Pattern Recognition 76 (2018) 323–338

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Deep visual tracking: Review and experimental comparison

Peixia Li, Dong Wang, Lijun Wang, Huchuan Lu*

School of Information and Communication Engineering, Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, China

ARTICLE INFO

Article history:
Received 3 May 2017
Revised 19 October 2017
Accepted 5 November 2017
Available online 6 November 2017

Keywords:
Visual tracking
Deep learning
CNN
RNN
Pre-training
Online learning

ABSTRACT

Recently, deep learning has achieved great success in visual tracking. The goal of this paper is to review the state-of-the-art tracking methods based on deep learning. First, we introduce the background of deep visual tracking, including the fundamental concepts of visual tracking and related deep learning algorithms. Second, we categorize the existing deep-learning-based trackers into three classes according to network structure, network function and network training. For each categorize, we explain its analysis of the network perspective and analyze papers in different categories. Then, we conduct extensive experiments to compare the representative methods on the popular OTB-100, TC-128 and VOT2015 benchmarks. Based on our observations, we conclude that: (1) The usage of the convolutional neural network (CNN) model could significantly improve the tracking performance. (2) The trackers using the convolutional neural network (CNN) model to distinguish the tracked object from its surrounding background could get more accurate results, while using the CNN model for template matching is usually faster. (3) The trackers with deep features perform much better than those with low-level hand-crafted features. (4) Deep features from different convolutional layers have different characteristics and the effective combination of them usually results in a more robust tracker. (5) The deep visual trackers using end-to-end networks usually perform better than the trackers merely using feature extraction networks. (6) For visual tracking, the most suitable network training method is to per-train networks with video information and online fine-tune them with subsequent observations. Finally, we summarize our manuscript and highlight our insights, and point out the further trends for deep visual tracking.

1. Introduction

Online visual tracking is a very critical issue in computer vision and video processing, which has numerous realistic applications including navigation, surveillance, robotics, traffic control, augmented reality, to name a few. Many efforts have been done in last decades, however, it is still a challenging task to develop a robust and efficient tracker due to difficulties from partial occlusion, illumination variation, background clutter, motion blur, viewpoint change and so on.

Traditional tracking algorithms usually focus on developing robust appearance model from the perspectives of hand-crafted features, online learning algorithms, or both. Some milestones include IVT [1], MIL [2], TLD [3], APGL1 [4], SCM [5], ASLAS [6], STRUCK [7], and KCF [8]. However, the reports on large-scale benchmark evaluations (both OTB-100 [9], TC128 [10] and VOT2015 [11]) suggest that the performance of these traditional algorithms is far from the requirement of realistic applications.

Over the last five years, deep learning [12] have achieved an impressive suite of results thanks to their success on automatic feature extraction via multi-layer nonlinear transformations, especially in computer vision [13,14], speech recognition [15,16] and natural language processing [17,18]. Motivated by these breakthroughs, several deep-learning-based trackers (e.g., FCNT [19], MDNet [20], STCT [21], SINT [22], SiameFC [23], C-COT [24], GOTURN [25], TCNN [26], ADNet [27] and SANet [28]) have demonstrated the potential advantages for significantly improving the tracking performance. The performance on the OTB-100 [9] dataset is constantly refreshed by the tracking methods based on deep learning (such as DeepSRDCF [29], HCFT [30] and HDT [31]). The MDNet [20] tracker is the winner of VOT2015 competition. All the top-4 trackers in the VOT2016 competition, including C-COT [24], TCNN [26], SSAT[1] and MLDF[2], are based on deep neural networks.

Although a variety of tracking methods based on deep learning have been presented, until now, there exists no work to make

* Corresponding author.
E-mail address: lhchuan@dlut.edu.cn (H. Lu).

[1] SSAT is an extended version of MDNet [20].
[2] MLDF is designed based on FCNT [19] and STCT [21].

https://doi.org/10.1016/j.patcog.2017.11.007
0031-3203/

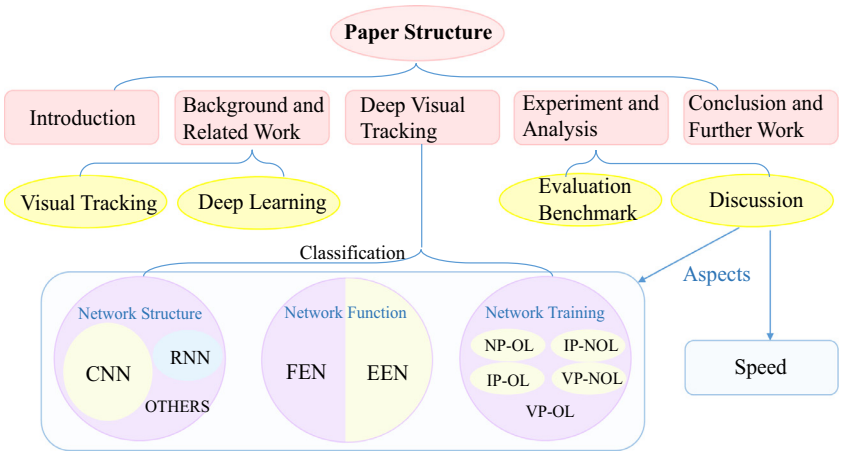

Fig. 1. The structure of this paper.

a detailed survey, comprehensive evaluation and insightful analysis on these deep trackers. In this work, we review existing deep-learning-based tracking algorithms and evaluate them on recent benchmarks. Furthermore, we attempt to address several important issues: (1) What is the connection and difference between these deep-learning-based trackers? We present three classification rules to review existing deep visual trackers according to the structures of network, the functions of network and network training methods; (2) Why is deep learning suitable for visual tracking? In this paper, we conduct a comprehensive evaluation on existing open-source deep-learning-based trackers, and further explain the reason why deep feature is more robust than hand-crafted feature and why interaction between different parts of the network could improve tracking performance; (3) How to better leverage deep networks for visual tracking and what is the future direction of the development on deep visual tracking? By evaluating existing deep trackers and some top-ranked baseline methods, we make some useful conclusions and point out the possible research directions. In order to construct a large-scale and fair comparison, we test 16 deep trackers and 6 baseline methods on the popular OTB-100, TC-128 and VOT2015 benchmarks. The comparison results indicate that: (a) The trackers using the convolutional neural network (CNN) model to distinguish the tracked object from its local background could get more accurate results, while using CNN for template matching is usually faster. (b) The trackers with deep features usually perform much better than those with low-level hand-crafted features. (c) For the CNN model, the trackers using multiple convolutional layers usually perform better than those using a single convolutional layer. (d) The end-to-end networks usually perform better than the feature extraction networks in designing a robust deep tracking method. (e) For visual tracking, the most suitable network training method is to per-train networks with video information and online fine-tune them with subsequent observations.

The rest of the paper is organized as follows. In Section 2, we briefly introduce some fundamental concepts on online visual tracking and some closely related deep learning algorithms. In Section 3, we review the existing trackers based on deep learning from three aspects: network structure, network function, and network training. In Section 4, we report the experiment evaluations of different deep visual trackers and baseline methods, and provide some discussions and analysis on them. Finally, we summarize our paper and point out some further directions in Section 5. Here, we provide Fig. 1 for a clear understand.

2. Background and related work

2.1. Fundamental concepts of online visual tracking

Online visual tracking aims to track any object in realistic scenes. Broadly speaking, a visual tracking method consists of two main components: a motion model that describes the states of an object over time and predicts its likely state (e.g., Kalman filter [32] and particle filter [33,34]); and an observation model that depicts the appearance information of the tracked object and verifies predictions in each frame [35]. Some researches have demonstrated that the observation model plays a more important role than the motion model [36].

From the perspective of the observation model, the existing trackers usually can be categorized into either generative (e.g., [1,37–40]) or discriminative (e.g., [3,7,41–46]) methods. Generative methods focus on searching for the regions that are the most similar to the tracked object, including template-based [32,38,47], subspace-based [1], sparse representation [48,49], to name a few. While discriminative trackers usually consider tracking as a classification problem that distinguishes the tracked objects from its local surrounding backgrounds. Several classic machine learning techniques have been attempted to solve the tracking problem, such as boosting [50,51], support vector machine [52], naive bayes [42], random forest [53], multiple instance learning [2], metric learning [54], structured learning [7], latent variable learning [55], correlation filter [44] and so on. In [36], Wang et al. demonstrate that feature extraction is also a very important issue in designing a robust tracker (such as the tracker with HOG features performs much better than that with Haar-like features). From the research experiences of the traditional tracking algorithms, we can conclude that any progress of feature extraction methods or machine learning techniques may facilitate developing the tracking method. Thus, we believe that the usage of deep learning could improve the tracking performance since the deep learning techniques have been shown powerful abilities on feature extraction and object classification.

To the best of our knowledge, there exist some surveys and evaluations on traditional tracking algorithms [8,11,35,56–59], which provide useful insights and conclusions for visual tracking. However, until now, no work has been done to review the deep-learning-based trackers, which are all recently published (after 2013) and shown to achieve state-of-the-art performance on the popular benchmarks. We note that our work is the first attempt to make a comprehensive review and exhaustive evaluation of existing deep visual trackers on large-scale benchmarks, which will

facilitate the readers' understanding the benefits of deep learning for visual tracking.

2.2. Related deep learning algorithms

Deep learning has been intensively studied and demonstrated remarkable success in a wide range of computer vision areas, including image classification, object detection, semantic segmentation, image caption, pose estimation, saliency detection, edge detection, *etc.* This section mainly discusses recent progresses in image classification and object detection, which are highly correlated to visual tracking. Readers are referred to deep learning survey papers for more comprehensive reviews.

Image classification is one of the earliest computer vision tasks where DNNs have shown their strong learning capabilities. In [60], an 8-layer CNN model is trained in an end-to-end manner on the ImageNet classification data set [61] and delivers record-breaking performance. It further indicates that the applications of dropout layers [62], Rectified Linear Units (ReLUs) [63], and data augmentation techniques can effectively facilitate easier network training and significantly reduces overfitting. Following [60], deeper networks with more sophisticated architectures [64,65] have been designed, yielding higher classification accuracy. To alleviate the problem of exploding/vanishing gradients, Ioffe et al. [66] propose a Batch Normalization method, which is able to accelerate network training and improve the final performance by reducing covariate shift. Later on, He et al. [67] formulate network layers as learning residual functions with short-cut connections, which makes gradients flow through multiple layers easier and allows the training of extremely deep networks. The above progresses have significantly benefit many vision tasks including visual tracking. In particular, the pre-trained deep features on large scale image classification data sets capture high-level semantic meaning of objects, have strong generalization ability across domains, and are widely explored in other areas.

Object detection is another research topic where DNN-based methods have achieved state-of-the-art performance. Most existing DNN-based detectors [68,69] are implemented in a two-step manner by first generating a set of candidate regions and then classifying them into object categories or background with DNNs. Fast R-CNN [69] improves the efficiency by developing the ROI-pooling layer to extract features from shared convolutional feature maps. In [70], the region proposal network is proposed and trained end-to-end with Fast R-CNN to generate high-quality region proposals. Instead of performing detection with CNN-based classifiers, [71] formulates the prediction of bounding boxes and class probabilities as a regression problem with DNNs, which delivers superior performance at a significantly accelerated speed. Some recent methods also focus on addressing object detection at the instance-level [72] or in video sequences [73] with deep-learning-based approaches. It should be noted that object detection and visual tracking are essentially different in that object detection aims to distinguish objects of different categories, while visual tracking is designed to locate objects of interests in an class-agnostic manner. Nonetheless, they are also highly correlated. For instance, some recent visual tracking methods [74] pre-train networks on object detection data sets. Others [75,76] leverage object detection results or region proposals to facilitate more accurate online tracking.

3. Deep visual tracking

In this work, we attempt to review and discuss the existing deep-learning-based trackers from three different perspectives: (1) network structure: the kinds of deep neural networks exploited in developing online visual tracker; (2) network function: the role of deep neural networks in a designed tracking system; and (3) network training: the methods and data for training or fine-tuning the deep neural networks to meet the requirements of online visual tracking.

3.1. Network structure

Networks with different structures focus on solving different tasks. The convolutional neural network (CNN) has been demonstrated to be effective for feature extraction and achieved great success on image classification. While the recurrent neural network (RNN) is able to remember previous states and establish temporal connection, which is suitable for sequence modeling. For visual tracking, from the perspective of network structure, we can roughly category the exiting trackers into three classes: CNN-based trackers, RNN-based trackers, and trackers based on other networks (Fig. 2).

3.1.1. CNN

The CNN model [60,64,65,67] is very suitable for developing robust appearance model in the tracking task, due to its powerful ability on feature extraction and image classification. Similar with the traditional trackers, the CNN-based tracking methods can also be either discriminative or generative. The discriminative method aims to conduct a binary classification with the CNN model for effectively distinguishing the tracked object from its surrounding backgrounds. The generative one focuses on learning a robust similarity function to accurately match the object template within a given search region. We note that the former one as CNN-C for highlighting the classification task and the later one as CNN-M for highlighting the matching process.

CNN-C: To develop a CNN-C-based tracker, a convenient way is to replace hand-crafted features with deep features extracted from the CNN model in the traditional tracking framework, such as correlation filter, online SVM and so on. For the VGGNet [64] model, Ma et al. [30] find that the outputs of the last convolutional layer encode the semantic information and such representations are robust to significant appearance variations. However, their spatial resolution is too coarse to precisely localize the tracked objects. In contrast, earlier convolutional layers provide more precise localization but are less invariant to appearance changes. Thus, they construct three correlation-filter-based classifiers with three different convolutional layers (Conv3-4, Conv4-4 and Conv5-4), and combine their corresponding response maps to identify the tracked object. The network structure is shown as Fig. 3. The DeepSRDCF [29] method combines activations from the convolutional layer of a CNN model with the SRDCF [101] framework. Different with image classification, the results of the algorithm suggest that activations from the first layer could provide superior tracking performance compared to the deeper layers and the convolutional features provide improved results compared standard hand-crafted features. Some similar ideas have been presented in [24,27,31,86,97]. In [76], Zhu et al. use the CNN model similar with Faster R-CNN to generate object proposals, and then the proposals are put into an online Sturctured SVM [102] to obtain the object state. In [95], Hong et al. adopt the CNN model to produce discriminative saliency maps, and combine them with online SVM to learn a robust appearance model. In addition, the deep networks can be not only used for extracting visual features but also for conducting classification (such as [19,21,74,99]).

CNN-M: Different from the CNN-C-based trackers, the CNN-M ones use convolutional neural networks to learn effective matching functions. In [22], Tao et al. propose a siamese network model to match the object template and candidates for visual tracking, in which the optimal state can be determined based on the highest matching score. After that, Bertinetto et al. [23] develop a fully

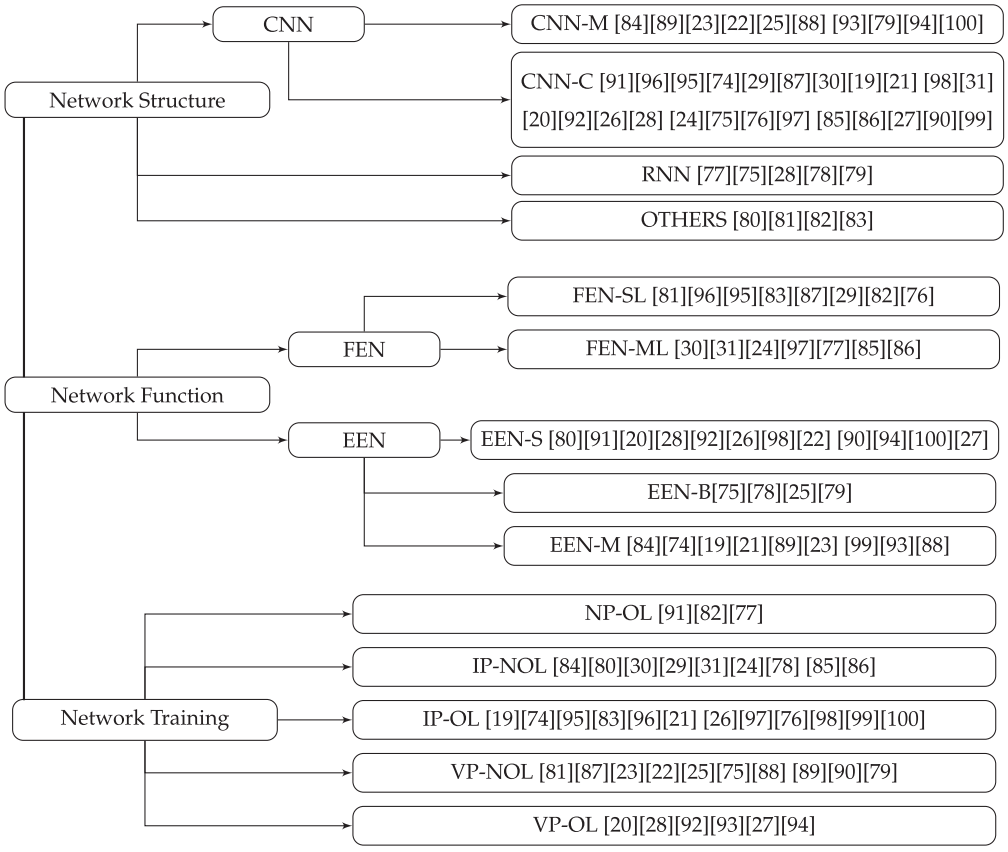

Fig. 2. Structure of three classification methods and algorithms in each category. See Refs. [84,87,90,96].

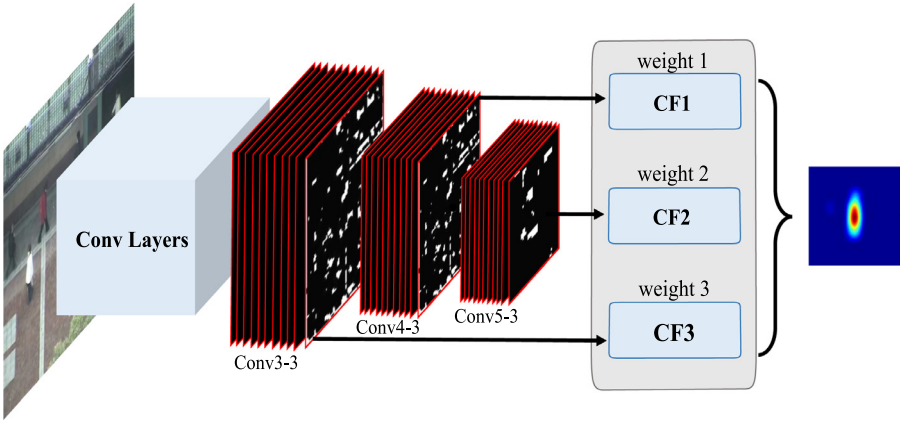

Fig. 3. The structure of HCFT [30].

connected siamese network to match the object template and current search region in a convolutional manner, shown as Fig. 4. Besides, Chen et al. [89] present a more generic approach utilizing a novel two-flow convolutional neural network to include two inputs (one is the object image patch, another is the search region patch) and produce a response map that predicts how likely the object appears in a specific location. [79,94,100] also use two-path network to evaluate the object location. Recently, some trackers apply the satisfactory performance of correlation filter to deep neural network. Valmadre et al. [88] interpret correlation filter as a differentiable layer in a siamese network. A closed-form solution via the fourier domain ensures the network being trained end to end. Similar idea can also be found in [93].

3.1.2. RNN

The recurrent neural network (RNN) model is suitable for sequence modeling since its neuron's output can be directly applied to itself in the next time. Motivated by some research works on handwriting recognition [103,104] or speech recognition [15,16], some attempts have been done to exploit semantic information among spatial configurations and temporal association among frames in visual tracking. Cui et al. [77] propose a RNN-based method to produce a confidence map and apply it to correlation filter. The RNN model is trained from four different directions, which makes the appearance model robust to partial occlusion. Fan et al. [28] adopt a RNN model similar with the RTT [77] method, in which the features produced by RNN are added into a CNN network to get a robust feature representation. The

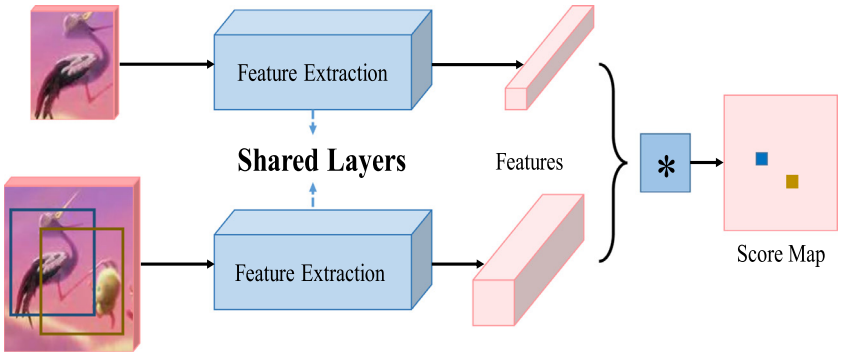

Fig. 4. The structure of SiameFC [23].

above-mentioned two papers mainly focus on establishing spatial relationship among different image parts using the RNN model. Besides, there exist some attempts to construct temporal correlation with RNN. Ning et al. [75] investigate the regression capability of long short-term memory (LSTM) in the temporal domain, and propose to concatenate high-level visual features produced by convolutional networks with region information. Gan et al. [78] feed the feature vector of the input frame into a recurrent neural network. The recurrent neural network effectively updates its internal memory vector based on the previous memory vector, previous location of an object and the current frame. Daniel et al. [79] use a two-path network followed by two LSTM blocks to remember the object appearance and motion.

3.1.3. Others

In addition to the CNN- and RNN-based trackers, some researchers have attempted to develop robust tracking algorithms using other deep networks (especially the autoencoder network [105–107]). Wang et al. [80] propose the first deep-learning-based tracker that trains a stacked denoising autoencoder offline to learn generic image features. These features have been demonstrated to be robust against appearance variations. In [81], the proposed tracker learns complex-valued invariant representations from tracked sequential image patches via strong temporal slowness constraint and stacked convolutional autoencoders. The deep slow local representations are learned offline on unlabeled data and transferred to the observational model of their proposed tracker. Motivated by [80], Zhuang et al. [92] exploit the autoencoder network to construct deep feature representations and utilize the shallow subspace model to recovery partial occlusions or other unexpected noises. This method effectively combines both shallow and deep models and achieves better performance than the traditional DLT algorithm [80].

3.2. Network function

For visual tracking, deep networks can be not only used for extracting effective features but also adopted for evaluating the candidates of the tracked object. From this perspective, the tracking algorithms based on deep learning can be roughly classified into two categories (see Fig. 2): (1) feature extraction network (FEN), which merely uses deep networks to extract deep features, and then adopts the traditional method to learn the appearance model and locate the target; (2) end to end network (EEN), which not only uses deep networks for feature extraction but also for candidate evaluation. The outputs of the EEN methods can be in terms of probability map, heat map, candidate's score, object position or even bounding box directly.

3.2.1. Feature extraction network (FEN)

Motivated by the success of deep features on image classification, many researchers attempt to exploit deep networks for feature extraction in designing a tracking method. This kind of algorithms can be mainly divided into two aspects: FEN-SL (using deep features from a single layer); and FEN-ML (using deep features from multiple layers). *FEN-SL:* The DLT [80], Trans-DLT [83] and CNN-SVM [95] methods design deep networks for feature extraction and adopt the outputs of the networks as object features. The CNT [82] method propagates an input image forward in a CNN network to extract weak features, and then learns a classifier for distinguishing these features into positive and negative ones. The DeepSRDCF [29] method extracts features from the first layers of the VGG network, and combines deep features with the SRDCF framework to improve the tracking performance. The RPNT [76] method utilizes a network similar with Faster R-CNN to draw proposals and extracts the image features using the outputs of last convolutional layers of Faster R-CNN. The RTT [77] method applies RNN to get a saliency map of the search region, which helps the correlation filter to reduce the interference of background.

FEN-ML: Different layers of a deep network could provide multi-level feature description. Thus, it is reasonable to exploit multiple layers for feature extraction in developing a robust tracker. Ma et al. [30] observe that different convolutional layers of a typical CNN model provide multiple levels of abstraction in the feature hierarchies. Features in earlier layers retain higher spatial resolution for precise localization with low-level visual information. While features in latter layers capture more semantic information and less fine-grained spatial details. Thus, they extract features from three different layers and use a fix weight to combine the feature maps generated by those layers, shown as Fig. 3. The results have demonstrated that the features extracted from multiple layers perform better than the features from a single layer for developing a robust tracking method. After that, Ma et al. [97] use a designed network to replace the Conv3-3 layer in [30] and therefore improve the tracking performance. Qi et al. [31] extract features from six convolutional layers and combine these layers using an adaptive weight scheme. The RTT [77] method uses a combination of four output maps of the deep network trained from four different directions. In addition, the C-COT [24] algorithm proposes a joint learning framework to fuse deep features from different spatial pyramids. Based on C-COT [24], Martin et al. [86] aim to combine hand-crafted features with deep features and reduce the algorithm redundancy.

3.2.2. End to end network (EEN)

In contrast with the FEN-based methods, the EEN-based trackers train a network to conduct both feature extraction and candidate evaluation. According to the differences of network outputs, we can roughly divide the EEN-based methods into three cate-

Table 1
Abbreviations in network training.

Abbreviation	Full Name
NP	No Pre-trained
IP	Image Pre-trained
VP	Video Pre-trained
OL	Online Learning
NOL	No Online Learning

gories: EEN-S, EEN-M and EEN-B. Their outputs are object score (S), confidence map (M) and bounding box (B), respectively.

EEN-S: This kind of methods ([20,22,26,28,75,80,91]) generates a series of candidates using particle filter or sliding window schemes, and then produces the scores of these candidates for localizing the tracked object. The DeepTrack [91] method samples a set of candidates with a sliding window, and evaluates the likelihoods of these candidates using a CNN model designed by multiple convolutional layers and a joint learned fully-connected layer. In [22], the SINT method generates a lot of particles and calculates their similarity scores using the siamese network. The optimal state can be determined by the particle with the highest score. In addition to outputting score, the ADNet [27] uses a network to output an action by reinforcement learning, which will guide the bounding box to object.

EEN-M: This kind of trackers usually exploits deep networks to generate a confidence map (or probability map, response map, heat map) and then uses other methods to localize the tracked object. In [19], two deep networks are designed with the Conv4-3 and Conv5-3 layers of the VGG-16 model and then used to calculate the response maps. In [21], the last convolutional layers are jointly combined to produce the final heat map. The SO-DLT [74] method first pre-trains a CNN model to recognize what is an object and then generates a probability map instead of producing a simple class label. The SiameFC [23] tracker utilizes a pre-trained fully convolutional siamese network, the inputs of which are the object template and current search region (Fig. 4). In each frame, this method generates a response map regarding the tracked object with convolution operation. The similar structure is also adopted in the YCNN [89] method.

EEN-B: This kind of algorithms learns an end-to-end network to directly produce the bounding box (or position) of the tracked object in each frame (such as [25,75,78,79]).

3.3. Network training

The network training is also a critical issue for developing a robust deep-learning-based tracker, which may be used to transfer visual prior, online learning, or both. According to the manners of pre-training and online learning, we can divide the existing deep visual trackers into five categories: NP-OL, IP-NOL, IP-OL, VP-NOL and VP-OL (shown in Fig. 2). It should be noted that the 'OL' or 'NOL' here is only for the network rather than the whole tracker. The detailed explanations are presented in Table 1.

NP-OL: For exploiting deep learning to solve the tracking problem, an intuitive idea is to replace traditional appearance model with some deep networks (such as CNN). Li et al. [91] present a deep-learning-based tracker with a CNNs pool, where each CNN is set up to obtain the scores of different particles in every frame. The CNT [82] method collects a series of positive and negative patches to learn many image filters, and builds a robust appearance model using convolutional operators. These trackers can merely use a small number of layers to design the appearance model due to the limit of training data.

IP-NOL: The lack of labeled data in online learning limits the performance of deep networks. Thus, it is reasonable to transfer the visual prior using existing networks pre-trained by massive natural images. The well-known VGGNet [64] (offline trained on the large-scale ImageNet dataset) is widely used to design the tracking methods because of their satisfactory performance on image classification. In [29], Danelljan et al. improve the SRDCF tracker by adopting the outputs of the VGGNet's first convolutional layer as object features. To further exploit the abundant information within the VGGNet model, many research works [24,30,31] combine multi-level features from different convolutional layers to develop effective appearance models.

IP-OL: There also exist many research works [74,75,83,89,95,97,98] to simultaneously exploit the advantage of both offline pre-training and online learning. Some of them adopt the existing offline pre-trained networks and introduce additional structures online trained with subsequent data during the tracking process. The network in the FCNT [19] method is comprised of VGG-16 convolutional layers and an additional network with three designed layers. The parameters of former layers are pre-trained with the ImageNet dataset and fixed during the tracking process. The later layers are flexible enough to capture the appearance change with online update (the similar idea is presented in [21]). In [26], Nam et al. effectively combine pre-trained convolutional layers and multiple fully-connected layers with a tree structure to achieve good tracking performance. The other algorithms offline train a CNN structure and online fine-tune it to meet the requirement of the specific tracking task, such as SO-DLT [74], DST [81], CNN-Tracker [98], Trans-DLT [83], DNT [99] and DRT [100].

VP-NOL: In essence, visual tracking is a sequential inference problem regarding video data. Thus, pre-training deep models with video information may facilitate dealing with the tracking problem. There have been some attempts to train deep networks with tracking videos. The SINT [22] method exploits a siamense network offline trained by tracking videos to learn a matching function, and integrates this matching function with the particle filter method to construct the entire tracking framework. After that, Bertinetto et al. [23] use tracking videos to train a fully convolutional siamese network, which localizes the accurate positions of the tracked object under the convolution framework. Similarly, Held et al. [25] present a image-comparison tracking framework based on two-frame architectures with traditional convolutional layers and fully connected layers trained in an offline manner. The traditional convolutional layers aim to extract robust deep feature representations, and the fully connected layers focus on learning a complex feature comparison between the object template and the candidate samples. This method achieves very fast performance in tracking generic objects.

VP-OL: Besides the VP-NOL-based trackers, there exist some attempts to combine video pre-training and online learning for developing an effective tracking method. By using a large set of tracking videos with ground truths, the MDNet [20] method pre-trains a CNN model with shared layers and multiple branches of domain-specific layers to obtain a generic object representation, shown as Fig. 5. Domain-specific layers correspond to individual training sequences, where each branch is responsible for conducting a binary classification to identify the tracked object in each domain. Then, each domain is trained in the network iteratively to obtain a generic object representation in the shared layers. During the tracking process, the MDNet method constructs a new network to combine the pre-trained shared layers and online updates this network to capture the appearance change of the tracked object. The SANet [28] method exploits the same idea with the MDNet tracker, and introduces an additional RNN-based structure to further enhance object representation.

Recently, deep reinforcement learning have drawn more attention in visual tracking, which is suitable when we are lack of train-

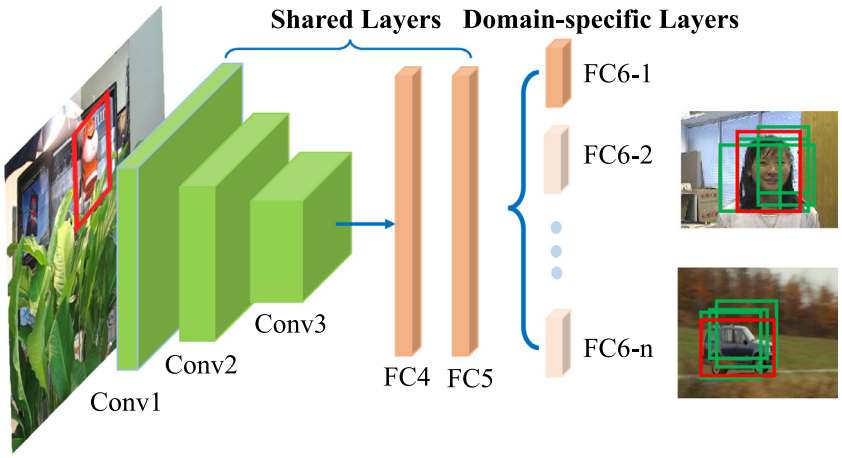

Fig. 5. The structure of MDNet [20].

ing labels or have delayed labels. Yun et al. [27] attempt to learn an agent which could evaluate the moving direction of bounding box through unsupervised learning in a frame. The part of whole model is pre-trained offline with videos. Then, the network is updated online. Different from [27], Janghoon et al. [94] propose an agent for template choosing. The author applies policy-based method to train the model, making it be able to choose the best one.

4. Experiment and analysis

In this section, we conduct a comprehensive experimental evaluation on the recent OTB-100 [9], TC-128 [10] and VOT2015 [11] datasets, which explains the benefits of exploiting deep learning in visual tracking. In the following subsections, we first introduce the adopted benchmarks and the tracking methods to be compared, and then report the experimental comparisons on OTB-100 [9], TC-128 [10] and VOT2015 [11]. Finally, we conduct the detailed discussions on our experiments, and provide the useful insights and conclusions. It should be mentioned that all trackers are tested on PC with a 3.4GHz CPU and a GTX 1080 GPU with 8G memory. All compared trackers are implemented with matlab for fair speed comparison. The project of this paper can be found in http://ice.dlut.edu.cn/lu/publications.html.

4.1. Evaluation benchmark

This subsection introduces the adopted dataset (OTB-100 [9], TC-128 [10], VOT2015 [11]) and deep-learning-based trackers to be evaluated in our manuscript. In this work, we collect around 16 trackers based on deep learning and 6 trackers as baseline to perform a comprehensive experimental comparison on these two popular benchmarks.

OTB-100: The OTB-100 [9] dataset is presented by Wu et al., which has been one of most commonly used benchmarks in evaluating online visual trackers. This dataset includes 100 challenging video clips annotated with different attributes, such as Illumination Variation (IV), Scale Variation (SV), Occlusion (OCC), Deformation (DEF), Motion Blur (MB), Fast Motion (FM), In-Plane Rotation (IPR), Out-of-Plane Rotation (OPR), Out-of-View (OV), Background Clutters (BC), and Low Resolution (LR). By the 11 different attributes, we can analyze the performance of trackers in different aspects. The performance of the 23 trackers is quantitatively validated via two metrics [9] including distance precision (DP)(%) rate at a threshold of 20 pixels and overlap success (OS)(%) at an overlap threshold 0.5. The DP value denotes the percentage that the centre location error (i.e., the Euclidean distance between the center of the tracked target and that of the ground truth) is smaller than a certain threshold in the sequence. The OS value is calculated by the ratio of successfully tracked frames. Usually, the overlap score between the tracking bounding box R_T and the ground truth R_G is larger than a pre-defined threshold (such as 0.5), the target is regarded as being tracked successfully. Fig. 6 illustrates an overall comparison of all 16 deep trackers and 6 traditional top-ranked trackers; while Tables 3 and 4 report the attribute-based performance of these trackers for highlighting the trackers' abilities in handling different challenges.

TC-128: The TC-128 [10] dataset is presented by Liang et al. and focus on color information. This benchmark contains 128 color sequences with ground truth and challenge factor annotations, including Illumination Variation (IV), Scale Variation (SV), Occlusion (OCC), Deformation (DEF), Motion Blur (MB), Fast Motion (FM), In-Plane Rotation (IPR), Out-of-Plane Rotation (OPR), Out-of-View (OV), Background Clutters (BC), and Low Resolution (LR). The 11 challenge factors as well as the evaluation metrics here are the same with the OTB-100 [9] dataset. Fig. 7 illustrates an overall comparison of all 16 deep trackers and 6 traditional top-ranked trackers; while Fig. 8 reports the attribute-based performance of these trackers for highlighting the trackers' abilities in handling different challenges. Besides, we test the speeds of the 22 trackers on this benchmark and report the comparison results in Fig. 11.

VOT2015: The VOT2015 [11] dataset consists of 60 short sequences annotated with 6 different attributes including occlusion, illumination change, motion change, size change, camera motion and unassigned. The major difference between VOT2015 [11] and OTB-100 [9] is that the VOT2015 challenge provides a re-initialization protocol (i.e., trackers are reset with ground-truths in the middle of evaluation if tracking failures are observed). In this paper, we evaluate 15 deep-learning-based trackers and 7 baseline ones in terms of both accuracy and robustness (AR) rank and expected average overlap (EAO) metrics. It should be noted that the 15 trackers here are the same as trackers tested on OTB-100 and TC-128 except for MCPF [85], the code of which is packaged and cannot be changed. The AR rank is created by ranking the trackers over each sequence and averaging the rank lists according to the quantized accuracy and robustness. It does convert the accuracy and robustness to equal scales, thus, the averaged rank cannot be interpreted as a concrete tracking application result. To address this issue, the EAO [11] rule is also introduced to rank different tracking algorithms. It estimates how accurate the estimated bounding box is after a certain number of frames are processed since initialization. Fig. 9 reports both accuracy-robust rank (AR) and expected accuracy overlap (EAO) plots of different trackers and Table 5 provides accuracy scores and failure times of different trackers for each individual attribute on the VOT2015 dataset.

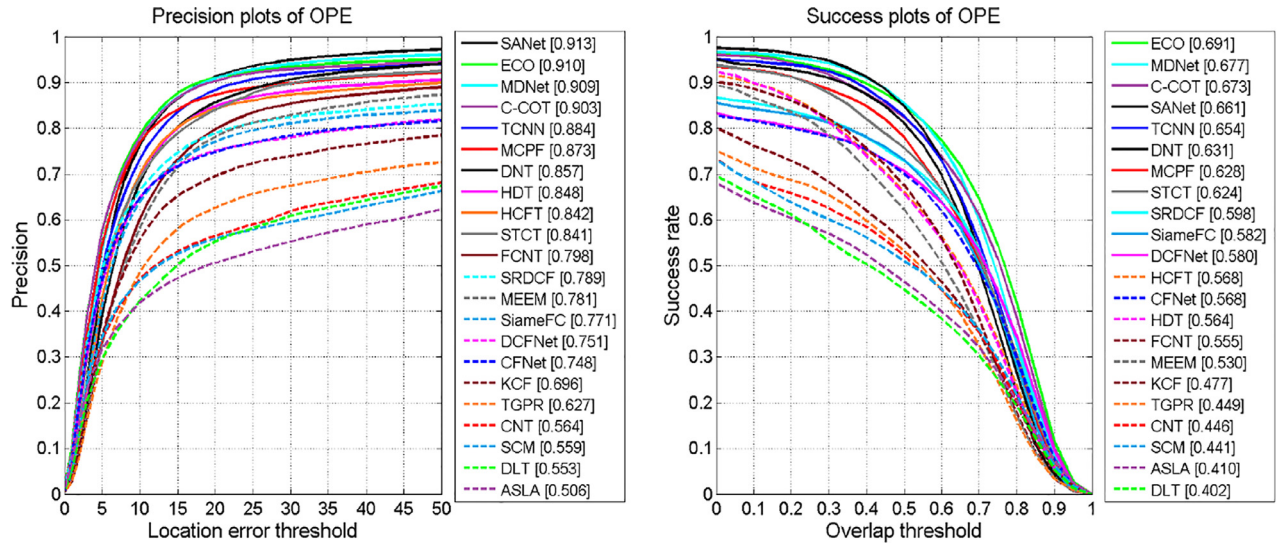

Fig. 6. Distance precision plots (DP) and the area under curve (AUC) over OTB-100 benchmark sequences on 22 tested trackers using one-pass evaluation (OPE).

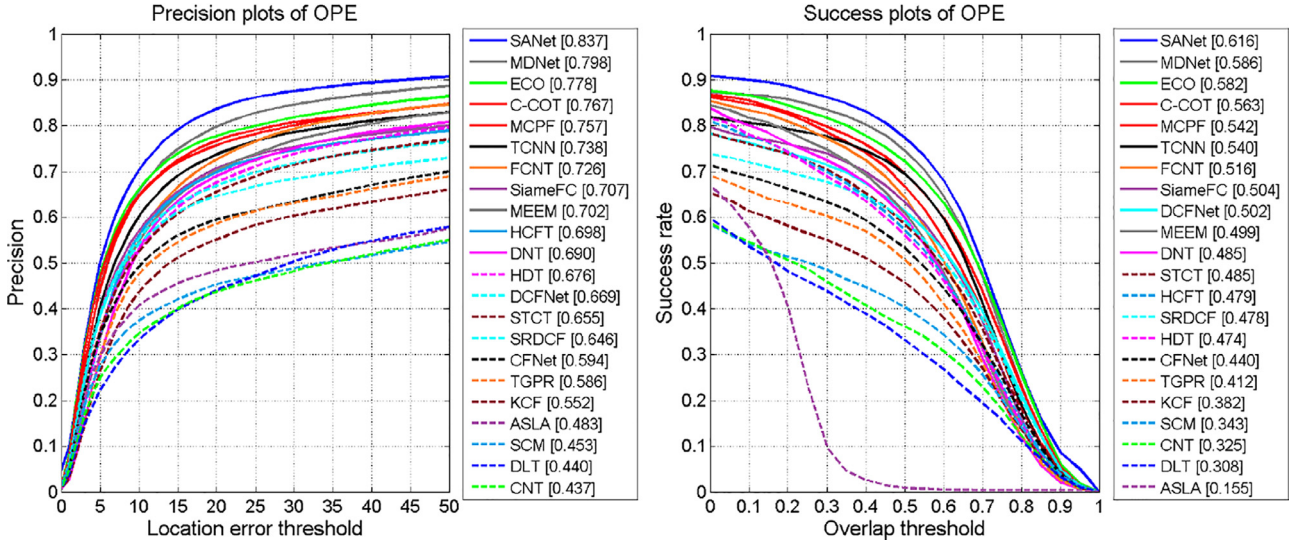

Fig. 7. Distance precision plots (DP) and the area under curve (AUC) over TC-128 benchmark sequences on 22 tested trackers using one-pass evaluation (OPE).

Tracking Algorithm: In our experiment, we collect 16 deep visual trackers, the source codes or benchmark results of which have been publicly available already. These methods include ECO [86], CFNet [88], MCPF [85], DNT [99], DCFNet [93], MDNet [20], SANet [28], TCNN [26], C-COT [24], STCT [21], FCNT [19], HCFT [30], HDT [31], SiameFC [23], CNT [82] and DLT [80] (the detailed information can be found in Table 2).[3] In addition, we select 6 baseline trackers including MEEM [43][4], KCF [44][5], TGPR [55][6], SCM [5][7], ASLA [6][8], and SRDCF [101][9]. These methods have achieved top performance on the above-mentioned three benchmarks within the tracking methods with low-level hand-crafted features. In our experiments, we run the source codes (with same parameters) or use tracking results provided by the original authors to conduct experimental comparisons. Specially, all trackers are re-tested on TC-128 [10] for fair speed comparison.

4.2. Discussions

Analysis of Network Structure: Different network structures have different characteristics. Before exploiting the popular CNN methods, some trackers have attempted to use an autoencoder or a simply designed network for constructing robust appearance models (such as DLT [80] and CNT [82]). However, these trackers just achieved comparable results with top-ranked traditional tracking methods (see Figs. 6 and 9, which have not demonstrated the benefits of deep learning for visual tracking. Most existing deep trackers use deep convolutional neural networks (CNN) to develop appearance models. Some of them use CNN to distinguish the object from the background (i.e., CNN-C), while others use CNN to match candidates with object model (i.e., CNN-M). Distinguishing positive samples with negative ones is easier than matching two feature patches, so CNN-C trackers perform more accurate than CNN-M trackers (see Figs. 6 and 9). While CNN-M trackers are usually

[3] We note that both SiameFC [23], CFNet [88] methods have multiple different implementation versions. Here, SiameFC means SiameFC_3s and CFNet means CFNet_Conv2.

[4] http://cs-people.bu.edu/jmzhang/MEEM/MEEM.html.

[5] http://www.robots.ox.ac.uk/~joao/circulant/index.html.

[6] http://www.dabi.temple.edu/~hbling/code/TGPR.htm.

[7] http://ice.dlut.edu.cn/lu/Project/cvpr12_scm/cvpr12_scm.htm.

[8] http://ice.dlut.edu.cn/lu/Project/cvpr12_jia_project/cvpr12_jia_project.htm.

[9] https://www.cvl.isy.liu.se/research/objrec/visualtracking/regvistrack/.

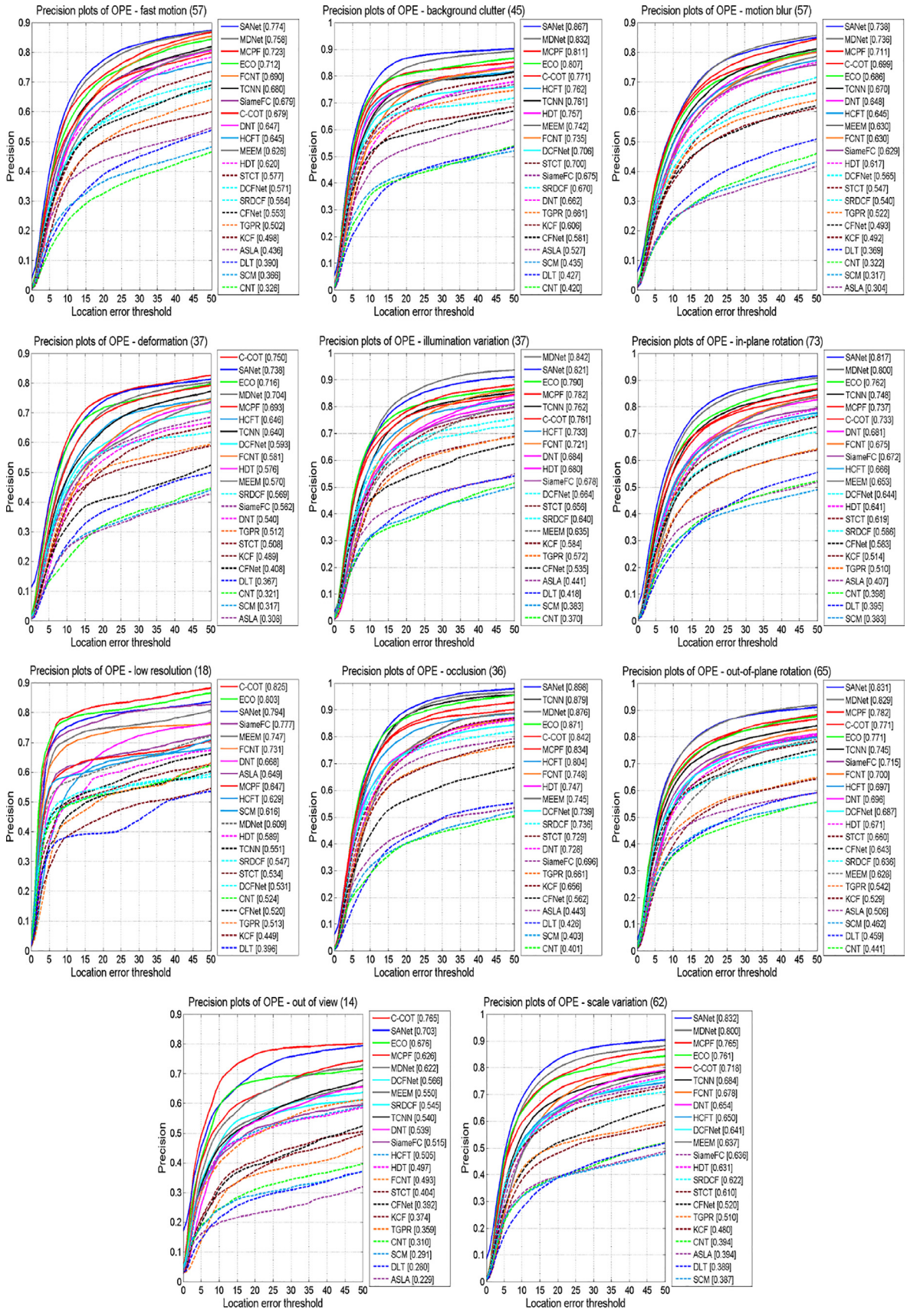

Fig. 8. Distance precision plots (DP) in different challenges over TC-128 benchmark sequences on 22 tested trackers using one-pass evaluation (OPE).

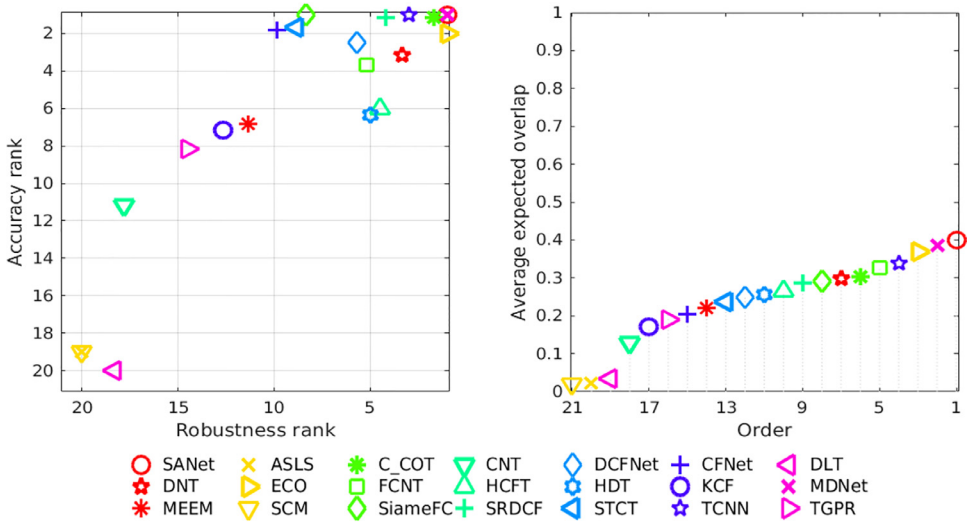

Fig. 9. Accuracy-robust rank (AR) and expected accuracy overlap (EAO) over VOT2015.

Table 2
Detailed information of deep visual trackers evaluated in our paper. Abb - Abbreviation, NS - Network Structure, NF - Network Function, C - Code (M - Matlab, m - Matconvnet, c - Caffe).

Abb	Full Name	NS	NF	NT	C	Resource Link
ECO [86]	ECO: Efficient Convolution Operators for Tracking	CNN-C	FEN-ML	IP-NOL	M+m	http://www.cvl.isy.liu.se/research/objrec/visualtracking/ecotrack/index.html
CFNet [88]	End-to-end representation learning for Correlation Filter based tracking	CNN-M	EEN-M	VP-NOL	M+m	http://www.robots.ox.ac.uk/Üluca/cfnet.html
MCPF [85]	Multi-task Correlation Particle Filter for Robust Visual Tracking	CNN-C	FEN-ML	IP-NOL	M+m	http://nlpr-web.ia.ac.cn/mmc/homepage/tzzhang/mcpf.html
DNT [99]	Dual Deep Network for Visual Tracking	CNN-C	EEN-M	IP-OL	M+c	http://ice.dlut.edu.cn/lu/publications.html
DCFNet [93]	DCFNet: Discriminant Correlation Filters Network for Visual Tracking	CNN-M	EEN-M	VP-OL	M+m	https://github.com/foolwood/DCFNet
MDNet [20]	Learning Multi-Domain Convolutional Neural Networks for Visual Tracking	CNN-C	EEN-S	VP-OL	M+m	https://github.com/HyeonseobNam/MDNet
STCT [21]	STCT: Sequentially Training Convolutional Networks for Visual Tracking	CNN-C	EEN-M	IP-OL	M+c	https://github.com/scott89/STCT
FCNT [19]	Visual Tracking with Fully Convolutional Networks	CNN-C	EEN-M	IP-OL	M+c	http://scott89.github.io/FCNT/
HCFT [30]	Hierarchical Convolutional Features for Visual Tracking	CNN-C	FEN-ML	IP-NOL	M+m	https://sites.google.com/site/jbhuang0604/publications/cf2
HDT [31]	Hedged Deep Tracking	CNN-C	FEN-ML	IP-NOL	M+m	https://sites.google.com/site/yuankiqi/hdt/
SiameFC [23]	Fully-Convolutional Siamese Networks for Object Tracking	CNN-M	EEN-M	VP-NOL	M+m	http://www.robots.ox.ac.uk/~luca/siamese-fc.html
C-COT [24]	Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking	CNN-C	FEN-ML	IP-NOL	M+m	http://www.cvl.isy.liu.se/research/objrec/visualtracking/conttrack/index.html
CNT [82]	Robust Visual Tracking via Convolutional Networks Without Training	OTHERS	FEN-SL	NP-OL	M	http://kaihuazhang.net/
DLT [80]	Learning A Deep Compact Image Representation for Visual Tracking	OTHERS	FEN-SL	IP-NOL	M	http://winsty.net/dlt.html
TCNN [26]	Modeling and Propagating CNNs in a Tree Structure for Visual Tracking	CNN-C	EEN-S	IP-OL	M+m	http://home.unist.ac.kr/professor/bhhan/
SANet [28]	SANet: Structure-Aware Network for Visual Tracking	CNN-C, RNN	EEN-S	VP-OL	M+m	http://www.dabi.temple.edu/~hbling/publication-selected.htm

faster (see Fig. 11), because most trackers use the siamese network to model prior information instead of online fine-tuning. Compared with traditional trackers, the CNN-based deep tracking algorithms perform significantly better than the traditional methods in terms of both overall performance (Figs. 6 and 9) and different challenges (Tables 3, 4 and 5). This is mainly because deep features have stronger representation ability than hand-crafted features. Hand-crafted feature can also be seen as the shallow features of deep network. They include less semantic information compared with deep features. Besides, end-to-end training makes the classifier or matcher fit well with deep features. The usage of the CNN model has achieved satisfactory performance in visual tracking, however, it cannot model sequential information. The RNN model can depict the temporal or spatial relationships among members. In addition, there are also some attempts to use for modeling temporal relationships among continuous frames, such as ROLO [75] and RNNT [78]. But their performance is not satisfactory and requires to be improved. Overall, for visual tracking, the usage of the RNN model is far from being effectively exploited and will be a future research direction.

Analysis of Network Application: Generally, deep networks are used for extracting visual features (i.e., FEN) or outputting probability scores and maps (i.e., EEN) in visual tracking. The first kind of trackers integrates deep features with traditional appearance models (e.g., KCF and SVM).

Based on the experimental comparisons, we can conclude that the usage of deep features could significantly improve the tracking performance in comparison with the trackers with hand-crafted

Table 3
Average precision scores of different trackers for each individual attribute on the OTB-100 dataset.

	IV	SV	OCC	DEF	MB	FM	IPR	OPR	OV	BC	LR	Overall
SANet[28]	0.926	0.891	0.866	0.899	0.858	0.853	0.903	0.906	0.790	0.931	0.882	0.913
ECO[86]	0.914	0.881	0.908	0.859	0.904	0.865	0.892	0.907	0.913	0.942	0.888	0.910
MDNet[20]	0.915	0.893	0.857	0.899	0.879	0.869	0.910	0.900	0.822	0.924	0.854	0.909
C-COT[24]	0.884	0.882	0.904	0.859	0.906	0.870	0.877	0.899	0.895	0.882	0.885	0.903
TCNN[26]	0.920	0.870	0.831	0.848	0.869	0.843	0.895	0.880	0.772	0.878	0.890	0.884
MCPF[85]	0.881	0.864	0.862	0.816	0.841	0.827	0.888	0.867	0.764	0.823	0.918	0.873
DNT[99]	0.871	0.824	0.792	0.855	0.775	0.781	0.855	0.864	0.779	0.829	0.827	0.857
HDT[31]	0.820	0.811	0.774	0.821	0.794	0.802	0.844	0.805	0.663	0.844	0.766	0.848
HCFT[30]	0.832	0.802	0.778	0.792	0.797	0.792	0.865	0.816	0.677	0.843	0.786	0.842
STCT[21]	0.829	0.819	0.785	0.823	0.779	0.770	0.815	0.819	0.694	0.842	0.784	0.841
FCNT[19]	0.777	0.763	0.731	0.760	0.726	0.708	0.830	0.800	0.629	0.750	0.755	0.798
SRDCF[101]	0.792	0.749	0.735	0.734	0.782	0.762	0.745	0.742	0.597	0.775	0.613	0.789
MEEM[43]	0.740	0.740	0.741	0.754	0.722	0.728	0.794	0.794	0.685	0.746	0.605	0.781
SiameFC[23]	0.736	0.739	0.722	0.690	0.724	0.741	0.742	0.756	0.669	0.690	0.815	0.771
DCFNet[93]	0.722	0.743	0.755	0.671	0.686	0.673	0.738	0.755	0.727	0.741	0.767	0.751
CFNet[88]	0.694	0.715	0.674	0.643	0.687	0.695	0.767	0.734	0.533	0.724	0.787	0.748
KCF[44]	0.719	0.639	0.630	0.617	0.618	0.620	0.701	0.677	0.501	0.713	0.546	0.696
TGPR[55]	0.615	0.569	0.581	0.584	0.522	0.503	0.653	0.633	0.402	0.596	0.539	0.627
CNT[82]	0.558	0.514	0.515	0.502	0.368	0.380	0.539	0.561	0.388	0.607	0.522	0.564
SCM[5]	0.605	0.550	0.541	0.512	0.322	0.320	0.533	0.554	0.409	0.579	0.484	0.559
DLT[80]	0.570	0.540	0.490	0.486	0.454	0.421	0.529	0.551	0.486	0.561	0.661	0.553
ALSA[6]	0.524	0.515	0.434	0.453	0.284	0.307	0.495	0.515	0.384	0.531	0.489	0.506

Table 4
Average success rate scores of different trackers for each individual attribute on the OTB-100 dataset.

	IV	SV	OCC	DEF	MB	FM	IPR	OPR	OV	BC	LR	Overall
SANet[28]	0.677	0.640	0.635	0.630	0.663	0.642	0.639	0.649	0.600	0.669	0.592	0.661
ECO[86]	0.713	0.669	0.680	0.633	0.718	0.678	0.655	0.673	0.660	0.700	0.617	0.691
MDNet[20]	0.689	0.661	0.646	0.649	0.686	0.667	0.655	0.661	0.626	0.676	0.591	0.677
C-COT[24]	0.682	0.658	0.674	0.614	0.716	0.673	0.627	0.652	0.648	0.652	0.619	0.673
TCNN[26]	0.678	0.641	0.621	0.615	0.681	0.648	0.645	0.640	0.583	0.629	0.610	0.654
MCPF[85]	0.628	0.604	0.620	0.570	0.597	0.583	0.620	0.619	0.553	0.601	0.598	0.628
DNT[99]	0.637	0.598	0.597	0.620	0.609	0.600	0.615	0.630	0.591	0.603	0.543	0.631
HDT[31]	0.535	0.489	0.528	0.543	0.563	0.550	0.555	0.533	0.472	0.578	0.420	0.564
HCFT[30]	0.552	0.491	0.537	0.532	0.580	0.558	0.567	0.543	0.484	0.585	0.437	0.568
STCT[21]	0.644	0.594	0.590	0.603	0.625	0.607	0.567	0.582	0.530	0.635	0.527	0.624
FCNT[19]	0.543	0.506	0.515	0.529	0.555	0.558	0.558	0.549	0.470	0.529	0.442	0.555
SRDCF[101]	0.613	0.565	0.559	0.544	0.610	0.595	0.544	0.550	0.460	0.583	0.480	0.598
MEEM[43]	0.517	0.474	0.504	0.489	0.545	0.525	0.529	0.525	0.488	0.519	0.335	0.530
SiameFC[23]	0.568	0.557	0.543	0.506	0.568	0.569	0.557	0.558	0.506	0.523	0.573	0.582
DCFNet[93]	0.581	0.570	0.573	0.497	0.564	0.545	0.557	0.575	0.557	0.569	0.551	0.580
CFNet[88]	0.544	0.539	0.516	0.473	0.567	0.553	0.568	0.542	0.414	0.549	0.590	0.568
KCF[44]	0.479	0.399	0.443	0.436	0.456	0.448	0.469	0.453	0.393	0.498	0.307	0.477
TGPR[55]	0.445	0.386	0.423	0.426	0.418	0.402	0.459	0.451	0.326	0.424	0.338	0.449
CNT[82]	0.459	0.410	0.409	0.389	0.349	0.334	0.409	0.429	0.347	0.482	0.394	0.446
SCM[5]	0.486	0.432	0.420	0.381	0.320	0.311	0.412	0.421	0.324	0.457	0.347	0.441
DLT[80]	0.426	0.401	0.336	0.319	0.389	0.353	0.381	0.396	0.350	0.383	0.464	0.402
ALSA[6]	0.433	0.408	0.367	0.362	0.282	0.291	0.393	0.411	0.321	0.432	0.341	0.410

features, especially comparing DeepSRDCF with SRDCF, HCFT with KCF (see Fig. 10), and so on. Similar with human brain, convolution layers in CNN use parameters sharing and local connectivity, which makes them more useful for image feature extraction. Some deep trackers use convolutional features from a single layer (i.e., FEN-SL) while others use a combination of both lower and deeper convolutional layers (i.e., FEN-ML). Based on our empirical observations, trackers using features from multiple layers (e.g., HCFT [30] and HDT [31]) usually perform better than those using features from a single layer (such as DLT [80]). Besides, we can also conclude the same result from self-comparison of HCFT [30] (see Fig. 10). The second kind of deep trackers exploit deep networks to produce probability score (i.e., EEN-S), map (i.e., EEN-M) or even bounding box (i.e., EEN-B), according to which the tracked object can be easily located in the search region. Most EEN-S trackers (e.g., MDNet [20], SANet [28] and TCNN [26]) are designed based on the particle filter framework. They have achieved amazing performance as shown in Table 3, however, it is time-consuming to pass every par-

Table 5
Accuracy scores and failure times of different trackers for each individual attribute on the VOT2015 dataset. ('A'- accuracy score, 'R'- failure times, 'CM'- camera_motion, 'US'- unsigned, 'ILL'- illum_change, 'MC'- motion_change, 'OCC'- occlusion, 'SC'- size_change).

Name	CM		US		ILL		MC		OCC		SC		Pooled	
	A	R	A	R	A	R	A	R	A	R	A	R	A	R
SANet	0.61	18.13	0.66	5.20	0.68	1.07	0.56	15.73	0.54	13.93	0.56	11.20	0.61	43.07
ECO	0.56	15.00	0.59	5.00	0.64	1.00	0.49	17.00	0.41	17.00	0.50	10.00	0.54	43.00
MDNet	0.61	20.00	0.65	6.20	0.68	1.07	0.56	15.73	0.54	13.93	0.56	11.20	0.60	45.93
C_COT	0.56	17.00	0.56	10.00	0.66	1.00	0.49	19.00	0.51	18.00	0.51	12.00	0.54	52.00
TCNN	0.59	25.67	0.63	7.13	0.67	2.27	0.56	21.67	0.51	16.20	0.54	15.67	0.59	57.53
DNT	0.56	30.04	0.58	15.87	0.50	2.20	0.50	23.20	0.43	17.53	0.48	14.67	0.54	72.87
HDT	0.51	29.00	0.58	16.00	0.44	3.00	0.48	31.00	0.43	17.00	0.38	17.00	0.51	80.00
HCFT	0.52	24.00	0.56	17.00	0.45	3.00	0.48	30.00	0.43	16.00	0.37	16.00	0.50	73.00
STCT	0.58	44.40	0.61	13.27	0.65	3.87	0.50	26.60	0.44	31.87	0.50	16.80	0.56	94.33
FCNT	0.51	31.33	0.56	10.33	0.49	2.93	0.50	22.47	0.51	18.33	0.42	14.00	0.52	64.20
SRDCF	0.56	32.00	0.62	15.00	0.68	5.00	0.49	25.00	0.48	19.00	0.51	13.00	0.56	71.00
MEEM	0.49	42.00	0.58	24.00	0.48	5.00	0.47	38.00	0.47	26.00	0.36	30.00	0.50	107.00
SiameFC	0.56	32.00	0.60	15.00	0.67	2.00	0.51	30.00	0.47	20.00	0.51	20.00	0.55	84.00
DCFNet	0.55	32.00	0.60	23.00	0.59	4.00	0.49	30.00	0.48	17.00	0.49	15.00	0.55	92.00
CFNet	0.56	60.00	0.63	22.00	0.72	6.00	0.51	48.00	0.43	18.00	0.52	24.00	0.56	126.00
KCF	0.49	57.00	0.54	38.00	0.49	7.00	0.46	54.00	0.47	22.00	0.37	26.00	0.49	146.00
TGPR	0.46	58.87	0.55	29.53	0.45	7.20	0.45	51.07	0.44	26.80	0.36	34.53	0.48	134.60
CNT	0.48	111.67	0.57	46.00	0.59	8.67	0.42	91.88	0.34	34.67	0.41	42.33	0.48	230.88
SCM	0.25	364.33	0.24	331.53	0.26	38.60	0.22	241.60	0.24	105.93	0.21	202.07	0.23	978.20
DLT	0.17	132.00	0.16	48.00	0.17	15.00	0.17	94.00	0.17	51.00	0.17	66.00	0.17	274.00
ASLS	0.26	360.07	0.25	345.53	0.25	41.40	0.23	237.00	0.25	103.20	0.23	204.73	0.24	990.47

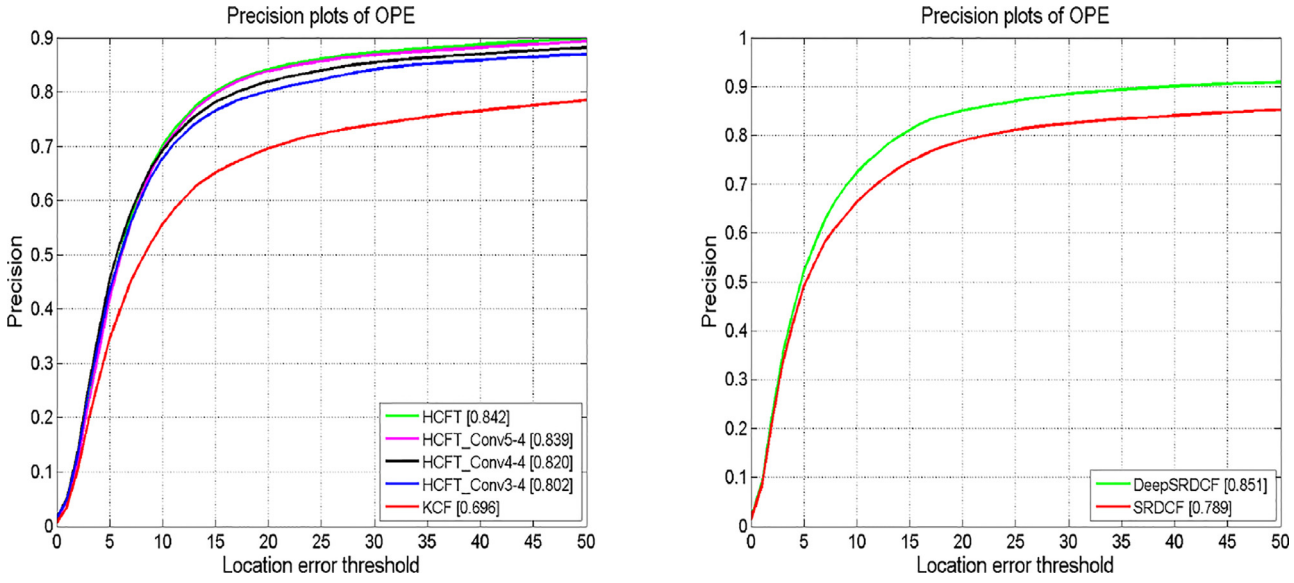

Fig. 10. Distance precision plots (DP) over OTB-100 benchmark sequences on HCFT & KCF (left) and DeepSRDCF & SRDCF (right).

ticle through the whole network. To address this issue, some EEN-M algorithms (such as STCT [21], FCNT [19] and SO-DLT [74]) adopt the search region as the input and the probability map as the output, which are able to exploit the convolution operation to speed up the trackers. In general, the EEN trackers perform better than the FEN ones, which demonstrates that deep networks are more useful than traditional ones to be a classifier or matcher.

Analysis of Network Training: For deep-learning-based trackers, the role of network training is to transferring visual prior via pre-training, capturing the appearance change (during the tracking process) via online learning, or both them. The deep visual tracker with image-based pre-training is able to effectively transfer visual prior within natural images and capture much powerful feature descriptions. These methods are usually integrated with either online updating traditional models (i.e., IP-NOL) or online fine-tuning deep networks (i.e., IP-OL) to depict the visual variations of the tracked object during the tracking processing. It can be seen from our experiment evaluation that these methods (e.g., TCNN, ECO, C-COT, STCT, FCNT, HCFT, HDT) achieve significant improvements in comparison with traditional trackers based on hand-crafted features. In addition, the trackers merely with video-based pre-training (such as SINT and SiameFC) achieve satisfactory

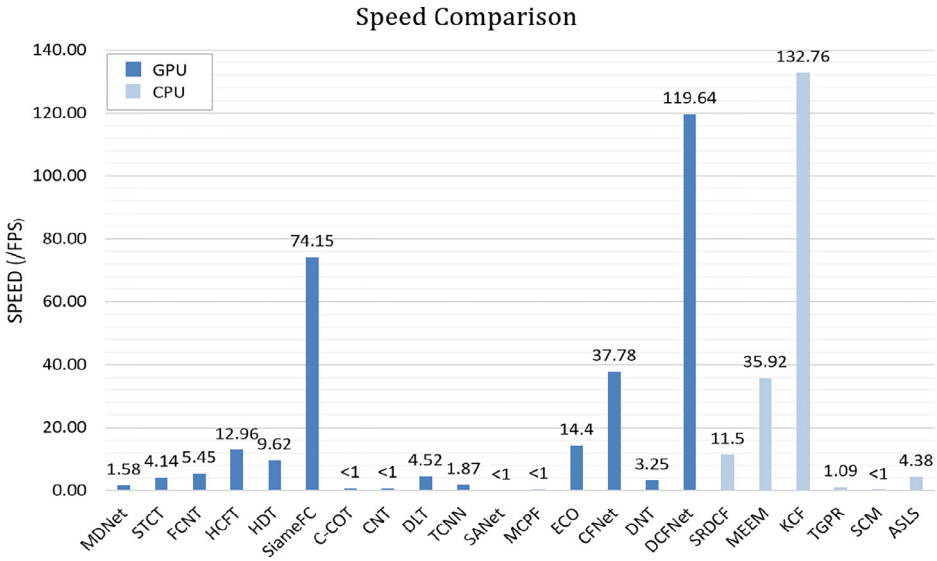

Fig. 11. Speed comparison over TC-128 benchmark sequences.

performance. The deep models of these trackers are pre-trained before tracking and are not updated during the tracking process. Their most important advantage is that it does not require additional computation and storage costs. Thus, further improving the performance of VP-NOL methods will be a very interesting research direction. Finally, we can observe that the VP-OL methods (MDNet and SANet) perform much better than the other ones with simple classifier, since their adopted deep networks not only include abundant visual semantic information within natural images and tracking videos but also capture the appearance change of the object of interest being tracked. Besides, online fine-tuning could make the tracker further capture the appearance variations of the object during the tracking process, thereby improving the tracking performance. We can conclude that it will be a better way for develop a robust deep tracker to exploit both video-based pre-training and online learning. In practice, it also requires to design effective network structures and training methods to balance accuracy and speed.

Analysis of Tracking Speed: Speed is also an important aspect for online visual tracking, which is mainly effected by model complexity and update frequency. Generally, deep feature extraction needs more time than hand-crafted feature extraction. But the truth is not the deeper the network is, the better the features are. Because the resolution of tracking video is lower than that of image classification or object detection, so very deep network will make much lost of information. Most state-of-the-art trackers (MDNet, SANet, ADNet) use VGG-M instead of VGG-16 or VGG-19. It makes tracker faster and more accurate. Besides, the lack of prior information in visual tracking need the tracker conserve template or update model to fit different sequences. But it is often time-consuming to update the deep network frame by frame. Thus, some trackers (FCNT, ECO) update the network every few frames and others (SiameFC, DCFNet, CFNet) apply siamese network in visual tracking. They use one path of network to model template, replacing online update of classifier. We can observe from Fig. 11 that these methods are much faster than others.

5. Conclusion and further work

In this work, we review the recently proposed visual trackers based on deep learning and conduct extensive experiments to evaluate the existing deep-learning-based tracking methods in comparison with some traditional baseline algorithms. The main contributions of this work are three-folds. First, we review the existing deep visual trackers in three aspects including network structure, network function and network training, and then discuss these trackers from each perspective. Second, we conduct extensive experiments to compare the representative methods on the popular OTB-100, TC-128 and VOT2015 benchmarks. This large-scale evaluation facilitates the readers' understanding the benefits of deep learning for visual tracking (especially in comparison with traditional trackers with hand-crafted features). Third, we analyze the results obtained by different deep trackers and obtain the following useful insights and conclusions, which will facilitate the researchers' designing their own trackers based on deep learning. Although deep learning has been used in visual tracking and achieved promising improvements compared with traditional trackers, there also exist many topics to be investigated. First, the deep features have much redundancy which limits both speed and accuracy improvement. It will be a promising direction to reduce the redundancy in deep visual tracking. Second, most trackers use VGG network. Developing more effective network structures should be noticed. Third, the lack of training data need more focus on unsupervised or weakly supervised learning. Reinforcement learning or exploiting generative adversarial networks to generate more training samples will improve the tracking performance. Besides, the transfer ability of model is pretty important in visual tracking. We may see some new directions such as one-shot learning appearing in tracking. In conclusion, improving tracking efficiency and solving the lack of training data will be new directions.

Acknowledgement

This paper is supported by the Natural Science Foundation of China no. 61725202, no. 61502070, and no. 61472060.

References

[1] D.A. Ross, J. Lim, R. Lin, M. Yang, Incremental learning for robust visual tracking, Int. J. Comput. Vis. 77 (1–3) (2008) 125–141.
[2] B. Babenko, M. Yang, S.J. Belongie, Robust object tracking with online multiple instance learning, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1619–1632.
[3] Z. Kalal, K. Mikolajczyk, J. Matas, Tracking-learning-detection, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2012) 1409–1422.
[4] C. Bao, Y. Wu, H. Ling, H. Ji, Real time robust L1 tracker using accelerated proximal gradient approach, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1830–1837.

[5] W. Zhong, H. Lu, M. Yang, Robust object tracking via sparse collaborative appearance model, IEEE Trans. Image Process. 23 (5) (2014) 2356–2368.
[6] X. Jia, H. Lu, M. Yang, Visual tracking via coarse and fine structural local sparse appearance models, IEEE Trans. Image Process. 25 (10) (2016) 4555–4564.
[7] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M. Cheng, S.L. Hicks, P.H.S. Torr, Struck: structured output tracking with kernels, IEEE Trans. Pattern Anal. Mach. Intell. 38 (10) (2016) 2096–2109.
[8] Y. Wu, J. Lim, M. Yang, Online object tracking: a benchmark, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2411–2418.
[9] Y. Wu, J. Lim, M. Yang, Object tracking benchmark, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1834–1848.
[10] P. Liang, E. Blasch, H. Ling, Encoding color information for visual tracking: algorithms and benchmark, IEEE Trans. Image Process. 24 (12) (2015) 5630–5644.
[11] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernández, T. Vojír, G. Häger, G. Nebehay, R.P. Pflugfelder, The visual object tracking VOT2015 challenge results, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2015, pp. 564–586.
[12] Y. LeCun, Y. Bengio, G.E. Hinton, Deep learning, Nature (2015) 436–444.
[13] R.B. Girshick, J. Donahue, T. Darrell, J. Malik, Region-based convolutional networks for accurate object detection and segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 38 (1) (2016) 142–158.
[14] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: a unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.
[15] S. Kim, T. Hori, S. Watanabe, Joint ctc-attention based end-to-end speech recognition using multi-task learning, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 2017, pp. 4835–4839.
[16] Z. Wu, C. Valentini-Botinhao, O. Watts, S. King, Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 2015, pp. 4460–4464.
[17] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, G.E. Hinton, Grammar as a foreign language, in: Proceedings of the Advances in Neural Information Processing Systems, 2015, pp. 2773–2781.
[18] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, Clin. Orthopaedics Related Res. (2014). arXiv:abs/1409.0473.
[19] L. Wang, W. Ouyang, X. Wang, H. Lu, Visual tracking with fully convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3119–3127.
[20] H. Nam, B. Han, Learning multi-domain convolutional neural networks for visual tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4293–4302.
[21] L. Wang, W. Ouyang, X. Wang, H. Lu, STCT: sequentially training convolutional networks for visual tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1373–1381.
[22] R. Tao, E. Gavves, A.W.M. Smeulders, Siamese instance search for tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1420–1429.
[23] L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, P.H.S. Torr, Fully-convolutional siamese networks for object tracking, in: Proceedings of the European Conference on Computer Vision Workshops, 2016, pp. 850–865.
[24] M. Danelljan, A. Robinson, F.S. Khan, M. Felsberg, Beyond correlation filters: learning continuous convolution operators for visual tracking, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 472–488.
[25] D. Held, S. Thrun, S. Savarese, Learning to track at 100 FPS with deep regression networks, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 749–765.
[26] H. Nam, M. Baek, B. Han, Modeling and propagating cnns in a tree structure for visual tracking, Clin. Orthopaedics Related Res. (2016). arXiv:abs/1608.07242.
[27] S. Yun, J. Choi, Y. Yoo, K. Yun, J.Y. Choi, Action-decision networks for visual tracking with deep reinforcement learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2711–2720.
[28] H. Fan, H. Ling, Sanet: structure-aware network for visual tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017.
[29] M. Danelljan, G. Häger, F.S. Khan, M. Felsberg, Learning spatially regularized correlation filters for visual tracking, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4310–4318.
[30] C. Ma, J. Huang, X. Yang, M. Yang, Hierarchical convolutional features for visual tracking, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3074–3082.
[31] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, M. Yang, Hedged deep tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4303–4311.
[32] D. Comaniciu, V.R. Member, P. Meer, Kernel-based object tracking, IEEE Trans. Pattern Anal. Mach. Intell. 25 (5) (2003) 564–575.
[33] P. Pérez, C. Hue, J. Vermaak, M. Gangnet, Color-based probabilistic tracking, in: Proceedings of the European Conference on Computer Vision, 2002, pp. 661–675.
[34] Y. Li, H. Ai, T. Yamashita, S. Lao, M. Kawade, Tracking in low frame rate video: a cascade particle filter with discriminative observers of different life spans, IEEE Trans. Pattern Anal. Mach. Intell. 30 (10) (2008) 1728–1740.
[35] X. Li, W. Hu, C. Shen, Z. Zhang, A.R. Dick, A. van den Hengel, A survey of appearance models in visual object tracking, ACM Trans. Intell. Syst. Technol. 4 (4) (2013) 58:1–58:48.
[36] N. Wang, J. Shi, D. Yeung, J. Jia, Understanding and diagnosing visual tracking systems, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3101–3109.
[37] J. Santner, C. Leistner, A. Saffari, T. Pock, H. Bischof, PROST: parallel robust online simple tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 723–730.
[38] J. Kwon, K.M. Lee, Tracking by sampling and integratingmultiple trackers, IEEE Trans. Pattern Anal. Mach. Intell. 36 (7) (2014) 1428–1441.
[39] D. Wang, H. Lu, C. Bo, Visual tracking via weighted local cosine similarity, IEEE Trans. Cybern. 45 (9) (2015) 1838–1850.
[40] C. Chen, S. Li, H. Qin, A. Hao, Real-time and robust object tracking in video via low-rank coherency analysis in feature space, Pattern Recognit. 48 (9) (2015) 2885–2905.
[41] F. Yang, H. Lu, M. Yang, Robust superpixel tracking, IEEE Trans. Image Process. 23 (4) (2014) 1639–1651.
[42] K. Zhang, L. Zhang, M. Yang, Fast compressive tracking, IEEE Trans. Pattern Anal. Mach. Intell. 36 (10) (2014a) 2002–2015.
[43] J. Zhang, S. Ma, S. Sclaroff, MEEM: robust tracking via multiple experts using entropy minimization, in: Proceedings of the European Conference on Computer Vision, 2014b, pp. 188–203.
[44] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters, IEEE Trans. Pattern Anal. Mach. Intell. 37 (3) (2015) 583–596.
[45] C. Xu, W. Tao, Z. Meng, Z. Feng, Robust visual tracking via online multiple instance learning with fisher information, Pattern Recognit. 48 (12) (2015) 3917–3926.
[46] L. Zhang, P.N. Suganthan, Robust visual tracking via co-trained kernelized correlation filters, Pattern Recognit. 69 (2017) 82–93.
[47] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-based tracking using the integral histogram, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 798–805.
[48] X. Mei, H. Ling, Robust visual tracking and vehicle classification via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (11) (2011) 2259–2272.
[49] D. Wang, H. Lu, M.-H. Yang, Online object tracking with sparse prototypes, IEEE Trans. Image Process. 22 (1) (2013) 314–325.
[50] H. Grabner, H. Bischof, On-line boosting and vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 260–267.
[51] S. Avidan, Ensemble tracking, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2) (2007) 261–271.
[52] S. Avidan, Support vector tracking, IEEE Trans. Pattern Anal. Mach. Intell. 26 (8) (2004) 1064–1072.
[53] A. Saffari, C. Leistner, J. Santner, M. Godec, H. Bischof, On-line random forests, in: Proceedings of the IEEE International Conference on Computer Vision, 2009, pp. 1393–1400.
[54] N. Jiang, W. Liu, Y. Wu, Learning adaptive metric for robust visual tracking, IEEE Trans. Image Process. 20 (8) (2011) 2288–2300.
[55] J. Gao, H. Ling, W. Hu, J. Xing, Transfer learning based visual tracking with gaussian processes regression, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 188–203.
[56] M.S. Arulampalam, S. Maskell, N.J. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking, IEEE Trans. Signal Process. 50 (2) (2002) 174–188.
[57] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Comput. Surv. 38 (4) (2006) 13.
[58] H. Yang, L. Shao, F. Zheng, L. Wang, Z. Song, Recent advances and trends in visual tracking: a review, Neurocomputing 74 (18) (2011) 3823–3831.
[59] A.W.M. Smeulders, D.M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah, Visual tracking: an experimental survey, IEEE Trans. Pattern Anal. Mach. Intell. 36 (7) (2014) 1442–1468.
[60] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 1106–1114.
[61] J. Deng, W. Dong, R. Socher, L. Li, K. Li, F. Li, Imagenet: a large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.
[62] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, Clin. Orthopaedics Related Res. (2012). arXiv:abs/1207.0580.
[63] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the International Conference on Machine Learning, 2010, pp. 807–814.
[64] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Clin. Orthopaedics Related Res. (2014). arXiv:abs/1409.1556.
[65] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S.E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[66] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 448–456.
[67] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[68] R.B. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[69] R.B. Girshick, Fast R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[70] S. Ren, K. He, R.B. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[71] J. Redmon, S.K. Divvala, R.B. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
[72] K. He, G. Gkioxari, P. Dollár, R.B. Girshick, Mask R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2017.
[73] K. Kang, W. Ouyang, H. Li, X. Wang, Object detection from video tubelets with convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 817–825.
[74] N. Wang, S. Li, A. Gupta, D. Yeung, Transferring rich feature hierarchies for robust visual tracking, Clin. Orthopaedics Related Res. (2015). arXiv:abs/1501.04587.
[75] G. Ning, Z. Zhang, C. Huang, Z. He, X. Ren, H. Wang, Spatially supervised recurrent convolutional neural networks for visual object tracking, Clin. Orthopaedics Related Res. (2016). arXiv:abs/1607.05781.
[76] G. Zhu, F. Porikli, H. Li, Robust visual tracking with deep convolutional neural network based object proposals on pets, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 1265–1272.
[77] Z. Cui, S. Xiao, J. Feng, S. Yan, Recurrently target-attending tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1449–1458.
[78] G. Zhu, F. Porikli, H. Li, Robust visual tracking with deep convolutional neural network based object proposals on PETS, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 1265–1272.
[79] D. Gordon, A. Farhadi, D. Fox, Re3 : real-time recurrent regression networks for object tracking, Clin. Orthopaedics Related Res. (2017). arXiv abs/1705.06368
[80] N. Wang, D. Yeung, Learning a deep compact image representation for visual tracking, in: Proceedings of the Advances in Neural Information Processing Systems, 2013, pp. 809–817.
[81] J. Kuen, K. Lim, C. Lee, Self-taught learning of a deep invariant representation for visual tracking via temporal slowness principle, Pattern Recognit. 48 (10) (2015) 2964–2982.
[82] K. Zhang, Q. Liu, Y. Wu, M. Yang, Robust visual tracking via convolutional networks without training, IEEE Trans. Image Process. 25 (4) (2015) 1779–1792.
[83] L. Wang, T. Liu, G. Wang, K.L. Chan, Q. Yang, Video tracking using learned hierarchical features, IEEE Trans. Image Process. 24 (4) (2015) 1424–1435.
[84] J. Fan, W. Xu, Y. Wu, Y. Gong, Human tracking using convolutional neural networks, IEEE Transactions on Neural Networks 21 (10) (2010) 1610–1623.
[85] T. Zhang, C. Xu, M.-H. Yang, Multi-task correlation particle filter for robust object tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4335–4343.
[86] M. Danelljan, G. Bhat, F.S. Khan, M. Felsberg, ECO: efficient convolution operators for tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6638–6646.
[87] C. Ma, X. Yang, C. Zhang, M. Yang, Learning a temporally invariant representation for visual tracking, Proceedings of the IEEE International Conference on Image Processing (2015) 857–861.
[88] J. Valmadre, L. Bertinetto, J.F. Henriques, A. Vedaldi, P.H.S. Torr, End-to-end representation learning for correlation filter based tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2805–2813.
[89] K. Chen, W. Tao, Once for all: a two-flow convolutional neural network for visual tracking, Clin. Orthopaedics Related Res. (2016). arXiv abs/1604.07507
[90] J. Choi, H.J. Chang, S. Yun, T. Fischer, Y. Demiris, J.Y. Choi, Attentional correlation filter network for adaptive visual tracking, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017) 4807–4816.
[91] H. Li, Y. Li, F. Porikli, Deeptrack: learning discriminative feature representations by convolutional neural networks for visual tracking, in: Proceedings of the British Machine Vision Conference, 2014, pp. 1–11.
[92] B. Zhuang, L. Wang, H. Lu, Visual tracking via shallow and deep collaborative model, Neurocomputing 218 (2016) 61–71.
[93] Q. Wang, J. Gao, J. Xing, M. Zhang, W. Hu, Dcfnet: discriminant correlation filters network for visual tracking, Clin. Orthopaedics Related Res. (2017). arXiv abs/1704.04057
[94] J. Choi, J. Kwon, K.M. Lee, Visual tracking by reinforced decision making, Clin. Orthopaedics Related Res. (2017). arXiv abs/1702.06291
[95] S. Hong, T. You, S. Kwak, B. Han, Online tracking by learning discriminative saliency map with convolutional neural network, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 597–606.
[96] M. Hahn, S. Chen, A. Dehghan, Deep tracking: Visual tracking using deep convolutional networks, Clinical Orthopaedics and Related Research (2015). arXiv abs/1512.03993
[97] C. Ma, Y. Xu, B. Ni, X. Yang, When correlation filters meet convolutional neural networks for visual tracking, IEEE Signal Process. Lett. 23 (10) (2016) 1454–1458.
[98] Y. Chen, X. Yang, B. Zhong, S. Pan, D. Chen, H. Zhang, Cnntracker: online discriminative object tracking via deep convolutional neural network, Appl. Soft Comput. 38 (2016) 1088–1098.
[99] Z. Chi, H. Li, H. Lu, M. Yang, Dual deep network for visual tracking, IEEE Trans. Image Process. 26 (4) (2017) 2005–2015.
[100] J. Gao, T. Zhang, X. Yang, C. Xu, Deep relative tracking, IEEE Trans. Image Process. 26 (4) (2017) 1845–1858.
[101] M. Danelljan, G. Häger, F.S. Khan, M. Felsberg, Learning spatially regularized correlation filters for visual tracking, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4310–4318.
[102] M.B. Blaschko, C.H. Lampert, Learning to localize objects with structured output regression, in: Proceedings of the European Conference on Computer Vision, 2008, pp. 2–15.
[103] D.C. Ciresan, U. Meier, L.M. Gambardella, J. Schmidhuber, Convolutional neural network committees for handwritten character classification, in: Proceedings of the International Conference on Document Analysis and Recognition, 2011, pp. 1135–1139.
[104] D.C. Ciresan, U. Meier, Multi-column deep neural networks for offline handwritten chinese character classification, in: Proceedings of the International Joint Conference on Neural Networks, 2015, pp. 1–6.
[105] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408.
[106] R. Socher, J. Pennington, E.H. Huang, A.Y. Ng, C.D. Manning, Semi-supervised recursive autoencoders for predicting sentiment distributions, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2011, pp. 151–161.
[107] J. Zhang, S. Shan, M. Kan, X. Chen, Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 1–16.

Peixia Li received the B.E. degree in electronic information engineering in 2016 and now pursuits the M.Sc. degree in signal and information processing from the Dalian University of Technology, Dalian, China. Her research interests include visual tracking and deep learning.

Dong Wang received the B.E. degree in electronic information engineering and the Ph.D. degree in signal and information processing from the Dalian University of Technology (DUT), Dalian, China, in 2008 and 2013, respectively. He is currently a Associate Professor with the School of Information and Communication Engineering, DUT. His current research interests include face recognition, interactive image segmentation, and object tracking.

Lijun Wang received the B.E. degree from the Dalian University of Technology, Dalian, China, in 2013, where he is currently pursuing the Ph.D. degree in signal and information processing. His current research interests include deep learning, visual saliency, and object tracking.

Huchuan Lu received the M.Sc. degree in signal and information processing and the Ph.D. degree in system engineering from the Dalian University of Technology (DUT), Dalian, China, in 1998 and 2008, respectively. He joined as a Faculty Member with DUT, in 1998, where he is currently a Full Professor with the School of Information and Communication Engineering. His current research interests include computer vision and pattern recognition with a focus on visual tracking, saliency detection, and segmentation. Dr. Lu is an Associate Editor of the IEEE TRANSACTIONS ON CYBERNETICS and a senior member of IEEE.