A Real-time Multiple People Tracking System in a Complex Environment

Shi-Jinn Horng
Department of Computer Science and Information Engineering, Asia University, Taichung Taiwan
Department of Medical Research, China Medical University Hospital, China Medical University, Taichung Taiwan
e-mail: horngsj@yahoo.com.tw

Hu-Ke Li
Department of Computer Science and Information Engineering,
National Taiwan University of Science and Technology,
Taipei, 10607, Taiwan, R.O.C
e-mail: 1742640235@qq.com

***Abstract*—Multiple Object Tracking is a major research field of computer vision due to increasing demand. Its application has become more and more extensive. The model proposed in this paper is an improved version of the traditional Deep Sort, which is mainly divided into two parts, the object detection part and the target tracking part. YOLOv5 (PA), the improved version of YOLOv5, is used as the front object detection model and it was trained specifically for the category of "person" in the CrowdHuman dataset, which greatly improved the detection accuracy of the model in a complex environment. Based on the Deep Sort tracking architecture, the Re-ID accuracy of the model was improved by using Mahalanobis distance, Hungarian algorithm, Aligned ReID, etc., and the tracking was predicted by Kalman filtering. In this paper, we use videos from the MOT20 dataset as the main test scenario. While achieving good MOTA and MOTP, the running speed of this model is guaranteed to achieve the effect of real-time.**

Keywords-Deep learning; target tracking; MOTA; MOTP.

I. INTRODUCTION

In recent years, with the rapid development of neural networks, deep learning has gradually received attention from all walks of life. In particular, with the rise of AlphaGo, developed by Google's DeepMind in 2017, deep learning has become a fierce competitor to other traditional algorithms.

Due to its rich and diverse datasets, computer vision has become one of the most important and rapidly developing application fields of deep learning, with a large number of applications with strong practicability, wide coverage and rapid development, for example, image classification, object detection, object tracking and semantic segmentation. These technologies have been fully integrated into every corner of our lives and continue to develop and gradually change our lives.

When taking monitoring as an example, whether it is outdoor traffic or indoor dense crowd, face recognition, ultra-distant object identification and target tracking technologies have been integrated into the monitoring system. They provide more accurate and diverse data, making surveillance more than just a picture. In particular, Multiple Object Tracking (MOT) has gradually developed and been more and more integrated into autonomous driving, pedestrian detection and other fields.

Based on the MOT20 dataset provided by MOT Challenge [1], this paper uses Deep Sort architecture with our YOLOv5 (PA) algorithm to carry out multi-target tracking. Through continuous trial and improvement, it can still complete high accuracy target tracking in the MOT20 dataset.

II. RELATED WORK

MOT has many related research directions, such as object detection, Single Object Tracking (VOT/SOT), Multi-Object Multi-Camera Tracking (MTMCT), Person Re-ID and multi-object single-camera tracking. The most commonly used datasets are MOT Challenge and Duke MTMC.

MOT Challenge, an advanced MOT competition with people as the main detection target, has been continuously developed since MOT15. In recent years, algorithms appearing on MOT Challenge are more and more accurate, among which the classic ones are Sort [2] and Deep Sort [3], which are dedicated to multi-target tracking. In addition, object detection algorithms such as Mask RCNN [4] also start to develop in this direction. All of these algorithms performed well in various competitions that year, until MOT20, the new dataset of MOT Challenge, appeared. Compared with previous MOT16 or MOT17, MOT20 has a huge difference. In MOT17, the number of persons in the same frame is roughly distributed between 10 and 30, but in MOT20, the number of persons in almost every frame is far more than 50, or even more than 100, and mutual occlusion between targets occurs more frequently. The proportion of pedestrians to the whole picture shrinks dramatically. All these present a more severe challenge to the original multi-target tracking algorithm.

In addition to Sort and Deep Sort, which are currently favored by the industry, many new algorithms have emerged in previous MOT Challenge competitions, attracting much attention. For example, MOTDT [5] made some modifications to the traditional Deep Sort. By learning Deep Sort, JDE [6] embedded the detection network into the model to improve the speed. However, these models are mainly developed in the direction of detection accuracy, using very complex algorithms, but often ignore the speed. Therefore, this paper not only

ISBN: 978-1-68558-074-2

focuses on improving the accuracy, but also controlling the running speed, so that the model can achieve real-time.

III. SYSTEM MODEL

This model is divided into two parts when making MOT. The first part is called object detection block, which identifies persons in the picture with YOLOv5 (PA) and acquires Bounding boxes. The second part, called the object tracking block, does continuous tracking of objects through Deep Sort.

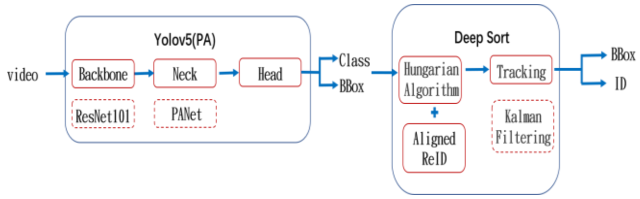

Figure 1. Model structure.

A. Detection

As shown in Figure 1, the object detection model we used is YOLOv5 (PA), which is a new model we improved based on the traditional YOLOv5 model. YOLO series is a very famous one-stage object detection model. Compared with the two-stage model, like R-CNN series, the detection accuracy is similar, but the detection speed is faster. Two-stage models divide the orientation of Bounding Box and object classification into two stages in the process of object detection, while YOLO combines them into one, which greatly improves the detection speed and meets the demand of real-time with high frame rate.

The YOLO series was continuously updated and improved from YOLOv1 [7], followed by YOLOv5, then Ultralytics [8] team, which is based on the YOLOv4 model for adjustment and improvement.

In this paper, a new YOLOv5(PA) model is obtained.

Backbone

This part is used to preliminarily extract the features of the target object. ResNet101 is used in this paper [9]. Because in this step, the network only learns some low-level features, and these features are often very similar. Therefore, it can save a lot of time in the initial training of the model by borrowing ready-made models and replacing the steps of pre-training with transfer learning.

Neck

The main function of this part is Feature enhancement. In YOLOv5 model, Feature Pyramid Networks are added [10], which is also called Feature Pyramid. In FPN of YOLOv5, as shown in Figure 2, FPN will convolve the input image first, and obtain feature maps of 76*76, 38*38 and 19*19 with different sizes in the process of convolution. Then, the feature map of 19*19 was upsampled twice (nearest neighbor interpolation method), and finally, these feature maps obtained by upsampling were added to the original feature map of the same size.

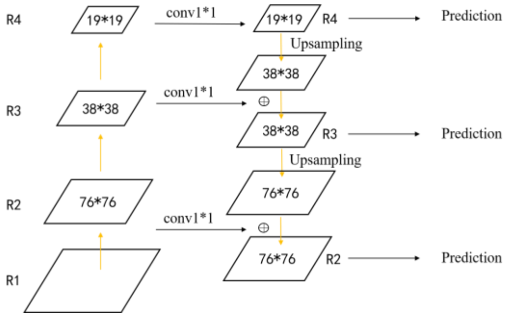

Figure 2. The FPN in YOLOv5.

Before addition, the model will conduct 1*1 convolution for the original feature map. Take R2 and P2 in Figure 2 as an example, the original size of R2 is 76*76*C1, while P2 is 76*76*C2. When C1≠C2, these two feature maps cannot be added together. Therefore, a 1*1 convolution of R2 is required to make C1 equal to C2. In this paper, the channel number is set to 225. Finally, three feature maps with different sizes were obtained, namely 19*19*225, 38*38*225 and 76*76*225. Object detection will be conducted for feature map model of each size, and the final detected target will be integrated into a map. These three feature maps of different sizes correspond to boxes of three sizes respectively. The method of convolution, up-sampling and finally addition adopted by FPN can combine the low-level features learned by neural network in the early stage and the high-level features learned in the later stage, so that the model can obtain more comprehensive target features in classification.

Head

It is the part responsible for output results in the whole model. In the YOLO series, only object classes and Bounding boxes are required. In this paper, we only focus on the object whose class is classified as "person".

B. Tracking

After receiving Bounding boxes from YOLOv5(PA), Deep Sort first selects detection targets and Bounding boxes according to confidence score. In this paper, MOT20 standard is adopted, and targets with confidence below 0.5 will be excluded (the range of confidence is 0~1).
Mahalanobis distance is mainly used to measure the feature distance between two persons during feature comparison in the subsequent Re-ID.

The Hungarian algorithm aims at finding the maximum number of matches and matching as many targets as possible. From the perspective of MOT, it means to associate each target as much as possible between adjacent frames.
Kalman filter is used for trajectory prediction in Deep Sort. It can predict the target's state at time t according to the target's state at time t-1.

IV. IMPROVED METHOD

A. *Anchor Box Size*

In the Neck part of YOLOv5, FPN provides three anchor boxes of different sizes for each feature map of different sizes, altogether nine anchor box sizes. These anchor boxes of different sizes are mainly used to match targets of different shapes and sizes, and match NMS to select the final Bounding box. However, NMS is not selected in this model, and the accuracy of Bounding Box in model is more dependent on the accuracy of Anchor Box given at the beginning. In addition, if the original anchor box size is continued to be used, plural targets may be boxed into a Bounding box. Therefore, we need to improve the size of the original Anchor box to get the anchor box that is more suitable for MOT20. The improved size of the anchor box will be closer to the size of individual pedestrian targets in MOT20, so as to improve the accuracy of Bounding box selection of model frames.

This is only a significant improvement on datasets like MOT20, which are mostly from a surveillance overhead view, with relatively small targets. Therefore, we adjusted the size of Anchor Box to be close to the target average size in MOT20. However, in MOT17, MOT16 and other datasets, a good result can be obtained even without this step adjustment. The reasons are as follows: The original anchor box size of YOLOv5 is suitable for traditional datasets such as COCO and ImageNet, and the target size of such datasets will not be too small compared to the whole image. The target sizes of MOT16 and MOT17 datasets are also close to those of COCO or ImageNet datasets, so better detection results can be obtained.

Although this change in this paper is only for the MOT20 dataset, its generality is not limited to this dataset. At present, MOT is more and more applied in the field of crowd monitoring and traffic flow monitoring.

As shown in Figure 3, the original Anchor Box of YOLOv5 can be roughly divided into three different proportions: a, b and c. We removed the Anchor Box for class c because people of this proportion are not possible in the MOT20 dataset. Category a and b are exactly corresponding to persons with standing and sitting positions. Therefore, we reserve these two categories and adjust their proportions to better match the average Bounding Box size of persons in MOT20. Among them, we further subdivide category a with different proportions to obtain two categories a1 and a2, which can take into account both taller persons and shorter persons.

B. *PANet*

In addition, we also improved FPN by adding the architecture of PANet [13] after FPN. As shown in Figure 4, we changed the model with PANet architecture into YOLOv5 (PA). FPN improves the traditional feature extraction method by adding high-level features to the shallow feature map, it is more conducive to classification. Also, PANet's improvement on FPN is to add low-level features into the deep feature map to improve the target positioning accuracy of the model, that is, Bounding Box accuracy. Why is the low-level feature helpful for target positioning? Because the low-level features are mostly features such as edge shapes, these features are particularly important when the model is doing instance segmentation, especially pixel-level segmentation. The improvement of PANet lies in its better transmission of low-level features than FPN.

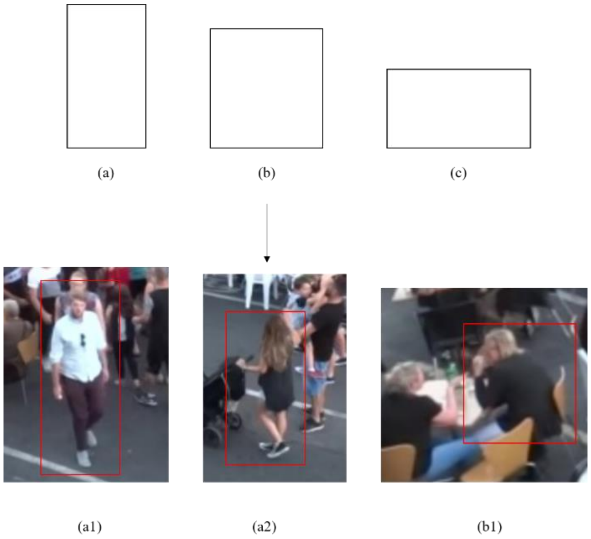

Figure 3. The types of new anchor box size

As shown in Figure 4, we can see the process of low-level feature transmission in FPN through the orange arrow. Low-level features can be transmitted to P4 only through the convolution of R2, R3 and R4. Although only THREE layers of R2, R3 and R4 are drawn in the schematic diagram, in fact, YOLOv5, R2, R3 and R4 all contain a large number of residual blocks. Therefore, the low-level features actually enter dozens or hundreds of network layers during transmission. In this process, the low-level features are inevitably lost, and few of them can be successfully transmitted. Compared with P4, although the process of low-level feature transferring to P3 and P2 has undergone fewer convolution, it still has dozens of layers.

As shown in Figure 4, the process of low-level feature transmission in PANet is represented by a green arrow. The low-level feature is first transmitted from R1 to N2 through P2 through 1*1 convolution twice, and then transmitted to N4 through two more convolution times. The convolutional network here is a very simple single-layer convolutional network, rather than a large set of residual blocks. Compared with FPN, it passes through very few layers and has few feature loss, so it can transmit more complete low-level features to the deep network. Experiments show that the accuracy of Bounding Box selection is improved after the addition of PANet. Moreover, compared with FPN, the amount of computation increased by PANet architecture is almost negligible, so it does not affect the overall running speed of the model.

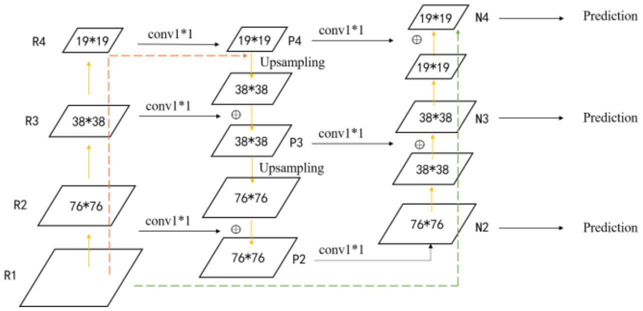

Figure 4. Architecture after adding PANet.

C. Aligned ReID

In the traditional Deep Sort architecture, kalman filter is used to predict the trajectory, and a simple CNN composed of six residual blocks acts as the function of Re-ID to do feature matching. This method has achieved good results in MOT16 and MOT17. However, in MOT20, due to the large number of targets, the trajectories of different targets overlap and a large number of interludes also appear. However, the original Re-ID model cannot perform feature matching well, which makes it easy for kalman filter to appear IDSW phenomenon due to matching errors in trajectory prediction. Therefore, we try to replace the Re-ID model and use a more effective model to enhance feature learning and clustering among targets with the same feature, so that the model can better track targets.

In the end, after many attempts, we chose Aligned ReID [14] as the new Re-ID approach and ResNet50 as the Re-ID model. Aligned ReID not only compared global features of persons, but also used dynamic programming to align local features. We will explain dynamic programming in the future, and the so-called local feature alignment refers to matching the local features of two targets one by one to facilitate the subsequent calculation of feature distance.

In general, Aligned ReID studied the local features of the target, associated all the local features as global features, and then did the final feature comparison between the two targets directly by global features. Compared with the Re-ID method which only considers the global features, the Re-ID method not only satisfies the integrity of the whole feature of the target but also gives good consideration to the local differences. Aligned ReID not only noted local features and human spatial structure as well as methods that considered only local features, but also greatly reduced computation time and cost by using only global features in the end.

As shown in Figure 5, the image is cut into multiple regions of the same size, and Aligned ReID sequentially compares the distances of each two small regions in the feature space. Starting from the comparison between the first layer in Figure 5(a) and the first layer in Figure 5(b), the first layer in Figure 5(a) corresponds to the head of the pedestrian while the first layer in Figure 5(b) is only the background. Obviously, the feature distance between the two layers is large. So Aligned ReID continues to compare the first layer of Figure 5(a) with the second layer of Figure 5(b). And so on, until a certain layer with the lowest feature distance of the first layer in Figure 5(b) is found in Figure 5(a). At this point, it is judged that the features of the two layers are similar, and then the two layers are matched together, as shown in the thick arrow in the figure. In Figure 5, we can see that matching the first layer in Figure 5(a) is the fourth layer in Figure 5(b).

Then, the second layer in Figure 5(a) is compared with each layer in Figure 5(b) in sequence, and the layer in Figure 5(b) with the shortest feature distance from the second layer is selected and matched. Finally, each layer in Figure 5(a) has the corresponding layer with the shortest feature distance in Figure 5(b), which completes the local feature alignment between the two images. However, in the final feature comparison of the two images, the first, second and third layers of Figure 5(b) will not participate in the comparison if they are not matched. In this way, the model will be less disturbed by environmental factors when comparing targets, especially its identification ability of the same pedestrian in different situations will be greatly improved, so that the feature distance between multiple images of the same pedestrian will be closer, and it will not be easy to lose or follow the wrong target in target tracking, effectively reducing IDSW.

a_i and b_j are used to represent the feature vectors of layer I in Figure 5(a), and layer J in Figure 5(b), respectively. After normalizing them, the feature distance $d_{a,b}$ between any two regions can be calculated by formula,

$$d_{i,j} = \frac{e^{\|a_i - b_j\|^2} - 1}{e^{\|a_i - b_j\|^2} + 1} \qquad i,j \epsilon\, 1,2,3,\cdots,H.$$

where H represents the number of region division. Take Figure 5 as an example, H=7.

Figure 5(c) shows us the process of feature alignment of Figure 5(a) and Figure 5(b) by dynamic programming. Each point in the figure represents a characteristic distance. For example, the point (1, 1) in the upper left corner represents the characteristic distance between the first layer in Figure 5(a) and the first layer in Figure 5(b). As shown in Figure 5(c), it takes at least 13 points to go from point (1, 1) to point (7, 7). Connecting these 13 points constitutes a "path", and the length of this path is the sum of the characteristic distances represented by each point passing through. We use I →j to indicate that the i-layer of 3-3(a) corresponds to the j-layer of Figure 5(b), and the final correspondence between Figure 5(a) and Figure 5(b) is 1→4, 2→5, 3→5, 4→6, 5→6, 6→7 and 7→7. For the model to find the shortest path, the points (1,4), (2,5), (3,5), (4,6), (5,6), (6,7), and (7,7) representing the seven sets of corresponding relationships must all be on that path.

We use $S_{i,j}$ to represent the shortest distance from (i, j) to (1,1) at any point in Figure 5(c).

$$S_{i,j} = \begin{cases} d_{i,j} & i = 1, j = 1 \\ S_{i-1,j} + d_{i,j} & i \neq 1, j = 1 \\ S_{i,j-1} + d_{i,j} & i = 1, j \neq 1 \\ \min\left(S_{i,j-1}, S_{i-1,j}\right) + d_{i,j} & i \neq 1, j \neq 1 \end{cases}$$

For example, in Figure 5, Aligned ReID's goal is to find the shortest path from (7, 7) to (1, 1), which is the minimum $S_{7,7}$. Note that this shortest path does not represent the final characteristic distance of the two graphs in Aligned ReID.

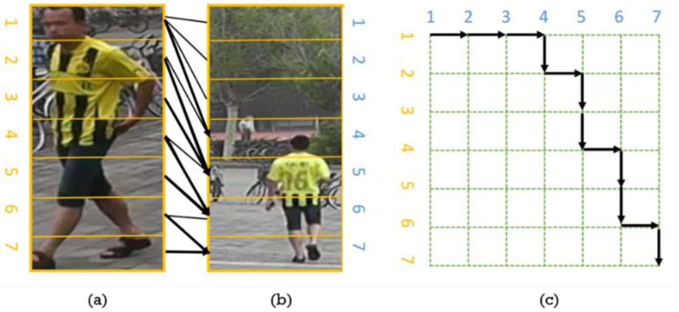

Figure 5. Aligned ReID to find shortest paths [14].

V. Experiments

A. Datasets

In this paper, there are two datasets involved in model training: CrowdHuman [15] for training object detection model YOLOv5 (PA), and Market-1501 [16] for training Aligned ReID.

1) CrowdHuman

We used the CrowdHuman database, which is open source by Cut Technology, a leading AI unicorn company in China. Why CrowdHuman?

Although the traditional YOLOv5 [18] model trained from COCO [17] dataset performed well in MOT16 and MOT17, due to the huge increase in the difficulty of MOT20 dataset, it could not achieve such satisfactory results in MOT20.In MOT20, the difficulty of multi-target tracking of this dataset is much higher than that of several MOT datasets due to the increase of the number of persons in the same frame, the decrease of the proportion of persons to the whole graph and the frequent occurrence of mutual occlusion among persons. Although COCO is a good dataset of quantity and quality, the models trained from it still do not meet the MOT20 requirements.

After comparing with several datasets, we finally chose to use CrowdHuman instead of COCO to train the model. We believe that CrowdHuman is currently the most suitable dataset for training pedestrian detection models, especially dense crowds. Here are some comparisons between the CrowdHuman dataset and the COCO dataset.

As shown in Table I, CrowdHuman far outnumbered COCO in the data volume of persons, and the average number of people in each graph was also higher than COCO, which was more suitable for MOT20 with a dense crowd. Although the COCO dataset is also a well-known object detection dataset with a wide range of content, the proportion of the dataset allocated to persons is not as large as other pedestrian-specific datasets.

The MOT20 dataset is taken in a very different way from the original MOT16 and MOT17 datasets, with cameras instead of close street shots. The pedestrian in the picture is relatively small and difficult to detect. Moreover, most of the persons in the picture are shot from a overlooking Angle, which is not common in COCO. In COCO dataset, the target size of persons is usually moderate relative to the whole image, so it is difficult for the model trained with COCO dataset to accurately detect the persons of small size, and they are often taken as the background, resulting in "missing report". However, in CrowdHuman's dataset, there are not only conventional images like COCO, but also many images taken at a distance or from an overhead perspective, which greatly improves the model's ability to detect persons in various situations.

There is a very deadly marking method for MOT tasks in the COCO data set when marking Ground Truth, which we call "crowd marking". COCO data assembly makes multiple persons with similar positions share an Bounding Box, and then gives this big Box a label of "person", that is, multiple targets are judged as one target. This is not acceptable in MOT, where any MOT task wants to isolate as many targets as possible.

TABLE I. COPARISION OF DATA AMONG DATA SETS

	Caltech	KITTI	City Persons	COCO Persons	Crowd Human
images	42,782	3,712	2,975	**64,115**	15,000
persons	13,674	2,322	19,238	257,252	**339,565**
persons/image	0.32	0.63	6.47	4.01	**22.64**

In addition, the most special part of The CrowdHuman data set is its Grounding Truth. It provides three different Bounding boxes, namely Head Box, Full Box, Visible Box, which are abbreviated as HBox, FBox, VBox in the following. This is a unique feature of the dataset. HBox is specifically for the head, which will not be used in this paper, while FBox and VBox are for the whole body of persons. The difference is the following: as the name implies, VBox boxes the part of persons that can be seen, while FBox boxes the whole pedestrian, including not only the visible part but also the blocked part. This labeling method does not exist in other datasets, as shown in Figure 6. The training of models using such datasets can well achieve the purpose of modal labeling (Amodal).

Modal labeling is a concept proposed by Deng Z. and Jan Latecki L. [19], originally used for 3D object detection, which refers to enclosing invisible parts of objects in Bounding boxes at the same time. For example, in the detection, only the upper body of a pedestrian is uncovered while the lower body is covered, but the Bounding Box area is extended to the foot of the pedestrian it deems to be completed completely by the modal labeling.

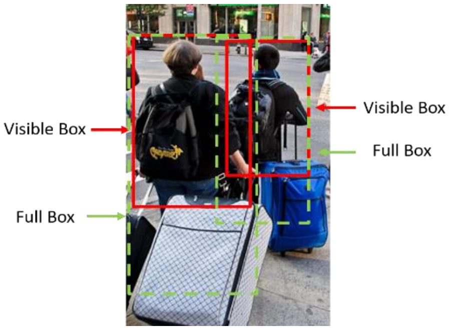

Figure 6. Two different boxes in the CrowdHuman dataset.

2) *Market-1501*

Since MOT20 was much more difficult than before, we retrained a better Re-ID model to replace the original Deep Sort module responsible for this function with the Market-1501 dataset. Market-1501 is a dataset dedicated to Person Re-ID released by Tsinghua University of China in 2015. There are 32668 images in this dataset, and each image is 64*128 in size. There is a totally 1501 different persons. There were 751 people in the training set with an average of 17.2 images per person, and 750 people in the test set with an average of 26.3 images per person. The dataset was captured by six cameras, each with a different resolution. Even the image of the same pedestrian has a large number of different features and environmental interference factors. As shown in Figure 7, each column shows the same pedestrian in different poses with different cameras and shooting angles.

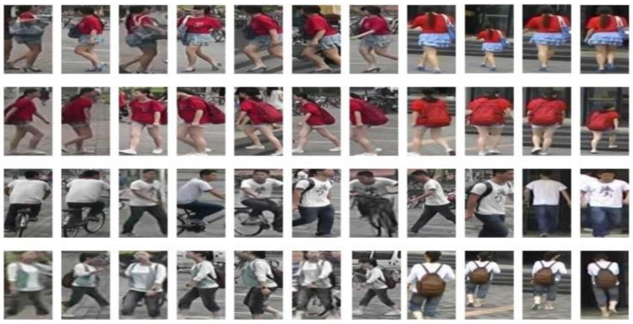

Figure 7. Market-1501 dataset.

B. *Results*

Using MOT20 video as input, the model identifies each pedestrian frame by frame, assigns each pedestrian a unique ID, and keeps the ID corresponding to the pedestrian until the pedestrian leaves the frame. In the MOT20 competition of MOT Challenge, MOTer [20] is the model with many leading data, and we take it as the benchmark for comparison. As shown in Table II, our model's MOTPI and MOTPC were 35.6 and 77.9, respectively, after the training of CrowdHuman dataset. By changing the size of Anchor Box and adding PANet architecture, we improved YOLOv5 and obtained YOLOv5(PA), which further improved the Bounding accuracy of Bounding boxes, contributing greatly to the improvement of MOTPI and MOTPC. But we are still 2% behind Moter in MopI.

With Aligned ReID, we compared local features between targets in addition to global features. Therefore, the model obtained excellent ReID ability, reduced the target mismatching between before and after frames, reduced the occurrence of IDSW, and MOTA was significantly improved to 69.7, surpassing the traditional Sort20 and MOTer.

Moreover, it is worth mentioning that the operation speed of MOTer model is 1 second/frame, that is, it takes about 1 second to process an image. Generally speaking, it is considered that 1 second /12 frames is about 12 images output per second to be a smooth picture. Therefore, it is difficult for MOTer model to achieve real-time, which is fatal in real-time monitoring. However, our model relies on YOLOv5 (PA)'s super fast computing speed to stabilize the processing time of each frame at about 0.1 seconds, which is far superior to MOTer in this point and can achieve basic real-time tracking. Figure 8 shows the results on MOT 20. Table II shows the results compared to some existing results.

TABLE II. THE EXPERIMENTAL RESULTS

	MOTA↑	MOTPI↑	MOTPC↓	s/ frame
MOTer [20]	58.6	**79.8**	/	1.0
Sort20 [2]	42.7	78.5	/	0.7
ours	**69.7**	77.9	35.6	**0.1**

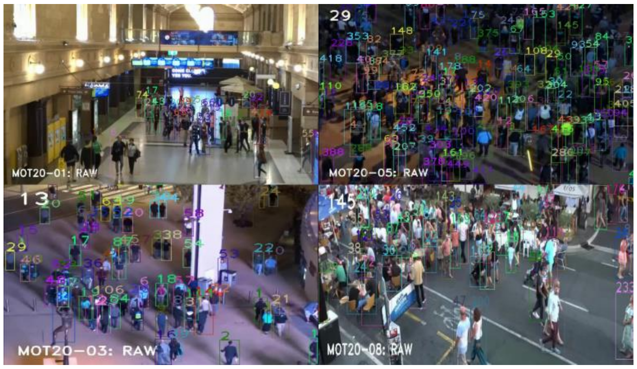

Figure 8. Results in MOT20.

VI. Conclusions and Future Work

In this paper, experimental results show that the performance of the proposed method outperforms that of other approaches. During the testing, for the dataset like MOT20, it is still hard to get better performance due to many people. Hence, in future work, a good method to resolve the crowded people needs to be proposed.

ACKNOWLEDGMENT

This work was supported in part by the National Science and Technology Council under contract numbers 111-2221-E-011 -134 -, 111-2218-E-011 -011 –MBK.

REFERENCES

[1] MOT challenge, https://motchallenge.net/

[2] A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft, "Simple online and realtime tracking," *IEEE International Conference on Image Processing (ICIP)*, 2016.

[3] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *IEEE International Conference on Image Processing (ICIP)*, 2017.

[4] K. He, G. Gkioxari, P. Dollar and R. Girshick, "Mask R-CNN," *IEEE International Conference on Computer Vision*, 2017, pp. 2961-2969.

[5] L. Chen, H. Ai, Z. Zhuang and C. Shang, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," *IEEE International Conference on Multimedia and Expo (ICME)*, 2018.

[6] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," arXiv preprint arXiv:1909.12605 2.3 (2019): 4.

[7] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779-788.

[8] Ultralytics： https://ultralytics.com/

[9] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778.

[10] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature pyramid networks for object detection," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117-2125.

[11] B. Keni and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP Journal on Image and Video Processing*, 2008, pp. 1-10.

[12] R. Ergys *et al.*, "Performance measures and a data set for multi-target, multi-camera tracking," *European Conference on Computer Vision*, Springer, Cham, 2016.

[13] S. Liu, L. Qi, H. Qin, J. Shi and J. Jia, "Path aggregation network for instance segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759-8768.

[14] X. Zhang *et al.*, "Alignedreid: Surpassing human-level performance in person re-identification," arXiv preprint arXiv:1711.08184, 2017.

[15] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "Crowdhuman: A benchmark for detecting human in a crowd," arXiv preprint arXiv:1805.00123, 2018.

[16] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang and Q. Tian, "Scalable person re-identification: A benchmark," *IEEE International Conference on Computer Vision*, 2015, pp. 1116-1124.

[17] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," *European Conference on Computer Vision*, Springer, Cham, 2014.

[18] Yolov5: https://github.com/ultralytics/yolov5

[19] D. Zhuo and L. Jan Latecki, "Amodal detection of 3d objects: Inferring 3d bounding boxes from 2d ones in rgb-depth images," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5762-5770.

[20] Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, and X. Alameda-Pineda, "TransCenter: Transformers with dense queries for multiple-object tracking," arXiv preprint arXiv:2103.15145, 2021.

[21] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "Yolov4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[22] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," *18th International Conference on Pattern Recognition (ICPR'06)*, Vol. 3, IEEE, 2006.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, issue 6, June 2016, pp. 1137-1149.