Develop and evaluate a method for tracking pedestrians and analyzing their motion in MOT16 dataset

<z5380108 Dian Jin>
<z5385997 Lei Chen>
<z5327713 Guoling Dong>
<z5291577 Yunyi Li>
<z5329125 Xu Guo>

Ⅰ. Introduction

Public security about pedestrians has drawn widespread attention by official and private departments. Countless researchers put their efforts on exploring this domain and obtained the expected result to improve the social security of pedestrians[1]. Meanwhile, video detection has played a crucial role in traffic safety management. Object detection, one of the majority and essential tasks of video detection, deploys the classification of objects and marks the current position of objects. There exists the urgent evacuation and flustered enormous flow of people in a sporadic emergency or violence, the problem is the fact that pedestrians expect to escape from possible exit in a flash unlike rigid objects which are unmovable[2]. The change of randomness in human motion, joint posture, clothing, lighting, mutual occlusion, complex backgrounds causes the extremely tough judgment of their position detection[3]. Fortunately, the idea of a data-driven method was introduced [4], and made a remarkable breakthrough in related algorithms for monitoring and judgment of escape in the panic crowd.

Computer vision can be defined as extracting high-level information from images and video. Nowadays it is in the process of changing many industries. Computer vision systems consist of three tasks: image transformation, image analysis, and image understanding. On the basis of the computer vision system, the object tracking task is realized.

Figure 1: Static view of street corner

The video frame of the dataset of this task is composed of static shots and moving shots on the street. We will also encounter the challenge of shape change and contour change caused by the change of body posture, background clutter interference, identification of crowd, pedestrian trajectory crossing and real-time monitoring of pedestrian data statistics. These challenges can lead to identification difficulties, for example when someone walking in a group may be obscured by a partner and may be difficult to identify. In addition, object detection and tracking helps to monitor every pedestrian in a certain state of motion. At present, there are a variety of pedestrian detection and tracking algorithms. This project will analyze the existing mainstream methods from traditional manual feature methods to deep learning methods to achieve an efficient system for high-precision pedestrian detection and tracking on the given dataset.

Pedestrian detection is the study of predicting pedestrians by recognizing the features of images, and pedestrian tracking is the study of monitoring the movement trajectory of pedestrians and recording the path. The combination of pedestrian detection and tracking system enables it to sense, analyze and detect various posture of pedestrians. After training on the dataset, the system can overcome the challenges of object form group and background

interference, and has excellent performance in recording pedestrian tracks and data statistics.

Ⅱ. Literature Review

1. Object Detection

Object detection is the study of predicting pedestrians by recognizing the features of images, and pedestrian tracking is the study of monitoring the movement trajectory of pedestrians and recording the path. The combination of the pedestrian detection and tracking system enables to sense, analyze and detect various posture of pedestrians. Various methods used for pedestrian detection in this project are: Background Subtraction, MOT and YOLO. The best detection method is selected by combining tracking algorithms such as SORT, and an efficient pedestrian tracking algorithm is constructed. The feature of the program is that it can calculate the data of pedestrian detection in every frame of image, and track the movement of pedestrians. After training on the dataset, the system can overcome the challenges of objects from group and background interference, and has excellent performance in recording pedestrian tracks and data statistics.

2. Pedestrian Detection

Pedestrian detection methods including handcrafted, deep learning, and hybrid methods, usually consist of three consecutive steps: proposal generation, classification (and regression), and post-processing. Note that not all methods have these three steps. Without loss of generality, we will discuss these three steps in detail. [5]

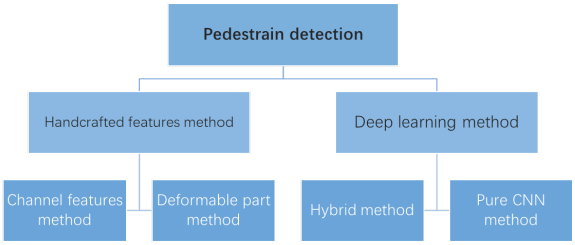


Figure 2: Two different classes of single-spectral pedestrian detection approaches

3. Background Subtraction

Background subtraction (BSM) is one of the most popular methods for object detection. This algorithm is usually used to divide the scene into background and foreground, compare the foreground object with the frame without the object to find the moving part in the video, and create a distance matrix. In theory, all it does is to compare the difference between the values of two frames in a video to a threshold. If the difference between the values of two frames is greater than a preset threshold (usually predefined using the first few frames of the video), the result is marked as the detection of a moving object [8].

4. Histogram of oriented gradients (HOG)

HOG is one of the most widely used handcrafted feature methods in pedestrian detection. HOG descriptors significantly outperform existing feature sets for human detection [6]. HOG feature describes the appearance of gradient directions in a local image focused on the structure or shape of the image object. It provides edge orientation by dividing the image into small local regions and calculating gradient and orientation. And generates histograms for each region.

5. Multiple object tracking (MOT)

MOT is an important research topic in computer vision which has a wide range of applications [7]. The goal is to estimate the movement of the object shown in the video. Formulating MOT as multi-task learning for object detection and re-identification in a single network, which can realize the joint optimization of the two tasks and has high computational efficiency.

6. FairMot

FairMOT is based on the anchor-free object detection architecture. It is not a naive combination of CenterNet and re-ID, while the re-ID and detection are treated equally in FairMOT, which has a simple network structure for extracting re-ID features and detecting objects. FairMOT addresses the fairness issue between re-ID and detection and achieves a high level of detection and tracking accuracy.

7. YOLOv5

The You Only Look Once (YOLO) algorithm is famous for its object detection characteristic and is a widely used algorithm. Redmon et al. launched the first YOLO version in 2015 [10]. In the past years, Subsequent versions have been published by scholars, including YOLOv1 to YOLOv5. In this

experiment, YOLOv5 is applied which was introduced in 2020.
Compared with other versions of YOLO, the advantage of the YOLOv5 is flexible control of model size, application of Hardswish activation function, and data enhancement [11]. YOLO V5 provides each batch of training data through the data loader, while enhancing the training data. The data loader performs three types of data enhancement: scaling, color space adjustment, and Mosaic enhancement. YOLO V5 multiple network architectures are more flexible to use, have a very lightweight model size, and excel in accuracy. But there is also the possibility that small objects may not be detected as accurately as large ones.

Ⅲ. Method

A. Object Detection using You Only Look Once

Essentially, Yolov5 is based on CNN neural network architecture. The model only predicts 1 class as per the latest update that uses the pedestrian class for implementing many tasks since we have trained the dataset before. We deploy pre-trained weights and configurations in the model. In this case, the implementation of the framework and its corresponding output vector will be introduced.
To be noticed, we didn't deploy deepsort in this experiment.

1) *The prediction Vector*
Initially, S x S grid of cells is delivered by dividing the object gathered by detection in which the cell named 'responsible cell' gives the centroid of the object describing its location. The prediction of B bounding boxes and C class probabilities are computed. The majority of components involved in the bounding box are known as (x, y, w, h, confidence). (x, y) coordinates are the center of the box and it is then normalized in the range of [0,1] with (w, h) coordinates relative to the image scale. The parameter confidence is introduced as:

$$\Pr(\text{Object}) * \text{IOU}(\text{pred}, \text{truth})$$

The confidence score is 0 if there is no object in the corresponding grid. Otherwise the confidence score is produced by intersection over union (IOU) employed logistic regression method to compute the bounding box detection.

The conditional probabilities are then computed in which the computation delivers a total of S x S x C class probabilities. Hence, the output vector will produce S x S x (B*5+C) tensor.

Here, the prediction has also finished through 3 various scales and then plenty of labels based on the confidence matrix delivered to the computation of bounding box classification.

2) Network
YOLOv5 network structure consists of four parts: input, backbone, neck and prediction. The network structure is shown in Figure 3. According to different widths and depths, this network model is divided into YOLOv5s, YOLOv5m, YOLOv5m and YOLOv5x. In this experiment, the network of YOLOv5s used in this Yolov5 method. Its characteristic is that it runs fast, but the accuracy rate is lower than that of the other three networks [9].

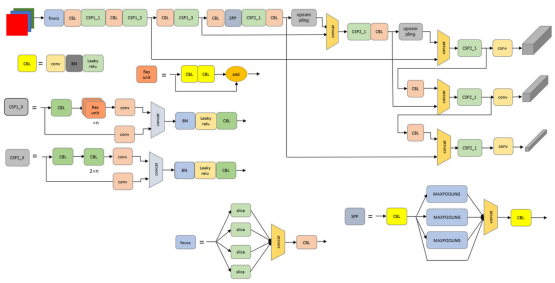


Figure 3: Structure of YOLOv5 network [9]

B. FairMOT

FairMOT is built on top of CenterNet. Unlike other frameworks, FairMOT, it is a one shot tracker and treats the detection and re-ID tasks equally. It has a simple network structure which consists of two homogeneous branches for detecting objects and extracting re-ID features, respectively. For Task 1 and Task 2 in this project, we can use FairMOT tracker to produce the predicted image with bounding box and ID in one step.

1) Backbone Network

The FairMOT uses ResNet-34 as backbone to get a balance between accuracy and speed. An enhanced version of Deep Layer Aggregation is applied to the backbone to fuse multi-layer features. This version of DLA has more skip connection between low-level and high-level features, and all convolution layers in up-sampling modules are replaced by deformable convolution to better alleviate the alignment issue[7].

2) Detection Branch

The detection branch is built on top of CenterNet. Three parallel heads are appended to DLA-34 to estimate heatmaps, object center offsets and bounding box sizes. Each head is implemented by applying a 3 × 3 convolution (with 256 channels) to the output features of DLA-34, followed by a 1 × 1 convolutional layer which generates the final targets. The heatmap head is responsible for estimating the locations of the object centers. The box offset head estimates a continuous offset relative to the object center for each pixel to localize objects more precisely. The box size head is responsible for estimating height and width of the target box at each location[7].

3) Re-ID Branch

Re-ID branch is to distinguish objects and give them a unique ID. It took a convolution layer with 128 kernels on top of backbone features to extract re-ID features for each location. The re-ID features are learned through a classification task, and all object instances of the same identity in the training set are treated as the same class.

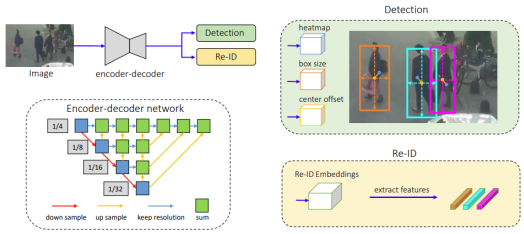


Figure 4: Overview of the structure of FairMOT,

C. Background Subtraction

The moving objects are highlighted by Background Subtraction algorithms through extracting the features of them by subtracting the background of two subsequent frames. The difference between two frames is determined by their absolute difference, then it can highlight objects in motion. Threshold is used to split the foreground and background while the image has been converted to grayscale. Morphological image processing plays a vital role in cleaning the noise, 'cv2.erosion' helps clean the noise. The whole objects on the cleaned foreground are obtained by searching connectivity which is delivered by deploying the centroids using 'cv2.connectedComponentwithStats' function. Upon obtaining the centroids, the bounding box can be drawn. This can help detect and track pedestrians efficiently.

IV. Experimental setup

The tracking component of this assignment was developed on the Anaconda platform under python 3.9 and OpenCV version 4.2.0 for object tracking techniques such as Background Subtraction, Yolov5 and FairMOT. There are several libraries employed in this assignment like numpy, torch and matplotlib. Thresholding and low pass filter build up the background subtraction algorithm. Yolov5 and FairMOT is an open source model in which Yolov5 is the pre-trained model that enables users to detect and track the objects and label them. This model is made up of TensorFlow and labels the whole objects detected by coco.txt and gt format labeled file. The FairMOT model trains the image contained in the dataset to deliver results.txt through the open source server in which results.txt labeled the objects.

Task1 Track Pedestrian

This task employs a python-based solution to detect and track pedestrians in the frames of video taking above methods. Moreover, this task obtains the trajectory for each pedestrian and draws the bounding box with a unique color corresponding to a typical person.

1) Detection of pedestrians on image

a) Yolov5

The split training set is trained through the open source of YOLOv5. For our detection, we set the default object confidence threshold = 0.25, default IOU threshold for NMS = 0.45, default image size 640, and hyperparameter epoch = 10. Best.pt is delivered after training the model.

Then, The YOLO uses the pre-trained weights 'Best.pt' and configuration 'person.yaml' to detect. Since the coordinator of the object plays a crucial role in the following task. Therefore 'detect.py' in source code is edited to produce all the pedestrians corresponding to its coordinator that prepare the bounding box drawn in the following part.

b) FairMOT

Basically, We pass the dataset to the fairMOT server, the dataset there will train the dataset and generate a result.txt, and then we extract the data stored in result.txt which contain the coordinator of the label and perform other stages like Yolov5.

c) Background Subtraction

The moving objects are highlighted by Background Subtraction algorithms through extracting the features of them by subtracting the background of two subsequent frames. Threshold = 70 (=20 in test set 01) is used to split the foreground and background while the image has been converted to grayscale in test set 07. After we gather the raw foreground, it contains plenty of noises since this video frame exists. Therefore, morphological image processing plays an vital role in cleaning the noise, 'cv2.erosion' with 5x5 kernel and 3 (1 in test set 01) times help clean the noise. We won't worry if it will clean or cause bias on objects we want to detect, since the main object will still be visible since pedestrians obtain a certain connected area while noise just occupies a piece of aea. The whole objects on the cleaned foreground are obtained by searching connectivity which is delivered by deploying the centroids using 'cv2.connectedComponentwithStats' function.

This is under testset07, it will be a different threshold in another testset.

2) Obtain the trajectory for each pedestrian & Draw the bounding box

a) Yolov5

The coordinator of the object will be delivered by the first task, the bounding box is generated by this. The key to obtain the trajectory for each pedestrian is to determine whether there is the same person in various frames. The principle of frame difference method employed in this task. Firstly, we observe the video frame to figure out there is no pedestrian walking at a high speed. Then, the distance of bounding boxes between two frames will not be so far. We set the distance threshold = 250 (the distance of the same person moving in two continuous frames won't exceed this threshold value). The program will determine there is the same person if the distance between the previous bounding box's centroid and the next one under this distance threshold (250). If so, this pedestrian will inherit the previous frame's attribute and label, that is, the same color and label will be displayed in the bounding box of the same person in various continuous frames. Therefore, the trajectory will be drawn on the pedestrians' centroid under the same person determined by this. Otherwise, it will generate a new color and label in a different person. Also, the trajectory will be interrupted and begin a new one from the new pedestrian's centroid.

b) Background Subtraction

Apart from the same tracking the trajectory method with Yolov5, there is one more preprocessing step for Background Subtraction to draw the bounding box, since there might still be some noise to be detected and saved, so we set the area threshold to make the too small area disappeared which less than threshold = 4500 (1600 in test set 01). That also means the farest pedestrians we can detect is at least 4500 in dimension.

c) FairMOT

We draw a trace using 'Plot Tracking' in which we create a TXT text for each video sequence, with each line representing a target and containing 9 values for segmentation. For the previous values, conf fills 1, which is the same as the GT meaning of the training set. The last three values represent 3D MOT. Since we evaluate 2D data sets, the last three values x,y, and z are filled with -1, and the corresponding frame,

ID, and Bbox coordinates all start with 1. Evaluate trace performance using the evaluation scripts provided with MOTchallenge.

Task2 Count Pedestrian

a) Yolov5 & Background Subtraction

Based on task 1, the program will generate a new label corresponding to a new pedestrian. The count of unique pedestrians increased by 1 under the process of assigning a new label. The total count of pedestrians can be delivered as the total number of bounding boxes. We allow users to draw rectangular regions within the video window employed by 'selectROI' (an opencv function). Users can draw and choose the area of the rectangle in the first frame. selectROI will generate the coordinator of this frame, then it can detect if someone has the coordinator of the corner go through this rectangle, the count will increase by 1. Then we can see how many pedestrians go through this area from the word shown on the left top sides by the 'putText' function.

b) FairMOT

SetMouseCallback () creates a mouse callback function in OpenCV. The mouse response function is called three times by clicking the left button, releasing the mouse, and moving the mouse (including clicking the mouse in situ without moving) Bind the callback function to the window.

Task3 Analyze Pedestrian

It can be seen from the video frame that people form in groups at a very close distance. Walk-in-group pedestrians are determined by the difference between the centroid of the previous and current frame is less than the distance threshold = 250. After execution, it will generate two lists of coordinates of formed group and non-formed group respectively. The number of walk-alone pedestrians are those objects in the list of non-formed groups, and the number of walk-in-groups pedestrians will be delivered the difference between the total pedestrians in the current frame and people walking alone. The bounding box of each group will be recorded as a minimum coordinator and the maximum length（min x, min y, max w, max h）through the traverse. The new formed group, also deployed in a frame difference method, is determined by the different areas of the previous group and this current group with a padding area threshold = 100.

The occurrence of the formed group will be highlighted as a new form label If the function doesn't detect the area difference from the previous frame. Otherwise, it won't be highlighted as it is an already existing group. Group destruction occurs when the ratio of length and width change a lot from the previous recorded group area. To deal with the occurrence of pedestrians entering or leaving, four certain-size rectangles have been drawn on all the sides of the frame. When pedestrians touch any one of those, it will be highlighted as entering or leaving.

Ⅴ. Result and Analysis

In this research, the team has implemented four various methods and YOLO performs the most efficiently. This conduction is based on the evaluation factors we analyze below (Table 1.).

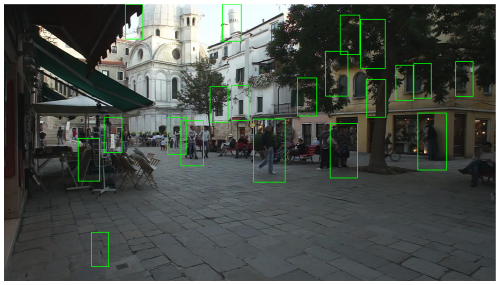
Evaluation Factors	***HOG+SVM***	***Background Subtraction***	***YOLO***	***FairMOT***
Detect Individual Pedestrians	***NO***	***YES***	***YES***	***YES***
Detect Pedestrian in a group together	***NO***	***YES***	***YES***	***YES***
Detect all the pedestrians in a frame	***NO***	***NO***	***YES***	***YES***
Detect group form and destruct	***NO***	***NO***	***YES***	***YES***

Table 1. Comparison of detection method

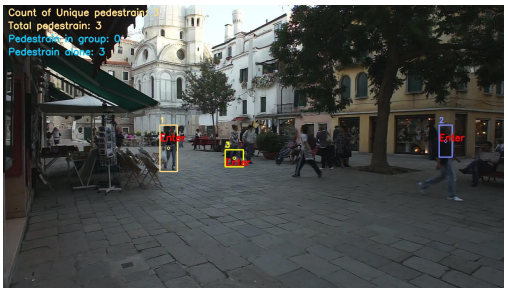
As a result, Yolov5 and FairMOT have proven the best performance which satisfies all the requirements and evaluation factors. We decided to proceed with Yolov5 to finish all the tasks because FairMOT requires a huge memory and configuration to proceed.

After that, we generate the quantitative result which is accuracy, mean average precision and other metrics.

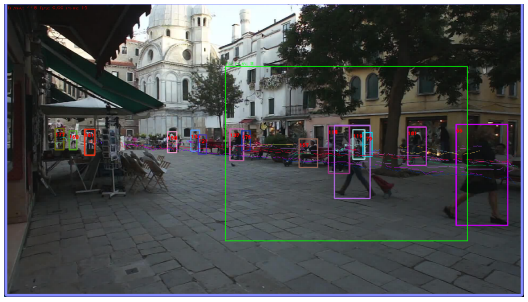
The qualitative result of accuracy of counting pedestrians and building their centroid are shown below.



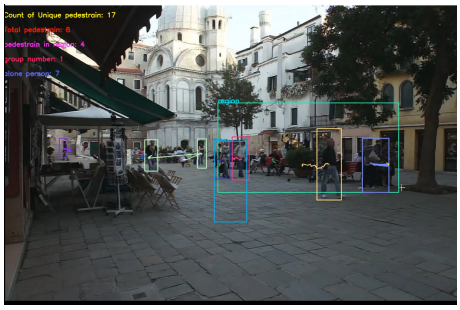
SVM+HOG



Background Subtraction

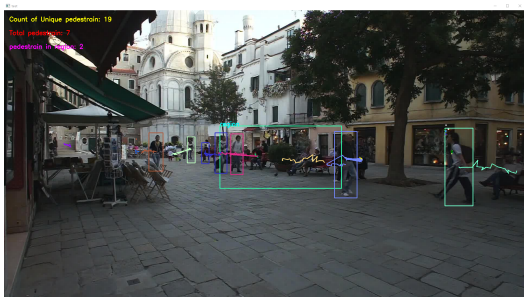


FairMOT

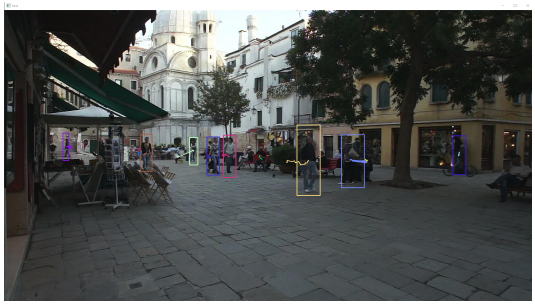


Yolov5

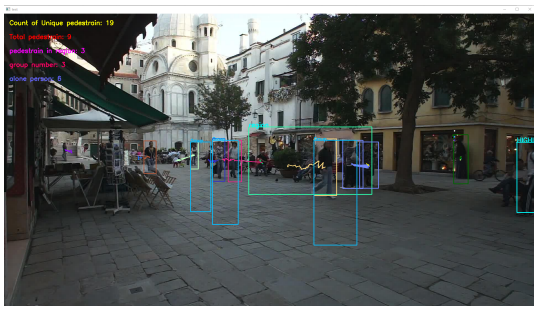
Fig 5 Accuracy of 4 methods

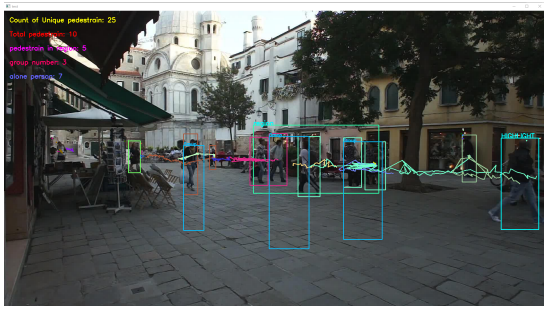


task 1, detect and draw the bounding box and trajectory with unique color per the same person



task 2. Count of unique, total, current and box user drawn pedestrian





task 3. Highlight form, destruct group and enter or leave the camera

Labels	Precision	Recall	mAP	Accuracy
3975	0.838	0.616	0.764	0.858

Table 2. The evaluation table of Yolov5

Here, we can observe quantitative results from the table before, the accuracy of software to predict pedestrians is 85.8% under 3975 labels. mAP stands for mean average precision, which is the area under recall and precision curve imply the performance of object detection model in which the range from those 4 metrics is 0 to 1. Here 0.764 mAP demonstrates a reliable detection in our software.

VI. Discussion

For Task 1, the team went on to three different detection methods. And by visually inspecting the results of the detection and comparing the quantitative measurement and other real-life factors, we found that YOLO v5 is most efficient and suitable for this task. In this part of the task, the subtraction algorithm had an unexpected performance in which it can not accurately detect the pedestrians. In subtracting the foreground object, we need to subtract the background and compare the difference between origin and background. That means a threshold needs to be determined. It's hard to find an appropriate static threshold for all the images, because the background average is changing frame by frame. The result is that this algorithm often detects only half of an individual pedestrian when the brightness of another half is close to the background. And after the erosion, although It removed many left background objects, it also wiped off some of the 'small pedestrians'(the pedestrians far away from the camera).

For Task 2, we use the labels generated from the trained YOLO v5 model from the previous task to represent the pedestrians. The counting is mostly correct, but in some rare cases when two or more pedestrians stagger passing by each other the method we take may produce duplicate counting. The main reason is that the algorithm decides whether a pedestrian needs to be uniquely counted by comparing the label in the current frame and the last frame. If two labels are within a certain threshold, the algorithm will consider them as the same person. If two pedestrians come across for more than a frame, one of the pedestrians is not detected during these frames, then this pedestrian will be detected as a new pedestrian. Another reason is that if there is a waggle in the camera, the relative positions of the same pedestrians will be changing a lot in this frame, and that would make the difference between two frames higher than the threshold and produce many duplicate counts.

For Task 3, the quality of result is not good in both the accuracy and the visualization. For accuracy problems, the similar problem in Task 2 occurred. When two pedestrians meet and cross by each other, the model will predict them as one pedestrian. Another problem is that when the algorithm we took decides whether several pedestrians are walking in a group, it compares the distance between the pedestrians. If the distance is below a certain threshold, the pedestrians are defined as in group. This threshold is a fixed number, but the pedestrians close to the camera are much bigger than the pedestrians far away from the camera in size. So, the

pedestrians very close to the camera can hardly be recognized in groups. Another visualization problem is that the algorithm will highlight the pedestrians entering the camera. But if a pedestrian is walking along the edge of the theme, this pedestrian will be continuously highlighted.

Ⅶ. Conclusion

Thus, the given dataset is used to detect, track and count the number of pedestrians accurately using YOLO v5. Task 2 was implemented using the model trained in Task 1 and by defining a function to compare labels predicted by model between frames. Although task 3 finished, there still exist some problems because the accuracy of executing this part is not really high.

The results for Task 1 can be further improved by configuring the hyperparameters of the YOLO v5 model. Such as increasing the batch size in every epoch during the training or exploring more training data, and seeking for a better number of NMS. Task 2 and Task 3 can be improved by using the DEEPSORT to tract and ID the pedestrians. Also for Task 3, using a dynamic distance threshold to decide whether several pedestrians are walking in groups might also increase the counting accuracy.

Contribution

(A) Laurence (Dian) Jin
Implement the task 1, 2, 3 of Background Subtraction Algorithm. Report of Experimental setup of background subtraction, Result and Analysis, some of formatting, Method of half part of yolov5, Background subtraction algorithm. Half part of introduction. Total report integration and polish.

(B) Lei Chen
Implement the method to provide quantitative measurement for YOLO in task 1. Report of Discussion, Conclusion, part of the Method and some formatting.

(C) Xu Guo
The experimental detection and matching are completed by YOLO and frame difference method. Complete tasks 1, 2 and 3 with code. Wrote the YOLO related part of the experimental setup and the method of half part of yolov5 network in the report.

(D) YunYi Li
Implement the code of FairMOT with task 1, 2 and part of 3 but not feasible in part 3, write a brief description of setup FairMOT

(E) GongLing Dong
Report of half part of introduction, Literature review and ppt

References

[1] Z. Li and W. A. Xu, “Pedestrian evacuation within limited-space buildings based on different exit design schemes,” Safety Science, vol. 124, Article ID 104575, 2020.

[2] Zhihong Li, Yang Dong, Yanjie Wen, Han Xu, Jiahao Wu, "A Deep Pedestrian Tracking SSD-Based Model in the Sudden Emergency or Violent Environment", Journal of Advanced Transportation, vol. 2021, Article ID 2085876, 13 pages, 2021.https://doi.org/10.1155/2021/2085876

[3] Hui Li, Yun Liu, Chuanxu Wang, Shujun Zhang, Xuehong Cui, "Tracking Algorithm of Multiple Pedestrians Based on Particle Filters in Video Sequences", Computational Intelligence and Neuroscience, vol. 2016, Article ID 8163878, 17 pages, 2016.https://doi.org/10.1155/2016/8163878

[4] J. Kampars and J. Grabis, “Near real-time big-data processing for data driven applications,” in 2017 International Conference on Big Data Innovations and Applications (Innovate-Data), Prague, Czech Republic, August 2017.

[5] Cao, J, Pang, Y, Xie, J, Khan, FS & Shao, L 2021, ‘From Handcrafted to Deep Features for Pedestrian Detection: A Survey’, IEEE transactions on pattern analysis and machine intelligence, vol. PP, pp. 1–1.

[6] Dalal, N & Triggs, B 2005, ‘Histograms of Oriented Gradients for Human Detection’, in IEEE Computer Society, pp. 886–893.

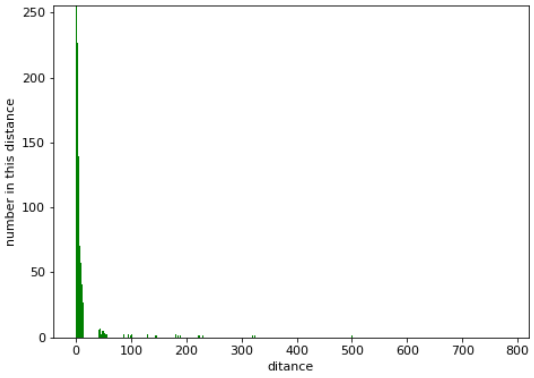
[7] Zhang, Y, Wang, C, Wang, X, Zeng, W & Liu, W 2021, ‘FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking’, arXiv.org.

[8] Krishnamurthi, R, Kumar, A & Gill, SS 2022, Autonomous and connected heavy vehicle technology, Academic Press, London.

[9] Yolov5network: Xue, J., Zheng, Y., Dong-Ye, C. et al. Improved YOLOv5 network method for remote sensing image-based ground objects recognition. Soft Comput (2022).

[10] Redmon, J. , Divvala, S. , Girshick, R. , & Farhadi, A. . (2016). You only look once: unified, real-time object detection.

[11] Jiang, P, Ergu, D, Liu, F, Cai, Y & Ma, B 2022, ‘A Review of Yolo Algorithm Developments’, Procedia computer science, vol. 199, pp. 1066–1073.



The shortest distance between the previous frame and current frame in the training set of 600 images.

Appendix

```
compute_accuracy(original, results, total_labels)
```

```
0.8586163522012579
```

```
               Class     Images     Labels          P          R     mAP@.5 mAP@
                 all        300       3975      0.838      0.616      0.764      0.411
Speed: 0.3ms pre-process, 2.4ms inference, 2.8ms NMS per image at shape (32, 3, 640, 640)
Results saved to runs/val/exp10
```

Result evaluation output by yolo