A Simple Baseline for Multi-Object Tracking

Yifu Zhang[1], Chunyu Wang[2], Xinggang Wang[1], Wenjun Zeng[2], and Wenyu Liu[1]

[1] Huazhong University of Science and Technology
{yifuzhang,xgwang,liuwy}@hust.edu.cn
[2] Microsoft Research Asia
{chnuwa,wezeng}@microsoft.com

Abstract. There has been remarkable progress on object detection and re-identification in recent years which are the core components for multi-object tracking. However, little attention has been focused on accomplishing the two tasks in a single network to improve the inference speed. The initial attempts along this path ended up with degraded results mainly because the re-identification branch is not appropriately learned. In this work, we study the essential reasons behind the failure, and accordingly present a simple baseline to addresses the problems. It remarkably outperforms the state-of-the-arts on the public datasets at 30 fps. We hope this baseline could inspire and help evaluate new ideas in this field. The code and the pre-trained models are available at https://github.com/ifzhang/FairMOT.

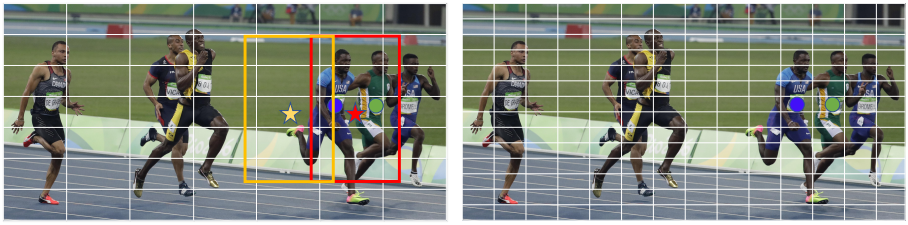
Keywords: One-shot MOT, Simple Baseline, Anchor-free

1 Introduction

Multi-Object Tracking (MOT) has been a longstanding goal in computer vision [3,37,6,40]. The goal is to estimate the trajectories of multiple objects of interest in videos. The successful resolution of the task can benefit many applications such as action recognition, public security, sport videos analysis, elderly care, and human computer interaction.

The state-of-the-art methods [23,46,11,3,37,6,40] usually address the problem by two separate models: the *detection* model first localizes the objects of interest by bounding boxes in the images, and 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 the features. There has been remarkable progress on object detection [27,12,44,26] and Re-ID [43,6] respectively in recent years which in turn boosts the tracking performance. However, those methods cannot perform inference at video rate because the two networks do not share features.

With the maturity of multi-task learning [15], the *one-shot* methods which jointly detect objects and learn Re-ID features have began to attract more attention [35,33]. Since most features are shared for the two models, they have



(a) ***anchor-based*** Re-ID feature extraction (a) ***anchor-free*** Re-ID feature extraction

Fig. 1: (a) The yellow and red anchors are responsible for estimating the same ID (the person in blue shirt) although the image patches are very different. In addition, the anchor-based methods usually operate on a coarse grid. So there is a high chance that the features extracted at the anchor (red or yellow star) are not aligned with the object center. (b) The anchor-free approach suffers less from the ambiguities.

the potential to notably reduce the inference time. However, the accuracy of the one-shot methods usually drops remarkably compared to the two-step ones. In particular, the number of ID switches increases a lot as will be shown in the experimental section. The results show that combining the two tasks is not trivial and should be treated carefully.

Instead of using bags of tricks to improve the tracking accuracy, we study the reasons behind the failure, and present a simple yet effective baseline. Three factors which are critical to the tracking results are identified.

(1) Anchors don't fit Re-ID The current one-shot trackers [35,33] are all based on anchors since they are modified from object detectors [26,12]. However, the anchors are not suitable for learning Re-ID features for two reasons. First, multiple anchors, which correspond to different image patches, may be responsible for estimating the identity of the same object. This causes severe ambiguities for the network. See Figure 1 for illustration. In addition, the feature map is usually down-sampled by 8 times to balance the accuracy and speed. This is acceptable for detection but is too coarse for ReID because the object centers may not align with the features extracted at coarse anchor locations for predicting the object's identity. We solve the problem by treating the MOT problem as a pixel-wise keypoint (object center) estimation and identity classification problem on top of a *high-resolution* feature map.

(2) Multi-Layer Feature Aggregation This is particularly important for MOT because the Re-ID features need to leverage low-level and high-level features to accommodate both small and large objects. We observe in our experiment that this is helpful to reduce identity switches for the one-shot methods due to the improved ability to handle scale variations. Note that the improvement is less significant for the two-step methods because objects will have similar scales after the cropping and resizing operations.

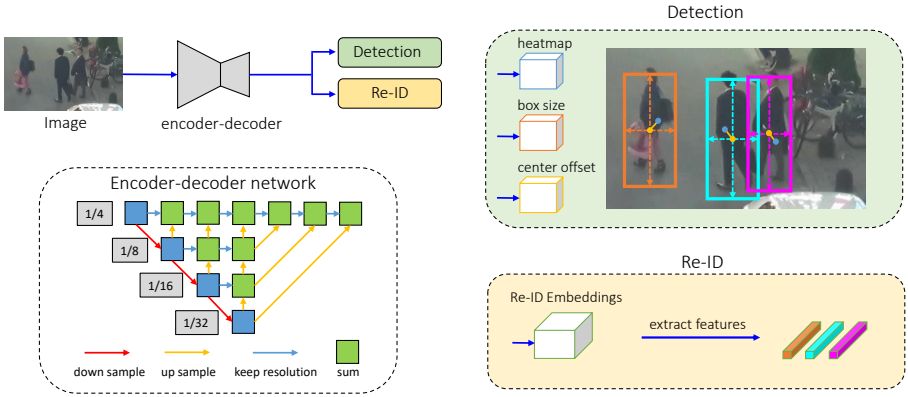


Fig. 2: Overview of our one-shot MOT tracker. The input image is first fed to an encoder-decoder network to extract high resolution feature maps (stride=4). Then we add two simple parallel heads for predicting bounding boxes and Re-ID features, respectively. The features at the predicted object centers are extracted for temporal bounding box linking.

(3) Dimensionality of the ReID Features The previous ReID methods usually learn high dimensional features, and have achieved promising results on their benchmarks. However, we find that lower-dimensional features are actually better for MOT because it has fewer training images than ReID (we cannot use the ReID datasets because they only provide cropped person images). Learning low-dimensional features helps reduce the risk of over-fitting to small data, and improves the tracking robustness.

We present a simple baseline which jointly considers the above three factors. Note that we do not claim algorithmic novelty over the previous works. Instead, our contributions lie in first identifying the challenges behind the one-shot trackers, and then putting together a number of techniques and concepts that are developed in different areas of computer vision to address the challenges which are *overlooked* in the previous MOT works.

The overview of our approach is shown in Figure 2. We first adopt an *anchor-free* object detection approach to estimate the object centers [44,17,45,9] on a high-resolution feature map. The elimination of anchors alleviates the ambiguity problem and the use of a high-resolution feature map enables the Re-ID features to be better aligned with the object centers. Then we add a parallel branch for estimating the pixel-wise Re-ID features which are used to predict the objects' identities. In particular, we learn low-dimensional Re-ID features which not only reduce the computation time but also improve the robustness of feature matching. We equip the backbone network [13] with the Deep Layer Aggregation operator [41] to fuse features from multiple layers in order to deal with objects of different scales.

We evaluate our approach on the MOT Challenge benchmark via the evaluation server. It ranks first among all online trackers on the 2DMOT15 [18], MOT16 [24], MOT17 [24] and MOT20 [7] datasets. In fact, it also outperforms the offline trackers on the 2DMOT15, MOT17 and MOT20 datasets (MOT20 is the newest dataset and no previous works have reported results on it). In spite of the strong results, the approach is very simple and runs at 30 FPS. We hope it could be used as a strong baseline in this field. The code as well as the pre-trained models will be released.

2 Related Work

In this section, we briefly review the related works on MOT by classifying them into the two-step and one-shot methods, respectively. We discuss the pros and cons of the methods and compare them to our approach.

2.1 Two-Step MOT Methods

The state-of-the-art MOT methods such as [37,40,23,46,11] often treat object detection and Re-ID as two separate tasks. They first apply the CNN detectors such as [27,12,26] to localize all objects of interest in the images by a number of boxes. Then in a separate step, they crop the images according to the boxes and feed them to the identity embedding network to extract Re-ID features, and link the boxes to form multiple tracks. The works usually follow a standard practice for box linking 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 [36] and Hungarian algorithm [16] to accomplish the linking task. A small number of works such as [23,46,11] use more complicated association strategies such as group models and RNNs.

The advantage of the two-step methods is that they can use the most suitable model for each task, respectively, without making compromises. In addition, they can crop the image patches according to the detected bounding boxes and resize them to the same size before predicting Re-ID features. This helps to handle the scale variations of objects. As a result, these approaches [40] have achieved the best performance on the public datasets. However, they are usually very slow because both object detection and Re-ID feature embedding need a lot of computations without sharing between them. So it is hard to achieve video rate inference which is required in many applications.

2.2 One-Shot MOT Methods

With the maturity of multi-task learning [15,25,30] 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 the inference time through sharing most of the computation. For example, Track-RCNN [33] adds a Re-ID head on top

of Mask-RCNN [12] and regresses a bounding box and a Re-ID feature for each proposal. The JDE [35] is introduced on top of the YOLOv3 [26] framework which achieves near video rate inference.

However, the tracking accuracy of the one-shot methods is usually lower than that of the two-step methods. We find this is because the learned Re-ID features are not optimal which leads to a large number of ID switches. We deeply investigate the reasons and find that the identity embedding features extracted at anchors are not aligned with the object centers which causes severe ambiguities. To address the problem, we propose to use *anchor-free* approaches for both object detection and identity embedding which significantly improves the tracking accuracy on all benchmarks.

3 The Technical Approach

In this section, we present the details for the backbone network, the object detection branch and the Re-ID feature embedding branch, respectively.

3.1 Backbone Network

We adopt the ResNet-34 [13] as our backbone in order to strike a good balance between the accuracy and speed. To accommodate objects of different scales, a variant of Deep Layer Aggregation (DLA) [44] is applied to the backbone as shown in Figure 2. Different from the original DLA [41], it has more skip connections between the low-level and high-level features which is similar to the Feature Pyramid Network (FPN) [19]. In addition, all convolution layers in the up-sampling module are replaced by the deformable convolution layers such that they can dynamically adapt the receptive field according to the 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 the 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 $W = W_{\text{image}}/4$.

3.2 Object Detection Branch

Following [44], we treat object detection as a center-based based bounding box regression task on a high-resolution feature map. In particular, three parallel regression heads are appended to the backbone network 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 feature maps of the backbone network, followed by a 1×1 convolutional layer which generates the final targets.

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 location in the heatmap and the object center.

Center Offset Head This head is responsible for localizing the objects more precisely. Recall that the stride of the feature map is four which will introduce non-negligible quantization errors. Note that the benefits for object detection performance may be marginal. But it is critical for tracking because the Re-ID features should be extracted according to the accurate object centers. We find in our experiments that the careful alignment of the ReID features with object centers is critical for the performance.

Box Size Head This head is responsible for estimating the height and width of the target bounding box at each anchor location. This head is not directly related to the Re-ID features but the localization precision will impact the evaluation of the object detection performance.

3.3 Identity Embedding Branch

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

3.4 Loss Functions

Heatmap Loss 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 σ_c represents the standard deviation. The loss function is defined as pixel-wise logistic regression with focal loss [20]:

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

where $\hat{M}$ is the estimated heatmap, and α, β are the parameters.

Offset and Size Loss We denote the outputs 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 can compute its size as $\mathbf{s}^i = (x_2^i - x_1^i, y_2^i - y_1^i)$. Similarly, the GT offset can be 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:

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

Identity Embedding Loss We treat object identity embedding as a classification task. In particular, all object instances of the same identity in the training set are treated as one 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 an identity feature vector $\mathbf{E}_{x^i, y^i}$ at the location and learn to map it to a class distribution vector $\mathbf{p}(k)$. Denote the one-hot representation of the GT class label as $\mathbf{L}^i(k)$. Then we compute the softmax 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.

3.5 Online Tracking

In this section, we explain the inference of our model and how we perform box tracking with the detection results and identity embeddings.

Network Inference The network takes an image of size 1088×608 as input which is the same as the previous work JDE [35]. 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.

Online Box Linking We use the standard online tracking algorithm to achieve box linking. We initialize a number of tracklets based on the estimated boxes in the first frame. In the subsequent frames, we link the boxes to the existing tracklets according to their distances measured by Re-ID features and IoU's. We also use Kalman Filter 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 [4,14].

4 Experiments

4.1 Datasets and Metrics

Following the previous works such as [35], we compose a large training dataset by combining the training images from six public datasets for human detection and search. In particular, the ETH [10] and the CityPerson [42] datasets only provide bounding box annotations so we only train the detection branch on them. The CalTech [8], MOT17 [24], CUHK-SYSU [39] and PRW [43] datasets provide both bounding box and identity annotations on which we train both of the detection and identity embedding branches. Since some videos in the ETH dataset also appear in the testing set of the MOT16 dataset, we remove them from the training dataset for fair comparison. In some ablative experiments, we propose to train our model on a smaller dataset to save the computation cost which will be described clearly.

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. As in [35], We use Average Precision (AP) for evaluating the detection performance, and True Positive Rate (TPR) at a false accept rate of 0.1 for evaluating the Re-ID features. We use the CLEAR metric [2] and IDF1 [28] to evaluate the tracking accuracy.

4.2 Implementation Details

We use a variant of DLA-34 proposed in [44] as our default backbone. The model parameters pre-trained on the COCO detection dataset [21] are used to initialize our model. We train our model with the Adam optimizer for 30 epochs with a starting learning rate of $1e-4$. The learning rate decays to $1e-5$ and $1e-6$, at 20 and 27 epochs, respectively. 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 takes about 30 hours on two RTX 2080 GPUs.

4.3 Ablative Study

Anchor-based vs. Anchor-free The previous one-shot trackers are based on anchors which suffer from the mis-alignment problem as described in the previous sections. In this section, we numerically validate the argument by constructing an anchor-based baseline on top of our approach by replacing the detection branch with the anchor-based method used in [35]. We keep the rest of the factors the same for the two approaches for fair comparison. Note that the models in this section are trained on the large training dataset because the anchor-based method obtains very bad results when we use small datasets for training. The results are shown in Table 1.

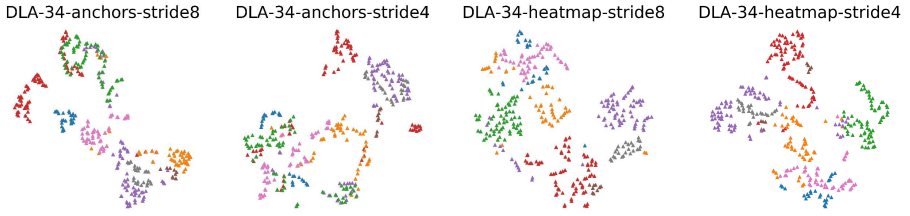


Fig. 3: We plot the Re-ID features of all persons in the testing set learned by four models using t-SNE [22]. The features of the same person are represented by the same color.

Table 1: Evaluation of the anchor-based and anchor-free methods on the validation videos of the MOT15 dataset. The large training dataset is used and all models are trained for 10 epochs. ↑ means the larger the better and ↓ means the smaller the better. The best results are in **bold**.

Backbone	stride	Head	MOTA↑	IDF1↑	IDs↓	Prec↑	Rec↑	AP↑	TPR↑
DLA-34	2	anchor-free	71.9	70.3	93	91.7	79.8	87.2	56.5
DLA-34	4	anchor-based	64.9	62.1	137	87.9	76.4	81.9	73.6
DLA-34	4	anchor-free	**75.9**	**72.3**	**93**	94.2	**81.6**	**88.2**	80.8
DLA-34	8	anchor-based	65.5	66.3	139	91.8	73.1	83.4	75.3
DLA-34	8	anchor-free	67.3	64.9	109	**94.8**	72.2	85.1	**85.5**

We can see that the anchor-based method obtains consistently lower *MOTA* scores than our anchor-free method for different strides. For example, when the stride is 8, the anchor-free method achieves a significantly better *TPR* score than the anchor-based baseline (85.5% vs. 75.3%) meaning that the Re-ID features of the anchor-free method have clear advantages. The main reason is that the mis-alignment between the anchors and object centers causes severe ambiguities to the learning of the network.

It is noteworthy that increasing the feature map resolution for the anchor-based method even degrades the *MOTA* score. This is because there will be more *unaligned* positive anchors when we use high-resolution feature maps which makes the network training even more difficult. We do not show the results for the stride of 2 because the significantly increased number of anchors exceed the memory capacity of our GPUs.

In contrast, our anchor-free approach suffers less from the mis-alignment issue and achieves notably better *MOTA* score than the anchor-based one. In particular, the number of ID switches decreases significantly from 137 to 93 for the stride of four. More importantly, our approach benefits a lot when we decrease the stride from 8 to 4. Further decreasing the stride to 2 begins to

Table 2: Evaluation of different backbones on the 2DMOT15 dataset. The best results are shown in **bold**.

Backbone	MOTA↑	IDF1↑	IDs↓	Prec↑	Rec↑	FPS↑	AP↑	TPR↑
ResNet-34	30.7	41.3	372	74.6	48.8	**47.3**	61.9	35.0
ResNet-34-FPN	34.0	45.2	320	77.1	50.3	36.1	67.3	40.9
ResNet-50	34.6	42.8	432	81.9	46.7	32.0	62.8	35.4
HRNet-W32	37.9	52.8	189	83.9	47.8	22.2	65.7	63.8
DLA-34	**40.4**	**53.9**	**136**	**83.9**	**50.7**	31.0	**68.3**	**67.3**

Table 3: The impact of backbones for objects of different scales. *Small*: area smaller than 6000; *Medium*: area from 6000 to 25000; *Large*: area larger than 25000.

Backbone	AP^S	AP^M	AP^L	TPR^S	TPR^M	TPR^L	IDs^S	IDs^M	IDs^L
ResNet-34	32.6	60.2	72.6	28.8	32.2	22.5	48	131	149
ResNet-34-FPN	**39.3**	**63.9**	75.1	38.3	41.5	34.2	49	121	104
ResNet-50	33.0	59.8	71.1	29.7	43.7	30.3	37	162	172
HRNet-W32	35.6	60.2	**78.7**	60.1	67.9	**59.7**	**23**	49	97
DLA-34	36.2	62.9	78.3	**61.9**	**71.2**	55.2	25	**47**	**41**

degrade the results because the introduction of lower-level features makes the representation less robust to appearance variations.

We also visualize the Re-ID features learned by different models in Figure 3. We can see that the features of different identities are mixed for the anchor-based approach, especially when the stride is 4. In contrast, they are well separated for our anchor-free approach.

Multi-Layer Feature Aggregation This section evaluates the impact of multi-layer feature aggregation in the backbone networks. In particular, we evaluate a number of backbones such as the vanilla ResNet [13], Feature Pyramid Network (FPN) [19], High-Resolution Network (HRNet) [31] and DLA-34 [44]. The remaining factors of the approaches are controlled to be the same for fair comparison. The stride of the final feature map is 4 for all methods in this experiment. In particular, We add three up-sampling operations for the vanilla ResNet to obtain the stride-4 feature map. We split the training subset of the 2DMOT15 dataset into 5 training videos and 6 validation videos following the practice of the previous work [38]. The large scale training dataset is not used here in order to reduce the computation cost.

The results are shown in Table 2. We can see that DLA-34, which is built on top of the ResNet-34, achieves a notably better *MOTA* score than the vanilla

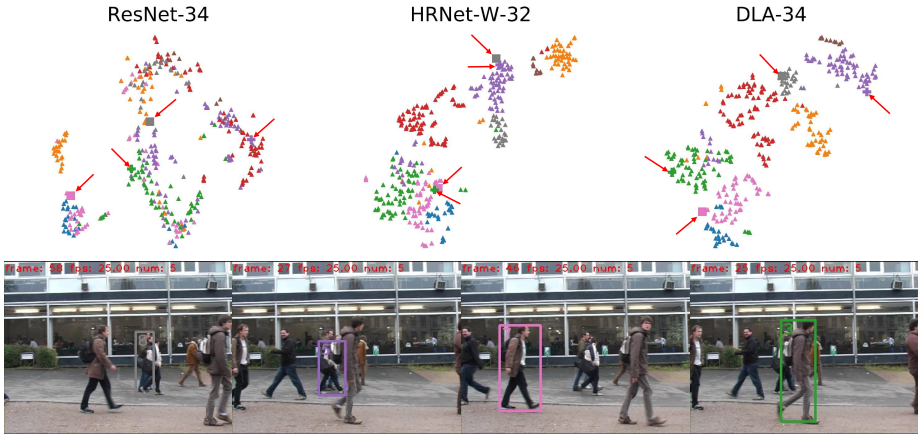


Fig. 4: We plot the Re-ID features of all persons learned by three backbone networks, respectively, using t-SNE [22]. The features of the same person are represented by the same color. The features learned by DLA-34 has clear advantages in terms of discriminative ability. We highlight the features of four different people by red arrows. The appearance of the four people correspond to the boxes of different colors as shown in the bottom images.

ResNet-34. In particular, *TPR* increases significantly from 35.0% to 67.3% which in turn decreases the number of ID switches (*IDs*) from 372 to 136. The experimental results suggest that the discriminative ability of the Re-ID features improves due to the multi-layer feature fusion.

By comparing the results of ResNet-34 and ResNet-50, we can see that using a larger network also improves the overall *MOTA* score. However, if we look into the detailed metrics, we find that the improvement is mainly from the enhanced detection results measured by *AP*. However, the Re-ID features barely benefit from the larger network. For example, *TPR* only improves from 35.0% to 35.4%. In contrast, the number is 67.3% for DLA-34. The results demonstrate that multi-layer fusion has clear advantages over using deeper networks in terms of improving the identity embeddings.

We also compare to other multi-layer fusion methods such as HRNet [31] and FPN [19]. Both approaches achieve better *MOTA* scores than ResNet-34. The improvement not only comes from the enhanced detection results, but also from the improved discriminative ability of the Re-ID features. For example, *TPR* increases from 35.0% to 63.8% for HRNet.

The DLA-34 model has additional gains over FPN and HRNet. We find that the deformable convolution in DLA-34 is the main reasons for the gap because it can alleviate the mis-alignment issue caused by down-sampling for small objects. As shown in Table 3, we can see that DLA-34 mainly outperforms HRNet on small and middle sized objects.

We visualize the Re-ID features of all persons in the testing set in Figure 4 by t-SNE [22]. We can see that the features learned by the vanilla ResNet-34 are not discriminative since the features of different identities are mostly mixed together. This will cause a large number of ID switches in the linking stage. The Re-ID features learned by HRNet become better except that the pink and green points are largely confused. In addition, the Re-ID features of DLA-34 are more discriminative than the two baseline methods.

The Re-ID Feature Dimension The previous works usually learn 512 dimensional features without ablation study. However, we find in our experiments that the feature dimension actually plays an important role. In general, to avoid over-fitting, training high-dimensional Re-ID features requires a large number of training images which is not available for the one-shot tracking problem. The previous two-step approaches suffer less from the problem because they could leverage the abundant Re-ID datasets which provide *cropped* person images. The one-shot methods including ours cannot use them because it requires original uncropped images. One solution is to reduce its dependence on data by reducing the dimensionality of Re-ID features.

We evaluate multiple choices of dimensionality in Table 4. We can see that *TPR* consistently improves when the dimension decreases from 512 to 128 which demonstrates the advantages of using low-dimensional features. Further reducing the dimensionality to 64 begins to decrease *TPR* because the representative ability of the Re-ID features suffers. Although the changes for *MOTA* score are very marginal, the number of ID switches actually decreases significantly from 210 to 136. This actually plays a critical role in improving the user experience. The inference speed is also slightly improved by reducing the dimensionality of the Re-ID features. It is noteworthy that the argument of using lower-dimensional Re-ID features only holds when we have access to a small number of training data. The gap caused by the feature dimensionality will become smaller when the number of training data increases.

Table 4: Evaluation of the Re-ID feature dimensions on the 2DMOT15 dataset.

Backbone	dim	MOTA ↑	IDF1 ↑	IDs ↓	FPS↑	TPR ↑
DLA-34	512	40.4	52.1	210	28.7	61.5
DLA-34	256	40.2	**55.1**	157	30.6	63.5
DLA-34	128	**40.4**	53.9	**136**	31.0	**67.3**
DLA-34	64	40.4	51.1	165	**31.7**	61.0

4.4 The State-of-the-arts

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

One-Shot MOT Methods There are only two published works, *i.e.* JDE [35] and TrackRCNN [33], that jointly perform object detection and identity feature embedding. In particular, TrackRCNN requires additional segmentation annotations and reports results using a different metric for the segmentation task. So in this work, we only compare to JDE.

Table 5: Comparison to the state-of-the-art one-shot trackers on two datasets. The results on MOT16-*test* are obtained from the MOT challenge server.

Dataset	Method	MOTA↑	IDF1↑	IDs↓	FPS↑
MOT15 *train*	JDE [35]	67.5	66.7	218	22.5
	FairMOT(ours)	**77.1**	**76.0**	**80**	**30.9**
MOT16 *test*	JDE [35]	64.4	55.8	1544	18.5
	FairMOT(ours)	**68.7**	**70.4**	**953**	**25.9**

For fair comparison, we use the same data for training and testing as in [35]. Specifically, we use 2DMOT15-*train* and MOT16-*test* for validation. The CLEAR metric [2] and IDF1 [28] are used to measure the performance. The results are shown in Table 5. We can see that our approach remarkably outperforms JDE [35] on both datasets. In particular, the number of ID switches reduces from 218 to 80 which is big improvement in terms of improving the 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.

Two-Step MOT Methods We compare our approach to the state-of-the-art *online* trackers including the two-step methods on the MOT Challenge dataset in Table 6 [3]. 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. We finetune our model for 10 epochs on each of the dataset before doing testing. All of the results are obtained on the MOT challenge evaluation server.

Our approach ranks first among all *online* trackers on the four datasets. In fact, it also achieves the highest *MOTA* score among all *online* and *offline* trackers on the 2DMOT15 and MOT17 datasets, respectively. This is a very strong

[3] online tracker means it only uses the information before current frame for tracking; offline tracker could use the whole video.

Table 6: Comparison to the state-of-the-arts under the "private detector" protocol. It is noteworthy that the computation time (Hz) only counts for the association step for the two-step trackers. But for the one-shot trackers, it counts for the whole system. The one-shot trackers are labeled by "*".

Dataset	Tracker	MOTA↑	IDF1↑	MT↑	ML↓	IDs↓	Hz↑
MOT15	MDP_SubCNN[38]	47.5	55.7	30.0%	18.6%	628	2.1
	CDA_DDAL[1]	51.3	54.1	36.3%	22.2%	544	1.3
	EAMTT[29]	53.0	54.0	35.9%	19.6%	7538	11.5
	AP_HWDPL[5]	53.0	52.2	29.1%	20.2%	708	6.7
	RAR15[11]	56.5	61.3	45.1%	14.6%	**428**	5.1
	Ours*	**59.0**	**62.2**	**45.6%**	**11.5%**	582	**30.5**
MOT16	EAMTT[29]	52.5	53.3	19.9%	34.9%	910	12.2
	SORTwHPD16[3]	59.8	53.8	25.4%	22.7%	1423	**59.5**
	DeepSORT_2[37]	61.4	62.2	32.8%	**18.2%**	781	17.4
	RAR16wVGG[11]	63.0	63.8	**39.9%**	22.1%	**482**	1.6
	VMaxx[34]	62.6	49.2	32.7%	21.1%	1389	6.5
	JDE*[35]	64.4	55.8	35.4%	20.0%	1544	18.5
	TAP[46]	64.8	**73.5**	38.5%	21.6%	571	39.4
	CNNMTT[23]	65.2	62.2	32.4%	21.3%	946	11.2
	POI[40]	66.1	65.1	34.0%	20.8%	805	9.9
	Ours*	**68.7**	70.4	39.5%	19.0%	953	25.9
MOT17	SST[32]	52.4	49.5	21.4%	30.7%	8431	6.3
	Ours*	**67.5**	**69.8**	**37.7%**	**20.8%**	**2868**	**25.9**
MOT20	Ours*	**58.7**	**63.7**	**66.3%**	**8.5%**	**6013**	**13.2**

result considering that our approach is very simple. In addition, our approach achieves video rate inference. In contrast, most high-performance trackers such as [11,40] are usually slower than ours.

5 Conclusion

We present a simple baseline for one-shot multiple object tracking. We start by studying why the previous methods such as [35] fails 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. We present a simple anchor-free approach which outperforms the previous state-of-the-arts on several benchmark datasets with 30 fps. We hope it could inspire and evaluate new ideas in this field.

References

1. Bae, S.H., Yoon, K.J.: 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 (2017)
2. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. EURASIP Journal on Image and Video Processing **2008**, 1–10 (2008)
3. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing (ICIP). pp. 3464–3468. IEEE (2016)
4. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 2544–2550. IEEE (2010)
5. Chen, L., Ai, H., Shang, C., Zhuang, Z., Bai, B.: Online multi-object tracking with convolutional neural networks. In: 2017 IEEE International Conference on Image Processing (ICIP). pp. 645–649. IEEE (2017)
6. Chen, L., Ai, H., Zhuang, Z., Shang, C.: Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: 2018 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2018)
7. Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., Leal-Taixe, L.: Cvpr19 tracking and detection challenge: How crowded can it get? arXiv preprint arXiv:1906.04567 (2019)
8. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 304–311. IEEE (2009)
9. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: ICCV. pp. 6569–6578 (2019)
10. Ess, A., Leibe, B., Schindler, K., Van Gool, L.: A mobile vision system for robust multi-person tracking. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2008)
11. Fang, K., Xiang, Y., Li, X., Savarese, S.: Recurrent autoregressive networks for online multi-object tracking. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 466–475. IEEE (2018)
12. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
14. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE transactions on pattern analysis and machine intelligence **37**(3), 583–596 (2014)
15. Kokkinos, I.: 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 (2017)
16. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)
17. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: ECCV. pp. 734–750 (2018)

18. Leal-Taixé, L., Milan, A., Reid, I., Roth, S., Schindler, K.: Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv preprint arXiv:1504.01942 (2015)
19. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
20. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
21. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
22. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
23. Mahmoudi, N., Ahadi, S.M., Rahmati, M.: Multi-target tracking using cnn-based features: Cnnmtt. Multimedia Tools and Applications **78**(6), 7077–7096 (2019)
24. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: Mot16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016)
25. Ranjan, R., Patel, V.M., Chellappa, R.: Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. T-PAMI **41**(1), 121–135 (2017)
26. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
27. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
28. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: European Conference on Computer Vision. pp. 17–35. Springer (2016)
29. Sanchez-Matilla, R., Poiesi, F., Cavallaro, A.: Online multi-target tracking with strong and weak detections. In: European Conference on Computer Vision. pp. 84–99. Springer (2016)
30. Sener, O., Koltun, V.: Multi-task learning as multi-objective optimization. In: NIPS. pp. 527–538 (2018)
31. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: CVPR. pp. 5693–5703 (2019)
32. Sun, S., Akhtar, N., Song, H., Mian, A.S., Shah, M.: Deep affinity network for multiple object tracking. IEEE transactions on pattern analysis and machine intelligence (2019)
33. Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B.B.G., Geiger, A., Leibe, B.: Mots: Multi-object tracking and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7942–7951 (2019)
34. Wan, X., Wang, J., Kong, Z., Zhao, Q., Deng, S.: 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 (2018)
35. Wang, Z., Zheng, L., Liu, Y., Wang, S.: Towards real-time multi-object tracking. arXiv preprint arXiv:1909.12605 (2019)
36. Welch, G., Bishop, G., et al.: An introduction to the kalman filter (1995)
37. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE international conference on image processing (ICIP). pp. 3645–3649. IEEE (2017)

38. Xiang, Y., Alahi, A., Savarese, S.: Learning to track: Online multi-object tracking by decision making. In: Proceedings of the IEEE international conference on computer vision. pp. 4705–4713 (2015)
39. Xiao, T., Li, S., Wang, B., Lin, L., Wang, X.: Joint detection and identification feature learning for person search. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3415–3424 (2017)
40. Yu, F., Li, W., Li, Q., Liu, Y., Shi, X., Yan, J.: Poi: Multiple object tracking with high performance detection and appearance feature. In: European Conference on Computer Vision. pp. 36–42. Springer (2016)
41. Yu, F., Wang, D., Shelhamer, E., Darrell, T.: Deep layer aggregation. In: CVPR. pp. 2403–2412 (2018)
42. Zhang, S., Benenson, R., Schiele, B.: Citypersons: A diverse dataset for pedestrian detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3213–3221 (2017)
43. Zheng, L., Zhang, H., Sun, S., Chandraker, M., Yang, Y., Tian, Q.: Person re-identification in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1367–1376 (2017)
44. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)
45. Zhou, X., Zhuo, J., Krahenbuhl, P.: Bottom-up object detection by grouping extreme and center points. In: CVPR. pp. 850–859 (2019)
46. Zhou, Z., Xing, J., Zhang, M., Hu, W.: Online multi-target tracking with tensor-based high-order graph matching. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 1809–1814. IEEE (2018)