PEDESTRIANS TRACKING ALGORITHM BASED ON YOLOV5 AND DEEPSORT

Yicheng Fan
School of Computer Science and Engineering@UNSW Sydney
z5375674@ad.unsw.edu.au

Yuchen Zhang
School of Computer Science and Engineering@UNSW Sydney
line 5: z5111402@ad.unsw.edu.au

Zhenxuan Liang
School of Computer Science and Engineering@UNSW Sydney
z5386953@ad.unsw.edu.au

Xiangyang Xu
School of Computer Science and Engineering@UNSW Sydney
z5387048@ad.unsw.edu.au

Yanchang Lu
School of Computer Science and Engineering@UNSW Sydney
z5327448@ad.unsw.edu.au

***Abstract*— Pedestrian tracking and detection has been widely used in many scenes, such as autonomous driving, security protection. Its difficulty lies in target detection and multi-target tracking. In this project, the method we used is Yolov5+DeepSort, by using yolov5 for target detection and DeepSort for multi-target tracking. We validate our model on the STEP dataset, the results show that our model has a good performance in pedestrian detection and tracking.**

Keywords*— *Pedestrian tracking, YOLOv5, DeepSort

INTRODUCTION

Pedestrian detection is a research hotspot in the field of computer vision in recent years, and it is also a difficulty in the field of target detection. Its purpose is to recognize and locate pedestrians in images, which is widely used in many fields. Such as traffic safety, driverless cars avoid traffic accidents by detecting pedestrians in advance and avoiding them in time; Also in security protection, pedestrian detection is used to prevent suspicious persons from entering; In terms of public place management, pedestrian detection and statistics are used to optimize the allocation of human and material resources.

Our project can divide into 3 parts, track pedestrians, count the number of pedestrians over time, analyses the behavior of pedestrians.

The dataset we used to train our model is download from CSDN which is a world-renowned Chinese IT technology exchange platform. For this dataset, it contains 803 images, each with 900*600 pixels. These images contain multiple pedestrian targets. Since this dataset does not contain video, we used the video in STEP dataset to test.

LITERATURE REVIEW

The main difficulties of pedestrian detection and tracking are target recognition and multi-target tracking. When a pedestrian appears, we need to automatically track it. When it disappears, our detection of it should also end at the same time. In this process, we need to realize dynamic tracking, and multi-target tracking and solve the difficulties of how to continue tracking when pedestrians block each other.

A. Target detection

For target detection methods, since Ross Girshick proposed r-cnn in 2013, people have proposed fast r-cnn, faster r-cnn, mask r-cnn, SSD, Yolo and other algorithms in just a few years. Among them, the two-step target detection method (r-cnn series algorithm) needs to generate many candidate frames first, and then use convolutional neural network to classify and regression the candidate frames; The single-step detection method (SSD, Yolo series algorithm) directly uses the regression method in the convolutional neural network to predict the location and category of the target in one step. Although the target detection method of two-step detection has higher accuracy in most scenes, it needs to be carried out in two steps. Therefore, this method will cost a lot of time and expensive hardware cost and is not suitable for real-time detection of video files. The network speed of Yolo series is faster, which can adapt to the detection of real-time video and has stronger generalization ability.

Yolov5 is a very suitable tool for target detection. Its network structure can be divided into input, backbone, neck and prediction. Compared with V3, V5 adds Focus structure and CSP structure in backbone, fpn+pan structure in neck, and GIOU_Loss in prediction, these improvements make V5 have high accuracy and detection speed while maintaining the lightweight of the model compared with V3 and V4.

B. Multi-target tracking

At present, with the application of neural network in target detection, the problem of target tracking is mostly realized by detection, that is, tracking by detection. Sort algorithm is a paper on multi object tracking (MOT) presented by Alex Bewley of Queensland University of technology at the ICIP conference in 2016.

SORT is a multi-target tracking algorithm, which can effectively associate targets and improve the real-time performance of tracking. Its core is the combination of Kalman filter and Hungarian algorithm, which can achieve better tracking effect. It uses the Kalman filter algorithm to predict the state of the detection box in the next frame and matches the state with the detection result in the next frame to realize vehicle tracking. But in this way, once the object is occluded or not detected for other reasons, the state information predicted by Kalman filter will not match the detection result, and the tracking segment will end in advance. After the occlusion is over, the vehicle detection may be

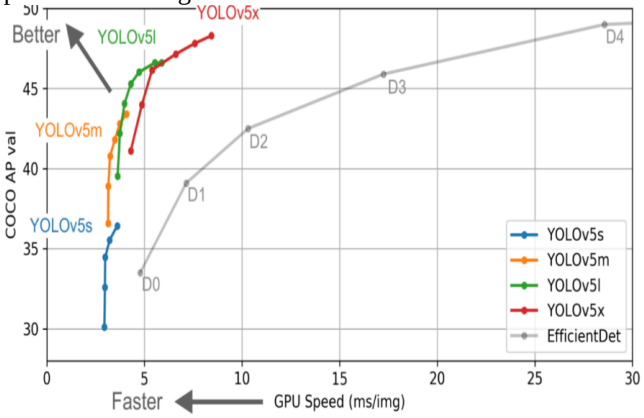
restarted, so SORT can only assign a new ID number to the object, representing the beginning of a new tracking segment.

To solve this problem, DeepSort algorithm is proposed, it uses the appearance features of the previously detected object. When the object receives occlusion and the occlusion ends, use the previously saved appearance features to match the object, assign ID numbers, and reduce ID switching.

METHOD

YOLOV5

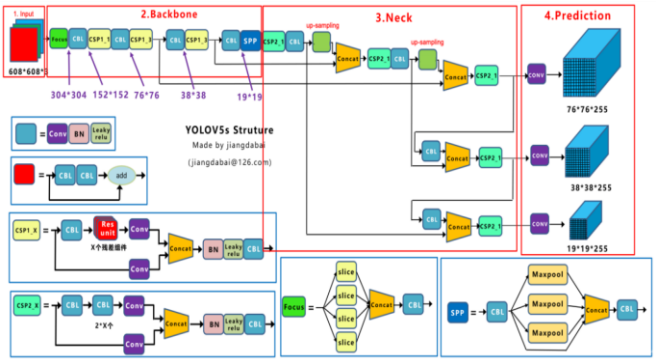
Yolov5 have four versions: yolov5s, yolov5m, yolov5l and yolov5x. Yolov5s network is the network with the smallest depth and the smallest width of feature map in yolov5 series. This is the main reason why we use it. The latter three are all deepened and widened on this basis, so in this project, we use yolov5s. The following[1] is the performance diagram of the four models.



[1]

Yolov3's network structure is divided into four parts: Input, backbone,Neck and Prediction. Yolov4 has made many innovations based on yolov3. For example, Input adopts Mosaic data enhancement, the Backbone adopts CSPDarknet53, Mish activation function, Dropblock and other methods. Neck adopts SPP+FPN+PAN structure, and Output adopts CIOU_ Loss, DIOU_ NMS operation. The structure of yolov5 is very similar to yolov4.

The following picture [2] is an example of the process of a yolov5 structure. After we enter a dataset. In the input part, the algorithm performs data enhancement on the image, initializes anchor boxes, and scales the image. Finally get a 608*608*3 image matrix. Then, in Backbone, the input is slicing into 304×304×12 feature map. After a convolution operation with 32 kernels, it finally becomes 304×304×32 feature map. This feature map will be input into the CSPDarkNet. Then enter the Neck structure. Drawing on the idea of residual network, the improved network faces the problem of network degradation due to the increase of depth. The feature map from the shallow layer or the previous level is conducive to positioning, and the deep feature map contains high-level semantic information. Finally in the prediction stage. yolov5 uses CIOU_Loss as the loss function of the Bounding box, so as to realize the back-propagation training model.



[2]

Yolov5s' four structures. Input uses Mosaic data enhancement, Adaptive anchor box calculation and adaptive image scaling. Backbone use Focus structure and CSPstructure. Neck use SPP+FPN+PAN structure. Prediction use GIOU_Loss.

Mosaic data enhancement

This step is mainly to add noise to the data set, so as to prevent overfitting in the training.

Adaptive anchor box calculation and Adaptive image scaling

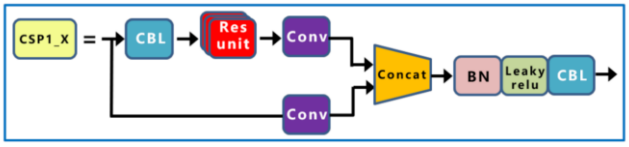
Adaptive anchor box calculation can calculate the best anchor box [3]. Then input it into image scaling. This method can uniformly scale the input pictures with different lengths and widths to the standard size. The improvement of yolov5 is that the black edges at both ends of the processed image become less, and the amount of calculation will also be reduced during reasoning, that is, the speed of target detection will be improved.

```
anchors:
  - [116,90, 156,198, 373,326]  # P5/32
  - [30,61, 62,45, 59,119]  # P4/16
  - [10,13, 16,30, 33,23]  # P3/8
```

[3]

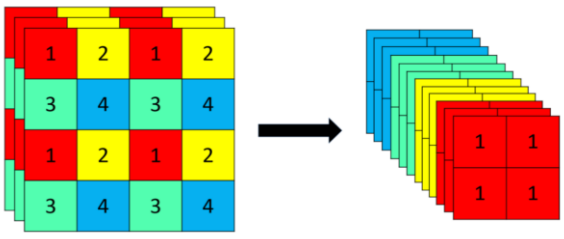
Backbone

Drawing on the idea of CSPNet, we have CSPDarkNet [4].



[4]

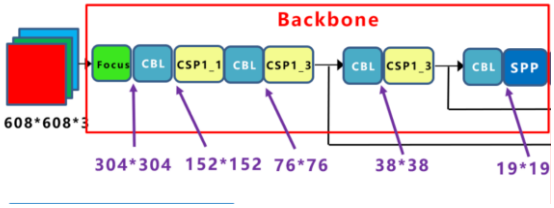
Focus structure:



[5]

The key of Focus structure comparison is slicing operation. For example, in the above figure [5], 4×4×3 becomes 2×2×12.

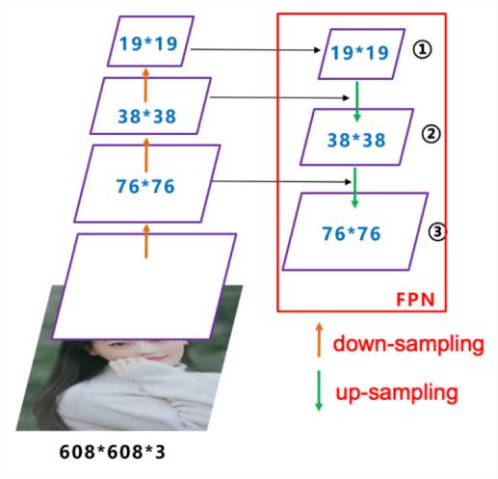
CSP structure: yolov5 designs two CSP structures [6], taking yolov5s network as an example. CSP1 structure is applied to backbone network, another CSP2 structure is applied to NECK.



[6]

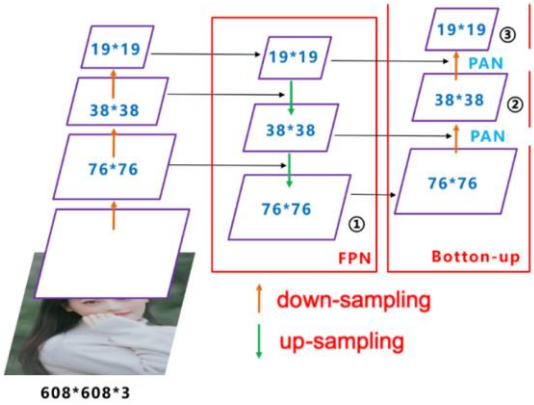
Neck: The current Neck in yolov5, like that in yolov4, adopts SPP+FPN+PAN structure.
SPP: Drawing on the idea of residual network, the improved network faces the problem of network degradation due to the increase of depth and alleviates the disappearance of gradient.

FPN: A feature pyramid model that combines multi-level features to solve multi-scale problems. The feature map from the shallow layer or the previous level is conducive to positioning, and the deep feature map contains high-level semantic information. As shown in the figure [7], FPN is top-down, and the high-level feature information is transferred and fused by up-sampling to obtain the feature map for prediction.



[7]

PAN: In order to solve the effect of small targets, the FPN network is supplemented (the supplementary point is mainly for positioning information), FPN belongs to down-sampling, and PAN belongs to up-sampling. FPN+PAN learns from It is the PANet of CVPR in 18 years, which was mainly used in the field of image segmentation at that time, but the ability of feature extraction can be further improved in yolov5.



[8]

Prediction

Finally, CIOU_Loss is adopted in yolov5 to perform the loss function of the bounding box.

$$CIOU\_Loss=1-CIOU=1-\left(IOU-\frac{Distance\_2^2}{Distance\_C^2}-\frac{v^2}{(1-IOU)+v}\right)$$

Where v is a parameter that measures the consistency of aspect ratio, which we can also define as:

$$v=\frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}}-\arctan\frac{w^{p}}{h^{p}}\right)^2$$

DeepSORT:

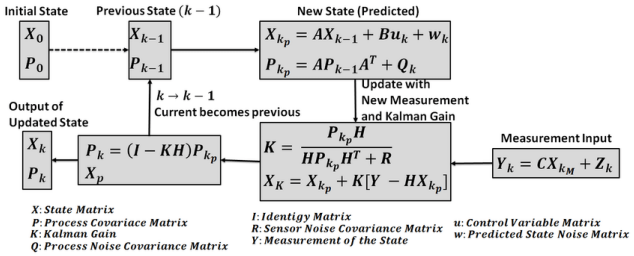
Our project uses DeepSORT algorithm as a pedestrian tracker. After using YOLOv5 to get different pedestrian detections in each frame, it is necessary to track the same target in consecutive frames and generate trajectories in turn. In this paper, the Kalman filter is used to predict the position of the target in the next frame, and then use the Hungarian algorithm to correlate the data, the workflow is as follows:

1. Get Bounding Box from YOLO detector and generate detections.
2. Kalman filter prediction.
3. Use Hungarian algorithm to match the predicted track with the detection result in the current frame.
4. Kalman filter update.

Kalman filter: The advantage of Kalman filtering is that the model can be applied to any dynamic system with uncertain information, to make educated predictions about where the system will go next, and to always indicate the real situation even with noise disturbances. As shown in Kalman filter image below, when using Kalman filtering for state prediction, set the previous state as $X_{k-1} = (P_{k-1}, V_{k-1})$, where p and v represent the position and velocity respectively, then when the current state prediction is performed under the Gaussian distribution, two quantities need to be introduced: the mean: $X_{k-1,}$ and covariance matrix: P_{k-1} .Considering the problems of external control and external noise interference, the state prediction equation of Kalman filtering can be obtained according to the kinematics formula and related mathematical calculation, as shown in the following formula:

$$\begin{cases}\hat{x}_k = \boldsymbol{F}_k \cdot \hat{x}_{k-1} + \boldsymbol{B}_k \boldsymbol{u}_k \\ \boldsymbol{P}_k = \boldsymbol{F}_k \cdot \boldsymbol{P}_{k-1} \cdot \boldsymbol{F}_k^{\mathrm{T}} + \boldsymbol{Q}_k\end{cases}$$

In this equation, F_k is the kinematic coefficient matrix; B_k is the external control matrix; uk is the external control quantity; Qk is the covariance matrix of external noise. Equation (4) shows that the current new optimal estimate is obtained from the previous optimal estimate plus known external control variables, and the new uncertainty is obtained from the previous uncertainty plus external environmental disturbances.



[9]

In order to let the Kalman filter model work iteratively, some parameters in the model need to be updated to ensure that the tracking is in real-time and accurate. Specifically, let the detection output of YOLOv5 be Z_k, and its covariance matrix be R_k. Combining the previous predictions, we have two Gaussian distributions: the prediction part is: $(\mu_0, \Sigma_0) = (\boldsymbol{H}_k\hat{x}_k, \boldsymbol{H}_k\boldsymbol{P}_k\boldsymbol{H}_k^{\mathrm{T}})$, where H_k is the covariance matrix of the Gaussian distribution of the prediction part; the detection part is: $(\mu_1, \Sigma_1) = (\boldsymbol{Z}_k, \boldsymbol{R}_k)$, Through a series of mathematical operations, the state update equation can be obtained as:

$$\begin{cases} \hat{x}'_k = \hat{x}_k + K \cdot (\boldsymbol{Z}_k - \boldsymbol{H}_k\hat{x}_k) \\ \boldsymbol{P}'_k = \boldsymbol{P}_k - \boldsymbol{K}\boldsymbol{H}_k\boldsymbol{P}_k \\ \boldsymbol{K} = \boldsymbol{P}_k\boldsymbol{H}_k^{\mathrm{T}} \cdot (\boldsymbol{H}_k\boldsymbol{P}_k\boldsymbol{H}_k^{\mathrm{T}} + \boldsymbol{R}_k)^{-1} \end{cases}$$

In the equation, K represents the Kalman gain. It can be seen that X'_k is the new optimal estimate, this estimate and P'_k can be put into the next prediction and update equation to iterate continuously. The Hungarian algorithm is a combinatorial optimization algorithm for polynomial time task allocation problems. The mathematical description is as follows: Let the matrix R_{ij} represent the efficiency of the M_i person doing the J job. Now it is necessary to select n independent elements from the matrix, that is, the row and column coordinates of the selected elements are not repeated, and finally the sum of these elements is maximized. Make $r = \max(r_{ij})$, $x_{ij} = r - r_{ij}$ Then the problem is equivalent to selecting n independent elements in the matrix A composed of X_{ij} to make their sum minimum. The specific steps of the algorithm are as follows:

1. Subtract the smallest element of A from each element in matrix A to obtain a non-negative matrix A1 with at least one zero element (reduce for each row or column go to the smallest element of this row and this column).
2. Find the smallest set S1 of lines, the number of lines m1, containing all non-zero elements in A1. If m1 = n, then there are n independent zero elements, and these n positions in A constitute the solution.
3. If m1 < n, then let h1 denote the smallest element in A1 that is not in S1, and h1 is greater than 0 at this time. For each line in S1, add h1 to each element covered by these lines; subtract h1 from each element in the remaining A1 to get a new matrix A2.
4. Repeat steps 2 and 3, replacing A1 with A2. Every time after the operation of step 3, the sum of all elements of the matrix will be reduced by n $(n\text{-}n_k) * h_k$, the solution can be found after iterations.

The algorithm sequence for pedestrian tracking in this project is as follows: First, YOLOv5 detects multiple pedestrians M and their bounding boxes and coordinates in the current frame, and then uses Kalman filtering to estimate the pedestrian position of the current frame based on the tracking results of the previous frame to obtain N prediction results and N trajectories. Finally, the Hungarian algorithm is used to match the detection results with the prediction results, and the data association is completed in the images of multiple frames to form the tracking trajectory.

RESULT

Different results can be obtained by using different weight files and modifying the parameters of the tracker code, which were evaluated by our group.

When using weight files for training, we found that different weight files have different effects on pedestrian detection, and the yolov5s.pt file that comes with yolov5 is a training weight that includes 80 different categories of objects, including Pedestrian classification. Another weight file is best. pt is the result of training on the dataset of our group.

Figure 1 Output using best.pt

Figure 2 Output using yolov5s.pt

The total number of pedestrians detected by the two weight files is shown below:

	Current frame	**Tracking total**
Yolov5s.pt	11 people	12 people
Best.pt	20 people	21 people

Table 1 Pedestrians detection comparison

It is obvious that the weight file generated by a specific dataset was much better than the given weight file from the yolov5 source code.

For a more in-depth study of the accuracy of this model, we also record the pedestrian tracking states before, during, and after occlusion respectively.

The detect comparison is shown below.

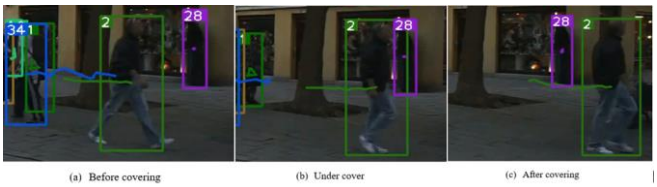


Figure 3 Pedestrians detect states

In the YOLOv5-DeepSORT model, although there is pedestrian occlusion, the target still maintains its original ID after being occluded, which makes the pedestrian track more accurate and the target trajectory more reliable and robust. The next optimization process is to adjust the confidence threshold, we set the confidence threshold to be from 0.2 to 0.7 with an addition of 0.1 each time and observe the change of bounding box ID. Below shows the result of different confidence thresholds.

Figure 4 Detection of different confidence thresholds

As the value of the confidence threshold increases, models are starting to look more for pedestrians than stationary passersby. As can be seen from the above figure, when the threshold is low, the detected persons No. 9 and 39 are not walking, and when the conf threshold is too high, the distant pedestrians will also become difficult to be detected (Fig 5).

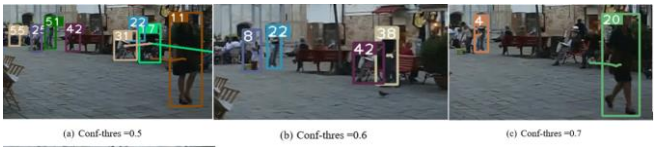


Figure 5 distant pedestrians detect

Finally, we collected the total number of pedestrians detected under each confidence threshold as shown in the table below. Although the number of detected pedestrians is decreasing, we believe that the optimized number is reasonable and necessary.

Conf-thres	Tracking total
0.2	52 people
0.3	50 people
0.4	46 people
0.5	39 people
0.6	33 people
0.7	23 people

Table 2 Total pedestrians detected with different conf-thres

Therefore, we chose the confidence threshold of 0.6. After optimization, all detected pedestrians are bounded by bounding boxes of different colours with the corresponding trajectories, while the text on the left side of the video shows the content that the project needs.

Figure 6 Text content shown on video

The first four lines of the text show the total number of pedestrians at this time and the total number of pedestrians so far in the video, when a pedestrian comes in or leaves from this camera, the tracker will also record the number as one. We recorded the number of pedestrians walking alone and in groups, and the tracker will also alert us when someone joined or leaves the group.

The last function is to show the total count of pedestrians who are within a manually created frame. The red frame in this picture was drawn manually and the tracker can accurately report that there are 5 people in this region.

Figure 7 Final output of the project

So far, we achieved all the functions required by the task by using yolov5 and deep sort techniques.

DISCUSSION

Our model reached better result by fine-tune YOLOv5-DeepSORT based on pre-trained model YOLOV5s and our own dataset, however, as Figure8 shows, our model is not performing well when facing more difficult scenario.

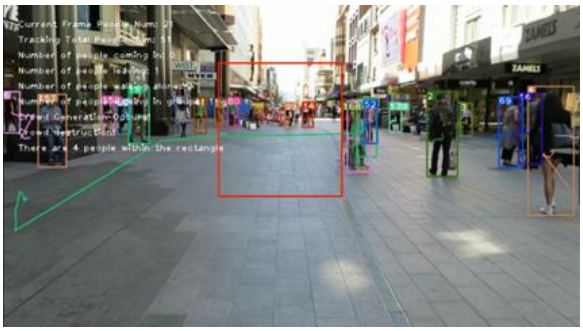


Figure 8 wrong trajectory on the second test video

Unlike the first test video, the second video was shot while moving, there is constant pedestrians entering and leaving the frame, and a large number of pedestrians covering each other. It is a hard task for our model, resulting wrong trajectory appears on some frame.

Another limitation is the number label of the pedestrians may change after they are covered up then appeared again. As Figure9 shows, the pedestrian with the label 228 was reassigned to 31 after the he was covered by the singing lady.

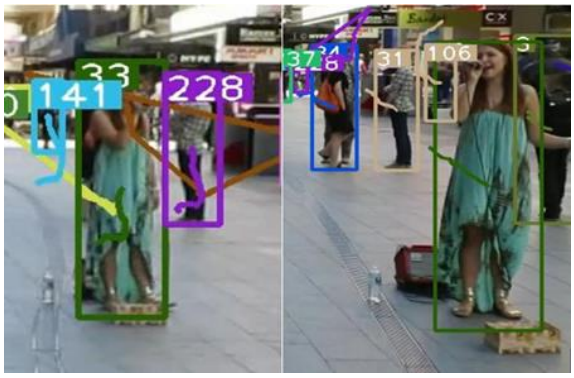


Figure 9 reassign label to the same pedestrian

In our opinion, two limitations above are caused by the tracking algorithm. After YOLOv5 assign ID to each pedestrian, the DeepSORT model extracts the pedestrian features detected in the current frame and compares them with the features of the previous frames to assign an ID to the pedestrian detected in the current frame. DeepSort will get erroneous results if the scene has too much interference. Hence the same pedestrian will be assigned a new ID after being masked.

Additionally, our model cannot identify pedestrians who are too small in the video (Figure10). Through fine-tune the model and add extra dataset, we have already improved the result. But it is not enough, one possible way to fix the problem is that improve the video quality of input. However, it will decrease the recognition time of each frame which is very importance in real-time tracking. A good model should have a high recognition rate even with low input quality, which is also our future goal.

Figure 10 small pedestrians cannot be recognized

We tried to work on the problem, but we find it difficult to satisfy all the scenario on a single model. Track real-time object is still a big challenge for us. Nevertheless, we will continue to work to achieve better results.

CONCLUSION

This report shows our work on the object tracking in real-time videos project. We achieved all tasks based on YOLOv5-DeepSORT. To achieve better result, we fine-tune YOLOv5-DeepSORT on our additional dataset. Experiment shows the improvement of our model compared with pre-train weight YOLOv5s. On the other hand, our model still cannot identify all pedestrians and cannot accurately track all trajectories. We are looking forward to exploring more popular approach in the object tracking area.

REFERENCES

[1] Ultralytics, “Ultralytics/yolov5: Yolov5 in PyTorch > ONNX > CoreML > TFLite,” GitHub. [Online]. Available: https://github.com/ultralytics/yolov5. [Accessed: 05-Aug-2022].

[2]-[8] D. Jiang, “A complete explanation of the core basic knowledge of Yolov5 of the Yolo series,” zhihu. [Online]. Available: https://zhuanlan.zhihu.com/p/172121380. [Accessed: 05-Aug-2022].

[9] A. Ghosh and About Abhishek GhoshAbhishek Ghosh is a Businessman, “Kalman filter to stabilize sensor readings,” The Customize Windows, 19-Mar-2019. [Online]. Available: https://thecustomizewindows.com/2019/03/kalman-filter-to-stabilize-sensor-readings/. [Accessed: 04-Aug-2022].