



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

Yifu Zhang<sup>1</sup> · Chunyu Wang<sup>2</sup> · Xinggang Wang<sup>1</sup> · Wenjun Zeng<sup>2</sup> · Wenyu Liu<sup>1</sup>

Received: 18 January 2021 / Accepted: 3 August 2021 / Published online: 3 September 2021  
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

Multi-object tracking (MOT) is an important problem in computer vision which has a wide range of applications. Formulating MOT as multi-task learning of object detection and re-ID in a single network is appealing since it allows joint optimization of the two tasks and enjoys high computation efficiency. However, we find that the two tasks tend to compete with each other which need to be carefully addressed. In particular, previous works usually treat re-ID as a secondary task whose accuracy is heavily affected by the primary detection task. As a result, the network is biased to the primary detection task which is not *fair* to the re-ID task. To solve the problem, we present a simple yet effective approach termed as *FairMOT* based on the anchor-free object detection architecture CenterNet. Note that it is not a naive combination of CenterNet and re-ID. Instead, we present a bunch of detailed designs which are critical to achieve good tracking results by thorough empirical studies. The resulting approach achieves high accuracy for both detection and tracking. The approach outperforms the state-of-the-art methods by a large margin on several public datasets. The source code and pre-trained models are released at <https://github.com/ifzhang/FairMOT>.

**Keywords** FairMOT · Multi-object tracking · One-shot · Anchor-free · Real-time inference

## 1 Introduction

Multi-Object Tracking (MOT) has been a longstanding goal in computer vision (Bewley et al. 2016; Wojke et al. 2017; Chen et al. 2018a; Yu et al. 2016). The goal is to estimate trajectories for objects of interest presented in videos. The

successful resolution of the problem can immediately benefit many applications such as intelligent video analysis, human computer interaction, human activity recognition, and even social computing.

Most of the existing methods such as Mahmoudi et al. (2019), Zhou et al. (2018), Fang et al. (2018), Bewley et al. (2016), Wojke et al. (2017), Chen et al. (2018a), Yu et al. (2016) attempt to address the problem by two separate models: the *detection* model firstly detects objects of interest by bounding boxes in each frame, then the *association* model extracts re-identification (re-ID) features from the image regions corresponding to each bounding box, links the detection to one of the existing tracks or creates a new track according to certain metrics defined on features.

There has been remarkable progress on object detection (Ren et al. 2015; He et al. 2017; Zhou et al. 2019a; Redmon and Farhadi 2018) and re-ID (Zheng et al. 2017a; Chen et al. 2018a) respectively recently which in turn boosts the overall tracking accuracy. However, these two-step methods suffer from scalability issues. They cannot achieve real-time inference speed when there are a large number of objects in the environment because the two models do not share features

---

Communicated by Bumsu Ham.

---

Yifu Zhang and Chunyu Wang have contributed equally to this work.

---

✉ Xinggang Wang  
xgwan@hust.edu.cn

Yifu Zhang  
yifuzhang@hust.edu.cn

Chunyu Wang  
chnuwa@microsoft.com

Wenjun Zeng  
wezeng@microsoft.com

Wenyu Liu  
liuwu@hust.edu.cn

<sup>1</sup> Huazhong University of Science and Technology, Wuhan, China

<sup>2</sup> Microsoft Research Asia, Beijing, China

and they need to apply the re-ID models for every bounding box independently in the video.

With the maturity of multi-task learning (Kokkinos 2017; Chen et al. 2018b), one-shot trackers which estimate objects and learn re-ID features using a single network have attracted more attention (Wang et al. 2020b; Voigtlaender et al. 2019). For example, Voigtlaender et al. (2019) add a re-ID branch to Mask R-CNN to extract a re-ID feature for each proposal (He et al. 2017). It reduces inference time by re-using backbone features for the re-ID network. But the performance drops remarkably compared to the two-step models. In fact, the detection accuracy is still good but the tracking performance drops a lot. For example, the number of ID switches increases by a large margin. The result suggests that combining the two tasks is a non-trivial task and should be treated carefully.

In this paper, we investigate the reasons behind the failure, and present a simple yet effective solution. Three factors are identified to account for the failure. The first issue is caused by anchors. Anchors are originally designed for object detection (Ren et al. 2015). However, we show that anchors are not suitable for extracting re-ID features for two reasons. First, anchor-based one-shot trackers such as Track R-CNN (Voigtlaender et al. 2019) overlook the re-ID task because they need anchors to first detect objects (i.e. using RPN Ren et al. 2015) and then extract the re-ID features based on the detection results (re-ID features are useless when detection results are incorrect). So when competition occurs between the two tasks, it will favor the detection task. Anchors also introduce a lot of ambiguity during training the re-ID features because one anchor may correspond to multiple identities and multiple anchors may correspond to one identity, especially in crowded scenes.

The second issue is caused by feature sharing between the two tasks. Detection task and re-ID task are two totally different tasks and they need different features. In general, re-ID features need more low-level features to discriminate different instances of the same class while detection features need to be similar for different instances. The shared features in one-shot trackers will lead to feature conflict and thus reduce the performance of each task.

The third issue is caused by feature dimension. The dimension of re-ID features is usually as high as 512 (Wang et al. 2020b) or 1024 (Zheng et al. 2017a) which is much higher than that of object detection. We find that the huge difference between dimensions will harm the performance of the two tasks. More importantly, our experiments suggest that it is a generic rule that learning low-dimensional re-ID features for “joint detection and re-ID” networks achieves both higher tracking accuracy and efficiency. This also reveals the difference between the MOT task and the re-ID task, which is overlooked in the field of MOT.

In this work, we present a simple approach termed as *FairMOT* which elegantly address the three issues as illustrated

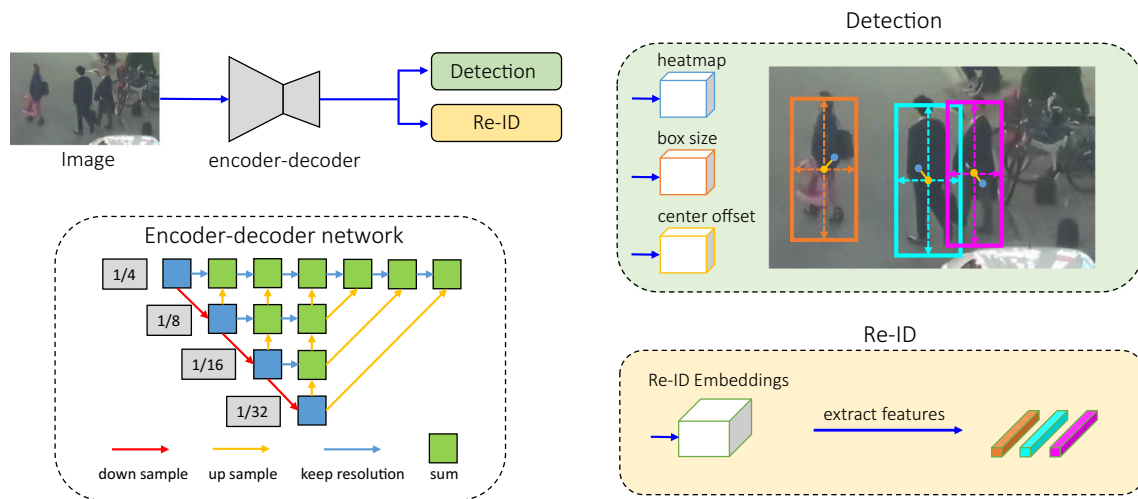
in Fig. 1. *FairMOT* is built on top of CenterNet (Zhou et al. 2019a). In particular, the detection and re-ID tasks are treated equally in *FairMOT* which essentially differs from the previous “detection first, re-ID secondary” framework. It is worth noting that it is not a naive combination of CenterNet and re-ID. Instead, we present a bunch of detailed designs which are critical to achieve good tracking results by thorough empirical studies.

Figure 1 shows an overview of *FairMOT*. It has a simple network structure which consists of two *homogeneous* branches for detecting objects and extracting re-ID features, respectively. Inspired by Zhou et al. (2019a, b), Law and Deng (2018), Duan et al. (2019), 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 two-stage cascaded style. So *FairMOT* eliminates the unfair disadvantage of the detection branch as reflected in Table 1, effectively learns high-quality re-ID features and obtains a good trade-off between detection and re-ID.

We evaluate *FairMOT* on the MOT Challenge benchmark via the evaluation server. It ranks first among all trackers on the 2DMOT15 (Leal-Taixé et al. 2015), MOT16 (Milan et al. 2016), MOT17 (Milan et al. 2016) and MOT20 (Dendorfer et al. 2020) datasets. When we further pre-train our model using our proposed single image training 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.

Our contributions are as follows:

- We empirically demonstrate that the prevalent anchor-based one-shot MOT architectures have limitations in terms of learning effective re-ID features which has been overlooked. The issues severely limit the tracking performance of those methods.
- We present *FairMOT* to address the fairness issue. *FairMOT* is built on top of CenterNet. Although the adopted techniques are mostly not novel by themselves, we have new discoveries which are important to MOT. These are both novel and valuable.
- We show that the achieved fairness allows our *FairMOT* to obtain high levels of detection and tracking accuracy and outperform the previous state-of-the-art methods by a large margin on multiple datasets such as 2DMOT15, MOT16, MOT17 and MOT20.



**Fig. 1** 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 detect-

ing objects and extracting re-ID features, respectively. The features at the predicted object centers are used for tracking

**Table 1** Comparison of different re-ID feature extraction (sampling) strategies on the validation set of MOT17

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)		<b>69.1</b>	72.8	<b>299</b>	94.4
FairMOT (Center-BI)		68.8	<b>74.3</b>	303	<b>94.9</b>
FairMOT (Two-Stage)	✓	69.0	68.2	388	90.5

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

## 2 Related Work

The best-performing MOT methods (Bergmann et al. 2019; Brasó and Leal-Taixé 2020; Hornakova et al. 2020; Yu et al. 2016; Mahmoudi et al. 2019; Zhou et al. 2018; Wojke et al. 2017; Chen et al. 2018a; Wang et al. 2020b; Voigtlaender et al. 2019) usually follow the tracking-by-detection paradigm, which first detect objects in each frame and then associate them over time. We classify the existing works into two categories based on whether they use a single model or separate models to detect objects and extract association features. We discuss the pros and cons of the methods and compare them to our approach.

### 2.1 Detection and Tracking by Separate Models

#### 2.1.1 Detection Methods

Most benchmark datasets such as MOT17 (Milan et al. 2016) provide detection results obtained by popular methods such as DPM (Felzenszwalb et al. 2008), Faster R-CNN (Ren et al. 2015) and SDP (Yang et al. 2016) such that the works that

focus on the tracking part can be fairly compared on the same object detections. Some works such as Yu et al. (2016), Wojke et al. (2017), Zhou et al. (2018), Mahmoudi et al. (2019) use a large private pedestrian detection dataset to train the Faster R-CNN detector with VGG-16 (Simonyan and Zisserman 2014) as backbone, which obtain better detection performance. A small number of works such as Han et al. (2020) use more powerful detectors which are developed recently such as Cascade R-CNN (Cai and Vasconcelos 2018) to boost the detection performance.

#### 2.1.2 Tracking Methods

Most of the existing works focus on the tracking part of the problem. We classify them into two classes according to the type of cues used for association.

**Location and Motion Cues based Methods** SORT (Bewley et al. 2016) first uses Kalman Filter (Kalman 1960) to predict future locations of the tracklets, computes their overlap with the detections, and uses Hungarian algorithm (Kuhn 1955) to assign detections to tracklets. IOU-Tracker (Bochinski et al. 2017) directly computes the overlap between the

tracklets (of the previous frame) and the detections without using Kalman filter to predict future locations. The approach achieves 100K fps inference speed (detection time not counted) and works well when object motion is small. Both SORT and IOU-Tracker are widely used in practice due to their simplicity.

However, they may fail in challenging cases of crowded scenes and fast motion. Some works such as Xiang et al. (2015), Zhu et al. (2018), Chu and Ling (2019), Chu et al. (2019) leverage sophisticated single object tracking methods to get accurate object locations and reduce false negatives. However, these methods are extremely slow especially when there are a large number of people in the scene. To solve the problem of trajectory fragments, Zhang et al. (2020) propose a motion evaluation network to learn long-range features of tracklets for association. MAT (Han et al. 2020) is an enhanced SORT, which additionally models the camera motion and uses dynamic windows for long-range re-association.

**Appearance Cues based Methods** Some recent works (Yu et al. 2016; Mahmoudi et al. 2019; Zhou et al. 2018; Wojke et al. 2017) propose to crop the image regions of the detections and feed them to re-ID networks (Zheng et al. 2017b; Hermans et al. 2017; Luo et al. 2019a) to extract image features. Then they compute the similarity between tracklets and detections based on re-ID features and use Hungarian algorithm (Kuhn 1955) to accomplish assignment. The method is robust to fast motion and occlusion. In particular, it can re-initialize lost tracks because appearance features are relatively stable over time.

There are also some works (Bae and Yoon 2014; Tang et al. 2017; Sadeghian et al. 2017; Chen et al. 2018a; Xu et al. 2019) focusing on enhancing appearance features. For example, Bae and Yoon (2014) propose an online appearance learning method to handle appearance variations. Tang et al. (2017) leverage body pose features to enhance the appearance features. Some methods (Sadeghian et al. 2017; Xu et al. 2019; Shan et al. 2020) propose to fuse multiple cues (i.e. motion, appearance and location) to get more reliable similarity. MOTDT (Chen et al. 2018a) proposes a hierarchical data association strategy which uses IoU to associate objects when appearance features are not reliable. A small number of works such as Mahmoudi et al. (2019), Zhou et al. (2018), Fang et al. (2018) also propose to use more complicated association strategies such as group models and RNNs.

**Offline Methods** The offline methods (or batch methods) (Zhang et al. 2008; Wen et al. 2014; Berclaz et al. 2011; Zamir et al. 2012; Milan et al. 2013; Choi 2015; Brasó and Leal-Taixé 2020; Hornakova et al. 2020) often achieve better results by performing global optimization in the whole sequence. For example, Zhang et al. (2008) build a graphical model with nodes representing detections in all frames. The

optimal assignment 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. (2011) 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. (2013) 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. MPNTrack (Brasó and Leal-Taixé 2020) proposes trainable graph neural networks to perform a global association of the entire set of detections and make MOT fully differentiable. Lif\_T (Hornakova et al. 2020) formulates MOT as a lifted disjoint path problem and introduces lifted edges for long range temporal interactions, which significantly reduces id switches and re-identify lost.

**Advantages and Limitations** For the methods which perform detection and tracking by separate models, the main advantage 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 (Yu et al. 2016; Henschel et al. 2019) 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.

## 2.2 Detection and Tracking by a Single Model

With the quick maturity of multi-task learning (Kokkinos 2017; Ranjan et al. 2017; Sener and Koltun 2018) in deep learning, joint detection and tracking using a single network has begun to attract more research attention. We classify them into two classes as discussed in the following.

**Joint Detection and Re-ID** The first class of methods (Voigtlaender et al. 2019; Wang et al. 2020b; Liang et al. 2020; Pang et al. 2021; Lu et al. 2020) perform object detection and re-ID feature extraction in a single network in order to reduce inference time. For example, Track-RCNN (Voigtlaender et al. 2019) adds a re-ID head on top of Mask R-CNN (He et al. 2017) and regresses a bounding box and a re-ID feature for each proposal. Similarly, JDE (Wang et al. 2020b) is built on top of YOLOv3 (Redmon and Farhadi 2018) which achieves near video rate inference. However, the accuracy of these one-shot trackers is usually lower than that of the two-step ones.

**Joint Detection and Motion Prediction** The second class of methods (Feichtenhofer et al. 2017; Zhou et al. 2020; Pang



et al. 2020; Peng et al. 2020) learn detection and motion features in a single network. D&T (Feichtenhofer et al. 2017) propose a Siamese network which takes input of adjacent frames and predicts inter-frame displacements between bounding boxes. Tracktor (Bergmann et al. 2019) directly exploits the bounding box regression head to propagate identities of region proposals and thus removes box association. Chained-Tracker (Peng et al. 2020) proposes an end-to-end model using adjacent frame pair as input and generating the box pair representing the same target. These box-based methods assume that bounding boxes have a large overlap between frames, which is not true in low-frame rate videos. Different from these methods, CenterTrack (Zhou et al. 2020) predicts the object center displacements with pair-wise inputs and associate by these point distances. It also provides the tracklets as an additional point-based heatmap input to the network and is then able to match objects anywhere even if the boxes have no overlap at all. However, these methods only associate objects in adjacent frames without re-initializing lost tracks and thus have difficulty handling occlusion cases.

Our work belongs to the first class. We investigate the reasons why one-shot trackers get degraded association performance and propose a simple approach to address the problems. We show that the tracking accuracy is improved significantly without heavy engineering efforts. A concurrent work CStrack (Liang et al. 2020) also aims to alleviate the conflicts between the two tasks from the perspective of features, and propose a cross-correlation network module to enable the model to learn task-dependent representations. Different from CStrack, our method tries to address the problem from three perspectives in a systematic way and obtains notably better performances than CStrack. CenterTrack (Zhou et al. 2020) is also related to our work since it also uses center-based object detection framework. But CenterTrack does not extract appearance features and only links objects in adjacent frames. In contrast, *FairMOT* can perform long-range association with the appearance features and handle occlusion cases.

**Multi-task Learning** There is a large body of literature (Liu et al. 2019; Kendall et al. 2018; Chen et al. 2018b; Guo et al. 2018; Sener and Koltun 2018) on multi-task learning which may be used to balance the object detection and re-ID feature extraction tasks. Uncertainty (Kendall et al. 2018) uses task-dependent uncertainty to automatically balance the single-task losses. MGDA is proposed in Sener and Koltun (2018) to update the shared network weights by finding a common direction among the task-specific gradients. Grad-Norm (Chen et al. 2018b) controls the training of multi-task networks by simulating the task-specific gradients to be of similar magnitude. We evaluate these methods in the experimental sections.

## 2.3 Video Object Detection

Video Object Detection (VOD) (Feichtenhofer et al. 2017; Luo et al. 2019b) is related to MOT in the sense that it leverages tracking to improve object detection performances in challenging frames. Although these methods were not evaluated on MOT datasets, some of the ideas may be valuable for the field. So we briefly review them in this section. Tang et al. (2019) 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 Han et al. (2016), Kang et al. (2016, 2017), Tang et al. (2019), Pang et al. (2020). One main limitation of these tube-based methods is that they are extremely slow especially when there are a large number of objects in videos.

## 3 Unfairness Issues in One-Shot Trackers

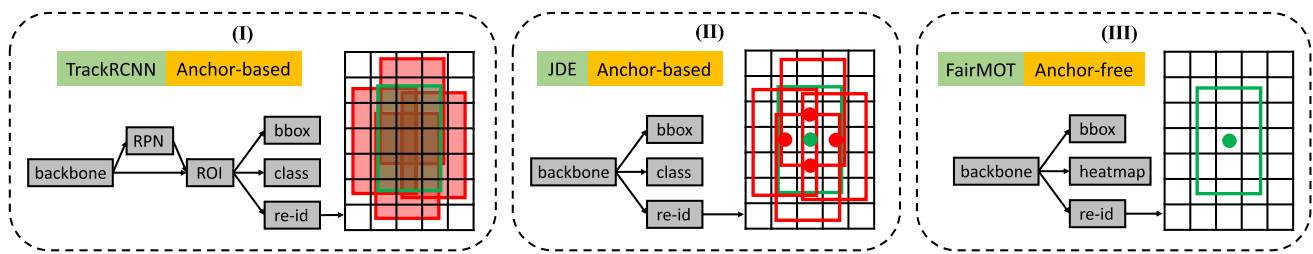
In this section, we discuss three unfairness issues that arise in the existing one-shot trackers which usually lead to degraded tracking performances.

### 3.1 Unfairness Caused by Anchors

The existing one-shot trackers such as Track R-CNN (Voigtlaender et al. 2019) and JDE (Wang et al. 2020b) are mostly anchor-based since they are directly modified from anchor-based object detectors such as YOLO (Redmon and Farhadi 2018) and Mask R-CNN (He et al. 2017). However, we find that the anchor-based design is not suitable for learning re-ID features which result in a large number of ID switches in spite of the good detection results. We explain the problem from three perspectives in the following.

**Overlooked re-ID task** Track R-CNN (Voigtlaender et al. 2019) operates in a cascaded style which first estimates object proposals (boxes) and then pools features from them to estimate the corresponding re-ID features. The quality of re-ID features heavily depends on the quality of proposals during training (re-ID features are useless if proposals are not accurate). As a result, in the training stage, the model is seriously biased to estimate accurate object proposals rather than high quality re-ID features. So the standard “detection first, re-ID secondary” design of the existing one-shot trackers makes the re-ID network not fairly learned.

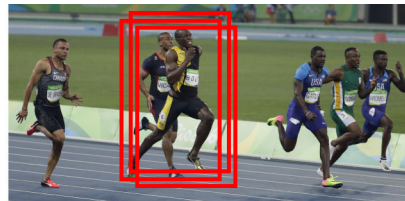
**One anchor corresponds to multiple identities** The anchor-based methods usually use ROI-Align to extract features from proposals. Most sampling locations in ROI-Align may belong to other disturbing instances or background as shown in Fig. 2. As a result, the extracted features are not optimal



(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. 2** **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**, **c**

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 a single point, i.e. the estimated object centers.

**Multiple anchors correspond to one identity** In both Voigtlaender et al. (2019) and Wang et al. (2020b), 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 Fig. 2 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.

### 3.2 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 features to estimate object classes and positions but re-ID requires low-level appearance features to distinguish differ-

ent instances of the same class. From the perspective of the multi-task loss optimization, the optimization objectives of detection and re-ID have conflicts. Thus, it is important to balance the loss optimization strategy of the two tasks.

### 3.3 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) high-dimensional re-ID features 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) the MOT task is different from the re-ID task. The MOT task only performs a small number of one-to-one matchings between two consecutive frames. The re-ID task needs to match the query to a large number of candidates and thus requires more discriminative and high-dimensional re-ID features. So in MOT we do not need that high-dimensional features; (3) learning low dimensional re-ID features improves the inference speed as will be shown in our experiments.

## 4 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.

### 4.1 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) (Zhou et al. 2019a) is applied to the backbone to fuse multi-layer features as shown in Fig. 1. Different from original DLA (Yu et al. 2018), it has more skip connections between low-level and high-level features which is similar to the Feature Pyramid Network (FPN) (Lin et al. 2017a). 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 (Cheng et al. 2020), can be used in our framework to provide fair features for both detection and re-ID.

### 4.2 Detection Branch

Our detection branch is built on top of CenterNet (Zhou et al. 2019a) but other anchor-free methods such as Duan et al. (2019), Law and Deng (2018), Dong et al. (2020), Yang et al. (2019) 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  $3 \times 3$  convolution (with 256 channels) to the output features of DLA-34, followed by a  $1 \times 1$  convolutional layer which generates the final targets.

#### 4.2.1 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  $(\tilde{c}_x^i, \tilde{c}_y^i) = (\lfloor \frac{c_x^i}{4} \rfloor, \lfloor \frac{c_y^i}{4} \rfloor)$ . Then the heatmap response at the location  $(x, y)$  is computed as  $M_{xy} = \sum_{i=1}^N \exp \frac{-(x - \tilde{c}_x^i)^2 + (y - \tilde{c}_y^i)^2}{2\sigma_c^2}$  where  $N$  represents the number of objects in the image and  $\sigma_c$  represents the standard deviation. The loss function is defined as pixel-wise logistic regression with focal loss (Lin et al. 2017b):

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

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

#### 4.2.2 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  $\hat{S} \in \mathbb{R}^{2 \times H \times W}$  and  $\hat{O} \in \mathbb{R}^{2 \times H \times W}$ , 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:

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

where  $\lambda_s$  is a weighting parameter and is set 0.1 as the original CenterNet (Zhou et al. 2019a).

### 4.3 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  $\mathbf{E} \in \mathbb{R}^{128 \times H \times W}$ . The re-ID feature  $\mathbf{E}_{x,y} \in \mathbb{R}^{128}$  of an object centered at  $(x, y)$  can be extracted from the feature map.

### 4.3.1 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  $(\tilde{c}_x^i, \tilde{c}_y^i)$ . We extract the re-ID feature vector  $\mathbf{E}_{\tilde{c}_x^i, \tilde{c}_y^i}$  and use a fully connected layer and a softmax operation 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:

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

where  $K$  is the number of all the identities in the training data. 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.

### 4.4 Training FairMOT

We jointly train the detection and re-ID branches by adding the losses (i.e., Eqs. (1), (2) and (3)) together. In particular, we use the uncertainty loss proposed in Kendall et al. (2018) to automatically balance the detection and re-ID tasks:

$$L_{\text{detection}} = L_{\text{heat}} + L_{\text{box}}, \quad (4)$$

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

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 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 single image training method to train *FairMOT* on image-level object detection datasets such as COCO (Lin et al. 2014) and CrowdHuman (Shao et al. 2018). Different from CenterTrack (Zhou et al. 2020) that takes two simulated consecutive frames as input, we only take a single image as input. We assign each bounding box a unique identity and thus regard each object instance in the dataset as a separate class. We apply different transformations to the whole image including HSV augmentation, rotation, scaling, translation and shearing. The single image training method has significant empirical values. First, the pre-trained model

on the CrowdHuman dataset can be directly used as a tracker and get acceptable results on MOT datasets such as MOT17 (Milan et al. 2016). This is because the CrowdHuman dataset can boost the human detection performance and also has strong domain generalization ability. Our training of the re-ID features further enhances the association ability of the tracker. Second, we can finetune it on other MOT datasets and further improve the final performance.

### 4.5 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.

#### 4.5.1 Network Inference

The network takes a frame of size  $1088 \times 608$  as input which is the same as the previous work JDE (Wang et al. 2020b). On top of the predicted heatmap, we perform non-maximum suppression (NMS) based on the heatmap scores to extract the peak keypoints. The NMS is implemented by a simple  $3 \times 3$  max pooling operation as in Zhou et al. (2019a). 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.

#### 4.5.2 Online Association

We follow MOTDT (Chen et al. 2018a) and use a hierarchical online data association method. We first initialize a number of tracklets based on the detected boxes in the first frame. Then in the subsequent frame, we link the detected boxes to the existing tracklets using a two-stage matching strategy. In the first stage, we use Kalman Filter (Kalman 1960) and re-ID features to obtain initial tracking results. In particular, we use Kalman Filter to predict tracklet locations in the following frame and compute the Mahalanobis distance  $D_m$  between the predicted and detected boxes following DeepSORT (Wojke et al. 2017). We fuse the Mahalanobis distance with the cosine distance computed on re-ID features:  $D = \lambda D_r + (1 - \lambda) D_m$  where  $\lambda$  is a weighting parameter and is set to be 0.98 in our experiments. Following JDE (Wang et al. 2020b), we set Mahalanobis distance to infinity if it is larger than a threshold to avoid getting trajectories with large motion. We use Hungarian algorithm (Kuhn 1955) with a matching threshold  $\tau_1 = 0.4$  to complete the first stage matching.



In the second stage, for unmatched detections and tracklets, we try to match them according to the overlap between their boxes. In particular, we set the matching threshold  $\tau_2 = 0.5$ . We update the appearance features of the tracklets in each time step to handle appearance variations as in Bolme et al. (2010), Henriques et al. (2014). Finally, we initialize the unmatched detections as new tracks and save the unmatched tracklets for 30 frames in case they reappear in the future.

## 5 Experiments

### 5.1 Datasets and Metrics

There are six training datasets briefly introduced as follows: the ETH (Ess et al. 2008) and CityPerson (Zhang et al. 2017) datasets only provide box annotations so we only train the detection branch on them. The CalTech (Dollár et al. 2009), MOT17 (Milan et al. 2016), CUHK-SYSU (Xiao et al. 2017) and PRW (Zheng et al. 2017a) 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 MOT17 which are removed from the training dataset for fair comparison. The overall training strategy is described in Sect. 4.4, which is the same as Wang et al. (2020b). For the self-supervised training of our method, we use the Crowd-Human dataset (Shao et al. 2018) which only contains object bounding box annotations.

We evaluate our approach on the testing sets of four benchmarks: 2DMOT15, MOT16, MOT17 and MOT20. We use Average Precision (AP) to evaluate detection results. Following Wang et al. (2020b), we use True Positive Rate (TPR) at a false accept rate of 0.1 for evaluating re-ID features. In particular, we extract re-ID features which correspond to ground truth boxes and use each feature to retrieve  $N$  most similar candidates. We report the true positive rate at false accept rate 0.1 ( $TPR@FAR=0.1$ ). Note that  $TPR$  is not affected by detection results and faithfully reflects the quality of re-ID features. We use the CLEAR metric (Bernardin and Stiefelhagen 2008) (i.e.  $MOTA$ ,  $IDs$ ) and  $IDFI$  (Ristani et al. 2016) to evaluate overall tracking accuracy.

### 5.2 Implementation Details

We use a variant of DLA-34 proposed in Zhou et al. (2019a) as our default backbone. The model parameters pre-trained on the COCO dataset (Lin et al. 2014) are used to initialize our model. We train our model with the Adam optimizer (Kingma and Ba 2014) for 30 epochs with a starting learning rate of  $10^{-4}$ . The learning rate decays to  $10^{-5}$  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  $1088 \times 608$  and the feature map resolution is  $272 \times 152$ . The training step takes about 30 hours on two RTX 2080 Ti GPUs.

### 5.3 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.

#### 5.3.1 Anchors

We evaluate four strategies for sampling re-ID features from the detected boxes which are frequently used by previous works (Wang et al. 2020b; Voigtlaender et al. 2019). The first strategy is ROI-Align used in Track R-CNN (Voigtlaender et al. 2019). 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 (Wang et al. 2020b). 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 (He et al. 2017) to extract the backbone features based on the detected bounding boxes and then use a re-ID head (a fully connected layer) to get re-ID features. The main difference between the two-stage approach and the one-stage “ROI-Align” approach is that the re-ID features of the two-stage approach rely on the detection results while those of the one-stage approach do not during training.

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  $IDFI$  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

**Table 2** Comparison of different loss weighting strategies on the validation set of the MOT17 dataset

Loss Weighting	MOTA↑	IDF1↑	IDs↓	AP↑	TPR↑
Fixed	<b>69.6</b>	71.6	387	<b>81.9</b>	93.8
Uncertainty-task	69.1	72.8	<b>299</b>	81.2	94.4
Uncertainty-branch	68.5	73.3	319	81.0	96.0
MGDA-UB	63.6	67.9	355	78.5	<b>97.0</b>
GradNorm	69.5	<b>73.8</b>	311	81.3	95.1

The best results are shown in bold

it samples features at more accurate locations. The two-stage approach harms the quality of the re-ID features.

### 5.3.2 Balancing Multi-task Losses

We evaluate different methods for balancing the losses of different tasks including Uncertainty (Kendall et al. 2018), GradNorm (Chen et al. 2018b) and MGDA-UB (Sener and Koltun 2018). We also evaluate a baseline with fixed weights obtained by grid search. We implement two versions for the uncertainty-based method. The first is “Uncertainty-task” which learns two parameters for the detection loss and re-ID loss, respectively. The second is “Uncertainty-branch” which learns four parameters for the heatmap loss, box size loss, offset loss and re-ID losses, respectively.

The results are shown in Table 2. We can see that the “Fixed” method gets the best *MOTA* and *AP* but the worst *IDs* and *TPR*. It means that the model is biased to the detection task. MGDA-UB has the highest *TPR* but the lowest *MOTA* and *AP*, which indicates that the model is biased to the re-ID task. Similar results can be found in Wang et al. (2020b), Vandenhenne et al. (2021). GradNorm gets the best overall tracking accuracy (highest *IDF1* and second highest *MOTA*), meaning that ensuring different tasks to have similar gradient magnitude is helpful to handle feature conflicts. However, GradNorm takes longer training time. So we use the simpler Uncertainty method which is slightly worse than GradNorm in the rest of our experiments.

### 5.3.3 Multi-layer Feature Fusion

We compare a number of backbones such as vanilla ResNet (He et al. 2016), Feature Pyramid Network (FPN) (Lin et al. 2017a), High-Resolution Network (HRNet) (Wang et al. 2020a), DLA (Zhou et al. 2019a), HarDNet (Chao et al. 2019) and RegNet (Radosavovic et al. 2020). 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 and RegNet to obtain feature maps of stride four.

We divide these backbones into two categories, one without multi-layer fusion (i.e. ResNet and RegNet) and one with (i.e. FPN, HRNet, DLA and HarDNet).

The results are shown in Table 3. We also list the ImageNet (Russakovsky et al. 2015) classification accuracy *Acc* in order to demonstrate that a strong backbone in one task does not mean it will also get good results in MOT. So detailed studies for MOT are necessary and useful.

By comparing the results of ResNet-34 and ResNet-50, we find that blindly using a larger network does not notably improve 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. By comparing ResNet-50 and RegNetY-4.0GF, we can find that using a even more powerful backbone also achieves very limited gain. The re-ID metric *TPR* of RegNetY-4.0GF is the same as ResNet-50 (91.9) while the ImageNet classification accuracy improves a lot (79.4 vs. 77.8). All these results suggest that directly using a larger or a more powerful network cannot always improve 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%. By comparing RegNetY-4.0GF-FPN with RegNetY-4.0GF, we can see that adding multi-layer feature fusion structure (Lin et al. 2017a) to RegNet brings considerable gains (+1.9 *MOTA*, +1.3 *IDF1*, -36.9% *IDs*, +2.2 *AP*, +2.3 *TPR*), which suggests that multi-layer feature fusion has clear advantages over simply using larger or more powerful 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. Similar conclusions can be obtained from the results of HRNet-W18. The results validate that feature fusion (FPN, DLA and HRNet) 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 5, we can see that DLA-34 mainly outperforms HRNet-W18 on middle and large size objects. When we further use a more

**Table 3** Comparison of different backbones on the validation set of MOT17 dataset

Backbone	w/ MLFF	MOTA↑	IDF1↑	IDs↓	AP↑	TPR↑	Acc↑
ResNet-34		63.6	67.2	435	75.1	90.9	75.2
ResNet-50		63.7	67.7	501	75.5	91.9	77.8
RegNetY-4.0GF		63.9	68.0	407	75.8	91.9	<b>79.4</b>
ResNet-34-FPN	✓	64.4	69.6	369	77.7	94.2	75.2
RegNetY-4.0GF-FPN	✓	65.8	69.3	257	78.0	94.3	<b>79.4</b>
HRNet-W18	✓	67.4	74.3	315	80.5	94.6	76.8
DLA-34	✓	69.1	72.8	299	81.2	94.4	76.9
HarDNet-85	✓	<b>71.2</b>	<b>74.5</b>	<b>198</b>	<b>82.6</b>	<b>95.8</b>	77.0

“MLFF” is short for multi-layer feature fusion. “Acc” is short for ImageNet classification accuracy. The results of the ImageNet classification accuracy are from the original papers of the backbone networks. The best results are shown in bold

**Table 4** Demonstration of *feature conflict* between the detection and re-ID tasks on the validation set of the MOT17 dataset

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	<b>69.1</b>	<b>72.8</b>	<b>299</b>	<b>81.2</b>	<b>94.4</b>

“-det” means only the detection branch is trained and the re-ID branch is randomly initialized. The best results are shown in bold

powerful backbone HarDNet-85 with more multi-layer feature fusion structures, we achieve even better results than DLA-34 (+2.1 MOTA, +1.7 IDF1, -33.8% IDs, +1.4 AP, +1.4 TPR). Although HRNet-W18, DLA-34 and HarDNet-85 get lower ImageNet classification accuracy than ResNet-50 and RegNetY-4.0GF, they achieve much higher tracking accuracy. Based on the experimental results above, we believe that multi-layer feature fusion is the key to solve the “feature” issue.

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 4 that the detection result measured by AP improves by 1 point 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 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.

### 5.3.4 Feature Dimension

The previous one-shot trackers such as JDE (Wang et al. 2020b) usually learns 512 dimensional re-ID features follow-

ing 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 conduct experiments on different one-shot trackers and find it is a generic rule that low-dimensional (i.e. 64) re-ID features achieves better performance than high-dimensional (i.e. 512) re-ID features.

We evaluate multiple choices for re-ID feature dimension of JDE and *FairMOT* in Table 6. For JDE, we can see that 64 achieves better performance than 512 on all the metrics. For *FairMOT*, we can see that 512 achieves higher *IDF1* and *TPR* scores which indicates that higher dimensional re-ID features lead to stronger discriminative ability. However, the MOTA score 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. Different from the re-ID task, low-dimensional re-ID features achieves better performance and efficiency on the MOT task.

### 5.3.5 Data Association Methods

This section evaluates the three ingredients in the data association step including bounding box IoU, re-ID features and Kalman Filter (Kalman 1960). These are used to compute the similarity between each pair of detected boxes. With that we use Hungarian algorithm (Kuhn 1955) to solve the assignment problem. Table 7 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

**Table 5** The impact of different backbones on objects of different scales

Backbone	AP <sup>S</sup>	AP <sup>M</sup>	AP <sup>L</sup>	TPR <sup>S</sup>	TPR <sup>M</sup>	TPR <sup>L</sup>	IDs <sup>S</sup>	IDs <sup>M</sup>	IDs <sup>L</sup>
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	<b>93.3</b>	166	71	90
HRNet-W18	<b>51.1</b>	63.7	85.7	<b>94.2</b>	<b>92.5</b>	93.1	168	<b>55</b>	<b>56</b>
DLA-34	46.8	<b>65.1</b>	<b>88.8</b>	92.7	91.2	91.8	<b>134</b>	64	70

*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

**Table 6** Evaluation of re-ID feature dimensions of JDE and FairMOT on the validation set of MOT17

Method	Dim	MOTA ↑	IDF1 ↑	IDs ↓	AP ↑	TPR ↑	FPS ↑
JDE	512	59.9	64.1	536	73.3	76.8	22.2
JDE	64	<b>60.3</b>	<b>65.0</b>	<b>474</b>	<b>73.6</b>	<b>82.0</b>	<b>24.4</b>
FairMOT	512	68.5	<b>73.7</b>	312	80.9	<b>94.6</b>	24.1
FairMOT	64	<b>69.2</b>	73.3	<b>283</b>	<b>81.3</b>	94.3	<b>26.8</b>

The best results of the same method are shown in bold

**Table 7** Evaluation of the three ingredients in the data association model

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
✓	✓	✓	<b>69.1</b>	<b>72.8</b>	<b>299</b>

The backbone is DLA-34 The best results are shown in bold

is important to leverage box IoU, re-ID features and Kalman filter to obtain good tracking performance.

We also present a detailed runtime breakdown of different components including detection, re-ID matching, Kalman Filter and IoU matching. We test runtime on sequences with different density (average number of pedestrians per frame). The results are shown in Fig. 4. The time spent on joint detection and re-ID is minimally affected by density. The time spent on Kalman Filter and IoU matching are around 1 ms or 2 ms and can be ignored. The time spent on re-ID matching increases linearly with the increase of density. This is because a large amount of time is cost on updating the re-ID feature of each tracklet.

### 5.3.6 Visualization of Re-ID Similarity

We use re-ID similarity maps to demonstrate the discriminative ability of re-ID features in Fig. 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 Sects. 5.3.1 and 5.3.3 respectively. By comparing the similarity maps of ResNet-34 and ResNet-34-det, we can see that training the re-ID branch

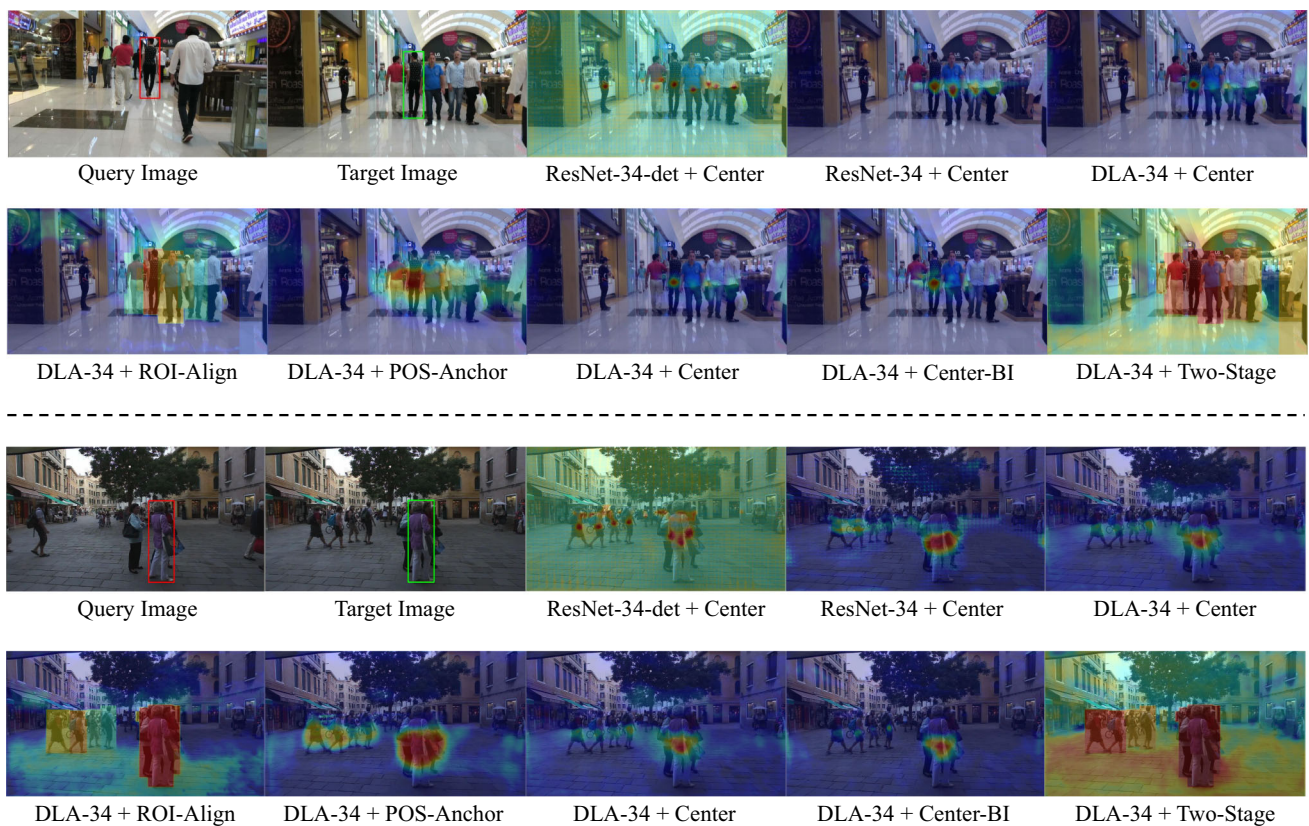
is important. By comparing DLA-34 and ResNet-34, we can see that multi-layer feature aggregation can get more discriminative re-ID features. Among all the sampling strategies, the proposed Center and Center-BI can better discriminate the target object from surrounding objects in crowded scenes.

## 5.4 Single Image Training

We first pre-train *FairMOT* on the CrowdHuman dataset (Shao et al. 2018). In particular, we assign a unique identity label for each bounding box and train *FairMOT* using the method described in Sect. 4.4. Then we finetune the pre-trained model on the target dataset MOT17.

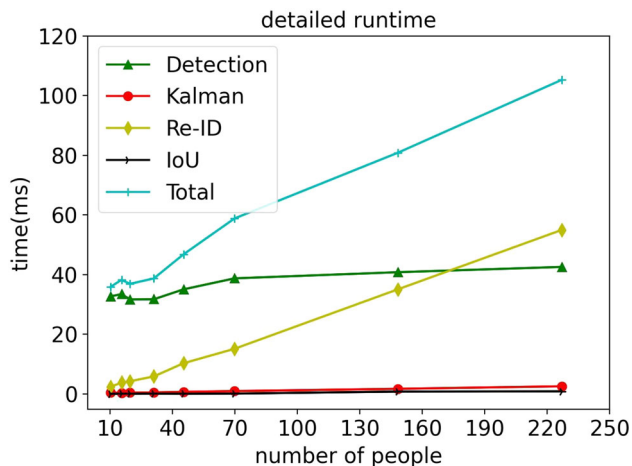
Table 8 shows the results. First, the pre-trained model can be directly used as a tracker and get acceptable results on MOT datasets such as MOT17. This is because the CrowdHuman dataset can boost the human detection performance and also has strong domain generalization ability. Our training of the re-ID features further enhances the association ability of the tracker. Second, pre-training on CrowdHuman outperforms directly training on the MOT17 dataset by a large margin. Third, the single image training model even outperforms the model trained on the “MIX” and MOT17 datasets with identity annotations. The results validate the effectiveness of the proposed single image pre-training, which saves





**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 Sect. 5.3.1) 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



**Fig. 4** Time spent on different parts of our whole MOT system. We run tracking on sequences with different density from the MOT17 dataset and the MOT20 dataset

lots of annotation efforts and makes *FairMOT* more attractive in real applications.

**Table 8** Effects of single image training on the validation set of MOT17

Training Data	MOTA $\uparrow$	IDF1 $\uparrow$	IDs $\downarrow$	AP $\uparrow$	TPR $\uparrow$
CH*	64.1	64.9	476	80.5	79.9
MOT17	67.5	69.9	408	79.6	93.4
CH*+MOT17	<b>71.1</b>	<b>75.6</b>	327	<b>83.0</b>	93.6
MIX+MOT17	69.1	72.8	<b>299</b>	81.2	<b>94.4</b>

“CH” and “MIX” stand for CrowdHuman and the composed five datasets introduced in Sect. 5.1, respectively. \*Means no identity annotations are used. The best results are shown in bold

## 5.5 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.

### 5.5.1 Comparing with One-Shot SOTA MOT Methods

There are two published works of JDE (Wang et al. 2020b) and TrackRCNN (Voigtlaender et al. 2019) that jointly per-

**Table 9** Comparison of the state-of-the-art one-shot trackers on the 2DMOT15 dataset

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

“MIX” represents the large scale training dataset and “MOT17 Seg” stands for the 4 videos with segmentation labels in the MOT17 dataset. The best results of the same training data are shown in bold

**Table 10** Comparison of the state-of-the-art methods under the “private detector” protocol

Dataset	Tracker	MOTA↑	IDF1↑	MT↑ (%)	ML↓ (%)	IDs↓	FPS↑
MOT15	MDP_SubCNN (Xiang et al. 2015)	47.5	55.7	30.0	18.6	628	<1.7
	CDA_DDAL (Bae and Yoon 2017)	51.3	54.1	36.3	22.2	544	<1.2
	EAMTT (Sanchez-Matilla et al. 2016)	53.0	54.0	35.9	19.6	7538	<4.0
	AP_HWDPL (Chen et al. 2017)	53.0	52.2	29.1	20.2	708	6.7
	RAR15 (Fang et al. 2018)	56.5	61.3	45.1	14.6	<b>428</b>	<3.4
	TubeTK* (Pang et al. 2020)	58.4	53.1	39.3	18.0	854	5.8
	FairMOT (Ours)*	<b>60.6</b>	<b>64.7</b>	<b>47.6</b>	<b>11.0</b>	591	<b>30.5</b>
MOT16	EAMTT (Sanchez-Matilla et al. 2016)	52.5	53.3	19.9	34.9	910	<5.5
	SORTwHPD16 (Bewley et al. 2016)	59.8	53.8	25.4	22.7	1423	<8.6
	DeepSORT_2 (Wojke et al. 2017)	61.4	62.2	32.8	18.2	781	<6.4
	RAR16wVGG (Fang et al. 2018)	63.0	63.8	39.9	22.1	<b>482</b>	<1.4
	VMaxx (Wan et al. 2018)	62.6	49.2	32.7	21.1	1389	<3.9
	TubeTK* (Pang et al. 2020)	64.0	59.4	33.5	19.4	1117	1.0
	JDE* (Wang et al. 2020b)	64.4	55.8	35.4	20.0	1544	18.5
	TAP (Zhou et al. 2018)	64.8	<b>73.5</b>	38.5	21.6	571	<8.0
	CNNMTT (Mahmoudi et al. 2019)	65.2	62.2	32.4	21.3	946	<5.3
	POI (Yu et al. 2016)	66.1	65.1	34.0	20.8	805	<5.0
MOT17	CTrackerV1* (Peng et al. 2020)	67.6	57.2	32.9	23.1	1897	6.8
	FairMOT (Ours)*	<b>74.9</b>	72.8	<b>44.7</b>	<b>15.9</b>	1074	<b>25.9</b>
	SST (Sun et al. 2019)	52.4	49.5	21.4	30.7	8431	<3.9
	TubeTK* (Pang et al. 2020)	63.0	58.6	31.2	19.9	4137	3.0
	CTrackerV1* (Peng et al. 2020)	66.6	57.4	32.2	24.2	5529	6.8
	CenterTrack* (Zhou et al. 2020)	67.8	64.7	34.6	24.6	<b>2583</b>	17.5
	FairMOT (Ours)*	<b>73.7</b>	<b>72.3</b>	<b>43.2</b>	<b>17.3</b>	3303	<b>25.9</b>
	FairMOT (Ours)*	<b>61.8</b>	<b>67.3</b>	<b>68.8</b>	<b>7.6</b>	<b>5243</b>	<b>13.2</b>
	FairMOT (Ours)*	<b>61.8</b>	<b>67.3</b>	<b>68.8</b>	<b>7.6</b>	<b>5243</b>	<b>13.2</b>
	FairMOT (Ours)*	<b>61.8</b>	<b>67.3</b>	<b>68.8</b>	<b>7.6</b>	<b>5243</b>	<b>13.2</b>

It is noteworthy that FPS considers both detection and association time. The one-shot trackers are labeled by “\*”. The best results of each dataset are shown in bold

form object detection and identity feature embedding. We compare our approach to both of them. Following the previous work (Wang et al. 2020b), 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 Sect. 5.1. 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 use the 4 videos to train our model. The CLEAR metric

(Bernardin and Stiefelhagen 2008) and IDF1 (Ristani et al. 2016) are used to measure their performance.

The results are shown in Table 9. We can see that our approach remarkably outperforms JDE (Wang et al. 2020b). 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 (Voigtlaender et al. 2019), their detection results are slightly better than ours (with lower





**Fig. 5** 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 11** Results of the MOT17 test set when using different datasets for training

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	<b>73.2</b>	3345
MOT17+MIX+CH	73K	740K	8.7K	<b>73.7</b>	72.3	<b>3303</b>

“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

*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.

### 5.5.2 Comparing with Other SOTA MOT Methods

We compare our approach to the state-of-the-art trackers including the two-step methods in Table 10. 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 (Fang et al. 2018; Yu et al. 2016) are usually slower than ours. Our approach also ranks second under a very recent local MOT metric ALTA (Valmadre et al. 2021), which further indicates that our approach achieves very high tracking performance (Table 10).

### 5.5.3 Training Data Ablation Study

We also evaluate the performance of *FairMOT* using different amount of training data (Table 11). 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 (Wang et al. 2020b), we can achieve 72.9 *MOTA*, which remarkably outperforms JDE. In addition, when we perform single image training 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.

## 5.6 Qualitative Results

Figure 5 visualizes several tracking results of *FairMOT* on the test set of MOT17 (Milan et al. 2016). 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 (Bewley et al. 2016; Bochinski et al. 2017) 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. Our method can detect small objects accurately as shown in the results of MOT17-07 and MOT17-14.

## 6 Summary and Future Work

Start from studying why the previous one-shot methods (Wang et al. 2020b) 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 dimension 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 single image 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 (Nos. 61733007 and 61876212) and MSRA Collaborative Research Fund. We thank all the anonymous reviewers for their valuable suggestions.



## References

- Bae, S. H., & Yoon, K. J. (2014). 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* (pp. 1218–1225).
- Bae, S. H., & Yoon, K. J. (2017). Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3), 595–610.
- Berclaz, J., Fleuret, F., Turetken, E., & Fua, P. (2011). Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9), 1806–1819.
- Bergmann, P., Meinhardt, T., & Leal-Taixe, L. (2019). Tracking without bells and whistles. In *ICCV* (pp. 941–951).
- Bernardin, K., & Stiefelhagen, R. (2008). Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008, 1–10.
- Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016). Simple online and realtime tracking. In *ICIP* (pp. 3464–3468). IEEE.
- Bochinski, E., Eiselein, V., & Sikora, T. (2017). High-speed tracking-by-detection without using image information. In *2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS)* (pp. 1–6). IEEE.
- Bolme, D. S., Beveridge, J. R., Draper, B. A., & Lui, Y. M. (2010). Visual object tracking using adaptive correlation filters. In *CVPR* (pp. 2544–2550). IEEE.
- Brasó, G., & Leal-Taixé, L. (2020). Learning a neural solver for multiple object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6247–6257).
- Cai, Z., & Vasconcelos, N. (2018). Cascade r-cnn: Delving into high quality object detection. In *CVPR* (pp. 6154–6162).
- Chao, P., Kao, C. Y., Ruan, Y. S., Huang, C. H., & Lin, Y. L. (2019). Hardnet: A low memory traffic network. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3552–3561).
- Chen, L., Ai, H., Shang, C., Zhuang, Z., & Bai, B. (2017). Online multi-object tracking with convolutional neural networks. In *2017 IEEE international conference on image processing (ICIP)* (pp. 645–649). IEEE.
- Chen, L., Ai, H., Zhuang, Z., & Shang, C. (2018a). Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *2018 IEEE international conference on multimedia and expo (ICME)* (pp. 1–6). IEEE.
- Chen, Z., Badrinarayanan, V., Lee, C. Y., & Rabinovich, A. (2018b). GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML, PMLR* (pp. 794–803).
- Cheng, B., Xiao, B., Wang, J., Shi, H., Huang, T. S., & Zhang, L. (2020). Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *CVPR*.
- Choi, W. (2015). Near-online multi-target tracking with aggregated local flow descriptor. In *Proceedings of the IEEE international conference on computer vision* (pp. 3029–3037).
- Chu, P., Fan, H., Tan, C. C., & Ling, H. (2019). Online multi-object tracking with instance-aware tracker and dynamic model refreshment. In *2019 IEEE winter conference on applications of computer vision (WACV)* (pp. 161–170). IEEE.
- Chu, P., & Ling, H. (2019). Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In *ICCV* (pp. 6172–6181).
- Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., & Leal-Taixé, L. (2020). Mot20: A benchmark for multi object tracking in crowded scenes. arXiv preprint [arXiv:2003.09003](https://arxiv.org/abs/2003.09003).
- Dollár, P., Wojek, C., Schiele, B., & Perona, P. (2009). Pedestrian detection: A benchmark. In *CVPR* (pp. 304–311). IEEE.
- Dong, Z., Li, G., Liao, Y., Wang, F., Ren, P., & Qian, C. (2020). Centripetalnet: Pursuing high-quality keypoint pairs for object detection. In *CVPR* (pp. 10519–10528).
- Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). Centernet: Keypoint triplets for object detection. In *ICCV* (pp. 6569–6578).
- Ess, A., Leibe, B., Schindler, K., & Van Gool, L. (2008). A mobile vision system for robust multi-person tracking. In *CVPR* (pp. 1–8). IEEE.
- Fang, K., Xiang, Y., Li, X., & Savarese, S. (2018). Recurrent autoregressive networks for online multi-object tracking. In *WACV* (pp. 466–475). IEEE.
- Feichtenhofer, C., Pinz, A., & Zisserman, A. (2017). Detect to track and track to detect. In *Proceedings of the IEEE international conference on computer vision* (pp. 3038–3046).
- Felzenszwalb, P., McAllester, D., & Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *CVPR* (pp. 1–8). IEEE.
- Guo, M., Haque, A., Huang, D. A., Yeung, S., & Fei-Fei, L. (2018). Dynamic task prioritization for multitask learning. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 270–287).
- Han, S., Huang, P., Wang, H., Yu, E., Liu, D., Pan, X., & Zhao, J. (2020). Mat: Motion-aware multi-object tracking. arXiv preprint [arXiv:2009.04794](https://arxiv.org/abs/2009.04794).
- Han, W., Khorrami, P., Paine, T. L., Ramachandran, P., Babaeizadeh, M., Shi, H., Li, J., Yan, S., & Huang, T. S. (2016). Seq-nms for video object detection. arXiv preprint [arXiv:1602.08465](https://arxiv.org/abs/1602.08465).
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *ICCV* (pp. 2961–2969).
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *CVPR* (pp. 770–778).
- Henriques, J. F., Caseiro, R., Martins, P., & Batista, J. (2014). High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3), 583–596.
- Henschel, R., Zou, Y., & Rosenhahn, B. (2019). Multiple people tracking using body and joint detections. In *CVPRW*.
- Hermans, A., Beyer, L., & Leibe, B. (2017). In defense of the triplet loss for person re-identification. arXiv preprint [arXiv:1703.07737](https://arxiv.org/abs/1703.07737).
- Hornakova, A., Henschel, R., Rosenhahn, B., & Swoboda, P. (2020). Lifted disjoint paths with application in multiple object tracking. In *International conference on machine learning, PMLR* (pp. 4364–4375).
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1), 35–45.
- Kang, K., Li, H., Xiao, T., Ouyang, W., Yan, J., Liu, X., & Wang, X. (2017). Object detection in videos with tubelet proposal networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 727–735).
- Kang, K., Ouyang, W., Li, H., & Wang, X. (2016). Object detection from video tubelets with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 817–825).
- Kendall, A., Gal, Y., & Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR* (pp. 7482–7491).
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- Kokkinos, I. (2017). Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR* (pp. 6129–6138).
- Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2), 83–97.

- Law, H., & Deng, J. (2018). Cornernet: Detecting objects as paired keypoints. In *ECCV* (pp. 734–750).
- Leal-Taixé, L., Milan, A., Reid, I., Roth, S., & Schindler, K. (2015). Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv preprint [arXiv:1504.01942](https://arxiv.org/abs/1504.01942).
- Liang, C., Zhang, Z., Lu, Y., Zhou, X., Li, B., Ye, X., & Zou, J. (2020). Rethinking the competition between detection and reid in multi-object tracking. arXiv preprint [arXiv:2010.12138](https://arxiv.org/abs/2010.12138).
- Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017a). Feature pyramid networks for object detection. In *CVPR* (pp. 2117–2125).
- Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017b). Focal loss for dense object detection. In *ICCV* (pp. 2980–2988).
- Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *ECCV* (pp. 740–755). Springer.
- Liu, S., Johns, E., & Davison, A. J. (2019). End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1871–1880).
- Lu, Z., Rathod, V., Votel, R., & Huang, J. (2020). Retinatrack: Online single stage joint detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14668–14678).
- Luo, H., Gu, Y., Liao, X., Lai, S., & Jiang, W. (2019a). Bag of tricks and a strong baseline for deep person re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*.
- Luo, H., Xie, W., Wang, X., & Zeng, W. (2019b). Detect or track: Towards cost-effective video object detection/tracking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 8803–8810.
- Mahmoudi, N., Ahadi, S. M., & Rahmati, M. (2019). Multi-target tracking using cnn-based features: Cnnmtt. *Multimedia Tools and Applications*, 78(6), 7077–7096.
- Milan, A., Leal-Taixé, L., Reid, I., Roth, S., & Schindler, K. (2016). Mot16: A benchmark for multi-object tracking. arXiv preprint [arXiv:1603.00831](https://arxiv.org/abs/1603.00831).
- Milan, A., Roth, S., & Schindler, K. (2013). Continuous energy minimization for multitarget tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1), 58–72.
- Pang, B., Li, Y., Zhang, Y., Li, M., & Lu, C. (2020). 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* (pp. 6308–6318).
- Pang, J., Qiu, L., Li, X., Chen, H., Li, Q., Darrell, T., & Yu, F. (2021). Quasi-dense similarity learning for multiple object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 164–173).
- Peng, J., Wang, C., Wan, F., Wu, Y., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., & Fu, Y. (2020). Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *European conference on computer vision* (pp. 145–161). Springer.
- Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., Dollár, P. (2020). Designing network design spaces. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp 10428–10436
- Ranjan, R., Patel, V. M., & Chellappa, R. (2017). Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *T-PAMI*, 41(1), 121–135.
- Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint [arXiv:1804.02767](https://arxiv.org/abs/1804.02767).
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91–99).
- Ristani, E., Solera, F., Zou, R., Cucchiara, R., & Tomasi, C. (2016). Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV* (pp. 17–35). Springer.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3), 211–252.
- Sadehian, A., Alahi, A., & Savarese, S. (2017). Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *Proceedings of the IEEE international conference on computer vision* (pp. 300–311).
- Sanchez-Matilla, R., Poiesi, F., & Cavallaro, A. (2016). Online multi-target tracking with strong and weak detections. In *ECCV* (pp. 84–99). Springer.
- Sener, O., & Koltun, V. (2018). Multi-task learning as multi-objective optimization. In *NIPS* (pp. 527–538).
- Shan, C., Wei, C., Deng, B., Huang, J., Hua, X. S., Cheng, X., & Liang, K. (2020). Fgagt: Flow-guided adaptive graph tracking. arXiv preprint [arXiv:2010.09015](https://arxiv.org/abs/2010.09015).
- Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., & Sun, J. (2018). Crowdhuman: A benchmark for detecting human in a crowd. arXiv preprint [arXiv:1805.00123](https://arxiv.org/abs/1805.00123).
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556).
- Sun, S., Akhtar, N., Song, H., Mian, A. S., & Shah, M. (2019). Deep affinity network for multiple object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43, 104–119.
- Tang, P., Wang, C., Wang, X., Liu, W., Zeng, W., & Wang, J. (2019). Object detection in videos by high quality object linking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(5), 1272–1278.
- Tang, S., Andriluka, M., Andres, B., & Schiele, B. (2017). Multiple people tracking by lifted multicut and person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3539–3548).
- Valmadre, J., Bewley, A., Huang, J., Sun, C., Sminchisescu, C., & Schmid, C. (2021). Local metrics for multi-object tracking. arXiv preprint [arXiv:2104.02631](https://arxiv.org/abs/2104.02631).
- Vandenhende, S., Georgoulis, S., Van Gansbeke, W., Proesmans, M., Dai, D., & Van Gool, L. (2021). Multi-Task learning for dense prediction tasks: A survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. <https://doi.org/10.1109/TPAMI.2021.3054719>.
- Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B. B. G., Geiger, A., & Leibe, B. (2019). Mots: Multi-object tracking and segmentation. In *CVPR* (pp. 7942–7951).
- Wan, X., Wang, J., Kong, Z., Zhao, Q., & Deng, S. (2018). Multi-object tracking using online metric learning with long short-term memory. In *2018 25th IEEE international conference on image processing (ICIP)* (pp. 788–792). IEEE.
- Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., et al. (2020). Deep high-resolution representation learning for visual recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. <https://doi.org/10.1109/TPAMI.2020.2983686>.
- Wang, Z., Zheng, L., Liu, Y., Li, Y., & Wang, S. (2020b). Towards real-time multi-object tracking. In *Computer vision—ECCV 2020: 16th European conference*, Glasgow, UK, August 23–28, 2020, *Proceedings, Part XI* 16 (pp. 107–122). Springer.
- Wen, L., Li, W., Yan, J., Lei, Z., Yi, D., & Li, S. Z. (2014). Multiple target tracking based on undirected hierarchical relation hypergraph. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1282–1289).
- Wojke, N., Bewley, A., & Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. In *2017 IEEE interna-*

- tional conference on image processing (ICIP) (pp. 3645–3649). IEEE.
- Xiang, Y., Alahi, A., & Savarese, S. (2015). Learning to track: Online multi-object tracking by decision making. In *ICCV* (pp. 4705–4713).
- Xiao, T., Li, S., Wang, B., Lin, L., & Wang, X. (2017). Joint detection and identification feature learning for person search. In *CVPR* (pp. 3415–3424).
- Xu, J., Cao, Y., Zhang, Z., & Hu, H. (2019). Spatial-temporal relation networks for multi-object tracking. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3988–3998).
- Yang, F., Choi, W., & Lin, Y. (2016). Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2129–2137).
- Yang, Z., Liu, S., Hu, H., Wang, L., & Lin, S. (2019). Reppoints: Point set representation for object detection. In *ICCV* (pp. 9657–9666).
- Yu, F., Li, W., Li, Q., Liu, Y., Shi, X., & Yan, J. (2016). Poi: Multiple object tracking with high performance detection and appearance feature. In *ECCV* (pp. 36–42). Springer.
- Yu, F., Wang, D., Shelhamer, E., & Darrel, I. T. (2018). Deep layer aggregation. In *CVPR* (pp. 2403–2412).
- Zamir, A. R., Dehghan, A., & Shah, M. (2012). Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *European conference on computer vision* (pp. 343–356). Springer.
- Zhang, L., Li, Y., & Nevatia, R. (2008). Global data association for multi-object tracking using network flows. In *2008 IEEE conference on computer vision and pattern recognition* (pp. 1–8). IEEE.
- Zhang, S., Benenson, R., & Schiele, B. (2017). Citypersons: A diverse dataset for pedestrian detection. In *CVPR* (pp. 3213–3221).
- Zhang, Y., Sheng, H., Wu, Y., Wang, S., Lyu, W., Ke, W., & Xiong, Z. (2020). Long-term tracking with deep tracklet association. *IEEE Transactions on Image Processing*, 29, 6694–6706.
- Zheng, L., Zhang, H., Sun, S., Chandraker, M., Yang, Y., & Tian, Q. (2017a). Person re-identification in the wild. In *CVPR* (pp. 1367–1376).
- Zheng, Z., Zheng, L., & Yang, Y. (2017b). A discriminatively learned cnn embedding for person reidentification. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 14(1), 1–20.
- Zhou, X., Koltun, V., & Krähenbühl, P. (2020). Tracking objects as points. In *European conference on computer vision* (pp. 474–490). Springer.
- Zhou, X., Wang, D., & Krähenbühl, P. (2019a). Objects as points. arXiv preprint [arXiv:1904.07850](https://arxiv.org/abs/1904.07850).
- Zhou, X., Zhuo, J., & Krahenbuhl, P. (2019b). Bottom-up object detection by grouping extreme and center points. In *CVPR* (pp. 850–859).
- Zhou, Z., Xing, J., Zhang, M., & Hu, W. (2018). Online multi-target tracking with tensor-based high-order graph matching. In *2018 24th International Conference on Pattern Recognition (ICPR)* (pp. 1809–1814). IEEE.
- Zhu, J., Yang, H., Liu, N., Kim, M., Zhang, W., & Yang, M. H. (2018). Online multi-object tracking with dual matching attention networks. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 366–382).

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.