FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking

Yifu Zhang*, Chunyu Wang*, Xinggang Wang†, Wenjun Zeng, Wenyu Liu

Abstract—There has been remarkable progress on object detection and re-identification (re-ID) in recent years which are the key components of multi-object tracking. However, little attention has been focused on jointly accomplishing the two tasks in a single network. Our study shows that the previous attempts ended up with degraded accuracy mainly because the re-ID task is not fairly learned which causes many identity switches. The unfairness lies in two-fold: (1) they treat re-ID as a secondary task whose accuracy heavily depends on the primary detection task. So training is largely biased to the detection task but ignores the re-ID task; (2) they use ROI-Align to extract re-ID features which is directly borrowed from object detection. However, this introduces a lot of ambiguity in characterizing objects because many sampling points may belong to disturbing instances or background. To solve the problems, we present a simple approach *FairMOT* which consists of two homogeneous branches to predict pixel-wise objectness scores and re-ID features. The achieved fairness between the tasks allows *FairMOT* to obtain high levels of detection and tracking accuracy and outperform previous state-of-the-arts by a large margin on several public datasets. The source code and pre-trained models are released at https://github.com/ifzhang/FairMOT.

Index Terms—Multi-object tracking, one-shot, anchor-free, real-time.

1 Introduction

Multi-Object Tracking (MOT) has been a longstanding goal in computer vision [1], [2], [3], [4] which aims to estimate trajectories for objects of interest in videos. The successful resolution of the problem can benefit many applications such as video analysis, action recognition, smart elderly care, and human computer interaction.

The existing methods such as [1], [2], [3], [4], [5], [6], [7] often address the problem by two separate models: the *detection* model firstly localizes the objects of interest by bounding boxes in each frame, then the *association* model extracts re-identification (re-ID) features for each bounding box and links it to one of the existing tracks according to certain metrics defined on features. There has been remarkable progress on object detection [8], [9], [10], [11] and re-ID [3], [12] respectively in recent years which in turn significantly boosts the overall tracking performance. However, those methods cannot perform real-time inference especially when there are a large number of objects because the two models do not share features and they need to apply the re-ID models for every bounding box in the video.

With maturity of multi-task learning [13], one-shot trackers which estimate objects and learn re-ID features using a single network have attracted more attention [14], [15]. For example, Voigtlaender *et al*. [15] propose to add a re-ID branch on top of Mask R-CNN to obtain proposals' re-ID features using ROI-Align. It reduces inference time by re-using the backbone features for the re-ID network. Unfortunately, the tracking accuracy drops remarkably compared to the two-step ones. In particular, the number of ID switches increases by a large margin. The result suggests that combining the two tasks is a non-trivial problem and should be treated carefully. In this paper, we aim to deeply understand the reasons behind the failure, and present a simple yet effective approach. In particular, three factors are identified.

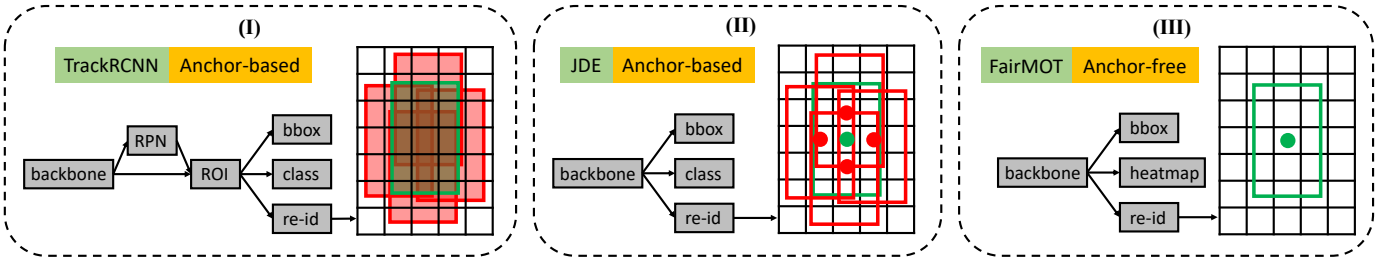
1.1 Unfairness Caused by Anchors

The existing one-shot trackers such as Track R-CNN [15] and JDE [14] are mostly anchor-based since they are directly modified from anchor-based object detectors such as YOLO [11] and Mask R-CNN [9]. However, we find in this study that the anchor-based framework is not suitable for learning re-ID features which result in a large number of ID switches in spite of the good detection results.

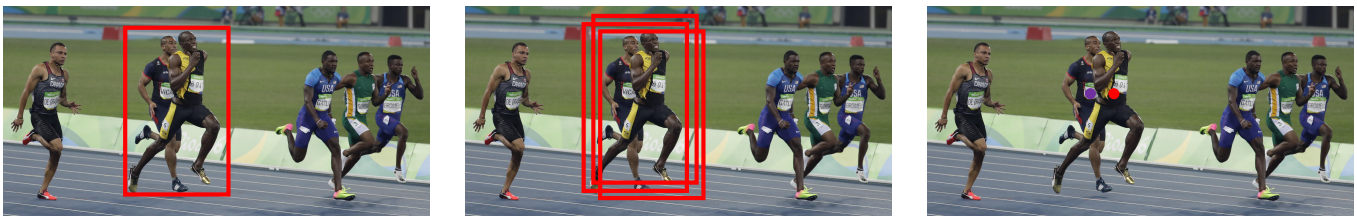
Overlooked re-ID task: Track R-CNN [15] operates in a cascaded style which first estimates object proposals (boxes) and then pools re-ID features from the proposals to estimate the corresponding re-ID features. It is worth noting that the quality of re-ID features heavily depends on the quality of proposals. As a result, in the training stage, the model is seriously biased to estimate accurate object proposals rather than high quality re-ID features. To summarize, this de facto standard "detection first, re-ID secondary" framework makes the re-ID network not fairly learned.

One anchor corresponds to multiple identities: The anchor-based methods usually use ROI-Pool or ROI-Align to extract features from each proposal. Most sampling locations

- *Y. Zhang, X. Wang and W. Liu are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China.*
  Email: {yifuzhang, xgwang, liuwy}@hust.edu.cn
- *C. Wang and W. Zeng are with Microsoft Research Asia.*
  E-mail: {chnuwa, wezeng}@microsoft.com
- ** Equal contribution. † Corresponding author.*



(a) *Comparison of the existing one-shot trackers and FairMOT*



(b) *One anchor contains multiple identities* (c) *Multiple anchors response for one identity* (d) *One point for one identity*

Fig. 1. (a) Track R-CNN treats detection as the primary task and re-ID as the secondary one. Both Track R-CNN and JDE are anchor-based. The red boxes represent positive anchors and the green boxes represent the target objects. The three methods extract re-ID features differently. Track R-CNN extracts re-ID features for all positive anchors using ROI-Align. JDE extracts re-ID features at the centers of all positive anchors. FairMOT extracts re-ID features at the object center. (b) The red anchor contains two different instances. So it will be forced to predict two conflicting classes. (c) Three different anchors with different image patches are response for predicting the same identity. (d) FairMOT extracts re-ID features only at the object center and can mitigate the problems in (b) and (c).

in ROI-Align may belong to other disturbing instances or background as shown in Figure 1. As a result, the extracted features are not optimal in terms of accurately and discriminatively representing the target objects. Instead, we find in this work that it is significantly better to only extract features at the estimated object centers.

Multiple anchors correspond to one identity: In both [15] and [14], multiple adjacent anchors, which correspond to different image patches, may be forced to estimate the same identity as long as their IoU is sufficiently large. This introduces severe ambiguity for training. See Figure 1 for illustration. On the other hand, when an image undergoes small perturbation, *e.g.*, due to data augmentation, it is possible that the same anchor is forced to estimate different identities. In addition, feature maps in object detection are usually down-sampled by $8/16/32$ times to balance accuracy and speed. This is acceptable for object detection but it is too coarse for learning re-ID features because features extracted at coarse anchors may not be aligned with object centers.

1.2 Unfairness Caused by Features

For one-shot trackers, most features are shared between the object detection and re-ID tasks. But it is well known that they actually require features from different layers to achieve the best results. In particular, object detection requires deep and abstract features to estimate object classes and positions but re-ID focuses more on low-level appearance features to distinguish different instances of the same class. We empirically find that multi-layer feature aggregation is effective to address the contradiction by allowing the two tasks (network branches) to extract whatever features they need from the multi-layer aggregated features. Without multi-layer fusion, the model will be biased to the primary detection branch and generates low-quality re-ID features. In addition, multi-layer fusion, which fuses features from layers with different receptive fields, also improves the capability to handle object scale variation which is very common in practice.

1.3 Unfairness Caused by Feature Dimension

The previous re-ID works usually learn very high dimensional features and have achieved promising results on the benchmarks of their field. However, we find that learning lower-dimensional features is actually better for one-shot MOT for three reasons: (1) although learning high dimensional re-ID features may slightly improve their capability to differentiate objects, it notably harms the object detection accuracy due to the competition of the two tasks which in turn also has negative impact to the final tracking accuracy. So considering that the feature dimension in object detection is usually very low (class numbers + box locations), we propose to learn low-dimensional re-ID features to balance the two tasks; (2) when training data is small, learning low dimensional re-ID features reduces the risk of over-fitting. The datasets in MOT are usually much smaller than those in the re-ID area. So it is favorable to decrease feature dimensions; (3) learning low dimensional re-ID features improves the inference speed as will be shown in our experiments.

1.4 Overview of FairMOT

In this work, we present a simple approach termed as *FairMOT* to jointly address the three fairness issues. It essentially differs from the previous "detection first, re-ID secondary" framework because the detection and re-ID tasks are treated equal in *FairMOT*. Our contributions are three-fold. Firstly, we empirically demonstrate and discuss the challenges faced by the previous one-shot tracking frameworks which have been overlooked but severely limit their performance. Second, on top of the anchor-less object detection methods such as [10], we introduce a framework to fairly balance the detection and re-ID tasks which significantly outperforms the previous methods without bells and whistles. Finally, we also present a self supervised learning approach to train *FairMOT* on large scale detection datasets which improves its generalization capability. This has significant empirical values.

Figure 2 shows an overview of *FairMOT*. It adopts a very simple network structure which consists of two *homogeneous* branches for detecting objects and extracting re-ID features, respectively. Inspired by [10], [16], [17], [18], the detection branch is implemented in an *anchor-free* style which estimates object centers and sizes represented as position-aware measurement maps. Similarly, the re-ID branch estimates a re-ID feature for each pixel to characterize the object centered at the pixel. Note that the two branches are completely homogeneous which essentially differs from the previous methods which perform detection and re-ID in a cascaded style. So *FairMOT* eliminates the unfair advantage of the detection branch as reflected in Table 3, effectively learns high-quality re-ID features and obtains a good trade-off between detection and re-ID for better MOT results.

It is also worth noting that *FairMOT* operates on high-resolution feature maps of strides four while the previous anchor-based methods operate on feature maps of stride 32. The elimination of anchors as well as the use of high-resolution feature maps better aligns re-ID features to object centers which significantly improves the tracking accuracy. The dimension of re-ID features is set to be only 64 which not only reduces computation time but also improves tracking robustness by striking a good balance between the detection and re-ID tasks. We equip the backbone network [19] with the Deep Layer Aggregation operator [20] to fuse features from multiple layers in order to accommodate both branches and handle objects of different scales.

We evaluate *FairMOT* on the MOT Challenge benchmark via the evaluation server. It ranks first among all trackers on the 2DMOT15 [21], MOT16 [22], MOT17 [22] and MOT20 [23] datasets. When we further pre-train our model using our proposed self supervised learning method, it achieves additional gains on all datasets. In spite of the strong results, the approach is very simple and runs at 30 FPS on a single RTX 2080Ti GPU. It sheds light on the relationship between detection and re-ID in MOT and provides guidance for designing one-shot video tracking networks.

2 Related Work

We first review the related work on MOT including both deep learning and non-deep learning based ones. Then we briefly talk about video object detection since it is also related to object tracking. We discuss the pros and cons of the methods and compare them to our approach.

2.1 Non-deep Learning MOT Methods

Multi-object tracking can be classified into online methods [1], [24], [25], [26], [27] and batch methods [28], [29], [30], [31], [32], [33] based on whether they rely on future frames. Online methods can only use current and previous frames while batch methods use the whole sequence.

Most online methods assume object detection is available and focus on the data association step. For example, SORT [1] first uses Kalman Filter [34] to predict future object locations, computes their overlap with the detected objects in future frames, and finally adopts Hungarian algorithm [35] for tracking. IOU-Tracker [24] directly associates detections in neighboring frames by their spatial overlap without using Kalman filter and achieves 100K fps inference speed (detection time not counted). Both SORT and IOU-Tracker are widely used in practice due to their simplicity. However, they may fail for challenging scenarios such as crowded scenes and fast camera motion due to lack of re-ID features. Bae *et al*. [26] apply Linear Discriminant Analysis to extract re-ID features for objects which achieves more robust tracking results. Xiang *et al*. [25] formulate online MOT as Markov Decision Processes (MDPs) and leverage online single object tracking and reinforcement learning to decide birth/death and appearance/disappearance of tracklets.

The class of batch methods have achieved better results than the online ones due to its effective global optimization in the whole sequence. For example, Zhang *et al*. [28] build a graphical model with nodes representing detections in all frames for multi-object tracking. The global optimum is searched using a min-cost flow algorithm, which exploits the specific structure of the graph to reach the optimum faster than Linear Programming. Berclaz *et al*. [29] also treat data association as a flow optimization task and use the K-shortest paths algorithm to solve it, which significantly speeds up computation and reduces parameters that need to be tuned. Milan *et al*. [31] formulate multi-object tracking as minimization of a continuous energy and focus on designing the energy function. The energy depends on locations and motion of all targets in all frames as well as physical constraints.

2.2 Deep Learning MOT Methods

The rapid development of deep learning has motivated researchers to explore modern object detectors instead of using the baseline detection results provided by the benchmark datasets. For example, some best performing methods such as [2], [4], [5], [6], [7] treat object detection and re-ID as two separate tasks. They first apply CNN-based object detectors such as Faster R-CNN [8] and YOLOv3 [11] to localize all objects of interest in input images. Then in a separate step,

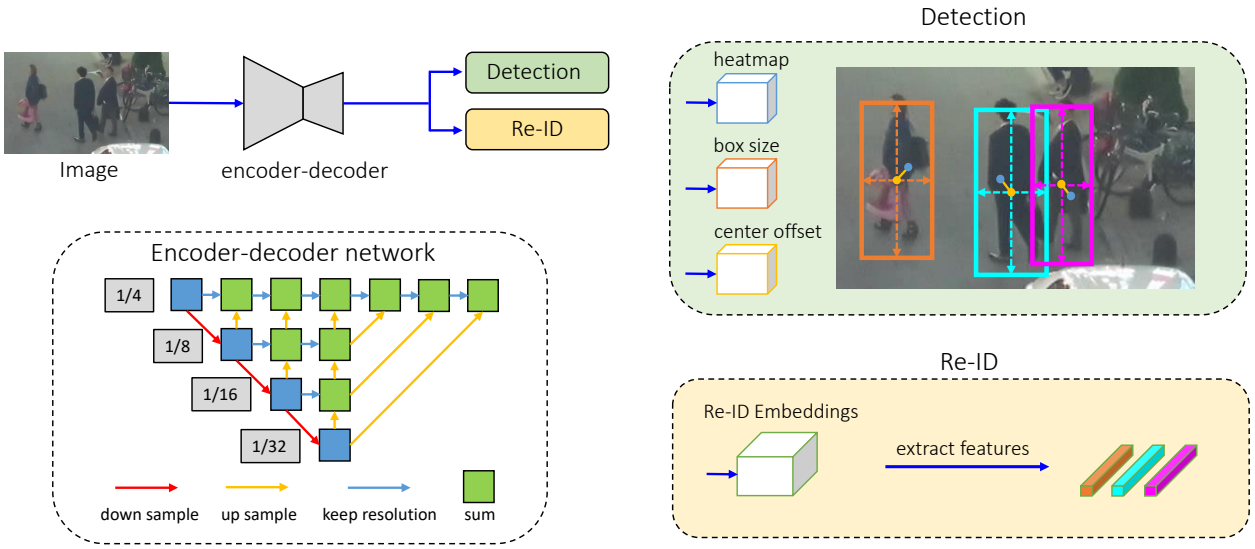


Fig. 2. Overview of our one-shot tracker *FairMOT*. The input image is first fed to an encoder-decoder network to extract high resolution feature maps (stride=4). Then we add two homogeneous branches for detecting objects and extracting re-ID features, respectively. The features at the predicted object centers are used for tracking.

they crop the images according to the boxes and feed them to an identity embedding network to extract re-ID features which are used to link the boxes over time. The linking step usually follows a standard practice which first computes a cost matrix according to the re-ID features and Intersection over Unions (IoU) of the bounding boxes and then uses the Kalman Filter [34] and Hungarian algorithm [35] to accomplish the linking task. A small number of works such as [5], [6], [7] also propose to use more complicated association strategies such as group models and RNNs.

The main advantage of the two-step methods is that they can develop the most suitable model for each task separately without making compromise. In addition, they can crop the image patches according to the detected bounding boxes and resize them to the same size before estimating re-ID features. This helps to handle the scale variations of objects. As a result, these approaches [4] have achieved the best performance on the public datasets. However, they are usually very slow because the two tasks need to be done separately without sharing. So it is hard to achieve video rate inference which is required in many applications.

With the quick maturity of multi-task learning [13], [36], [37] in deep learning, one-shot MOT has begun to attract more research attention. The core idea is to simultaneously accomplish object detection and identity embedding (re-ID features) in a single network in order to reduce inference time. For example, Track-RCNN [15] adds a re-ID head on top of Mask R-CNN [9] and regresses a bounding box and a re-ID feature for each proposal. Similarly, JDE [14] is built on top of YOLOv3 [11] which achieves near video rate inference. However, the accuracy of the one-shot trackers is usually lower than that of the two-step ones.

Our work also belongs to one-shot tracker. Different from the previous works, we deeply investigate the reasons behind the failure and find that the re-ID task is treated unfairly compared to the detection task from three aspects. On top of that, we propose FairMOT which achieves a good balance between the two tasks. We show that the tracking accuracy is improved significantly without heavy engineering efforts.

Video Object Detection (VOD) is related to MOT in the sense that it leverages object tracking to improve object detection [38], [39] in challenging frames. For example, Tang *et al.* [40] detect object tubes in videos which aims to enhance classification scores in challenging frames based on their neighboring frames. The detection rate for small objects increases by a large margin on the benchmark dataset. Similar ideas have also been explored in [40], [41], [42], [43], [44].One main limitation of these tube-based methods is that they are extremely slow especially where there are a large number of objects in videos.

3 FAIRMOT

In this section, we present the technical details of *FairMOT* including the backbone network, the object detection branch, the re-ID branch as well as training details.

3.1 Backbone Network

We adopt ResNet-34 as backbone in order to strike a good balance between accuracy and speed. An enhanced version of Deep Layer Aggregation (DLA) [10] is applied to the backbone to fuse multi-layer features as shown in Figure 2. Different from original DLA [20], it has more skip connections between low-level and high-level features which is similar to the Feature Pyramid Network (FPN) [45]. In addition, convolution layers in all up-sampling modules are replaced by deformable convolution such that they can dynamically adjust the receptive field according to object scales and poses. These modifications are also helpful to alleviate the alignment issue. The resulting model is named DLA-34. Denote the size of input image as $H_{\text{image}} \times W_{\text{image}}$, then the output feature

map has the shape of $C \times H \times W$ where $H = H_{\text{image}}/4$ and $W = W_{\text{image}}/4$. Besides DLA, other deep networks that provide multi-scale convolutional features, such as Higher HRNet [46], can be used in our framework to provide fair features for both detection and re-ID.

3.2 Detection Branch

Our detection branch is built on top of CenterNet [10] but other anchor-free methods such as [16], [18], [47], [48] can also be used. We briefly describe the approach to make this work self-contained. In particular, three parallel heads are appended to DLA-34 to estimate heatmaps, object center offsets and bounding box sizes, respectively. Each head is implemented by applying a 3×3 convolution (with 256 channels) to the output features of DLA-34, followed by a 1×1 convolutional layer which generates the final targets.

3.2.1 Heatmap Head

This head is responsible for estimating the locations of the object centers. The heatmap based representation, which is the de facto standard for the landmark point estimation task, is adopted here. In particular, the dimension of the heatmap is $1 \times H \times W$. The response at a location in the heatmap is expected to be one if it collapses with the ground-truth object center. The response decays exponentially as the distance between the heatmap location and the object center.

For each GT box $\mathbf{b}^i = (x_1^i, y_1^i, x_2^i, y_2^i)$ in the image, we compute the object center (c_x^i, c_y^i) as $c_x^i = \frac{x_1^i + x_2^i}{2}$ and $c_y^i = \frac{y_1^i + y_2^i}{2}$, respectively. Then its location on the feature map is obtained by dividing the stride $(\widetilde{c}_x^i, \widetilde{c}_y^i) = (\lfloor \frac{c_x^i}{4} \rfloor, \lfloor \frac{c_y^i}{4} \rfloor)$. Then the heatmap response at the location (x, y) is computed as $M_{xy} = \sum_{i=1}^{N} \exp^{-\frac{(x-\widetilde{c}_x^i)^2 + (y-\widetilde{c}_y^i)^2}{2\sigma_c^2}}$ where N represents the number of objects in the image and σ_c represents the standard deviation. The loss function is defined as pixel-wise logistic regression with focal loss [49]:

$$L_{\text{heat}} = -\frac{1}{N} \sum_{xy} \begin{cases} (1 - \hat{M}_{xy})^{\alpha} \log(\hat{M}_{xy}), & M_{xy} = 1; \\ (1 - M_{xy})^{\beta} (\hat{M}_{xy})^{\alpha} \log(1 - \hat{M}_{xy}) & \text{otherwise}, \end{cases} \tag{1}$$

where $\hat{M}$ is the estimated heatmap, and α, β are the pre-determined parameters in focal loss.

3.2.2 Box Offset and Size Heads

The box offset head aims to localize objects more precisely. Since the stride of the final feature map is four, it will introduce quantization errors up to four pixels. This branch estimates a continuous offset relative to the object center for each pixel in order to mitigate the impact of down-sampling. The box size head is responsible for estimating height and width of the target box at each location.

Denote the output of the *size* and *offset* heads as $\hat{S} \in R^{W \times H \times 2}$ and $\hat{O} \in R^{W \times H \times 2}$, respectively. For each GT box $\mathbf{b}^i = (x_1^i, y_1^i, x_2^i, y_2^i)$ in the image, we compute its size as $\mathbf{s}^i = (x_2^i - x_1^i, y_2^i - y_1^i)$. Similarly, the GT offset is computed as $\mathbf{o}^i = (\frac{c_x^i}{4}, \frac{c_y^i}{4}) - (\lfloor \frac{c_x^i}{4} \rfloor, \lfloor \frac{c_y^i}{4} \rfloor)$. Denote the estimated size and offset at the corresponding location as $\hat{\mathbf{s}}^i$ and $\hat{\mathbf{o}}^i$, respectively. Then we enforce l_1 losses for the two heads:

$$L_{\text{box}} = \sum_{i=1}^{N} \|\mathbf{o}^i - \hat{\mathbf{o}}^i\|_1 + \|\mathbf{s}^i - \hat{\mathbf{s}}^i\|_1. \tag{2}$$

3.3 Re-ID Branch

Re-ID branch aims to generate features that can distinguish objects. Ideally, affinity among different objects should be smaller than that between same objects. To achieve this goal, we apply a convolution layer with 128 kernels on top of backbone features to extract re-ID features for each location. Denote the resulting feature map as $\mathbf{E} \in R^{128 \times W \times H}$. The re-ID feature $\mathbf{E}_{x,y} \in R^{128}$ of an object centered at (x, y) can be extracted from the feature map.

3.3.1 Re-ID Loss

We learn re-ID features through a classification task. All object instances of the same identity in the training set are treated as the same class. For each GT box $\mathbf{b}^i = (x_1^i, y_1^i, x_2^i, y_2^i)$ in the image, we obtain the object center on the heatmap $(\widetilde{c}_x^i, \widetilde{c}_y^i)$. We extract the re-ID feature vector $\mathbf{E}_{\widetilde{c}_x^i, \widetilde{c}_y^i}$ and learn to map it to a class distribution vector $\mathbf{P} = \{\mathbf{p}(k), k \in [1, K]\}$. Denote the one-hot representation of the GT class label as $\mathbf{L}^i(k)$. Then we compute the re-ID loss as:

$$L_{\text{identity}} = -\sum_{i=1}^{N} \sum_{k=1}^{K} \mathbf{L}^i(k) \log(\mathbf{p}(k)), \tag{3}$$

where K is the number of classes. During the training process of our network, only the identity embedding vectors located at object centers are used for training, since we can obtain object centers from the objectness heatmap in testing.

3.4 Training FairMOT

We jointly train the detection and re-ID branches by adding the losses (i.e., Eq. (1), Eq. (2) and Eq. (3)) together. In particular, we use the uncertainty loss proposed in [50] to automatically balance the detection and re-ID tasks:

$$L_{\text{detection}} = L_{\text{heat}} + L_{\text{box}}, \tag{4}$$

$$L_{\text{total}} = \frac{1}{2}\left(\frac{1}{e^{w_1}} L_{\text{detection}} + \frac{1}{e^{w_2}} L_{\text{identity}} + w_1 + w_2\right), \tag{5}$$

where w_1 and w_2 are learnable parameters that balance the two tasks. Specifically, given an image with a few objects and their corresponding IDs, we generate ground-truth heatmaps, box offset and size maps as well as one-hot class representation of the objects. These are compared to the estimated measures to obtain losses to train the whole network.

In addition to the standard training strategy presented above, we propose a weakly supervised learning method to train FairMOT on image-level object detection datasets such as COCO. Inspired by [51], we regard each object instance in the dataset as a separate class and different transformations of the same object as instances in the same class. The adopted transformations include HSV augmentation, rotation,

scaling, translation and shearing. We pre-train our model on the CrowdHuman dataset [52] and then finetune it on the MOT datasets. With this self supervised learning approach, we further improve the final performance.

3.5 Online Inference

In this section, we present how we perform online inference, and in particular, how we perform association with the detections and re-ID features.

3.5.1 Network Inference

The network takes a frame of size 1088×608 as input which is the same as the previous work JDE [14]. On top of the predicted heatmap, we perform non-maximum suppression (NMS) based on the heatmap scores to extract the peak keypoints. We keep the locations of the keypoints whose heatmap scores are larger than a threshold. Then, we compute the corresponding bounding boxes based on the estimated offsets and box sizes. We also extract the identity embeddings at the estimated object centers. In the next section, we discuss how we associate the detected boxes over time using the re-ID features.

3.5.2 Online Association

We follow the standard online tracking algorithm to associate boxes. We first initialize a number of tracklets based on the estimated boxes in the first frame. Then in the subsequent frame, we link the detected boxes to the existing tracklets according to their cosine distances computed on Re-ID features and their box overlap by bipartite matching [35]. We also use Kalman Filter [34] to predict the locations of the tracklets in the current frame. If it is too far from the linked detection, we set the corresponding cost to infinity which effectively prevents from linking the detections with large motion. We update the appearance features of the trackers in each time step to handle appearance variations as in [53], [54].

4 Experiments

4.1 Datasets and Metrics

There are six training datasets briefly introduced as follows: the ETH [55] and CityPerson [56] datasets only provide box annotations so we only train the detection branch on them. The CalTech [57], MOT17 [22], CUHK-SYSU [58] and PRW [12] datasets provide both box and identity annotations which allows us to train both branches. Some videos in ETH also appear in the testing set of the MOT16 which are removed from the training dataset for fair comparison. The overall training strategy is described in Section 3.4, which is the same as [14]. For the self-supervised training of our method, we use the CrowdHuman dataset [52] which only contains object bounding box annotations.

We extensively evaluate a variety of factors of our approach on the testing sets of four benchmarks: 2DMOT15, MOT16, MOT17 and the recently released MOT20. Following the common practices in MOT, we use Average Precision (*AP*) for evaluating detection performance, and True Positive Rate (*TPR*) at a false accept rate of 0.1 for rigorously evaluating re-ID features with ground-truth detections. We use the CLEAR metric [59] and *IDF1* [60] to evaluate overall tracking accuracy.

TABLE 1
Comparison of different re-ID feature extraction (sampling) strategies on the validation set of MOT17. The rest of the models are kept the same for fair comparison. ↑ means the larger the better and ↓ means the smaller the better. The best results are shown in **bold**.

Feature Extraction	Anchor	MOTA↑	IDF1↑	IDs↓	TPR↑
FairMOT (ROI-Align)	✓	68.7	71.0	331	93.1
FairMOT (POS-Anchor)	✓	69.0	70.3	434	93.9
FairMOT (Center)		**69.1**	72.8	**299**	94.4
FairMOT (Center-BI)		68.8	**74.3**	303	**94.9**
FairMOT (Two-Stage)	✓	69.0	68.2	388	90.5

4.2 Implementation Details

We use a variant of DLA-34 proposed in [10] as our default backbone. The model parameters pre-trained on the COCO dataset [61] are used to initialize our model. We train our model with the Adam optimizer [62] for 30 epochs with a starting learning rate of e^{-4}. The learning rate decays to e^{-5} at 20 epochs. The batch size is set to be 12. We use standard data augmentation techniques including rotation, scaling and color jittering. The input image is resized to 1088×608 and the feature map resolution is 272×152. The training step takes about 30 hours on two RTX 2080 Ti GPUs.

4.3 Ablative Studies

In this section, we present rigorous studies of the three critical factors in *FairMOT* including anchor-less re-ID feature extraction, feature fusion and feature dimensions by carefully designing a number of baseline methods.

4.3.1 Fairness Issue in Anchors

We evaluate four strategies for sampling re-ID features from the detected boxes which are frequently used by previous works [14] [15]. The first strategy is ROI-Align used in Track R-CNN [15]. It samples features from the detected proposals using ROI-Align. As discussed previously, many sampling locations deviate from object centers. The second strategy is POS-Anchor used in JDE [14]. It samples features from positive anchors which may also deviate from object centers. The third strategy is "Center" used in *FairMOT*. It only samples features at object centers. Recall that, in our approach, re-ID features are extracted from discretized low-resolution maps. In order to sample features at accurate object locations, we also try to apply Bi-linear Interpolation (Center-BI) to extract more accurate features.

We also evaluate a two-stage approach to first detect object bounding boxes and then extract re-ID features. In the first stage, the detection part is the same as our *FairMOT*. In the second stage, we use ROI-Align [9] to extract the backbone

TABLE 2
Comparison of different backbones on the validation set of MOT17 dataset. The best results are shown in **bold**.

Backbone	MOTA↑	IDF1↑	IDs↓	AP↑	TPR↑
ResNet-34	63.6	67.2	435	75.1	90.9
ResNet-50	63.7	67.7	501	75.5	91.9
ResNet-34-FPN	64.4	69.6	369	77.7	94.2
HRNet-W18	67.4	**74.3**	315	80.5	**94.6**
DLA-34	**69.1**	72.8	**299**	**81.2**	94.4

features based on the detected bounding boxes and then use a re-ID head (a fully connected layer) to get re-ID features.

The results are shown in Table 1. Note that the five approaches are all built on top of *FairMOT*. The only difference lies in how they sample re-ID features from detected boxes. First, we can see that our approach (Center) obtains notably higher *IDF1* score and True Positive Rate (*TPR*) than ROI-Align, POS-Anchor and the two-stage approach. This metric is independent of object detection results and faithfully reflects the quality of re-ID features. In addition, the number of ID switches (*IDs*) of our approach is also significantly smaller than the two baselines. The results validate that sampling features at object centers is more effective than the strategies used in the previous works. Bi-linear Interpolation (Center-BI) achieves even higher *TPR* than Center because it samples features at more accurate locations. The two-stage approach harms the quality of the re-ID features.

4.3.2 Fairness Issue in Features

We aim to study the effectiveness of multi-layer feature fusion in addressing the unfairness issue in features. To that end, we compare a number of backbones such as vanilla ResNet [19], Feature Pyramid Network (FPN) [45], High-Resolution Network (HRNet) [63] and DLA-34 [10] in terms of re-ID features and detection accuracy. Note that the rest of the factors of these approaches such as training datasets are all controlled to be the same for fair comparison. In particular, the stride of the final feature map is four for all methods. We add three up-sampling operations for vanilla ResNet to obtain feature maps of stride four.

The results are shown in Table 2. By comparing the results of ResNet-34 and ResNet-50, we surprisingly find that using a larger network only slightly improves the overall tracking result measured by *MOTA*. In particular, the quality of re-ID features barely benefits from the larger network. For example, *IDF1* only improves from 67.2% to 67.7% and *TPR* improves from 90.9% to 91.9%, respectively. In addition, the number of *ID switches* even increases from 435 to 501. All these results suggest that using a larger network adds very limited values to the final tracking accuracy.

In contrast, ResNet-34-FPN, which actually has fewer parameters than ResNet-50, achieves a larger *MOTA* score than ResNet-50. More importantly, *TPR* improves significantly from 90.9% to 94.2% which suggests that multi-layer feature fusion has clear advantages over simply using larger networks. In addition, DLA-34, which is also built on top of ResNet-34 but has more levels of feature fusion, achieves an even larger *MOTA* score. In particular, *TPR* increases significantly from 90.9% to 94.4% which in turn decreases the number of ID switches (*IDs*) from 435 to 299. The results validate that feature fusion (both FPN and DLA) effectively improves the discriminative ability of re-ID features. On the other hand, although ResNet-34-FPN obtains equally good re-ID features (*TPR*) as DLA-34, its detection results (*AP*) are significantly worse than DLA-34. We think the use of deformable convolution in DLA-34 is the main reason because it enables more flexible receptive fields for objects of different sizes - it is very important for our method since *FairMOT* only extracts features from object centers without using any region features. We can only get 65.0 *MOTA* and 78.1 *AP* when replacing all the deformable convolutions with normal convolutions in DLA-34. As shown in Table 4, we can see that DLA-34 mainly outperforms HRNet-W18 on middle and large size objects.

To validate the existence of *feature conflict* between the detection and re-ID tasks, we introduce a baseline ResNet-34-det which only trains the detection branch (re-ID branch is randomly initialized). We can see from Table 3 that the detection result measured by *AP* improves by a large margin if we do not train the re-ID branch which shows the conflict between the two tasks. In particular, ResNet-34-det even gets higher *MOTA* score than ResNet-34 because the metric

TABLE 3
Demonstration of *feature conflict* between the detection and re-ID tasks on the validation set of the MOT17 dataset. "-det" means only the detection branch is trained and the re-ID branch is randomly initialized.

Backbone	MOTA↑	IDF1↑	IDs↓	AP↑	TPR↑
ResNet-34	63.6	67.2	435	75.1	90.9
ResNet-34-det	63.7	60.4	597	76.1	36.7
DLA-34	**69.1**	**72.8**	**299**	**81.2**	**94.4**

TABLE 4
The impact of different backbones on objects of different scales. *Small*: area smaller than 7000 pixels; *Medium*: area from 7000 to 15000 pixels; *Large*: area larger than 15000 pixels. The best results are shown in **bold**.

Backbone	AP^S	AP^M	AP^L	TPR^S	TPR^M	TPR^L	IDs^S	IDs^M	IDs^L
ResNet-34	40.6	57.8	85.2	91.7	85.7	88.8	190	87	118
ResNet-50	39.7	59.4	86.0	91.3	85.3	89.0	248	91	124
ResNet-34-FPN	45.9	61.0	85.4	90.7	91.5	**93.3**	166	71	90
HRNet-W18	**51.1**	63.7	85.7	**94.2**	**92.5**	93.1	168	**55**	**56**
DLA-34	46.8	**65.1**	**88.8**	92.7	91.2	91.8	**134**	64	70

TABLE 5
Evaluation of re-ID feature dimensions on the validation set of MOT17. The best results are shown in **bold**.

Backbone	dim	MOTA ↑	IDF1 ↑	IDs ↓	FPS↑	AP↑	TPR ↑
DLA-34	512	68.5	**73.7**	312	24.1	80.9	**94.6**
DLA-34	256	68.5	72.5	337	26.1	81.1	94.3
DLA-34	128	69.1	72.8	299	26.6	81.2	94.4
DLA-34	64	**69.2**	73.3	**283**	**26.8**	**81.3**	94.3

TABLE 6
Evaluation of the three ingredients in the data association model. The backbone is DLA-34.

Box IoU	Re-ID Features	Kalman Filter	MOTA ↑	IDF1 ↑	IDs ↓
✓			67.8	67.2	648
	✓		68.1	70.3	435
	✓	✓	68.9	71.8	342
✓	✓	✓	**69.1**	**72.8**	**299**

TABLE 7
Effects of self supervised learning on the validation set of MOT17. "CH" and "MIX" stand for CrowdHuman and the composed five datasets introduced in Section 4.1, respectively. * means no identity annotations are used.

Training Data	MOTA ↑	IDF1 ↑	IDs ↓	AP↑	TPR ↑
MOT17	67.5	69.9	408	79.6	93.4
CH*+MOT17	**71.1**	**75.6**	327	**83.0**	93.6
MIX+MOT17	69.1	72.8	**299**	81.2	**94.4**

favors better detection than tracking results. In contrast, DLA-34, which adds multi-layer feature fusion over ResNet-34, achieves better detection as well as tracking results. It means multi-layer feature fusion helps alleviate the *feature conflict* problem by allowing each task to extract whatever it needs for its own task from the fused features.

4.3.3 Fairness Issue in Feature Dimensionality

The previous one-shot trackers usually learn 512 dimensional re-ID features following the two-step methods without ablation study. However, we find in our experiments that the feature dimension actually plays an important role in balancing detection and tracking accuracy. Learning lower dimensional re-ID features causes less harm to the detection accuracy and improves the inference speed.

We evaluate multiple choices for re-ID feature dimensionality in Table 5. We can see that 512 achieves the highest *IDF1* and *TPR* scores which indicates that higher dimensional re-ID features lead to stronger discriminative ability. However, it is surprising that the *MOTA* score consistently improves when we decrease the dimension from 512 to 64. This is mainly caused by the conflict between the detection and re-ID tasks. In particular, we can see that the detection result (*AP*) improves when we decrease the dimension of re-ID features. In our experiments, we set the feature dimension to be 64 which strikes a good balance between the two tasks.

4.3.4 Data Association Methods

This section evaluates the three ingredients in the data association step including bounding box IoU, re-ID features and Kalman Filter [34]. These are used to compute the similarity between each pair of detected boxes. With that we use Hungarian algorithm [35] to solve the assignment problem. Table 6 shows the results. We can see that only using box IoU causes a lot of *ID switches*. This is particularly true for crowded scenes and fast camera motion. Using re-ID features alone notably increases *IDF1* and decreases the number of *ID switches*. In addition, adding Kalman filter helps obtain smooth (reasonable) tracklets which further decreases the number of *ID switches*. When an object is partly occluded, its re-ID features become unreliable. In this case, it is important to leverage box IoU, re-ID features and Kalman filter to obtain good tracking performance.

4.3.5 Visualization of Re-ID Similarity

We use re-ID similarity maps to demonstrate the discriminative ability of re-ID features in Figure 3. We randomly choose two frames from our validation set. The first frame contains the query instance and the second frame contains the target instance that has the same ID. We obtain the re-ID similarity maps by computing the cosine similarity between the re-ID feature of the query instance and the whole re-ID feature map of the target frame, as described in Section 4.3.1 and Section 4.3.2 respectively. By comparing the similarity maps of ResNet-34 and ResNet-34-det, we can see that training the re-ID branch is important. By comparing DLA-34 and ResNet-34, we can see that multi-layer feature aggregation can get more discriminative re-ID features. Among all the sampling strategies, the proposed Center and Center-BI can better discriminate the target object from surrounding objects in crowded scenes.

4.4 Self-supervised Learning

We first pre-train *FairMOT* on the CrowdHuman dataset [52]. In particular, we assign a unique identity label for each bounding box and train *FairMOT* using the method described in section 3.4. Then we finetune the pre-trained model on the target dataset MOT17.

Table 7 shows the results. First, pre-training via self-supervised learning on CrowdHuman outperforms directly training on the MOT17 dataset by a large margin. Second, the self-supervised learning model even outperforms the fully-supervised model trained on the "MIX" and MOT17 datasets. The results validate the effectiveness of the proposed self-supervised pre-training, which saves lots of annotation efforts and makes *FairMOT* more attractive in real applications.

4.5 Results on MOTChallenge

We compare our approach to the state-of-the-art (SOTA) methods including both the one-shot methods and the two-step methods.

4.5.1 Comparing with One-Shot SOTA MOT Methods

There are only two published works of JDE [14] and Track-RCNN [15] that jointly perform object detection and identity feature embedding. We compare our approach to both of them. Following the previous work [14], the testing dataset contains 6 videos from 2DMOT15. *FairMOT* uses the same training data as the two methods as described in their papers. In particular, when we compare to JDE, both *FairMOT* and JDE use the large scale composed dataset described in Section 4.1. Since Track R-CNN requires segmentation labels to train the network, it only uses 4 videos of the MOT17 dataset which has segmentation labels as training data. In this case, we also

Fig. 3. Visualization of the discriminative ability of the re-ID features. Query instances are marked as red boxes and target instances are marked as green boxes. The similarity maps are computed using re-ID features extracted based on different strategies (*e.g.*, Center, Center-BI, ROI-Align and POS-Anchor as described in Section 4.3.1) and different backbones (*e.g.*, ResNet-34 and DLA-34). The query frames and target frames are randomly chosen from the MOT17-09 and the MOT17-02 sequence.

use the 4 videos to train our model. The CLEAR metric [59] and IDF1 [60] are used to measure their performance.

The results are shown in Table 8. We can see that our approach remarkably outperforms JDE [14]. In particular, the number of ID switches reduces from 218 to 80 which is big improvement in terms of user experience. The results validate the effectiveness of the *anchor-free* approach over the previous *anchor-based* one. The inference speed is near video rate for the both methods with ours being faster. Compared with Track R-CNN [15], their detection results are slightly better than ours (with lower *FN*). However, *FairMOT* achieves much higher *IDF1* score (64.0 vs. 49.4) and fewer *ID switches* (96 vs. 294). This is mainly because Track R-CNN follows the "detection first, re-ID secondary" framework and use anchors which also introduce ambiguity to the re-ID task.

4.5.2 Comparing with Two-Step SOTA MOT Methods

We compare our approach to the state-of-the-art trackers including the two-step methods in Table 9. Since we do not use the public detection results, the "private detector" protocol is adopted. We report results on the testing sets of the 2DMOT15, MOT16, MOT17 and MOT20 datasets, respectively. Note that all of the results are directly obtained from the official MOT challenge evaluation server.

Our approach ranks first among all *online* and *offline* trackers on the four datasets. In particular, it outperforms other methods by a large margin. This is a very strong result especially considering that our approach is very simple. In addition, our approach achieves video rate inference. In contrast, most high-performance trackers such as [4], [7] are usually slower than ours.

TABLE 8
Comparison of the state-of-the-art one-shot trackers on the 2DMOT15 dataset. "MIX" represents the large scale training dataset and "MOT17 Seg" stands for the 4 videos with segmentation labels in the MOT17 dataset.

Training Data	Method	MOTA↑	IDF1↑	IDs↓	FP↓	FN↓	FPS↑
MIX	JDE [14]	67.5	66.7	218	1881	**2083**	26.0
	FairMOT(ours)	**77.2**	**79.8**	**80**	**757**	2094	**30.9**
MOT17 Seg	Track R-CNN [15]	69.2	49.4	294	1328	**2349**	2.0
	FairMOT(ours)	**70.2**	**64.0**	**96**	**1209**	2537	**30.9**

4.5.3 Training Data Ablation Study

We also evaluate the performance of *FairMOT* using different amount of training data. We can achieve 69.8 *MOTA* when only using the MOT17 dataset for training, which already outperforms other methods using more training data. When we use the same training data as JDE [14], we can achieve 72.9 *MOTA*, which remarkably outperforms JDE. In addition, when

Fig. 4. Example tracking results of our method on the test set of MOT17. Each row shows the results of sampled frames in chronological order of a video sequence. Bounding boxes and identities are marked in the images. Bounding boxes with different colors represent different identities. Best viewed in color.

TABLE 9
Comparison of the state-of-the-art methods under the "private detector" protocol. It is noteworthy that FPS considers both detection and association time. The one-shot trackers are labeled by "*".

Dataset	Tracker	MOTA↑	IDF1↑	MT↑	ML↓	IDs↓	FPS↑
MOT15	MDP_SubCNN [25]	47.5	55.7	30.0%	18.6%	628	<1.7
	CDA_DDAL [64]	51.3	54.1	36.3%	22.2%	544	<1.2
	EAMTT [65]	53.0	54.0	35.9%	19.6%	7538	<4.0
	AP_HWDPL [66]	53.0	52.2	29.1%	20.2%	708	6.7
	RAR15 [7]	56.5	61.3	45.1%	14.6%	**428**	<3.4
	TubeTK* [44]	58.4	53.1	39.3%	18.0%	854	5.8
	FairMOT (Ours)*	**60.6**	**64.7**	**47.6%**	**11.0%**	591	**30.5**
MOT16	EAMTT [65]	52.5	53.3	19.9%	34.9%	910	<5.5
	SORTwHPD16 [1]	59.8	53.8	25.4%	22.7%	1423	<8.6
	DeepSORT_2 [2]	61.4	62.2	32.8%	18.2%	781	<6.4
	RAR16wVGG [7]	63.0	63.8	39.9%	22.1%	**482**	<1.4
	VMaxx [67]	62.6	49.2	32.7%	21.1%	1389	<3.9
	TubeTK* [44]	64.0	59.4	33.5%	19.4%	1117	1.0
	JDE* [14]	64.4	55.8	35.4%	20.0%	1544	18.5
	TAP [6]	64.8	**73.5**	38.5%	21.6%	571	<8.0
	CNNMTT [5]	65.2	62.2	32.4%	21.3%	946	<5.3
	POI [4]	66.1	65.1	34.0%	20.8%	805	<5.0
	CTrackerV1* [68]	67.6	57.2	32.9%	23.1%	1897	6.8
	FairMOT (Ours)*	**74.9**	72.8	**44.7%**	**15.9%**	1074	**25.9**
MOT17	SST [69]	52.4	49.5	21.4%	30.7%	8431	<3.9
	TubeTK* [44]	63.0	58.6	31.2%	19.9%	4137	3.0
	CTrackerV1* [68]	66.6	57.4	32.2%	24.2%	5529	6.8
	CenterTrack* [70]	67.3	59.9	34.9%	24.8%	**2898**	22.0
	FairMOT (Ours)*	**73.7**	**72.3**	**43.2%**	**17.3%**	3303	**25.9**
MOT20	FairMOT (Ours)*	**61.8**	**67.3**	**68.8%**	**7.6%**	**5243**	**13.2**

TABLE 10
Results of the MOT17 test set when using different datasets for training. "MIX" represents the large scale dataset mentioned in part 4.1 and "CH" is short for the CrowdHuman dataset. All the results are obtained from the MOT challenge server. The best results are shown in **bold**.

Training Data	Images	Boxes	Identities	MOTA↑	IDF1↑	IDs↓
MOT17	5K	112K	0.5K	69.8	69.9	3996
MOT17+MIX	54K	270K	8.7K	72.9	**73.2**	3345
MOT17+MIX+CH	73K	740K	8.7K	**73.7**	72.3	**3303**

we perform self supervised learning on the CrowdHuman dataset, the *MOTA* score improves to 73.7. The results suggest that our approach is not data hungry which is a big advantage in practical applications.

4.6 Qualitative Results

Figure 4 visualizes several tracking results of *FairMOT* on the test set of MOT17 [22]. From the results of MOT17-01, we can see that our method can assign correct identities with the help of high-quality re-ID features when two pedestrians cross over each other. Trackers using bounding box IOUs [1], [24] usually cause identity switches under these circumstances. From the results of MOT17-03, we can see that our method perform well under crowded scenes. From the results of MOT17-08, we can see that our method can keep both correct identities and correct bounding boxes when the pedestrians are heavily occluded. The results of MOT17-06 and MOT17-12 show that our method can deal with large scale variations. This mainly attributes to the using of multi-layer feature aggregation. The results of MOT17-07 and MOT17-14 show that our method can detect small objects accurately.

5 Conclusion

Start from studying why the previous single-shot methods (*e.g.*, [14]) fail to achieve comparable results as the two-step methods, we find that the use of anchors in object detection and identity embedding is the main reason for the degraded results. In particular, multiple nearby anchors, which correspond to different parts of an object, may be responsible for estimating the same identity which causes ambiguities for network training. Further, we find the feature unfairness issue and feature conflict issue between the detection and re-ID tasks in previous MOT frameworks. By addressing these problems in an anchor-free single-shot deep network, we propose *FairMOT*. It outperforms the previous state-of-the-art methods on several benchmark datasets by a large margin in terms of both tracking accuracy and inference speed. Besides, FairMOT is inherently training data-efficient and we propose self-supervised training of multi-object trackers only using bounding box annotated images, which both make our method more appealing in real applications.

Acknowledgements

This work was in part supported by NSFC (No. 61733007 and No. 61876212) and MSRA Collaborative Research Fund.

REFERENCES

[1] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *ICIP*. IEEE, 2016, pp. 3464–3468.

[2] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[3] L. Chen, H. Ai, Z. Zhuang, and C. Shang, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018, pp. 1–6.

[4] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "Poi: Multiple object tracking with high performance detection and appearance feature," in *ECCV*. Springer, 2016, pp. 36–42.

[5] N. Mahmoudi, S. M. Ahadi, and M. Rahmati, "Multi-target tracking using cnn-based features: Cnnmtt," *Multimedia Tools and Applications*, vol. 78, no. 6, pp. 7077–7096, 2019.

[6] Z. Zhou, J. Xing, M. Zhang, and W. Hu, "Online multi-target tracking with tensor-based high-order graph matching," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 1809–1814.

[7] K. Fang, Y. Xiang, X. Li, and S. Savarese, "Recurrent autoregressive networks for online multi-object tracking," in *WACV*. IEEE, 2018, pp. 466–475.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017, pp. 2961–2969.

[10] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[11] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[12] L. Zheng, H. Zhang, S. Sun, M. Chandraker, Y. Yang, and Q. Tian, "Person re-identification in the wild," in *CVPR*, 2017, pp. 1367–1376.

[13] I. Kokkinos, "Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," in *CVPR*, 2017, pp. 6129–6138.

[14] Z. Wang, L. Zheng, Y. Liu, and S. Wang, "Towards real-time multi-object tracking," *arXiv preprint arXiv:1909.12605*, 2019.

[15] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "Mots: Multi-object tracking and segmentation," in *CVPR*, 2019, pp. 7942–7951.

[16] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *ECCV*, 2018, pp. 734–750.

[17] X. Zhou, J. Zhuo, and P. Krahenbuhl, "Bottom-up object detection by grouping extreme and center points," in *CVPR*, 2019, pp. 850–859.

[18] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *ICCV*, 2019, pp. 6569–6578.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[20] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *CVPR*, 2018, pp. 2403–2412.

[21] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *arXiv preprint arXiv:1504.01942*, 2015.

[22] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[23] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "Mot20: A benchmark for multi object tracking in crowded scenes," *arXiv:2003.09003[cs]*, Mar. 2020, arXiv: 2003.09003. [Online]. Available: http://arxiv.org/abs/1906.04567

[24] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2017, pp. 1–6.

[25] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *ICCV*, 2015, pp. 4705–4713.

[26] S.-H. Bae and K.-J. Yoon, "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1218–1225.

[27] C. Dicle, O. I. Camps, and M. Sznaier, "The way they move: Tracking multiple targets with similar appearance," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2304–2311.

[28] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[29] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 9, pp. 1806–1819, 2011.

[30] A. R. Zamir, A. Dehghan, and M. Shah, "Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs," in *European Conference on Computer Vision*. Springer, 2012, pp. 343–356.

[31] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 58–72, 2013.

[32] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S. Z. Li, "Multiple target tracking based on undirected hierarchical relation hypergraph," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1282–1289.

[33] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3029–3037.

[34] G. Welch, G. Bishop *et al.*, "An introduction to the kalman filter," 1995.

[35] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[36] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *T-PAMI*, vol. 41, no. 1, pp. 121–135, 2017.

[37] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in *NIPS*, 2018, pp. 527–538.

[38] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3038–3046.

[39] H. Luo, W. Xie, X. Wang, and W. Zeng, "Detect or track: Towards cost-effective video object detection/tracking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8803–8810.

[40] P. Tang, C. Wang, X. Wang, W. Liu, W. Zeng, and J. Wang, "Object detection in videos by high quality object linking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 5, pp. 1272–1278, 2019.

[41] W. Han, P. Khorrami, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and T. S. Huang, "Seq-nms for video object detection," *arXiv preprint arXiv:1602.08465*, 2016.

[42] K. Kang, W. Ouyang, H. Li, and X. Wang, "Object detection from video tubelets with convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 817–825.

[43] K. Kang, H. Li, T. Xiao, W. Ouyang, J. Yan, X. Liu, and X. Wang, "Object detection in videos with tubelet proposal networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 727–735.

[44] B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu, "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.

[45] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017, pp. 2117–2125.

[46] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang, "Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation," in *CVPR*, 2020.

[47] Z. Dong, G. Li, Y. Liao, F. Wang, P. Ren, and C. Qian, "Centripetalnet: Pursuing high-quality keypoint pairs for object detection," in *CVPR*, 2020, pp. 10 519–10 528.

[48] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "Reppoints: Point set representation for object detection," in *ICCV*, 2019, pp. 9657–9666.

[49] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017, pp. 2980–2988.

[50] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *CVPR*, 2018, pp. 7482–7491.

[51] N. Zhao, Z. Wu, R. W. Lau, and S. Lin, "Distilling localization for self-supervised representation learning," *arXiv preprint arXiv:2004.06638*, 2020.

[52] 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.

[53] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *CVPR*. IEEE, 2010, pp. 2544–2550.
[54] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 3, pp. 583–596, 2014.
[55] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "A mobile vision system for robust multi-person tracking," in *CVPR*. IEEE, 2008, pp. 1–8.
[56] S. Zhang, R. Benenson, and B. Schiele, "Citypersons: A diverse dataset for pedestrian detection," in *CVPR*, 2017, pp. 3213–3221.
[57] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *CVPR*. IEEE, 2009, pp. 304–311.
[58] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, "Joint detection and identification feature learning for person search," in *CVPR*, 2017, pp. 3415–3424.
[59] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.
[60] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *ECCV*. Springer, 2016, pp. 17–35.
[61] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*. Springer, 2014, pp. 740–755.
[62] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[63] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *arXiv preprint arXiv:1908.07919*, 2019.
[64] S.-H. Bae and K.-J. Yoon, "Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 595–610, 2017.
[65] R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro, "Online multi-target tracking with strong and weak detections," in *ECCV*. Springer, 2016, pp. 84–99.
[66] L. Chen, H. Ai, C. Shang, Z. Zhuang, and B. Bai, "Online multi-object tracking with convolutional neural networks," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 645–649.
[67] X. Wan, J. Wang, Z. Kong, Q. Zhao, and S. Deng, "Multi-object tracking using online metric learning with long short-term memory," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 788–792.
[68] J. Peng, C. Wang, F. Wan, Y. Wu, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, "Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking," *arXiv preprint arXiv:2007.14557*, 2020.
[69] S. Sun, N. Akhtar, H. Song, A. S. Mian, and M. Shah, "Deep affinity network for multiple object tracking," *IEEE transactions on pattern analysis and machine intelligence*, 2019.
[70] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," *arXiv:2004.01177*, 2020.