IET Computer Vision

Review Article

Deep learning for multiple object tracking: a survey

ISSN 1751-9632
Received on 17th September 2018
Revised 11th November 2018
Accepted on 3rd January 2019
E-First on 1st May 2019
doi: 10.1049/iet-cvi.2018.5598
www.ietdl.org

Yingkun Xu[1], *Xiaolong Zhou*[1], *Shengyong Chen*[1,2] ✉, *Fenfen Li*[3]

[1]*College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, People's Republic of China*
[2]*School of Computer Science and Engineering, Tianjin University of Technology, Tianjin 300384, People's Republic of China*
[3]*Computer and Software Engineering, Huaiyin Institute of Technology, Huaian, People's Republic of China*
✉ *E-mail: sy@ieee.org*

Abstract: Deep learning has been proved effective in multiple object tracking, which confronts the difficulties of frequent occlusions, confusing appearance, in-and-out objects, and lack of enough labelled data. Recently, deep learning based multi-object tracking methods make a rapid progress from representation learning to network modelling due to the development of deep learning theory and benchmark setup. In this study, the authors summarise and analyse deep learning based multi-object tracking methods which are top-ranked in the public benchmark test. First, they investigate functionality of deep networks in these methods, and classify the methods into three categories as description enhancement using deep features, deep network embedding, and end-to-end deep network construction. Second, they review deep network structures in these methods, and detail the usage and training of these networks for multi-object tracking problem. Through experimental comparison of tracking results in the benchmarks in total and by group, they finally show the effectiveness of deep networks for tracking employed in different manners, and compare the advantages of these networks and their robustness under different tracking conditions. Moreover, they analyse the limitations of current methods, and draw some useful conclusions to facilitate the exploration of new directions for multi-object tracking.

1 Introduction

Developing efficient and robust systems to approximate human vision mechanism is essential for computer vision techniques. Recently, deep learning based algorithms have made promising advances to this target. In the field of image classification, Krizhevsky *et al.* [1] make a breakthrough in ILSVRC 2012 competition. They reduce the error rate nearly ten percent in comparison with conventional methods. Then, new deeper neural network architectures are proposed for image classification [2–5], and human vision is overwhelmed in this task [2, 3]. Deep learning methods are proved available for related tasks of image classification. Rapid progress is made in face recognition [6, 7], person re-identification [8, 9], action recognition [10, 11], image semantic segmentation [12], and so on. The success of deep learning in image categorisation is attributed to huge data collection and labelling, acceleration of parallel computation, and new deep neural network architectures. Motivated by the success of deep learning in image classification, tasks of object localisation, object detection, and motion prediction are also benefited from deep learning [13–17].

All the advances in computer vision tasks mentioned above inspire improvement in visual tracking problem, which aims to localise and link to trajectory for specific objects. However, it is not straightforward to utilise deep learning algorithms for object tracking. Because tracking a manually selected object is a semi-supervised learning task, there is a lack of enough samples to learn features for the targets. The early deep learning based tracking algorithm [18] is inferior compared with correlation filter [19] or sparse principal component analysis [20]. In order to adopt the deep learning for object tracking, various strategies have been explored [21–26]. These strategies can be roughly classified into three aspects. (i) Extensive samples are constructed to promote the feature learning for object tracking [21, 22]. (ii) Features from low layers or multiple layers from deep convolutional neural networks (CNNs) are extracted, which appear more discriminative than high level features [23, 24]. (iii) End-to-end deep networks are designed and trained to obtain the tracking results directly [25, 26]. These deep learning methods promote tracking performance impressively. Recently, Li *et al.* [27] give a review to summarise deep learning based algorithms for tracking single object.

In contrast with tracking single specific object, multiple object tracking (MOT) is more complicated. Besides of the difficulties in tracking single object, tracking multiple objects in one category needs to create new tracked objects using detection results, re-identify lost objects when they appear again, or terminate objects when they go out of the field of view of camera. In addition, the problems of occlusion, background clutter, pose changing are more sophisticated than those in tracking single object [28]. To cope with these challenges, some deep learning based methods are proposed [29–44]. For example, it is feasible to replace conventional hand-crafted features by features extracted from deep neural network to associate detection results, even though the features are learned from tasks of classification or recognition [32, 33]. Moreover, it is proved that the performance can be improved when the attributes of MOT, such as temporal and spatial attention map or temporal order, are explored [36, 37]. Furthermore, some end-to-end deep learning architectures are designed to extract the features not only for appearance descriptor but also for motion information [34, 38, 39].

Although deep learning methods are effective for MOT problem, there is much room to improve the tracking performance with the power of deep learning by considering its great success in the fields of image classification and recognition [45]. Therefore, it is necessary to summarise and analyse existing deep learning based MOT algorithms to pave the way of study deep learning methods for multi-object tracking further.

In this paper, we first review the existing deep learning based MOT methods, and then discuss how the deep learning methods improve the performance of MOT, what the main functionality of deep learning in multi-object tracking is, and how they are implemented. Finally, we attempt to suggest some directions to improve the tracking performance further.

The remaining of this paper is organised as follows. In Section 2, we review the related works. In Section 3, we illustrate general frameworks of MOT, and give a brief overview of deep learning algorithms within these frameworks. In Sections 4 and 5, we investigate these tracking methods in details. The recent multi-object tracking benchmarks are illustrated in Section 6, and tracking results are compared on the benchmarks, through which we discuss the effectiveness of deep learning in MOT, and give some directions for further exploration. The paper is summarised in the final section.

2 Related works

In this section, we review the main works for MOT and present some related survey papers about it. MOT is an important topic in the field of computer vision. It is formulated variously when the inputs, goals or optimisations are different. According to the way of trajectory generation, MOT algorithms can be roughly classified as offline multi-object tracking and online multi-object tracking.

Offline methods collect overall detections in an entire sequence beforehand, and treat MOT as partitions of the detection set. Maximising posterior probability of the partitions can be converted to different formulations through assumptions in different level. Considering the linkages between detections of neighbouring frames as edges between nodes, graph models are constructed to describe the tracking problem. The graphs can be solved by min-cost network flow [46–51], k-shortest path [52], and sub-graph decomposition based on multi-cut [53] or multi-clique [54]. To add the relative constraints, the maximisation of probability can be formulated as inference of Markov random fields, in which the constraints can be imposed to describe the dependency and exclusion for motion and occlusion [55–58] or the relations between discrete measurements and continuous states of trajectories [59, 60]. Except of global inference of tracks, some methods build the tracking results in a hierarchical manner [61–63]. During this multi-stage procedures, the scenario information can be estimated [61], and discriminative features are updated [62, 63].

In online tracking methods, trackers receive current image data and detection results in each frame. It needs to decide which tracks the detections should be associated to, or whether they should be utilised to create new tracks. Treating online tracking as states estimation of joint distribution, the Bayesian inference can be employed. Through assumption of Gaussian distribution for transition or Gaussian Mixture for density, the joint probabilistic data-association filter [64] and Gaussian mixture probability hypothesis density filter [65, 66] are used to estimate and update the states of tracked objects. For non-Gaussian cases, particle filter methods are utilised by considering the exclusion between objects [67] and updating the discriminative features [68]. The process of states' updating can be approximated as matching between tracks and detections. Different costs for matching are designed using appearance features of tracklets [69], learned by random forests [70], or based on Markov decision process [71]. A special category of online tracking is near-online methods [72–74], which look forward for several frames and then make decisions for states' updating. The decisions are made according to accumulated likelihood [72], energy minimisation [73], or re-ranking results by historical tracks [74].

Although object tracking problem is studied for decades, it is applied for radar system or service as a stage for behaviour analysis in surveillance system in early time. There are some review papers related to these systems [75–78]. For human motion capture and crowd behaviour analysis system, individual tracking is employed to estimate human pose and find anomaly behaviour [75, 76]. For surveillance system, human and vehicle tracking must be considered for complex scenario analysis [77, 78]. In tracking algorithms, tracking general single object is investigated more widely than multi-object tracking. There are some papers published to survey the tracking systems and experiments [27, 79–83]. The overall tracking system is specified in [79], while the appearance models and trends of next development are discussed in [80, 81]. To compare the results of different methods fairly and promote the tracking results in various environments, the benchmark for tracking is built [82] and experimental evaluations are given [83]. Further, the state-of-art methods based on deep learning are summarised and compared in [27].

By contrast, there are less literature reviews concentrating on the MOT methods and analysing its challenges. One progress is the development of recent benchmarks for MOT task [84, 85], which collect several new high-resolution video samples and some video sequences used in conventional methods. Common detections and evaluation metrics are used for fair comparison. Leal-Taixe *et al.* [45] analyse the results of recent trackers from the aspects of sample difficulties and various tracking errors, and attempt to propose some inspiring directions, including deep learning based association. Luo *et al.* [28] give a review for MOT, but recent state-of-the-art methods based on deep learning are not concerned. Instead, we focus on investigation of deep-learning based MOT algorithms, which are competitive and top-ranked recently on the MOT benchmarks.

3 Deep learning based multi-object tracking overview

To facilitate the understanding of deep learning based MOT algorithms reviewed in this work, we illustrate them using their abbreviated names within online and offline tracking frameworks in Fig. 1. We organise these methods according to the functionality of deep networks and the way of state optimisation. The details about these methods are listed in Table 1.

In online MOT framework, main challenges lie in how to learn robust associating metric of linking the detections to tracks, when to create new tracks by distinguishing the true detecting results from false positive ones, and when to terminate the lost tracks. Specifically designed deep learning methods provide promising solutions to these challenges. DeepSort [33] and CDA-DDAL [86] learn the appearance features from person re-identification task to associate with detections. RAN [40] and AMIR [39] predict motion and appearance features by auto-regression and matching classification using Long Short-Term Memory (LSTM) networks. STAM-MOT [36] applies spatial and temporal attention map to handle the partial occlusion problem in tracking. recurrent neural network LSTM (RNN-LSTM) [34] designs end-to-end deep neural network to learn the association between tracks and detections, statement update, initialisation and termination of tracks like particle filter as in AP-HWDPL [35]. To find the optimal location of objects, STAM-MOT [36] employs the dense searching strategies, which are utilised commonly in tracking single object. Two deep learning based methods using MHT framework are MHT-DAM [42] and MHT-bLSTM [87], in which the CNN and bilinear LSTM networks are employed to learn appearance features.

In the offline MOT framework, the main challenges are how to construct the graph and network, and how to optimise the global labelling problem of them. In order to handle the problems of offline tracking, detection-to-detection pairwise similarity as well as those between short tracks is learned in SiameseCNN [32] and CNNTCM [30]. To obtain more accurate similarity metric, the additional information, such as temporal sequential order, is considered in Quad-CNN [37]. These sequential features are strengthened in GCRA [41]. Further, researchers find that network optimisation, such as multi-cut flow [53], can be improved by deep flow features in JointMC [31] and lifted cut edges in LMP [43]. Besides, network flow can also be optimised globally in end-to-end learning manner in DeepNetWork [38].

Through exploration of deep learning algorithms in MOT frameworks, it is proved that deep learning is effective in improving tracking performance from aspects of tracking prediction and data association. Similar as in image recognition tasks, learning appearance features automatically by deep CNN can promote discrimination and robustness to occlusion both for online tracking [33, 35, 36] and offline optimisation [32, 37]. Besides, learning motion features by recurrent neural network (RNN) are helpful to promote the accuracy of motion prediction [39, 40], thus to improve the performance of bipartite matching between tracks and detections. By comparison with appearance and motion features learning, relatively less end-to-end learning algorithms are explored for the complex components both in online and offline tracking frameworks. It is desired that more approaches can be inspired from the work of RNN-LSTM [34] and DetNetFlow [38], which obtain the tracking results directly by end-to-end learning process.

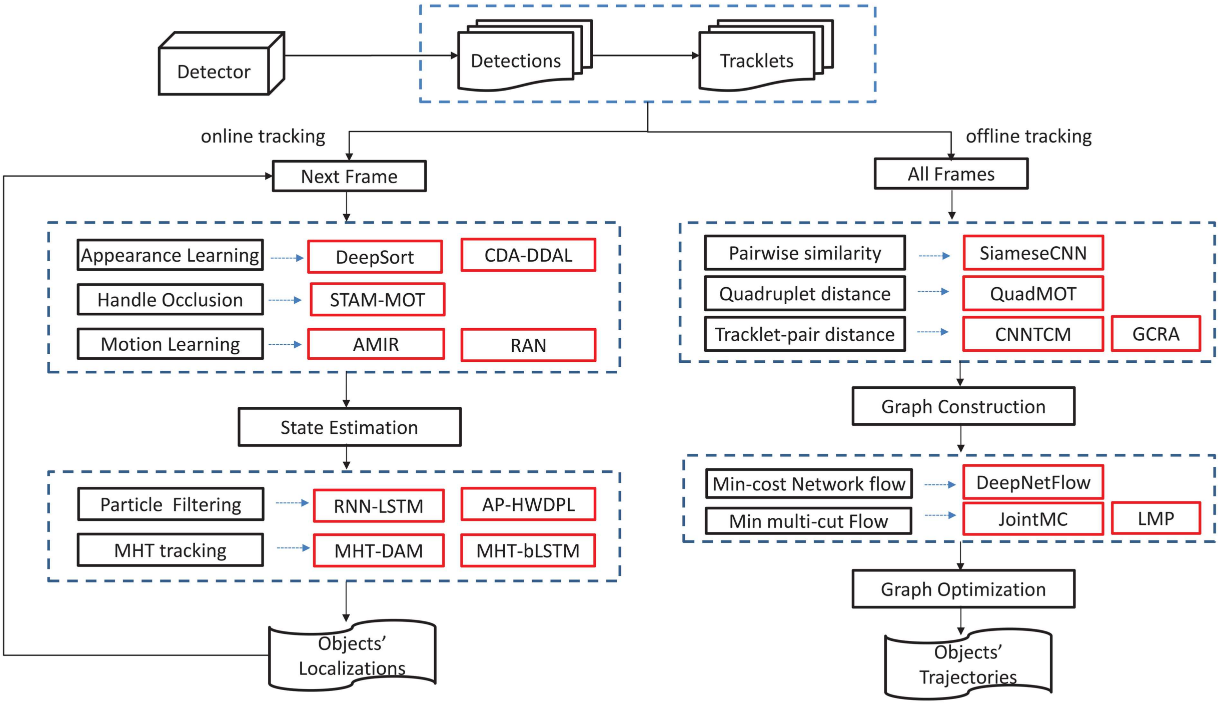

Fig. 1 *Framework of multi-object tracking and deep learning based algorithms investigated in this paper*

Table 1 List of detailed information of multi-object tracking methods investigated in this paper

Method	Year	Full name	Online	Networks	Web link
SiameseCNN [32]	2016	learning by tracking: Siamese CNN for robust target association	×	CNN	https://lealtaixe.github.io/
joint learning of CNNs and temporally constrained metrics(CNNTCM) [30]	2016	joint learning of CNNs and temporally constrained metrics for tracklet association	×	CNN	http://www.eee.ntu.edu.sg/
JointMC [31]	2016	multi-person tracking by multicut and deep matching	×	CNN	https://ps.is.tuebingen.mpg.de/
LMP [43]	2017	multiple people tracking with lifted multicut and person re-identification	×	CNN	https://ps.is.tuebingen.mpg.de/
QuadMOT [37]	2017	multi-object tracking with quadruplet CNNs	×	CNN	http://cvlab.postech.ac.kr/~jeany/
DeepNetFlow [38]	2017	deep network flow for multi-object tracking	×	CNN	http://www.nec-labs.com/
generation cleaving and re-connection association (GCRA) [41]	2018	trajectory factory: tracklet cleaving and re-connection by deep siamese Bi-GRU for MOT	×	GRU	http://www.idm.pku.edu.cn/
DeepSort [33]	2017	simple online and realtime tracking with a deep association metric	✓	CNN	https://github.com/nwojke/deep_sort/
MHT-DAM [42]	2015	multiple hypothesis tracking revisited	✓	CNN	http://rehg.org/mht/
appearance model with DeeP learning from Hua Wei (AP-HWDPL) [35]	2017	online multi-object tracking with CNNs	✓	CNN	http://media.cs.tsinghua.edu.cn/
spatial-temporal attention mechanism for multiple object tracking (STAM-MOT) [36]	2017	online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism	✓	CNN	https://wlouyang.github.io/
confidence-based data association and discriminative deep appearance learning (CDA-DDAL) [86]	2018	confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking	✓	CNN	https://cvl.gist.ac.kr/project/cmot.html/
RNN-LSTM [34]	2017	online multi-target tracking using recurrent neural networks	✓	RNN + LSTM	https://bitbucket.org/amilan/rnntracking/
recurrent autoregressive networks (RAN) [40]	2018	recurrent autoregressive networks for online multi-object tracking	✓	LSTM	https://ai.stanford.edu/~kuanfang/
appearance, motion, and interaction RNNs (AMIR) [39]	2017	tracking the untrackable: learning to track multiple cues with long-term dependencies	✓	LSTM	http://web.stanford.edu/~alahi/
multiple hypothesis tracking with bilinear long short-term memory (MHT-bLSTM) [87]	2018	multi-object tracking with neural gating using bilinear LSTM	✓	LSTM	http://web.engr.oregonstate.edu/~lif/

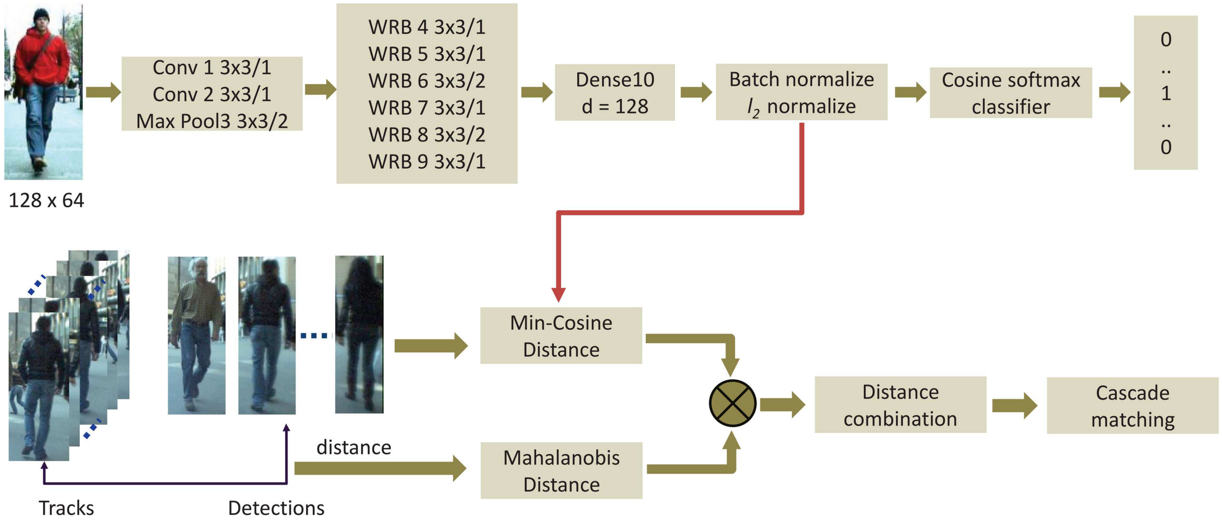

Fig. 2 *Framework of deep SORT [33]. In this paper, the deep features extracted from WRNs for classification are utilised to enhance the matching between detections and tracks (WRB – residual block of WRN)*

As stated above, there are different functionalities of deep networks when they are employed for MOT. Basically, deep features which are learned from the related tasks, e.g. person re-identification, can be combined to promote the tracking performance [32, 33]. Additionally, the specific neural networks can be designed to promote discrimination between different objects or regressive accuracy for one object [35, 40]. Thus, features learned from these neural networks are robust to disturbance during tracking. Further, some end-to-end deep learning architectures for MOT are proposed [34, 38, 39]. These architectures try to model the whole procedure of tracking, always with some assumptions for tracking components. For example, RNN-LSTM [34] assumes global existing probability for objects, and DeepNetWork [38] assumes first order dependency for linking.

4 Deep learning based multi-object tracking analysis

In this section, we analyse and compare deep learning based MOT methods according to deep learning functionalities in tracking framework. We roughly classify the methods into three categories: (i) Multi-object tracking enhancement using deep network features, in which the semantic features are extracted from deep neural network designed for related tasks, and used to replace conventional handcrafted features within previous tracking framework. In most cases, these features extracted from deep networks are effective to promote tracking performance. (ii) Multi-object tracking with deep network embedding, in which the core part of tracking framework is designed using a deep neural network. For instance, the outputs of deep networks are designed as multiple classification scores of detection to different tracks, and binary deep classifier is constructed to represent whether two detections belong to same object or not. (iii) Multi-object tracking with end-to-end deep neural network learning, in which the deep networks are designed directly to obtain the tracking results. Generally, it is hard to obtain multi-object tracking results by only one network because there are some intertwined sub-modules in MOT tracking. Several works attempt to implement this target by making some assumptions such as Markov property, fixed distributions and so on.

4.1 Multi-object tracking enhancement using deep features

The success of deep neural network learning for image classification is due to powerful ability of deep feature learning. These deep features have rich semantic information and are discriminative between different categories. These deep features not only promote the performance of classification, but also are effective to other related tasks, like object detection and image segmentation [88]. Motivated by the effectiveness of deep features, they can be employed to promote the performance of MOT.

Similar as object detection [88] in which CNN is utilised to extract features for region proposals, the deep features extracted from AlexNet [1] are employed in multiple hypotheses tracking (MHT) framework [72]. MHT tracking framework keeps multiple association hypotheses, and constructs a hypothesis tree. Scoring function is designed to select the best hypothesis as track results. Kim *et al.* [42] extend MHT method with appearance features using multi-output regularised least square method. The appearance features are dimensionally reduced from 4096-dimensional deep features. To increase discrimination, Wojke *et al.* [33] employ the deep features extracted from a wide residual network (WRN) for person re-identification task, which are l2-normalised 128-dimensional features before cosine softmax classifier layer [89]. These deep features can be utilised to compute a min cosine distance between detections and track. Through a combination of this cosine distance and motional Mahalanobis distance, the fusing dissimilarities for matching are obtained. The whole tracking framework utilises cascading matching steps according to track age as shown in Fig. 2. This tracking method using deep features from WRN can obtain competitive performance for online tracks while keeping real-time speed.

Considering the goal of feature learning in tracking is to assess the similarity between detections and tracks, Siamese CNN architecture with two same branches is well-suitable to learn the matching features for MOT. There are three types of Siamese CNN topologies: two same branches with one cost layer, two same branches with some shared CNN layers upon, and stacked two stream data forming inputs for CNN layers. Leal-Taixe *et al.* [32] compare these three topologies, and use the third architecture to extract the deep features. By fusing the deep features and motion information with gradient boosting algorithm, they formulate the tracking problem as linear programming and solve it efficiently. Wang *et al.* [30] extend the first architecture of Siamese network to learn the associating affinities between tracklets. The loss of the network is designed as hinge loss of Mahalanobis distance between deep features of sample pairs. To impose temporal constraints for distance learning, segment-wise Mahalanobis distance matrixes are proposed and online multi-task learning algorithm is employed to optimise them. When the matrices are updated, the Siamese network is fine-tuned using samples from tracklets. The whole tracking process is modelled as generalised linear assignment problem, and solved by soft-assign method. Bae and Yoon [86] also utilise the first type of Siamese network to learn affinities for tracklets to replace previous features from ILDA [69]. Different from [30, 32], they employ the online tracking framework, in which the association between tracklets and detections are cascaded in two stages from high confident tracklet to low one.

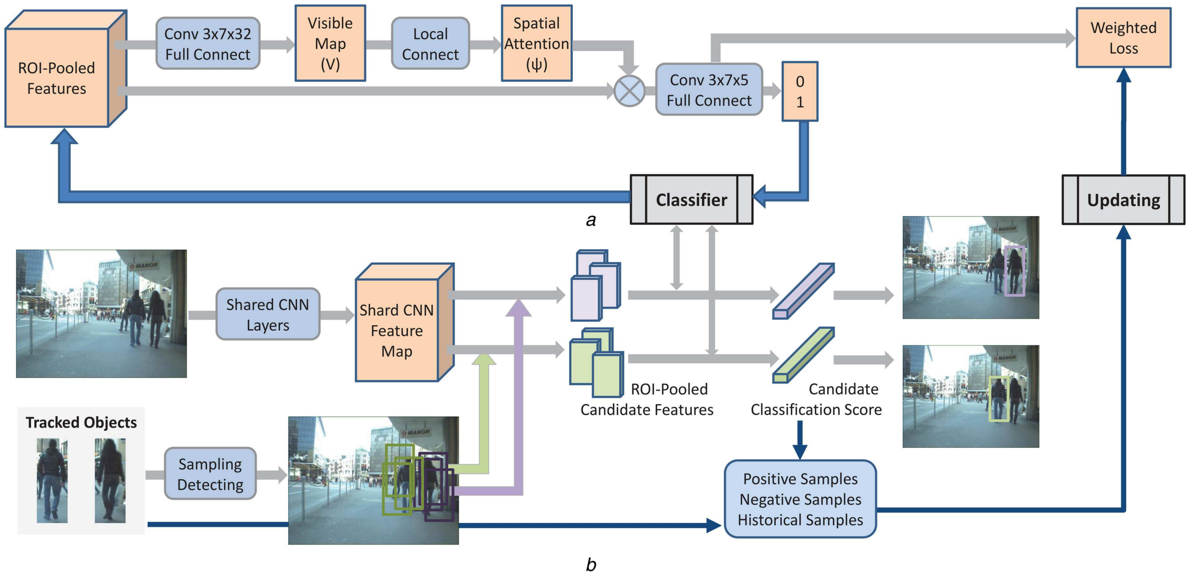

Fig. 3 *Framework of STAM-MOT [36]. In this paper*
(a) Deep CNN is constructed to learn the spatial attention and object specific classifier, and ***(b)*** Sampling based searching method is used to find the best candidate

As stated above, pairwise images can be input into Siamese networks to learn affinities. They are also used to learn optical flow features by deep networks [90]. It is proved that the optical flow features are effective for object tracking and image association [71, 73]. Because deep learning methods can obtain more smoothing and robust optical flow results than traditional algorithms [90], deep optical flow features are expected to promote the tracking performance. Tang *et al.* construct matching cost between detections through deep matching features, and promote association results within multi-cut framework [31]. Furthermore, they discuss that direct matching cost between long-term frames with deep optical flow may lose valid path information and not be reliable for tracking. They add lifted edges to encode deep re-identification features, which are helpful constraints for tracking [43].

4.2 Multi-object tracking with deep network embedding

Comparing with enhancing tracking methods using deep features, it is more efficient to learn the critical components for tracking using deep neural networks. In this category, the networks are designed and embedded as key part of tracking framework, and are trained using samples from tracking data. According to the task of network learning, we can roughly classify these MOT methods with deep network embedding into three types: discriminative deep network learning for MOT (DN-MOT), deep metric learning for MOT (DM-MOT), and generative deep network learning for MOT (GN-MOT).

DN-MOT: Tracking-by-detection is commonly adopted paradigm in state-of-the-art tracking methods. In these methods, object trackers optimise discriminative models and search the best locations in the next frames according to the models. Because deep networks are widely employed for discriminative tasks, it is natural to extend the discriminative deep network models to tracking task. For instance, Chen *et al.* [35] propose an object-specific particle filtering framework for MOT. To track each object, two CNN based classifiers are constructed. These two classifiers utilise the features from different layers of VGG-16 model [5] based faster-RCNN [13] as inputs. The first classifier extracts features from top-layer to classify person instance, whereas the second extracts features from low-layer and compares with object's historical features to decide whether they are same or not. The confidences of these two classifiers are combined to obtain weights for particles, and the final tracking results are obtained using particle filtering. In this method, the networks are offline trained, whereas the objects' historical features are online updated. Similar as in [35], Chu *et al.* [36] build MOT framework using object-specific trackers, and each tracker looks for the best candidate among image patch samples and neighbouring detections through an online updated classifier (Fig. 3). In order to handle occlusion between objects, spatial attention features are learned based on visible map using convolution and full connection layers. The spatial attention maps weight the samples' features to promote accuracy of the classifier. In order to reduce time-consuming computation, a CNN feature map is shared to extract the features by ROI pooling layers. The main difference between the work of [35, 36] is that the former attaches a category classifier to reduce samples, while the latter does not consider category classifier but occlusion features. It shows that this category classification is helpful for multi-object tracking task as discussed in Section 6.

Besides of using deep learning for classification tasks, deep networks are utilised to learn regression models. Object detection and single object tracking task can be modelled as regression tasks, and learned with deep networks [14, 91]. In contrast, there are few works in MOT which are modelled by regression learning. However, it is possible to improve the tracking precision by considering the regressive loss. In [37], the regressive loss of bounding box is attached to ranking loss to improve tracking robustness. In [29], Fang model the tracking problem as bounding box regression task using RNN. However, they find that this method cannot obtain convincing results for MOT problems because there are lots of occlusions and similar objects in MOT task.

DM-MOT: MOT methods using detection association need to learn which tracks the incoming detections belong to or whether two detections are from same objects or not. This task can be treated as image patch verification problem. Similar as face recognition [6] and person re-identification [9], learning accurate affinity model or distance metric is desirable for this kind of problems. In addition to using the deep features from neural network for person re-identification [33], it is expected to design suitable deep metric learning networks and learn them for MOT problems directly. Some works have attempted to achieve this goal by fusing the motion features. Son *et al.* [37] propose an extensive Siamese network using quadruplets of image patches as inputs. These image patches are extracted for three detections from one same object and another different one. Thus, the outputs of the network construct a ranking relationship among triple distances reflecting temporal orders for one object and gaps between different ones. The network fuses motion information and appearance features by combination of their distance metrics, and then a connection graph is constructed and solved by min–max label propagation algorithm. Xiang *et al.* [44] propose a triplet loss based CNN to learn the distance metric between trackers and detections. Different from [37], the motion features are learned using difference between LSTM prediction and detections in next frame. Appearance features are extracted from deep networks for person re-identification, and concatenated with motional features to form inputs to the deep metric learning network, which is optimised by a triplet loss [92]. In the tracking process, this distance metric between trackers and detections constructs the cost of bipartite graph, which can be solved efficiently by Hungarian algorithm [39].

Besides of distance metric learning between detection pairs, some researchers attempt to learn distance metric between two tracklets. Ma *et al.* [41] extend Siamese network for tracklet and learn the distance between tracklet pairs. This network extracts features for each detection in tracklets and transfers the features to a bidirectional GRU networks. The output of the bidirectional GRU network is temporally pooled and constructs the overall features for tracklets in Euclidean space. During tracking process, tracklets are generated firstly, and they are cleaved to short sub-tracklets according to local distance between bidirectional GRU outputs. Finally, these sub-tracklets are re-connected to long trajectories using similarity between temporal pooling global features.

GN-MOT: Although deep neural networks are commonly designed for discriminative learning, it is promising to extend deep network for generative learning. Some researchers make exciting progress in this field [93–96]. They show that appropriate parameters for data distribution can be estimated through deep network learning. For MOT problem, some works try to employ deep generative learning to promote tracking performance. Fang *et al.* [40] model posterior probability of object motion status and appearance features with Gaussian distribution using auto linear regression. The parameters are learned using GRU networks, in which the hidden layers are updated in each frame and used to determine the mean and deviation of the distributions for next frame. During tracking process, motion and appearance information are treated as independent, and their joint probability for matching object tracks to detections is calculated. Finally, the greedy matching algorithm [71] is employed to find the best matching results while a preset threshold is used to cut off some low-probability matching results.

Instead of training generative model for prediction, Fernando *et al.* [97] construct an LSTM based generative model for detection confidence map. The image sequence in ten frames is inputted into an encoder composed by stacked convolution layers. A pixel-wise probability map is outputted using Generative Adversarial Network (GAN) model through a decoder following an LSTM layer. Similar as [39], Fernando *et al.* [97] utilise LSTM to do a prediction for object trajectory. This prediction module contains two parts. One is a short-term prediction for association. Another is a long-term prediction for trajectory updating. During tracking process, the object trackers are associated with detections generated from the GAN model above, and non-associated detections are used to create new objects. When the objects are lost for more than ten frames, they are terminated and deleted from tracking system.

4.3 Multi-object tracking with end-to-end deep network learning

In contrast with tracking single object, there are more intertwined components in MOT problems including construction of the relationship between detections, updating the states of tracked objects, and critical submodules of how to initiate new tracks and terminate lost objects and so on. It is hard to model all of these components in one framework and learn them as a whole. Recently, through simplifying tracking process and making some prior assumptions, some end-to-end learning approaches are proposed to implement this target.

In online MOT task, the states of tracked objects can be estimated using recursive Bayesian filter composed of prediction and updating from observations. Inspired by Ondruska and Posner [98], Milan *et al.* [34] extend RNN to model these procedures as shown in Fig. 4. The states of objects, current observations, their matching matrix, as well as existence probabilities are inputted into the network. The predicted states and updated results are outputted, as well as the new existence probabilities, which decide whether the objects should be terminated. To compute the matching matrix, a group of LSTM based networks are designed. Each one is used to model the matching process between one object's state and current observations. The RNN plus a list of LSTMs is trained with end-to-end fashion using sampled track segments. Although the simulations obtain promising tracking results, the tracking results in real MOT evaluation dataset are not well convincing. The reason lies in at least three aspects. First, this approach only considers motion features, while appearance features are omitted. Second, the models of initiation and termination do not consider context information. Third, the training samples are insufficient to learn an optimised model at once. To cope with these problems, Sadeghian *et al.* [39] design a hierarchical RNN structure network to integrate motion, appearance and interaction features for each tracked object. This structure contains three sub LSTM networks, which not only predicts long-term motion features, but also extracts multi-frame appearance and contextual features for trackers. The features of these LSTM networks are concatenated, and inputted to the top LSTM network to obtain the final matching probabilities between trackers and new detections in each frame. To learn this hierarchical RNN structure, the three sub LSTM networks are pre-trained individuals, and fine-tuned from the top RNN. Thus, the whole network structure is trained end-to-end as a matching classifier. The experimental results show it is more robust by considering appearance features than using only motion features, and can also achieve state-of-the-art performance for person re-identification task. Because there are at most six historical frames used to infer the optimal tracking results in RNNs [39], which is not intuitive, Kim *et al.* [87] further explore the detailed operations in LSTM network for learning appearance features. They add a multiplication layer between hidden states and input features to express the incremental regression module, and form a bilinear LSTM module to match between tracks and detections. The modified LSTM presents effectiveness for appearance but not for motion features. Thus, the deep appearance features extracted from bilinear LSTM and motion features from conventional LSTM modules are combined to get the final matching classifier, and trained in end-to-end manner. The tracking employs MHT framework to obtain online tracking results, and the performance is competitive.

In globally optimised MOT, tracking is treated as optimal partition of detection set, which can be modelled using probabilistic graph, network flow and so on. Schulter *et al.* [38] construct an end-to-end deep learning min-cost network flow. The loss function of the deep architecture is defined as weighted l_2 distance of edge labels. This loss is differentiable at network flow cost, which is calculated as deep neural network outputs. Thus, the min-cost network flow with its edges built on multi-layers forms a deep architecture model, and can be optimised using deep learning paradigm. Experimental results show that the global tracking using min-cost network flow is promoted by the deep features. Inspired by this work, it is expected that other global tracking algorithms using graph models or network flows could be extended and promoted by deep architecture.

5 Deep network structure and training

In this section, we discuss deep network structures and their training strategies for MOT task. As there are always a large number of parameters in deep networks, it is critical to train the network correctly and appropriately. There are different network structures according to the functionality and roles in tracking system. These structures mainly include CNN, RNNs, and their integration and variations. Because the training strategies are largely dependent on the network structures, we review the training strategies in the following two sections.

5.1 CNN-based multi-object tracking and training

CNNs are widely used for image classification and recognition because of its outstanding ability for feature learning. When training the CNN models, different objective functions are utilised and various training datasets are needed according to specific tasks.

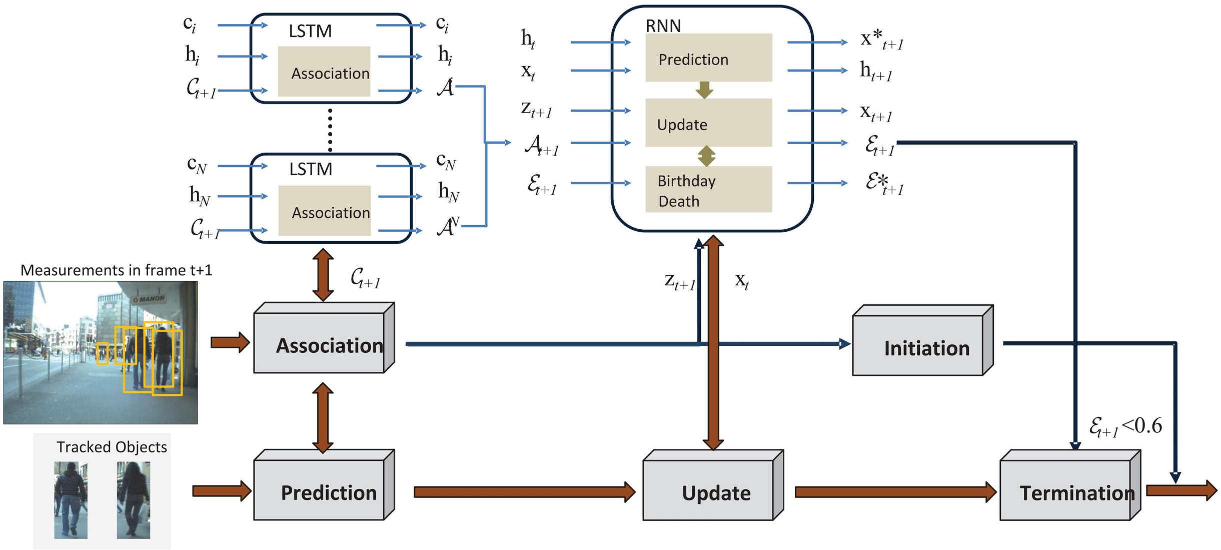

Fig. 4 *Framework of RNN-LSTM tracking [34]. In this paper, an RNN based network is constructed to learn prediction, updated states, and probability of termination. An LSTM based network is used to find best association between detections and objects*

In MOT system, training a CNN model can be roughly classified as training beforehand using classification datasets, training holistically with tracking data, or pre-training initially and then fine-tuning.

A straightforward method to improve tracking performance is to replace hand-crafted features by features extracted from existing CNN models [31, 42] or those trained using classification datasets [33, 43]. Some classification datasets are proved effective to train and obtain high-performance deep CNN models for the task of object tracking, such as image classification dataset ImageNet [4] and person re-identification datasets CUHK03 [99], MARS [100]. For instance, deep-SORT [33] tracking method learns the WRN model via the MARS [100] dataset to obtain a matching metric for person detection pairs, which is applicable for tracking task.

By comparison with person recognition task, there are a large number of missing detections, false alarms, and partial detections in real tracking context. Therefore, it is expected to learn CNN models using real tracking data [32, 35, 37]. In [32], data samples utilise detection pairs. The positive samples use two detections from one tracked object, while negative samples use two detections from two different objects or any patches in the image background. The sampling strategy in [35] is similar, but the patches in the surrounding background with overlapping rate <0.5 are utilised as the negative. In Quad-CNN [37], the main target loss function of deep network is ranking loss. Different from [35], the constraint of temporal orders between detections from same objects is imposed. Thus, one sample is composed of triple detections from two tracks, and hinge loss for ranking instead of cross-entropy loss for classification is employed for training.

For some nested CNNs, it is hard to obtain optimised network models just using end-to-end training. An effective technique is to pre-train the sub networks partially and then fine-tune the whole network finally. In STAM-MOT [36], using existing VGG-16 network [5] as shared layer, each object attaches a CNN based classifier branch, which is composed by three sub networks: a visible map, spatial attention and a classifier. These sub networks are pre-trained initially, and the whole networks are fine-tuned after tracking samples have been collected. In some tracking methods, the network models need to update according to tracking states. This online learning process essentially follows pre-training and fine-tuning paradigm. In CNNTCM [30] tracker, image sequence is split as temporal segments. The Siamese CNN based distance metrics for tracklet pairs in different segments are supposed to be changed but correlated. Thus, the CNN models need to be updated segment by segment.

5.2 RNN-based multi-object tracking and training

By comparison with CNNs, RNNs are suitable for sequence modelling and can predict the next state according to historical information. Thus, it is natural to improve tracking performance by RNNs. However, how to use RNN to integrate appearance and motion features is not straightforward and how to train an RNN model is always difficult. To simplify learning the appearance feature model, some existing CNNs, such as VGG-16 [5], are always employed to extract features as input for RNNs. To train the RNNs appropriately, it can use the strategy of pre-training sub networks partially and then fine-tuning the whole system.

An inspiring method to model object's long-term motion and appearance features is the combination of RNN and LSTM [34, 39, 87]. To learn the states' prediction and matching probability between each object and detection, a modified RNN and an LSTM are employed and trained using mean square error and negative log-likelihood error in [34]. To accommodate the appearance features, LSTM and its extended bilinear version are utilised in [39, 87], which are trained as binary classifiers. These methods pre-train the LSTMs for motion and appearance features individually, and then fine-tuning them using training data for tracking in end-to-end manner.

Recently, GRU based RNNs are utilised to model the reliability and similarity for tracking sequence [41], and do regression for tracking prediction [40]. In [41], two GRU branches form a Siamese structure, which is learned using contrastive loss and cross-entropy loss from global and local aspects. In [40], the hidden layer of GRU network is extended to express the parameters of distribution for motion and appearance features. To train the model, the negative log-likelihood of training samples from MOT tracking datasets is minimised as the loss function.

6 Experiment and discussion

In this section, we summarise the experimental results on MOT2015 and MOT2016 datasets, which are commonly used benchmarks in the recent multi-object tracking algorithms. To analyse the influence of different conditions to tracking and robustness of tracking methods, we divide the image sequences of these datasets into some groups according to their properties. Some useful and inspiring conclusions are obtained through experimental comparison on these groups.

6.1 Datasets and evaluation metrics

The benchmarks for MOT challenge proposed by Milan *et al.* [84, 85] are widely used in state-of-the-art multi-object tracking methods [30–44, 87, 101]. The two datasets, MOT2015 and MOT2016, contain a number of image sequences taken from different scenarios with various distributions of pedestrian detections. The pedestrian detectors are ACF [102] for MOT2015 and DPM [103] for MOT2016. Most of the videos are taken from existing tracking datasets, such as PETS2009 [104], ETHMS [105] and KITTI [106]. Although some other datasets, such as Stanford Drone [107], are employed in several tracking algorithms [39], we conduct the analysis on the MOT datasets for fair comparisons.

There are 22 and 14 image sequences in MOT2015 and MOT2016 datasets individually. Half of the sequences are used for training and the others for testing. We list the test image sequences in Table 2. It is illustrated that the videos are captured from different conditions with low frame rate or real time speed in moving or fixed platforms. Because the crucial modules in object trackers are motion prediction and appearance feature learning, it is necessary to contain different motion and appearance patterns to evaluate tracking performance in the testing datasets. Generally, it is different to predict the motion patterns for videos with low frame rate or on moving platform than with high-rate or from static cameras, therefore we describe these video properties in Table 2. It is similar for appearance conditions, and thus the illumination and weather conditions are listed, as well as the average density of persons at each frame, which always affects whether there are lots of occlusions in the scene. To analyse the different robustness of the tracking algorithms for different conditions, we divide the video sequences into some groups, and compare the tracking results for each group quantitatively.

To evaluate the performance for MOT algorithms, two sets of measurements are mainly used: the CLEAR metrics [108] and the VACE metrics [109]. The former aims to measure the overall performance for all predicted trajectories, which mainly includes MOTA (accuracy of tracking) and MOTP (precision of tracking). The latter is used to describe the individual metrics from different aspects, which contains FP (false positive), FN (false negative), FAF (false alarm per frame), MT (mostly tracked), ML (mostly lost), IDS (number of ID switches), and Frag (number of fragments). Among all the metrics, MOTA is considered as the most agreed with human assess than others [45], which is defined as

$$\text{MOTA} = 1\frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}$$

where g_t, m_t, fp_t and mme_t indicate the number of ground truth, missing tracks, false positives and mismatches in frame t, respectively, [108]. Noticing that MOTA metric is average tracking accuracy for each true tracked object in each frame, we can calculate MOTA metric for any video subset S as below:

$$\text{MOTA}(S) = \sum_{k=1}^{n} \alpha^k \text{MOTA}^k$$

where $\alpha^k = Len(k)*Den(k)/\sum_k (Len(k)*Den(k))$, $Len(k)$ and $Den(k)$ are frame length and density for the kth video sequence in the set S. Similarly, FAF, MT and ML for sequence subset can be calculated as sum of individual metrics by weighting number of trajectories or frame length. In order to analyse different tracking algorithms, we give the tracking results based on MOT benchmark groups, where each group corresponds to video sequence subset under one same condition.

6.2 Tracking results and analysis

Because most of the deep learning based MOT methods do not open source code, but publish their testing results in MOT benchmarks [84, 85], we collect the evaluation results from the site, and then reorganise the data by groups according to different attributes, which affect tracking results from aspects of motion or appearance features.

We illustrate the tracking methods and the comparing results on test sequences of MOT2015 and MOT2016 in Fig. 5. The MOTA metrics for all sequences using each tracking method are connected as a polyline. The names of these sequences are listed below x-axis in descending order of average MOTA, which is drawn with a black dashed polyline. The names of tracking methods are shown in the legend in descending order of overall MOTA metrics.

The detailed evaluation results are illustrated in Table 3, where two additional metrics IDF1 and FPS are considered to show the correctness of detection identification and tracking speed. From results of MOT2015, the end-to-end deep learning methods can get the best results, and deep network embedded methods outperform those methods only using deep features as representation, except for the methods RNN-LSTM and AP-HWDPL, in which the former does not model appearance features and the latter employs a particle filter instead of using detections as input. From results of MOT2016, global optimised methods, LMP and GCRA, exhibit better performance than other methods, including RNN based end-to-end methods, which utilise the online tracking frameworks. Besides, MOTA results in MOT2016 have less deviation than those in MOT2015. The reason may be because detection for MOT 2016 is more stable than that in MOT2015, since MOTP of MOT2016 is larger than MOT2015. While AP-HWDPL and LMP are the best in most evaluated metrics, AMIR presents outstanding in both datasets, whenever the detectors are ACF or DPM.

To analyse the advantages of tracking algorithms and the impact of different conditions for tracking in detail, we split the test video

Table 2 Testing video sequences and their main properties included in the MOT2015 [84] and MOT2016 [85] benchmarks

Dataset	Sequences	Length	Tracks	FPS	Platform	Viewpoint	Density	Weather
MOT 2015	TUD-Crossing	201	13	25	static	horizontal	5.5	cloudy
	PETS09-S2L2	436	42	7	static	high	22.1	cloudy
	ETH-Jelmoli	440	45	14	moving	low	5.8	sunny
	ETH-Linthescher	1194	197	14	moving	low	7.5	sunny
	ETH-Crossing	219	26	14	moving	low	4.6	cloudy
	AVG-TownCentre	450	226	2.5	static	high	15.9	cloudy
	ADL-Rundle-1	500	32	30	moving	horizontal	18.6	sunny
	ADL-Rundle-3	625	44	30	static	horizontal	16.3	sunny
	KITTI-16	209	17	10	static	horizontal	8.1	sunny
	KITTI-19	1059	62	10	moving	horizontal	5	sunny
	Venice-1	450	17	30	static	horizontal	10.1	sunny
MOT 2016	MOT16-01	450	23	30	static	horizontal	14.2	cloudy
	MOT16-03	1500	148	30	static	high	69.7	night
	MOT16-06	1194	221	14	moving	low	9.7	sunny
	MOT16-07	500	54	30	moving	horizontal	32.6	cloudy
	MOT16-08	625	63	30	static	horizontal	26.8	sunny
	MOT16-12	900	86	30	moving	horizontal	9.2	indoor
	MOT16-14	750	164	25	moving	high	24.6	sunny

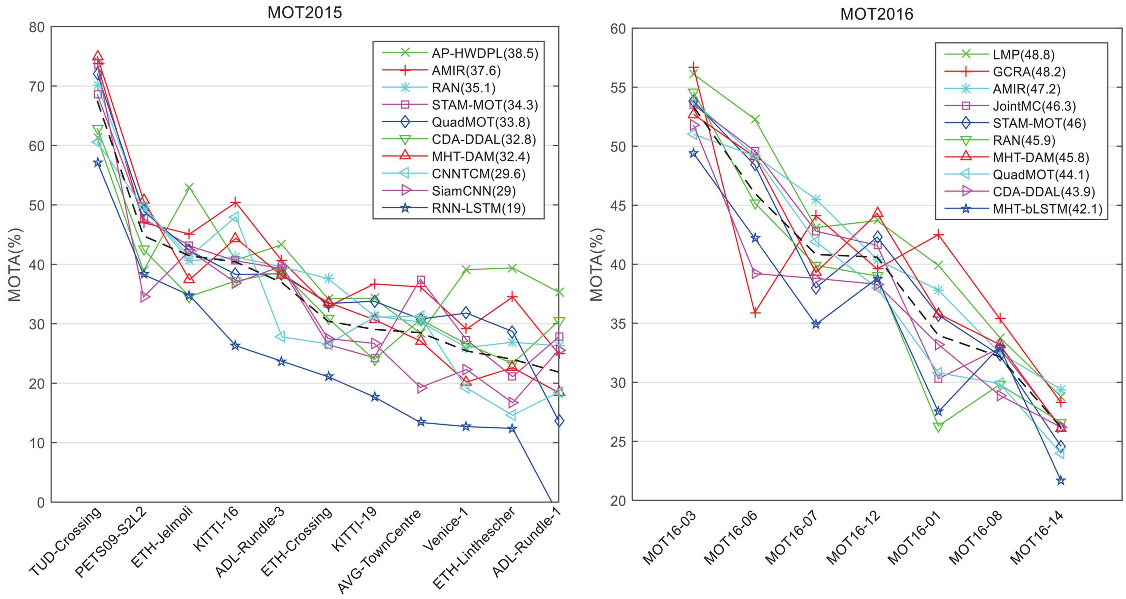

Fig. 5 *Performance of deep learning based multi-object tracking for each sequence on MOT2015 and MOT2016*

Table 3 Evaluation results using CLEAR and VACE for main deep learning based multi-object tracking methods

Method	MOTA	MOTP	IDF1	FAF	MT	ML	FP	FN	IDS	Frag	FPS
MOT2015											
AP-HWDPL [35]	**38.51**	72.6	47.1	**0.70**	8.73%	37.45%	**4005**	33,203	586	1263	6.7
AMIR [39]	37.56	71.7	46.0	1.38	15.81%	**26.77%**	7933	**29,397**	1026	2024	1.9
RAN [40]	35.10	70.9	45.4	1.18	13.04%	42.31%	6771	32,717	381	1523	5.4
STAM-MOT [36]	34.34	70.5	**48.3**	0.88	11.41%	43.39%	5154	34,848	**348**	1463	0.5
QuadMOT [37]	33.81	**73.4**	40.4	1.36	12.89%	36.88%	7898	32,061	703	1430	3.7
CDA-DDAL [86]	32.81	70.7	38.8	0.87	9.71%	42.16%	4983	35,690	614	1583	2.3
MHT-DAM [42]	32.34	71.8	45.3	1.57	**15.96%**	43.82%	9064	32,060	435	**826**	0.7
CNNTCM [30]	29.63	71.8	36.8	1.34	11.25%	43.97%	7786	34,733	712	943	1.7
SiamCNN [32]	29.06	71.2	34.3	0.90	8.46%	48.41%	5160	37,798	639	1316	52.8
RNN-LSTM [34]	18.98	71.0	17.1	1.99	5.53%	45.65%	11,578	36,706	1490	2081	**165.2**
MOT2016											
LMP [43]	**48.75**	**79.0**	**51.3**	1.13	**18.17%**	40.06%	6654	**86,245**	481	**595**	0.5
GCRA [41]	48.15	77.5	48.6	0.87	12.90%	41.10%	5104	88,586	821	1117	**2.8**
AMIR [39]	47.17	75.8	46.3	**0.45**	13.95%	41.62%	**2681**	92,856	774	1675	1.0
jointMC [31]	46.29	75.7	46.3	1.08	15.55%	**39.64%**	6373	90,914	657	1114	0.8
STAM-MOT [36]	45.96	74.9	50.0	1.18	14.62%	43.61%	6895	91,117	**473**	1422	0.2
RAN [40]	45.88	74.8	48.8	1.15	13.18%	41.90%	6871	91,173	648	1992	0.9
MHT-DAM [42]	45.82	76.3	46.1	1.10	16.22%	43.22%	6412	91,758	590	781	0.8
QuadMOT [37]	44.10	76.4	38.3	1.09	14.62%	44.93%	6388	94,775	745	1096	1.8
CDA-DDAL [86]	43.88	74.7	45.1	1.08	10.66%	44.40%	6450	95,175	676	1795	0.5
MHT-bLSTM [87]	42.09	75.9	47.8	1.98	14.88%	44.41%	11,637	93,172	753	1156	1.8

sequences into four groups according to different conditions. For each group, we recount the quantitative tracking results using metrics MOTA, FAF, MT, ML, and FNF (false negative per frame). In addition, we count the gap between MOTA of each group and overall MOTA as dMOTA. Because dMOTA is the average change of accuracy for each true object, it reflects the tracking difficulty under one special condition. We illustrate the results in Tables 4 and 5.

First, we explore the impact of camera motion to tracking results in Table 4. Because motion features reflect object movement and changes of background, it is more complex to learn and predict motion in moving platform than using static camera, which can be approved from the results that the dMOTA metrics in static scenes are all positive, whereas they are negative for a moving platform. Besides, in static platform, most of CNN and RNN based methods achieve approximate results, while RNN based methods are more robust in moving platforms. What' more, the globally optimised algorithms combining high order feature learning, such as GCRA and LMP, present promising results in static platforms, and are expectative for moving cameras.

Second, we divide the test video sequences into two groups according to the lighting conditions in Table 2: videos taken outside on sunny days, and videos taken under low-lighting conditions, including indoor, at night or on cloudy weather. The appearance features of tracked objects under sunny weather are more complicated, because there are lots of illumination changes caused by bright sunlight and shadows. We count the tracking results in these two groups in Table 5. It can be seen that all tracking methods except AP-HWDPL obtain better results under low-light condition than on sunny weather. Maybe it is because the appearance features are easy to keep stable under low-lighting conditions, which is helpful to calculate correct similarity, whereas there are rich contours on sunny weather, which is benefit to generate sufficient candidates for tracking states such as in AP-HWDPL method. In the former conditions, most of trackers present comparable results, while LMP and GCRA outperform the others.

Table 4 Quantitative comparison between static camera and moving camera for main deep learning based MOT methods

Method	MOTA	dMOTA	FAF	FNF	MT	ML	MOTA	dMOTA	FAF	FNF	MT	ML
MOT2015			Static camera						Moving camera			
AP-HWDPL	38.85	0.34	1.01	7.66	8%	24%	**38.07**	−0.44	**0.47**	4.41	9%	51%
AMIR	**41.51**	3.94	1.49	**6.69**	**22%**	**17%**	32.57	−4.99	1.29	**3.97**	9%	**36%**
RAN	39.73	4.63	1.32	7.29	16%	34%	29.24	−5.86	1.07	4.52	10%	51%
STAM-MOT	40.70	6.36	1.10	7.38	15%	28%	26.30	−8.05	0.75	5.09	8%	58%
QuadMOT	40.00	6.19	1.24	7.22	16%	26%	25.97	−7.84	1.45	4.38	9%	48%
CDA-DDAL	37.20	4.39	1.17	7.73	13%	29%	27.25	−5.56	0.65	5.09	7%	55%
MHT-DAM	38.50	6.16	1.62	7.14	18%	35%	24.54	−7.80	1.53	4.43	**14%**	53%
CNNTCM	35.56	5.93	1.81	7.32	15%	29%	22.12	−7.51	1.03	5.09	8%	59%
SiamCNN	32.64	3.58	**0.86**	8.69	8%	38%	24.52	−4.54	0.92	5.04	9%	59%
RNN-LSTM	25.40	6.42	2.31	8.09	5%	33%	10.86	−8.13	1.79	5.14	6%	58%
MOT2016			Static camera						Moving camera			
LMP	52.36	3.61	1.58	**21.90**	**23%**	26%	**40.31**	−8.44	0.77	**8.93**	**16%**	**47%**
GCRA	**53.20**	5.05	1.10	21.94	22%	**24%**	36.34	−11.81	0.68	9.60	9%	49%
AMIR	50.21	3.04	**0.57**	23.95	16%	26%	40.07	−7.10	**0.37**	9.33	13%	49%
JointMC	49.64	3.35	1.42	23.40	19%	26%	38.45	−7.83	0.81	9.17	14%	46%
STAM-MOT	50.07	4.12	2.17	22.45	21%	26%	36.33	−9.63	0.39	9.96	12%	52%
RAN	49.93	4.06	1.75	22.92	18%	25%	36.39	−9.48	0.71	9.61	11%	49%
MHT-DAM	49.30	3.48	1.85	23.13	19%	28%	37.68	−8.14	0.50	9.63	15%	50%
QuadMOT	47.22	3.12	1.49	24.49	16%	30%	36.81	−7.29	0.77	9.49	14%	52%
CDA-DDAL	47.87	3.99	1.43	24.27	15%	29%	34.55	−9.32	0.83	9.77	9%	51%
MHT-bLSTM	46.15	4.07	3.40	23.12	21%	27%	32.58	−9.51	0.86	10.06	12%	52%

Table 5 Quantitative comparison under sunny and low-light condition for main deep learning based MOT methods

Method	MOTA	dMOTA	FAF	FNF	MT	ML	MOTA	dMOTA	FAF	FNF	MT	ML
MOT2015			Low-light						Sunny			
AP-HWDPL	35.99	−2.52	1.17	7.81	5%	28%	**39.63**	1.12	**0.55**	5.14	11%	45%
AMIR	43.72	6.16	1.62	**6.12**	**24%**	**18%**	34.83	−2.74	1.30	**4.78**	10%	**34%**
RAN	42.85	7.75	1.35	6.78	19%	37%	31.65	−3.45	1.12	5.33	9%	46%
STAM-MOT	**43.93**	9.59	**1.08**	6.88	15%	30%	30.08	−4.26	0.84	5.78	9%	53%
QuadMOT	42.63	8.82	1.47	6.51	18%	29%	29.89	−3.92	1.33	5.26	9%	43%
CDA-DDAL	38.61	5.80	1.56	7.05	13%	29%	30.23	−2.58	0.66	5.91	7%	52%
MHT-DAM	42.33	9.99	1.39	6.78	19%	39%	27.90	−4.44	1.62	5.18	**14%**	48%
CNNTCM	42.09	12.46	1.38	6.73	16%	33%	24.09	−5.54	1.34	5.80	8%	52%
SiamCNN	30.67	1.61	1.13	8.57	7%	42%	28.34	−0.72	0.82	5.94	9%	53%
RNN-LSTM	29.06	10.08	1.78	7.96	5%	38%	14.50	−4.48	2.07	5.88	6%	51%
MOT2016			Low-light						Sunny			
LMP	53.01	4.26	1.44	**17.46**	**22%**	29%	**36.41**	−12.35	0.71	**10.81**	**15%**	48%
GCRA	**53.47**	5.32	1.08	17.61	20%	**28%**	32.72	−15.42	0.58	11.52	8%	50%
AMIR	51.23	4.06	**0.57**	19.01	18%	30%	35.41	−11.76	**0.30**	11.36	11%	50%
JointMC	50.39	4.11	1.46	18.50	20%	30%	34.39	−11.89	0.58	11.27	12%	**46%**
STAM-MOT	50.34	4.39	1.82	18.16	19%	31%	33.25	−12.71	0.31	11.78	11%	53%
RAN	50.54	4.67	1.44	18.45	17%	30%	32.35	−13.52	0.80	11.44	10%	50%
MHT-DAM	49.78	3.96	1.49	18.70	21%	33%	34.34	−11.48	0.55	11.33	13%	50%
QuadMOT	48.16	4.06	1.35	19.46	16%	35%	32.35	−11.75	0.72	11.51	13%	52%
CDA-DDAL	48.53	4.66	1.33	19.36	16%	35%	30.39	−13.49	0.77	11.80	7%	51%
MHT-bLSTM	45.98	3.89	2.77	18.94	18%	33%	30.82	−11.27	0.91	11.57	13%	52%

In the latter conditions, AMIR shows more robust to illumination changing when comparing with others.

Similar as camera motion and illumination conditions, the frame rate and view point also affect the difficulties for tracking. By comparison, well-designed deep learning methods using high-order features for motion and appearance can obtain more robust and stable trackers than others. For example, AMIR tracker and LMP tracker, in which the former is an end-to-end learned RNN based tracker, and the latter is globally optimised method with lifted edges encoding the long-term constraints for matching.

6.3 Discussions

The aim of MOT task is to find specific objects in the field of view of cameras, and estimate their moving trajectories from coming in to leaving out the field-of-view. The complexity of this task lies in many aspects. First, the object detections are not always correct or precise. Besides, there are challenges to decide which objects are true incomers or leavers, which detection pairs should be associated, and how to compensate the missing detections. To cope with these problems, traditional MOT methods design and combine effective features to learn similarity between detections [62–71], and estimate the optimal states by graphical models or flow optimisation [46–60]. By comparison, deep learning based methods employ data-driven mechanism to learn the affinity models for detection association. There are different strategies to embed deep learning into tracking algorithms. For instance, deep network

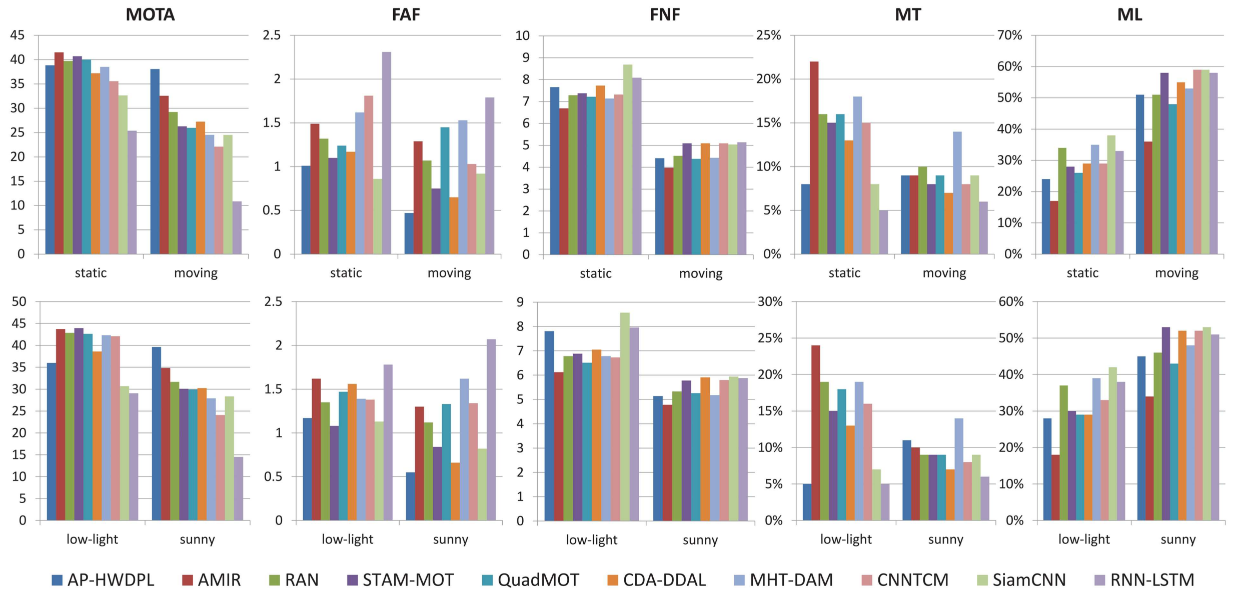

Fig. 6 *Bar chart of evaluation results with different metrics in MOT2015 using deep learning multi-object tracking methods*

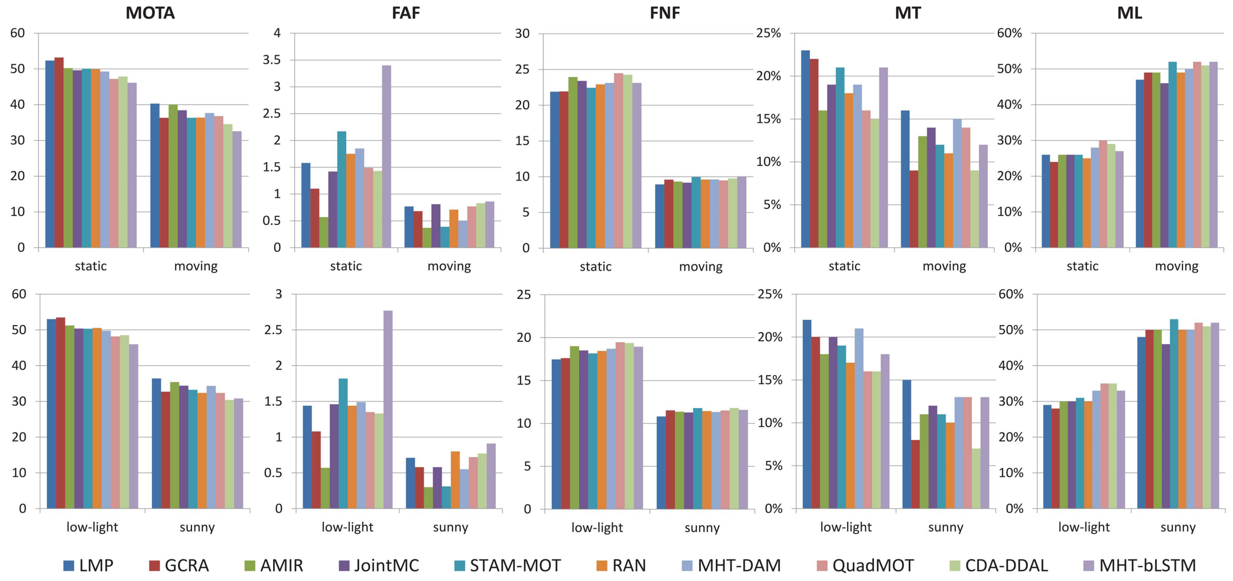

Fig. 7 *Bar chart of evaluation results with different metrics in MOT2016 using deep learning multi-object tracking methods*

classifiers can be used to assess the matching between detections, and deep network features can be employed to replace hand-crafted features within conventional MOT frameworks, such as SiamCNN [32] and JointMC [31].

Deep learning strategies obtain remarkable promotion compared with the baseline methods using same frameworks but no deep features [53, 70], however there are big distinction among these deep learning embedded tracking methods as shown in Table 3. The reason lies in at least three aspects. First, different deep network structures are used in these tracking methods, in which the features of various layers are extracted to describe the objects' appearance. Second, various models are applied in tracking task, which are different at how close they are to catch the essence of tracking problem. Third, the distributions of samples for tracking task are not exactly consistent with those for recognition or localisation. Therefore, transferring the deep features from recognition tasks to tracking directly is not optimal. To apply deep learning for MOT problem more appropriately, tracking-specific deep networks are designed as in [36, 37] and so on. These networks embedded in tracking are proved effective to enhance the appearance descriptor learning. In addition, the recurrent networks provide possible deep models for motion prediction [34, 39, 40], which present a feasible and robust way to promote tracking results.

Although great efforts have been made to design the task-specific deep networks for tracking, the tracking algorithms are not well-suitable for complex conditions as shown in Figs. 6 and 7, which illustrate the MOTA results as well as other metrics under different conditions. To overcome these issues, some research directions would be expectative in future works. First, learning high-order features with deep networks is necessary to reduce the high rate of ML metric, which is closely related with the drops of MOTA under serious conditions. Second, scenario learning is helpful to distinguish the foreground objects and promote motion prediction, especially for the moving platform. Although scenario learning is explored widely in the field of 3D reconstruction, it is less discussed in object tracking task, which is a valuable topic to combine these two tasks. Third, there are lots of false negatives in tracking results as illustrated in the figures, which affect the final tracking results more largely than false positives under the same conditions even for the end-to-end deep learning tracking methods [34, 39]. One exception is APHWDPL [35] which uses detection results only for tracking initiation obtains promising results in moving platform, which indicates tracking by raw image information is a feasible way to handle this issue. Inspired by Fernando *et al.* [97], it can be seen that the advances of deep generative models would pave the way for the new end-to-end deep learning based tracking framework from raw information to trajectories.

7 Conclusion

Great advances of deep learning methods are made in the fields of image recognition, object detection and person re-identification, which also benefit to the development of multi-object tracking. In this paper, we summarise deep learning based multi-object tracking methods, which are top-ranked in public benchmarks. The contribution of this paper lies in three aspects. First, the usage of deep learning for multi-object tracking is organised, and the mechanisms of deep feature transferring, neural network embedding and end-to-end network training are analysed based on existing methods, by which the rules to design new tracking framework are inspired. Second, we investigate the roles of deep networks in tracking framework, and explore the issues of training these networks. Third, comparisons between these multi-object tracking methods are presented and reorganised according to common datasets and evaluations. The advantages and limitations of the methods are stressed. From the analysis of experimental evaluation, it can be seen that there is much room to improve the tracking results by deep learning paradigm. Some useful insights are given in this paper. On one side, it is far from enough labelled datasets to train satisfied models for tracking under all conditions. A possible way can be paved by the generative networks which are outstanding to promote the generalisation for deep learning models. On the other side, to cope with declined tracking results in complex environment such as on moving platform, the integrated network models are required to learn the features of these dynamic scenes. In addition, to adapt to the changing conditions further, learning high-order or online transferred features are expected for the tracked objects.

8 Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61876168, 61403342, U1509207), in part by Foundation of Zhejiang Educational Committee (Y201839708), in part by Fund Project of Jiangsu Engineering Key Laboratory of Mobile Network Technology (JSWLW-2017-012), in part by Zhejiang Provincial Natural Science Foundation of China (LY18F030020).

9 References

[1] Krizhevsky, A., Sutskever, I., Hinton, G.E.: 'Imagenet classification with deep convolutional neural networks'. Proc. Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 2012, pp. 1097–1105

[2] He, K., Zhang, X., Ren, S., *et al.*: 'Delving deep into rectifiers: surpassing human-level performance on imagenet classification'. Proc. IEEE Int. Conf. Computer Vision, Santiago, Chile, 2015, pp. 1026–1034

[3] He, K., Zhang, X., Ren, S., *et al.*: 'Deep residual learning for image recognition'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016, pp. 770–778

[4] Deng, J., Dong, W., Socher, R., *et al.*: 'Imagenet: a large-scale hierarchical image database'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 248–255

[5] Simonyan, K., Zisserman, A.: 'Very deep convolutional networks for large-scale image recognition', CoRR, 2014, abs/1409.1556

[6] Zhang, X., Fang, Z., Wen, Y., *et al.*: 'Range loss for deep face recognition with long-tailed training data'. Proc. IEEE Int. Conf. Computer Vision, Venice, Italy, 2017, pp. 5419–5428

[7] Liu, W., Wen, Y., Yu, Z., *et al.*: 'Sphereface: deep hypersphere embedding for face recognition'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Honolulu, HI, USA, 2017, pp. 6738–6746

[8] Chung, D., Tahboub, K., Delp, E.J.: 'A two stream siamese convolutional neural network for person re-identification'. Proc. IEEE Int. Conf. Computer Vision, Venice, Italy, 2017, pp. 1992–2000

[9] Zhou, S., Wang, J., Wang, J., *et al.*: 'Point to set similarity based deep feature learning for person reidentification'. Proc. IEEE Int. Conf. Computer Vision, Venice, Italy, 2017, pp. 5028–5037

[10] Fernando, B., Gavves, E., Oramas, J., *et al.*: 'Rank pooling for action recognition', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017, **39**, (4), pp. 773–787

[11] Zhao, S., Liu, Y., Han, Y., *et al.*: 'Pooling the convolutional layers in deep convnets for video action recognition', *IEEE Trans. Circuits Syst. Video Technol.*, 2018, **28**, (8), pp. 1839–1849

[12] Geng, H., Zhang, H., Xue, Y., *et al.*: 'Semantic image segmentation with fused CNN features', *Optoelectron. Lett.*, 2017, **13**, (5), pp. 381–385

[13] Ren, S., He, K., Girshick, R.B., *et al.*: 'Faster R-CNN: towards real-time object detection with region proposal networks', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017, **39**, (6), pp. 1137–1149

[14] Dai, J., Li, Y., He, K., *et al.*: 'R-FCN: object detection via region-based fully convolutional networks'. Proc. Advances in Neural Information Processing Systems, Barcelona, Spain, 2016, pp. 379–387

[15] Redmon, J., Divvala, S.K., Girshick, R.B., *et al.*: 'You only look once: unified, real-time object detection'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016, pp. 779–788

[16] Liu, W., Anguelov, D., Erhan, D., *et al.*: 'SSD: single shot multibox detector'. European Conf. on Computer Vision, Amsterdam, The Netherlands, 2016, pp. 21–37

[17] Fan, L., Huang, W., Gan, C., *et al.*: 'End-to-end learning of motion representation for video understanding'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 1–8

[18] Wang, N., Yeung, D.Y.: 'Learning a deep compact image representation for visual tracking'. Proc. Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 2013, pp. 809–817

[19] Henriques, J.F., Caseiro, R., Martins, P., *et al.*: 'High-speed tracking with kernelized correlation filters', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2015, **37**, (3), pp. 583–596

[20] Chen, D., Zhu, M., Wang, H.: 'Visual tracking based on the sparse representation of the PCA subspace', *Optoelectron. Lett.*, 2017, **13**, (5), pp. 392–396

[21] Nam, H., Han, B.: 'Learning multi-domain convolutional neural networks for visual tracking'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016, pp. 4293–4302

[22] Li, J., Zhou, X., Chan, S., *et al.*: 'Object tracking using a convolutional network and a structured output SVM', *Comput. Vis. Media*, 2017, **3**, (4), pp. 325–335

[23] Ma, C., Huang, J.B., Yang, X., *et al.*: 'Hierarchical convolutional features for visual tracking'. Proc. IEEE Int. Conf. Computer Vision, Santiago, Chile, 2015, pp. 3074–3082

[24] Danelljan, M., Robinson, A., Khan, F.S., *et al.*: 'Beyond correlation filters: learning continuous convolution operators for visual tracking'. European Conf. on Computer Vision, Amsterdam, The Netherlands, 2016, pp. 472–488

[25] Valmadre, J., Bertinetto, L., Henriques, J., *et al.*: 'End-to-end representation learning for correlation filter based tracking'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Honolulu, HI, USA, 2017, pp. 5000–5008

[26] Choi, J., Chang, H.J., Fischer, T., *et al.*: 'Context-aware deep feature compression for high-speed visual tracking'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 479–488

[27] Li, P., Wang, D., Wang, L., *et al.*: 'Deep visual tracking: review and experimental comparison', *Pattern Recognit.*, 2018, **76**, pp. 323–338

[28] Luo, W., Xing, J., Milan, A., *et al.*: 'Multiple object tracking: a literature review', arXiv preprint arXiv:1409. 7618, 2014

[29] Fang, K.: 'Track-RNN: joint detection and tracking using recurrent neural networks'. Proc. Advances in Neural Information Processing Systems, Barcelona, Spain, 2016, pp. 1–8

[30] Wang, B., Wang, L., Shuai, B., *et al.*: 'Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association'. Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 2016, pp. 1–8

[31] Tang, S., Andres, B., Andriluka, M., *et al.*: 'Multi-person tracking by multicut and deep matching'. European Conf. on Computer Vision, Amsterdam, The Netherlands, 2016, pp. 100–111

[32] Leal-Taixe, L., Canton-Ferrer, C., Schindler, K.: 'Learning by tracking: siamese CNN for robust target association'. Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 2016, pp. 33–40

[33] Wojke, N., Bewley, A., Paulus, D.: 'Simple online and realtime tracking with a deep association metric'. Proc. Int. Conf. on Image Processing, Beijing, China, 2017, pp. 3645–3649

[34] Milan, A., Rezatofighi, S.H., Dick, A.R., *et al.*: 'Online multi-target tracking using recurrent neural networks'. Proc. AAAI, San Francisco, CA, USA, 2017, vol. 2, pp. 4225–4232

[35] Chen, L., Ai, H., Shang, C., *et al.*: 'Online multi-object tracking with convolutional neural networks'. Proc. Int. Conf. on Image Processing, Beijing, China, 2017, pp. 645–649

[36] Chu, Q., Ouyang, W., Li, H., *et al.*: 'Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism'. Proc. IEEE Int. Conf. Computer Vision, Venice, Italy, 2017, pp. 4846–4855

[37] Son, J., Baek, M., Cho, M., *et al.*: 'Multi-object tracking with quadruplet convolutional neural networks'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Honolulu, HI, USA, 2017, pp. 5620–5629

[38] Schulter, S., Vernaza, P., Choi, W., *et al.*: 'Deep network flow for multi-object tracking'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Honolulu, HI, USA, 2017, pp. 2730–2739

[39] Sadeghian, A., Alahi, A., Savarese, S.: 'Tracking the untrackable: learning to track multiple cues with long-term dependencies'. Proc. IEEE Int. Conf. Computer Vision, Venice, Italy, 2017, pp. 300–311

[40] Fang, K., Xiang, Y., Li, X., *et al.*: 'Recurrent autoregressive networks for online multi-object tracking'. Proc. IEEE Winter Conf. on Applications of Computer Vision, Lake Tahoe, NV, USA, 2018, pp. 466–475

[41] Ma, C., Yang, C., Yang, F., *et al.*: 'Trajectory factory: tracklet cleaving and re-connection by deep siamese Bi-GRU for multiple object tracking', arXiv preprint arXiv:1804.04555, 2018

[42] Kim, C., Li, F., Ciptadi, A., *et al.*: 'Multiple hypothesis tracking revisited'. Proc. IEEE Int. Conf. Computer Vision, Santiago, Chile, 2015, pp. 4696–4704

[43] Tang, S., Andriluka, M., Andres, B., *et al.*: 'Multiple people tracking by lifted multicut and person reidentification'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Honolulu, HI, USA, 2017, pp. 3539–3548

[44] Xiang, J., Zhang, G., Hou, J., *et al.*: 'Multiple target tracking by learning feature representation and distance metric jointly', arXiv preprint arXiv:1802.03252, 2018

[45] Leal-Taixe, L., Milan, A., Schindler, K., *et al.*: 'Tracking the trackers: an analysis of the state of the art in multiple object tracking', arXiv preprint arXiv:1704.02781, 2017

[46] Zhang, L., Li, Y., Nevatia, R.: 'Global data association for multi-object tracking using network flows'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Anchorage, AK, USA, 2008, pp. 1–8

[47] Pirsiavash, H., Ramanan, D., Fowlkes, C.C.: 'Globally-optimal greedy algorithms for tracking a variable number of objects'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 2011, pp. 1201–1208

[48] Chari, V., Lacoste-Julien, S., Laptev, I., *et al.*: 'On pairwise costs for network flow multi-object tracking'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Boston, MA, USA, 2015, pp. 5537–5545

[49] Dehghan, A., Tian, Y., Torr, P.H., *et al.*: 'Target identity-aware network flow for online multiple target tracking'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Boston, MA, USA, 2015, pp. 1146–1154

[50] Wang, X., Turetken, E., Fleuret, F., *et al.*: 'Tracking interacting objects using intertwined flows', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016, **38**, pp. 2312–2326

[51] Butt, A.A., Collins, R.T.: 'Multi-target tracking by Lagrangian relaxation to min-cost network flow'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Portland, OR, USA, 2013, pp. 1846–1853

[52] Berclaz, J., Fleuret, F., Turetken, E., *et al.*: 'Multiple object tracking using k-shortest paths optimization', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011, **33**, (9), pp. 1806–1819

[53] Tang, S., Andres, B., Andriluka, M., *et al.*: 'Subgraph decomposition for multi-target tracking'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Boston, MA, USA, 2015, pp. 5033–5041

[54] Dehghan, A., Modiri-Assari, S., Shah, M.: 'GMMCP tracker: globally optimal generalized maximum multi clique problem for multiple object tracking'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Boston, MA, USA, 2015, pp. 4091–4099

[55] Wu, M., Peng, X., Zhang, Q., *et al.*: 'Motion constraint Markov network model for multi-target tracking', *Optoelectron. Lett.*, 2008, **4**, (5), pp. 375–378

[56] Yang, B., Huang, C., Nevatia, R.: 'Learning affinities and dependencies for multi-target tracking using a CRF model'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 2011, pp. 1233–1240

[57] Yang, B., Nevatia, R.: 'An online learned CRF model for multi-target tracking'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Providence, RI, USA, 2012, pp. 2034–2041

[58] Milan, A., Schindler, K., Roth, S.: 'Detection-and trajectory-level exclusion in multiple object tracking'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Portland, OR, USA, 2013, pp. 3682–3689

[59] Milan, A., Roth, S., Schindler, K.: 'Continuous energy minimization for multitarget tracking', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014, **36**, (1), pp. 58–72

[60] Milan, A., Schindler, K., Roth, S.: 'Multi-target tracking by discrete-continuous energy minimization', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016, **38**, (10), pp. 2054–2068

[61] Huang, C., Wu, B., Nevatia, R.: 'Robust object tracking by hierarchical association of detection responses'. European Conf. on Computer Vision, Marseille, France, 2008, pp. 788–801

[62] Kuo, C.H., Huang, C., Nevatia, R.: 'Multi-target tracking by on-line learned discriminative appearance models'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010, pp. 685–692

[63] Kuo, C.H., Nevatia, R.: 'How does person identity recognition help multi-person tracking?'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 2011, pp. 1217–1224

[64] Bar-Shalom, Y.: '*Tracking and data association*' (Academic Press, Boston, 1988)

[65] Zhou, X., Li, Y., He, B., *et al.*: 'GM-PHD-based multi-target visual tracking using entropy distribution and game theory', *IEEE Trans. Ind. Inf.*, 2014, **10**, (2), pp. 1064–1076

[66] Zhou, X., Yu, H., Liu, H., *et al.*: 'Tracking multiple video targets with an improved GM-PHD tracker', *Sensors*, 2015, **15**, (12), pp. 30240–30260

[67] Khan, Z., Balch, T., Dellaert, F.: 'MCMC-based particle filtering for tracking a variable number of interacting targets', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2005, **27**, (11), pp. 1805–1819

[68] Breitenstein, M.D., Reichlin, F., Leibe, B., *et al.*: 'Online multiperson tracking-by-detection from a single, uncalibrated camera', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011, **33**, (9), pp. 1820–1833

[69] Bae, S.H., Yoon, K.J.: 'Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 1218–1225

[70] Leal-Taixe, L., Fenzi, M., Kuznetsova, A., *et al.*: 'Learning an image-based motion context for multiple people tracking'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 3542–3549

[71] Xiang, Y., Alahi, A., Savarese, S.: 'Learning to track: online multi-object tracking by decision making'. Proc. IEEE Int. Conf. Computer Vision, Santiago, Chile, 2015, pp. 4705–4713

[72] Reid, D.: 'An algorithm for tracking multiple targets', *IEEE Trans. Autom. Control*, 1979, **24**, (6), pp. 843–854

[73] Choi, W.: 'Near-online multi-target tracking with aggregated local flow descriptor'. Proc. IEEE Int. Conf. Computer Vision, Santiago, Chile, 2015, pp. 3029–3037

[74] Xu, Y., Qin, L., Huang, Q.: 'Coupling reranking and structured output SVM co-train for multitarget tracking', *IEEE Trans. Circuits Syst. Video Technol.*, 2016, **26**, (6), pp. 1084–1098

[75] Zhan, B., Monekosso, D.N., Remagnino, P., *et al.*: 'Crowd analysis: a survey', *Mach. Vis. Appl.*, 2008, **19**, (5–6), pp. 345–357

[76] Moeslund, T.B., Granum, E.: 'A survey of computer vision-based human motion capture', *Comput. Vis. Image Underst.*, 2001, **81**, (3), pp. 231–268

[77] Wang, X.: 'Intelligent multi-camera video surveillance: a review', *Pattern Recognit. Lett.*, 2013, **34**, (1), pp. 3–19

[78] Zhu, H., Yuen, K.V., Mihaylova, L., *et al.*: 'Overview of environment perception for intelligent vehicles', *IEEE Trans. Intell. Transp. Syst.*, 2017, **18**, (10), pp. 2584–2601

[79] Yilmaz, A., Javed, O., Shah, M.: 'Object tracking: a survey', *ACM Comput. Surv.*, 2006, **38**, (4), pp. 1–45

[80] Yang, H., Shao, L., Zheng, F., *et al.*: 'Recent advances and trends in visual tracking: a review', *Neurocomputing*, 2011, **74**, (18), pp. 3823–3831

[81] Li, X., Hu, W., Shen, C., *et al.*: 'A survey of appearance models in visual object tracking', *ACM Trans. Intell. Syst. Technol.*, 2013, **4**, (4), p. 58

[82] Wu, Y., Lim, J., Yang, M.H.: 'Online object tracking: a benchmark'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Portland, OR, USA, 2013, pp. 2411–2418

[83] Smeulders, A.W., Chu, D.M., Cucchiara, R., *et al.*: 'Visual tracking: an experimental survey', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014, **36**, (7), pp. 1442–1468

[84] Leal-Taixe, L., Milan, A., Reid, I., *et al.*: 'Motchallenge 2015: towards a benchmark for multi-target tracking', arXiv preprint arXiv:1504.01942, 2015

[85] Milan, A., Leal-Taixe, L., Reid, I., *et al.*: 'MOT16: a benchmark for multi-object tracking', arXiv preprint arXiv:1603.00831, 2016

[86] Bae, S.H., Yoon, K.J.: 'Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018, **40**, (3), pp. 595–610

[87] Kim, C., Li, F., Rehg, J.M.: 'Multi-object tracking with neural gating using bilinear LSTM'. European Conf. on Computer Vision, Munich, Germany, 2018, pp. 200–215

[88] Girshick, R., Donahue, J., Darrell, T., *et al.*: 'Rich feature hierarchies for accurate object detection and semantic segmentation'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 580–587

[89] Wojke, N., Bewley, A.: 'Deep cosine metric learning for person re-identification'. Proc. IEEE Winter Conf. on Applications of Computer Vision, Lake Tahoe, NV, USA, 2018, pp. 748–756

[90] Weinzaepfel, P., Revaud, J., Harchaoui, Z., *et al.*: 'Deepflow: large displacement optical flow with deep matching'. Proc. IEEE Int. Conf. Computer Vision, Sydney, NSW, Australia, 2013, pp. 1385–1392

[91] Held, D., Thrun, S., Savarese, S.: 'Learning to track at 100 FPS with deep regression networks'. European Conf. on Computer Vision, Amsterdam, The Netherlands, 2016, pp. 749–765

[92] Cheng, D., Gong, Y., Zhou, S., *et al.*: 'Person re-identification by multi-channel parts-based CNN with improved triplet loss function'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016, pp. 1335–1344

[93] Larochelle, H., Murray, I.: 'The neural autoregressive distribution estimator'. Proc. Int. Conf. on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 2011, pp. 29–37

[94] Gregor, K., Danihelka, I., Mnih, A., *et al.*: 'Deep autoregressive networks', arXiv preprint arXiv:1310.8499, 2013

[95] Goodfellow, I., Pouget-Abadie, J., Mirza, M., *et al.*: 'Generative adversarial nets'. Proc. Advances in Neural Information Processing Systems, Montreal, Canada, 2014, pp. 2672–2680

[96] Fernando, T., Denman, S., Sridharan, S., *et al.*: 'Task specific visual saliency prediction with memory augmented conditional generative adversarial networks'. Proc. IEEE Winter Conf. on Applications of Computer Vision, Lake Tahoe, NV, USA, 2018, pp. 1539–1548

[97] Fernando, T., Denman, S., Sridharan, S., *et al.*: 'Tracking by prediction: a deep generative model for mutli-person localisation and tracking'. Proc. IEEE Winter Conf. on Applications of Computer Vision, Lake Tahoe, NV, USA, 2018, pp. 1122–1132

[98] Ondruska, P., Posner, I.: 'Deep tracking: seeing beyond seeing using recurrent neural networks'. Proc. AAAI, Phoenix, AZ, USA, 2016, pp. 3361–3367

[99] Li, W., Zhao, R., Xiao, T., *et al.*: 'Deepreid: deep filter pairing neural network for person re-identification'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 152–159

[100] Zheng, L., Bie, Z., Sun, Y., *et al.*: 'Mars: a video benchmark for large-scale person re-identification'. European Conf. Computer Vision, Amsterdam, The Netherlands, 2016, pp. 868–884

[101] Chen, L., Ai, H., Zhuang, Z., *et al.*: 'Real-time multiple people tracking with deeply learned candidate selection and person re-identification'. IEEE Int. Conf. on Multimedia and Expo (ICME), San Diego, CA, USA, 2018, pp. 1–6

[102] Dollar, P., Appel, R., Belongie, S., *et al.*: 'Fast feature pyramids for object detection', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014, **36**, (8), pp. 1532–1545

[103] Felzenszwalb, P.F., Girshick, R.B., McAllester, D., *et al.*: 'Object detection with discriminatively trained part-based models', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010, **32**, (9), pp. 1627–1645

[104] Ferryman, J., Shahrokni, A.: 'PETS2009: dataset and challenge'. IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance (PETS), Miami, FL, USA, 2009, pp. 1–6

[105] Andriluka, M., Roth, S., Schiele, B.: 'People-tracking-by-detection and people-detection-by-tracking'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Anchorage, AK, USA, 2008, pp. 1–8

[106] Geiger, A., Lenz, P., Urtasun, R.: 'Are we ready for autonomous driving? The KITTI vision benchmark suite'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Providence, RI, USA, 2012, pp. 3354–3361

[107] Robicquet, A., Sadeghian, A., Alahi, A., *et al.*: 'Learning social etiquette: human trajectory understanding in crowded scenes'. European Conf. Computer Vision, Amsterdam, The Netherlands, 2016, pp. 549–565

[108] Kasturi, R., Goldgof, D., Soundararajan, P., *et al.*: 'Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics, and protocol', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009, **31**, (2), pp. 319–336

[109] Li, Y., Huang, C., Nevatia, R.: 'Learning to associate: hybrid boosted multi-target tracker for crowded scene'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 2953–2960