SportsTrack

Jie Wang[*], Xiaodong Yang, Pengyu Zhou, Ting Wang, Yanming Zhang
BOE AIoT CTO
[*]*Corresponding to: bluetornado@zju.edu.cn*

Abstract—**This technical report introduces a Multi-object tracking algorithm for sports scene, where the background is complicated, players possess rapid motion and the camera lens moves fast. Our solution finally achieve 76.264 HOTA in SportsMOT Challenge at ECCV 2022. The source code and the pre-trained models are available at https://github.com/vghost2008/sportstrack.**

Index Terms—**Mutli-object tracking, Tracking-by-detection, SportsMOT**

1. Introduction

Our algorithm is an online algorithms, similar to byte-track[1] and Bot-SORT[2], we use the tracking-by-detection paradigm. In order to adapt to the sports scene, we propose a new highly robust tracker, SportsTrack, the main contributions of our work can be summarized as follows:

1) Since targets in motion scenes often come out with motion blur, the confidence value of motion blurred target is usually low, and if we directly use a matching method similar to the byte-track[1] by high and low confidence bboxes in order, we may encounter the situation which blurred tracking objects matched with incorrectly detected high-confidence bbox, so we introduce a three-stage matching strategy, first using all detection targets to match with the tracking objects, but using a stricter threshold, followed by matching using high-confidence detection bboxes, and finally matching using low-confidence detection bboxes.
2) Since motion scenes often have extreme occlusion, in this case, the detection model cannot detect all objects. To handle this situation, we introduce the 'crowded target', a 'crowded target' should be determined by calculating IOU between tracks. if it is a crowded target, we allow one-to-many correspondence between crowded tracking objects and detection objects, while non-crowded tracking objects, similar to other trackers[1], [2], only allow one-to-one correspondence.
3) For the lost objects we classify them into two categories: lost in the center area of the image and lost at the edge of the image, for the lost objects in the center area of the image we use a similar processing strategy to byte-track[1], etc. For the lost objects at the edge of the image, we no longer update their state by Kalman filter, while for the new tracked objects (e.g., track length less than 10) appear from the edge of the image and are in approximately the same orientation as an edge-lost tracking object, then we compute its ReID distance, and if its ReID distance is less than the specified threshold, we consider them to be actually the same tracking object.
4) To avoid generating incorrect new tracking targets, a detection target can only be a new tracking target if satisfied: 1. high confidence 2. does not match any tracking target 3. IOU value with other matched detection targets less than a specific threshold (e.g. 0.5).

2. Method

2.1. Algorithm processing flow

In this section, we present our four main improvements for the multi-object tracking. The pipeline of our algorithm is presented in Fig1. We describe our algorithmic processing flow in detail below.

1) Initialize the trace list to be empty.
2) The following processing is performed for each frame of the input image in turn.
    a) Detect all athletes in the image using the object detection model. In the postprocessing of object detection, all bboxes with confidence greater than 0.1 are retained, NMS (non-maximum suppression) processing is performed between targets, and duplicates are counted only when the IOU between targets is greater than a specified threshold (e.g., 0.7). Detection results are high recall and low precision, which are important for stable tracking, while we eliminate the adverse effects of incorrect detections on tracking by a subsequent progressive matching strategy.
    b) Calculate the ReID features for each detected bboxes.
    c) Predicting the motion state of all tracks using the Kalman filter, but excluding tracks lost at the edges.

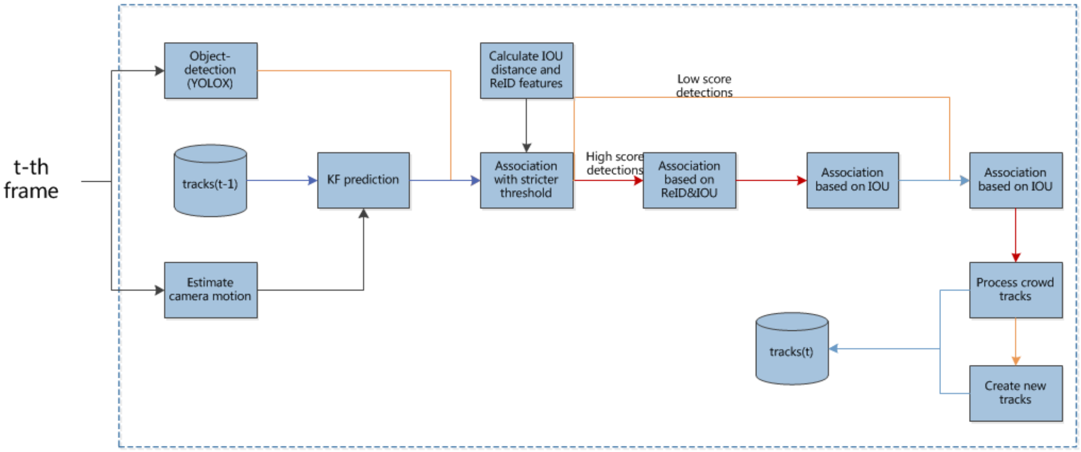

Figure 1. Overview of ours SportsTrack

d) Merging tracking sequences: If the length of the sequence of unlost tracks is less than 30 and it appears after the lost tracks, and the distance between its appearing angle (taking the image center as the origin) and the lost angle(taking the image center as the origin) of the lost track is less than 90 degrees, it is considered that it may be the same track, and for this case, the ReID distance of two sequences is calculated by taking the lost track the latest 60 ReID records (the actual length may be less than 60), take the latest 10 ReID records of the unlost track, calculate the ReID distance in pairs, count the number of ReID distance less than the specified threshold (e.g. 0.2), if the number is greater than 3, then the two tracking sequences are considered as the same sequence.

e) Calculate the IOU between any two unlost tracks, and if the IOU between the two targets is greater than 0.45, they are considered as crowded targets, and for crowded targets, find the detection target with the largest IOU, and if its IOU is greater than the specified threshold (e.g., 0.6), set the corresponding detection target as a candidate matching target for the crowded targets.

f) Calculate the IOU distance between the tracks and the detection target, assume we have M tracks and N detection targets, then the dimensional size of their IOU distance matrix D is $M \times N$.

g) Calculate the ReID distance between the tracks and the detection bboxes, and let the feature vector of the ith tracks be e_i, the ReID feature vector of the jth detection target is f_j, Then the distance matrix E of ReID is defined as

$$E_{ij} = e_i f_j^T$$

where T denotes the vector transpose operation.

h) Calculate the hybrid distance D1 based on the IOU distance and ReID distance:

$$D1 = \alpha D + (1 - \alpha) E$$

where α is 0.9.

i) Using the Hungarian algorithm matching tracks and detection bboxes with a matching threshold of 0.05 and D1 as the loss.

j) For matched tracks-detection targets pairs, update the Kalman filter state of the tracks by the corresponding detection bboxes. For unmatched detection targets, they are divided into two groups of high and low confidence using a specified threshold (e.g., 0.6) according to their detection confidence value.

k) A new hybrid distance D2 is calculated using the unmatched tracks and the high-confidence detection targets using their IOU distance D^H and ReID distance E^H:

$$D2 = (1 - \alpha) D^H + \alpha E^H$$

l) Using the Hungarian algorithm matching tracks and detection bboxes with a matching threshold of 0.3 and D2 as the loss.

m) For the tracks-detection target pairs matched in the previous step, update the Kalman filter state of the tracks by the corresponding detection target.

n) Further use the unmatched tracking target and the unmatched high confidence detection target to calculated a new hybrid distance D3 using its IOU distance D^{H1} and ReID distance E^{H1}:

$$D3 = \alpha D^{H1} + (1 - \alpha)E^{H1}$$

o) Using the Hungarian algorithm matching tracks and detection targets with a matching threshold of 0.7 and D3 as the loss.

p) For the tracks-detection target pair matched in the last step, update the kalman filter state of the tracks by the corresponding detection bboxes.

q) A new hybrid distance D4 is calculated using the unmatched tracking target and the low confidence detection target using its IOU distance D^L and ReID distance E^L:

$$D4 = \alpha D^L + (1 - \alpha)E^L$$

r) Using the Hungarian algorithm matching tracks and detection bboxes with a matching threshold of 0.7 and D4 as the loss.

s) For the tracks-detection target pair matched in the previous step, update the kalman filter state of the tracks by the corresponding detection target. For an unmatched tracking target, check its tracking length, and if its starting tracking frame number is not the first frame and its length is one, it is considered as a mistracked object.

t) For other unmatched tracks, check if satisified: 1.they are 'crowded target' 2.candidate matching object bboxes are settled, and if both conditions are satisfied, update these unmatched tracking targets with the corresponding candidate matching objects and set them to tracking status.

u) The remaining unmatched tracks, if its state in the last frame is the tracking state then calculate its missing position, let the image width be W, height be H, let boundary width $b = 60$, if the traking object is lost at image's center area $(b, b, W - b, H - b)$, then it is considered lost in the center area of the image, otherwise it is considered lost at the edge area of the image, for the tracks lost in edge area calculate its missing angle, the calculation formula is:

$$A = atan2(y - \frac{H}{2}, x - \frac{W}{2})$$

where x, y are the coordinates of the tracks' centroid.

v) For lost tracks, check the length of time that they have been lost, and if a tracking target has been lost for more than 120 consecutive frames, it is considered permanently lost and is removed from the tracking list.

w) For the unmatched high-confidence detection bboxes in the previous step, use NMS process these detections, the processing threshold is set to 0.45, and next the IOU between the remained high-confidence detections and matched detection targets in previous steps is greater than 0.45 is removed.

x) The remaining high-confidence detection targets after the previous processing step is added to the tracking list as a new tracking target.

In the actual implementation, we use a mixture of key point distance and IOU distance, and the optimal value of each threshold is searched on the sportsmot val dataset by a random search algorithm.

2.2. Training Strategies

In the algorithm, there are 4 models need to be trained using the dataset.

1) Object detection model.
2) The ReID model for main algorithm (feature dim size is 2048).
3) keypoints detection model.
4) The ReID model for post-processing algorithm (feature dim size is 512).

The training methods of the corresponding models are described as follows.

2.2.1. Object detection model. We use YOLOX[3] as our detection model, we use the yolox-x configuration, the image input size is 1440×800, we use the official weight trained on COCO as pre-training weight, we only train the head, the backbone and neck are frozen during training, the training data are sportsmot train and val dataset, training epochs=50, batch size=40, the learning rate strategy is yoloxwarmcos, the initial learning rate is $\frac{0.01}{64}$, the optimizer is SGD, momentum=0.9.

2.2.2. ReID model for main algorithm. We use Fast-ReID[4] as our ReID model, of which we use the sbs_S50 configuration, with an image input size 128×384, we use the official pre-trained weights which is trained on imagenet, training data are sportsmot train and val dataset, total training 38 epochs, batch size=160, learning rate strategy is CosineAnnealingLR, the initial learning rate is 0.00035, the optimizer is Adam, momentum=0.9.

2.2.3. Keypoints detection model. We use hrnet as our keypoints detecter[5], We use the configuration pose_hrnet_w48. We use the official pretained weight which was trained on COCO.

2.2.4. ReID model for post-process. The post-process ReID model uses deep-person-ReID[6], [7], [8], of which we use the osnet_ain_1x0 configuration, with an image input size of 128×256, we use the official pre- trained weights

TABLE 1. RESULTS ON SPORTSMOT CHALLENGE

Method	HOTA	AssA	DetA	MOTA
SportsTrack	76.264	73.538	79.180	89.316

which is trained on imagenet, training data are sportsmot train and val dataset and dukemtmcreid, total training 300 epochs, batch size=256, learning rate strategy is CosineAnnealingLR, initial learning rate is 0.0003, using random_flip and random_erase transforms, the optimizer is Adam, momentum=0.9, beta1=0.9, beta2=0.99. fixebase_epoch is 50, and total train 50 epochs, open_layers=classifier; loss function is softmax with label_smooth.

2.3. Post-processing

Our main algorithm is **ONLINE** algorithm, and the post-processing is **OFFLINE** algorithm. The post-processing process is summarized as follows.

1) For all tracking objects, the tracking quality is evaluated by calculating the ReID features of the object at each frame, and then calculating the average ReID characteristics of the object, and if the variance of the distance between the object's ReID and its average ReID is greater than a specified threshold, such as 0.2, it is considered as a less stable tracking object, otherwise it is considered as a stable tracking object.
2) For two stable tracking objects sequences, calculate pairwise values of ReID distances of two sequences, Count the number of value which less than the specified threshold (e.g., 0.3). If the number is greater than half of length of the pairwise value results, they are considered to be the same tracking object sequence and are merged, e.g., if the length of tracks object a is M and the length of tracks object b is N, then there are $M \times N$ ReID distances between them, if the number of ReID distance less than the specified threshold is more than $\frac{M \times N}{2}$, then they are considered as the same tracking object.
3) Interpolate the trace results and remove some too short trace sequences.

2.4. Results

The main results is shown in Table1

References

[1] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," 2022.

[2] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "Bot-sort: Robust associations multi-pedestrian tracking," *arXiv preprint arXiv:2206.14651*, 2022.

[3] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[4] L. He, X. Liao, W. Liu, X. Liu, P. Cheng, and T. Mei, "Fastreid: A pytorch toolbox for general instance re-identification," *arXiv preprint arXiv:2006.02631*, 2020.

[5] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," 2019.

[6] K. Zhou and T. Xiang, "Torchreid: A library for deep learning person re-identification in pytorch," *arXiv preprint arXiv:1910.10093*, 2019.

[7] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," in *ICCV*, 2019.

[8] K. Zhou, Y. Yang, A. Cavallaro, and Xiang, "Learning generalisable omni-scale representations for person re-identification," 2021.