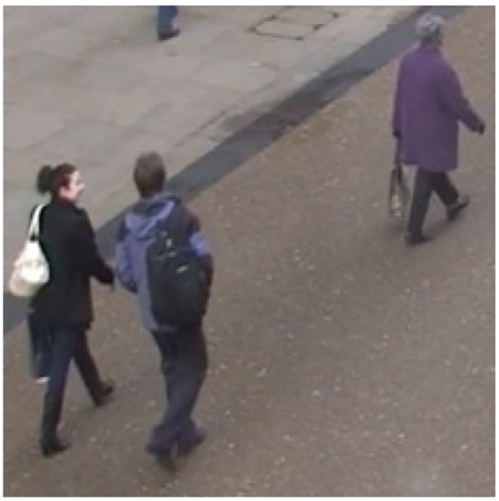

Figure 3. A frame of the Oxford Town Center dataset we have used is shown.

Figure 4. There is a point in each pedestrians bounding box center. In our output, we delete each person's bounding boxes on the left and put a point in the center of the bounding boxes. On the right, only those points are visible.

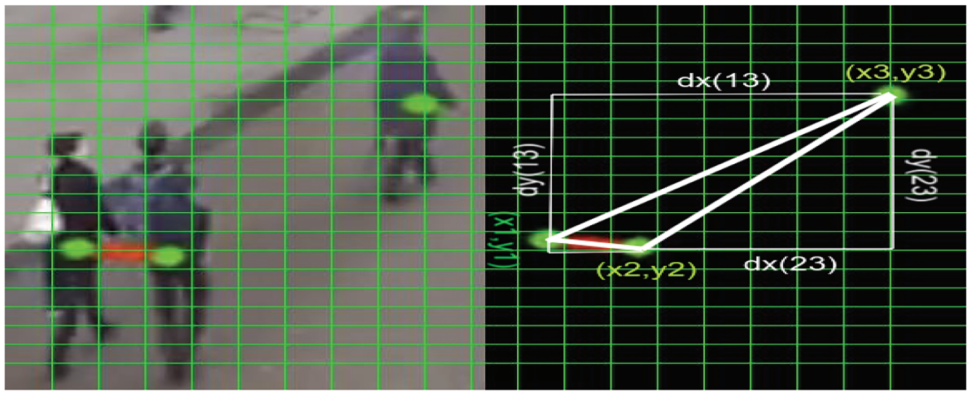

Figure 5. Measuring the distance between pedestrians with Euclidean distance formula. In this visual, it is seen how we calculate the distance between people. It applies Euclidian distance between people and gives a warning when it goes below a certain threshold.

Figure 4 illustrates output of the model and it calculates same time distances between humans. We illustrate the detected the pedestrians and alert levels with green dots and red lines. The distance between 2 humans, even if it is a little bit lower than the threshold, the model will make a line between these people. This regulates which specific area, there is a social distancing violation through camera. The threshold is fixed to 45 based on our dataset angle which was projected during the video capturing. There is no fixed threshold and can be changed based on the camera angle. If the camera is faced at larger distances, the threshold should be considerably reduced because the accuracy declines as the camera is placed far away from the crowd.

4. Results from experiments

Here, results are provided that got acquired from the experiments for object detection and distance estimation tasks. All the results are based on model trained on COCO dataset. We tested our framework on two separate datasets: COCO test-dev dataset and Oxford Town Centre dataset.

4.1. Object detection

We compare a variety of object detection methods, we introduced in Section 2, with regard to speed and accuracy. The most appropriate method is selected for real-time video sequence analysis for social distancing estimation.

4.1.1. YOLO v3 vs. SSD

YOLO v3 could have more good performance from the SSD with regard to accuracy and speed as reported in previous studies.[9] Hence, it got decided to frame the inference of YOLO v3 and SSD to work in the same scenario to see which is better. The dataset on which YOLO v3 and SSD are trained is MS COCO dataset on "person" class. The dataset contains images and its corresponding annotation files in the COCO format. We used pretrained models.

The models of the SSD and YOLO v3 were trained and evaluated with the COCO dataset [15]. Using mean average precision (AP) every learning classes evaluation base got conducted. The classes within the base training represented by AP which disposes the final accuracy for each class. Table 1 hand over outputs from the evaluation and the inference of SSD and YOLO v3. A common $AP_{@k}$ calculation formula is written as

$$AP_{@k} = \frac{1}{GTP} \sum_{i=1}^{k} \frac{TPseen}{i} \tag{14}$$

the query total amount of positives of ground truth is indicated as GTP and the amount of positives that is true seen till k is indicated as TP seen.[10]

In Figure 6 and Table 1 we can see that YOLO v3 is more accurate and has more FPS than SSD. These 2 parameters are crucial for model selection, because we want to run our model on real-time video sequences.

[9]Synded (2020). The YOLO v3 Object Detection Network Is Fast! [online]. Website www.syncedreview.com/2018/03/27/the-yolov3-object-detection-network-is-fast [accessed 20 September 2020].

[10]Towards Data Science Inc. (2020). Breaking Down Mean Average Precision (mAP) [online]. Website www.towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52 [accessed 20 September 2020].

Table 1. Performance of trained model (SSD and YOLO v3) COCO dataset and prediction on Oxford Town Centre dataset.

Architecture	mAP	FPS [12]
SSD[5]	46.5	19
YOLO v3[4]	55.3	35

Figure 6. SSD inference (left) vs. YOLO v3 inference (right). There is a comparison of SSD and path in this visual. YOLO is seen to give better and faster results than SSD. That is the reason why no SSD was used.

4.1.2. Faster RCNN vs. YOLO v3

We run a similar comparison between RCNN and YOLO v3 methods for detection of object. Table 2 indicates that faster RCNN has similar accuracy to YOLO v3. On the flip side, the FPS is lower than the YOLO v3.[11] This concluded that if the work needs high resolution precision, faster RCNN can be deployed. But if the speed and inference are essential, YOLO v3 would be optimal which is more fast than RCNN and more fast than SDD. On the contrary, accuracy and speed are the main parameters, then SSD is an ideal one. YOLO v3 on the other hand accuracy is nearly good as fast RCNN. It can be said it is a good replacement. The model of the faster RCNN and model of YOLO v3 got evaluated was trained with learning base of COCO. In Figure 7 we can see that YOLO v3 is more accurate and has more FPS than faster RCNN.

Table 2. Faster RCNN and YOLO v3 inference.

Architecture	mAP	FPS [12]
Faster RCNN[6]	59.1	6
YOLO V3[4]	55.3	35

4.2. Distance estimation

After training the same dataset with faster RCNN, SSD and YOLO v3, it is seen that the model of more fast RCNN got the minimal lost with maximum AP, anyway, got the most low FPS, then it is useless for the real life situations. Moreover, as the compression to YOLO v3, SSD got better outputs with balanced AP, training time, and FPS score. As illustrated in Figure 8, the model is capable to detect each and every person present in the frame. The model works for both image format and video format due to its high FPS.

[11]Towards Data Science Inc. (2020). R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms [online]. Website www.towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e [accessed 21 September 2020].

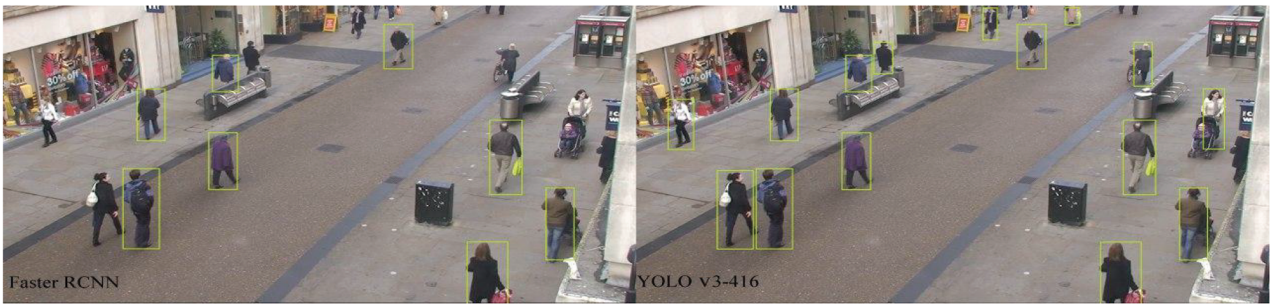

Figure 7. Faster RCNN inference (left) vs. YOLO v3 inference (right). This vision is a comparison of faster RCNN and path. Where it is seen, YOLO is faster than faster RCNN, but worse than faster RCNN in terms of accuracy. We chose the path for this comparison since we were going to make a real time system.

Figure 8. One point in each pedestrians bounding box center. The distance is calculated between the center points. On the right half side we can track the points.

Table 3 defines the FPS comparisons and prediction comparisons among different architectures. TOP-1 defines that each predicted class is same as the target label. TOP-5 defines the best label that is target from five predictions (the other five has the best probabilities).

Table 3. Comparison of backbones. Accuracy and FPS for various networks.The ResNet-151 backbone has highest accuracy but lowest FPS and Darknet-19 has highest FPS but lower accuracy.For comparing the accuracy ResNet-151 and Darknet-53 near to each other but in FPS Darknet-53 is better. The optimal choice is Darknet-53 and we selected it.

Backbone	Top-1	Top-5	FPS [4]
Darknet-19[13]	74.1	91.8	171
ResNet-101[10]	77.1	93.7	53
ResNet-151[10]	77.6	93.8	37
Darknet-53[4]	77.2	93.8	78

The video is nothing but sequence of frames. The model plots the distance between each and every person in the frame. Suppose person 'A' is in the vicinity of person 'B' and the model computes the distance between them. If the distance is in the range of the fixed threshold, it marks as red alert, if it lies outside the range, it marks as safe zone. Here, person 'B' is taken as base reference. The similar approach is followed for multiple people. Even in case of multiple people, each person is taken base reference with respective to the latter present

in their vicinity. For example, 10 persons, A, B, C, D, E, F, G, H, I, J. If A, B, C, D are together, each one of them is a base reference to each other. This multiple independent reference approach can give accurate results without leaving any leaf turned down.

The trained model of YOLO v3 is then used for watching the social distance on the surveillance camera and even can be deployed on edge devices for real-time standalone surveillance. We have tested the model with Oxford Town Centre dataset and real-time CCTV footage deployed at several civil areas. Our model enhances a real-time deployment ready on both high-end server and edge devices which can operate remotely such as Nvidia Jetson Nano.

As shown in Figure 8, the output of the model is useful and meaningful in terms of distancing plotting and measuring between each person for a easy vision based surveillance for monitoring social distancing. We can see all the results in Table 4. YOLO v3 method is the most accurate and fastest model for detecting pedestrians and measuring the distances between them. The results showed in Figure 8, is a commercial complex from Oxford Town Centre dataset, where the crowd is being monitored and corresponding results, i.e. distance between each person if they are near to each other is being drawn with a red line, which indicates an alert. Or else, a colored point is being printed for each person. This not only gives the social distancing monitoring, but also the density present in each street. Based on these results, the authorities can control the area to avoid formation of dense crowd.

Table 4. Comparing the SSD, faster RCNN and YOLO v3 architectures on the MS COCO dataset with parameters like mean average precision (mAP) and FPS. Faster RCNN has highest mAP but lowest FPS. For the comparison of SSD and YOLO v3, YOLO v3 is better than SSD on mAP and FPS. For the optimal choice we selected YOLO v3.

Architecture	mAP	FPS [12]
SSD [5]	46.5	19
Faster RCNN[6]	59.1	6
YOLO v3[4]	55.3	35

Figures 8 and 9 illustrate the prediction output taken from a street. The model plots the distance between each person. As Figure 9 camera is more far from humans than Figure 8 camera we gave lower threshold value than Figure 8. We tried many times to find optimal value for threshold.

Figure 9. Distance calculation for an example frame, where pedestrians are marked using bounding box center. The distance is calculated between the center points with Euclidean distance. Our algorithm detects a violation of social distancing measures.

5. Discussion

We suggest a method for person detection as well as distance estimation in street view videos. This work aims providing a vision-based surveillance in all scenarios and plots the distance between each person that are captured in the video frame. We make a comparison of state-of-the-arts on detection of objects architectures and YOLO v3 outperformed SSD and faster RCNN in terms of inference FPS and accuracy. The model also provides an edge device compatibility which can be deployed as a standalone device and can be monitored remotely anywhere and anytime. Furthermore the distances acquired from the model can be used as statistical parameters. Statistical physics and epidemic processes research got a strong history together [16, 17]. Informed predictions about epidemic spreading can be made using values obtained from basic mathematical models fitted to the statistical data [18].

YOLO v3 outperforms with the rest SSD and Faster RCNN on object detection, after rigorous trail and error on Oxford Town Centre dataset, threshold is fixed on person-to-person minimum distance, the threshold can be varied accordingly. Often the threshold depends on the angle of the camera, and position of the camera from the ground level. Hence, the angle and the position of the camera from the ground, threshold can be changed accordingly.

Although YOLO v3 outperformed the other state-of-the-arts methods, there are several limitations of this model. The video quality is essential factor for providing accurate assessment of and depth factor is important too. With the appropriate datasets availability, we aim to validate the calculated distances at each frame. One natural application of our framework could be flagging the violations of social distances with a fixed threshold and creating an alarm mechanism. With the current developments in today's society we believe automatic detection tools such as the one proposed in our paper would be an essential tool in protecting the society in spread of pandemic diseases.

The biggest impact from this work would be a remote monitoring surveillance on social distancing in the civilian areas. This surveillance could help in monitoring the areas whether social distancing is maintained in order to stop the human-to-human transmission of corona virus. Many governments have been lifting up the lockdown, but the social distancing norm remains the same in almost all countries. Hence, the model can be deployed near airports, shopping complexes, workplaces, restaurants, public transport deports, markets, and several civilian prone areas where dense crowd can be expected during their economic activity. As the vaccine has not yet been formed, more care is required to maintain the flattened covid curve. To achieve this, stopping the super spreading is the only option left out to all nations. Hence, strict surveillance is the foremost priority for the governments to ensure human-to-human transmission does not occur, especially due to breaking social distancing norm.

The future work lies in deploying mask detection and temperature detection of each person in the crowd. Combining multiple models in an ensemble learning framework can be instrumental in an end-to-end pipeline for complete monitoring on each person. A person unknowingly may have symptoms of coronavirus disease, and can be traced easily and stop the unintentional transmission. A demo of the system and the source code can be found on our Github page.[12]

[12]Github Inc. (2020). Subjective-analysis-of-social-distance-monitoring-using-YOLO-V3-architecture1and-crowd-tracking-syst [online]. Website www.github.com/muraatozbek/distance [accessed 15 October 2020].

References

[1] Amodei D, Ananthanarayanan S, Anubhai R , J Bai, E Battenberg et al. Deep speech 2: end-to-end speech recognition in English and Mandarin. Proceedings of the 33rd International Conference on Machine Learning 2016; 48: 173-182.

[2] Pouyanfar S, Sadiq S, Yilin Y, Haiman T, Yudong T et al. A survey on deep learning: algorithms, techniques, and applications. Association for Computing Machinery Computing Surveys 2018; 51 (5): 1-20. doi: 10.1145/3234150.

[3] Chen Z, Khemmar R, Decoux B, Atahouet A, Ertaud J. Real time object detection, tracking, and distance and motion estimation based on deep learning: Application to smart mobility. Eighth International Conference on Emerging Security Technologies (EST); Colchester, United Kingdom; 2019. pp. 1-6.

[4] Redmon J, Farhadi A. YOLOv3: an incremental improvement. arXix 2018; 1804.02767.

[5] Wei L, Dragomir A, Dumitru E, Christian S, Scott R et al. SSD: single shot multibox detector. In: European Conference on Computer Vision; Rome, Italy; 2016. pp. 21-37. doi: 10.1007/978-3-319-46448-0_2

[6] Shaoqing R, Kaiming H, Ross G, Jian S. Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems 2015; 28: 91-99.

[7] Girshick R, Donahue J, Darrell T, Malik J.Rich. Feature hierarchies for accurate object detection and semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition; Ohio,USA; 2015. pp. 580-587.

[8] Girshick R. Fast R-CNN. In: IEEE International Conference on Computer Vision (ICCV); Montreal, Canada; 2015. pp. 1440-1448. doi: 10.1109/ICCV.2015.169

[9] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations(ICLR); San Diego,USA; 2015. pp. 1-20.

[10] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition(CVPR); Las Vegas, NV, USA; 2016. pp. 770-778. doi: 10.1109/CVPR.2016.90

[11] Palacin J, Palleja T, Tresanchez M, Sanz R, Llorens J et al. Real-time tree-foliage surface estimation using a ground laser scanner. IEEE Transactions on Instrumentation and Measurement 2007; 56(4): 1377-1383. doi: 10.1109/TIM.2007.900126

[12] Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: unified, real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); Las Vegas, NV, USA; 2016. pp 779-788. doi: 10.1109/CVPR.2016.91

[13] Redmon J, Farhadi A. YOLO9000: Better, faster, stronger. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); Honolulu, HI; 2017. pp. 6517-6525. doi: 10.1109/CVPR.2017.690

[14] Everingham M, Gool L, Williams K, Winn J, Zisserman A. The Pascal visual object classes (VOC) challenge. International Journal of Computer Vision 2010; 88 (2): 303-338. doi: 10.1007/s11263-009-0275-4

[15] Lin T, Maire M, Belongie S, Bourdev L, Girshick R et al. Microsoft COCO: common objects in context. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (editors). Computer Vision – ECCV 2014. Lecture Notes in Computer Science, Vol. 8693. Cham, Switzerland: Springer, 2014. doi: 0.1007/978-3-319-10602-1_48

[16] Pastor-Satorras R, Castellano C, Van P, Vespignani A. Epidemic processes in complex networks. Review Modern Physics 2015; 87 (3): 925-979. doi: 10.1103/RevModPhys.87.925

[17] Wang Z, Bauch T, Bhattacharyya S, Onofrio A, Manfred P et al. Statistical physics of vaccination. Physics Reports 2016; 664: 1-113. doi: 10.1016/j.physrep.2016.10.006

[18] Petropoulos F, Makridakis S. Forecasting the novel coronavirus COVID-19. PLoS ONE 2020; 15. doi: 10.1371/journal.pone.0231236