The Application of Yolov4 And A New Pedestrian Clustering Algorithm to Implement Social Distance Monitoring During The COVID-19 Pandemic

Junxiao Li1,* and Ziao Wu2
1Xidian University, Shanxi, China
2University of Electronic Science and Technology of China, Chengdu, China

*Corresponding author: lijunxiao@xidian.stu.edu.cn

Abstract. Based on the global spread of COVID-19 epidemic, this paper implements an application of social distance monitoring in public places, aiming to control the spread of virus by controlling the social distance of pedestrians. The application mainly uses YOLOv4 object detection algorithm and DeepSORT multiple objects tracking algorithm to detect and label pedestrians, and uses affine transformation to calibrate the scene to a more intuitive bird’s-eye view. Based on the analysis of pedestrian movement, the author poses a novel pedestrian clustering algorithm to avoid the impact of peers on the monitoring results. Finally, three indicators are selected to classify the pedestrian to analyse and evaluate the risk of virus infection in a certain place. In this paper, comparing the yolov4 algorithm with the most suitable performance of other research results, the author indirectly concludes that yolov4 is the most suitable method in the application of social distance monitoring. And the new pedestrian clustering algorithm plays a role in the application, which improves the practicability of the application. Through the experiment, it is found that the social distance in the scene can be monitored in real time and accurately, and there is a correlation between the average distance of pedestrian clusters, the density of pedestrian clusters and the infection index, which can be used to evaluate the safety of places and prevent the spread of virus.

1. Introduction
COVID-19 was officially named by the World Health Organization on January 12, 2020 [1]. It was first discovered in Wuhan, China in December 2019 [2]. Its main symptoms [3] are fever or mild cough, pneumonia, and if very serious, even death.

According to the World Health Organization (WHO) technical report [5], the main modes of transmission of COVID-19 are direct transmission, aerosol transmission, contact transmission and mother-to-child transmission. The main ways to prevent the virus are to wear masks, wear protective clothing, reduce outdoor activities, and maintain effective social distance [6].

Among them, controlling the social distance between people has become the simplest and most effective way to prevent the spread of the virus. Social distance [7] refers to the different spatial distances between individuals, groups, and between individuals due to different degrees of closeness or distance. Maintaining social distancing can block the path of virus transmission and prevent the increase of risks of infection. According to the WHO, maintaining a certain social distance is an effective measure to
prevent the COVID-19 pandemic, and it is recommended that people maintain a social distance of at least 1 meter between them [8].

For ordinary people, home isolation is the most effective preventive measure, but for those who maintain social stability in the handicraft and manufacturing industries people have to gather together to continue working, and for the long-term development of social economy and people’s lives, it is unlikely that going out in mildly epidemic areas will be explicitly prohibited. Therefore, based on the above background, in order to ensure the safety of the working environment around employees and ensure their social distance in the workplace, Landing AI has developed a social distance detection tool that supports artificial intelligence, which can monitor people from real-time images captured by cameras whether a safe distance is maintained [9].

On April 23, 2020, with the advent of YOLOv4 [10], the accuracy and future of the object detection method have been redefined again. This experiment is also based on the introduction of YOLOv4 and the inspiration of Landing AI. In the study of Narinder Singh Punn’s team [11], they compared current mainstream object detection algorithms, including Faster R-CNN, SSD, YOLO, and so on, and finally chose the YOLOv3 algorithm, which have a better performance in terms of accuracy and speed of in social distance monitoring. This experiment is also based on the predecessors, directly compare the performance of YOLOv3 and YOLOv4, to determine which method to use in our experiment of pedestrian detection.

However, including Landing AI and other social distance detection algorithms based on YOLOv3 mainly focus on the distance between pedestrians, without considering that people who know each other are also close in social distance, in that case distance between pedestrians has little impact on the spread of the virus. If there are many peers at the same time, it may lead to inaccurate results. Therefore, based on the practicability and rationality of social distance detection in real scenes, a new pedestrian clustering algorithm is proposed in this paper. By judging the characteristics of the pedestrian’s movement state after object tracking, peers can be identified. The purpose of this innovation is to reduce the inaccuracy caused by the close distance between peers when calculating the social distance.

In the current study, we will perform the application that detects people’s social distance in outdoor places during COVID-19, and assess the risk of infection in this area, as well as provide suggestions for people who are too close to each other.

In Section 2, we will introduce some related works. In Section 3 and 4, we will focus on object detection method, YOLOv4, and object tracking method, Deep-SORT. In Section 5, we will mention the calibration operation. In Section 6, we will propose a novel pedestrian clustering algorithm. In Addition, we will introduce social distance monitoring and propose several indicators to analyze the results in Section 7. Our steps and results will be shown in Section 8 and Section 9. In Section 10, we will draw our final conclusion and discuss research directions and improvement measures in Section 11.

2. Related Works

Pedestrian detection is an application in the field of object detection. It belongs to the most basic steps of many computer vision tasks. Looking back at the existing target detection results, it can be divided into two categories: one-stage and two stage, as shown in Fig. 1.

Two stage means that the detection should be completed in two steps, that is, locating first, then classification. The most famous algorithm is R-CNN [12] [13]. R-CNN is the first algorithm to apply deep learning to the field of object detection. The basic principle is to extract all ROI of the image for selective search by region proposal, and use SVM to determine the object category in the ROI, and finally use the regression to refine the ROI’s position. However, R-CNN will generate a lot of redundancy when extracting features of all ROI, which has a serious speed bottleneck. But this problem is also well solved by using SPPNet in the next Fast R-CNN [14]. Subsequently, Faster R-CNN [15] [16] chose to use RPN network instead of selective search, which once again greatly improved the detection speed.
One stage does not need to search for candidate areas separately, and it is in place at one time. Compared with the accurate detection results of R-CNN based on region, one stage detector is fast. However, it usually needs to tradeoff between accuracy and real-time processing speed. When the object is too close or too small, the detection results will be inaccurate. SSD [17] uses VGG16 [18] network as feature extractor, and then regresses on the feature map to get the location and category of objects. The other is YOLO method [19], which uses Darknet to do feature detection, then partially smooths the feature map and stitches it with another lower resolution feature map, and uses convolution kernel to predict. Then, the proposed YOLOv2 [20] improves the map from 63.4 to 78.6 by using clustering to extract the prior frame scale and constrain the predicted border position. Another yolo9000 [20] continues to enhance the ability to recognize categories while maintaining the speed and accuracy.

The real improvement of YOLO class algorithm is YOLOv3 [21], which adopts a new network structure, Darknet-53 [22]. The residual module is added to the network to solve the gradient problem of deep-seated network, and multi-scale features are used for object detection so that more fine-grained features can be detected. Finally, softmax is changed to logistic when predicting object categories; it is used to support the prediction of multi-label objects.

The main carrier of social distance detection in this experiment is pedestrians, so the technology of Multiple Object Tracking will bring great help to this experiment. Multiple Object Tracking (MOT) technology is mainly to give a continuous image sequence, find out the moving objects in each frame, and give the same object an accurate and specific ID. With the continuous development of target detection technology, tracking based Detection has gradually emerged as a leader in the MOT field. However, by dealing with the optimization problem of the whole process, the multi-object tracking algorithm, flow network formulations [23] [24], and probabilistic graphical models [25] cannot be applied to real-time scenes that determine ID in every time step; therefore, they losing competitiveness due to the rise of target detection and the gradual demand for real-time. At the same time, the more traditional Multiple Hypothesis Tracking (MHT) [26] and the Joint Probabilistic Data Association Filter (JPDAF) [27] [28] have received renewed attention due to the emergence of Tracking based Detection and the characteristics of data association on a frame-by-frame basis.

Among these algorithms, SORT [29], fully known as Simple Online and Real-time Tracking, can match the SOTA algorithm in 2016 with a speed of 260hz, which is 20 times faster than the former, although it only combines the ordinary Kalman filter with the Hungarian algorithm. In the experiment, the algorithm uses Fast R-CNN [14] and traditional pedestrian detection model ACF [30] to detect, and only uses the location and size of the detection frame to solve the motion prediction and data association. However, because sort ignores the surface features of the detected object, it can only come in effect when the uncertainty of the pose estimation is low. But this also prompted the DeepSORT algorithm [31].

For the detection of social distance, this task was first proposed by the Landing AI company [9], in order to ensure the safety of the working environment around employees and ensure their social distance in the workplace. They have developed a social distance detection tool that supports AI. Through the three main steps of calibration, monitoring and measurement, they can monitor whether people keep a safe distance from the real-time images captured by the camera. Due to the call and inspiration of Landing AI, the number of researches on social distance detection has increased suddenly. Basile Roth,
a graduate student from Montreal, Quebec, Canada, did a similar study [32]. However, we found that the detection algorithm used in the above experiment encountered a major problem, that is, one cannot identify the people of the same group, and will always mistakenly classify the peers as susceptible to infection. However, we know that the risk of virus transmission does not exist in the close contact of a group of peers. So, in the next module, we propose a new clustering algorithm to solve this problem.

Then Narinder Singh Punn team [11] also explored the performance of object detection algorithm, and selected improved YOLOv3 and DeepSORT to detect social distance. Dongfang Yang, Ekim Yurtsever team [33] from the Ohio State University have made an early warning system for social distance. By linear regression between the social density and the number of people who are too close to each other in three different situations, the maximum social density \( \rho \) in the place is obtained to avoid the spread of the virus. At the same time, they also considered four moral factors into the early warning system, which enhanced the universality and practicability of the task. And with the advent of YOLOv4 [10], the accuracy and future of object detection methods are redefined. In this experiment, we will also focus on comparing the differences between YOLOv4 and YOLOv3 in social distance detection, and show the advantages of YOLOv4 algorithm.

3. The Object Detection Algorithm: YOLOv4

The main purpose of YOLOv4 [10] is to create a fast object detection system suitable for the actual working environment. It is hoped that it can be well trained and used on a conventional GPU. In the training process, bag-of-freebies and bag-of-specials are used to improve and optimize training results and training speed.

YOLOv4 method features mainly include:

3.1. Selection of Architecture

A target detection model consists of three parts:

- **Backbone.** It is used to extract input shallow features (edges, colours, etc.), this module can learn from the trained network;
- **Neck.** It is used to enhance the understanding and extraction of features. Processing, combining and analysing the extracted shallow features, and optimizing according to the target of the model;
- **Head.** Outputting according to the needs of the model, such as classifier, detection frame, image segmentation, etc.

The goal is to find the best balance between the input network resolution, the number of convolution layers, the parameter number \((\text{filtersize}^2 \times \text{filters} \times \text{channel/groups})\) and the number of layer outputs (filters). We choose CSPDarknet53 [34] backbone, SPP (add module) [35], PANet(path-aggregation neck) [36] and YOLOv3-head (anchor based) [21] as YOLOv4 architecture.

CSP is a new kind of backbone which can enhance the learning ability of CNN. The main technique is to divide the underlying feature mapping into two parts: one is through dense block and transition layer, the other is combined with transmission feature mapping to the next stage.

CSPResNetXt50 performs better than CSPDarkNet53 in classification, but worse in detection. It is suggested that the model with larger receptive field and larger parameters should be selected as the backbone. Therefore, by comparing CSPResNetXt50 [34], CSPDarkNet53 [34] and EfficientNet-B3 [37] through experiments, it shows that CSPDarkNet53 is more suitable as the backbone of detection model.

SPP [35] comes from Kaiming. He’s SPP Net, mainly because it significantly increases the receptive field, separates the most important context functions, and hardly reduces the network operation speed. And PANet [36] is mainly the improvement of feature fusion.

3.2. Selection of BoF and BoS

**Bag of Freebies:** apply some training techniques to improve the accuracy of the model without changing the complexity of the model.

**Bag of Specials:** the insertion module is used to enhance some attributes and significantly improve the accuracy of target detection.
Bag of Freebies (BoF) for backbone: CutMix and Mosaic data augmentation;

Bag of Specials (BoS) for backbone: Mish activation, Cross-stage partial connections (CSP), Multi-input weighted residual connections (MiWRC); Bag of Freebies (BoF) for backbone: CutMix and Mosaic data augmentation;

Bag of Freebies (BoF) for detector: CloU-loss, CmBN, DropBlock regularization, Mosaic data augmentation, Self-Adversarial Training, eliminate grid sensitivity, using multiple anchors for a single ground truth, Co-sine annealing scheduler, Optimal hyperparameters, Random training shapes;

Bag of Specials (BoS) for detector: Mish activation, SPP-block, SAM-block, PAN path-aggregation block, DIoU-NMS.

3.3. Additional Improvements

In order to make the designed detector more suitable for the training of single GPU, the following other designs and improvements have been made:

- Bag of Freebies (BoF) for backbone: CutMix and Mosaic data augmentation;
- A new method of data amplification Mosaic and self-confrontation training (SAT); Mosaic represents a new data enhancement method, which mixes four training images. So four different contexts are mixed, while CutMix only mixes two input images. This makes it possible to detect objects other than normal contexts. In addition, batch normalization calculates activation data from four different images on each layer. This significantly reduces the need for a large batch size. Self-advanced training (SAT) also represents a new data amplification technology, which operates in two forward operation stages. In the first stage, the neural network changes the original image rather than the weight of the network. In this way, the neural network attacks itself, changes the original image, and creates the deceptive illusion that there is no object in the image. In the second stage, the neural network trains and detects objects in the usual way.
- In the application of genetic algorithm, we choose the best super parameter;
- Some external methods have been modified, including SAM, PAN and CmBN. CmBN is a modified version of CBN [38], which only collects statistics between mini-batches in a single batch. Changing SAM [39] from spatial-wise attention to point-wise attention. Changing the shortcut connection of PAN to connection.

4. The Object Tracking Algorithm: DeepSORT

The previous SORT [29] algorithm used a simple Kalman filter to process the relevance of frame-by-frame data, and used the Hungarian algorithm to measure the relevance, so that it achieved better performance in high-frame real-time conditions. But the SORT algorithm ignores the surface features of the detected object, and it is more accurate only when the uncertainty of the object pose estimation is low.

References are cited in the text just by square brackets [1]. Two or more references at a time may be put in one set of brackets [3, 4]. The references are to be numbered in the order in which they are cited in the text and are to be listed at the end of the contribution under heading references, see our example below.

Therefore, in DeepSORT [31], such following improvements have been proposed:

- A new metric that combines motion and appearance information is used to replace the original correlation metric;
- A CNN network [40] is used to train and extract features on a large-scale pedestrian data set.

4.1. Track Handling and State Estimation

Initially, our tracking scenario is defined on the eight-dimensional state space \((u, v, \gamma, h, u, v, \gamma, h)\) that contains the bounding box centre position \((u, v)\), aspect ratio \(\gamma\), height \(h\), and their respective velocities in image coordinate system, and taking the boundary coordinates \((u, v, \gamma, h)\) as the direct observation value of the target state.
For the detected track \( k \), count the number of frames dependent on the last successful association. And judge whether the track has left the scene or just entered the scene.

4.2. Assignment Problem
The template is designed so that author affiliations are not repeated each time for multiple authors of the same affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization). This template was designed for two affiliations.

Represents the degree of motion matching between the \( j \)-th detection and the \( i \)-th track, where \( S_i \) is the covariance matrix of the observation space at the current moment predicted by the Kalman filter, and \( y_i \) is the prediction of the track at the current moment.

4.2.1. Motion Metric: Mahalanobis distance is used to evaluate the predicted Kalman state and the newly entered state:

\[
    d^{(1)}(i, j) = (d_j - y_i)^T S_i^{-1} (d_j - y_i)
\]

Represents the degree of motion matching between the \( j \)-th detection and the \( i \)-th track, where \( S_i \) is the covariance matrix of the observation space at the current moment predicted by the Kalman filter, and \( y_i \) is the prediction of the track at the current moment observation, and \( d_j \) is the state of the \( j \)-th detection \((u, v, \gamma, h)\).

Considering the continuity of motion, the detection can be filtered by the Mahalanobis distance. In the paper, the 0.95 quantile of the chi-square distribution is used as the threshold \( t^{(1)} = 0.4877 \), we can define a threshold function:

\[
    b_{i,j}^{(1)} = \mathbb{1}[d^{(1)}(i, j) \leq t^{(1)}]
\]

4.2.2. Appearance Metric: When the uncertainty of the target movement is low, the Mahalanobis distance is a good correlation metric. However, in practice, for example, when the camera moves, it will cause a large number of unmatched Mahalanobis distances, which will make this metric invalid. Therefore, we integrate the second metric for each bounding box detection frame \( d_j \); also, we calculate a surface feature descriptor \( r_j, |r_j| = 1 \), we will create a gallery to store the latest \( L_k = 100 \) descriptors of track, namely \( R_k = \{r_k^{(i)}\}_{i=1}^{L_k} \). And then we use the minimum cosine distance between the \( i \)-th track and the \( j \)-th track as the second measurement scale.

\[
    d^{(2)}(i, j) = \min \{1 - r_j^T r_k^{(i)} | r_k^{(i)} \in R_i\}
\]

Of course, we can use a threshold function to express:

\[
    b_{i,j}^{(2)} = \mathbb{1}[d^{(2)}(i, j) \leq t^{(2)}]
\]

Next, we merge these two scales into:

\[
    c_{i,j} = \lambda d^{(1)}(i, j) + (1 - \lambda) d^{(2)}(i, j)
\]

\[
    b_{i,j} = \prod_{m=1}^{2} b_{i,j}^{(m)}
\]

4.2.3. Matching Cascade: The paper also proposes a cascaded matching strategy to improving matching accuracy, mainly because when a target is occluded for a long time, the uncertainty of Kalman filtering
will greatly increase, causing the probability of continuous prediction to diffuse, with an assumption that the variance matrix is a normal distribution. If the continuous forecast is not updated, the variance of the normal distribution will become larger and larger. Then the points farther from the mean Euclidean distance may get the same Mahalanobis distance value as the points closer to the previous distribution.

5. Semi-automatic Calibration
Before testing in this experiment, we also need to perform perspective transformation on the scene coordinates under the camera’s perspective to make it a top view that can be easily observed and calculated.

The traditional calibration method needs to set the boundary coordinates before transformation in advance to calculate the transformation matrix $M$, and then perform coordinate transformation on specific coordinate points.

The general transformation formula is:

$$[x', y', w'] = [u, v, w] \cdot \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

where $(u, v)$ are the original image pixel coordinates, $(x = x'/w', y = y'/w')$ are the transformed image pixel coordinates.

However, this method has certain limitations. It can only be successfully applied in specific scenes, and cannot adapt to the differences brought about by scene changes. Therefore, the method adopted in this experiment is a similar automatic calibration step. Before each social distance detection, the boundary coordinates before transformation are selected in the order of upper left, lower left, lower right, and upper right through the computer interface, and different transformation matrices $M$ are calculated for specific scenarios, to improve the scope of application of this experiment.

6. Pedestrian Clustering
Our social testing of moving pedestrians is mainly to judge the distance between the two. Being too close will increase the risk of virus infection. However, there are always pedestrians who come out together. Peers, a group of people who know and walk together in public, will not increase the risk of infection among them. If such groups are regarded as dangerous people with a greater risk of infection, that will increase people's panic about the dangers of public places. Therefore, we do not want to judge people in the same group as dangerous people because of their closeness. And we hope to use the clustering algorithm to classify the peers into a certain group for analysis. Common clustering algorithms mainly refer to the distance between data points. In this experiment, the distance between data points is a direct parameter to judge the risk. So, we need a suitable clustering algorithm for judging pedestrian groups.

Consider below that the order of magnitude of pedestrians in the video will not be greater than $10^2$, and each pedestrian is used as a data point centroid. The data point is obtained after the object detection and calibration in the previous step. It has two-dimensional coordinates and changes with time, $[x, y]$. Because the coordinate position of pedestrians is constantly changing, if we want to discuss the changes of a pedestrian or two pedestrians with respect to time, we need to add an array called $memory$ to each object to store the coordinate information of the object in the $max_memory_size$ frame.

Next, we will define several advanced parameters:

- $f_t$: The frame speed vector, is a vector whose size is the change of pixels in a unit frame, and the direction is from the initial state to the final state. It abstractly described the movement of pedestrians;
- $d_{xy}$: The distance vector is also a vector, the size is the Euclidean distance between two points, and the direction is that one-point points to the other point. It describes the positional relationship between two points at a certain time;
- $d(x, y)$: Euclidean distance, is a scalar quantity that reflects the difference between data;
- $\theta(vector_i, vector_j)$: The vector angle is a scalar, which $\theta \in (0, 180)$.

And the calculation method is:

$$
\theta(f_{r_1}^{(1)}, f_{r_2}^{(2)}) = \begin{cases} 
|\text{angle}_1 - \text{angle}_2|, & \text{if } \text{angle}_1 \ast \text{angle}_2 \geq 0 \\
|\text{angle}_1| + |\text{angle}_2|, & \text{if } \text{angle}_1 \ast \text{angle}_2 < 0 \text{ and result} \leq 180 \\
360 - |\text{angle}_1| - |\text{angle}_2|, & \text{else situations}
\end{cases}
$$

Where $r = |a_1| + |a_2|$, $\text{angle}_1 = \arctan\left(\frac{f_{r_1}^{(1)} y}{f_{r_1}^{(1)} x}\right)$, $\text{angle}_2 = \arctan\left(\frac{f_{r_2}^{(2)} y}{f_{r_2}^{(2)} x}\right)$.

Based on the object parameters defined above, it can be concluded that pedestrians classified into the same group need to meet the following conditions:

- The distance between two pedestrians is less than the threshold, which is not only a prerequisite for peer judgment, but also a judgment condition for social distance detection.

$$
d(\text{centroid}^{(1)}, \text{centroid}^{(2)}) \leq \text{threshold}^{(1)}
$$

- In the max_memory_size frames’ time range, if the speed direction $f \vec{r}_v$ of the two pedestrians is basically the same, and the speed $|f \vec{r}_v|$ is also basically the same, it can indicate that the two pedestrians have the same state of motion.

$$
\theta(f_{r_1}^{(1)}, f_{r_2}^{(2)}) \leq \text{threshold}^{(2)}
$$

$$
d(|f_{r_1}^{(1)}|, |f_{r_2}^{(2)}|) \leq \text{threshold}^{(3)}
$$

- Under the premise of the previous condition, if the angle between the two pedestrians’ sum of speed direction $f \vec{r}_v$ and the distance vector $d \vec{v}$ is approximately equal to $90^\circ$. We think that the pedestrians always exercise side by side.

$$
\theta(f_{r_1}^{(1)} + f_{i_2}^{(2)}, d \vec{v}) \in [90 - \epsilon, 90 + \epsilon]
$$

where the $\epsilon$ is the bias. If there are two people lining up to follow, we tend to think that they belong to two groups of pedestrians.

If the above three conditions are all met in the judgment stage, there are reasons to believe that the social distance in these pedestrian groups is not related to the risk of virus infection.

### 7. Social Distance Monitorings

#### 7.1. Distance Measuring

Considering the pedestrian coordinate after object detection and calibration:

$$
\text{centroids}(\text{centroid}^{(1)}, \text{centroid}^{(2)}, \text{centroid}^{(3)}, ...)
$$

We use the coordinates of the bottom midpoint in the bounding box, $(\text{centroid}^{(i)}_x, \text{centroid}^{(i)}_y)$ to indicate the position of each pedestrian. In particular, if a group of pedestrians is clustered as a group of pedestrians by the pedestrian clustering algorithm, the position of this group is represented by the mean value of the coordinates of the midpoint of the bottom edges of
all pedestrians. \( \frac{1}{m} \sum_{i=1}^{m} centroid^{(i)}_x, \frac{1}{m} \sum_{i=1}^{m} centroid^{(i)}_y \), where \( m \) is the number of pedestrians in group.

Below, we will describe the algorithm flow used to calculate social distance.

7.1.1. Calculating Euclidean distance: Based on the pedestrian ID detected by DeepSORT, calculating the distance between any two pedestrian groups to construct the Euclidean distance matrix. The formula for calculating Euclidean distance is:

\[
d_{i,j} = \| centroid^{(i)} - centroid^{(j)} \| \tag{13}
\]

And the Euclidean distance matrix is:

\[
D = \begin{pmatrix}
d_{1,1} & d_{1,2} & \cdots & d_{1,N} \\
d_{2,1} & d_{2,2} & \cdots & d_{2,N} \\
\vdots & \vdots & \ddots & \vdots \\
d_{N,1} & d_{N,2} & \cdots & d_{N,N}
\end{pmatrix} \tag{14}
\]

Where \( N \) is the number of pedestrians detected in a certain frame.

7.1.2. Traversing the matrix: Traversing the matrix, if the value of an element is greater than threshold, \( Mindistance \), we put the serial numbers of the two into the violate set.

7.1.3. Marking pedestrians: Before the clustering, we will mark the pedestrians who are not in \( violate \) in green first, indicating that these pedestrians are currently at a safe distance.

7.1.4. Using Clustering: Then we use the three conditions in the Pedestrian Clustering method to determine whether there are pedestrians belonging to the same peer from the violate. If the conditions are met, it will be marked as purple and deleted from the violation.

7.1.5. Monitoring: Finally, the remaining pedestrians in the \( violate \) are marked in red, indicating that the detected social distance is relatively close and the risk of infection is greater.

7.2. Infection Risk Analysis

Finally, we will extract some data (\( v \) represents the total number of pedestrians in the scene, and \( m \) represents the number of peers identified through clustering) through experiments to analyze the relationship between social distancing and infection risk, that is, whether controlling social distancing can effectively reduce the population density in public places and thereby truly reducing the risk of viral infection. And based on the experiment results, recommendations and assessments of the risk of infection for the area will be made.

First of all, we need to use the social density \( \rho = N/m^2 \) and \( \overline{d} \) to quantify the social distance more intuitively. \( \overline{d} = \frac{1}{N} \sum_{i=1}^{N} d_{min}^{(i)} \) where \( d_{min}^{(i)} = \min_{j=1,N} (d_{i,j}) \)

In addition, we will define a metric describing the risk of infection. \( v \) represents the number of pedestrians in the detected violence set.

\[
v = \sum_{i=1}^{N} \sum_{j=1,j\neq i}^{N} \mathbb{I}(d_{i,j}) \tag{15}
\]

Where \( \mathbb{I}(d_{i,j}) = 1 \) if \( d_{i,j} < Mindistance \), else 0.
Then we consider the probability of infection risk as \( \eta \).

\[
\eta = \frac{v-m}{N} \times 100\%
\]  

(16)

\( \eta \) also, can be regarded as the proportion of non-accompanied persons who are closer to the group of people.

8. Experiment
Through the above statement, we will implement this social distance monitoring application (Fig.2.) based on YOLOv4 + DeepSORT and Pedestrian Clustering algorithm.

First, we select four boundary points srt_coor on Oxford Town Centre Dataset [41], the transform matrix M is obtained by perspective transformation, and the camera angle is transformed into top view.

Next, we extract the marked image data of the person category from the VOC2007 dataset [42], which contains 6376 training set pictures and 1993 test set pictures and is often used for image recognition and image detection model testing. We divide the training data into two categories, 0 (False person) and 1 (True person). And we use the PC that the configuration is Inter Core i7-7700HQ 2.8GHz Quad-core and 16GB NVIDIA GeForce GTX 1050Ti GPU trains the YOLOv3 and YOLOv4 object detectors for pedestrian detection respectively. By comparing the performance of the two for pedestrian detection, the YOLOv4 algorithm with better accuracy and speed is selected to detect pedestrians.

Then, it uses the detected data to run DeepSORT MOT, and marks each pedestrian with a specific ID. Further, the Euclidean distance matrix \( D \) is calculated, and the pedestrians who are too close to each other are put into the violate.

The pedestrian clustering algorithm is used to judge the peers, which is marked as purple, and the remaining pedestrians are marked as red.

Finally, the average minimum distance \( \bar{d} \), social density \( \rho \) and custom infection risk probability \( \eta \) in the scene are calculated. Then we draw graph as well as table and draw a conclusion.

| Table 1. Performance Comparison Of RCNN, SSD AND YOLOV3 [11]. |
|------------------|-------|-------|-------|-------|
| Model            | TT (in sec.) | NoI | mAP  | TL    | FPS  |
| Faster RCNN      | 9651  | 12135 | 0.969 | 0.02  | 3    |
| SSD              | 2124  | 1200  | 0.691 | 0.22  | 10   |
| YOLO v3          | 5659  | 7560  | 0.846 | 0.87  | 23   |

| Table 2. Performance Comparison of YOLOv3 and YOLOv4. |
|------------------|-------|-------|
| Model            | TT (in sec.) | mAP  | TL    |
| YOLO v3          | 90484 | 0.77  | 0.184 |
| YOLO v4          | 79038 | 0.82  | 0.324 |
9. Result and Discussion

9.1. YOLOv3 and YOLOv4 Comparison

Through training, Fig. 3 shows the loss of the YOLOv3 and YOLOv4 detectors in each iteration of the 8000 epochs training process.

![The Loss of YOLOv3](image1) ![The Loss of YOLOv4](image2)

**Figure 3.** The Loss of the YOLOv3 and YOLOv4.

Before analysing the performance of each detector, we used the research results of the Narinder Singh Punn team [11]. They compared the detection performance of RCNN, SSD and YOLOv3 in Table I. Thus, if we can compare the performance of YOLOv3 and YOLOv4, we could indirectly compare the advantages and disadvantages of various detection methods.

Table I shows the parameters at the end of each training, including TT (Training Time), mAP (mean Average Precision), and FPS (Frame Per Second). It can be seen by comparison that YOLOv3 better balances the relationship between mAP and FPS than RCNN and SSD, due to the definition of those two. In addition, it can be observed that the larger the mAP, the smaller the FPS of the model. Compared with YOLOv3, Table II shows YOLOv4 not only maintains the relative balance of the two, but also has a significant improvement in value. So, we finally used the detector trained by the YOLOv4 method to implement the social distance monitoring application.

9.2. Social Distance Monitoring

First of all, we observe the effect of pedestrian clustering. In the Fig. 4, we can see that those marked with purple are peers, those marked with red are those who are too close to each other, and the green arrow and yellow arrow are the speed direction of each pedestrian respectively. According to the calculated movement state of pedestrians, when the speed direction $\vec{f_r}$ and speed of pedestrians $|\vec{f_r}|$ are the same, and the distance vector $\vec{d}$ and speed direction $\vec{f_r}$ are approximately vertical, they can be regarded as peers.

The calibrated image is shown in the Fig. 5. After the pedestrian is detected by the trained YOLOv4 detector, each pedestrian is tracked by DeepSORT, and the ID is marked. The final result of social distance monitoring is shown in the Fig. 6.

In the lower part of the image, violates’ ID and number will be displayed in real time, as well as peers. We will also calculate the average minimum distance $\bar{d}$, social density $\rho$ and probability of infection $\eta$ in the monitoring process and display them.

![Pedestrian clustering](image3) ![Calibrated image](image4)

**Figure 4.** Pedestrian clustering. **Figure 5.** Calibrated image.
9.3. Infection Risk Analysis

Finally, the calculated data are denoised and smoothed, and then the average value of every 30 frames is taken as a unit of data to draw 2D Histogram of $d$ & $\eta$, $\rho$ & $\eta$, and $d$ & $\rho$ (Fig. 8). Fig. 7 combines the changes of the three with time, the relationship between them is analysed.

It can be found that there is a negative correlation between the average minimum distance and social density as a measure of the same attribute. However, $\eta$, which represents the infection risk index, increases with the increase of social density and decreases with the increase of the average minimum distance. The relationship among them also conforms to the law in real life.

Therefore, in order to reduce the risk of COVID-19 infection from the perspective of social distance monitoring, attention should be paid to limiting the minimum distance between pedestrians in public places and avoiding excessive pedestrian density in local areas. If found, corresponding actions should be taken in time.

![Figure 7](image)

**Figure 7.** By observing the changes of $d^-$, $\eta$, $\rho$ over time, we can find that probability of infection increases with the increase of social density and decreases with the increase of the average minimum distance.

![Figure 8](image)

(a) 2D_Histogram_dv&eta  (b) 2D_Histogram_dv&rho  (c) 2D_Histogram_eta&rho

**Figure 8.** 2D_Histogram of average minimum distance, probability of infection and social density. We can find out $d^-$ and $\eta$ has a negative correlation, $d^-$ and $\rho$ has a negative correlation, $\eta$ and $\rho$ has a positive correlation.
10. Conclusions
In this paper, we use YOLOv4 object detection method, DeepSORT object tracking method and the newly pro- posed pedestrian clustering algorithm to realize the application of social distance monitoring in specific real-time scenes. Then, according to the monitoring results, there is a negative correlation between the social density (or average minimum distance) and the risk of virus infection; and this relationship can be used to prevent the spread of virus in public places in advance.

In the experiment, we also compare the performance of YOLOv3 and YOLOv4, and find that YOLOv4 algorithm achieves better results in real-time social distance monitoring task. The new pedestrian clustering algorithm improves the rationality and integrity of social distance detection, making it more valuable in the epidemic environment.

As for the future work, the real-time data and hazard levels obtained from the monitoring can be used to guide pedestrians to choose a suitable route. The establishment of an interactive user- end platform is a feasible direction in the future.

References
[1] W.H.O. Naming the coronavirus disease (covid-19) and the virus that causes it. http://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it/, 2020.
[2] W.H.O. Archived: Who timeline - covid-19. https://www.who.int/news-room/detail/27-04-2020-who-timeline-covid-19/, 2020.
[3] W.H.O. Q a on coronaviruses (covid-19): what are the symptoms of covid-19? https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/q-a-coronaviruses#:~:text=symptoms/, 2020.
[4] W.H.O. Modes of transmission of virus causing COVID-19- 19: implications for ipc precaution recommendations. https://www.who.int/news-room/commentaries/detail/modes-of-transmission-of-virus-causing-covid-19-implications-for-ipc-precaution-recommendations/, 29 March 2020.
[5] W.H.O. Coronavirus disease (covid-19) advice for the public. https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public/, 4 June 2020.
[6] LCSW Amy Morin and MD Daniel B. Block. What is social distancing? https://www.verywellmind.com/what-is-social-distancing-4799570/, March 13, 2020.
[7] W.H.O. Covid-19: physical distancing. https://www.who.int/westernpacific/emergencies/covid-19/information/physical-distancing/, 2020
[8] Landing AI. Landing ai creates an ai tool to help customers monitor social distancing in the workplace. https://landing.ai/landing-ai-creates-an-ai-tool-to-help-customers-monitor-social-distancing-in-the-workplace/, Apr 16, 2020.
[9] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020.
[10] Narinder Singh Punn, Sanjay Kumar Sonihbadra, and Sonali Agarwal. Monitoring covid-19 social distancing with person detection and tracking via fine-tuned yolo v3 and deepsort techniques. arXiv preprint arXiv:2005.01385, 2020.
[11] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. IEEE trans- actions on neural networks and learning systems, 30(11): 3212– 3232, 2019.
[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Ma- lik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014.
[13] Ross Girshick. Fast r-cnn. In Proceedings of the IEEE interna- tional conference on computer vision, pages 1440–1448, 2015.
[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Im- agenet classification with deep
convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016.

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[18] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

[19] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7263–7271, 2017.

[20] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv: 1804.02767, 2018.

[21] Joseph Redmon. Darknet: Open source neural networks in c. http://pjreddie.com/darknet/, 2013.

[22] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In 2008 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2008.

[23] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In CVPR 2011, pages 1201–1208. IEEE, 2011.

[24] Matthew J Beal, Nebojsa Jojic, and Hagai Attias. A graphical model for audiovisual object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(7):828–836, 2003.

[25] Samuel S Blackman. Multiple hypothesis tracking for multiple target tracking. IEEE Aerospace and Electronic Systems Magazine, 19(1): 5–18, 2004.

[26] Thomas Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. IEEE journal of Oceanic Engineering, 8(3): 173–184, 1983.

[27] B Habtemariam, Ratnasingham Tharmarasa, T Thayaparan, Mahendra Mallick, and Thia Kirubarajan. A multiple-detection joint probabilistic data association filter. IEEE Journal of Selected Topics in Signal Processing, 7(3):461–471, 2013.

[28] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In 2016 IEEE International Conference on Image Processing (ICIP), pages 3464–3468. IEEE, 2016.

[29] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. IEEE transactions on pattern analysis and machine intelligence, 36(8):1532–1545, 2014.

[30] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In 2017 IEEE international conference on image processing (ICIP), pages 3645–3649. IEEE, 2017.

[31] Basile Roth. A social distancing detector using a tensorflow object detection model, python and opencv. https://towardsdatascience.com/a-social-distancing-detector-using-a-tensorflow-low-object-detection-model-python-and-opencv-4450a431238/, 17 June 2020.

[32] Dongfang Yang, Ekim Yurtsever, Vishnu Renganathan, Keith A Redmill, and U Özgüner. A vision-based social distancing and critical density detection system for covid-19. Image video Process. DOI, 2020.

[33] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping- Yang Chen, Jun-Wei Hsieh, and
I-Hau Yeh. 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, pages 390–391, 2020.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9): 1904–1916, 2015.

[35] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 8759–8768, 2018.

[36] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946, 2019.

[37] Zhuliang Yao, Yue Cao, Shuxin Zheng, Gao Huang, and Stephen Lin. Cross-iteration batch normalization. arXiv preprint arXiv:2002.05712, 2020.

[38] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV), pages 3–19, 2018.

[39] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. arXiv preprint arXiv:1605.07146, 2016.

[40] Ben Benfold and Ian Reid. Stable multi-target tracking in real-time surveillance video. In CVPR 2011, pages 3457–3464. IEEE, 2011.

[41] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge 2007 (voc2007) results. 2007.