Multi-object Tracking Based on YOLOX and DeepSORT Algorithm

Guangdong Zhang, Wenjing Kang(✉), Ruofei Ma, and Like Zhang

Harbin Institute of Technology, Weihai 264209, Shandong, China
{kfjqq,maruofei}@hit.edu.cn

Abstract. The implementation of 5G/6G network provides high-speed data transmission with a peak transmission rate of up to 10 Gbit/s, which solves the problems of blurred video and low transmission rate in monitoring systems. Faster and higher-definition surveillance images provide good conditions for tracking multiple targets in surveillance video. In this context, this paper uses a two-stage processing algorithm to complete the multi-target tracking task based on the surveillance video in the 5G/6G network, realizing the continuous tracking of multiple targets and solving the problem of target loss and occlusion well. The first stage uses YOLOX to detect the target and passes the detection data to the DeepSORT algorithm of the second stage as the input of Kalman Filtering, and then use the deep convolutional network to extract the features of the detected frames and compare them with the previously saved features. The algorithm can better continuously track multiple targets in different scenarios and achieve the real-time effect of the processing of monitoring video, which has certain significance for solving the problems of large-scale dense pedestrian detection and tracking and pedestrian multi-object tracking for pedestrians in the future 5G/6G video surveillance network.

Keywords: Multi-object tracking · YOLO · Deep convolutional neural network · Kalman filter

1 Introduction

1.1 MOT Method

At present, there are mainly two kinds of multi-object tracking (MOT) methods, Tracking by Detection (TBD) and Detection Free Tracking (DFT) [1]. Figure 1 clearly shows the difference between the two types of algorithms. DFT is similar to single target tracking. It is necessary to manually mark the target in the first frame of the video when initializing the target, and then detect while tracking. The incompleteness of manual annotation may cause instability of tracking results, so compared with DFT, TBD is more efficient which is commonly used and is the most effective paradigm for MOT in current. The MOT based on the TBD strategy includes an independent detection process, a process in which detection results and tracker trajectories are connected. Number of TBD tracking

A. Li et al. (Eds.): 6GN 2022, LNICST 504, pp. 52–64, 2023.
https://doi.org/10.1007/978-3-031-36011-4_5

targets and types are related to the results of the detection algorithm, usually the detection results are unpredictable, so the performance of TBD basically depends on the quality of the detection results. Simple Online Realtime Tracking with Deep Association Metric (DeepSORT) [2] is a MOT algorithm based on the TBD strategy which implements tracking by designing an association strategy for detection results and tracking prediction results.

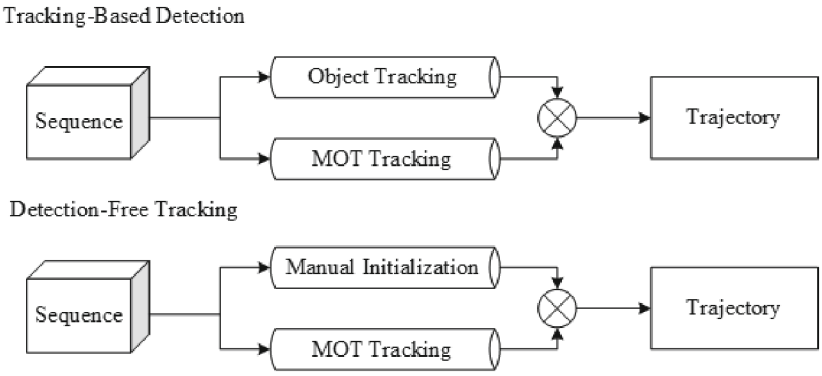

Fig. 1. MOT tracking strategy

1.2 Related Work

Figure 2 shows that the vast majority MOT algorithm consists of four steps. The two key components of MOT are object detection and data association. The estimated bounding box is realized by detection, while the identity is realized by association.

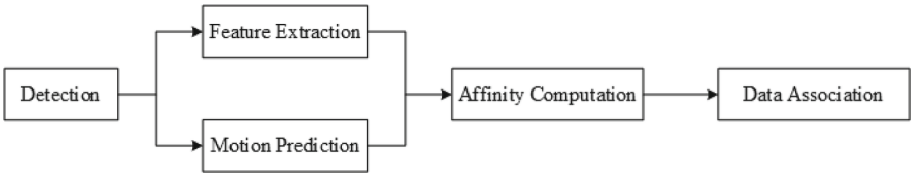

Fig. 2. MOT algorithms key steps

Many methods use detectors with higher performace to obtain higher tracking performance. A lot of methods used CenterNet [3] due to its simplicity and efficiency. A large number of methods [4] used the YOLO series detectors [7] because of its balance of accuracy and speed. The detection box on a single image is used to complete the tracking task by most methods This practice can also help to obtain a more accurate detection box. There are many other methods [9] to enhance the feature representation of subsequent frames by tracking the boxes in the previous frame. Several methods used

transformer-based [12] detectors [10] because it has the powerful ability to propagate boxes between frames.

As the core of multi-target tracking, the process of data association first computes the similarity, which is the basis for matching, between tracklets and detection boxes. The way of combining location and motion cues used by SORT [6] is very simple. Firstly, Kalman Filter [8] is used to predict the position of the trajectory in the new frame. Then the Intersection over Union (IOU) between the detection frame and the prediction frame is calculated as the similarity. The way that Sort matches the check boxes with the tracklets is once matching. The first step of MOTDT [5] is to match through appearance similarity, and the second part uses IOU similarity to match the previously mismatched trajectories.

2 Target Detecting Based on YOLOX

2.1 ConvNeXt

The structure of YOLOX network (Fig. 3) mainly consists of four parts including Imput, Backbone, Neck and Prediction. YOLOX's Backbone actually is a convolutional neural network which adopts Darknet53 network structure. It is used to forms image features and aggregate different fine-grained images.

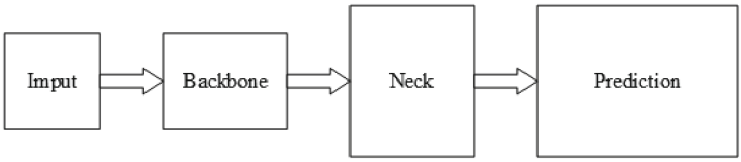

Fig. 3. YOLOX-Darknet53 structure

In this paper, the convolutional network in Backbone is replaced by ConvNeXt. In ConvNeXt, its optimization strategy draws on Swin-Transformer. Specific optimization strategies include: increase the number of training Epochs from 90 to 300, change the optimizer from SGD to AdamW, more complex data augmentation strategies including Mixup, CutMix, RandAugment, Random Erasing and so on, add regularization strategies, such as random depth [13], label smoothing [14] and so on.

Starting from ResNet-50, the five perspectives of macro design, ResNeXt, MobileNet v2, large convolution kernel, and detailed design are drawn from the ideas of Swin Transformer in turn, and then carried out on ImageNet-1K Training and evaluation, and finally get the core structure of ConvNeXt. In the macro design, the improvement of ConvNeXt is to adjust the ratio of the blocks of each Stage of ResNet-50 to 1:1:3:1, and the final number of blocks is (3, 3, 9, 3). This improvement increases the accuracy of ResNet-50 from 78.8% to 79.4%. ResNeXt is a more compromised solution, which improves the computational speed of the model by grouping convolutions (grouping channels and then convolving in groups). Similarly, the Self-Attention of Swin-Tranformer is also an operation unit in units of channels. The difference is that the separable convolution is

a learnable convolution kernel, and Self-Attention is a weight dynamically calculated according to the data.

In ConvNeXt, the idea of grouped convolution is also introduced. It replaces 3*3 convolutions with 3*3 grouped convolutions, which reduces GFLOPs from 4.4 to 2.4, but it also reduces accuracy from 79.5% to 78.3%. To compensate for the drop in accuracy, it increases the base channel count of ResNet-50 from 64 to 96. This operation increases GFLOPs to 5.3, but improves the accuracy to 80.5%. ConvNeXt also uses the structure of the inverse bottleneck layer. The bottleneck layer is a structure with a small middle and large ends, which was first used in the residual network. In MobileNet v2, a structure with large middle and small ends is used, which can effectively avoid information loss.

2.2 FPN and PAN

A series of network layers constitute the main structure of Neck. The function of this part is mixing and combining image features and transmit image features to the prediction layer. The neck structure of YOLOX-X is mainly composed of FPN and PAN (Fig. 4).

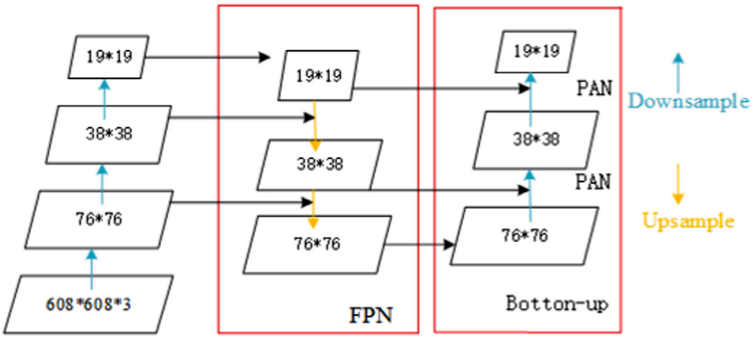

Fig. 4. FPN and PAN structure

Semantic information is transmitted by FPN from high-dimensional to low-dimensional. From top to bottom, FPN conveys strong semantic features at the high level, and enhances the whole pyramid, but only enhances semantic information and does not transmit location information. Aiming at this point, PAN adds a bottom-up pyramid behind FPN to complement FPN and transmits the powerful localization features of the underlying layer and semantic information from low-dimensional to high-dimensional again.

2.3 Decoupled Head

The Prediction part of the YOLOX structure, which Decoupled Head belongs to, can predict image features to generate boundary boxes and prediction categories. Compared with Yolo Head, Decoupled Head has faster convergence and higher accuracy. However, it should be noted that decoupling the detection head will increase the complexity of

the operation. After a trade-off between speed and performance, a 1x1 convolution is used for dimensionality reduction first, and two 3x3 convolutions are used in each of the latter two branches to adjust the network parameters to only increase a little. Decoupling the detection head has a deeper importance: YOLOX's network architecture can be integrated with many algorithmic tasks.

Extract the Decoupled Head 1 in Yolox-Darknet53 (Fig. 5). Passing through the previous Neck layer, the length and width of the Decoupled Head 1 input is 20*20. There are three branches in front of Concat:

(1) cls_output: Mainly for the category of the target box, predict the score. Because the COCO data set has a total of 80 categories, and it is mainly N two-category judgments, it becomes 20*20*80 size after being processed by the Sigmoid activation function.
(2) obj_output: It mainly judges whether the target frame is foreground or background, so it is processed by Sigmoid and becomes 20*20*1 size.
(3) reg_output: It mainly predicts the coordinate information (x, y, w, h) of the target frame, so the size is 20*20*4.

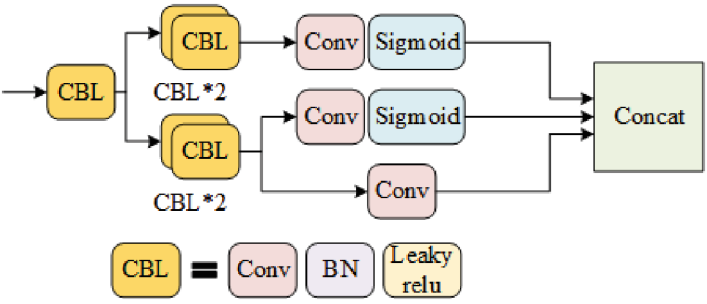

Fig. 5. Decoupled Head 1 structure

The last three outputs are fused together by Concat to obtain 20*20*85 feature information. Similarly, Decoupled Head 2 outputs feature information and performs Concat to obtain 40*40*85 feature information. Decoupled Head 3 outputs feature information and performs Concat to obtain 80*80*85 feature information. Then, perform the Reshape operation on the three pieces of information of Decoupled Head 1, 2 and 3, and perform the overall Concat to obtain the prediction information of 8400*85. After a Transpose, it becomes a two-dimensional vector information of 85*8400 size. Here 8400 refers to the number of prediction boxes, and 85 is the information of each prediction box (Fig. 6):

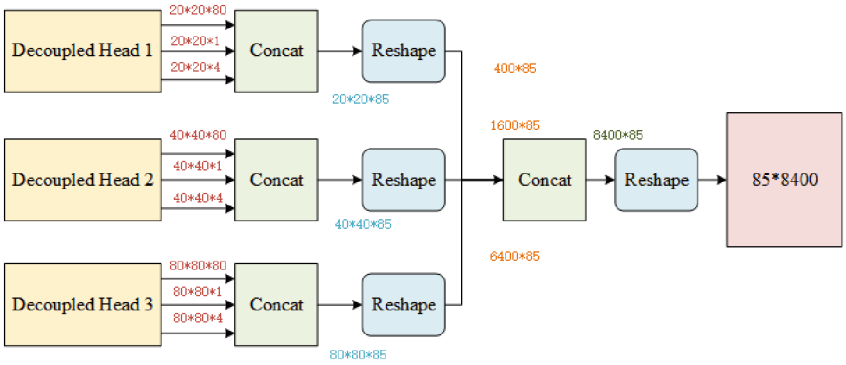

Fig. 6. Prediction structure

3 Target Tracking Based on DeepSORT Algorithm

3.1 Kalman Filter

DeepSORT is a very efficient classical MOT algorithm based on TBD strategy, which completes the task of identifying multiple targets according to the detection results and uses the detection recognition mechanism to help the tracker predicting the trajectory to achieve the function of connecting and distinguishing targets in adjacent images. Since DeepSORT recognizes multiple targets based on the detection algorithm, the tracking effect of DeepSORT is related to the result of target detection, and different detectors can achieve different tracking effects.

In the DeepSORT algorithm, the Kalman Filter is used to realize the tracker's prediction function of the motion trajectory, and the Hungarian Algorithm is used to realize the optimal distribution function of the detection results and the tracker's prediction results.

As a method of optimal state estimation, Kalman Filter plays an irreplaceable role in automatic control systems that need to be pre-judged. On the one hand, Kalman Filter has certain effects on system parameters containing noise and inaccurate observations. The fault tolerance of, on the other hand, is due to the overall performance of the prediction step of the Kalman Filter as far as possible to achieve the optimal estimation of the state value of the dynamic system. Tracking predicts the future state based on the current trajectory of the target, so the Kalman Filter is an essential part of the tracking algorithm.

Formula (1) represents a mathematical model of Kalman Filter that satisfies the basic assumptions of a discrete linear dynamic system.

$$x_k = \mathbf{A} * x_{k-1} + \mathbf{B} * u_k + w_{k-1} \;\; z_k = \mathbf{H} * x_k + v_k \tag{1}$$

where x_k is the system state matrix, $\mathbf{A}$ is the state transition equation, $\mathbf{H}$ is the state observation matrix, and w_{k-1} is the process noise. z is the observed amount of the state matrix obtained by the actual measurement, which corresponds to the state x_k obtained by the system simulation. $\mathbf{B}$ is the control input matrix. Two noise parameters, process noise and measurement noise, are introduced to achieve the fault tolerance of the Kalman

filter. v_k is the Gaussian measurement white noise, the covariances of the process noise w_{k-1} and the measurement noise v_k are $\mathbf{Q}$ and $\mathbf{R}$, which Satisfy formula (2).

$$p(w) \in N(0, \mathbf{Q}), p(v) \in N(0, \mathbf{R}) \tag{2}$$

In terms of prediction, the Kalman Filter uses the state prediction equation shown in formula (3) to calculate the predicted state value.

$$x_k^- = \mathbf{A} * x_{k-1}^- + \mathbf{B} * u_k \tag{3}$$

Kalman Filter performs state estimation by setting three state quantities: a priori state predicted value x_k^-, a posteriori optimal estimated value x_k and the actual value x_k. $\mathbf{K}$ is the gain for Kalman, which Indicates the proportion of prediction error to measurement error. The state update equation is used to calculate the optimal estimated value x_k of the state, as shown below:

$$x_k = \mathbf{K}(z_k - \mathbf{H} * x_k^-) \tag{4}$$

The cost function is calculated by the state estimation covariance shown below:

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}\mathbf{H}\mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}^T\mathbf{K}^T + \mathbf{K}(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R})\mathbf{K}^T \tag{5}$$

where $\mathbf{P}_k^-$ is the covariance of the true value and the predicted value, $\mathbf{P}_k$ is the covariance between the true value and the optimal estimated value. The Kalman gain matrix $\mathbf{K}$ under the optimal estimation condition is calculated by formula (6), and the estimation error variance matrix is calculated as formula (7). The calculation of the prediction covariance matrix $\mathbf{P}_k^-$ is shown in formula (8).

$$\mathbf{K} = \mathbf{P}_k^-\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R})^{-1} \tag{6}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}\mathbf{H}) * \mathbf{P}_k^- \tag{7}$$

$$\mathbf{P}_{k+1}^- = \mathbf{A}\mathbf{P}_k\mathbf{A}^T + \mathbf{Q} \tag{8}$$

3.2 Hungarian Algorithm

The Hungarian Algorithm is a method to find the optimal allocation. Its classical mathematical model is the assignment problem, and its general form is shown in formula (9), where c_{ij} is the efficiency matrix.

$$\begin{cases} min\ \mathrm{z} = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\ s.t.\ \sum_{j=1}^{n} x_{ij} = 1, j = 1, 2, ..., n \\ \sum_{i=1}^{n} x_{ij} = 1, i = 1, 2, ..., n \\ x_{ij} = 0\ or\ 1, i, j = 1, 2, ..., n \end{cases} \tag{9}$$

In order to solve the allocation problem of detection and tracking results in DeepSORT, the Hungarian Algorithm is used to find an optimal allocation with the least cost between the detection results of the detector and the tracking trajectories of the tracker. DeepSORT adopts the weighted Hungarian Algorithm to track the association between objects frame by frame, and uses the IoU distance as the weight of the Hungarian Algorithm. The setting of the IoU threshold also has a certain tolerance for short-term occlusion, but it can only be established when the obstacle is slightly larger than the target.

When DeepSORT works, the tracker first estimates the location of each target in the next image through Kalman Filtering, and then calculates the IoU with the prediction result of the tracker according to the recognition result of each detector in the next image. The IoU is taken as the cost matrix of the Hungarian Algorithm. The Hungarian algorithm is used to optimize the distribution of the trajectory of each target. When the overlap between the detection frame and the prediction frame is less than the IoU threshold, it refuses to match the two.

3.3 Algorithm Flow

DeepSORT reduces the frequency of ID-Switch by integrating appearance information. As can be seen from the Fig. 7, DeepSORT adds a cascade matching strategy based on the SORT algorithm, while considering the target distance and feature similarity, and adopts a verification mechanism for the newly generated tracking trajectory to eliminate erroneous prediction results. The core process of DeepSORT is consistent with SORT, and it follows the combination of prediction, observation, and update. In this paper, the convolutional network in this algorithm is replaced by OSNet for better performance.

The DeepSORT matching process is divided into the following situations:

(1) Kalman predicting and detecting match successfully.

After each frame of image in the video is predicted by Kalman filter, the predicted trajectory bounding box of all objects in the current frame is generated, and the detected and predicted bounding boxes are data-related according to the detection results of the detector in the current frame.

The estimated tracking trajectory bounding box is updated for the Kalman filter prediction result that has the corresponding detection result to be matched, and then the next frame is tracked, and the process of observation, prediction, and matching update is performed cyclically.

(2) Kalman predicting and detecting match unsuccessfully.

When the detection is missed, it is easy to lead to the situation that some tracking trajectories do not match the detection results, that is, the matching of the tracking trajectories is missing. At the same time, there is also a situation where the detection result lacks the matching tracking trajectory, which is easy to occur in the scene where a new target enters the camera's field of view. Since the new object entering the field of view has no past trajectory for Kalman filter prediction, the tracking trajectory is missing, resulting in the lack of detection matching. In addition, when an object is occluded for a long time and exceeds the life limit of consecutive matching failures, the algorithm will

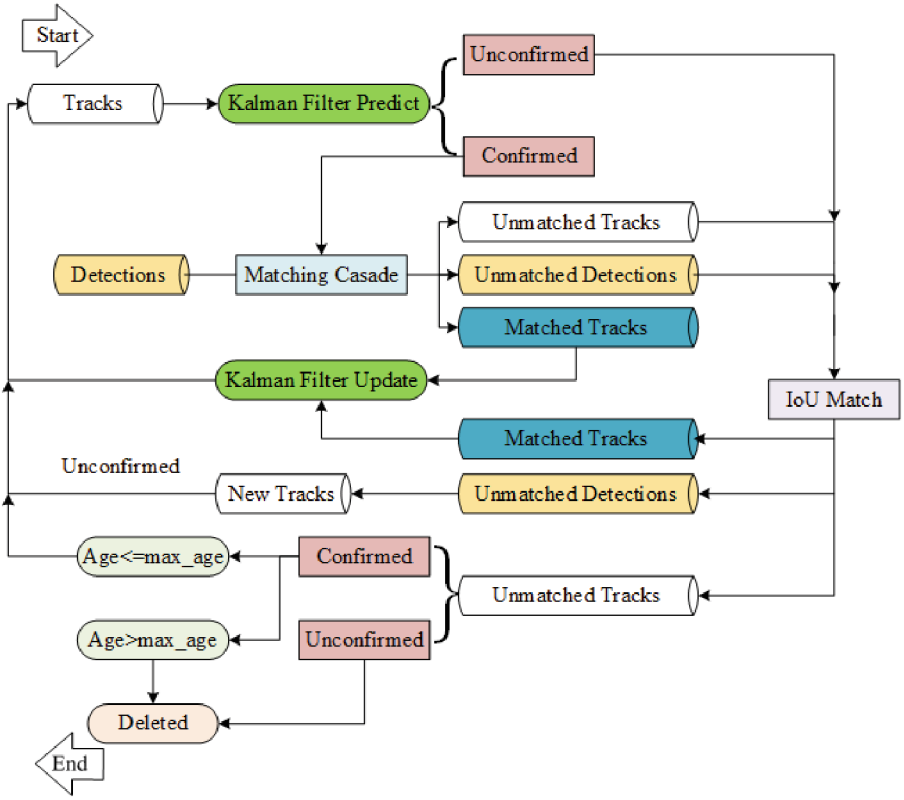

Fig. 7. Algorithm flow

consider that it will no longer appear in the lens and delete the object trajectory, which will also lead to a lack of detection matching.

For the prediction and detection bounding boxes that fail to match, DeepSORT will calculate the IoU again for secondary matching, and re-match the failed matching caused by interference factors such as calculation errors through secondary matching, so as to reduce the remaining detection and tracking results as much as possible. For the detection of secondary matching failure, a new trajectory is established and marked as an unreal trajectory.

After three matching inspections, if the matching is successful three times in a row, it is modified and marked as a real trajectory, and added to the trajectory set. For the tracking box that fails to be matched again, consider the case where the detector misses detection. If the track is marked as untrue, delete its track. If it is marked as true, set a lifetime for it. If the track is marked as true, it will still fail to match within the lifetime. The target has moved out of the shot, so the track is deleted.

4 Experiments

4.1 Setting

The detector is YOLOX-X whose backbone is replaced by ConvNeXt network. There is no doubt that the key point is to train the improved detector. The feature extraction threshold τ_f is set to 0.85 and matching threshold is 0.13. The input image size, the shortest side, the optimizer and other parameters used during multi-scale training adopte what is described in [11].

Training was performed on NVIDIA GTX2080Ti GPU for 80 epochs using a combined training schedule containing multiple datasets including Cityperson, ETHZ, MOT17 and CrowdHuman. Mosaic and Mixup are included in data augmentation.

After the training is completed, four videos are tested on the NVIDIA GTX1080 GPU. The loss function of the COVNeXt after training is as Fig. 8. Moreover, we evaluate the integrated tracker on MOT17 datasets under the "private detection" protocol. Both datasets contain training sets and test sets.

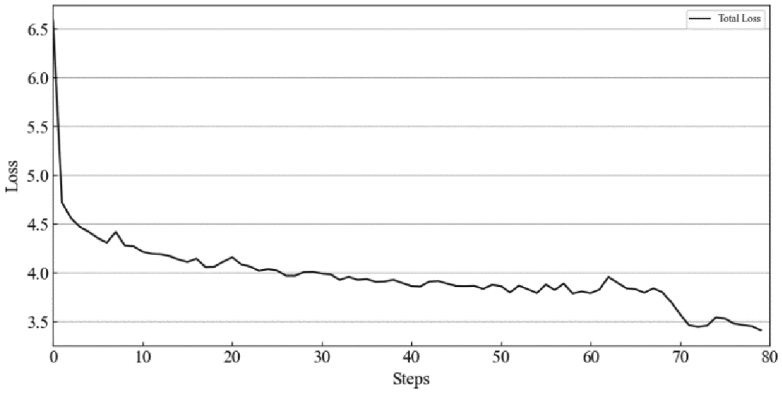

Fig. 8. Iterative loss value

4.2 Visualization Results

Four surveillance videos in different scenes including living area, teaching area and intersection, which are recorded by the camera in the format of 1080P and 30 FPS, are tracked using this algorithm. The algorithm can achieve the mission of tracking multiple targets. When occlusion occurs and the occluded object reappears, the algorithm can still identify and track the target according to the original identity information of the target, give the same ID at the same time (Fig. 9).

Scene 1 is two frames of images taken in the living area. It can be seen that the algorithm can effectively track almost all visible targets, and the targets can be re-identified and tracked after being occluded, and the same detection frame is given. The tracking effect of other scenes is also similar to scene 1. In the very dense environment of scene 3, a good tracking effect is also achieved.

Fig. 9. Visualization results in different scenarios

We compare the integrated tracker with state-of-the-art trackers on the test set of MOT17. The evaluate results are shown in Table 1. Our tracker achieves state-of-the-art performance under the "private detector" protocol. We get 74.9 MOTA, 67.0 IDF1 and 12819 FP.

Table 1. Comparison of the state-of-the-art methods under the "private detector" protocol.

Tracker	MOTA↑	HOTA↑	IDF1↑	FP↓	FN↓	IDs↓	Frag↓
PermaTrackpr	73.8	55.5	68.9	28998	115104	3699	6132
OCSORT	78.0	63.2	77.5	15129	107055	1950	**2040**
GSDT	66.2	55.5	38.7	43368	144261	3318	8046
StrongSORT	79.6	**64.4**	**79.5**	27876	86205	**1194**	1866
ByteTrack [11]	**80.3**	63.1	77.3	25491	**83721**	2196	2277
TraDes	69.1	52.7	63.9	20892	150060	3555	4833
Ours	74.9	58.1	67.0	**12819**	125874	3165	3474

5 Conclusion

Based on the realization and application of 5G/6G technology, this paper realizes multi-target tracking in different complexity scenarios through YOLOX detector and DeepSORT algorithm. The algorithm can better track a large number of targets continuously and solve the problem of target occlusion and loss to a certain extent. It has certain value in solving the multi-object tracking problem in surveillance video under 5G/6G network.

Acknowledgement. This work was supported partially by National Natural Science Foundation of China (Grant Nos. 61971156, 61801144), Shangdong Provincial Natural Science Foundation (Grant No. ZR2019QF003, ZR2019MF035, ZR2020MF141), the Fundamental Research Funds for the Central Universities, China (Grant No.HIT.NSRIF.2019081) and the Scientific Research Innovation Foundation in Harbin Institute of Technology at Weihai (Grant No. 2019 KYCXJJYB06).

References

1. Luo, W., Xing, J., Zhang, X.: Multiple object tracking: A literature review. Artif. Intell. **293,** 103448 (2021)
2. Wojke, N., Bewley, A., Paules, D.: Simple online and real-time tracking with a deep association metric. In: Proceedings of the 2017 IEEE International Conference on Image Processing, Beijing, China, pp. 3645–3649 (2017)
3. Zhou, X., Wang, X., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)
4. Wang, Z., Zheng, L., Liu, Y., Li, Y., Wang, S.: Towards real-time multi-object tracking. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12356, pp. 107–122. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58621-8_7
5. Chen, L., Ai, H., Zhuang, Z., Shang, C.: Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: 2018 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE (2018)
6. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: ICIP, pp. 3464–3468. IEEE (2016)

7. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
8. Kalman, R.E.: A new approach to linear filtering and prediction problems. J. Fluids Eng. **82**(1), 35–45 (1960)
9. Liang, C., Zhang, Z., Zhou, X., Zhou, Li, B., Lu, Y., Hu, W.: One more check: Making "fake background" be tracked again. arXiv preprint arXiv:2104.09441 (2021)
10. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision, pp. 213–229. Springer (2020)
11. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z.: ByteTrack: Multi-Object Tracking by Associating Every Detection Box. arXiv preprint arXiv:2110.06864 (2021)
12. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I.: Attention is all you need. In Advances in neural information processing systems, pp. 5998–6008 (2017)
13. Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.: Deep networks with stochastic depth. In: European conference on computer vision. Springer, Cham (2016)
14. Müller, R., Kornblith, S., Hinton, G.: When does label smoothing help? arXiv preprint arXiv:1906.02629 (2019)