A METHOD FOR JOINT DETECTION AND RE-IDENTIFICATION IN MULTI-OBJECT TRACKING

L. Huang, X. Shi*, J. Xiang**

Abstract: In order to better balance the detection accuracy and tracking speed, we propose an online balanced multi-object tracking method (BalMOT), which integrates object detection and appearance extraction into a single network, and can simultaneously output detection and appearance embedding. We also model the training of classification, regression, and embedding features as a multi-task training problem and each part is weighted based on the task-independent uncertainty method. In addition, we introduce the transition layer to optimize the repeated gradient information in the network and reduce the training cost. Through the training, our BalMOT system reaches 71.9 % multiple object tracking accuracy (MOTA) on the MOT17 challenge dataset, and the speed fluctuates between 17.4 $\sim$ 22.3 frames per second (FPS) according to the size of the input image.

Key words: *multi-object tracking, anchor-based, joint-learning, real-time*

Received: October 5, 2021 **DOI:** 10.14311/NNW.2022.32.017
Revised and accepted: December 31, 2022

1. Introduction

Multi-object tracking (MOT), a system to predict and track the motion trajectories of multiple targets in a video sequence, which has critical research significance in many fields, i.e., transport, smart cities, and national security. The mainstream idea of MOT research is divided into two steps: 1) the target detection step, the output of which is the input of the tracking module; 2) object tracking step, which includes target association and matching. The essence of tracking is to correlate the same target in the connected frames of the video and assign a unique ID value. The existing MOT methods such as [27, 32, 38, 37] are all composed of two independent models: the detection model first detects the target and locate multiple objects in each frame, then the association model extracts features based on the re-identification (re-id) domain model. However, the current MOT field has gradually abandoned this paradigm to balance tracking speed and accuracy. In order to save computation and optimize the network structure, the method combining detection with embedding/matching can be adopted, as shown in Fig. 1.

*Lilian Huang; Xu Shi; Jianhong Xiang – Corresponding author; Harbin Engineering University, College of Information and Communication Engineering, No. 145 Nantong Street, Nangang District, Harbin, China, 150001, b MIIT Key Laboratory of Advanced Marine Communication and Information Technology, E-mail: lilian_huang@163.com, shixuxz@hrbeu.edu.cn, xiangjianhong@hrbeu.edu.cn

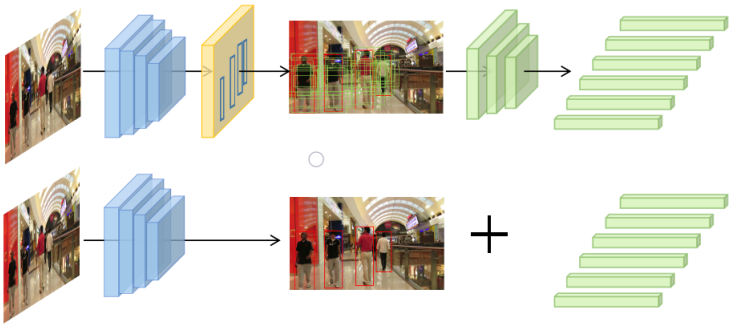


Fig. 1 *Comparison between the two-stage model and the BalMOT.*

In [33], the framework of the joint detector and embedding (JDE) learning is formally proposed for the first time. The architecture adopts feature pyramid network (FPN) [17] to adapt to multi-scale detection. The detection part uses an improved region proposal network (RPN) [28] network and introduces embedding to achieve joint learning. Although it has achieved good results, much remains to be improved. Initially the loss of feature information. Although FPN adopts the top-down feature propagation mode, and uses the strong semantic information of high-level features to improve the low-level features, the path from low-level structure to top-level features is very long, which makes it more difficult to obtain accurate positioning signals. Therefore, it is necessary to find a more appropriate framework for the integration of different feature layers. Low-level features are utilized in the system of [19, 18, 6, 10], but not propagated to enhance the whole feature level.

Furthermore, in the previous study of re-id problem, the method of learning high-dimensional features is usually adopted to obtain pedestrian appearance features. However, this method relies on a large datasets, while the relevant re-id datasets only provides cropped figure images, which is not conducive to the study of MOT.

In addition, many algorithms commonly use the method of deepening or widening the model to improve the ability of the model to extract fine-grained features, but it also brings a great test to the computing power of the machine and the real-time performance of multi-object tracking.

This paper introduces a balanced algorithm BalMOT, which enables the network to output the position of the bounding box and the embedding features of the objects in the box concurrently. Our approach improve the accuracy while ensuring the speed. Fig. 1 briefly illustrates the difference between the two-stage approach and our approach. The separation of detection module and embedding module lead to the poor alignment and much redundancy of the models. By sharing the same network, our algorithm can reduce the complexity of the model and further increase the accuracy of the MOT by promoting the fusion of sub-modules. Compared with the Faster Region-CNN+Embedding model in the person of interest (POI) algorithm, the optimized model achieves a running time of 18.16 FPS on the MOT16 test set, while the former only achieves <6 FPS. Meanwhile, our model

achieves MOTA = 71.9 % on the MOT17 validation set, which is 3.2 % higher than the MOTA = 68.7 % of QuasiDense [25] algorithm.

In order to build a more accurate real-time MOT system, our research is carried out from several aspects of dataset selection, network architecture, learning objectives, optimization strategies, and evaluation metrics. Firstly, in terms of data selection, through pre-training, we determine three datasets, i.e., MOT17, CUHK-SYSU, and PRW, to train our model. Secondly, we choose FPN as the infrastructure and discuss the influence of different networks on MOT performance. Then, we model the training process as anchor classification, bounding box regression and embedding problems of multi-task learning. We also adopt the task-independent uncertainty [15] to balance the weight of each task. Finally, we use the following public standards to evaluate the performance of the MOT. Clear [2] metric is adopted to evaluate the overall performance of the MOT system. The average precision (AP) is employed to evaluate the performance of the detector. The true positive rate (TPR) is to appraise the quality of the embedding. Our method is as fast as the joint detection and embedding (JDE), and our MOTA = 66.8 %, 18.16 FPS on the same MOT16 test.

The contribution of this paper can be summarized into the following aspects:

- We propose a balanced algorithm BalMOT, which makes the detection module and the embedding module share the same network. Through the fusion of sub modules, the algorithm improves the tracking accuracy while ensuring the speed.
- We introduce a bottom-up path aggregation network (bPaNet) into the network to enhance the feature level, and introduce a transition layer to reduce the extra computation caused by model deepening.
- Experiments have confirmed that the use of low-dimensional re-id features can better reduce the risk of overfitting caused by small data sets and improve the robustness of multi-target tracking models.
- Experiments on MOT16 and MOT17 demonstrate the advantage of our method over state-of-the-art MOT systems considering accuracy and speed.

2. Related work

With the development of deep learning technology, the application related to visual multi-object tracking has made enormous strides. The framework of MOT based on deep learning mostly presents two modes:

2.1 The tracking-by-detection model

With the continuous improvement of CNN model, deep learning has made rapid development in image classification task. A number of excellent open source deep neural networks, such as R-CNN detectors, SSD [18] and YOLO detectors [27, 5], have greatly enhanced the ability of object detection. Due to the enhancement of single frame image detection ability, MOT presents a trend from the initial

data association optimization algorithms with complex computation, such as joint probabilistic data association (JPDA) [16] and multiple hypothesis tracking (MHT) [4], to the DBT model which depends on the detection results. SORT [3] is one of the earliest MOT algorithms using CNN to detect pedestrians. Based on the traditional Hungarian algorithm, SORT replaces ACF detection with Fast R-CNN [11], which improves the accuracy of multi-object tracking by 18.9 %. Deepsort further extracts stable apparent features on the basis of SORT, which improves the performance of the algorithm under object occlusion and greatly reduces the speed of the algorithm. Yu et al. [37] proposed the POI algorithm and improved the Faster R-CNN by de-pooling and multi-scale feature extraction technology, so as to further improve the tracking accuracy and speed. Lu et al. [20] proposed the associative long short term memory networks (LSTM), which uses SSD detector to directly regress the position and category of the target, and generate association features to solve the problem that traditional LSTM cannot fundamentally simulate the association of objects between consecutive frames. Henschel et al. [13] proposed a off-line MOT model, which added a head detection to increase the accuracy under the condition that most of the human body is covered. Bergmann et al. [1] proposed Tracktor, which predicts the object position of the next frame through the regression head of the detector. However, the models in these tracking-by-detection (TBD) algorithms are complex and adopt multiple sub-modules, which ignoring the sharing information between modules.

2.2 The joint detection and tracking model

The joint detection and tracking algorithm emerged in recent two years not only reduces the complexity of TBD model, but also improves the accuracy of MOT. Its strategy is to integrate some functional modules in TBD model to reduce the extra inference time generated by phased processing and to increase the coupling degree between functional modules. Christoph et al. [9] first tried to improve the object detection network by adding tracking branch. Then, base on the region-based fully convolutional network (R-FCN) [7], the MOT task is transformed into the matching problem of the relative offset of the target position in two adjacent frames, which effectively improved the accuracy and speed of visual MOT. Nevertheless, it is still a two-stage MOT algorithm. In order to further integrate the tracking module, Bergmann et al. proposed a new joint detection tracking model, called Tracktor++, which uses a simple and light weight data association algorithm to match the bounding box and the prediction box, meanwhile, a deep neural network is used to generate the results of whole tracking sequence. Through fusion, the detection module has a greater impact on the MOT performance. Afterwards, inspired by Tracktor++, Huang et al. [14] further improved the motion model, the apparent model and the data association part to boost performance of Tracktor++. However, these model have limitations: low degree of fusion between functional modules. To address the problem, we propose a simple online multi-object tracking fusion algorithm, which can output the detection and appearance embedding simultaneously, and further improve the detection and tracking accuracy while ensuring the real-time running speed.

3. Technical methods

3.1 Backbone network

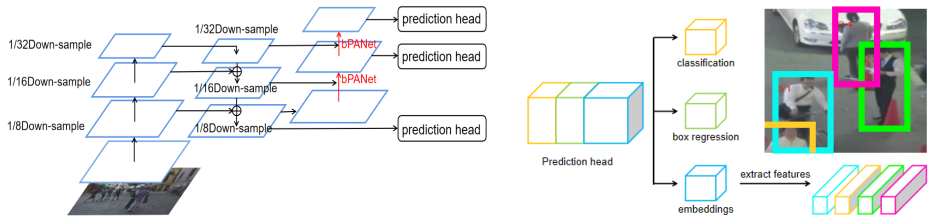
We use DarkNet53 [32] as our backbone to balance accuracy and speed. The size of the convolutional kernel in front of each cross stage partial (CSP) connection is 3×3, the stride $= 2$, which means a down-sampling operation. The output feature maps of three sizes respectively connected to a prediction head, as shown in the Fig. 2. Each prediction head consists of several stacked convolutional layers and outputs a dense prediction map. Denote the size of input image as $H_{\text{image}} \times W_{\text{image}}$, then output three feature maps with $\frac{1}{32}$, $\frac{1}{16}$, $\frac{1}{8}$ down-sampling rate respectively. The dense prediction map is divided into three parts:

a) the classification prediction map of size $2A \times H_{\text{image}} \times W_{\text{image}}$,

b) the bounding box regression coefficients of size $4A \times H_{\text{image}} \times W_{\text{image}}$,

c) the embedding feature prediction map of size $D \times H_{\text{image}} \times W_{\text{image}}$,

where A is the number of anchor templates allocated to each ratio, and D is the dimension of the embedding vector. Also, we add a bottom-up path (bPaNet), as shown in Fig. 3, to shorten the information path and use the precise positioning signals existing in the low-level to enhance the feature pyramid.



(a) Backbone overview (b) Prediction head

Fig. 2 *Overview of our one-shot tracker.(a) is the object detection part of the network architecture, "⊕" indicates concatenation, (b) is the prediction head added upon the bottom-up path augmentation.*

Aiming at the problem of multi-scale image detection, cropping or stretching methods are often used to meet the resolution requirements of the input image. However, this approach will change the size and aspect ratio of the input image, causing the original image to be distorted. Therefore, we connect an spatial pyramid pooling (SPP) [12] layer to the last convolutional layer to effectively avoid problems such as image distortion caused by image area clipping and scaling operations. For input images of different sizes and aspect ratios, SPP pool arbitrary size feature maps into fixed-size feature vectors. We set four pooling layers with the size of $\{1\times1; 5\times5; 9\times9; 13\times13\}$, and stride equals to 1. It will output four feature maps with the same size, thus improving the scale invariance of the image and reducing overfitting.

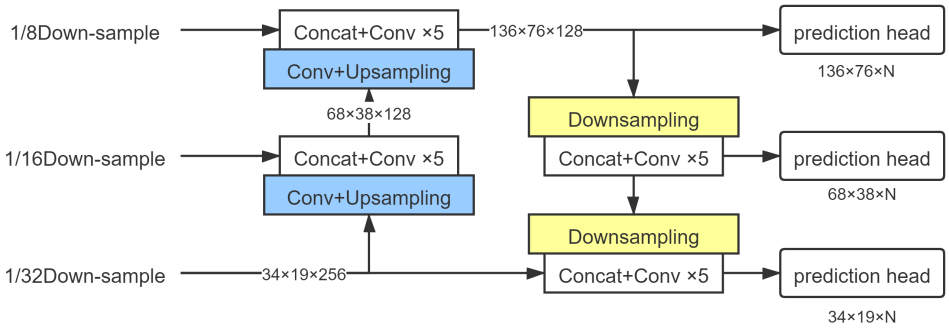
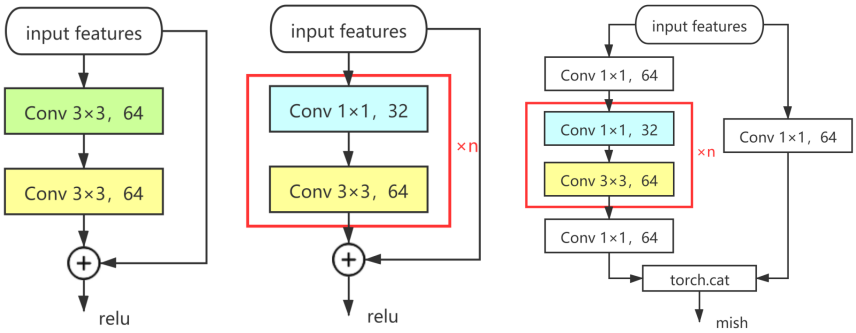


Fig. 3 *bPaNet.*

In order to solve the problem of increasing training time and decreasing tracking efficiency caused by the deepening of network layers, we introduce a transition layer after the residual block. As shown in the Fig. 4, the conventional residual block is composed of two convolution layers stacked, but this model introduces too many parameters, which is only suitable for small networks. In DarkNet53, the parameters of the residual block is modified, which is presented in Fig. 4(b). The computation is greatly reduced through a convolution with the size of 1×1 and the number of 32 channels. On the basis of (b), our model added a transition layer, whose input features are firstly transformed through two-channel 1×1 convolution, so as to improve the feature reusability and halve the number of channels to reduce the computation. Then, the balance between detection and speed is realized through the residual block.



(a) PRB (b) IRB (c) RBTL

Fig. 4 *The optimized residual block, which solves the problem of gradient disappearance and a mass of parameters caused by network deepening. "PRB" is the abbreviation for plain residual block, "IRB" is for the improved residual block, "RBTL" is for the residual block with transition layer.*

3.2 Detection branch

Our detection branch adopts a similar structure to RPN, but with two differences. First, we redesign the anchors in terms of number, proportion, and aspect ratio so that they could accommodate various targets (pedestrians). As a rule of thumb, all anchors are set to an aspect ratio of 1:3. The number of anchor templates is set to 12, corresponding to A = 4 for each scale, and the size of anchors range from $8 \times 2^{\frac{1}{2}} \approx 11$ to $8 \times 2^{\frac{12}{2}} \approx 512$. Secondly, in the real-time detection of moving targets, it is of great significance to use dual thresholds to distinguish foreground/background and reduce the misseing detection. By visualization, we determine the intersection of union (IoU) > 0.5, at which time the ground truth is approximately regarded as foreground, and this value is also consistent with the common settings of general object detection. Furthermore, $IoU < 0.4$ must be satisfied, in which case the ground truth is considered as background. The false alarm under severe occlusion can be effectively suppressed by dual-threshold method.

Object detection includes two tasks: classification and regression, where the classification loss function adopts the cross-entropy loss and the bounding box regression loss function employs the smooth-L1 loss. The coding method of the regression target is the same as [28].

3.3 Embedding branch

The purpose of embedding branch is to capture the features of different targets. Ideally, the metric distance of the same target between any frames in a video is less than different targets, indicated as $d(y_{n_t}, y_{n_{t+\triangle t}}) < d(y_{n_t}, y_{n'_{t+\triangle t}}), \forall (y_{n_t}, y_{n_{t+\triangle t}}, y_{n'_{t+\triangle t}}) \in y(x)$, where $y(x) \in R^{n \times D}$ is the corresponding embedding vector extracted from the dense embedding map, n is the number of targets in a single frame of video, and D is the demension of the embedding vector. To achieve this goal, triplet loss [29] can be used.

$$L_{\text{triplet}} = \sum_i \max[\|y(x_i^{\text{a}}) - y(x_i^{\text{p}})\|_2^2 - \|y(x_i^{\text{a}}) - y(x_i^{\text{n}})\|_2^2 + \alpha, 0], \tag{1}$$

where x_i^{a} represents the input anchor data, x_i^{p} represents the positive samples of the same category as x_i^{a}, x_i^{n} denotes the negative samples of the different category, and α is a margin item. For the convenience of research, the margin is ignored here.

Although the triplet loss is widely used, it still has the slow convergence speed problem, which needs to adopt the difficult sample mining method to accelerate the training speed. Therefore, we apply the tuplet loss proposed by [30] to the embedding model.

$$\begin{aligned} L_{\text{tuplet}} &= \log\left(1 + \sum_i \exp(y(x_i^{\text{a}})y(x_i^{\text{p}}) - y(x_i^{\text{a}})y(x_i^{\text{n}}))\right) \\ &= -\log \frac{\exp(y(x_i^{\text{a}})y(x_i^{\text{n}}))}{\exp(y(x_i^{\text{a}})y(x_i^{\text{n}})) + \sum_i \exp(y(x_i^{\text{a}})y(x_i^{\text{p}}))}. \end{aligned} \tag{2}$$

The above equation is similar to the cross-entropy loss,

$$L_{\text{tuplet}} = -\log \frac{\exp(y(x_i^{\text{a}})w(x_i^{\text{n}}))}{\exp(y(x_i^{\text{a}})w(x_i^{\text{n}})) + \sum_i \exp(y(x_i^{\text{a}})w(x_i^{\text{p}}))}, \tag{3}$$

where $w(x_i^{\text{p}})$ is the class-wise weight of positive samples, and $w(x_i^{\text{n}})$ is the weight of negative samples. Therefore, we employ the cross-entropy loss to calculate the embedding loss.

3.4 Joint training

In the object detection task, the learning objective of each prediction head can be modeled as a multi-task learning problem. The joint target can be written as a linear weighted form of different loss weights and individual component losses,

$$L_{\text{total}} = \sum_{i}^{M} \sum_{j=\alpha,\beta,\gamma} \omega_j^i L_j^i, \tag{4}$$

where M is the number of prediction heads, and $\omega_j^i, i = 1, \ldots, M, j = \alpha, \beta, \gamma$ are loss weights.

Nevertheless, defining joint objectives in such a simple way would make the setting of the loss weights too rigid. The fixed weight will along with the whole training period and limit the learning of the task. As a result, in order to achieve the optimal match as far as possible, we choose the automatic learning scheme proposed in [15], where the learning objective can be written as:

$$\begin{aligned} L_{\text{total}} =& \sum_{i}^{M} \sum_{j=\alpha,\beta,\gamma} \frac{1}{2}\left(\frac{1}{e^{s_j^i}} L_j^i + s_j^i\right) \\ =& \frac{1}{2}\left(\frac{1}{e^{s^1_{\text{regression}}}} L^1_{\text{regression}} + \frac{1}{e^{s^2_{\text{classification}}}} L^2_{\text{classification}} + \frac{1}{e^{s^3_{\text{identity}}}} L^3_{\text{identity}} \right. \\ & \left. + s^1_{\text{regression}} + s^2_{\text{classification}} + s^3_{\text{identity}}\right), \end{aligned} \tag{5}$$

where s_j^i refers to the task-independent uncertainty of each loss function, which can be obtained through learning.

3.5 Online association

We follow the standard online tracking association strategy to associate boxes. Firstly, we recognize the position of the detection and corresponding appearance embedding in a frame of video. The detection boxes are filtered according to the confidence and non-maximum suppression (NMS). After that, we use the Kalman filter [34] to predict the mean value and covariance of the target in the current frame position. Then the target is matched with the existing trajectory according to the predicted results. Whether the targets are the same is determined by the similarity of appearance features and the detected target position between adjacent frames. We also update the appearance features for each matched target to improve model accuracy and robustness.

4. Experiment

4.1 Datasets and evaluation metrics

In our experiment, we use three training sets: PRW [38], CUHK-SYSU(CS) [36], and MOT17 [22]. The PRW released the full frames with annotations. The CS has detailed information of both the bounding box and the identity annotation, including 11k pictures, 55k ground truth, and 7k identities. MOT17 contains 7 training sequences, each of which provides three detection methods: DPM, Faster R-CNN, and SDP.

In order to evaluate the performance of the algorithm, we need to consider three aspects of performance: detection accuracy, embedding discrimination ability, and tracking performance of the entire MOT system. To evaluate the detection performance, we calculate average precision (AP) with an IoU threshold of 0.5 on the MOT17 validation set, and true positive rate (TPR) at a false accept rate of 0.1 for rigorously evaluating embedding features. We use the CLEAR metrics to evaluate the tracking performance of MOT, including tracking accuracy (MOTA), the total number of identity switches (IDs), the percentage of tracks tracked with more than 80 % of targets (MT), and the percentage of tracks lost with less than 20 % targets (ML). In addition, the ID F1 score (IDF1) is also used to measure the accuracy of trajectory recognition. In order to verify the robustness of our algorithm, we also compare the model under the "private detector" protocol of the MOT16 and MOT17 benchmarks.

4.2 Implementation details

We employ DarkNet53 as the backbone, and the network is trained with the standard stochastic gradient descent (SGD) method for 30 epochs. The learning rate is initialized to 10^{-2} and decays to 10^{-3} at the 15th and 10^{-4} at the 23rd epochs. The batch size is set to 16. The training step takes about 12 hours on eight GTX 1080 Ti GPUs. The testing and demo runs on a RTX 3080 GPU.

4.3 Experimental results

Performance analysis

On the basis of DarkNet53, we use the standard residual block and the residual block with transition layer respectively. Meanwhile, we analyze the network parameters and inference speed of the model, the results are shown in Tab. I. The time cost is measured for 1088×608 images using single RTX 3080 GPU and cuDNN v8 with Intel i7 10700KF@3.80 GHz. When the transition layer is introduced into the standard residual block, the accuracy of detection and tracking changes little, but the inference time cost is reduced by 10.77 ms. We can conclude that the model after introducing the transition layer can not only guarantee the accuracy but also avoid learning repeated gradient information, reduce the calculation amount and improve the running speed. Besides, considering the influence of the backbone activation function, we select a smoother Mish [23] function to further improve the accuracy and generalization of the model. From Tab. I, the network with Mish

activation function showed a better result, MOTA is increased by 1.1 %, IDF 1 is increased by 2.4 %, ID switch number is reduced by 232, even more to the point, these improvements only cost an extra 8.78 ms on inference time.

Method	MOTA↑ [%]	IDF1↑ [%]	IDs↓	AP↑ [%]	Time cost (ms)↓
Darknet53-ResB-Relu	68.8	*63.9*	*1452*	*82.18*	56.47
Darknet53-ResB+TL-Relu	*69.4*	61.3	1662	**82.98**	**45.70**
Darknet53-ResB+TL-Mish	**70.6**	**66.4**	**1220**	81.40	*54.48*

Tab. I *Ablation study on the validation set of MOT17.↑ means the larger the better, ↓ means the smaller the better. In each column, the optimal value is in bold, and the sub-optimal is in italic. "ResB" represents the residual block, "TL" is short for the transition layer.*

We also use DarkNet53 (it only covers the backbone and three prediction heads without any neck) as the baseline to discuss the effectiveness of different modules of the network. The results are shown in Tab. II. We can see that using SPP alone cannot improve the detection accuracy and MOTA. On the contrary, it reduces AP and MOTA by 0.7 % and 0.8 % respectively. The best result is obtained when bPANet and SPP are both introduced to the backbone, they combined to enlarge the receptive field and enhance the feature fusion capability. Ultimately, we get 71.5 % MOTA and 85.31 % AP. However, with the deepening of the network, FPS has been reduced to varying degrees.

Method	MOTA↑ [%]	IDF1↑ [%]	FP↓	FN↓	IDs↓	AP↑ [%]	FPS↑
Baseline	*70.6*	*66.4*	**3728**	28113	**1220**	*81.40*	**18.73**
Baseline+SPP	69.4	65.0	*4956*	*28103*	*1271*	80.70	*17.88*
Baseline+SPP+bPANet	**71.5**	**66.4**	5459	**25115**	1385	**85.31**	17.11

Tab. II *Ablation study on the validation set of MOT17. ↑ means the larger the better, ↓ means the smaller the better. In each column, the optimal value is in bold, and the sub-optimal is in italic.*

Re-id embedding feature dimension

As a rule of thumb, most of the previous pedestrian appearance embedding vector dimensions are set to 512. However, this is not necessarily applicable to all networks. To prevent over-fitting, the high dimensional re-id features need huge amounts of datasets to obtain fine-grained features. Due to the limitation of multi-object tracking dataset, we try to reduce over-fitting by using the smaller feature dimension of re-id to adapt to our network. From Tab. III, we can know that the impact of re-id feature dimension on the quality of embedding feature extraction is not monotonous. Generally, for the model trained by 512 dimensional embedding features, the TPR is 80.69 %. When the dimension is reduced to 256, the TPR decreases by 1.54 %. When the dimension decreases to 128, the TPR increases by 5.24 % to 85.93 %. In addition, the change of dimension has little impact on

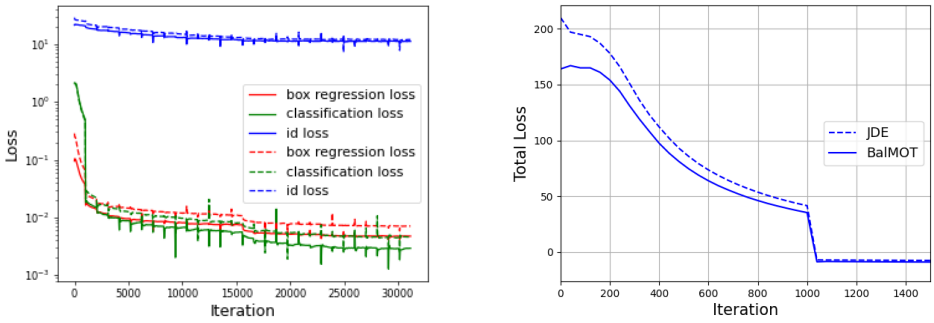
MOTA. The MOTA of 256 dimension is 0.3 % lower than that of 512 dimension, and the MOTA of 128 dimension is 0.3 % higher than 512 dimension. However, the reduction of dimension makes the number of ID switch increase, and the robustness of the model to object occlusion decrease.

Backbone	dim	MOTA↑ [%]	IDF1↑ [%]	IDs↓	FPS↑	TPR↑ [%]
DarkNet53	512	*70.6*	**66.4**	**1220**	18.73	*80.69*
DarkNet53	256	70.3	*66.4*	*1267*	*19.31*	79.15
DarkNet53	128	**70.9**	65.6	1395	**19.47**	**85.93**

Tab. III *Ablation study on the validation set of MOT17.↑ means the larger the better, ↓ means the smaller the better. In each column, the optimal value is in* ***bold****, and the sub-optimal is in italic.*

Comparison of the improved algorithm on the result of the loss

Fig. 5(a) shows the curves of the optimized algorithm and JDE in bounding box regression loss, classification loss, and ID loss. The dotted line represents the loss of each part under the JDE method, while the solid line denotes the improved algorithm. It can be seen that both the regression loss and the classification loss have significantly decreased, while the ID loss has little change. Fig. 5(b) is the total loss results. By optimizing the detection model, the total loss is obviously reduced, and the convergence is basically synchronized with JDE. The result of ours tends to be stable around batch size = 25.



(a) (b)

Fig. 5 *Comparisons results of different loss functions between our algorithm and JDE algorithm. (a) is the comparison chart of loss function between JDE-1088 and optimized algorithm, (b) is the comparison chart of total loss function between JDE-1088 and optimized algorithm.*

The comparison result of proposed BalMOT algorithm and JDE method

Our algorithm aims to use multi-layer feature fusion to solve the problem of feature imbalance. We compare our MOT system to JDE. For a fair comparison, we use the same embedding strategy and loss functions as JDE, introduce the optimized network to compare and analyze the performance of our MOT system under three datasets. In Tab. IV, we can get that the detection accuracy and the quality of embedding features are both improved. BalMOT-864 increases 6.1 % compared with JDE-864 on MOTA, and IDF1 increases 7.1 %. The higher the image resolution, the better our algorithm. When the input image is resized to 1088×608, our MOTA increases 7.1 % compared with JDE-1088 and IDF1 increases 10.6 %. Due to the deepening of network layers and the increase of computational complexity, the FPS decreases to some extent, but it still meets the real-time requirements.

Method	MOTA↑ [%]	IDF1↑ [%]	MT↑ [%]	ML↓ [%]	IDs↓	AP↑ [%]	TPR↑ [%]	FPS↑
JDE-864	62.1	56.9	34.4	16.7	1608	80.48	85.18	**24.1**
JDE-1088	64.4	55.8	35.4	20.0	*1544*	82.47	85.89	18.8
BalMOT-864	*68.2*	*64.0*	*47.6*	**7.5**	1581	*82.69*	**91.69**	*22.3*
BalMOT-1088	**71.9**	**67.8**	**55.1**	*7.7*	**1547**	**85.31**	*88.80*	17.40

Tab. IV *BalMOT vs JDE.Comparison of our improved algorithm and JDE-MOT on the validation set of MOT17.*

Comparison with the state-of-the-art MOT systems

Compare our algorithm with the state-of-the-art trackers under the "private data" protocol of the MOT16 and MOT17 benchmarks, respectively. The trackers consist of one-stage and two-stage. As can be seen in Tab. V, our method outperforms other trackers on any single test set, and achieves near real-time speed while improving

Dataset	Method	MOTA↑ [%]	IDF1↑ [%]	MT↑ [%]	ML↓ [%]	IDs↓	FPS↑
MOT16	SORTwHPD16 [3]	59.8	53.8	25.4	22.7	1423	<8.6
	DeepSORT_2 [35]	61.4	62.2	32.8	*18.2*	*781*	<8.1
	RAR16wVGG [8]	63.0	63.8	*39.9*	22.1	**482**	<1.5
	JDE [33]	64.4	55.8	35.4	20.0	1544	**18.8**
	TAP [39]	64.8	**73.5**	38.5	21.6	794	<8.2
	CNNMTT [21]	65.2	62.2	32.4	21.3	946	<6.4
	POI [37]	*66.1*	65.1	34.0	20.8	805	<6
	BalMOT(1088*608)	**66.8**	*65.9*	**42.2**	**9.5**	1516	*18.16*
MOT17	SST [31]	52.4	49.5	21.4	30.7	8431	6.3
	Tube_TK [24]	63.0	58.6	31.2	*19.9*	4137	3.0
	CTrackerV1 [26]	66.6	57.4	32.2	24.2	5529	6.8
	CSTrack	67.3	67.9	34.2	24.1	2994	16.9
	QuasiDense [25]	*68.7*	*66.3*	*40.6*	21.9	*3378*	**20.3**
	CSTrack++	70.6	71.6	37.5	18.7	3465	15.8
	BalMOT	**71.9**	**67.8**	**55.1**	**7.7**	**1547**	*17.40*

Tab. V *Comparison with the state-of-the-art online MOT systems under the private data protocol on the MOT16 and MOT17 benchmarks.*

tracking accuracy. Our results are all obtained under the condition that the input image size is 1088×608. The running speed is affected by the size of the input image. The higher the input image resolution, the slower the speed. Fig. 6 shows the tracking results of our algorithm on partial test sets of MOT16 and MOT17. MOT16 test results show that our model can still achieve good tracking results even when the pedestrian density is high, and the real-time running speed is inversely proportional to the number of pedestrian in a single frame. However, it is worth noting that although our model has a certain improvement compared with these trackers in the ID switching problem, the ID switching problem caused by a large area of target occlusion for a long time is still worth further study.

Fig. 6 *Examples of the performance of our algorithm on the test sets of MOT16 and MOT17. Both images are arranged by a chronological sequence of video frames, and different colors represent different IDs for tracking.*

5. Conclusion

In this paper, we propose a new online balanced multi-object tracking (BalMOT) algorithm that balances detection accuracy and tracking speed. We analyze the losing feature information and unreasonable re-id dimension in the joint detection and tracking framework, and further improve the detection accuracy and appearance feature extraction ability. Further, in order to reduce the extra computing cost caused by network deepening, we introduce a transition layer to solve a mass of inference computing problems. The proposed BalMOT is comparable to the state-of-the-art results. Our MOT system reaches 71.9 % MOTA on the MOT17 challenge dataset, and the speed fluctuates between $17.4 \sim 22.3$ FPS according to the size of the input image. Later, we will focus on ID switching, and try to employ self-supervised learning to enrich data information, so as to improve the learning ability of the system.

Disclosures

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the Fundamental Research Funds for the Central Universities (3072022CF0801) and the Joint Guidance Project of the Natural Science Foundation of Heilongjiang Province (Grant No. LH2020F022).

References

[1] BERGMANN P., MEINHARDT T., LEAL-TAIXE L. Tracking without bells and whistles. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 941–951.

[2] BERNARDIN K., STIEFELHAGEN R. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*. 2008, 2008, pp. 1–10.

[3] BEWLEY A., GE Z., OTT L., RAMOS F., UPCROFT B. *Simple online and realtime tracking In: ICIP*. 2016.

[4] BLACKMAN S.S. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*. 2004, 19(1), pp. 5–18.

[5] BOCHKOVSKIY A., WANG C.-Y., LIAO H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*. 2020.

[6] CAI Z., FAN Q., FERIS R.S., VASCONCELOS N. A unified multi-scale deep convolutional neural network for fast object detection. In: *European conference on computer vision*, 2016, pp. 354–370.

[7] DAI J., LI Y., HE K., SUN J. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*. 2016, 29.

[8] FANG K., XIANG Y., LI X., SAVARESE S. Recurrent autoregressive networks for online multi-object tracking. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 466–475.

[9] FEICHTENHOFER C., PINZ A., ZISSERMAN A. Detect to track and track to detect. In: *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3038–3046.

[10] FU C.-Y., LIU W., RANGA A., TYAGI A., BERG A.C. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*. 2017.

[11] GIRSHICK R. Fast r-cnn. In: *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[12] HE K., ZHANG X., REN S., SUN J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*. 2015, 37(9), pp. 1904–1916.

[13] HENSCHEL R., LEAL-TAIXÉ L., CREMERS D., ROSENHAHN B. Fusion of head and full-body detectors for multi-object tracking. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 1428–1437.

[14] HUANG P., HAN S., ZHAO J., LIU D., WANG H., YU E., KOT A.C. Refinements in motion and appearance for online multi-object tracking. *arXiv preprint arXiv:2003.07177*. 2020.

[15] KENDALL A., GAL Y., CIPOLLA R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7482–7491.

[16] KRISHNASWAMY S., VITULLO S., LAIDLER W., KUMAR M. Dynamic Joint Probabilistic Data Association Framework for Target Tracking with Ground Robots. In: *2020 American Control Conference (ACC)*, 2020, pp. 2076–2081.

[17] LIN T.-Y., DOLLÁR P., GIRSHICK R., HE K., HARIHARAN B., BELONGIE S. Feature pyramid networks for object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[18] LIU W., ANGUELOV D., ERHAN D., SZEGEDY C., REED S., FU C.-Y., BERG A.C. Ssd: Single shot multibox detector. In: *European conference on computer vision*, 2016, pp. 21–37.

[19] LONG J., SHELHAMER E., DARRELL T. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[20] LU Y., LU C., TANG C.-K. Online video object detection using association LSTM. In: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2344–2352.

[21] MAHMOUDI N., AHADI S.M., RAHMATI M. Multi-target tracking using CNN-based features: CNNMTT. *Multimedia Tools and Applications*. 2019, 78(6), pp. 7077–7096.

[22] MILAN A., LEAL-TAIXE L., REID I., ROTH S., SCHINDLER K. MOT16: a benchmark for multi-object tracking. arXiv e-prints. *arXiv preprint arXiv:1603.00831*. 2016.

[23] MISRA D. Mish: A self regularized non-monotonic neural activation function. *arXiv preprint arXiv:1908.08681*. 2019, 4(2), pp. 10–48550.

[24] PANG B., LI Y., ZHANG Y., LI M., LU C. Tubetk: Adopting tubes to track multi-object in a one-step training model. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6308–6318.

[25] PANG J., QIU L., LI X., CHEN H., LI Q., DARRELL T., YU F. Quasi-dense similarity learning for multiple object tracking. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 164–173.

[26] PENG J., WANG C., WAN F., WU Y., WANG Y., TAI Y., WANG C., LI J., HUANG F., FU Y. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In: *European conference on computer vision*, 2020, pp. 145–161.

[27] REDMON J., FARHADI A. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*. 2018.

[28] REN S., HE K., GIRSHICK R., SUN J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*. 2015, 28.

[29] SCHROFF F., KALENICHENKO D., PHILBIN J. Facenet: A unified embedding for face recognition and clustering. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[30] SOHN K. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*. 2016, 29.

[31] SUN S., AKHTAR N., SONG H., MIAN A., SHAH M. Deep affinity network for multiple object tracking. *IEEE transactions on pattern analysis and machine intelligence*. 2019, 43(1), pp. 104–119.

[32] WANG C.-Y., LIAO H.-Y.M., WU Y.-H., CHEN P.-Y., HSIEH J.-W., YEH I.-H. CSPNet: A new backbone that can enhance learning capability of CNN. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 390–391.

[33] WANG Z., ZHENG L., LIU Y., LI Y., WANG S. Towards real-time multi-object tracking. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, 2020, pp. 107–122.

[34] WELCH G., BISHOP G. An introduction to the Kalman filter. 1995.

[35] WOJKE N., BEWLEY A., PAULUS D. Simple online and realtime tracking with a deep association metric. In: *2017 IEEE international conference on image processing (ICIP)*, 2017, pp. 3645–3649.

[36] XIAO T., LI S., WANG B., LIN L., WANG X. Joint detection and identification feature learning for person search. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3415–3424.

[37] YU F., LI W., LI Q., LIU Y., SHI X., YAN J. Poi: Multiple object tracking with high performance detection and appearance feature. In: *European Conference on Computer Vision*, 2016, pp. 36–42.

[38] ZHENG L., ZHANG H., SUN S., CHANDRAKER M., YANG Y., TIAN Q. Person re-identification in the wild. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1367–1376.

[39] ZHOU Z., XING J., ZHANG M., HU W. Online multi-target tracking with tensor-based high-order graph matching. In: *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 1809–1814.