International Journal of Automation and Computing
www.ijac.net

18(3), June 2021, 311-333
DOI: 10.1007/s11633-020-1274-8

Advances in Deep Learning Methods for Visual Tracking: Literature Review and Fundamentals

Xiao-Qin Zhang Run-Hua Jiang Chen-Xiang Fan Tian-Yu Tong
Tao Wang Peng-Cheng Huang

College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, China

Abstract: Recently, deep learning has achieved great success in visual tracking tasks, particularly in single-object tracking. This paper provides a comprehensive review of state-of-the-art single-object tracking algorithms based on deep learning. First, we introduce basic knowledge of deep visual tracking, including fundamental concepts, existing algorithms, and previous reviews. Second, we briefly review existing deep learning methods by categorizing them into data-invariant and data-adaptive methods based on whether they can dynamically change their model parameters or architectures. Then, we conclude with the general components of deep trackers. In this way, we systematically analyze the novelties of several recently proposed deep trackers. Thereafter, popular datasets such as Object Tracking Benchmark (OTB) and Visual Object Tracking (VOT) are discussed, along with the performances of several deep trackers. Finally, based on observations and experimental results, we discuss three different characteristics of deep trackers, i.e., the relationships between their general components, exploration of more effective tracking frameworks, and interpretability of their motion estimation components.

Keywords: Deep learning, visual tracking, data-invariant, data-adaptive, general components.

Citation: X. Q. Zhang, R. H. Jiang, C. X. Fan, T. Y. Tong, T. Wang, P. C. Huang. Advances in deep learning methods for visual tracking: literature review and fundamentals. *International Journal of Automation and Computing*, vol.18, no.3, pp.311–333, 2021. http://doi.org/10.1007/s11633-020-1274-8

1 Introduction

Single object tracking is a fundamental and critical task in the fields of computer vision and video processing. It has various practical applications in areas such as navigation, robotics, traffic control, and augmented reality. Therefore, numerous efforts have been devoted to overcoming the challenges in the single-object tracking task and developing effective tracking algorithms. However, this task remains challenging because of the difficulty in balancing the effectiveness and efficiency of the tracking algorithms. In addition, existing algorithms are not sufficiently robust under complex scenes with multiple issues, e.g., background clutter, motion blur, viewpoint changes, and illumination variations.

Single object tracking aims at locating a given target in all frames of a video. To this end, tracking algorithms always extract certain features from the template of target appearance and a search frame, and then iteratively match these features to locate the object. For retaining effective target templates, the appearance of the object in the first frame is taken as the initialization and continuously updated during tracking. In contrast, the matching framework is manually designed and fixed during the entire tracking process. As a result, the extracted features are required to be representative to accurately distinguish the object from the background. However, because these extracted features cannot comprehensively reflect the characteristics of an object, conventional tracking algorithms[1–4] tend to have relatively poor performance. Therefore, the improvements of these conventional tracking algorithms are twofold: by exploring features that can better reflect the characteristics of the object and by proposing effective matching frameworks. For example, the template-based[1, 5, 6], subspace-based[7], and sparse-representation[8, 9] methods use certain elements to represent an object, rather than directly using cropped pixels or image patches. Frameworks such as boosting[10, 11], support vector machine[12], random forest[13], multiple instance learning[14], and metric learning[15] have also been used to enhance the matching ability of tracking algorithms.

With the advancements in deep learning mechanisms[16], numerous studies have been proposed to conduct computer vision[17, 18], speech recognition[19, 20], and natural language processing tasks[21, 22]. Motivated by these breakthroughs, the deep learning mechanisms have also been introduced for the single object tracking task[23–26]. Meanwhile, several tracking datasets, such as Object Tracking

Review
Manuscript received September 15, 2020; accepted December 30, 2020; published online March 4, 2021
Recommended by Associate Editor Nazim Mir-Nasiri
Colored figures are available in the online version at https://link.springer.com/journal/11633

Benchmark 2013 (OTB-2013)[27] and Visual Object Tracking 2013 (VOT-2013)[28], have been proposed to evaluate the performance of these tracking algorithms. With these developments, several papers reviewed the advancements and challenges in deep-learning-based tracking algorithms. However, according to our statistical results (see Table 1), none of these existing reviews discusses tracking methods that are recently published in top conferences and journals. In addition, existing reviews mostly concentrate on classifying deep trackers according to their methodologies or on evaluating their performance. It can also be noted that none of the existing reviews details specific components of existing deep trackers. For example, in the two latest reviews, Li et al.[32] present comprehensive classification results based on characteristics such as network architecture, network function, and training frameworks. Yao et al.[34] systematically detail methods that can jointly conduct the video object segmentation and the visual object tracking tasks. To facilitate the development of single object tracking algorithms based on deep learning, in this work, we conclude with the general components of existing deep-learning-based tracking algorithms and present the popular components of deep neural networks, which are proposed for improving the representative ability of the features in the deep neural networks. In addition, we compare the recently proposed deep trackers by collecting and analyzing their metrics on benchmark datasets. In this way, we provide some important observations. For example, through making comparisons, we find that attention mechanisms are widely used to combine the online-updating methods with offline-trained ones. On the other hand, we find that since different components in the deep trackers have their special characteristics, improving only a single component sometimes cannot facilitate the tracking process.

Table 1 Statistical results of existing reviews related to single object tracking algorithms. In the newest work column, the year of publications of the latest work is presented.

Papers	Published year	The newest work	Function
[29]	2013	2012	Classification
[30]	2017	2017	Classification
[31]	2018	2018	Evaluation
[32]	2019	2019	Classification
[33]	2018	2017	Classification
[34]	2019	2019	Classification
Ours	–	2020	Specification

The rest of this paper is organized as follows. In Section 2, we briefly introduce fundamental frameworks and novel mechanisms of deep learning methods. In Section 3, we present the general components of deep trackers. In Section 4, the most popular tracking datasets are detailed and compared with each other. We then present popular metrics used for evaluating the tracking performance on popular tracking datasets. With these metrics, we present and compare the performance of recently published deep trackers in Section 5. Based on these comparison results, we provide several observations in Section 6. Finally, Section 7 summarizes this work.

2 Deep learning models

Deep learning models (i.e., deep neural networks) have been widely studied and applied to several computer vision tasks[35–39], such as image classification[35], object detection[36], and image restoration[38]. In general, the pipeline of the deep neural networks can be seen in Fig. 1. During generating manually designed outputs, different inputs (e.g., a single image for image restoration or continuous frames for video captioning) are fed through a pre-processing module for data augmentation, which aims at alleviating the huge requirement of training data and enhancing the robustness of the networks. After that, several feature processing modules are used to capture the characteristics of the inputs. Based on the captured char-

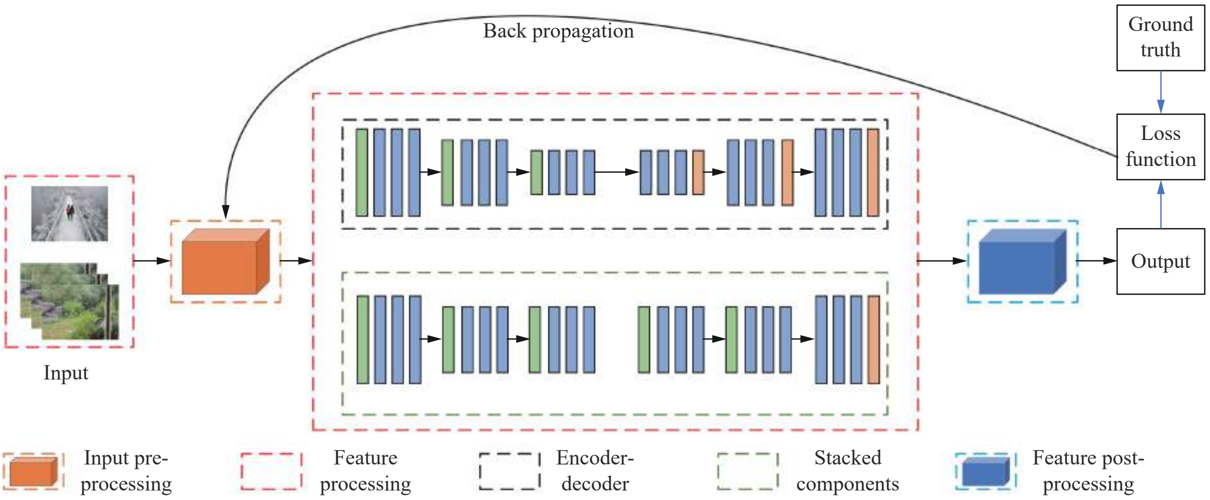


Fig. 1 General pipeline of a deep neural network. Colored figures are available in the online version.

acteristics and manual knowledge, a feature post-processing module is then used to generate outputs, which are supervised by computing the distance to the ground truth. Finally, the calculated loss function is used to update the network parameters through the back propagation. In this pipeline, it is easy to find that the most important part is the feature processing module, whose components are organized in encoder-decoder (denoted by a gray rectangle) or directly-stacked (denoted by a green rectangle) schemes. However, as reported in [40, 41], these two methods are time-consuming, error-prone and data-invariant, always leading to a poor generalization performance. Therefore, numerous feature processing components have been proposed to address the above issues. In addition, the skip connections between these components and the architectures of these components are also explored in the field of neural architecture search (NAS)[41]. According to whether the feature processing module changes its configurations with respect to specific inputs, we roughly split existing deep neural networks into two classes: data-invariant and data-adaptive methods.

2.1 Data-invariant methods

In the early stage of developing deep neural networks, the deep neural networks are static models, whose architectures are fixed and the parameters are iteratively updated in the training stage. Once the network is trained, its architectures and parameters are used to handle all the testing samples. Therefore, it is easy to find that these networks can be considered as data-invariant methods. Next, we detail three popular types of data-invariant methods and discuss their improvements.

2.1.1 Convolution neural networks

The convolution neural network is the first generation of deep neural networks. Given an input, the convolution neural network learns high-dimensional features from the input and generates supervised outputs with respect to the learned features. In this process, several layers, such as convolution layers, pooling layers, batch normalization layers, rectified linear units (ReLU), and fully connected layers, are used to first magnify the feature channel (sometimes also change the spatial resolution), and then gradually reduce the channel numbers. Therefore, the convolution layer, whose function is shown in Fig. 2, is the most basic component of convolution neural networks. In general, the convolution layer uses a trained kernel to convolute the entire input. As such, the resolutions of the outputs generated by convolution layers are decided by factors such as dimension, stride, dilation, and spatial resolution of the convolution kernel. All these factors are closely related to the receptive field of each convolution layer. Moreover, the factors of each convolution layer jointly influence the receptive field of convolution neural networks.

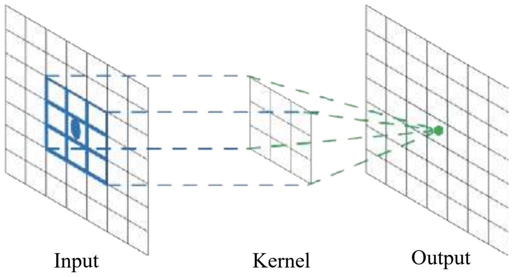


Fig. 2 Illustration of a convolution layer, which is the basic component of convolution neural networks. With different convolution kernels, the convolution layer can change the channels and spatial resolutions of the input feature.

To enlarge the receptive field of the convolution neural networks, the two simplest methods include increasing their depth or width. For example, in the heuristic work[42], multiple convolution layers with small convolution kernels are incorporated along with a max-pooling layer to form the VGG19 network. For obtaining a network as deep as possible, He et al.[43] employ the residual connections in the hierarchical convolution layers. Consequently, the Res101 and Res152[43] are approximately 10 times deeper than their former networks, and therefore significantly outperform their counterparts. Following these two representative works, designing deeper networks is considered as the most useful method to improve the model performance[44, 45]. However, this approach is highly empirical and is limited by computing resources. As a substitute, some methods try to increase the width of convolution neural networks to enlarge their receptive field. Szegedy et al.[46] carefully find out the optimal local sparse structure in a convolution neural network by using the Hebbian principle[47]. Based on this sparse structure, the Inception modules are introduced by implementing parallel branches. Later, Szegedy et al.[48] use hierarchical convolution layers with small kernels to replace convolution layers with large kernels and thus propose the Inception-v2. However, modules proposed in [46, 48] are too various to be used in different tasks. Therefore, Szegedy et al.[45] incorporate the existing Inception modules with residual connections to form the Inception-v4 and Inception-ResNet.

Although increasing the depth or width of neural networks yields a remarkable performance, these two methods are still unreasonable. To facilitate the designation of neural networks and alleviate the computation resource restriction, many studies have been conducted on the characteristics of the convolution layers. To name a few, Huo et al.[49] introduce a feature replay algorithm to learn the parameters of the convolution layers. Similarly, Jeong and Shin[50] modify the standard convolution layer with a channel-selectivity function and a spatial shifting function to dynamically emphasize important features, which have the highest influence on the output vector. Qiao et al.[51] also find that owing to the cascaded architecture of the convolution layers, it is unnecessary to update all the features during image recognition. As well as these methods that abandon irrelevant or unimportant features to improve overall performance[52], there are also some methods that improve their performance by exploring the interpretability of convolution neural networks[53–57]. Geirhos et al.[53] find that convolution neural networks (CNNs) trained with different datasets are biased toward image textures or shapes. For example, ImageNet-trained[54] CNNs are strongly biased toward image textures. By systematically exploring such a bias, Geirhos et al.[53] not only improve the performance of existing networks, but also improve their robustness.

2.1.2 Recurrent neural networks

Unlike the convolution neural networks, the recurrent neural networks (RNNs) are proposed to handle sequential data such as videos and natural language. During handling this information, each recurrent cell not only obtains the hidden states from previous cells, but also takes inputs based on the timestamp. Therefore, both short-term and long-term relationships between the sequential information are dynamically learned and transferred to the subsequent cells. To this end, RNNs maintain a vector of the activation for each timestamp, which makes most RNNs extremely deep. As a consequence, RNNs are difficult to train because of the exploding and the vanishing gradient problems[58, 59]. It is widely known that the first problem can be easily addressed by employing a hard constraint over the norm of these gradients[60, 61]. However, the vanishing gradient problem is more complicated, and two types of methods have been proposed to address this problem.

On the one hand, novel models such as long short-term memory (LSTM)[62] and gated recurrent unit (GRU)[63] are designed. Compared with LSTM, GRU makes it easier to forget long-term information, which is always irrelevant to recent inputs, and thus has better performance on most sophisticated tasks[64]. In detail, the gated recurrent unit contains a reset gate and an update gate, as shown in Fig. 3. In each timestamp, the reset gate takes the previous state h_{t-1} to drop any information that is irrelevant later in the future, while the update gate controls the degree of information that should be carried over from the previous hidden state to the current hidden state. When the reset gate is close to 0, the hidden state is forced to ignore the previous hidden state and reset with the current input X_t. Thus, compared with LSTM, GRU makes it easier to drop useless information from the previous hidden states. The same methodology is also used in later proposed RNNs[65, 66]. Other attempts to overcome the vanishing gradient problem involve using powerful second-order optimization algorithms to regularize weights of RNNs[67, 68] or carefully initialize the weights of RNNs[69]. In general, these above methods aim at proposing deeper recurrent neural networks, rather than improving the mechanisms of previous RNNs. As a consequence, several RNNs can only handle unidirectional temporal relationships between bidirectional sequential data.

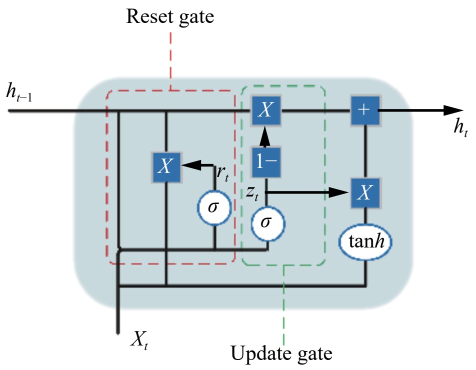


Fig. 3 Illustration of the gated recurrent unit (GRU). At each timestamp, GRU takes hidden state h_{t-1} from the previous cells and current inputs X_t to model the temporal information between sequential data[63].

To fully exploit the bidirectional temporal relationship, there are also several methods that incorporate other mechanisms into RNN. For example, for input sequences whose starts and ends are known in advance, the bidirectional RNNs can make full use of both forward and backward temporal relationships[70–78]. By using a CNN to learn hidden states for each recurrent cell, Liu et al.[73] propose a spatially variant recurrent network for several image restoration tasks. Recently, a fully regulated neural network (NN) with a double hidden layer structure is designed, and an adaptive global sliding-mode controller is proposed for a class of dynamic systems[77]. Overall, RNNs have been continuously enhanced to process the bidirectional relationships, and the forgetting mechanism is important for the above enhancements.

2.1.3 Graph neural networks

As we discussed earlier, CNNs and RNNs are widely used to process Euclidean data (e.g., RGB images) or sequential data. However, these two networks cannot handle data that are represented in the form of graphs. For example, in chemistry, molecules are represented as graphs, and their bioactivity needs to be identified as edges constructed between multiple nodes. In a citation network, articles are linked to each other via citations, and most articles can be categorized into different groups. Therefore, the wide application scenarios of graph data have imposed significant challenges along with opportunities for deep learning methods.

Recently, several graph neural networks (GNNs) have been proposed to address the above challenges. In general, an image can be considered as a fully connected graph, where all the pixels are connected by adjacent pixels. Similarly, as Fig. 4 illustrates, the standard convolution layer can also be seen as a special graph convolution layer, where all convolution kernels are connected in an undirected manner[79]. Therefore, it can be found that GNNs can be achieved by employing constraints in the kernels of traditional CNNs. Gori et al.[80] first propose a GNN-based method to process data with different characteristics, such as directed, undirected, labeled and cyclic

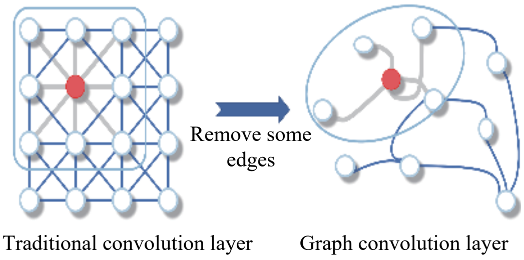


Fig. 4 Illustration of the graph convolution layer[79]

graphs. After this pioneering work, Scarselli et al.[81, 82] find that, with GNN, the representations of a single node can be obtained by propagating the neighbor information in an iterative manner until a stable fixed point is reached. With the above foundation, graphs are widely embedded into convolution layers and neural networks. Bruna et al.[83] develop the graph convolution based on the spectral graph theory. In [84], the graph kernels, whereby graphs or nodes can be embedded into the feature space using a mapping function, is proposed to study the random walk problem[85].

More recently, researchers find that Euclidean data and sequential data can also be represented by special graphs. Therefore, the graph neural networks are also used to handle these data[86−88]. For example, to capture topology and long-range dependencies of a lane graph, Liang et al.[86] extend existing graph convolutions with multiple adjacency matrices and along-lane dilation. By organizing the features in the different channels as nodes, a representative graph layer is proposed to dynamically sample the most representative features[88], leading to less computing consumption and more representative features. Since the components of different data are sparsely or closely related to their neighbors, most data can be represented in the form of different graphs. Hence, possible future research directions include transforming un-graph data into graphs and employing GNNs to handle these transformed graphs.

2.2 Data-adaptive methods

As we mentioned previously, early deep neural networks are designed in the static mode. That is, once the network is trained, its architecture and parameters will not change regardless of the inputs. Therefore, these static networks do not always generalize well, making practical applications difficult. To address this issue, several methods are proposed to change their components with respect to the input. Thus, we categorize these methods as data-adaptive methods and introduce several representative data-adaptive methods in the following sections.

2.2.1 Attention mechanisms

The field of neural language processing (NLP) has witnessed significant development of attention mechanisms, starting from the pioneering work[21]. Bahdanau et al.[21] introduce various attention factors and weight assignment functions. In [89], these factors are further considered, and the inner product of the vectors, which encode the query and key contents, is recommended for computing the attention weights. Later, the landmark work[90] proposes a new standard, and its follow-up studies demonstrate that relative positions can provide better generalization ability than the absolute positions[18,91−93]. Motivated by the success in the NLP tasks, the attention mechanisms are also employed in computer vision (CV) applications[94−99]. However, unlike the attention mechanisms used in NLP, the key and query of the attention mechanisms in CV refer to certain visual elements, and the formulation of attention mechanisms in CV is similar to Transformer[90].

As Fig. 5 shows, given a query element (i.e., the yellow dot) and several key elements (i.e., all colored dots), the attention mechanism aims at adaptively highlighting the key contents based on the attention weights that measure the compatibility of the query-key pairs. Therefore, based on the aggregated domain, existing attention mechanisms can be divided into two categories: spatial-wise and channel-wise attention mechanisms.

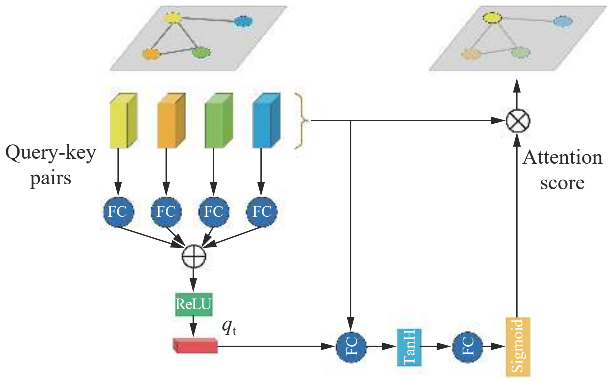


Fig. 5 An example of the attention mechanisms used in computer vision tasks. Given the query-key pairs, the attention mechanism computes scores for each key[94−99].

To name a few, an attention scaling network is introduced to learn attention scores of different regions according to an estimation result of the input images[100]. Zhang et al.[101] employ attention mechanisms in the generative adversarial network to model attention-driven and long-range dependency for image generation tasks. In contrast, Dai et al.[102] develop a novel attention module to adaptively rescale the channel-wise features by using second-order feature statistics. In general, the attention mechanism is biologically plausible and has been widely used in both CV and NLP fields. However, the best configuration of the query-key pairs remains unknown[103]. Hence, there is much room for improving attention mechanisms.

2.2.2 Dynamic neural networks

Dynamic neural networks, which can adjust the network architectures or network parameters depending on the corresponding inputs, have been recently studied in the computer vision field. In early studies, these dynamic neural networks are proposed for the image classification and semantic segmentation tasks by dropping blocks[104−107] or pruning channels[73, 108] for efficient inference. For example, in [109], a soft conditional gate is proposed to select scale transform paths for each layer. Wang et al.[105] attempt to skip several convolution blocks by using a reinforce learning gating function. It is possible to think that dynamic networks with gating functions, which adjust network architectures such as connections according to inputs, appear to be similar to the neural architecture search methods. In most NAS methods, the model parameters are iteratively initialized and trained, while the model architectures are continuously varied. In contrast, the parameters of the dynamic neural networks are all initialized before the training phase, and their architectures always remain unchanged. In addition, most NAS methods search model components based on predefined backbones, whereas the dynamic neural networks with the gating functions remove unnecessary components from predefined intact networks.

Compared with dynamic networks with gating functions, the other types of dynamic neural networks only learn parameters for certain components based on their inputs. Among these models, the most important one is [110], where the dynamic filter network is proposed to learn filtering operations such as local spatial transformations, selective blurring/deblurring, or adaptive feature extraction. After that, the method proposed in [111] dynamically learns two 1D kernels to replace standard 2D convolution kernels. According to their results, this method not only remarkably reduces the number of parameters, but also improves the performance of video frame interpolation. Since then, the dynamic neural networks with parameter learners are found to be useful for image/video restoration tasks. In [112], a dynamic neural network is designed to learn upsampling filters and residual images, which avoid the requirement of explicitly compensating for the motions of restored videos. Overall, the second type of dynamic neural networks can be illustrated as in Fig. 6, which is proposed in [113−115]. In Fig. 6, several standard convolution layers are used to learn the dynamic kernels and bias, which are respectively used to convolute a low-resolution image and further enhance the quality of the high-resolution output. As [115] discussed, a dynamic neural network that learns the parameters for convolution layers can better handle variational degradation in the inputs. Therefore, the dynamic neural networks can be further used to conduct various tasks. However, to the best of our knowledge, there is no work that effectively combines these two kinds of dynamic neural networks.

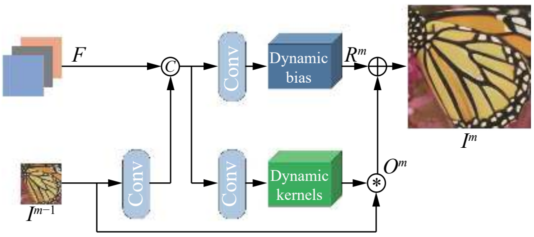


Fig. 6 An example of the dynamic neural network with a parameter learner. With the low-resolution input I^{m-1}, the dynamic neural network learns a set of convolution kernels and biases, which are used to generate the high-resolution output I^m[113−115].

2.2.3 Other neural networks

Recently, neurological research has significantly progressed and continues to reveal new characteristics of the biological neurons and brains. These new characteristics have led to several new types of artificial neural networks that can selectively make forward inference based on their inputs. For example, Dai et al.[116] find that using fixed convolution kernels to process different inputs inevitably limits CNN to model geometric transformation. Therefore, a 2D offset is learned from the preceding feature maps to regularize the learned convolution kernels. After this work, improved deformable convolutional networks such as Deformable ConvNet V2[117], SqueezeSeg V3[118] and Variational Context-Deformable ConvNets[119] are also proposed. For example, as shown in Fig. 7, the offsets are obtained by applying a standard convolution layer over the same input feature map, thus it has the same spatial resolution as that of the input feature map. When applied to different tasks, the standard convolution layer can be replaced by different operations. Therefore, the deformable convolution layer not only dynamically learns the feature maps, but can also be efficiently used for various tasks. On the other hand, the spiking neural networks (SNN) are introduced to mimic how information is encoded and processed in the human brain by employing spiking neurons as computation units[120, 121]. Unlike standard CNNs, SNNs use temporal aspects for the information transmission as in biological neural systems[122], thereby providing a sparse yet powerful computing ability[123]. However, due to the sparse nature of spike events, SNNs are always used for image classification, except in [121], where SNNs are successfully employed to

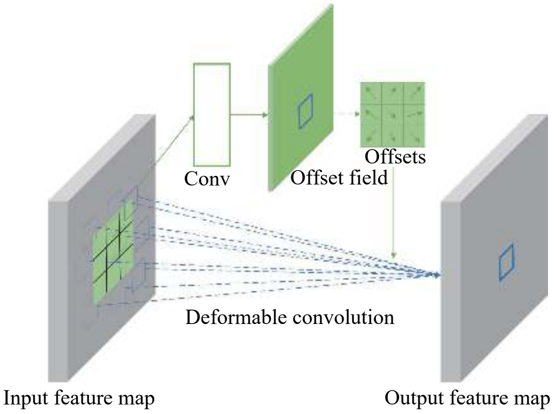


Fig. 7 Illustration of a 3 × 3 deformable convolution. Conv indicates the standard convolution layers[116−119].

the object detection task by using the DNN-to-SNN conversation method[124].

3 Deep tracker components

In this section, we detail several existing deep trackers, whose components can be generally concluded as in Fig. 8. First, by considering that most deep trackers take cascaded blocks (e.g., convolution layers or residual blocks) to extract the features of the tracked targets, the feature extraction module is concluded and discussed. Second, because tracking algorithms aim at locating the same target in different frames, the mechanisms of estimating the motion patterns are compared. Third, we also discuss how deep trackers obtain the bounding boxes, which indicate precise locations of the tracked targets. Finally, we discuss how loss functions influence the performance of deep trackers.

3.1 Feature extraction module

Feature extraction is important for tracking algorithms. In general, the extracted features should effectively and robustly represent the tracking target. However, such a requirement is difficult to meet due to challenges such as illumination variations or appearance variations of the tracked targets. To address these challenges, several modules are designed in previous methods to extract various local and statistical features. For example, Ross et al.[7] propose a tracking method to incrementally learn a low-dimensional subspace representation, which is concluded as gray features in [30]. By taking mean shift iterations, Comaniciu et al.[125] introduce the color distribution of probable target models and the target candidates. A texture feature is also proposed in [126]. Henriques et al.[127] extend the traditional RGB space to a 11-dimensional color space and use the principle component analysis method to extract features from this 11-dimensional color space. Later, Henriques et al.[128] further incorporate a correlation filter tracker with the kernel space and propose the histogram of oriented gradient feature. Except for these methods, other methods also employ multiple manually designed features to represent the tracked target[129]. However, these manually designed features only concentrate on certain characteristics of the tracking targets, thus, they are easy to be violated during tracking.

Since deep neural networks are effective at extracting features with a powerful representative ability, they become the substitutions of the above manually designed features. For example, hierarchical convolution layers are introduced into the correlation filter algorithm[130]. Experimental results reported in [130] indicate that low-level features contain more information of the target location, whereas high-level features include more semantic information and are more robust than low-level ones. Inspired by this observation, Qi et al.[131] extend the three convolution layers used in [130] to six layers, and take dynamic parameters to adaptively fuse features from these six layers. After these methods, the methodology of the correlation filter tracking algorithms is utilized to form the Siamese networks. In [132], the first Siamese tracker, i.e., Siamese-FC, is designed with two parallel branches, which respectively extract features from the first frame and the remaining ones. With these two branches, the features from the first frame are taken as a convolution kernel to scan all positions of the features from other frames. Finally, the response map is obtained, and the max value in this map is seen as the target location.

In general, the above deep trackers achieve better per-

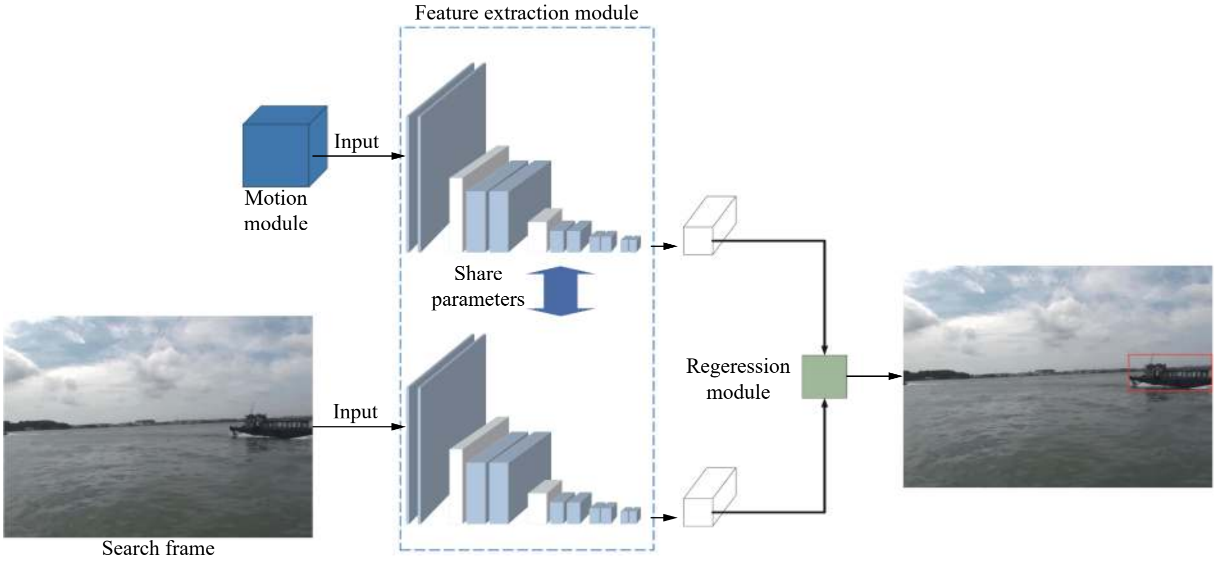


Fig. 8 General components of deep trackers. In each timestamp, the feature extraction module extracts features from target templates in the motion module and the search frame. Then these two kinds of features are fed through the regression module to generate the bounding box.

formance than most conventional trackers. However, with the development of deep neural networks, deep networks are also employed in learning-based trackers for further improving their performance and robustness. There are two types of methods for deepening learning-based trackers. The first type is to transfer pre-trained deep neural networks to learning-based trackers. Among these transfer-based methods, the most representative work is presented in [133], where the authors find that the learned feature representation for a target should remain spatially invariant. In addition, they theoretically find the zero-padding configuration of CNN trackers influences the spatial invariance restriction. Based on this work, Lukezic et al.[134] use a ResNet50 as a backbone to simultaneously conduct the visual object tracking and video object segmentation tasks. Similarly, the ResNet50 is also taken as a backbone to implicitly and dynamically encode the camera geometric relationship, and hence address missing target issues such as occlusion[135]. Chen et al.[136] view the tracking problem as a parallel classification and regression problem, and take a pre-trained classification network to solve the former problem. In contrast, the other kind of method directly constructs deep neural networks. By conducting extensive and systemic experiments, Zhang and Peng[137] find that the network stride, receptive field and spatial size of network output are important for constructing deep Siamese networks. They then propose the cropping-inside residual (CIR) units, down-sampling CIR unit, CIR-Inception and CIR-NeXt units to design deeper or wider Siamese networks than previous ones. Later, these units and observations proposed in [137] are also widely used to form Siamese trackers with deep architectures[138, 139]. However, Zhang and Peng[137] also indicate that the perceptual inconsistency between the target template and the search frame should be carefully designed for robust tracking. Therefore, for tracking algorithms, it is unnecessary to form extraction modules as deeply as possible. With this observation, several novel frameworks, such as generative adversarial network (GAN) and attention mechanisms are employed into deep trackers for effectively learning features. On the other hand, the above indication also shows that the motion pattern of tracked targets is another important consideration for deep trackers. Therefore, in the next subsection, the motion estimation module is discussed to explore such importance.

3.2 Motion estimation module

Unlike the image classification task, single object tracking aims at locating a given target in all frames. Therefore, the motion pattern between consecutive frames or tracked targets is important for enhancing the robustness and effectiveness of the tracking algorithms. In detail, for deep trackers such as Siamese networks, there is a branch that learns features related to previous target appearances. As Fig. 8 illustrates, the motion module is designed for constructing a target template, hence deep trackers can dynamically update the appearances of the tracked targets. For example, Ning et al.[76] first take an object detection method to choose candidate samples, which are then fed through LSTM blocks to generate object locations. Inside LSTM blocks, the context relationships between consecutive frames are used to select samples related to the targets. Reinforcement learning has also been used to capture motion patterns. In [63], an action-decision tracker (ADNet) is proposed to predict object locations by a learned agent. Specifically, this agent is trained to foresee movement and scale-change based on the present frame. However, since the motion estimation module tends to forget the appearance from the first frame, they are easily influenced by heavy occlusions or out-of-the-view movements.

To address the above problem, recently proposed motion estimation modules take both the search frame and other frames as inputs. For example, Teng et al.[140] propose a neural network to explicitly exploit object representations from each frame and changes among multiple frames in the same video. Thus, the proposed network could integrate object appearances with their motions for effectively capturing temporal variations among consecutive frames. In [141], all frames in each tracking sequence are used to train a reinforcement learning network, which aims at producing a continuous action for predicting the optimal object location. In contrast, Zhang et al.[101] introduce a motion estimation network to learn how to update the object template. In detail, this motion estimation network is provided with the first frame, the current frame, and the template of the current frame. Thus, the appearances from the first frame are always emphasized, leading to a more robust tracking performance than its counterparts. Li et al.[142] innovatively use gradient information between consecutive frames and the current frame. For better using of gradient information and avoiding of the over-fitting problem, they also propose a template generalization training method. According to their experimental analysis, motion estimation modules always learn a general template from the initial appearance, and continuously fine-tune the general template based on the appearances in the search frames. Therefore, it can be concluded that the appearance from the initial frame can help avoid the tracking-drifting problem to some extent. However, excessive information from the initial frame will influence the overall tracking performance (e.g., GRU tends to forget information from early times>tamps, thus are more effective and robust than LSTM). This indicates that balancing the importance of the initial frame and other frames still needs more exploration.

3.3 Regression module

With the motion estimation module, deep trackers can maintain a template of the tracking targets. Thus, the feature extraction module can learn features from this template and the search frame to locate tracking targets. However, since the features in deep neural networks are high-dimensional, regressing the bounding box from these extracted features also requires more researches. In general, for regressing a bounding box, a response map is first obtained via the following operation:

$$f_\theta(z,x) = \phi_\theta(z) * \phi_\theta(x) + b \tag{1}$$

from which ϕ indicates the feature extraction module with parameters θ, b denotes a bias term, z and x indicate the target template and search frame, respectively. Then, the maximum value in the response map is taken as the target location. In addition, for processing the scale variations, several scale parameters are first manually given and then gradually updated on the basis of the search frames or fixed factors.

Recently, it is demonstrated that the manually provided scale parameters cannot fully process the scale variations. In addition, with the deepening of the feature extraction module, more and more spatial information is lost, leading to poor tracking performance. Therefore, to alleviate the above issues, several methods take features from different levels of the feature extraction module to generate response maps, and dynamically fuse these response maps to obtain the final one. During this process, the multi-scale information in these features is incorporated together to handle the scale variation problem and improve the precision of the final response map. For example, in [143], several features are first used to obtain the final response map with multiple channels, and then, a classification sub-network and a regression one are used to decode the location and scale information of the object. Wang et al.[144] first take a semi-supervised video object segmentation network to obtain the segmentation results of the object, then use different branches to regress the bounding box and scores for each pixel. Fan and Lin[145] leverage high-level semantic information and low-level spatial information to obtain different response maps. These response maps are then used to progressively fine-tune the response map obtained by computing the final output features of the feature extraction module. The same methodology can also be found in [146]. In contrast, Ge et al.[147] search and select only partial features to make the response map. This is the first work that considers differences between features at different channels, rather than features at different levels, and according to their experimental results, the selected features could provide better performance than multi-level features. From this aspect, it seems that high-dimensional features should not be entirely used to obtain the response maps. Therefore, a possible research direction for the regression module is the selection of optimal features from the feature extraction module to generate the response map and scale parameters.

3.4 Loss function

After detailing the general components of deep trackers, in this section, we introduce popular loss functions used for training these deep trackers. First, we present statistical results of the loss functions. As listed in Table 2, among works recently published in top conferences or journals (e.g., *IEEE Transactions on Image Processing, IEEE Transactions on Pattern Analysis and Machine Intelligence*, and *European Conference on Computer Vision*), the cross entropy loss function, logistic loss function, and smooth L1 loss function are the three most popular loss functions. Because both the cross entropy loss function and logistic loss function supervise categories of each location and the latter loss function is a special version of the former one, we only introduce the cross entropy loss function, which is also the basis for the focal loss function. In addition, we discuss the smooth L1 loss function since it directly supervises the bounding box.

3.4.1 Cross entropy loss function

In general, the cross entropy loss function can be defined as follows:

$$Loss(p,q) = -\sum_x (p(x)\log q(x) + (1-p(x))\log(1-q(x))) \tag{2}$$

where p is the expected contribution of the sample x, and q is the generated contribution. However, for the single object tracking task, there are only two kinds of samples, i.e., target and background samples, which are respectively consisted of pixels belonging to the tracked target and other pixels. Therefore, in some tracking algorithms, the cross entropy loss function is used to force these algorithms to learn the discriminative features of the object and background. For example, in [148], the cross entropy loss function is incorporated along with the focal loss function to supervise the feature extraction module.

Table 2 Statistical results of loss functions used in recent deep trackers

Loss function	Number	Citations
Cross entropy loss	19	[136, 139, 143, 145, 148–162]
Logistic loss	10	[137, 141, 142, 144, 150, 162–166]
Smooth L1 loss	10	[133, 139, 145, 150, 152, 156, 159, 160, 167, 168]
L2 loss	6	[140, 147, 150, 167–169]
L1 loss	6	[158, 167, 170–173]
Focal loss	5	[148, 167, 169, 171, 162]
IoU loss	4	[136, 143, 148, 157]

In [161], this loss function is used to force a classifier to accurately classify pixels. However, since the cross entropy loss function only concentrates on the categories of each pixel, it cannot directly supervise tracking algorithms to generate accurate bounding boxes. Thus, the cross entropy loss function is typically used along with other loss functions such as the focal loss function or the IoU loss function.

3.4.2 Smooth L1 loss function

It is widely known the smooth L1 loss function is a special version of the L1 loss function, and the L1 loss function is the square root of the L2 loss function. Therefore, these three loss functions can be defined as

$$L1(x) = |x| \tag{3}$$

$$L2(x) = x^2 \tag{4}$$

$$SmoothL1(x) = \begin{cases} \lambda x^2, & \text{if } |x| < \beta \\ |x| - \gamma, & \text{otherwise} \end{cases} \tag{5}$$

where λ, β and γ are the pre-defined parameters. Clearly, compared with the L2 loss function, the smooth L1 loss function tends to generate relatively small values. In addition, the smooth L1 loss function tends to generate higher values than the L1 loss function, when $|x| < \beta$. This ensures that during training deep trackers, the gradient value is appropriate to get off the local optimal solution. However, although the smooth L1 loss function can directly supervise the bounding box, it tends to force deep trackers prior to short-term templates.

Therefore, not only the smooth L1 loss function but also its methodology are widely used to train deep trackers. For example, in [133], the smooth L1 loss function is used together with a classification loss function. In [150], the methodology of the smooth L1 loss function is used to update a short-term template, thus the model drifting and inconsistency of target template problems are avoided. Overall, even with an effective feature extraction module, motion estimation module, and regression module, the loss function is also important for obtaining an effective tracker.

However, since a single loss function cannot effectively supervise all modules, multiple loss functions are often jointly used in the weighted summation manner.

4 Visual tracking datasets

It is widely known that most deep learning methods rely on benchmark datasets with a large amount of labeled data. In addition, with the developments in deep learning trackers, previous datasets with limited data cannot fully validate their effectiveness. Therefore, several tracking datasets have been proposed in recent years (see Table 3). In this section, we discuss several conventional datasets and two newly proposed ones.

4.1 Object tracking benchmark datasets

In general, the Object Tracking Benchmark datasets are the most widely used datasets for evaluating tracking algorithms. In [27], a testing dataset namely OTB-2013 is the first proposed dataset to address and analyze the initialization problem of object tracking. In OTB-2013, there are a total of 51 sequences with manually annotated bounding boxes in each frame[1]. For further analyzing the tracking performance, all these 51 sequences are categorized with 11 attributes, i.e., illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutters, and low resolution. After that, Wu et al.[177] later extend OTB-2013 by adding other 48 additional videos to obtain OTB-2015, which is also denoted by OTB-100[2]. In addition, 50 difficult and representative sequences in OTB-100 are selected to form other datasets, namely OTB-50.

4.2 Visual object tracking datasets

The other datasets, which are denoted as Visual Object Tracking datasets[28, 176, 178, 179, 192–194], are also popular for tracker evaluation. Unlike OTB datasets, trackers are initialized by themself in this dataset, and they are allowed to be reinitialized when a failure is detected by comparing the predicted bounding box with the ground-truth annotations. In the first proposed VOT-2013[28], each frame of the released 16 sequences is labeled with different attributes. After VOT-2013, the VOT datasets are updated per year, and in the VOT-2019 challenge[179], there are in total 60 sequences.

4.3 Large-scale single object tracking dataset

A large-scale dedicated benchmark with a high quality of training and testing sequences is proposed in [175]. Considering that this dataset contains 1 400 videos with large-scale variants, it is called as large-scale single object tracking (LaSOT) Dataset. LaSOT is different from existing datasets, since it provides both visual bounding box annotations and rich natural language specifications. This dataset also takes the class imbalance into consideration, thus each video in LaSOT contains 2 512 frames and is categorized into only one class. In addition, LaSOT provides two different evaluation protocols. In the first protocol, all the 1 400 sequences are used for evaluation, and the tracking algorithms are allowed to use any sequence from the other dataset for training. In the second

[1] In the jogging video, two different targets are annotated. Therefore, there are totally 50 videos and 51 sequences.

[2] Except the jogging video, there is also a sequence (i.e., skating2) with two different annotated targets. Therefore, there are only 98 videos and 100 sequences.

Table 3 Statistical results of existing tracking datasets. The label indicates that this attribute is not considered. Several results are referred from [174, 175].

Dataset	Total			Train			Test		
	Classes	Videos	Boxes	Classes	Videos	Boxes	Classes	Videos	Boxes
OTB-2013[27]	10	50	29k	–	–	–	10	50	29k
VOT-2014[176]	11	25	10k	–	–	–	11	25	10k
OTB-2015[177]	22	100	59k	–	–	–	22	100	59k
VOT-2017[178]	24	60	21k	–	–	–	24	60	21k
VOT-2019[179]	30	60	19.9k	–	–	–	30	60	19.9k
ALOV++[180]	59	314	16k	–	–	–	59	314	16k
NUS PRO[181]	12	365	135k	–	–	–	12	365	135k
TColor128[182]	27	129	55k	–	–	–	27	129	55k
Nfs[183]	33	100	38k	–	–	–	33	100	38k
UAV123[184]	9	123	113k	–	–	–	9	123	113k
UAV20L[184]	5	20	59k	–	–	–	5	20	59k
OxUvA[185]	22	366	155k	–	–	–	22	366	155k
LaSOT[175]	70	1.4k	3.3M	70	1.1k	2.8M	70	280	685k
TrackingNet[186]	21	31k	14M	21	30k	14M	21	511	226k
MOT15[187]	1	22	101k	1	11	43k	1	11	58k
MOT16/17[188]	5	14	293k	5	7	200k	5	7	93k
KITTI[189]	4	50	59k	4	21	–	4	29	–
ILSVRC-VID[190]	30	5.4k	2.7M	30	5.4k	2.7M	–	–	–
YT-BB[191]	23	380k	5.6M	23	380k	5.6M	–	–	–
GOT-10k[174]	563	10k	1.5M	480	9.34k	1.4M	84	420	56k

protocol, both the training and testing sets are predefined to make fair comparisons.

4.4 GOT-10k dataset

Unlike the LaSOT, the GOT-10k dataset aims at providing a wide coverage of common moving objects in the wild. Specifically, GOT-10k contains over 10 000 video segments with more than 1.5 million manually labeled bounding boxes. These video segments have over 560 classes of moving objects and 87 motion patterns. To ensure a comprehensive and unbiased coverage of diverse moving objects, GOT-10k also uses the semantic hierarchy of WordNet[195] to guide class populations. Thus, each sequence in GOT-10k is labeled with 2D labels: object and motion classes. The former label denotes the target that will be tracked, whereas the other one describes the motion patterns of the target.

5 Performance evaluation

After introducing several datasets used for training and testing deep trackers, in this section, we first introduce several popular metrics that are used to evaluate the performance of these trackers. We then present quantitative results of recently proposed methods. Finally, we also discuss the changes and differences between methods that achieve state-of-the-art performance.

5.1 Evaluation metrics

As we previously discussed, four datasets, i.e., the OTB, VOT, LaSOT, and GOT-10k datasets, are widely used to train and test tracking algorithms. Therefore, in this subsection, we introduce several metrics used in these datasets, such as precision, success, robustness, and accuracy.

5.1.1 Precision

The precision metric is based on the most basic metric, i.e., center location error, which is defined as the average Euclidean distance between the center locations of the generated bounding box and the corresponding ground truth. Therefore, a simple method to evaluate the tracking performance is to summarize the average center location errors over the entire sequence. However, this simple method is unsuitable when the model drifting problem occurs, since the drifted bounding box is randomly distributed. Therefore, the precision metric is proposed to measure the percentage of frames, in which the estimated location is within the given threshold distance of the ground truth. The default threshold is 20 pixels. In addition, by varying this threshold, the precision plot can be obtained to make thorough comparisons.

5.1.2 Success and accuracy

Evaluating the tracking performance based on the center location error is sometimes ineffective. Therefore, the overlap ratio (also known as accuracy) between the generated bounding and labeled boxes is also considered for evaluation. Given a generated bounding box r_t and a labeled one r_a, their overlap ratio is defined as follows:

$$S = \frac{|r_t \cap r_a|}{|r_t \cup r_a|} \tag{6}$$

where $\cap$ and $\cup$ indicate the intersection and union of these two boxes, and $|\cdot|$ represents the number of pixels in the region. On the one hand, by averaging the values of accuracy in all frames, the average accuracy can be obtained to evaluate the tracking performance. On the other hand, for computing the metric success, a threshold t_o is also given, and the default value of t_o is always fixed as 0.5. Thus, frames whose overlap S is greater than t_o are seen as frames where the target is successfully tracked. By counting the number of these frames, the success can be calculated. However, since the default value of t_o cannot fully represent the overall performance, values of t_o are always varied from 0 to 1 for obtaining the success plot. For quantitative comparison, the success plot is always compared with the area under curve (AUC).

5.1.3 Robustness

For evaluating a tracking algorithm, a conventional method is to run this algorithm throughout a sequence with initialization from the ground truth position in the first frame, and then compute certain metrics such as precision and success. However, because several tracking algorithms are sensitive to the initialization, the above conventional method is partial. For this reason, the robustness metric is proposed. Specifically, there are two kinds of robustness metrics. The first kind of robustness metrics is calculated by initializing the ground truth position from different frames. Therefore, it is named the temporal robustness evaluation (TRE). In contrast, the second kind of robustness metrics are computed by perturbing the initialization from different positions in the first frame. Therefore, it is named the spatial robustness evaluation (SRE).

5.2 Quantitative results

In this subsection, we present metrics of recently proposed deep trackers on benchmark datasets such as OTB-2013, OTB-2015 and LaSOT. First, metrics such as precision and success applied to these three datasets are presented in Tables 4–6, from which we can find that visual tracking via adversarial learning (VITAL)[153], discriminative and robust online learning for Siamese visual tracking (DROL-RPN)[150], and Siam R-CNN[203] achieve the best performance on these datasets. In detail, on the OTB-2013 dataset, VITAL achieves the highest value of precision and comparable performance in term of success. By comparing the mechanisms of the two best methods, it can be found that both of them consider the imbalanced distributions of the target samples and background samples. In addition, VITAL additionally considers the overlap between the target samples. For addressing the above issues, VITAL takes the adversarial learning, while Yang et al.[149] use a novel online training strategy. As their performance indicates, the adversarial learning is more effective. On the OTB-2015 dataset, there are also two outperforming methods, i.e., Siamese visual tracking (DROL-RPN)[150] and deformable Siamese attention networks (SiamAttn)[139]. Among these two methods, the former one uses a Siamese attention mechanism to generate deformable self-attention and cross-attention weights, leading to an online-updated target and representative features from both the target template and the search frame. In contrast, the latter one combines an online module with an offline Siamese network via the attention mechanisms. Empirically speaking, by jointly extracting representative features from the target template and search frame, DROL-RPN is more effective and robust than SiamAttn. Similar to SiamAttn, the long-term tracking with meta-updater (LTMU)[208] also combines offline-trained Siamese architectures with the online-update-based trackers, which takes target appearance from the first frame and the previous appear-

Table 4 Precision and success of deep trackers on the OTB-2013 dataset. Red: the best result; blue: the second best result.

Trackers	Precision	Success
SiamFC+[137]	0.880	0.670
SiamRPN+[137]	0.920	0.670
ST-LSTM[140]	0.911	0.681
DP-Siam[141]	0.918	0.686
GDT[161]	0.938	0.711
Yang et al.[149]	0.941	0.702
Sa-Siam[163]	0.896	0.677
VITAL[153]	0.950	0.710
TADT[196]	0.896	0.680
MemTrack[197]	0.849	0.642
DSLT[169]	0.934	0.683
SiamFC[198]	0.809	0.607
StructSiam[165]	0.880	0.638
SSD[160]	0.813	0.637
CR-RE[199]	0.677	0.538
HCFT[200]	0.923	0.638
EMDSLT[166]	0.853	0.626
MDSLT[166]	0.815	0.600
LSSiam[162]	0.884	0.663

Table 5 Precision and success of deep trackers on the OTB-2015 dataset. Red: the best result; blue: the second best result.

Trackers	Precision	Success
SiamRPN++[133]	0.914	0.696
SiamFC+[137]	0.850	0.640
SiamRPN+[137]	0.900	0.670
SiamAttn[139]	0.926	0.712
ST-LSTM[140]	0.881	0.656
GradNet[142]	0.861	0.639
GDT[161]	0.910	0.683
DROL-RPN[150]	0.937	0.715
VITAL[153]	0.917	0.682
TADT[196]	0.866	0.660
MemTrack[197]	0.820	0.626
DSLT[169]	0.909	0.660
StructSiam[165]	0.851	0.621
CR-RE[199]	0.617	0.486
SiamRPN[152]	0.851	0.637
GCT[164]	0.854	0.648
UDT[201]	0.760	0.594
ROAM[168]	0.908	0.681
Siam R-CNN[202]	0.891	0.701
MetaCREST-01[155]	0.856	0.637
DaSiamRPN[203]	0.865	–
DRL-IS[204]	0.909	0.671
PG-Net[159]	0.892	0.691
PTAV[205]	0.862	0.632

Table 6 Precision and success of deep trackers on the LaSOT dataset. Red: the best result; blue: the second best result.

Trackers	Precision	Success
GradNet[142]	0.351	0.365
SiamCAR[143]	0.510	0.507
MDNet[24]	0.373	0.397
VITAL[153]	0.360	0.390
StructSiam[165]	0.333	0.335
ROAM[168]	0.368	0.390
ROAM++[168]	0.445	0.447
Siam R-CNN[202]	–	0.648
Dimp50[206]	0.564	0.568
LTMU[207]	0.572	0.572
GlobalTrack[208]	0.528	0.517
ATOM[209]	0.500	0.501

ance to model target templates. Evidently, not only mechanisms of the motion estimation module but also inputs to the motion estimation module are important for obtaining discriminative target templates.

Here, we also introduce historical results on the VOT challenges. As Fig. 9 illustrates, the best performance on VOT challenges is gradually increased. Among these metrics, the robustness has been remarkably enhanced in 2020. By analyzing the methodologies of these top trackers, it can be found that Siamese trackers are still the most popular methods. In addition, in 2020, most Siamese trackers are designed with multiple tasks, such as video segmentation and object detection, leading to superior precision and robustness. Therefore, multi-task learning is a possible methodology to design effective trackers and achieve semi-supervised/un-supervised trackers.

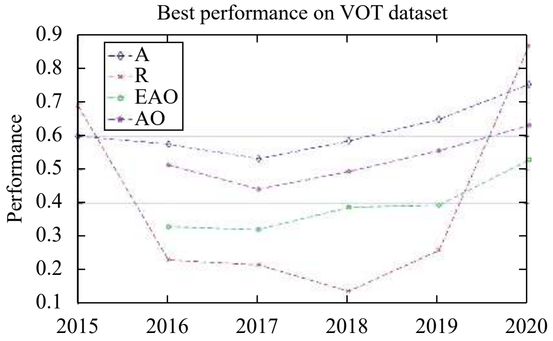


Fig. 9 Historical results on the VOT datasets. A and R indicate the accuracy and robustness metrics, while EAO and AO denote the expected average overlap and average overlap. All results are collected from official presentations of existing VOT challenges[28, 175, 176, 178, 179, 192, 193].

6 Discussions

6.1 Relationship among different components

Different components of deep trackers have different functions and characteristics. For the feature extraction module, the most important function is to extract representative features that can be used to separate the object from the background. Therefore, the feature extraction module should dynamically focus on the most salient parts of the object appearance. However, since the appearance of the tracked objects always varies along the entire sequence, the feature extraction module is easily fooled by these appearance variants. In addition, the environment around the object seriously influences this appearance. To address these two issues, improving the robustness of the feature extraction module and the regression module is helpful. For example, the global information of an object usually changes less than the local one. Thus, taking features generated by convolution layers with a big receptive field can help easily determine the coarse location of the object. However, tracking algorithms are always designed to precisely locate the given targets. As a result, the local information is also used to regress the fine locations of the object. It is easy to find that different features in the feature extraction module can be used to enhance performance of the regression module. Therefore, the feature extraction module and regression modules are complementary to each other. On the other hand, the motion estimation module, which maintains the previous target templates, is also based on the feature extraction module. This is because the target template is always updated with a certain loss function, which uses the feature extraction module to extract the features. However, the motion estimation module is somewhat different from the above two modules. This is because, for maintaining effective target templates, the appearances in all frames are useful. However, the above two modules must handle all frames to locate the tracking target. This contradiction is also the reason why several works iteratively emphasize the initial appearance during tracking. Overall, the feature extraction module is complementary with the regress module, whereas the motion estimation module is based on the feature extraction module and has its own special characteristics.

6.2 Exploration of more effective frameworks

In Section 3, the general pipeline of deep trackers is concluded in two steps: 1) The feature extraction module extracts features from the search frame and target templates of the motion estimation module, 2) The regression module takes these two kinds of features to generate the response map, and then estimates the max value in the response map to regress the bounding box. It is easy to find that this pipeline is similar to the correlation filter methods, which heavily rely on manually defined features. Therefore, the main difference between these two kinds of tracking algorithms lies in the extracted features. However, it is demonstrated that features extracted by deep neural networks are always redundant and sometimes noisy[52, 210–212]. Therefore, in the field of image classification, several mechanisms are proposed to address this issue and achieve better performance than previous classification methods. In contrast, the above issue has remained largely unexplored in the field of visual tracking. Therefore, we believe that more effective frameworks can be proposed by exploring the inherent characteristics of the features in these deep trackers.

6.3 Interpretability of motion estimation module

It is widely known that deep learning mechanisms are somewhat unreasonable, which is also a popular research point. However, existing research related to the interpretability of deep learning mechanisms mostly focus on the image classification task, which only concentrates on modeling the spatial relationships between different pixels. In contrast, these existing deep trackers not only take the deep learning mechanisms to extract the spatial information of the object, but also use them to estimate the motion patterns and update the target template. Therefore, it is easy to find why the deep learning mechanisms can be used to form the motion estimation module is still unknown. From this perspective, there is also a possible research point to enhance existing deep trackers. Last but not least, compared with modeling the spatial relationships between the pixels in the same frame, capturing the temporal relationships additionally considers relationships of pixels in different frames. This indicates that exploring the mechanisms and interpretability of the motion estimation module is also important for deep trackers.

7 Conclusions

In this work, we review several recently proposed single object tracking algorithms based on deep learning mechanisms. We also review traditional static convolution neural networks, recurrent neural networks, and the graph neural networks. Based on these static networks, we introduce several data-adaptive methods such as attention mechanisms and dynamic networks. After that, we detail the general components of deep trackers and present their experimental details. Finally, by systematically analyzing the experimental results, we provide three different research directions for exploring the mechanisms of deep-learning-based trackers.

Open Access

Acknowledgements

This work was supported by National Natural Science

Foundation of China (Nos. 61922064 and U2033210), Zhejiang Provincial Natural Science Foundation (Nos. LR17F030001 and LQ19F020005), the Project of Science and Technology Plans of Wenzhou City (Nos. C20170008 and ZG2017016).

References

[1] D. Comaniciu, V. Ramesh, P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003. DOI: 10.1109/TPAMI.2003.1195991.

[2] P. Perez, C. Hue, J. Vermaak, M. Gangnet. Color-based probabilistic tracking. In *Proceedings of the 7th European Conference on Computer Vision*, Springer, Copenhagen, Denmark, pp. 661–675, 2002. DOI: 10.1007/3-540-47969-4_44.

[3] D. Comaniciu, P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002. DOI: 10.1109/34.1000236.

[4] M. Isard, A. Blake. CONDENSATION-conditional density propagation for visual tracking. *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998. DOI: 10.1023/A:1008078328650.

[5] J. Kwon, K. M. Lee. Tracking by sampling and integratingmultiple trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1428–1441, 2014. DOI: 10.1109/TPAMI.2013.213.

[6] A. Adam, E. Rivlin, I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, New York, USA, pp. 798–805, 2006. DOI: 10.1109/CVPR.2006.256.

[7] D. A. Ross, J. Lim, R. S. Lin, M. H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, vol. 77, no. 1–3, pp. 125–141, 2008. DOI: 10.1007/s11263-007-0075-7.

[8] X. Mei, H. B. Ling. Robust visual tracking and vehicle classification via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2259–2272, 2011. DOI: 10.1109/TPAMI.2011.66.

[9] D. Wang, H. C. Lu, M. H. Yang. Online object tracking with sparse prototypes. *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 314–325, 2013. DOI: 10.1109/TIP.2012.2202677.

[10] H. Grabner, H. Bischof. On-line boosting and vision. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, New York, USA, pp. 260–267, 2006. DOI: 10.1109/CVPR.2006.215.

[11] S. Avidan. Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261–271, 2007. DOI: 10.1109/TPAMI.2007.35.

[12] S. Avidan. Support vector tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004. DOI: 10.1109/TPAMI.2004.53.

[13] A. Saffari, C. Leistner, J. Santner, M. Godec, H. Bischof. On-line random forests. In *Proceedings of the 12th IEEE International Conference on Computer Vision Workshops*, IEEE, Kyoto, Japan, pp. 1393–1400, 2009. DOI: 10.1109/ICCVW.2009.5457447.

[14] B. Babenko, M. H. Yang, S. Belongie. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2010. DOI: 10.1109/TPAMI.2010.226.

[15] N. Jiang, W. Y. Liu, Y. Wu. Learning adaptive metric for robust visual tracking. *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2288–2300, 2011. DOI: 10.1109/TIP.2011.2114895.

[16] Y. LeCun, Y. Bengio, G. Hinton. Deep learning. *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. DOI: 10.1038/nature14539.

[17] R. Girshick, J. Donahue, T. Darrell, J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016. DOI: 10.1109/TPAMI.2015.2437384.

[18] X. Q. Zhang, R. H. Jiang, T. Wang, P. C. Huang, L. Zhao. Attention-based interpolation network for video deblurring. *Neurocomputing*, 2020. DOI: 10.1016/j.neucom.2020.04.147.

[19] S. Kim, T. Hori, S. Watanabe. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, New Orleans, USA, pp. 4835–4839, 2017. DOI: 10.1109/ICASSP.2017.7953075.

[20] Z. 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 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, Brisbane, Australia, pp. 4460–4464, 2015. DOI: 10.1109/ICASSP.2015.7178814.

[21] D. Bahdanau, K. Cho, Y. Bengio. Neural machine translation by jointly learning to align and translate. [Online], Available: https://arxiv.org/abs/1409.0473, 2014.

[22] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, G. Hinton. Grammar as a foreign language. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, Cambridge, USA, pp. 2773–2781, 2015.

[23] L. J. Wang, W. L. Ouyang, X. G. Wang, H. C. Lu. Visual tracking with fully convolutional networks. In *Proceedings of IEEE International Conference on Computer Vision*, IEEE, Santiago, Chile, pp. 3119–3127, 2015. DOI: 10.1109/ICCV.2015.357.

[24] H. Nam, B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Las Vegas, USA, pp. 4293–4302, 2016. DOI: 10.1109/CVPR.2016.465.

[25] L. J. Wang, W. L. Ouyang, X. G. Wang, H. C. Lu. STCT: Sequentially training convolutional networks for visual tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Las Vegas, USA, pp. 1373–1381, 2016. DOI: 10.1109/CVPR.2016.153.

[26] R. Tao, E. Gavves, A. W. M. Smeulders. Siamese in-

stance search for tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Las Vegas, USA, pp. 1420–1429, 2016. DOI: 10.1109/CVPR.2016.158.

[27] Y. Wu, J. Lim, M. H. Yang. Online object tracking: A benchmark. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Portland, USA, pp. 2411–2418, 2013. DOI: 10.1109/CVPR.2013.312.

[28] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Cehovin, G. Nebehay, G. Fernandez, T. Vojir. The VOT2013 challenge: Overview and additional results. In *Proceedings of the 19th Computer Vision Winter Workshop*, Krtiny, Czech Republic, 2014.

[29] X. Li, W. M. Hu, C. H. Shen, Z. F. Zhang, A. Dick, A. Van Den Hengel. A survey of appearance models in visual object tracking. *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, Article number 58, 2013. DOI: 10.1145/2508037.2508039.

[30] H. C. Lu, P. X. Li, D. Wang. Visual object tracking: A survey. *Pattern Recognition and Artificial Intelligence*, vol. 31, no. 1, pp. 61–76, 2018. DOI: 10.16451/j.cnki.issn1003-6059.201801006. (in Chinese)

[31] P. X. Li, D. Wang, L. J. Wang, H. C. Lu. Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, vol. 76, pp. 323–338, 2018. DOI: 10.1016/j.patcog.2017.11.007.

[32] X. Li, Y. F. Zha, T. Z. Zhang, Z. Cui, W. M. Zuo, Z. Q. Hou, H. C. Lu, H. Z. Wang. Survey of visual object tracking algorithms based on deep learning. *Journal of Image and Graphics*, vol. 24, no. 12, pp. 2057–2080, 2019. (in Chinese)

[33] A. Brunetti, D. Buongiorno, G. F. Trotta, V. Bevilacqua. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing*, vol. 300, pp. 17–33, 2018. DOI: 10.1016/j.neucom.2018.01.092.

[34] R. Yao, G. S. Lin, S. X. Xia, J. Q. Zhao, Y. Zhou. Video object segmentation and tracking: A survey. *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 4, Article number 36, 2020. DOI: 10.1145/3391743.

[35] D. Ciregan, U. Meier, J. Schmidhuber. Multi-column deep neural networks for image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Providence, USA, pp. 3642–3649, 2012. DOI: 10.1109/CVPR.2012.6248110.

[36] Z. Q. Zhao, P. Zheng, S. T. Xu, X. D. Wu. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019. DOI: 10.1109/TNNLS.2018.2876865.

[37] J. Long, E. Shelhamer, T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Boston, USA, pp. 3431–3440, 2015. DOI: 10.1109/CVPR.2015.7298965.

[38] K. Zhang, W. M. Zuo, S. H. Gu, L. Zhang. Learning deep CNN denoiser prior for image restoration. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Honolulu, USA, pp. 3929–3938, 2017. DOI: 10.1109/CVPR.2017.300.

[39] X. O. Tang, X. B. Gao, J. Z. Liu, H. J. Zhang. A spatial-temporal approach for video caption detection and recognition. *IEEE Transactions on Neural Networks*, vol. 13, no. 4, pp. 961–971, 2002. DOI: 10.1109/TNN.2002.1021896.

[40] H. Geffner. Model-free, model-based, and general intelligence. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, AAAI Press, Stockholm, Sweden, pp. 10–17, 2018. DOI: 10.24963/ijcai.2018/2.

[41] T. Elsken, J. H. Metzen, F. Hutter. Neural architecture search: A survey. [Online], Available: https://arxiv.org/abs/1808.05377, 2018.

[42] K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, USA, 2015.

[43] K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Las Vegas, USA, pp. 770–778, 2016. DOI: 10.1109/CVPR.2016.90.

[44] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Honolulu, USA, pp. 4700–4708, 2017. DOI: 10.1109/CVPR.2017.243.

[45] C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi. Inception-v4, inception-ResNet and the impact of residual connections on learning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, AAAI Press, San Francisco, USA, pp. 4278–4284, 2016.

[46] C. Szegedy, W. Liu, Y. Q. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich. Going deeper with convolutions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Boston, USA, pp. 1–9, 2015. DOI: 10.1109/CVPR.2015.7298594.

[47] C. S. Brito, W. Gerstner. Nonlinear Hebbian learning as a unifying principle in receptive field formation. *PLoS computational biology*, vol. 12, no. 9, Article number e1005070, 2016.

[48] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Las Vegas, USA, pp. 2818–2826, 2016. DOI: 10.1109/CVPR.2016.308.

[49] Z. Y. Huo, B. Gu, H. Huang. Training neural networks using features replay. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ACM, Red Hook, USA, pp. 6660–6669, 2018.

[50] J. Jeong, J. Shin. Training CNNs with selective allocation of channels. In *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, pp. 3080–3090, 2019.

[51] S. Y. Qiao, Z. S. Zhang, W. Shen, B. Wang, A. Yuille. Gradually updated neural networks for large-scale image recognition. In *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, pp. 4188–4197, 2018.

[52] K. Han, Y. H. Wang, Q. Tian, J. Y. Guo, C. J. Xu, C. Xu. GhostNet: More features from cheap operations. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 1580–1589, 2020. DOI: 10.1109/CVPR42600.2020.

00165.

[53] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, W. Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. [Online], Available: https://arxiv.org/abs/1811.12231, 2018.

[54] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Miami, USA, pp. 248–255, 2009. DOI: 10.1109/CVPR.2009.5206848.

[55] J. Frankle, M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. [Online], Available: https://arxiv.org/abs/1803.03635, 2018.

[56] J. You, J. Leskovec, K. He, S. Xie. Graph structure of neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, Vienna, Austria, pp. 10881–10891, 2020.

[57] C. H. Xie, Y. X. Wu, L. Van Der Maaten, A. L. Yuille, K. M. He. Feature denoising for improving adversarial robustness. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 501–509, 2019. DOI: 10.1109/CVPR.2019.00059.

[58] S. Kanai, Y. Fujiwara, S. Iwamura. Preventing gradient explosions in gated recurrent units. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, USA, pp. 435–444, 2017.

[59] Y. Bengio, P. Simard, P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994. DOI: 10.1109/72.279181.

[60] T. Mikolov. Statistical language models based on neural networks. *Presentation at Google*, vol. 80, Article number 26, 2012.

[61] R. Pascanu, T. Mikolov, Y. Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning*, Atlanta, USA, pp. 1310–1318, 2013.

[62] S. Hochreiter, J. Schmidhuber. Long short-term memory. *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[63] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, pp. 1724–1734, 2014. DOI: 10.3115/v1/D14-1179.

[64] R. Jozefowicz, W. Zaremba, I. Sutskever. An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, JMLR.org, Lille, France, pp. 2342–2350, 2015.

[65] G. B. Zhou, J. X. Wu, C. L. Zhang, Z. H. Zhou. Minimal gated unit for recurrent neural networks. *International Journal of Automation and Computing*, vol. 13, no. 3, pp. 226–234, 2016. DOI: 10.1007/s11633-016-1006-2.

[66] B. W. Du, H. Peng, S. Z. Wang, M. Z. A. Bhuiyan, L. H. Wang, Q. R. Gong, L. Liu, J. Li. Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction. *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 972–985, 2020. DOI: 10.1109/TITS.2019.2900481.

[67] J. Martens. Deep learning via hessian-free optimization. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ACM, Madison, USA, pp. 735–742, 2010.

[68] J. Martens, I. Sutskever. Learning recurrent neural networks with hessian-free optimization. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, Madison, USA, pp. 1033–1040, 2011.

[69] I. Sutskever, J. Martens, G. Dahl, G. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on International Conference on Machine Learning*, Atlanta, USA, pp. 1139–1147, 2013.

[70] M. Schuster, K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997. DOI: 10.1109/78.650093.

[71] M. Sundermeyer, T. Alkhouli, J. Wuebker, H. Ney. Translation modeling with bidirectional recurrent neural networks. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, pp. 14–25, 2014.

[72] R. H. Jiang, L. Zhao, T. Wang, J. X. Wang, X. Q. Zhang. Video deblurring via temporally and spatially variant recurrent neural network. *IEEE Access*, vol. 8, pp. 7587–7597, 2019. DOI: 10.1109/ACCESS.2019.2962505.

[73] S. F. Liu, J. S. Pan, M. H. Yang. Learning recursive filters for low-level vision via a hybrid neural network. In *Proceedings of the 14th European Conference on Computer Vision*, Springer, Amsterdam, The Netherlands, pp. 560–576, 2016. DOI: 10.1007/978-3-319-46493-0_34.

[74] Z. H. Li, L. N. Yao, X. Q. Zhang, X. Z. Wang, S. Kanhere, H. Z. Zhang. Zero-shot object detection with textual descriptions. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, Honolulu, USA, vol. 33, pp. 8690–8697, 2019. DOI: 10.1609/aaai.v33i01.33018690.

[75] G. Papandreou, L. C. Chen, K. Murphy, A. L. Yuille. Weakly-and semi-supervised learning of a DCNN for semantic image segmentation. [Online], Available: https://arxiv.org/abs/1502.02734, 2015.

[76] G. H. Ning, Z. Zhang, C. Huang, X. B. Ren, H. H. Wang, C. H. Cai, Z. H. He. Spatially supervised recurrent convolutional neural networks for visual object tracking. In *Proceedings of IEEE International Symposium on Circuits and Systems*, IEEE, Baltimore, USA, pp. 1–4, 2017. DOI: 10.1109/ISCAS.2017.8050867.

[77] Y. D. Chu, J. T. Fei, S. X. Hou. Adaptive global sliding-mode control for dynamic systems using double hidden layer recurrent neural network structure. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 4, pp. 1297–1309, 2020. DOI: 10.1109/TNNLS.2019.2919676.

[78] R. Wang, S. M. Pizer, J. M. Frahm. Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 5555–5564, 2019. DOI: 10.1109/CVPR.2019.00570.

[79] Z. H. Wu, S. R. Pan, F. W. Chen, G. D. Long, C. Q.

Zhang, P. S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021. DOI: 10.1109/TNNLS.2020.2978386.

[80] M. Gori, G. Monfardini, F. Scarselli. A new model for learning in graph domains. In *Proceedings of IEEE International Joint Conference on Neural Networks*, IEEE, Montreal, Canada, pp. 729–734, 2005. DOI: 10.1109/IJCNN.2005.1555942.

[81] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009. DOI: 10.1109/TNN.2008.2005605.

[82] C. Gallicchio, A. Micheli. Graph echo state networks. In *Proceedings of International Joint Conference on Neural Networks*, IEEE, Barcelona, Spain, 2010. DOI: 10.1109/IJCNN.2010.5596796.

[83] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun. Spectral networks and locally connected networks on graphs. In *Proceedings of the 2nd International Conference on Learning Representations*, Banff, Canada, 2013.

[84] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, K. M. Borgwardt. Graph kernels. *The Journal of Machine Learning Research*, vol. 11, no. 40, pp. 1201–1242, 2010.

[85] T. Gartner, P. Flach, S. Wrobel. On graph kernels: Hardness results and efficient alternatives. In *Proceedings of the 16th Annual Conference on Learning Theory and 7th Kernel Workshop on Learning Theory and Kernel Machines*, Springer, Washington, USA, pp. 129–143, 2003. DOI: 10.1007/978-3-540-45167-9_11.

[86] M. Liang, B. Yang, R. Hu, Y. Chen, R. J. Liao, S. Feng, R. Urtasun. Learning lane graph representations for motion forecasting. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 541–556, 2020. DOI: 10.1007/978-3-030-58536-5_32.

[87] H. Y. Lee, L. Jiang, I. Essa, P. B. Le, H. F. Gong, M. H. Yang, W. L. Yang. Neural design network: Graphic layout generation with constraints. [Online], Available: https://arxiv.org/abs/1912.09421, 2019.

[88] C. Q. Yu, Y. F. Liu, C. X. Gao, C. H. Shen, N. Sang. Representative graph neural network. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 379–396, 2020. DOI: 10.1007/978-3-030-58571-6_23.

[89] M. T. Luong, H. Pham, C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Lisbon, Portugal, pp. 1412–1421, 2015. DOI: 10.18653/v1/D15-1166.

[90] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, USA, pp. 5998-6008, 2017.

[91] Z. H. Dai, Z. L. Yang, Y. M. Yang, J. Carbonell, Q. V. Le, R. Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. [Online], Available: https://arxiv.org/abs/1901.02860, 2019.

[92] P. Shaw, J. Uszkoreit, A. Vaswani. Self-attention with relative position representations. [Online], Available: https://arxiv.org/abs/1803.02155, 2018.

[93] X. Q. Zhang, T. Wang, J. X. Wang, G. Y. Tang, L. Zhao. Pyramid channel-based feature attention network for image dehazing. *Computer Vision and Image Understanding*, vol. 197–198, Article number 103003, 2020. DOI: 10.1016/j.cviu.2020.103003.

[94] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, pp. 2048–2057, 2015.

[95] T. Xu, P. C. Zhang, Q. Y. Huang, H. Zhang, Z. Gan, X. L. Huang, X. D. He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 1316–1324, 2018. DOI: 10.1109/CVPR.2018.00143.

[96] H. Hu, J. Y. Gu, Z. Zhang, J. F. Dai, Y. C. Wei. Relation networks for object detection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 3588–3597, 2018. DOI: 10.1109/CVPR.2018.00378.

[97] X. L. Wang, R. Girshick, A. Gupta, K. M. He. Non-local neural networks. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 7794–7803, 2018. DOI: 10.1109/CVPR.2018.00813.

[98] X. Z. Zhu, Y. J. Wang, J. F. Dai, L. Yuan, Y. C. Wei. Flow-guided feature aggregation for video object detection. In *Proceedings of IEEE International Conference on Computer Vision*, IEEE, Venice, Italy, pp. 408–417, 2017. DOI: 10.1109/ICCV.2017.52.

[99] F. Y. Xiao, Y. J. Lee. Video object detection with an aligned spatial-temporal memory. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 485–501, 2018. DOI: 10.1007/978-3-030-01237-3_30.

[100] X. H. Jiang, L. Zhang, M. L. Xu, T. Z. Zhang, P. Lv, B. Zhou, X. Yang, Y. W. Pang. Attention scaling for crowd counting. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 4706–4715, 2020. DOI: 10.1109/CVPR42600.2020.00476.

[101] H. Zhang, I. Goodfellow, D. Metaxas, A. Odena. Self-attention generative adversarial networks. In *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, USA, pp. 7354–7363, 2019.

[102] T. Dai, J. R. Cai, Y. B. Zhang, S. T. Xia, L. Zhang. Second-order attention network for single image super-resolution. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 11065–11074, 2019. DOI: 10.1109/CVPR.2019.01132.

[103] B. Y. Chen, P. X. Li, C. Sun, D. Wang, G. Yang, H. C. Lu. Multi attention module for visual tracking. *Pattern Recognition*, vol. 87, pp. 80–93, 2019. DOI: 10.1016/j.patcog.2018.10.005.

[104] Z. X. Wu, T. Nagarajan, A. Kumar, S. Rennie, L. S. Davis, K. Grauman, R. Feris. BlockDrop: Dynamic inference paths in residual networks. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 8817–8826, 2018. DOI: 10.1109/CVPR.2018.00919.

[105] X. Wang, F. S. Yu, Z. Y. Dou, T. Darrell, J. E. Gonzalez. SkipNet: Learning dynamic routing in convolutional networks. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 409–424, 2018. DOI: 10.1007/978-3-030-01261-8_25.

[106] N. Shazeer, K. Fatahalian, W. R. Mark, R. T. Mullapudi. HydraNets: Specialized dynamic architectures for efficient inference. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 8080–8089, 2018. DOI: 10.1109/CVPR.2018.00843.

[107] G. Huang, D. L. Chen, T. H. Li, F. Wu, L. Van Der Maaten, K. Q. Weinberger. Multi-scale dense networks for resource efficient image classification. [Online], Available: https://arxiv.org/abs/1703.09844, 2017.

[108] Z. H. You, K. Yan, J. M. Ye, M. Ma, P. Wang. Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, Vancouver, Canada, pp. 2133–2144, 2019.

[109] Y. W. Li, L. Song, Y. K. Chen, Z. M. Li, X. Y. Zhang, X. G. Wang, J. Sun. Learning dynamic routing for semantic segmentation. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 8553–8562, 2020. DOI: 10.1109/CVPR42600.2020.00858.

[110] B. De Brabandere, X. Jia, T. Tuytelaars, L. Van Gool. Dynamic filter networks. In *Proceedings of the 30th Conference on Neural Information Processing Systems*, Barcelona, Spain, pp. 667–675, 2016.

[111] S. Niklaus, L. Mai, F. Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of IEEE International Conference on Computer Vision*, IEEE, Venice, Italy, pp. 261–270, 2017. DOI: 10.1109/ICCV.2017.37.

[112] Y. Jo, S. W. Oh, J. Kang, S. J. Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 3224–3232, 2018. DOI: 10.1109/CVPR.2018.00340.

[113] X. Y. Xu, M. C. Li, W. X. Sun. Learning deformable kernels for image and video denoising. [Online], Available: https://arxiv.org/abs/1904.06903, 2019.

[114] B. Mildenhall, J. T. Barron, J. W. Chen, D. Sharlet, R. Ng, R. Carroll. Burst denoising with kernel prediction networks. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 2502–2510, 2018. DOI: 10.1109/CVPR.2018.00265.

[115] Y. S. Xu, S. Y. R. Tseng, Y. Tseng, H. K. Kuo, Y. M. Tsai. Unified dynamic convolutional network for super-resolution with variational degradations. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 12496–12505, 2020. DOI: 10.1109/CVPR42600.2020.01251.

[116] J. F. Dai, H. Z. Qi, Y. W. Xiong, Y. Li, G. D. Zhang, H. Hu, Y. C. Wei. Deformable convolutional networks. In *Proceedings of IEEE International Conference on Computer Vision*, IEEE, Venice, Italy, pp. 764–773, 2017. DOI: 10.1109/ICCV.2017.89.

[117] X. Z. Zhu, H. Hu, S. Lin, J. F. Dai. Deformable ConvNets V2: More deformable, better results. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 9308–9316, 2019. DOI: 10.1109/CVPR.2019.00953.

[118] C. F. Xu, B. C. Wu, Z. N. Wang, W. Zhan, P. Vajda, K. Keutzer, M. Tomizuka. SqueezeSegV3: Spatially-adaptive convolution for efficient point-cloud segmentation. [Online], Available: https://arxiv.org/abs/2004.01803, 2020.

[119] Z. T. Xiong, Y. Yuan, N. H. Guo, Q. Wang. Variational context-deformable convnets for indoor scene parsing. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 3992–4002, 2020. DOI: 10.1109/CVPR42600.2020.00405.

[120] W. Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997. DOI: 10.1016/S0893-6080(97)00011-7.

[121] S. Kim, S. Park, B. Na, S. Yoon. Spiking-YOLO: Spiking neural network for energy-efficient object detection. [Online], Available: https://arxiv.org/abs/1903.06530, 2019.

[122] Z. F. Mainen, T. J. Sejnowski. Reliability of spike timing in neocortical neurons. *Science*, vol. 268, no. 5216, pp. 1503–1506, 1995. DOI: 10.1126/science.7770778.

[123] G. Bellec, D. Salaj, A. Subramoney, R. Legenstein, W. Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Curran Associates Inc., Red Hook, USA, pp. 787–797, 2018.

[124] Y. Q. Cao, Y. Chen, D. Khosla. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, vol. 113, no. 1, pp. 54–66, 2015. DOI: 10.1007/s11263-014-0788-3.

[125] D. Comaniciu, V. Ramesh, P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Hilton Head Island, USA, pp. 142–149, 2000. DOI: 10.1109/CVPR.2000.854761.

[126] N. Dalal, B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, San Diego, USA, pp. 886–893, 2005. DOI: 10.1109/CVPR.2005.177.

[127] J. F. Henriques, R. Caseiro, P. Martins, J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *Proceedings of the 12th European Conference on Computer Vision*, Springer, Florence, Italy, pp. 702–715, 2012. DOI: 10.1007/978-3-642-33765-9_50.

[128] J. F. Henriques, R. Caseiro, P. Martins, J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015. DOI: 10.1109/TPAMI.2014.2345390.

[129] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, P. H. S. Torr. Staple: Complementary learners for real-time tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Las Vegas, USA, pp. 1401–1409, 2016. DOI: 10.1109/CVPR.2016.156.

[130] C. Ma, J. B. Huang, X. K. Yang, M. H. Yang. Hierarchical convolutional features for visual tracking. In *Proceedings of IEEE International Conference on Computer Vision*, IEEE, Santiago, Chile, pp. 3074–3082, 2015. DOI:

10.1109/ICCV.2015.352.

[131] Y. K. Qi, S. P. Zhang, L. Qin, H. X. Yao, Q. M. Huang, J. Lim, M. H. Yang. Hedged deep tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Las Vegas, USA, pp. 4303–4311, 2016. DOI: 10.1109/CVPR.2016.466.

[132] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, P. H. S. Torr. Fully-convolutional Siamese networks for object tracking. In *Proceedings of European Conference on Computer Vision*, Springer, Amsterdam, The Netherlands, pp. 850–865, 2016. DOI: 10.1007/978-3-319-48881-3_56.

[133] B. Li, W. Wu, Q. Wang, F. Y. Zhang, J. L. Xing, J. J. Yan. SiamRPN++: Evolution of Siamese visual tracking with very deep networks. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 4282–4291, 2019. DOI: 10.1109/CVPR.2019.00441.

[134] A. Lukezic, J. Matas, M. Kristan. D3S-A discriminative single shot segmentation tracker. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 7133–7142, 2020. DOI: 10.1109/CVPR42600.2020.00716.

[135] M. Y. Wu, H. B. Ling, N. Bi, S. H. Gao, Q. Hu, H. Sheng, J. Y. Yu. Visual tracking with multiview trajectory prediction. *IEEE Transactions on Image Processing*, vol. 29, pp. 8355–8367, 2020. DOI: 10.1109/TIP.2020.3014952.

[136] Z. D. Chen, B. N. Zhong, G. R. Li, S. P. Zhang, R. R. Ji. Siamese box adaptive network for visual tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 6668–6677, 2020. DOI: 10.1109/CVPR42600.2020.00670.

[137] Z. P. Zhang, H. W. Peng. Deeper and wider Siamese networks for real-time visual tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 4591–4600, 2019. DOI: 10.1109/CVPR.2019.00472.

[138] L. Y. Zheng, Y. Y. Chen, M. Tang, J. Q. Wang, H. Q. Lu. Siamese deformable cross-correlation network for real-time visual tracking. *Neurocomputing*, vol. 401, pp. 36–47, 2020. DOI: 10.1016/j.neucom.2020.02.080.

[139] Y. C. Yu, Y. L. Xiong, W. L. Huang, M. R. Scott. Deformable Siamese attention networks for visual object tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 6728–6737, 2020. DOI: 10.1109/CVPR42600.2020.00676.

[140] Z. Teng, J. L. Xing, Q. Wang, B. P. Zhang, J. P. Fan. Deep spatial and temporal network for robust visual object tracking. *IEEE Transactions on Image Processing*, vol. 29, pp. 1762–1775, 2019. DOI: 10.1109/TIP.2019.2942502.

[141] M. H. Abdelpakey, M. S. Shehata. DP-Siam: Dynamic policy Siamese network for robust object tracking. *IEEE Transactions on Image Processing*, vol. 29, pp. 1479–1492, 2019. DOI: 10.1109/TIP.2019.2942506.

[142] P. X. Li, B. Y. Chen, W. L. Ouyang, D. Wang, X. Y. Yang, H. C. Lu. GradNet: Gradient-guided network for visual object tracking. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Korea, pp. 6162–6171, 2019. DOI: 10.1109/ICCV.2019.00626.

[143] D. Y. Guo, J. Wang, Y. Cui, Z. H. Wang, S. Y. Chen. SiamCAR: Siamese fully convolutional classification and regression for visual tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 6269–6277, 2020. DOI: 10.1109/CVPR42600.2020.00630.

[144] Q. Wang, L. Zhang, L. Bertinetto, W. M. Hu, P. H. S. Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 1328–1338, 2019. DOI: 10.1109/CVPR.2019.00142.

[145] H. Fan, H. B. Ling. Siamese cascaded region proposal networks for real-time visual tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 7952–7961, 2019. DOI: 10.1109/CVPR.2019.00814.

[146] G. Bhat, J. Johnander, M. Danelljan, F. S. Khan, M. Felsberg. Unveiling the power of deep tracking. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 483–498, 2018. DOI: 10.1007/978-3-030-01216-8_30.

[147] S. M. Ge, Z. Luo, C. H. Zhang, Y. Y. Hua, D. C. Tao. Distilling channels for efficient deep tracking. *IEEE Transactions on Image Processing*, vol. 29, pp. 2610–2621, 2019. DOI: 10.1109/TIP.2019.2950508.

[148] Y. D. Xu, Z. Y. Wang, Z. X. Li, Y. Ye, G. Yu. SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, pp. 12549–12556, 2020. DOI: 10.1609/aaai.v34i07.6944.

[149] Y. Yang, G. Li, Y. Qi, Q. Huang. Release the power of online-training for robust visual tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, pp. 12645-12652, 2020. DOI: 10.1609/aaai.v34i07.6956.

[150] J. H. Zhou, P. Wang, H. Y. Sun. Discriminative and robust online learning for Siamese visual tracking. In *Proceedings of AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, AIAA, pp. 13017–13024, 2020.

[151] J. Choi, H. J. Chang, T. Fischer, S. Yun, K. Lee, J. Jeong, Y. Demiris, J. Y. Choi. Context-aware deep feature compression for high-speed visual tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 479–488, 2018. DOI: 10.1109/CVPR.2018.00057.

[152] B. Li, J. J. Yan, W. Wu, Z. Zhu, X. L. Hu. High performance visual tracking with Siamese region proposal network. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 8971–8980, 2018. DOI: 10.1109/CVPR.2018.00935.

[153] Y. B. Song, C. Ma, X. H. Wu, L. J. Gong, L. C. Bao, W. M. Zuo, C. H. Shen, R. W. H. Lau, M. H. Yang. Vital: VIsual tracking via adversarial learning. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 8990–8999, 2018. DOI: 10.1109/CVPR.2018.00937.

[154] Z. H. Lai, E. Lu, W. D. Xie. MAST: A memory-augmented self-supervised tracker. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 6479–6488 2020. DOI: 10.1109/CVPR42600.2020.00651.

[155] E. Park, A. C. Berg. Meta-tracker: Fast and robust online adaptation for visual object trackers. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 569–585, 2018. DOI: 10.1007/978-3-030-01219-9_35.

[156] X. P. Dong, J. B. Shen, L. Shao, F. Porikli. CLNet: A compact latent network for fast adjusting Siamese trackers. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 378–395, 2020. DOI: 10.1007/978-3-030-58565-5_23.

[157] Z. P. Zhang, H. W. Peng, J. L. Fu, B. Li, W. M. Hu. Ocean: Object-aware anchor-free tracking. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 771–787, 2020. DOI: 10.1007/978-3-030-58589-1_46.

[158] Y. Liu, R. T. Li, Y. Cheng, R. T. Tan, X. B. Sui. Object tracking using spatio-temporal networks for future prediction location. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 1–17, 2020. DOI: 10.1007/978-3-030-58542-6_1.

[159] B. Y. Liao, C. Y. Wang, Y. Y. Wang, Y. N. Wang, J. Yin. PG-Net: Pixel to global matching network for visual tracking. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 429–444, 2020. DOI: 10.1007/978-3-030-58542-6_26.

[160] L. H. Huang, X. Zhao, K. Q. Huang. Bridging the gap between detection and tracking: A unified approach. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Korea, pp. 3999–4009, 2019. DOI: 10.1109/ICCV.2019.00410.

[161] W. X. Liu, Y. B. Song, D. S. Chen, S. F. He, Y. L. Yu, T. Yan, G. P. Hancke, R. W. H. Lau. Deformable object tracking with gated fusion. *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 3766–3777, 2019. DOI: 10.1109/TIP.2019.2902784.

[162] Z. Y. Liang, J. B. Shen. Local semantic Siamese networks for fast tracking. *IEEE Transactions on Image Processing*, vol. 29, pp. 3351–3364, 2019. DOI: 10.1109/TIP.2019.2959256.

[163] A. F. He, C. Luo, X. M. Tian, W. J. Zeng. A twofold Siamese network for real-time object tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 4834–4843, 2018. DOI: 10.1109/CVPR.2018.00508.

[164] J. Y. Gao, T. Z. Zhang, C. S. Xu. Graph convolutional tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 4649–4659, 2019. DOI: 10.1109/CVPR.2019.00478.

[165] Y. H. Zhang, L. J. Wang, J. Q. Qi, D. Wang, M. Y. Feng, H. C. Lu. Structured Siamese network for real-time visual tracking. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 351–366, 2018. DOI: 10.1007/978-3-030-01240-3_22.

[166] K. P. Li, Y. Kong, Y. Fu. Visual object tracking via multi-stream deep similarity learning networks. *IEEE Transactions on Image Processing*, vol. 29, pp. 3311–3320, 2019. DOI: 10.1109/TIP.2019.2959249.

[167] G. T. Wang, C. Luo, X. Y. Sun, Z. W. Xiong, W. J. Zeng. Tracking by instance detection: A meta-learning approach. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 6288–6297, 2020. DOI: 10.1109/CVPR42600.2020.00632.

[168] T. Y. Yang, P. F. Xu, R. B. Hu, H. Chai, A. B. Chan. ROAM: Recurrently optimizing tracking model. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 6718–6727, 2020. DOI: 10.1109/CVPR42600.2020.00675.

[169] X. K. Lu, C. Ma, B. B. Ni, X. K. Yang, I. Reid, M. H. Yang. Deep regression tracking with shrinkage loss. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 353–369, 2018. DOI: 10.1007/978-3-030-01264-9_22.

[170] H. Z. Zhou, B. Ummenhofer, T. Brox. DeepTAM: Deep tracking and mapping. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 822–838, 2018. DOI: 10.1007/978-3-030-01270-0_50.

[171] X. Y. Zhou, V. Koltun, P. Krahenbuhl. Tracking objects as points. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 474–490, 2020. DOI: 10.1007/978-3-030-58548-8_28.

[172] Y. Sui, Z. M. Zhang, G. H. Wang, Y. F. Tang, L. Zhang. Exploiting the anisotropy of correlation filter learning for visual tracking. *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1084–1105, 2019. DOI: 10.1007/s11263-019-01156-6.

[173] H. Z. Zhou, B. Ummenhofer, T. Brox. DeepTAM: Deep tracking and mapping with convolutional neural networks. *International Journal of Computer Vision*, vol. 128, no. 3, pp. 756–769, 2020. DOI: 10.1007/s11263-019-01221-0.

[174] L. H. Huang, X. Zhao, K. Q. Huang. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. DOI: 10.1109/TPAMI.2019.2957464.

[175] H. Fan, L. T. Lin, F. Yang, P. Chu, G. Deng, S. J. Yu, H. X. Bai, Y. Xu, C. Y. Liao, H. B. Ling. LaSOT: A high-quality benchmark for large-scale single object tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 5374–5383, 2019. DOI: 10.1109/CVPR.2019.00552.

[176] M. Kristan, R. Pflugfelder, A. Leonardis, et al. The visual object tracking VOT2014 challenge results. In *Proceedings of the European Conference on Computer Vision*, Springer,Zurich,Switzerland,pp. 191–217,2014.DOI:10.1007/978-3-319-16181-5_14.

[177] Y. Wu, J. Lim, M. H. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015. DOI: 10.1109/TPAMI.2014.2388226.

[178] M. Kristan, A. Leonardis, J. Matas, et al. The visual object tracking VOT2017 challenge results. In *Proceedings of IEEE International Conference on Computer Vision Workshops*, IEEE, Venice, Italy, pp. 1949–1972, 2017. DOI: 10.1109/ICCVW.2017.230.

[179] M. Kristan, J. Matas, A. Leonardis, et al. The seventh visual object tracking VOT2019 challenge results. In *Proceedings of IEEE/CVF International Conference on Computer Vision Workshop*, IEEE, Seoul, Korea, pp. 2206–2241, 2019. DOI: 10.1109/ICCVW.2019.00276.

[180] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis*

and Machine Intelligence, vol. 36, no. 7, pp. 1442–1468, 2014. DOI: 10.1109/TPAMI.2013.230.

[181] A. N. Li, M. Lin, Y. Wu, M. H. Yang, S. C. Yan. NUS-PRO: A new visual tracking challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 335–349, 2015. DOI: 10.1109/TPAMI.2015.2417577.

[182] P. P. Liang, E. Blasch, H. B. Ling. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5630–5644, 2015. DOI: 10.1109/TIP.2015.2482905.

[183] H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, S. Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *Proceedings of IEEE International Conference on Computer Vision*, IEEE, Venice, Italy, pp. 1125−1134, 2017. DOI: 10.1109/ICCV.2017.128.

[184] M. Mueller, N. Smith, B. Ghanem. A benchmark and simulator for UAV tracking. In *Proceedings of 14th European Conference on Computer Vision*, Springer, Amsterdam, The Netherlands, pp. 445−461, 2016. DOI: 10.1007/978-3-319-46448-0_27.

[185] J. Valmadre, L. Bertinetto, J. F. Henriques, R. Tao, A. Vedaldi, A. W. M. Smeulders, P. H. S. Torr, E. Gavves. Long-term tracking in the wild: A benchmark. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 670−685, 2018. DOI: 10.1007/978-3-030-01219-9_41.

[186] M. Muller, A. Bibi, S. Giancola, S. Alsubaihi, B. Ghanem. TrackingNet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 300−317, 2018. DOI: 10.1007/978-3-030-01246-5_19.

[187] L. Leal-Taixe, A. Milan, I. Reid, S. Roth, K. Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. [Online], Available: https://arxiv.org/abs/1504.01942, 2015.

[188] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, K. Schindler. MOT16: A benchmark for multi-object tracking. [Online], Available: https://arxiv.org/abs/1603.00831, 2016.

[189] A. Geiger, P. Lenz, R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Providence, USA, pp. 3354−3361, 2012. DOI: 10.1109/CVPR.2012.6248074.

[190] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. H. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015. DOI: 10.1007/s11263-015-0816-y.

[191] E. Real, J. Shlens, S. Mazzocchi, X. Pan, V. Vanhoucke. YouTube-boundingBoxes: A large high-precision human-annotated data set for object detection in video. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Honolulu, USA, pp. 5296−5305, 2017. DOI: 10.1109/CVPR.2017.789.

[192] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, R. Pflugfelder, A. Gupta, A. Bibi, A. Lukezic, A. Garcia-Martin, A. Saffari, A. Petrosino, A. S. Montero. The visual object tracking VOT2015 challenge results. In *Proceedings of IEEE International Conference on Computer Vision Workshop*, IEEE, Santiago, Chile, pp. 1−23, 2015. DOI: 10.1109/ICCVW.2015.79.

[193] S. Had, R. Bowden, K. Lebeda. The visual object tracking VOT2016 challenge results. *Lecture Notes in Computer Science*, vol. 9914, pp. 777–823, 2016.

[194] M. Kristan, A. Leonardis, J. Matas, et al. The sixth visual object tracking VOT2018 challenge results. In *Proceedings of the European Conference on Computer Vision*, Springer, Munich, Germany, pp. 3−53, 2018. DOI: 10.1007/978-3-030-11009-3_1.

[195] G. A. Miller. *WordNet: An Electronic Lexical Database*. Cambridge, USA: MIT Press, 1998.

[196] X. Li, C. Ma, B. Y. Wu, Z. Y. He, M. H. Yang. Target-aware deep tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, CA, USA, pp. 1369−1378, 2019. DOI: 10.1109/CVPR.2019.00146.

[197] T. Y. Yang, A. B. Chan. Learning dynamic memory networks for object tracking. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 152−167, 2018. DOI: 10.1007/978-3-030-01240-3_10.

[198] X. P. Dong, J. B. Shen. Triplet loss in Siamese network for object tracking. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 459−474, 2018. DOI: 10.1007/978-3-030-01261-8_28.

[199] Y. Sui, Y. F. Tang, L. Zhang, G. H. Wang. Visual tracking via subspace learning: A discriminative approach. *International Journal of Computer Vision*, vol. 126, no. 5, pp. 515–536, 2018. DOI: 10.1007/s11263-017-1049-z.

[200] C. Ma, J. B. Huang, X. K. Yang, M. H. Yang. Robust visual tracking via hierarchical convolutional features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2709–2723, 2019. DOI: 10.1109/TPAMI.2018.2865311.

[201] N. Wang, Y. B. Song, C. Ma, W. G. Zhou, W. Liu, H. Q. Li. Unsupervised deep tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 1308−1317, 2019. DOI: 10.1109/CVPR.2019.00140.

[202] P. Voigtlaender, J. Luiten, P. H. S. Torr, B. Leibe. Siam R-CNN: Visual tracking by re-detection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 6578−6588, 2020. DOI: 10.1109/CVPR42600.2020.00661.

[203] Z. Zhu, Q. Wang, B. Li, W. Wu, J. J. Yan, W. M. Hu. Distractor-aware Siamese networks for visual object tracking. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 101−117, 2018. DOI: 10.1007/978-3-030-01240-3_7.

[204] L. L. Ren, X. Yuan, J. W. Lu, M. Yang, J. Zhou. Deep reinforcement learning with iterative shift for visual tracking. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 684−700, 2018. DOI: 10.1007/978-3-030-01240-3_42.

[205] H. Fan, H. B. Ling. Parallel tracking and verifying. *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 4130–4144, 2019. DOI: 10.1109/TIP.2019.2904789.

[206] G. Bhat, M. Danelljan, L. Van Gool, R. Timofte. Learning discriminative model prediction for tracking. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Korea, pp. 6182−6191,

2019. DOI: 10.1109/ICCV.2019.00628.

[207] K. N. Dai, Y. H. Zhang, D. Wang, J. H. Li, H. C. Lu, X. Y. Yang. High-performance long-term tracking with meta-updater. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, WA, USA, pp. 6298–6307, 2020. DOI: 10.1109/CVPR42600.2020.00633.

[208] L. H. Huang, X. Zhao, K. Q. Huang. GlobalTrack: A simple and strong baseline for long-term tracking. [Online], Available: https://arxiv.org/abs/1912.08531, 2019.

[209] M. Danelljan, G. Bhat, F. S. Khan, M. Felsberg. ATOM: Accurate tracking by overlap maximization. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, CA, USA, pp. 4660–4669, 2019. DOI: 10.1109/CVPR.2019.00479.

[210] J. F. Han, P. Luo, X. G. Wang. Deep self-learning from noisy labels. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Korea, pp. 5138–5147, 2019. DOI: 10.1109/ICCV.2019.00524.

[211] F. Q. Liu, Z. Y. Automatic "ground truth" annotation and industrial workpiece dataset generation for deep learning. *International Journal of Automation and Computing*, vol. 17, no. 4, pp. 539–550, 2020. DOI: 10.1007/s11633-020-1221-8.

[212] Q. Fu, X. Y. Chen, W. A survey on 3D visual tracking of multicopters. *International Journal of Automation and Computing*, vol. 16, no. 6, pp. 707–719, 2019. DOI: 10.1007/s11633-019-1199-2.

Xiao-Qin Zhang received the B. Sc. degree in electronic information science and technology from Central South University, China in 2005, and Ph. D. degree in pattern recognition and intelligent system from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China in 2010. He is currently a professor in Wenzhou University, China. He has published more than 80 papers in international and national journals, and international conferences, including IEEE T-PAMI, IJCV, IEEE T-IP, IEEE T-IE, IEEE T-C, ICCV, CVPR, NIPS, IJCAI, AAAI, etc.

His research interests include pattern recognition, computer vision and machine learning.

E-mail: zhangxiaoqinnan@gmail.com (Corresponding author)

ORCID iD: 0000-0003-0958-7285

Run-Hua Jiang received the B. Sc. degree in network engineering from Department of Information Science, Tianjin University of Finance and Economy, China in 2017. He is currently a graduate student in computer software and theory at College of Computer Science and Artificial Intelligence, Wenzhou University, China.

His research interests include several computer vision tasks, such as image/video restoration, crowd counting, visual understanding, and video question answering.

E-mail: ddghjikle1@gmail.com

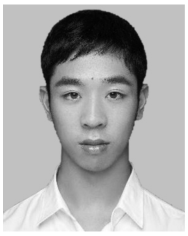
ORCID iD: 0000-0003-2402-8684

Chen-Xiang Fan received the B. Sc. degree in information and computing science from Department of Information and Computing Science, Ludong University, China in 2020. He is currently a graduate student majoring in computer software and theory at College of Computer Science and Artificial Intelligence, Wenzhou University, China.

His research interests include machine learning, recommendation system and object tracking.

E-mail: fanchenx@163.com

ORCID iD: 0000-0002-8793-1726



Tian-Yu Tong is currently an undergraduate student in data science and big data technology at College of Computer Science and Artificial Intelligence, Wenzhou University, China.

His research interests include big data technology, pattern recognition and machine learning.

E-mail: joker_tongtianyu@163.com

ORCID iD: 0000-0001-7045-8867

Tao Wang received the B. Sc. degree in information and computing science from Hainan Normal University, China in 2018. He is currently a graduate student at College of Computer Science and Artificial Intelligence, Wenzhou University, China.

His research interests include several topics in computer vision, such as image/video quality restoration, adversarial learning, visual tracking, image-to-image translation, reinforcement learning.

E-mail: taowangzj@gmail.com

ORCID iD: 0000-0002-0202-0174

Peng-Cheng Huang received the B. Sc. degree in electrical engineering and automation from Department of Modern Science and Technology, China Metrology University, China in 2018. He is currently a graduate student in computer software and theory at College of Computer Science and Artificial Intelligence, Wenzhou University, China.

His research interests include image and video processing, pattern recognition and machine learning.

E-mail: fshhppcc@gmail.com

ORCID iD: 0000-0003-3695-2022