Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Associative affinity network learning for multi-object tracking*

Liang MA‡, Qiaoyong ZHONG, Yingying ZHANG, Di XIE, Shiliang PU

Hikvision Research Institute, Hangzhou 310000, China

E-mail: {maliang6, zhongqiaoyong, zhangyingying7, xiedi, pushiliang.hri}@hikvision.com

Received June 4, 2020; Revision accepted Oct. 8, 2020; Crosschecked July 22, 2021

Abstract: We propose a joint feature and metric learning deep neural network architecture, called the associative affinity network (AAN), as an affinity model for multi-object tracking (MOT) in videos. The AAN learns the associative affinity between tracks and detections across frames in an end-to-end manner. Considering flawed detections, the AAN jointly learns bounding box regression, classification, and affinity regression via the proposed multi-task loss. Contrary to networks that are trained with ranking loss, we directly train a binary classifier to learn the associative affinity of each track-detection pair and use a matching cardinality loss to capture information among candidate pairs. The AAN learns a discriminative affinity model for data association to tackle MOT, and can also perform single-object tracking. Based on the AAN, we propose a simple multi-object tracker that achieves competitive performance on the public MOT16 and MOT17 test datasets.

Key words: Multi-object tracking; Deep neural network; Affinity learning
https://doi.org/10.1631/FITEE.2000272 **CLC number:** TP391

1 Introduction

Multi-object tracking (MOT) estimates the locations of multiple objects and maintains their identities across frames. It is a challenging task and has a wide range of applications such as visual surveillance, human action recognition, and autonomous driving. Generally speaking, MOT can be categorized into two main frameworks, namely tracking by detection (Tang et al., 2017) and joint tracking and detection (Feichtenhofer et al., 2017). In either framework, data association is the main challenge of MOT. A typical data association approach first builds a graph with each node being a track or detection and each edge representing pairwise associative affinity, and then solves the graph using different optimization methods (Yang B and Nevatia, 2014; Lan et al., 2016; Milan et al., 2017b; Schulter et al., 2017).

The tracking by detection framework uses the power of recent advancements in object detectors based on deep neural networks (Feichtenhofer et al., 2017; Bergmann et al., 2019a; Zhou et al., 2020). However, if a detector is given a priori, the key to the success of a multi-object tracker is the associative affinity model that estimates the associative affinity of tracks and detections. The associative affinity model for data association consists of two components, i.e., feature representation and affinity metric (Sun et al., 2021), that is, what kind of feature is used to represent each target's appearance and motion cue, and how likely are these features to be associated with each other.

During the past few years, researchers in the MOT community focused mainly on feature representation learning, while metrics such as the Euclidean and cosine distances have been used to

‡ Corresponding author

* Project supported by the National Key Research and Development Program of China (No. 2020AAA0109004) and the Zhejiang Postdoc Sponsorship

ORCID: Liang MA, https://orcid.org/0000-0003-4228-258X

measure feature similarity. The appearance feature has drawn much attention since the rise of visual tracking applications. Hand-crafted features including color histograms (Nummiaro et al., 2003), histogram of oriented gradient (HOG) (Dalal and Triggs, 2005), and local binary pattern (LBP) (Wang XY et al., 2009) have been widely adopted. Recently, convolutional neural networks (CNNs) have been employed as feature descriptors (Kim et al., 2015; Fang et al., 2018). Meanwhile, the motion feature is evolving from the constant velocity model (Xiang Y et al., 2015; Wojke et al., 2017) to a long short-term memory (LSTM) network (Sadeghian et al., 2017).

Researchers have also attempted to find a robust appearance model for data association. A typical way is to borrow ideas from the person re-identification (ReID) task. Contrastive loss, triplet loss, and their variants have been exploited to learn representative features (Hermans et al., 2017; Ristani and Tomasi, 2018; Yin et al., 2020). However, for objects under heavy occlusion, features learned in this way may not be reliable enough. Distinguishing objects under various degrees of occlusion is a critical issue for MOT. Some pioneering works attempted to learn an affinity metric for MOT (Wang B et al., 2016; Chu P and Ling, 2019; Sun et al., 2021), which has been verified by metric learning for other tasks (Chen S et al., 2018, 2019; Duan et al., 2020).

Inspired by previous works, this work aims to learn feature representation and affinity metrics jointly in an end-to-end framework. In particular, we propose a novel deep neural network architecture named the associative affinity network (AAN), which is able to directly estimate the associative affinity of a track-detection pair as illustrated in Fig. 1. To learn the associative affinity, we propose affinity loss, which is essentially a binary cross-entropy loss applied to track-detection pairs. In addition, we introduce matching cardinality loss to enforce mutual exclusion among different pairs. Notably, AAN can also be used as a drop-in replacement for an affinity model in other data association frameworks (Kim et al., 2015; Rezatofighi et al., 2015; Schulter et al., 2017).

To validate the benefits of AAN to MOT, we develop a simple multi-object tracker where data association is modeled as bipartite graph matching based on the association affinity matrix predicted by AAN. Moreover, we show that AAN reveals its ability to act as a single-object tracker to discover high-quality detections in the presence of missed detections. Extensive experiments show that with the simple tracker, AAN achieves competitive tracking performance on the MOT16 and MOT17 datasets.

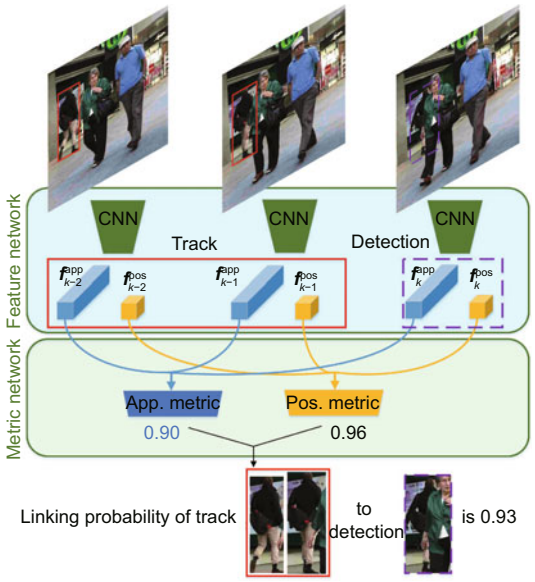

Fig. 1 Pipeline of the associative affinity network that computes the affinity of a track-detection pair via the jointly learned feature and metric networks

Our main contributions are summarized as follows:

1. We propose a deep affinity model for MOT that learns feature representation and affinity metrics jointly in an end-to-end manner.

2. The proposed AAN is capable of single-object tracking (SOT), which helps maintain trajectory consistency in the presence of missed detections.

3. With a simple multi-object tracker, AAN achieves competitive results on the challenging MOT16 and MOT17 datasets.

2 Related work

In the MOT community, researchers have tried to integrate different features to improve the tracking performance (Xiang J et al., 2016; Sadeghian et al., 2017; Son et al., 2017). Most works intuitively focused on appearance feature extraction motivated by the person ReID task, which has been shown to be effective by learning from a person ReID dataset to reduce the number of identity switches (Wojke et al.,

2017; Ristani and Tomasi, 2018; Bergmann et al., 2019a). With regard to motion feature extraction, a non-Markovian approach was proposed to impose global consistency by using behavioral patterns for tracking (Maksai et al., 2017). The aforementioned works tried to learn features to distinguish targets, but these features may not be as effective for some cases, e.g., targets under heavy occlusion and closely spaced targets with similar appearance. These difficult cases would cause mismatches and affect the trajectory consistency.

Rather than learning appearance features independently, some works aimed to learn feature representation and metrics jointly to directly distinguish positive and negative pairs (Leal-Taixé et al., 2016; Wang B et al., 2016; Ma et al., 2018). Recently, Sun et al. (2021) proposed the deep affinity network (DAN) to jointly model appearance and affinity in an end-to-end fashion. DAN uses a VGG-like network as an appearance feature extractor and applies an affinity estimator to obtain pairwise affinity. Moreover, Chu P and Ling (2019) proposed FAMNet, in which a multi-dimensional assignment sub-network was built to estimate the set of optimal assignments given the hypothesis trajectory affinity. Despite the multi-dimensional assignment sub-network, they adopted graph multicut to solve the affinity matrix for association. Yin et al. (2020) unified the Siamese network based SOT and object identity classification in a triplet network named UMA, to learn discriminative affinity with multi-task learning. However, the Siamese network based SOT is time-consuming under dense trajectories.

To further exploit the end-to-end learning potential of deep CNNs in this direction, we propose the AAN to learn both appearance and position features as well as their affinity metrics in a simple unified framework. Compared to the aforementioned end-to-end framework (Chu P and Ling, 2019; Yin et al., 2020; Sun et al., 2021), our AAN is composed of a feature network that is built on a detector CNN by adding a new branch for appearance extraction, and a much simpler metric network that has been shown to be effective for affinity estimation, as illustrated in Fig. 1. Our proposed AAN further narrows the gap between object detection and tracking tasks.

In addition, a few works have attempted to learn the affinity model and solve the affinity graph at the same time (Milan et al., 2017b; Schulter et al., 2017). Although this direction is promising, there is still a performance gap when compared to conventional multi-stage methods. Recent works have also revealed the difficulty in sufficiently tackling combinatorial optimization problems using neural networks (Milan et al., 2017a; Emami and Ranka, 2018). The proposed AAN sheds light on end-to-end data association with competitive performance, and will be further explored for joint affinity modeling and graph optimization for a more complete framework.

3 Associative affinity network

3.1 Overview

As illustrated in Fig. 1, AAN is a deep CNN architecture composed of two sub-networks, i.e., a feature network and a metric network. The feature network maps a detection into its feature representation, while the metric network maps a pair of features to their affinity. In this section, we describe the two sub-networks in detail, followed by the AAN training strategy.

3.2 Feature network

In the feature network, we design two branches to extract the appearance feature and motion feature of each region of interest (RoI), respectively.

Regarding detections as RoIs in an image, we adopt an architecture similar to mask R-CNN (He et al., 2017). As illustrated in Fig. 2, we add a new branch to extract each RoI's appearance feature. By appending a new fully connected (FC) layer, we obtain appearance feature $\boldsymbol{f}^{\mathrm{app}}$ of dimension D. Meanwhile, the existing classification and bounding box regression branches are preserved, which helps in refining the input RoIs. With the refinement, we can obtain more stable and accurate confidence score as well as the position of each RoI. It is worth noting that all three branches share the same input features, which makes the network very efficient. With multi-task learning, the appearance feature branch also benefits from the box refinement branches.

Considering the fact that the target motion pattern in the image plane is too difficult to learn (Tang et al., 2017), we use optical flow to guide the spatial-temporal alignment of each RoI's position. This approach is similar to the position feature used in previous work (Bullinger et al., 2017; Son et al., 2017;

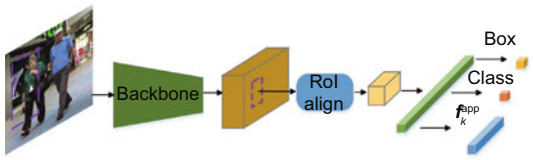

Fig. 2 Appearance feature network

Zhang et al., 2020). The difference is that rather than mask- and pixel-level flows, we use RoI-level flow to align RoIs from the current frame to the reference frame.

Denote $\boldsymbol{v}^{\mathcal{R}}$ as the RoI-level flow (including x and y directions) for region $\mathcal{R}$. $\boldsymbol{F}_{k^*\to k}$ is the optical flow from frame k^* to frame k computed using FlowNet2 (Ilg et al., 2017). Then, we compute the RoI-level flow by

$$\boldsymbol{v}^{\mathcal{R}}_{k^*\to k}=\frac{1}{|\mathcal{R}|}\Sigma_{(x,y)\in\mathcal{R}}\boldsymbol{F}_{k^*\to k}(x,y), \tag{1}$$

where $|\mathcal{R}|$ is the number of pixels within region $\mathcal{R}$. As shown in Fig. 3, the RoI-level flow is calculated by taking the mean optical flow of all pixels within the RoI. Then, position features of different frames are aligned to the reference frame for affinity computation. Given the refined position of an RoI in its own frame, we use four coordinates $(x_{\min},\ x_{\max},\ y_{\min},\ y_{\max})$ to identify the position feature. The alignment of the position feature $\boldsymbol{f}^{\text{pos}}_{k^*\to k}$ to the reference frame k is represented by

$$\boldsymbol{f}^{\text{pos}}_{k^*\to k}=(x_{\min}+v^x, x_{\max}+v^x, y_{\min}+v^y, y_{\max}+v^y), \tag{2}$$

where $\boldsymbol{v}$ is short for $\boldsymbol{v}_{k^*\to k}$, and the superscripts x and y denote flow directions.

3.3 Metric network

In common multi-object trackers, history templates of a track are exploited to capture richer information. We follow this idea and design the metric network to describe the affinity between a track and a detection.

A track can be seen as a collection of detections that belong to the same target. Given a track of length T and a detection, there are two alternative approaches for computing their affinity. In early fusion the features of the T frames in the track are fused into a detection-compatible feature, and then the pairwise detection-detection affinity is computed, whereas in late fusion the detection-detection

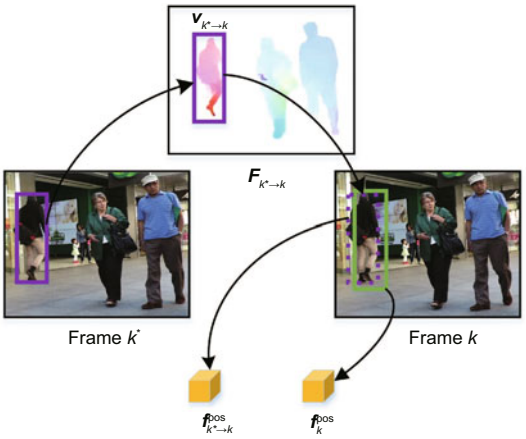

Fig. 3 Position feature alignment. The solid purple box of frame k^* is aligned to frame k by RoI-level flow denoted as the dashed purple box. The green box of frame k remains the same. References to color refer to the online version of this figure

distance between each frame of the track and the detection is first computed, and then these T distances are fused into the final affinity. To avoid information loss, we follow the latter fusion by taking the mean value of the T affinities as the fused affinity.

The metric network consists of two parallel branches that learn the appearance and position metrics, respectively. The input of the appearance metric network is the appearance feature $\boldsymbol{f}^{\text{app}}$ of dimension D, while the input of the position metric network is the aligned four-dimensional (4D) position feature $\boldsymbol{f}^{\text{pos}}$. To model the affinity metric, we use a tiny sub-network similar to that in Han et al. (2015). As shown in Fig. 4, it consists of three FC layers. Rectified linear unit (ReLU) nonlinearity is applied to the first two FC layers. For both appearance and position metric networks, the number of units in the first two FC layers is set equal to the appearance feature dimension D. The last FC layer contains a single unit, followed by sigmoid activation. Although the appearance and position metric networks have identical architecture, their weights are learned independently without sharing. For either metric network, given a pair of features, we first compute their squared element-wise difference, and feed it into the network. After computing the affinities $s_1, s_2, \ldots, s_T$ between each track template and the given detection, their mean value $\bar{s}$ is used as the final affinity of input track and detection.

Note that we have actually tried several architectures for the appearance and position metric

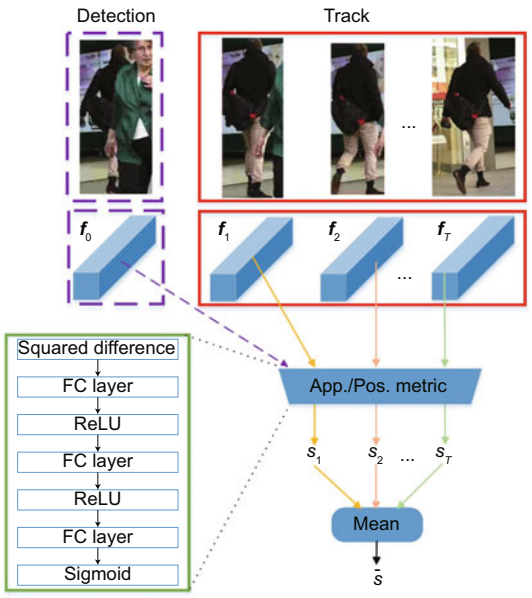

Fig. 4 Appearance metric network (App. metric) and position metric network (Pos. metric). s_i is the affinity of feature f_i (i^{th} frame of track) and f_0 (detection)

networks, such as varying the depth and the number of hidden units of FC layers, using an LSTM unit to encode the track features and then matching the encoded feature with the detection feature as in Sadeghian et al. (2017). Also, we have tried learnable weights to fuse the appearance and position affinity matrices. However, our experiments show that the tracking performance rarely differs. Hence, for simplicity we make the two metric networks share the same architecture and use the mean value for affinity matrix fusion. We want to convey the fact that the improvement of performance comes from AAN itself rather than an elaborate design of network architectures.

3.4 Training

3.4.1 Multi-task loss

The proposed AAN has three output layers. The classification loss $\mathcal{L}_{\text{cls}}$ and bounding box regression loss $\mathcal{L}_{\text{reg}}$ are used to handle detection imperfection as mentioned in Section 3.2. Additionally, we introduce affinity loss $\mathcal{L}_{\text{a}}$ for associative affinity learning. For a mini-batch of image data, assume that there are M tracks and N detections, and that the metric network generates two $M \times N$ non-negative matrices (one for appearance affinity and the other for position affinity). Rows and columns of the matrices represent tracks and detections, respectively. For each affinity matrix, we apply the binary cross entropy (BCE) loss for elements in the matrix. Because the affinity matrix is overwhelmed by negative samples, we adopt online hard example mining (Shrivastava et al., 2016) to sample the same number of hard negatives as positives. In addition, we introduce the matching cardinality loss, which is a weighted L1 loss that each row should sum to the number of truly matched detections and each column should sum to the number of truly matched tracks.

In summary, the total loss is computed by

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{reg}} + \omega \cdot \mathcal{L}_{\text{a}}, \tag{3}$$

where affinity loss $\mathcal{L}_{\text{a}}$ is given as

$$\mathcal{L}_{\text{a}} = \mathcal{L}_{\text{BCE}}^{\text{app}} + \lambda \mathcal{L}_{\text{L1}}^{\text{app}} + \mathcal{L}_{\text{BCE}}^{\text{pos}} + \lambda \mathcal{L}_{\text{L1}}^{\text{pos}}. \tag{4}$$

To determine an appropriate weight ω for the affinity loss, we have done a grid search and it turns out that $\omega = 1$ is a good choice. Parameter λ should be much smaller than ω to act as a regularizer.

3.4.2 Data collection

Given an annotated video for MOT, we sample $(c+1)$ frames from a short clip of C ($C > c+1$) frames. All these $(c+1)$ frames are involved in training of the classification and bounding box regression branches. Input RoIs are sampled from the union of detections and ground-truth boxes. An RoI is considered as positive if it has bounding box intersection over union (IoU) of at least 0.5 to any ground-truth box and as negative otherwise.

For the training of affinity loss, we make each positive RoI inherit the corresponding target identity of the ground-truth box with the highest IoU. We sample tracks from c out of the $(c+1)$ frames and detections from the remaining one frame. The ground-truth affinity matrix is filled by comparing detection identity and track identity, i.e., "1" for pairs of the same identity and "0" for those of different identities.

4 Multi-object tracker

Within the tracking-by-detection framework, trackers can be categorized into two groups: batch mode and online mode. Trackers in batch mode solve

data association using video frames from future time steps, while trackers in online mode use only previous video frames and the current video frame. Generally, batch processing yields better performance than online processing, but it suffers from severe inefficiency. Hence, we present a multi-object tracker that follows the tracking-by-detection scheme and runs in online mode. Because effectiveness of AAN in the context of MOT is our main concern, we keep the multi-object tracker as simple as it can be, which will be presented in Section 4.3 in detail.

4.1 Data association

Given a set of tracks and detections at time k, the data association problem is to determine the detection to which each track should be linked. We construct a bipartite graph via a trained AAN and solve the bipartite matching problem using the Hungarian algorithm. The detection feature can be directly obtained in the AAN, whereas the track feature generation is a bit trickier.

Features of each track are extracted from the recorded history information within a time window of size T. Notice that the track length is c in training and will not exceed T in inference. The appearance feature is used directly, whereas the position feature should be aligned to the detection frame. We do not explicitly compute RoI-level flow from each of the T frames to the detection frame because it is too time-consuming. The RoI-level flow of bounding boxes from different frames to the detection frame is calculated by the cumulative sum of its recorded RoI-level flow. For example, assume that at detection frame k, the RoI-level flow of an RoI from frame $k-2$ to k is $\boldsymbol{v}_{(k-2)\to k} \approx \boldsymbol{v}_{(k-2)\to(k-1)} + \boldsymbol{v}_{(k-1)\to k}$. This trick offers us the benefit of computing optical flow between two adjacent frames only. Once a track is associated with a detection, we use the position shift of the target from the track's last frame to the detection frame as the updated RoI-level flow. This alleviates the cumulative flow error because the position of refined detection is more accurate.

Assume that there are M tracks and N detections. AAN takes detections for refinement and feature extraction, and then outputs the appearance and position affinity matrices of size $M \times N$ given external track features. The final affinity A is the mean of appearance and position affinities.

Following the online processing mode, we build a bipartite matching graph given the affinity matrix. The assignment is solved optimally using the Hungarian algorithm. Additionally, a minimum affinity τ_1 is imposed to reject assignments where the track-detection associative affinity is too low. For matched track-detection, detection is appended to the track as a new state. Data association is performed twice as in Wojke et al. (2017), first for confirmed tracks and detections, and then for tentative tracks and remaining detections.

Notably, in our experiments we find that during inference using $s_{\max} = \max(s_1, s_2, \ldots, s_T)$ instead of $\bar{s}$ shown in Fig. 4 also works, and it is more robust to track length. This can be seen as a greedy template matching strategy.

4.2 Single-object tracking

The goal of SOT for MOT is to maintain trajectory consistency in the presence of missed detections. CNN-based single-object trackers have been shown to be helpful for MOT in recent works (Chu Q et al., 2017). Zhang et al. (2020) proposed the FlowTracker module as SOT. Here, we show that AAN itself is capable of tracking a single object without any modification.

Given a track of length T at time k, we can use RoI-level flow to generate T candidate RoIs located at time k. These RoIs can be treated as predictions or pseudo detections with prior identity. AAN extracts features of these T candidate RoIs at time k and outputs the affinity of these predictions to the track itself, which we call self-affinity. Then, the predicted box with the highest affinity score is chosen as the track's new state. In other words, AAN uses motion information to generate potential candidates, and performs SOT by finding the best match from these candidates to the track itself. Note that we perform only SOT for targets not under heavy occlusion, and occlusion reasoning is handled as proposed in Andriyenko et al. (2011).

Using AAN for SOT can help reduce the number of false negatives, but at the risk of introducing more false positives. This is a critical issue for any single-object tracker caused by drifting when an object gets occluded and its feature slowly drifts away. Eventually the bounding box will drift to other objects or background. We set a self-affinity threshold τ_2, such that predictions with a self-affinity lower than τ_2 will

be eliminated to alleviate drifting. To avoid the predicted box being a duplicate of any actual detection, we set its self-affinity to zero if its predicted box has IoU to actual detections greater than γ.

In practice, we find that for each track, generating one candidate from the RoI in the last frame suffices. Moreover, SOT is performed only for tracks that have been confirmed but cannot be matched with any detection. In a common case, the detections miss targets as partial (slight) occlusion happens. With SOT, we are able to recover the missing target. When a target becomes heavily (say fully) occluded, the tracker keeps the last updated RoI-level flow to maintain the motion tendency, and does not update appearance or position features until the target is associated with any detection again. In this case, the last updated RoI-level flow would be recorded as the RoI-level flow for these missing frames and later used for spatial-temporal alignment.

4.3 Track management

For video frames $\{I_1, I_2, \ldots, I_K\}$ with detections $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K\}$, we present track management in our multi-object tracker as follows, including initialization, association, state update, termination, and postprocessing:

1. Initialization

At frame k, detections that are not associated with any confirmed or tentative tracks are legacy detections $\mathcal{D}_k^{\text{legacy}}$. Each legacy detection will be initiated as a tentative track. These tracks remain tentative until they are matched to detections in the next few frames. Each track maintains a queue of length $\leq T$ that records the track features for affinity estimation in association. At frame 1, each detection of $\mathcal{D}_1$ is initialized as a tentative track.

2. Association

At frame k ($k \geq 2$), we perform the following steps:

Step 1: compute optical flow $\boldsymbol{F}_{(k-1)\to k}$ from frame $k-1$ to frame k using pre-trained FlowNet2. For each confirmed track, we calculate the RoI-level flow of its previous state to generate pseudo detections $\mathcal{D}_k^{\text{fake}}$.

Step 2: concatenate real detections and pseudo detections $\mathcal{D}_k^{\text{all}} = \mathcal{D}_k \cup \mathcal{D}_k^{\text{fake}}$, and then use the feature network for feature extraction.

Step 3: compute the affinity of $\mathcal{D}_k^{\text{all}}$ to both confirmed tracks $\mathcal{T}_{\text{con}}$ and tentative tracks $\mathcal{T}_{\text{ten}}$ via the metric network given their features. We first adopt the Hungarian algorithm to perform data association of confirmed tracks $\mathcal{T}_{\text{con}}$ to real detections $\mathcal{D}_k$. Then, remaining detections $\mathcal{D}_k^{\text{remain}}$ (except those matched to confirmed tracks) are associated to tentative tracks $\mathcal{T}_{\text{ten}}$ via the Hungarian algorithm.

Step 4: for each confirmed track that is not matched to real detections, we use SOT to recover its current state if its self-affinity exceeds τ_2.

3. State update

For each confirmed track, its features are stored as the current state if it is matched, while the lost frame counter accumulates if it is unmatched to any detections $\mathcal{D}_k^{\text{all}}$. Each tentative track is changed into a confirmed track if matched in T_{ini} consecutive frames, and features are recorded if matched.

4. Termination

Confirmed tracks are terminated and appended to dead tracks $\mathcal{T}_{\text{dead}}$, if they are not detected for T_{lost} frames to avoid track number growth. Tentative tracks that fail to match any detection in their first T_{ini} frames are deleted.

5. Postprocessing

Any trajectory that is too short is removed because it is likely to be a false positive. In addition, trajectories of the same target may be disrupted, which commonly happens when an object is occluded for some time. We link trajectories of that kind before and after the temporal gap via bounding box interpolation assuming linear velocity. We deal with only missing parts of a small gap. Notice that our multi-object tracker is purely in online mode. Postprocessing in a short time window can be used in near-online mode.

The initialization of tentative tracks and termination of dead tracks are handled at the end of each time step. We adopt track initialization and termination as proposed in Wojke et al. (2017), with $T_{\text{ini}} = 3$ and $T_{\text{lost}} = 30$.

Note that our multi-object tracker does not have a specific technique for occlusion handling. Targets under mild occlusion can be detected and associated properly because of RoI-level flow, whereas targets under heavy occlusion are missing targets. These missing targets would be detected and associated properly once their states change to no occlusion or mild occlusion again. In postprocessing, bounding

box interpolation can recover targets under heavy occlusion for a short time period.

5 Experiments

5.1 Datasets

We evaluate the performance of our multi-object tracker for pedestrian tracking on MOT16 and MOT17 from the MOTChallenge benchmark (https://motchallenge.net/). The MOT17 dataset contains 14 video sequences in unconstrained environments, which are divided into two subsets, seven for training and seven for testing. Public object detections from three detectors are provided, namely DPM (Felzenszwalb et al., 2010), faster R-CNN (Ren et al., 2017), and SDP (Yang F et al., 2016). The ground-truth annotations of training sequences are released, and performance on the testing sequences is evaluated by the benchmark. The MOT16 dataset contains only DPM detections, but has the same images as MOT17. The ground-truth annotation of the MOT17 dataset is a bit more accurate than that of MOT16, but the difference is slight.

5.2 Implementation details

The training batch of AAN consists of four ($c = 3$) frames from a clip of eight ($C = 8$) consecutive frames sampled from multiple training videos. For each frame, we sample 128 RoIs of which half are positive and half are negative. The backbone network is VGG-16 pre-trained on the ImageNet classification task. RoIAlign is used to pool a 7×7 feature map for each RoI. The appearance feature dimension is $D = 128$. The weight in multi-task loss is $\lambda = 0.01$. We use stochastic gradient descent (SGD) for optimization with a learning rate of 1×10^{-3} and a weight decay of 5×10^{-4}. We train AAN for nine epochs and reduce the learning rate to 1×10^{-4} after the 5^{th} epoch. We select the best model as the model with the highest tracking measure MOTA (multi-object tracking accuracy) on the validation set.

Parameter settings of the multi-object tracker are as follows: $T = 20$ for track length, $\tau_1 = 0.5$ to reject wrong association, $\tau_2 = 0.8$, and $\gamma = 0.7$ for SOT. In postprocessing, trajectories with fewer than five frames are removed, and the temporal gap for interpolation is 15. Visualization of a challenging tracking scenario with target occlusions is presented in Fig. 5.

5.3 Evaluation metrics

The MOTChallenge employs multiple metrics for MOT performance evaluation, including MOTA↑, multi-object tracking precision (MOTP↑), mostly tracked targets (MT↑, percentage of ground-truth objects covered at least 80%), mostly lost targets (ML↓, percentage of ground-truth objects covered less than 20%), false positives (FP↓), false negatives (FN↓), ID switches (IDS↓), and the number of frames processed in one second (Hz↑). Identification F-measure (IDF1↑), identification precision (IDP↑), and identification recall (IDR↑) are also used to evaluate trajectory consistency (Ristani et al., 2016). For items with ↑, a higher score means a better result. For those with ↓, a lower score means a better result.

5.4 Ablation studies

We split the original training set of MOT17 into training and validation sets. The training set contains four sequences, i.e., MOT17-04, MOT17-10, MOT17-11, and MOT17-13, while the validation set contains the remaining three sequences, i.e., MOT17-02, MOT17-05, and MOT17-09. In addition to the classical train/val split protocol, we train on

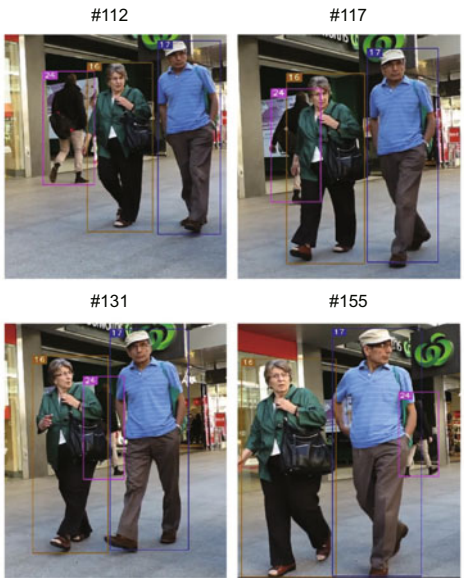

Fig. 5 Visualization of tracking results. Several consecutive frames of the MOT17-09 sequence are shown where occlusions happen

the union of training and validation sets denoted as trainval and validate on trainval. The two protocols are used to validate the generalization and fitting abilities of our approach.

To verify the advantage of AAN, we evaluate different variants of our approach (Table 1). For comparison, triplet loss is used to learn appearance feature extraction independently. Cosine similarity (normalized cosine distance into $[0, 1]$) and IoU are used as hand-crafted metrics for appearance and position features, respectively. App. and Pos. denote the appearance and position metrics proposed in Section 3.3 respectively, and affinity loss is described in Section 3.4.

5.4.1 Hand-crafted vs. learned metric

We notice that the learned appearance metric is superior to cosine similarity, whereas the learned position metric is inferior to IoU in both of the two settings as shown in Table 1. This indicates the necessity of training a binary classifier to fully use the appearance feature for matching. On the other hand, even we align the position feature with optical flow, it is still difficult to learn a discriminative metric to distinguish targets of various motion patterns. The reason for this lies in the complexity of motion under perspective projection and camera movement for monocular videos (Tang et al., 2017). For the trainval setting, the non-parametric IoU yields a 3.4% higher MOTA than the proposed position metric. We investigate our learned position metric, and find that it always outputs a lower value than the IoU for matched pairs. Learning motion pattern for tracking remains an open question.

5.4.2 Appearance feature & metric learning

To validate the appearance feature learned by AAN, we compare it with the feature learned by triplet loss. To rule out the impact of the metric, we fix it to the hand-crafted metric (Cos.+IoU). As shown in Table 1, the proposed affinity loss results in lower IDS and higher IDP than triplet loss. Hence, affinity loss helps learn a more discriminative appearance feature than triplet loss. A little benefit of triplet loss is that it helps reduce the number of FNs and obtain a higher number of MTs. However, because the appearance feature learned by triplet loss is not very discriminative, SOT could recover more missed detections and also introduce more FPs and IDSs. After all, joint learning of the appearance feature and metric still achieves the best tracking performance in terms of both MOTA (59.7%) and IDS (1089), denoted as affinity loss with App.+IoU in Table 1. Therefore, joint appearance feature and metric learning is beneficial for data association in MOT.

5.4.3 Abilities of fitting and generalization

As shown in Table 1, AAN with the App.+IoU configuration achieves the best performance (in terms of MOTA) in both train/val and trainval settings. Compared with triplet loss, AAN achieves an almost 27% decrease in IDS and a 2.6% increase in IDF1 in the trainval setting. This highlights the strong fitting and generalization abilities of AAN, and indicates that AAN would achieve more promising tracking performance if trained on larger datasets. Therefore, the following experiments are conducted with the trainval and App.+IoU configuration.

5.4.4 Impact of optical flow

We study the impact of optical flow for bounding box alignment when computing position affinity. A viable alternative is to use a Kalman filter for RoI-level motion estimation. As shown in Table 2, replacing optical flow with a Kalman filter results in a slight performance drop in terms of IDF1 and IDS, verifying the effectiveness of optical flow. Compared with the Kalman filter, RoI-level flow is more robust against fast-moving persons (which may also be caused by a fast-moving camera). The overall performance improvement is not significant because the MOT dataset barely contains fast-moving persons. For videos taken from a moving vehicle like sequence MOT17-13, the performance gain is significant (Table 3). However, such frames with fast movement occupy only a small portion of the whole dataset. Nevertheless, we incorporate RoI-level flow to make the proposed approach more robust against varying tracking scenarios, hopefully beyond the MOT dataset.

5.4.5 Impact of single-object tracking

To validate the importance of SOT using AAN, we deliberately disable the SOT function and

Table 1 Tracking performance of different feature and metric settings

Setting	Loss	Metric	IDF1↑ (%)	IDP↑	IDR↑	MT↑	ML↓	FP↓	FN↓	IDS↓	MOTA↑ (%)	MOTP↑
train/val	Triplet	Cos.+IoU	48.6	67.8%	37.9%	**166**	**205**	5182	**45 924**	534	44.2	79.0%
	Affinity	Cos.+IoU	47.2	69.7%	35.6%	130	237	3049	48 247	490	44.0	80.3%
	Affinity	Cos.+Pos.	36.2	63.5%	25.3%	50	341	**1599**	57 201	699	35.7	**80.8%**
	Affinity	App.+IoU	49.5	71.3%	37.9%	153	**205**	3638	46 888	**475**	**44.8**	79.7%
	Affinity	App.+Pos.	**50.0**	**72.0%**	**38.3%**	149	217	3772	47 014	**475**	44.6	79.8%
trainval	Triplet	Cos.+IoU	61.4	78.1%	50.6%	**563**	**408**	8905	**127 219**	1483	59.2	**86.9%**
	Affinity	Cos.+IoU	61.7	80.2%	50.2%	515	452	**5192**	131 118	1254	59.2	86.4%
	Affinity	Cos.+Pos.	58.8	76.8%	47.6%	466	486	7961	135 946	1588	56.8	84.2%
	Affinity	App.+IoU	**64.0**	**82.0%**	**52.5%**	551	439	6835	127 993	**1089**	**59.7**	85.8%
	Affinity	App.+Pos.	58.6	75.4%	47.9%	502	453	11 380	134 205	1535	56.3	82.6%

App. denotes the appearance metric; Pos. denotes the position metric; Cos. denotes the cosine similarity. ↑: a higher score means a better result; ↓: a lower score means a better result. The best results are in bold

Table 2 Impact of optical flow and single-object tracking (SOT) on tracking performance

Setting	Optical flow	SOT	IDF1↑ (%)	IDP↑	IDR↑	MT↑	ML↓	FP↓	FN↓	IDS↓	MOTA↑ (%)	MOTP↑
train/val	×	✓	49.2	71.1%	37.6%	**154**	296	3495	47 051	484	**44.8**	79.7%
	✓	×	48.5	**71.9%**	36.5%	146	215	**2942**	48 393	**468**	44.0	**80.1%**
	✓	✓	**49.5**	71.3%	**37.9%**	153	**205**	3638	**46 888**	475	**44.8**	79.7%
trainval	×	✓	63.7	81.7%	52.2%	**551**	446	6988	128 483	1151	59.4	85.8%
	✓	×	62.0	81.7%	50.0%	502	469	**5240**	136 151	**1061**	57.7	**86.3%**
	✓	✓	**64.0**	**82.0%**	**52.5%**	**551**	**439**	6835	**127 993**	1089	**59.7**	85.8%

↑: a higher score means a better result; ↓: a lower score means a better result. The best results are in bold

evaluate its tracking performance. As shown in Table 2, without SOT the number of FNs increases significantly. The proposed SOT using AAN is surprisingly simple but effective in boosting MOT performance by decreasing the number of FNs remarkably without introducing too many FPs or IDSs.

An intriguing fact is that we achieve similar performance on MOT16 and MOT17 in spite of their difference in detections. This is because using AAN for SOT boosts the tracking performance more significantly for detectors of low recall (i.e., DPM). See Table 4 for the detailed improvements over these detectors. Together with detection refinement, our approach is quite robust against detection quality.

5.5 Comparison with the state-of-the-art approaches

To evaluate our approach on the MOT16 and MOT17 testing sequences, we submit our results to the MOTChallenge benchmark evaluation server. Our performance on the two datasets is shown in Table 5. We also compare our tracker to tracktor++_v2 using the same private detections as provided in Bergmann et al. (2019b), denoted as Ours (private) in Table 5. Because optical flow is not necessary for MOT datasets, we also provide our method on the MOT17 test set without optical flow (w/o optical flow), which runs at 9.8 frames/s. It is clear that using better detection improves the performance of AAN significantly, indicating that we can enhance AAN by replacing the VGG-16 backbone with a stronger one like ResNet-101 with feature

Table 3 Impact of optical flow on MOT17-13 (trainval, App.+IoU)

Optical flow	IDF1↑ (%)	FP↓	FN↓	IDS↓	MOTA↑ (%)
×	58.5	**1741**	16 012	227	48.5
✓	**59.5**	1966	**15 451**	**198**	**49.6**

↑: a higher score means a better result; ↓: a lower score means a better result. The best results are in bold

Table 4 Impact of single-object tracking (SOT) for different detectors (trainval, App.+IoU)

Detector	SOT	IDF1↑ (%)	IDS↓	MOTA↑ (%)
DPM	×	55.2	**292**	50.2
	✓	**57.7**	303	**53.4**
Faster R-CNN	×	59.7	**356**	53.5
	✓	**62.1**	369	**55.2**
SDP	×	70.2	**413**	69.4
	✓	**71.4**	417	**70.4**

↑: a higher score means a better result; ↓: a lower score means a better result. The best results are in bold

Table 5 Results on the MOT16 and MOT17 test datasets (trainval, App.+IoU)

Dataset	Tracker	MOTA↑ (%)	IDF1↑ (%)	MT↑	ML↓	FP↓	FN↓	IDS↓	Hz↑
MOT16	UMA (Yin et al., 2020)	50.5	52.8	17.8%	33.7%	7587	81 924	685	5.0
	CRF_TRACK (Xiang J et al., 2021)	50.3	**54.4**	18.3%	35.7%	7148	82 746	702	1.5
	Tracktor++ (Bergmann et al., 2019a)	**54.4**	52.5	19.0%	36.9%	**3280**	79 149	682	1.5
	LMP (Tang et al., 2017)	48.8	51.3	18.2%	40.1%	6654	86 245	595	0.5
	GCRA (Ma et al., 2018)	48.2	48.6	12.9%	41.1%	5104	88 586	821	2.8
	MOTDT (Chen L et al., 2018)	47.6	50.9	15.2%	38.3%	9253	85 431	792	**20.6**
	NOMT (Choi, 2015)	46.4	53.3	18.3%	41.4%	9753	87 565	**359**	2.6
	STAM (Chu Q et al., 2017)	46.0	50.0	14.6%	43.6%	6895	91 117	473	0.2
	MHT_DAM (Kim et al., 2015)	45.8	46.1	16.2%	43.2%	6412	91 758	590	0.8
	LINF1 (Fagot-Bouquet et al., 2016)	41.0	45.7	11.6%	51.3%	7896	99 224	430	4.2
	Ours	53.8	53.4	**21.9%**	**32.0%**	6535	**76 880**	756	6.6
	Tracktor++_v2 (Bergmann et al., 2019b)	56.2	54.9	20.7%	35.8%	**2394**	76 844	**617**	1.6
	Ours (private)	**59.9**	**60.3**	**28.9%**	**22.3%**	11 018	**61 217**	939	5.8
MOT17	UMA (Yin et al., 2020)	53.1	54.4	21.5%	31.8%	22 893	239 534	2251	5.0
	CRF_TRACK (Xiang J et al., 2021)	53.1	53.7	**24.2%**	**30.7%**	27 194	234 991	2518	1.4
	FAMNet (Chu P and Ling, 2019)	52.0	48.7	19.1%	33.4%	14 138	253 616	3072	–
	Tracktor++ (Bergmann et al., 2019a)	53.5	52.3	19.5%	36.6%	**12 201**	248 047	2072	1.5
	DAN (Sun et al., 2021)	52.4	48.7	21.4%	**30.7%**	25 423	234 592	6431	6.3
	TAT (Shen et al., 2018)	51.5	–	20.6%	35.5%	–	–	2593	–
	FWT (Henschel et al., 2018)	51.3	47.6	21.4%	35.2%	24 101	247 921	2648	0.2
	jCC (Keuper et al., 2016)	51.2	**54.5**	20.9%	37.0%	25 937	247 822	**1802**	1.8
	MOTDT (Chen L et al., 2018)	50.9	52.7	17.5%	35.7%	24 069	250 768	2474	**18.3**
	MHT_DAM (Kim et al., 2015)	50.7	47.2	20.8%	36.9%	22 875	252 889	2865	0.9
	Ours	**53.9**	54.3	23.7%	32.0%	27 656	**230 042**	2386	5.4
	Tracktor++_v2 (Bergmann et al., 2019b)	56.3	55.1	21.1%	35.3%	**8866**	235 449	**1987**	1.5
	Ours (private)	**58.8**	**59.5**	**27.8%**	**24.5%**	30 888	**198 864**	2880	5.8
	Ours (private) w/o optical flow	58.6	59.2	**27.8%**	**24.5%**	31 467	199 275	3024	**9.8**

↑: a higher score means a better result; ↓: a lower score means a better result. The best results are in bold

pyramid networks (FPNs) or changing the feature network to Center-like networks used in Zhou et al. (2020).

Compared with state-of-the-art trackers, competitive performance is achieved in terms of MOTA, IDF1, and IDS, which represent the performance of tracking continuity. In addition, our approach achieves the first-tier MT and ML, which is attributed mainly to the single-object tracker using AAN.

5.6 Discussion

The experiments show that AAN is handy and useful for tracking multiple objects in videos. AAN exhibits outstanding performance: detection refinement for better localization and appearance feature representation, joint learning of features and metrics for a more discriminative model, and SOT to discover missed detections. AAN achieves these improvements by an end-to-end trained CNN. In addition, considering multi-task design, AAN has the potential to become a unified framework for joint detection and tracking.

6 Conclusions

We have proposed a new neural network architecture as an affinity model for the task of multi-object tracking. Appearance and position features and their affinity metrics have been learned jointly in an end-to-end manner. Based on an associative affinity network, we have introduced a single-object tracker that can maintain trajectory consistency in the presence of missed detections. With a simple multi-object tracker, competitive performance has been achieved on the MOT16 and MOT17 datasets.

Contributors

Liang MA and Qiaoyong ZHONG contributed to methodology, validation, and writing. Yingying ZHANG

contributed to experiment design. Di XIE and Shiliang PU contributed to supervision and project administration.

Compliance with ethics guidelines

Liang MA, Qiaoyong ZHONG, Yingying ZHANG, Di XIE, and Shiliang PU declare that they have no conflict of interest.

References

Andriyenko A, Roth S, Schindler K, 2011. An analytical formulation of global occlusion reasoning for multi-target tracking. IEEE Int Conf on Computer Vision Workshops, p.1839-1846. https://doi.org/10.1109/ICCVW.2011.6130472

Bergmann P, Meinhardt T, Leal-Taixé L, 2019a. Tracking without bells and whistles. IEEE/CVF Int Conf on Computer Vision, p.941-951. https://doi.org/10.1109/ICCV.2019.00103

Bergmann P, Meinhardt T, Leal-Taixé L, 2019b. Tracktor++_v2. Available from https://github.com/phil-bergmann/tracking_wo_bnw [Accessed on July 9, 2020].

Bullinger S, Bodensteiner C, Arens M, 2017. Instance flow based online multiple object tracking. IEEE Int Conf on Image Processing, p.785-789. https://doi.org/10.1109/ICIP.2017.8296388

Chen L, Ai HZ, Zhuang ZJ, et al., 2018. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. IEEE Int Conf on Multimedia and Expo, p.1-6. https://doi.org/10.1109/ICME.2018.8486597

Chen S, Gong C, Yang J, et al., 2018. Adversarial metric learning. Proc 27th Int Joint Conf on Artificial Intelligence, p.2021-2027. https://doi.org/10.24963/IJCAI.2018/279

Chen S, Luo L, Yang J, et al., 2019. Curvilinear distance metric learning. Proc 33rd Int Conf on Neural Information Processing Systems, p.4223-4232.

Choi W, 2015. Near-online multi-target tracking with aggregated local flow descriptor. IEEE Int Conf on Computer Vision, p.3029-3037. https://doi.org/10.1109/ICCV.2015.347

Chu P, Ling HB, 2019. FAMNet: joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. IEEE/CVF Int Conf on Computer Vision, p.6171-6180. https://doi.org/10.1109/ICCV.2019.00627

Chu Q, Ouyang WL, Li HS, et al., 2017. Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism. Proc IEEE Int Conf on Computer Vision, p.4846-4855. https://doi.org/10.1109/ICCV.2017.518

Dalal N, Triggs B, 2005. Histograms of oriented gradients for human detection. IEEE Computer Society Conf on Computer Vision and Pattern Recognition, p.886-893. https://doi.org/10.1109/CVPR.2005.177

Duan YQ, Lu JW, Zheng WH, et al., 2020. Deep adversarial metric learning. *IEEE Trans Image Process*, 29:2037-2051. https://doi.org/10.1109/TIP.2019.2948472

Emami P, Ranka S, 2018. Learning permutations with sinkhorn policy gradient. https://arxiv.org/abs/1805.07010

Fagot-Bouquet L, Audigier R, Dhome Y, et al., 2016. Improving multi-frame data association with sparse representations for robust near-online multi-object tracking. Proc 14th European Conf on Computer Vision, p.774-790. https://doi.org/10.1007/978-3-319-46484-8_47

Fang K, Xiang Y, Li XC, et al., 2018. Recurrent autoregressive networks for online multi-object tracking. IEEE Winter Conf on Applications of Computer Vision, p.466-475. https://doi.org/10.1109/WACV.2018.00057

Feichtenhofer C, Pinz A, Zisserman A, 2017. Detect to track and track to detect. IEEE Int Conf on Computer Vision, p.3057-3065. https://doi.org/10.1109/ICCV.2017.330

Felzenszwalb PF, Girshick RB, McAllester D, et al., 2010. Object detection with discriminatively trained part-based models. *IEEE Trans Patt Anal Mach Intell*, 32(9):1627-1645. https://doi.org/10.1109/TPAMI.2009.167

Han XF, Leung T, Jia YG, et al., 2015. MatchNet: unifying feature and metric learning for patch-based matching. IEEE Conf on Computer Vision and Pattern Recognition, p.3279-3286. https://doi.org/10.1109/CVPR.2015.7298948

He KM, Gkioxari G, Dollãr P, et al., 2017. Mask R-CNN. IEEE Int Conf on Computer Vision, p.2980-2988. https://doi.org/10.1109/ICCV.2017.322

Henschel R, Leal-Taixé L, Cremers D, et al., 2018. Fusion of head and full-body detectors for multi-object tracking. IEEE/CVF Conf on Computer Vision and Pattern Recognition Workshops, p.1509-1518. https://doi.org/10.1109/CVPRW.2018.00192

Hermans A, Beyer L, Leibe B, 2017. In defense of the triplet loss for person re-identification. https://arxiv.org/abs/1703.07737

Ilg E, Mayer N, Saikia T, et al., 2017. FlowNet 2.0: evolution of optical flow estimation with deep networks. IEEE Conf on Computer Vision and Pattern Recognition, p.1647-1655. https://doi.org/10.1109/CVPR.2017.179

Keuper M, Tang SY, Yu ZJ, et al., 2016. A multi-cut formulation for joint segmentation and tracking of multiple objects. https://arxiv.org/abs/1607.06317

Kim C, Li FX, Ciptadi A, et al., 2015. Multiple hypothesis tracking revisited. IEEE Int Conf on Computer Vision, p.4696-4704. https://doi.org/10.1109/ICCV.2015.533

Lan L, Tao DC, Gong C, et al., 2016. Online multi-object tracking by quadratic pseudo-Boolean optimization. Proc 25th Int Joint Conf on Artificial Intelligence, p.3396-3402.

Leal-Taixé L, Canton-Ferrer C, Schindler K, 2016. Learning by tracking: Siamese CNN for robust target association. IEEE Conf on Computer Vision and Pattern Recognition Workshops, p.418-425. https://doi.org/10.1109/CVPRW.2016.59

Ma C, Yang CS, Yang F, et al., 2018. Trajectory factory: tracklet cleaving and re-connection by deep Siamese Bi-GRU for multiple object tracking. IEEE Int Conf on Multimedia and Expo, p.1-6. https://doi.org/10.1109/ICME.2018.8486454

Maksai A, Wang XC, Fleuret F, et al., 2017. Non-Markovian globally consistent multi-object tracking. IEEE Int Conf on Computer Vision, p.2563-2573. https://doi.org/10.1109/ICCV.2017.278

Milan A, Rezatofighi SH, Garg R, et al., 2017a. Data-driven approximations to NP-hard problems. Proc 31st AAAI Conf on Artificial Intelligence, p.1453-1459.

Milan A, Rezatofighi SH, Dick A, et al., 2017b. Online multi-target tracking using recurrent neural networks. Proc 31st AAAI Conf on Artificial Intelligence, p.4225-4232.

Nummiaro K, Koller-Meier E, van Gool L, 2003. An adaptive color-based particle filter. *Image Vis Comput*, 21(1):99-110. https://doi.org/10.1016/S0262-8856(02)00129-4

Ren SQ, He KM, Girshick R, et al., 2017. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Patt Anal Mach Intell*, 39(6):1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

Rezatofighi SH, Milan A, Zhang Z, et al., 2015. Joint probabilistic data association revisited. IEEE Int Conf on Computer Vision, p.3047-3055. https://doi.org/10.1109/ICCV.2015.349

Ristani E, Tomasi C, 2018. Features for multi-target multi-camera tracking and re-identification. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.6036-6046. https://doi.org/10.1109/CVPR.2018.00632

Ristani E, Solera F, Zou R, et al., 2016. Performance measures and a data set for multi-target, multi-camera tracking. European Conf on Computer Vision, p.17-35. https://doi.org/10.1007/978-3-319-48881-3_2

Sadeghian A, Alahi A, Savarese S, 2017. Tracking the untrackable: learning to track multiple cues with long-term dependencies. IEEE Int Conf on Computer Vision, p.300-311. https://doi.org/10.1109/ICCV.2017.41

Schulter S, Vernaza P, Choi W, et al., 2017. Deep network flow for multi-object tracking. IEEE Conf on Computer Vision and Pattern Recognition, p.2730-2739. https://doi.org/10.1109/CVPR.2017.292

Shen H, Huang LC, Huang C, et al., 2018. Tracklet association tracker: an end-to-end learning-based association approach for multi-object tracking. https://arxiv.org/abs/1808.01562

Shrivastava A, Gupta A, Girshick R, 2016. Training region-based object detectors with online hard example mining. IEEE Conf on Computer Vision and Pattern Recognition, p.761-769. https://doi.org/10.1109/CVPR.2016.89

Son J, Baek M, Cho M, et al., 2017. Multi-object tracking with quadruplet convolutional neural networks. IEEE Conf on Computer Vision and Pattern Recognition, p.3786-3795. https://doi.org/10.1109/CVPR.2017.403

Sun SJ, Akhtar N, Song HS, et al., 2021. Deep affinity network for multiple object tracking. *IEEE Trans Patt Anal Mach Intell*, 43(1):104-119. https://doi.org/10.1109/TPAMI.2019.2929520

Tang SY, Andriluka M, Andres B, et al., 2017. Multiple people tracking by lifted multicut and person re-identification. IEEE Conf on Computer Vision and Pattern Recognition, p.3701-3710. https://doi.org/10.1109/CVPR.2017.394

Wang B, Wang L, Shuai B, et al., 2016. Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association. IEEE Conf on Computer Vision and Pattern Recognition Workshops, p.386-393. https://doi.org/10.1109/CVPRW.2016.55

Wang XY, Han TX, Yan S, 2009. An HOG-LBP human detector with partial occlusion handling. Proc IEEE 12th Int Conf on Computer Vision, p.32-39. https://doi.org/10.1109/ICCV.2009.5459207

Wojke N, Bewley A, Paulus D, 2017. Simple online and realtime tracking with a deep association metric. IEEE Int Conf on Image Processing, p.3645-3649. https://doi.org/10.1109/ICIP.2017.8296962

Xiang J, Sang N, Hou JH, et al., 2016. Hough forest-based association framework with occlusion handling for multi-target tracking. *IEEE Signal Process Lett*, 23(2):257-261. https://doi.org/10.1109/LSP.2015.2512878

Xiang J, Xu GH, Ma C, et al., 2021. End-to-end learning deep CRF models for multi-object tracking. *IEEE Trans Circ Syst Video Technol*, 31(1):275-288. https://doi.org/10.1109/TCSVT.2020.2975842

Xiang Y, Alahi A, Savarese S, 2015. Learning to track: online multi-object tracking by decision making. IEEE Int Conf on Computer Vision, p.4705-4713. https://doi.org/10.1109/ICCV.2015.534

Yang B, Nevatia R, 2014. Multi-target tracking by online learning a CRF model of appearance and motion patterns. *Int J Comput Vis*, 107(2):203-217. https://doi.org/10.1007/S11263-013-0666-4

Yang F, Choi W, Lin YQ, 2016. Exploit all the layers: fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. IEEE Conf on Computer Vision and Pattern Recognition, p.2129-2137. https://doi.org/10.1109/CVPR.2016.234

Yin JB, Wang WG, Meng QH, et al., 2020. A unified object motion and affinity model for online multi-object tracking. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.6767-6776. https://doi.org/10.1109/CVPR42600.2020.00680

Zhang JMY, Zhou SP, Chang X, et al., 2020. Multiple object tracking by flowing and fusing. https://arxiv.org/abs/2001.11180

Zhou XY, Koltun V, Krähenbühl P, 2020. Tracking objects as points. https://arxiv.org/abs/2004.01177