Information Sciences 597 (2022) 105–124

Contents lists available at ScienceDirect

Information Sciences

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

Residual memory inference network for regression tracking with weighted gradient harmonized loss

Huanlong Zhang [a], Jiapeng Zhang [a], Guohao Nie [a], Jilin Hu [b,*], W.J. (Chris)Zhang [c]

[a] College of Electrical Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China
[b] Department of Computer Science, Aalborg University, Aalborg 9220, Denmark
[c] Department of Mechanical Engineering and Division of Biomedical Engineering, University of Saskatchewan, Saskatoon, Canada

ARTICLE INFO

Article history:
Received 7 October 2021
Received in revised form 7 March 2022
Accepted 8 March 2022
Available online 12 March 2022

Keywords:
Long-short term memory
Residual network
Visual tracking

ABSTRACT

Recently, the memory mechanism has been widely implemented in target tracking. However, these trackers hardly balance the stability of long-term memory with the plasticity of short-term memory through an elegant and efficient mechanism. A residual memory inference network (RMIT) is proposed to exploit the history of target states and last visual features. Specifically, RMIT consists of a base layer and a residual memory layer by synergizing short-and long-term memories. The base layer can be regarded as Discriminative Correlation Filter (DCF) reformulation that maintains the short-term memory to accommodate rapid appearance changes. The residual memory layer can extend residual learning from the spatial domain to the Spatio-temporal domain via ConvLSTM to obtain long-term memory of the target appearance. To avoid model degradation due to sample imbalance, we introduce a weighted gradient harmonized loss to improve the discrimination of the tracker. Then, response scores can be served as a basis of the adaptive learning strategy to ensure the reliability of memory updates. The proposed method performs favorably and has been extensively validated on six benchmark datasets, including OTB-50/100, TC-128, UAV-123, and VOT-2016/2018 against several advanced methods.

1. Introduction

Object tracking has attracted more and more attention in the computer vision community recently [5]. It has been applied in intelligent monitoring, human-computer interaction, unmanned driving, and other fields. However, object tracking is still easy to fail due to the drastic appearance changes, such as target disappearance, occlusion, and violent deformation.

Recently, the memory mechanism has been successfully applied in object tracking to address appearance variation [12,26,34,48,49], which can be classified into short-and long-term memories. Short-term memory pays attention to the current state of the target, which is an essential feature to adapt to changes in appearance. For example, [12] proposes a short-term memory filter to handle the appearance variations for the target appearance, which is achieved by moving average algorithm with a high learning rate. Long-term memory contains the historical states of the target, which is robust for a tracker to deal with the violent deformation. For example, Discriminative Correlation Filters (DCFs) based methods [26,34] propose a correlation filter to maintain the long-term memory of the target, combined with a low learning rate. When the baseline tracker is unreliable, the conservative correlation filter can recover the target. Moreover, the online

* Corresponding author.
E-mail addresses: zzuli407@163.com (H. Zhang), hujilin1229@gmail.com (J. Hu).

https://doi.org/10.1016/j.ins.2022.03.047
0020-0255/© 2022 Elsevier Inc. All rights reserved.

trained classifiers can exploit reliable long-term memory for re-detection. Specifically, siamese-based approaches [48,49] use long-term memory to update the encoded target feature in the memory space.

Due to the different contributions of these two memories, many works have been proposed to construct an object tracker by coordinating them. Ma et al. [34] propose two different kinds of memories by using two correlation filters with different learning rates. Li et al. [26] propose a memory selection mechanism to allow the tracker to benefit from different memory patterns in different situations. MUSTer [20] builds a key point feature database to store the long-term memory, such that it can control the state of short-term memory and the final output. Fig. 1.

However, these algorithms are limited by the following aspects. Firstly, maintaining an additional long-term model is expensive. The two memory patterns come from different modules, so they benefit little from the less-used long-term memory [26,34]. However, the size of memory space expands over time [48,49], which increases the computational expenses and makes it difficult for the tracker to select the best target template from memory. Secondly, most trackers only utilize redundancy to address the accumulating error in the tracking loop. Although such an approach can improve results, a tracker can hardly have additional gain from the temporal information. Moreover, the complex memory selection mechanism still risks making the Tracking fail since the tracker always trusts the results from the re-detection. These limitations raise two questions: (1) whether temporal information in long-term memory can be used more effectively, such that it can compensate for short-term memory; and (2) whether these two memory models can be learned and applied adaptively in an end-to-end manner, rather than using a complex selection mechanism?

We propose an end-to-end regression tracking network, namely Residual Memory Inference network (RMIT) to address these two questions. It can provide a more affluent target appearance representation by simultaneously considering dual

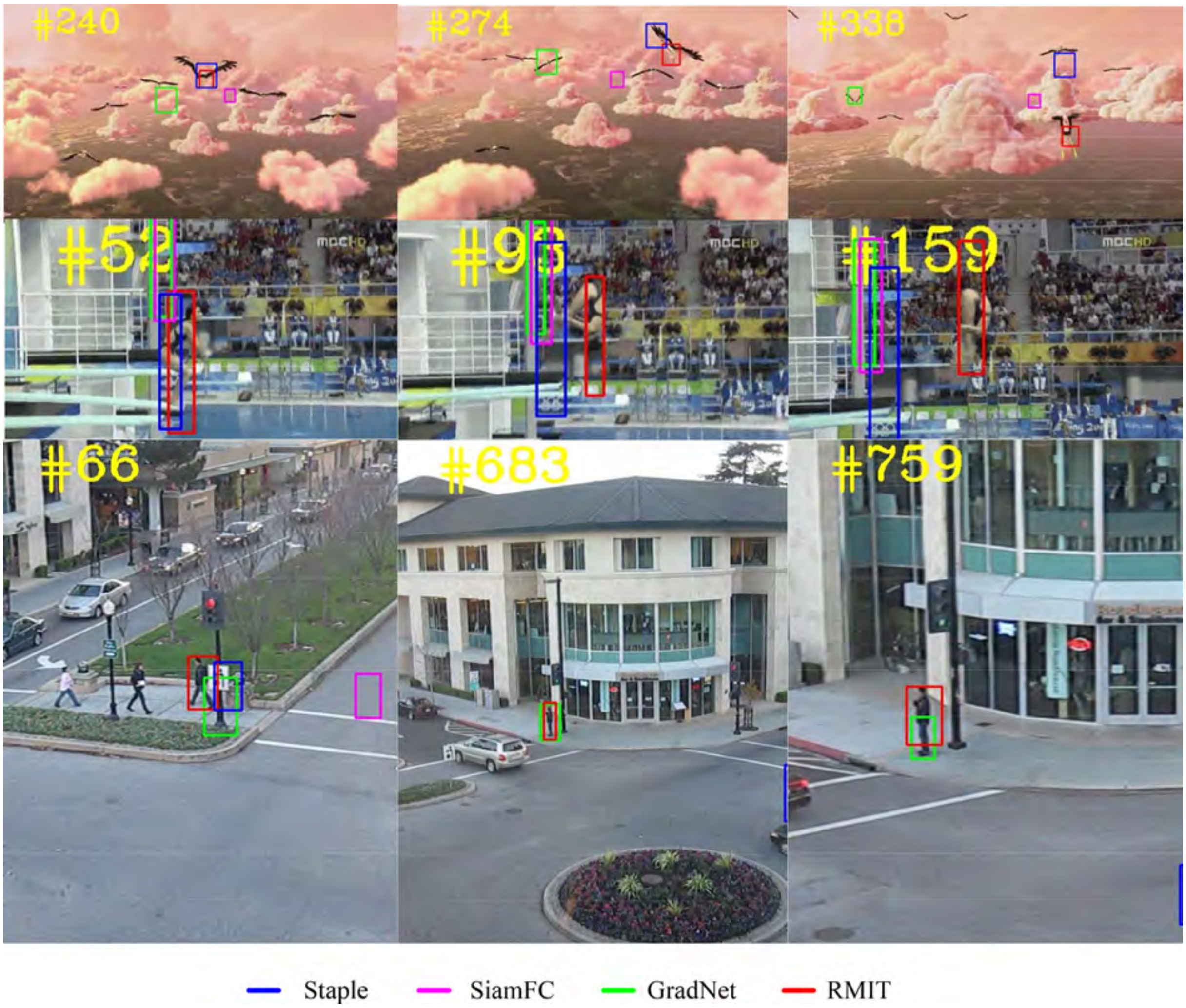


Fig. 1. Compared our proposed tracker with other state-of-the-art methods (Staple [4], SiamFC [3], and GradNet [28]) on several video sequences (Bird1, Diving, Human3) with object deformation and scale variation. And our algorithm efficiently handles these challenging situations compared to other approaches.

memory modes to improve the adaptability of visual Tracking. We first propose a base layer to maintain the short-term memory through a single-layer convolution network derived from a DCF equivalent network. Then, we propose a novel residual memory layer to learn the long-term memory of the target appearance, capturing the spatial variations of the target by learning the residual to complement the base layer. Moreover, we introduce a ConvLSTM into the residual memory layer to allow for the spatial information to propagate over time to store the long-term memory of the target appearance. Instead of using a complex memory selection mechanism on different tracking scenarios [34], we use residual learning to exploit the characteristics of two memory modes simultaneously. Therefore, RMIT can coordinate historical and recent states to infer the current target states, which is achieved by an optimal response map of different memories via two branches. To properly update our model and prevent the model from learning with incorrect features, e.g., occlusion, we employ a weighted gradient harmonized loss to improve the discriminative of the regression network by balancing the positive and negative samples. Finally, RMIT is integrated with an adaptive learning strategy to avoid updating memories with incorrect features and utilize more reliable response scores. The weighted gradient harmonized loss accelerates network convergence as well.

The main contributions of this work can be summarized as follows.

- We first propose an end-to-end residual memory inference network for deep regression tracking. More specifically, we construct a residual learning module to infer the current target state by considering both long and short memories of the target appearance.
- We then introduce a new weighted gradient coordinated loss algorithm to solve the data imbalance problem in regression network training. The proposed loss function can effectively improve the network's convergence efficiency and tracking accuracy.
- Finally, we validate the proposed method extensively on six benchmark datasets, including OTB-50/100, TC-128, UAV-123, and VOT-2016/2018. The results show that our RMIT tracker performs advantageously against other state-of-the-art trackers.

2. Related work

This section covers the discussion of tracking methods based on regression learning, target memory, and deep learning, respectively.

2.1. Tracking by regression learning

Regression learning tracker is to learn a mapping function from regularly dense samples of target objects to soft labels [32]. Discriminative correlation filters (DCFs) build an efficient tracker by learning a ridge regression classifier. Some works have been proposed for Tracking by combining multiple sources of information. Considering the color attributes have a good performance when confronting some challenges such as background clutter and improving the discriminant ability of the tracker, Danelljan et al. [13] propose a more robust tracker by appending color attributes of the target. However, it cannot provide sufficient information for the tracker to deal with fast motion and scale change challenges. Henriques et al. [18] propose another correlation filter for object tracking by exploiting the Histogram of Oriented Gradients (HOG) feature and the circulant structure of training samples. The matrix operation in Fourier space is transformed into the Hadamard product of vectors in this work, which significantly reduces the number of operations and increases computation speed. However, since the target box is already set in this algorithm, the tracker drifts during the tracking process when the tracking target size changes, leading to the tracking failure. In addition to these techniques, many other methods are proposed to improve the tracker's performance further, e.g., spatial regularization [10], scale estimation [12], and multiple dimensional features [18,50].

However, all the methods mentioned above are only to learn DCFs for the tracker, independent of the feature extraction. To enable the regression tracker to be trained in an end-to-end manner, Chen et al. [7] formulate the correlation filter as a convolution operation in the spatial domain. Song et al. [42] propose residual learning to address the drastic appearance changes of the target, which is the main reason for the performance degradation of the tracker. Lu et al. [32] propose a shrinkage loss to address the problem of data imbalance in training regression networks, which can further exploit multi-level semantic abstraction. Unlike these deep regression trackers, RMIT focuses on exploring the rich stream information in continuous frames to improve tracking accuracy, formulated as an end-to-end framework for prediction and optimization. Moreover, we introduce a weighted gradient coordination factor into the L2 loss function, which can be adjusted adaptively to the imbalanced sample instead of a fixed distribution [7,32].

2.2. Tracking by target memory

Memory-based trackers are introduced to adapt to appearance changes, including long-and short-term memories. Most trackers are constructed with well-designed criteria that benefit from different memory patterns. Li et al. [26] propose a dual-memory selection model to choose applicable memory modes in different scenarios. Xue et al. [46] propose a metric strategy to select memory states by measuring the motion distance between two frames that coincides with the poses. However, the tracking performance of these algorithms relies heavily on the design criteria, which can whether be accurately judged or not. Li et al. [27] propose a bottom-up and top-down integration tracking framework by concatenating both short-and long-term memories, avoiding making a selection between them. Thus, it can handle unknown objects' structural and appearance variations online. Nevertheless, the short-and long-term memory in cascade form does not allow the long-term tracker to take full advantage of the information in the frames.

Recurrent networks that can facilitate Tracking have been applied in visual Tracking, e.g., Long Short-Term Memory (LSTM). Baik et al. [2] propose a Siamese tracking framework that combined an offline trained RNN network to predict the target features of the next frame. Unlike the existing frameworks, RMIT synchronizes the long-and short-term memories in a residual learning way. Moreover, the proposed tracker considers Spatio-temporal correlations, which can then be utilized for Tracking.

2.3. Tracking by deep learning

More recently, numerous deep learning-based tracking methods have been developed. Bertinetto et al. [3] propose a Siamese network, namely SiamFC, to determine the target state by comparing the feature similarity between the exemplar image and the search area. SiamFC establishes an end-to-end, fully convoluted Siamese network that runs beyond real-time frame rates. However, SiamFC predicts the target scale by multi-scale test, and this method is difficult to estimate the target size effectively. Motivated by the regional proposals net in object detection, SiamRPN [5] is proposed to enable more robust tracking performance, discarding the time-consuming multi-scale testing in SiamFC.

Although all the methods mentioned above can achieve good results, none consider the benefit of temporal information between target motions. Gao et al. [14] propose an inter-frame attention mechanism to explore temporal information of the target appearance by exploiting a convolutional LSTM unit. Yang et al. [48] focus on the potential target by combining spatial attention mechanism and LSTM, which can then obtain the long-term memory by updating the encoded target feature in its memory space. It should be noted that LSTM introduces a cell state to store the long-term memory of previous information. Lee et al. [24] construct a recurrent convolutional network to reinforce temporal relevancy by learning relevant features extracted from previous frames. However, RMIT focuses on combining both the short-term memory, which is maintained by the single convolution layer, and the long-term memory gained by the LSTM network. Furthermore, RMIT can utilize more reliable response scores to update end-to-end via a weighted gradient harmonized loss.

3. Proposed method

In this Section, we introduce the proposed RMIT model in detail. Specifically, the residual memory inference network is introduced in Section 3.1. The proposed weighted gradient harmonized loss function is elaborated in Section 3.2, together with our adaptive learning strategy.

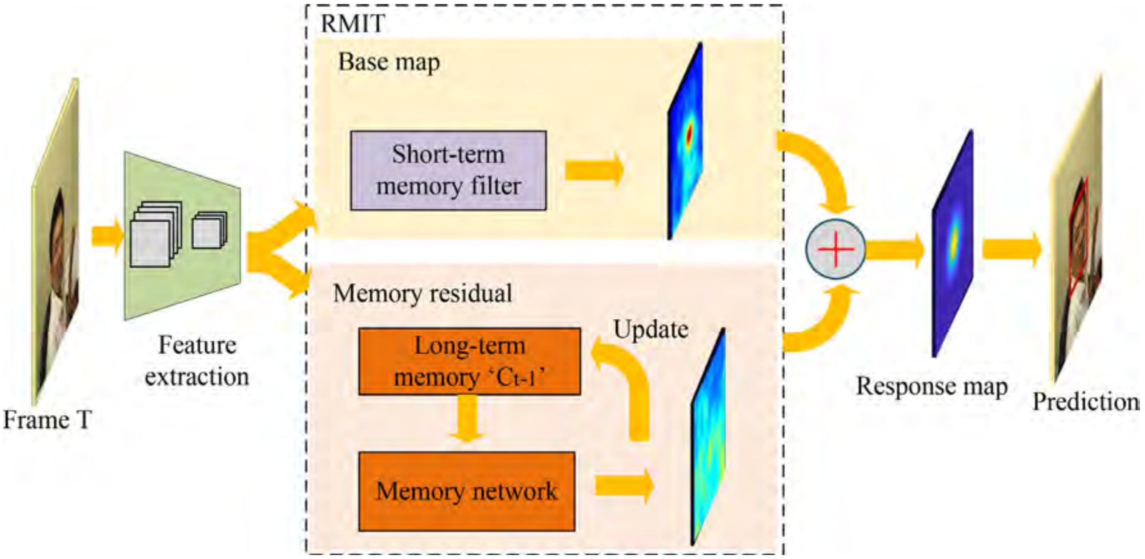


Fig. 2. Overview of the proposed tracker. RMIT contains base and memory residual branches and achieves target tracking by coordinating long-term and short-term memory in a residual learning way. Using weighted gradient harmonized loss, our tracker avoids model performance reduction degradation due to the sample imbalance problem to improve the discrimination.

3.1. Residual memory inference network

Fig. 2 gives an overview of RMIT, which consists of a base map and a memory residual branch. The base map branch, which only contains a short-term memory filter, will be introduced in Section 3.1.1. And the memory residual branch will be introduced in Section 3.1.2. Finally, the combination of these dual memory patterns will be presented in Section 3.1.3.

Model selection is a primary issue for data-driven models. It can be defined as the problem of selecting the data-driven model with the highest accuracy. Then model selection can be broadly divided into the following three types: 1) Choosing between different learning algorithms; 2) Setting the hyperparameters of a learning algorithm; 3) Choosing the structure of a learning algorithm [39]. We will discuss the structure choice of our method in this Section and leave the selection for hyperparameters in Section 5.3.3.

3.1.1. Short-term memory filter via convolutional regression

Inspired by [34,42], we first propose a convolutional regression network, which can be formulated as,

$$F(x) = w * f(x), \tag{1}$$

where $F(x)$ is the response map, $*$ denotes the convolution operator, w is the convolution filter, which is the short-term memory filter in Fig. 2.

Then, the loss function of online training can be formulated as follows [42,32]:

$$L(w) = \|F(x) - Y\|^2 + \lambda\|w\|^2, \tag{2}$$

where Y denotes the soft label of the regression target, which is usually generated by Gaussian functions, and λ is a regularization parameter.

In our paper, RMIT adopts reliable tracking results to adjust the convolutional filter in online regression. Thus, the convolution filter can be regarded as the short-term memory, which maintains radical appearance changes. Finally, this regression network can be trained by minimizing the loss function with gradient descent (GD).

The base layer can be regarded as the counterpart of DCFs, which is superior to correlation filtering from following two aspects: (1) The model complements the temporal information of the target by incremental learning with different networks, which have better scalability and less complexity; (2) The model can keep short-term tracker memory by adjusting the convolutional filter with tracking sets. Thus, our memory network is sensitive to changes in target appearance by updating the convolutional kernel in online regression to avoid rapid model degradation. Moreover, compared with general DCFs, the deep convolutional regression in the base layer eliminates boundary effects and has more substantial discrimination power with actual samples in the regression learning.

3.1.2. Long-term memory residual via ConvLSTM

Since short-term memory contains limited historical information, it is difficult for the tracker to adapt to changes in the appearance of targets over long periods. Thus, long-term memory can be exploited to locate the target, improving the stability of visual Tracking by incorporating historical information about the target. Some methods [34,48] maintain the target's long-term memory by either updating the tracker slowly or saving past states of the target. However, these approaches can only solve tracking problems in specific scenarios using long-term memory, such as occlusion or out-of-view. We introduce RNN for residual learning as the long-term memory network in RMIT from a Spatio-temporal perspective. However, it is difficult for the traditional RNN to maintain the target appearance information for too long due to the vanishing gradient problem. In contrast, we consider ConvLSTM in our RMIT, which can alleviate the negative effect of the vanishing gradient problem to some extent. The critical equations of ConvLSTM are as follows.

$$\begin{aligned}
i_t &= \sigma(W_{xi} * X_t + W_{hi} * h_{t-1} + W_{ci} \circ c_{t-1} + b_i),\\
f_t &= \sigma(W_{xi} * X_t + W_{hf} * h_{t-1} + W_{cf} \circ c_{t-1} + b_f),\\
c_t &= f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * h_{t-1} + b_c),\\
o_t &= \sigma(W_{xo} * X_t + W_{ho} * h_{t-1} + W_{co} \circ c_t + b_o),\\
h_t &= o_t \circ \tanh(c_t),
\end{aligned} \tag{3}$$

where t represents the frame index, X_t is input features extracted from the VGG16 Conv 3–3 layer and reduce the feature channels to 64 through Principal Component Analysis (PCA) dimensionality reduction, c_{t-1} is the output at the $t-1$ frame, W is weight matrix of the convLSTM, '$*$' and '$\circ$' denote the convolution operator the Hadamard product respectively. The LSTM first introduces a cell state c_t to store the long-term memory of the previous information. LSTM selectively forgets and remembers historical information by using a gating method without the vanishing problem. However, the general LSTM must unfold the frame to a 1-D vector as the input when processing video stream frames, which breaks the spatial information. The ConvLSTM, as an extension of LSTM with convolution underneath, overcomes this obstacle and propagates spatial information over time. This form of long-term memory allows feature regression to be carried out with spatial-temporal information.

3.1.3. Dual memory pattern combination via a residual framework

It is crucial to design an effective mechanism to switch between short-and long-term memories. Although the long-term memory can improve the performance in specific scenarios, short-term tracker still dominates the tracking. It is still challenging to leverage temporal information to assist in precise localization at the most moments. However, if the tracker always relies on long-term memory results, it also has the risk of failure when performing memory selection. Therefore, it is a significant point about combining both kinds of memories appropriately.

We focus on the potential benefits of temporal information to achieve target tracking by coordinating long-and short-term memories in a residual learning way. Considering the residual framework [42] can learn the variety of spatial inputs, RMIT utilizes a novel form of residual memory layers to learn the long-term memory of the target appearance. The residual learning building block is demonstrated in Fig. 3 (b). We implement memory-based tracking using long-term memory as compensation for short-term memory through a residual framework. The residual equation of RMIT is formulated as follows.

$$F(X_t) = F_M(X_t) + F_B(X_t), \tag{4}$$

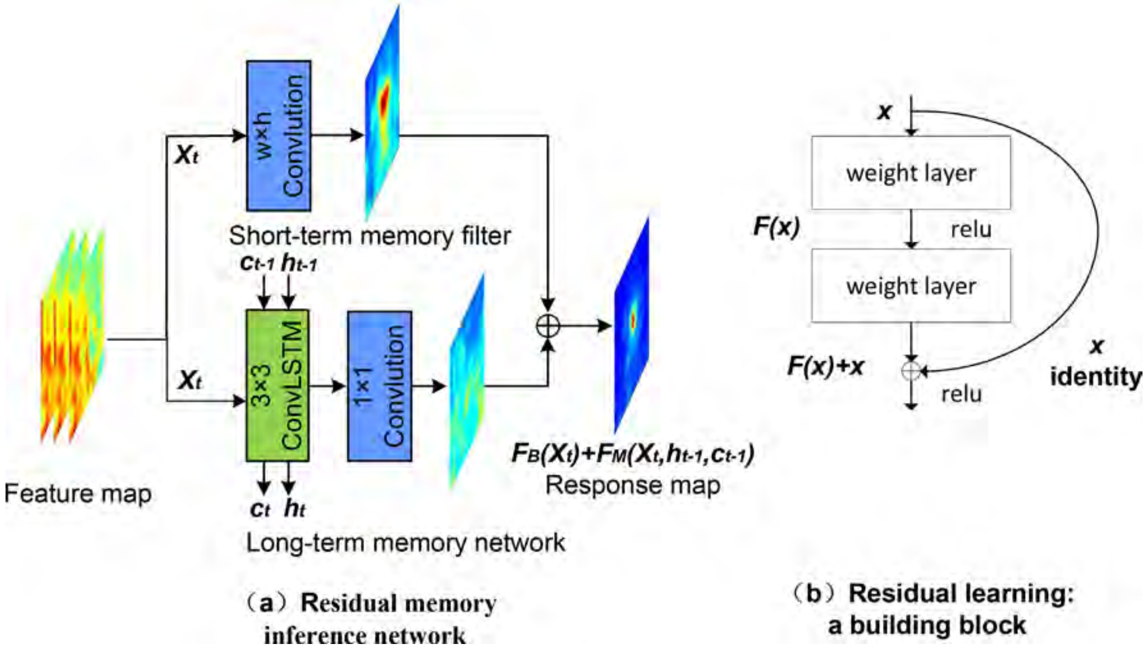
where X_t is input features, $F_M(X_t)$ represents our residual memory layer output, which is used to obtain long-term memory. $F_B(X_t)$ denotes the base layer output in Section 3.1.1, which is used to maintain short-term memory.

To accommodate the space-time features, we introduce a ConvLSTM into RMIT, where the ConvLSTM can model self-learned context information. The output of final RMIT model can be formulated as follows.

$$F(X_t) = F_M(X_t, c_{t-1}, F_M(X_{t-1})) + F_B(X_t), \tag{5}$$

Fig. 3 shows the residual blocks of RMIT and the residual learning building block. Compared with the residual learning building block, RMIT maintains the short-term memory of the target with its base layer, so it can maintain sensitivity to rapid changes. The target's long-term memory is introduced in the residual framework, which allows RMIT to learn the residuals of the target in the Spatio-temporal domain. Thus, it can compensate for the output of the base layer. More specifically, RMIT can utilize the historical state to update, not using the initial target only as in CREST [42]. Finally, RMIT can use rich temporal information for target inference, not just determining the spatial correlation between two frames.

RMIT has the following advantages compared with conventional memory-based tracking methods [49,48]. (1) The single-step regression framework does not require memory space [49,48]. It maintains long-and short-term memories for the target appearance by only adjusting the value in the convolutional filter. (2) The training set consists of tracking results according to adaptive learning strategy can then be utilized to fine-tune the network. Thus, our RMIT can save past changes to cope with the challenge of dramatic appearance changes through incremental learning. (3) The residual framework can adjust the contributions of both base and memory branches according to the apparent variation of the target. So, the response map can be as close to the regression target as possible. Compared with the memory selection strategy [34,26], it is easier to locate the target position by convolution with the maximum value of the response output. Therefore, RMIT can effectively exploit the potential benefits of long short-term memory and improve tracking performance.



(a) Residual memory inference network

(b) Residual learning: a building block

Fig. 3. Block diagrams of the RMIT and Residual learning building block. The cell state c_t in ConvLSTM is used to store the long-term memory of the previous information. h_t is the hidden state in ConvLSTM. X_t is the feature map of frame t.

3.2. Weighted gradient harmonized loss-based adaptive learning

RMIT conducts feature regression by integrating the outputs from the base and the residual memory layers. To overcome the problem of target missing, we construct an adaptive learning rule to reduce incorrect feature learning, especially in the case of complete occlusion. In Section 3.2.1, the weighted gradient harmonized loss is proposed to improve the reliability of the response scores. Then, an adaptive learning strategy is constructed to reduce incorrect feature learning in Section 3.2.2.

3.2.1. Weighted gradient harmonized loss function

One of the leading causes for the reduced discrimination is the extreme imbalance of positive-negative data in training regression networks. On one aspect, the large background regions in the input search patch can provide valuable contextual information to enhance the model's discriminating ability between the target and the background. On the other hand, it also introduces more negative samples, making it difficult for the regression model to accurately predict the positive results. Several works [32,31] on deep regression trackers introduce adjustment factors into the loss function to remove easy negative samples. However, such simple suppression depends on some hard manual-set thresholds, which is challenging to correct the current tracking object. Moreover, these hard thresholds are difficult to adjust dynamically, e.g., different loss values at different moments along the training process.

To address this problem, we introduce the gradient harmonized factor [25] into the loss function of the deep regression tracker. It can balance the proportion of different samples in the loss according to the gradient norm to reduce the overfitting of easy negatives. In particular, the gradient norm is first set to represent the sample attributes (e.g., easy or difficult), of which the L2 loss (Eq. (2)) can be formulated as follows.

$$g = \frac{\partial L}{\partial d} = 2|d|, \tag{6}$$

where $d = W \cdot X - Y$ is the error between the output and the label. Then, the gradient density function $GD(g)$ [25] that describes the distribution of the samples can be obtained by counting the number of samples at different gradient norms.

We then introduce the gradient harmonized factor β_i into the loss function of Eq. (2) to control the effect with a cost-sensitive weighting strategy [23], which can be formulated as follows.

$$L_{WGH-L} = \frac{\exp(Y_i)}{N} \sum_{I=1}^{N} \beta_i L(d_i), \tag{7}$$

where $\exp(Y_i)$ is the value of the soft label, which denotes an essential factor to highlight the valuable and rare samples. The weighted gradient harmonized loss approximates the L2 loss when the distribution of the gradient norm g is close to uniform distribution.

Next, we employ the signal-to-noise ratio [7] as a metric to evaluate the regression results, which can represent the accuracy of the regression model in predicting positiveness in visual tracking.

$$SNR(F(X_t)) = \exp(\max(F(X_t)) - \text{mean}(F(X_t))), \tag{8}$$

where SNR represents the signal-to-noise ratio, $F(X_t)$ denotes the response map. Using the maximum value of $F(x)$ to approximate the signal and the average value of $F(x)$ to approximate the noise. The search window is fetched from the first frame of the Bolt sequences, as shown in Fig. 4 (a). Since network training usually ends within a few hundred generations, we selected the first 100 iterations to demonstrate the effectiveness of the proposed method. It can be observed from Fig. 4 (b) that the convergence speed is significantly improved. Traditional L2 loss reduces regression errors for all samples, whether simple or difficult. The gradient harmonized factor adjusts for the contribution of a large number of simple samples to the loss. At the same time, $\exp(Y_i)$ additionally extends the positive sample weights to be accurately predicted.

3.2.2. Adaptive learning strategies for fine-tuning

Algorithm 1: Adaptive learning strategies for fine-tuning

```
Input: The most recent $m$ samples at time $t-1$, $Q_{t-1}$; Current sample pair, $q_t$;
Output: The most recent $m$ samples at a time $t$, $Q_t$;
Compute the Average Feature Energy Ratio of $q_t$, $AFER(q_t)$
if $AFER(q_t)$>th then
    $Q_t = \{q_{t-m+1}, q_{t-m+2} \cdots, q_t\}$
else
    $Q_t = Q_{t-1}$
end if
return $Q_t$
```

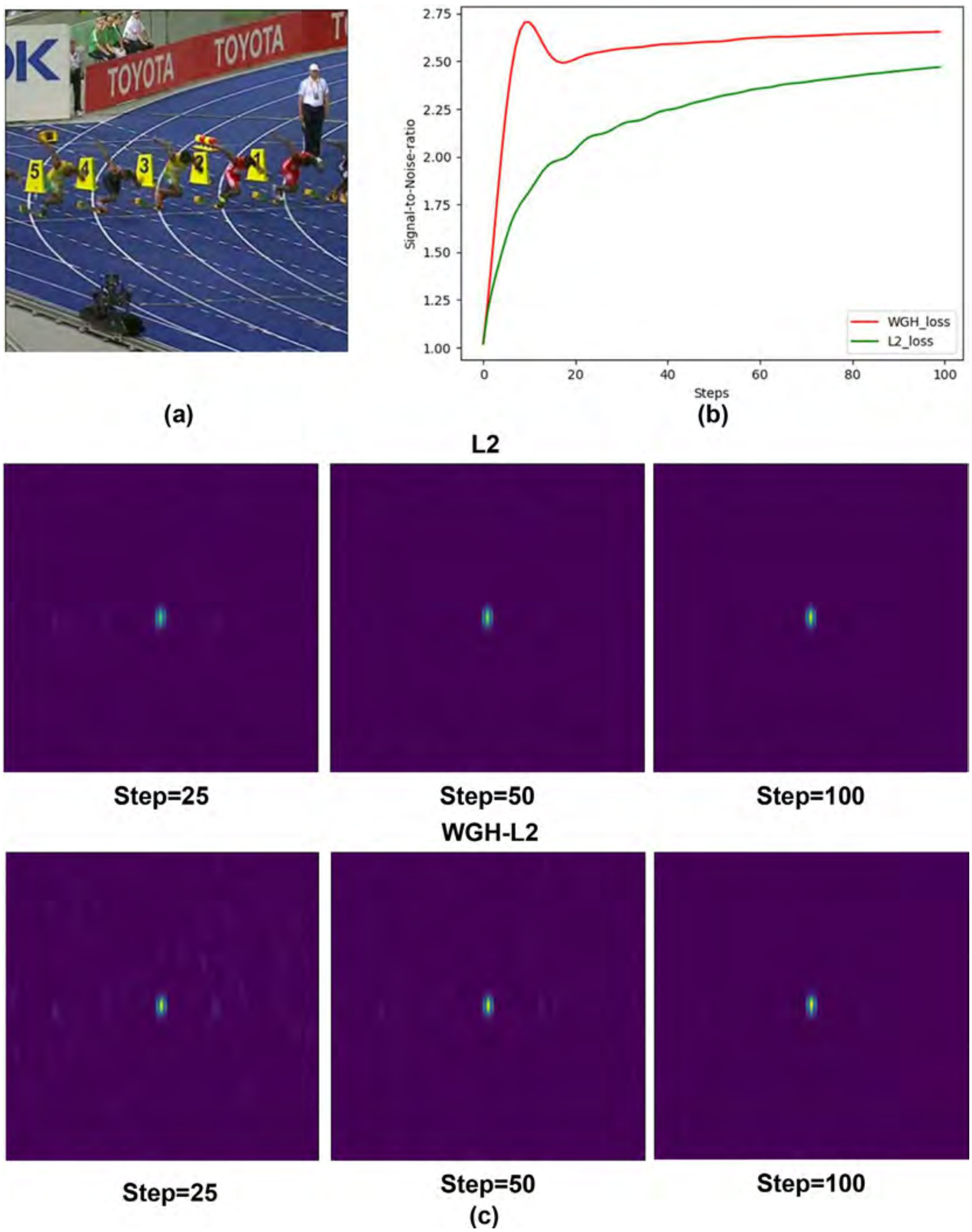


Fig. 4. Evaluation of convergence speed of gradient descent. (a) denotes the search patch of the first frame of the Bolt sequence. (b) represents a comparison of the regression results for different losses. (c) represents a visualization of the regression results.

To further improve the performance, we introduce rule-based learning in our tracking method. The training set can be expressed as follows.

$$Q_t = \{q_{t-m+1}, q_{t-m+2} \dots, q_t\}, \tag{9}$$

where Q_t is the training set consisting of the most recent m samples at a time t. q_t represents the sample pair consisting of the training patch and its response map. Since RNNs are utilized for long-term memory, the samples Q_t are arranged in chronological order. To avoid incorrect learning in difficult scenarios, such as occlusion or disappearance, the sample set Q_t at moment t is updated according to the following strategies as described in Algorithm 1.

$$Q_t = \begin{cases} \{q_{t-m+1}, q_{t-m+2} \ldots, q_t\}, AFER(q_t) > th \\ Q_{t-1}, \text{otherwise}, \end{cases} \tag{10}$$

where $AFER(\cdot)$ denotes the Average Feature Energy Ratio function [6], which is to calculate the effectiveness of response mapping and the confidence of tracking objects. The *AFER* value represents the reliability of the prediction target. During the tracking, when *AFER* of the current sample pair q_t is greater than the threshold *th*, we discard the farthest sample pair q_{t-m} and join the current one to maintain the sample size constant. Conversely, if the current sample pair is unreliable, the training set at time t keeps the same as the previous time moment. Using the above rules, the proposed RMIT network can avoid the learning of incorrect features in some scenarios, e.g., occlusion or out-of-view.

4. Tracking via RMIT

In this Section, we elaborate on how to integrate the residual memory inference network (Section 3.1) and adaptive learning (Section 3.2) into our tracker, which includes: 1) Initialization, 2) Tracking, and 3) Updating.

Initialization. Our tracker is initialized in the first frame. Specifically, a training patch is first extracted from the input frame with its target location. Then, the patch is fed into our framework to obtain the response mapping. Next, we adopt the Conv3–3 in the VGG-16 network to do the feature extraction, which is suitable for precise localization at the earlier layers that contain fine-grained spatial information. At the same time, all the parameters in RMIT are randomly initialized with standard Gaussian distribution.

Tracking. When a new frame appears, the search patch is obtained from the center of the prediction result in the previous frame. Then, the response map is acquired by feeding the search patch into the proposed network. The target position is determined by searching for a maximum value of the response output. Inspired by [42], we can obtain search patches at the center of the target's previous frame position with three different scales. Then, we resize these search patches to a fixed size. The response maps at multiple scales can be obtained simultaneously by computing these search patches. The target scale is updated according to the target maximum response correspondence scale, formulated as follows.

$$(w_{new}, h_{new}) = \zeta(w_{max}, h_{max}) + (1 - \zeta)(w_{old}, h_{old}), \tag{11}$$

where factor $\zeta$is used to penalize larger target changes. (w_{max}, h_{max}) means the target scale corresponding to the maximum response. (w_{old}, h_{old}) denotes the target scale at time $t - 1$.

Updating. After predicting the target, we construct a sample set of multiple recent reliable sample pairs and feed them into our network for training (Cf. Section 3.2). The sample-set is updated over time according to the rules in Section 3.2.2. Since RMIT allows long-term memory to be kept separately during the learning process, the network can infer the current target states from historical and recent states.

5. Experiments

In this Section, we first the details of the experiment. Then, we evaluate RMIT with other algorithms on several public datasets.

5.1. Experimental setups

Our RMIT tracker runs at 6 frames per second (FPS) on a PC with an i3 3.2 GHz CPU and a GeForce GTX 1060 GPU with TensorFlow. The training patch is set to five times the initial target size. Based on VGG16, the feature maps are extracted from the Conv3–3. The convolutional kernel size in the base branch is set to be the target size according to the feature extraction network. The regression target map is generated using a two-dimensional Gaussian function with a peak value of 1.0. In the propagation mechanism of ConvLSTM, we use a 3×3 convolution kernel instead of 1×1 in ConvLSTM. Because larger convolutional kernels have a wider receptive field, it is suitable to capture fast-moving or large objects. Afterward, a 1×1 convolution layer is utilized to adjust the shape of the output from the residual memory layer to match the output from the base layer. The learning rate of the Adam optimizer is set to 4e-8 to update the coefficients during training until the loss (Eq. (7)) is smaller than the given threshold, 0.01. Our network is converged from Gaussian initialization. In the online update phase, the model is updated every frame with a learning rate of 1e-9. In addition, we update the sample set as a training batch that contains time steps, whose size m is given to 5.

Error estimation estimates the error of the selected model on previously unseen data, relying only on the available data [39].

We use standard evaluation metrics as benchmarks to estimate model errors. For the OTB-50 [45], OTB-100 [44], UAV-123 [29], and TC-128 [30], we use the one-pass evaluation (OPE) with precision and success plots metrics. The tested tracker runs throughout a video sequence and performs the performance evaluation with only the initial state given at the first frame. This precision metric measures the ratio of frame locations whose distances are within a certain distance threshold from the ground truth. The success metric measures the overlap ratio between predicted and ground truth bounding boxes. In addition, the distance precision (DP) score at a threshold of 20 pixels is used to rank the tracker performance for comparisons. For VOT-2016 [36] and VOT-2018 [22] datasets, the performance is measured in terms of overlap, failures, and expected average overlap (EAO). The overlap means the average overlap while tracking successfully. Failures denote the number of failure times in the full Tracking. EAO is utilized to evaluate the overall performance of the tracker.

5.2. Ablation studies

We first perform an ablation analysis to compare the performance of the memory branch in the base branch, i.e., Base v.s. Base + Memory. Then, we try to evaluate the performance of the proposed weighted gradient harmonized loss. Due to the space limitation, our ablation studies are only conducted in OTB-100 dataset [44].

Fig. 5 shows the results of the base branch, the tracker with memory branch, and RMIT. The base branch is sensitive to current states, and its discriminability of the target gradually decreases as Tracking continues. RMIT introduces a long-term memory branch via the residual connection. This branch retains the past state of the target through the ConvLSTM network and is used for online Tracking. Furthermore, the residual connection can better coordinate dual memory patterns to balance the two advantages. The significant improvement of complete network structure in both accuracy and success rate proves the validity of the proposed approach where long-and short-term memory are combined in the form of residuals in Fig. 5.

We then analyze the effect of weighted gradient harmonized loss, whose results are shown in Fig. 4. We can observe that the performance of the RMIT is significantly improved through the integration of weighted gradient harmonized loss. It indicates that including a weighted gradient harmonized factor in L2 loss helps online tracking networks to converge. Compared to the basic regression loss, it also improves the tracker's accuracy with limited iterations.

Table 1 shows the influence of the new form of residual memory layer on the overall complexity of the algorithm. We evaluate the algorithm from two aspects of time complexity and space complexity, in which the space complexity and time complexity are reflected by the number of model parameters and the tracking speed of the algorithm, respectively. At the same time, DP scores were also taken into account to comprehensively evaluate the influence of the residual memory layer on the algorithm. As can be seen from Table 1, due to the introduction of convLSTM into the residual memory layer, the number of parameters in the model increased significantly, while the tracking speed decreased slightly. However, the residual memory layer positively affects the tracker's accuracy. Moreover, the distance precision of our tracker increased from 83.5% to 85.3%, realizing the high-precision tracking performance.

5.3. Overall performance

We evaluate the performance of the tracker on several benchmark datasets including OTB-50 [45], OTB-100 [44], UAV123 [29], TC-128 [30], VOT2016 [36] and VOT2018 [22].

5.3.1. Quantitative evaluation

OTB-50 Dataset. Fig. 6 shows the performance of RMIT together with 11 advanced trackers including HCFT [35], DCFNet [43], Staple [4], CNN-SVM [19], SiamFC [3], CSRDCF [42], HCFTstar [33], SRDCF [10], CREST [42], ECO-HC [9], SiamRPN [5]. We can observe that our approach outperforms DCFs trackers, HCFT, and HCFTstar using deep hierarchical features. Although HCFTstar designs additional detection mechanisms to cope with tracking failures, its performance is still worse than our

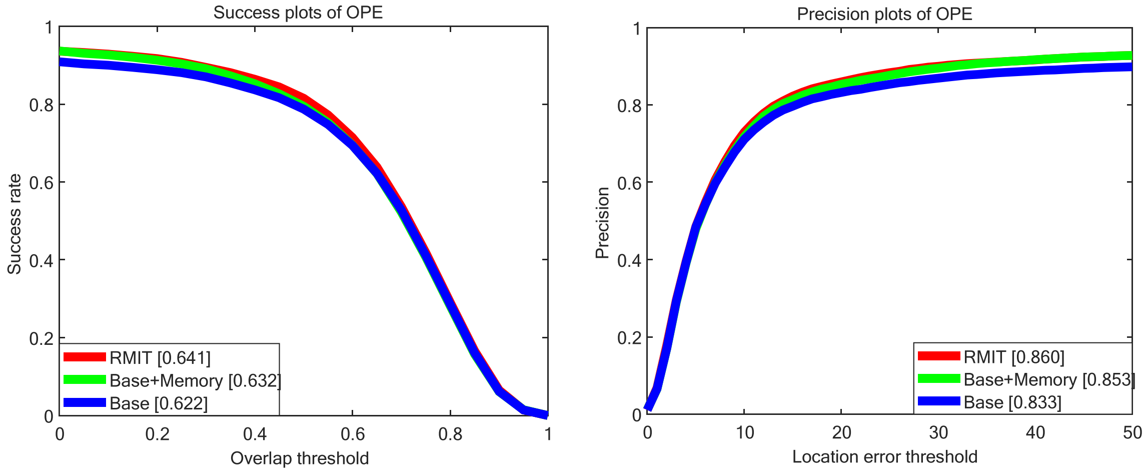


Fig. 5. Precision and success plots using one-pass evaluation on the OTB-100 dataset. The performance of the base branch is improved gradually with the integration of the memory branch and weighted gradient harmonized loss.

Table 1
The algorithm complexity analysis of all trackers are compared on DP scores (%), Time Complexity (FPS) and Space Complexity (params) on the OTB-100 Dataset.

Tracker	DP (%)	Time Complexity (FPS)	Space Complexity (params)
Base	83.3	6.5	5,377
Base + Memory	85.3	5.9	300,610

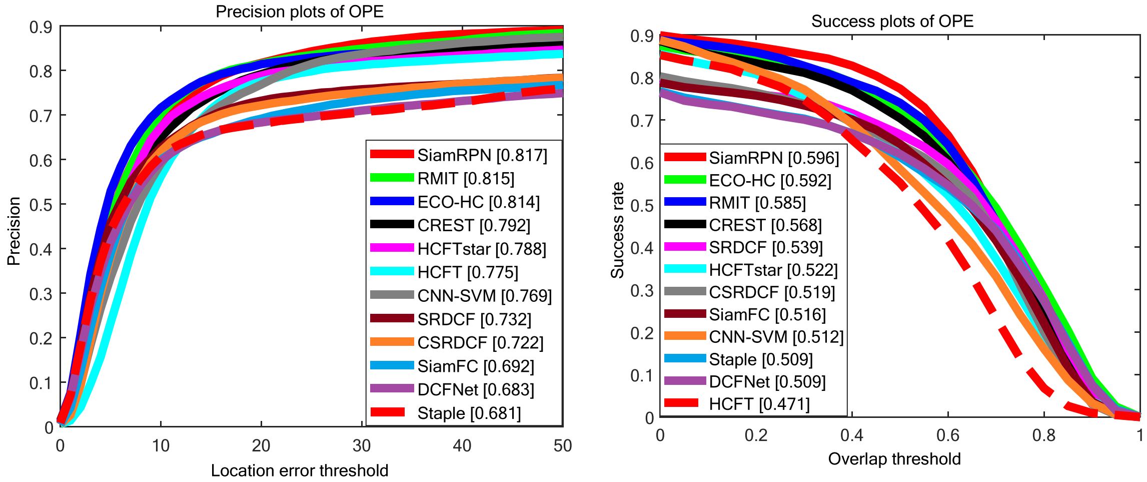


Fig. 6. The performance of all tracers are compared in precision and success plots on the OTB-50 Dataset.

method. Although SiamRPN obtains the best tracking results, the performance of RMIT, SiamRPN, and ECO-HC are similar in terms of distance accuracy and overlap threshold.

OTB-100 Dataset. Fig. 7 also shows the performance of RMIT together with the same 11 advanced trackers including HCFT [35], DCFNet [43], Staple [4], CNN-SVM [19], SiamFC [3], CSRDCF [42], HCFTstar [33], SRDCF [10], CREST [42], ECO-HC [9], SiamRPN [5]. We can observe that our RMIT tracker achieves optimal tracking accuracy with more video sequences. Moreover, we can also observe that our method outperforms SiamRPN in success rate, achieving a suboptimal performance. Then, we can observe that the RMIT tracker performs better in accuracy and success rate than the deep regression tracker CREST. It indicates that our approach integrates the Spatio-temporal information of the target for memory learning through ConvLSTM is better than CREST, where the spatial residuals learn the recent target changes and the temporal residuals provide only part of target information in the first frame.

Fig. 8 shows the performance of the tracker under different challenges (e.g., scale variation, out of view, and occlusion). We can observe that our tracker can better deal with the challenge out of view. The memory branch retains information

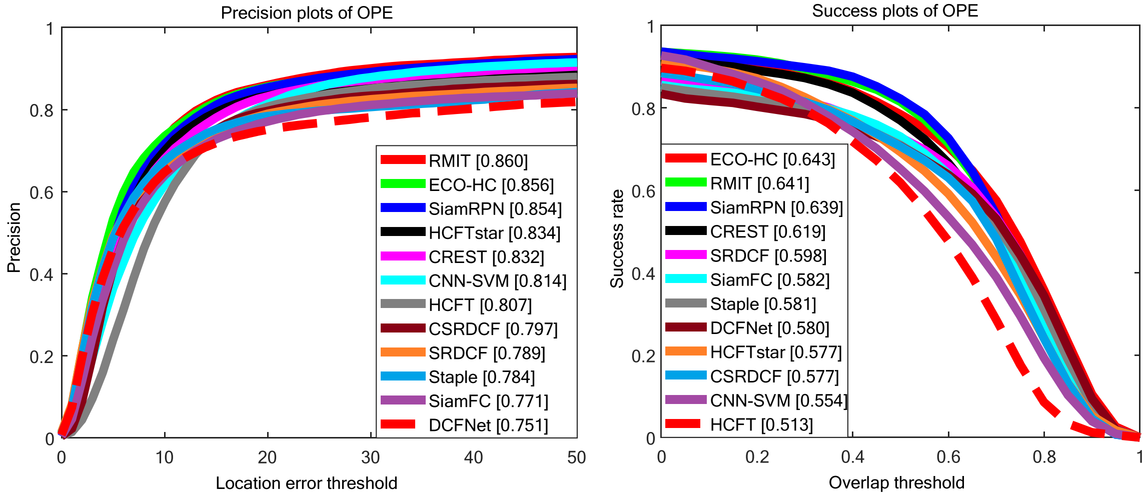


Fig. 7. The performance of all tracers are compared in precision and success plots on the OTB-100 Dataset.

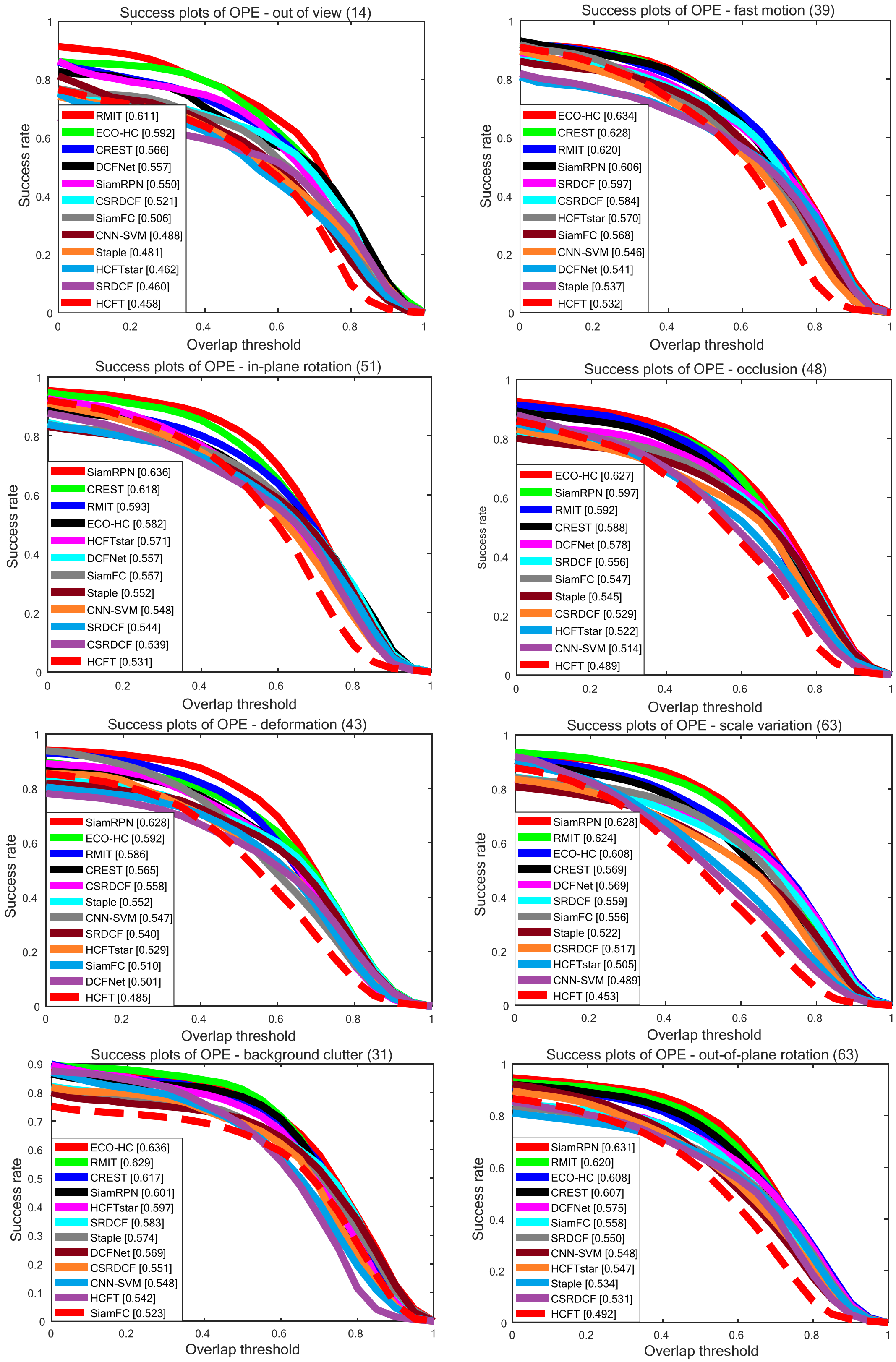


Fig. 8. The success plots the performance of the tracker in the eight tracking challenges.

about the target in the past moments. The use of target long-term representations can effectively mitigate erroneous updates when confronted with targets escaping out of view. In addition, Tracking is also more accessible with the help of target memory recovery. The proposed method is slightly inferior to the SiamRPN that uses a multi-branch network to regress the target boundary and validate the classification separately by borrowing from the RPN network. Our approach reduces the distance between CREST and SiamRPN, also regression trackers. By integrating the target history information to reason about the target, the gap between the label and the output is further reduced to make the prediction more accurate. Among other challenges, RMIT still occupies a relatively high position. To preserve the spatial resolution of the targets, we use shallower features, which make it challenging to obtain the best performance of our method in the face of some target deformations. As a result, RMIT only achieves a performance similar to that of ECO using many manual features.

The real-time performance of the tracker should also be considered to evaluate the tracker performance and can be represented by tracking speed. In Table 2, the runtime performance of some state-of-the-art trackers is compared on DP scores (%) and Speed (FPS). From Table 2, We observe that SiamFC achieves high speed, but since it is not able to update online, it is not well adapted to the challenges of the tracking process resulting in low accuracy. The CREST and our RMIT tracker perform better in accuracy than other algorithmic. However, the speed performance of CREST is not as satisfactory as RMIT. RMIT runs at 6.1 FPS and cannot reach the effectiveness of real-time Tracking. We believe that this result is as follows: (1) it takes much time for our tracker to obtain the initial parameters in the first frame. (2) the RMIT needs to be updated online in the tracking process. Overall, our RMIT tracker performs favorably and achieves an 86.0% DP score while running at 6.1 FPS.

TC-128 Dataset. Fig. 9 shows the results of RMIT tracker compared with 10 advanced trackers including TADT [16], KCF [18], DSST [12], HCFT [35], KCFDP [21], CREST [42], HCFTstar [33], LOT [40], ECO-HC [9] and CACF [37]. We can observe that our method obtains the optimal performance overall. The gap between RMIT and CREST is further widened. The training batch and loss can help the online tracker maintain sufficient training on the target, which can also help the optimizer converge in the right direction. Compared with other depth feature-based trackers, HCFT and HCFTstrar, RMIT is more accurate in regression of target labels. Overall, it can be observed from Fig. 9 that our tracker has achieved better performance in TC-128.

UAV-123 Dataset. In Fig. 10, Our RMIT tracker is compared with 10 advanced trackers including SiamRPN [5], CREST [42], SRDCF [10], MEEM [50], HCFT [35], SAMF [47], MUSTER [20], DSST [12], Struck [50] and KCF [18]. We can observe that the proposed method RMIT achieves suboptimal performance in both accuracy plot and success plot, and RMIT has significant advantages over the related filter trackers HCFT, DSST, and KCF. Then, compared with HCFT, which uses multiple layers of semantic abstraction, we can notice that RMIT gains more by using only Conv3–3 features. The residual learning fully exploits the potential ability of the model. Compared to the depth regression tracker CREST, RMIT considers the temporal information of the target via ConvLSTM, which in turn improves its performance. It demonstrates that the target's memory in the historical information can significantly improve tracking. Compared with SiamRPN, RMIT can also obtain close accuracy and success rate.

Table 2
The runtime performance of all trackers are compared on DP scores (%) and Speed (FPS) on the OTB-100 Dataset.

Tracker	SiamFC	CREST	CSRDCF	Staple	SRDCF	RMIT
DP (%)	77.1	83.2	79.7	78.4	78.9	86.0
Speed (FPS)	102.3	1.8	8.5	61.3	4.2	6.1

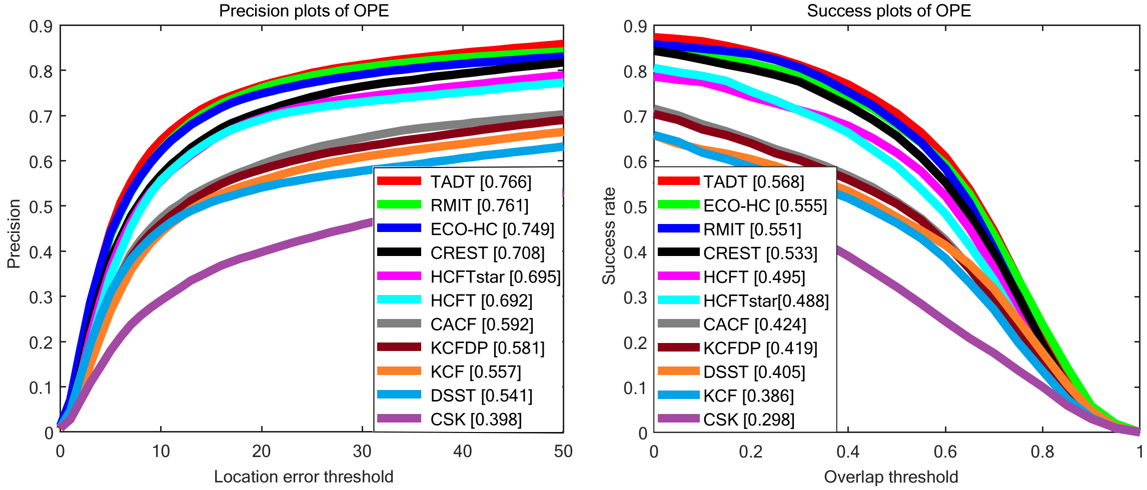


Fig. 9. The performance of all tracers are compared in precision and success plots on the TC-128 Dataset.

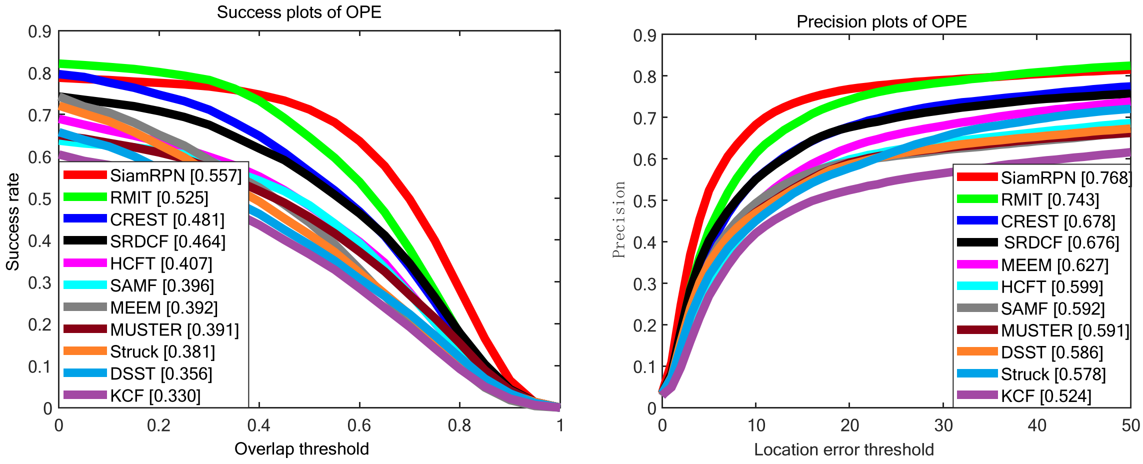


Fig. 10. The performance of all tracers are compared on precision and success plots on the UAV-123 Dataset.

VOT2016 Dataset. Table 3 shows the performance of RMIT tracker compared with 10 advanced trackers, including Staple [4], KCF [18], DSST [12], SRDCF [10], SiamAN [36], SAMF [47], MDNet [38], HCFT [35], DeepSRDCF [11], and DAT [41]. It can be observed that our RMIT tracker performs well. RMIT obtains a performance close to Staple's and outperforms other algorithms in terms of success rate metrics. Moreover, RMIT achieves the lowest failure evaluation metrics because of historical information as a tracker without re-detection steps.

VOT2018 Dataset. Table 4 shows the results of our RMIT tracker compared with 10 advanced trackers, including Staple [4], SRDCF [10], SiamFC [3], RSECF [8], MEEM [50], KCF [18], DSST [12], DSiam [15] DensSiam [1], DCFNet [43]. We can observe that RMIT achieves the first in EAO. Moreover, the proposed method has a considerable improvement compared with Staple. Moreover, in the metrics of Overlap and Failures, RMIT can rank at 2nd and 1st, respectively.

5.3.2. Qualitative evaluation

Fig. 11 shows some results of the top-performing trackers: ECO-HC [9], SiamRPN [5], CREST [42], HCFTstar [33] and SRDCF [10] and our RMIT tracker on 7 challenging sequences. In the first row of the Basketball sequence, the tracking scene is more

Table 3
Comparison with the advanced trackers on the VOT 2016 dataset. The results are presented in terms of Overlap,Failures and expected average overlap (EAO).

Tracker	Overlap	Failures	EAO
Staple	0.5403	23.8950	0.2941
SRDCF	0.5258	28.3167	0.2458
SiamAN	0.5311	29.8021	0.2358
SAMF	0.4965	37.7937	0.1851
MDNet	0.5386	21.0817	0.2579
KCF	0.4699	52.0310	0.1533
HCFT	0.4361	23.8569	0.2198
DSST	0.5245	44.8138	0.1805
DeepSRECF	0.5220	20.3462	0.2756
DAT	0.4581	28.3533	0.2166
RMIT	0.5352	16.5724	0.2892

Table 4
Comparison with the advanced trackers on the VOT 2018 dataset. The results are presented in terms of Overlap,Failures and expected average overlap (EAO).

Tracker	Overlap	Failures	EAO
Staple	0.5073	45.5561	0.1664
SRDCF	0.4634	66.7637	0.1158
SiamFC	0.4809	35.9983	0.1838
RSECF	0.4500	33.9447	0.2013
MEEM	0.4429	36.5823	0.1897
KCF	0.4364	52.4512	0.1347
DSST	0.3832	97.9973	0.0789
DSiam	0.4898	42.8737	0.1921
DensSiam	0.4436	46.9544	0.1708
DCFNet	0.4525	37.6251	0.1798
RMIT	0.4875	25.4464	0.2350

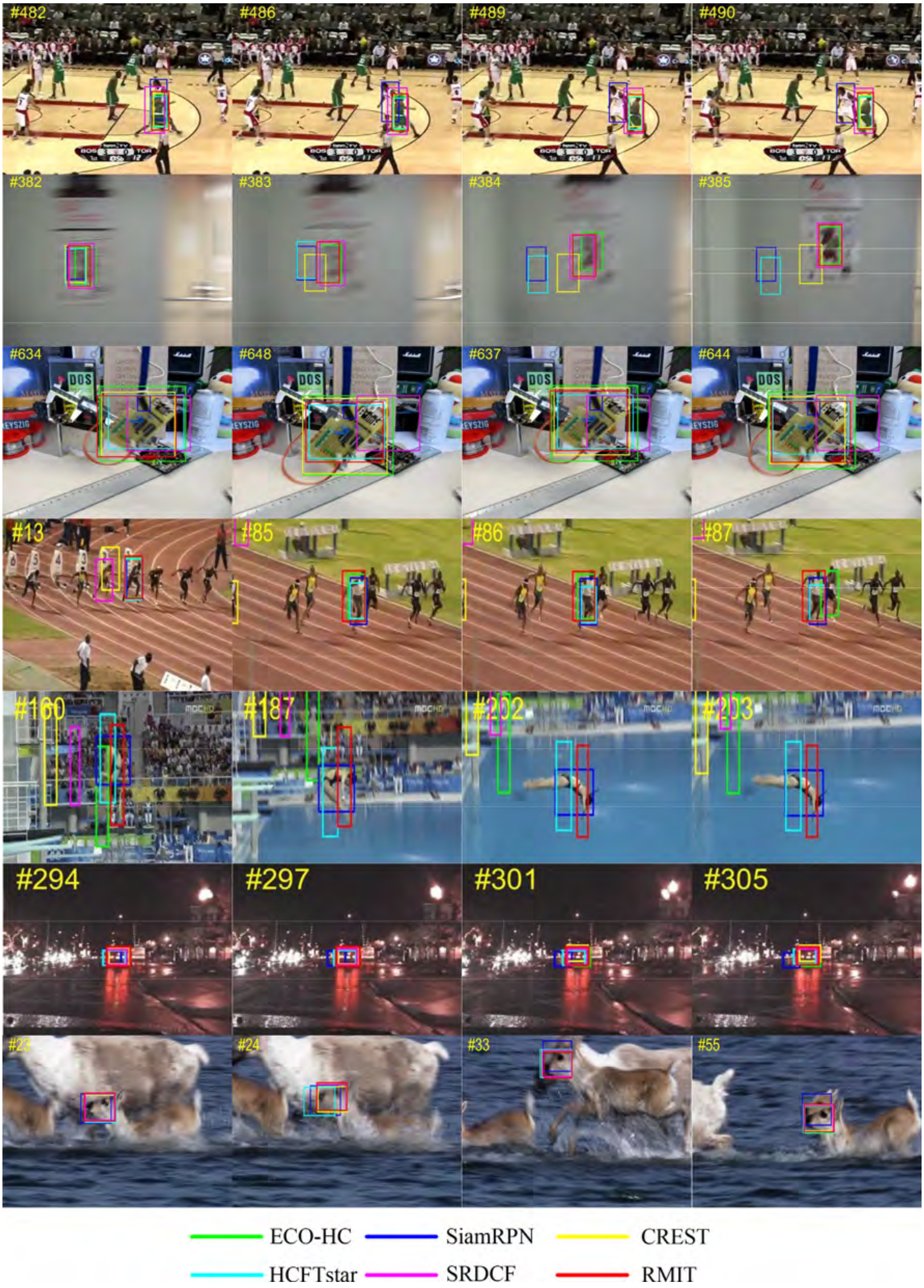


Fig. 11. Qualitative evaluation of our RMIT tracker, ECO–HC, SiamRPN, CREST, HCFTstar and SRDCF on 7 challenging sequences (from top to down: Basketball, BlurOwl, Board, Bolt2, Diving, CarDark and Deer, respectively). Our RMIT tracker performs favorably against the advanced trackers.

complex due to the interference. We can observe that SiamRPN drifts when similar objects appear, then focuses on the wrong target in subsequent frames. However, our method can maintain the Tracking of the target thoroughly.

In the BlurOwl sequences, the severe shaking of the camera produces a giant motion blur. It can be observed that SiamRPN, ECO-HC, and HCFTstar undergo a severe drift. ECO-HC uses hand-crafted features that make it difficult to describe the target in heavily ambiguous scenes correctly. HCFTstar, which uses multi-layer depth features, also has difficulty keeping track due to the lack of model discriminatory power. However, our RMIT algorithm can exploit the model's potential through residual memory learning.

In the Blot2 sequence, we observe that CREST cannot converge to the optimum, causing the tracker to remain local. The ECH-HC tracker drifts at frame #87 due to similar objects. However, the RMIT tracker can maintain good Tracking throughout.

In the Diving sequence, the target undergoes a rapid falling motion accompanied by severe deformation. It can be observed that the HCFTstar maintains tracking of the target by the re-detection module. Although our method can also maintain the basic Tracking because of the introduction of the target's history frame information, the lack of ability to adjust the aspect ratio leads to the inability to estimate the target shape accurately.

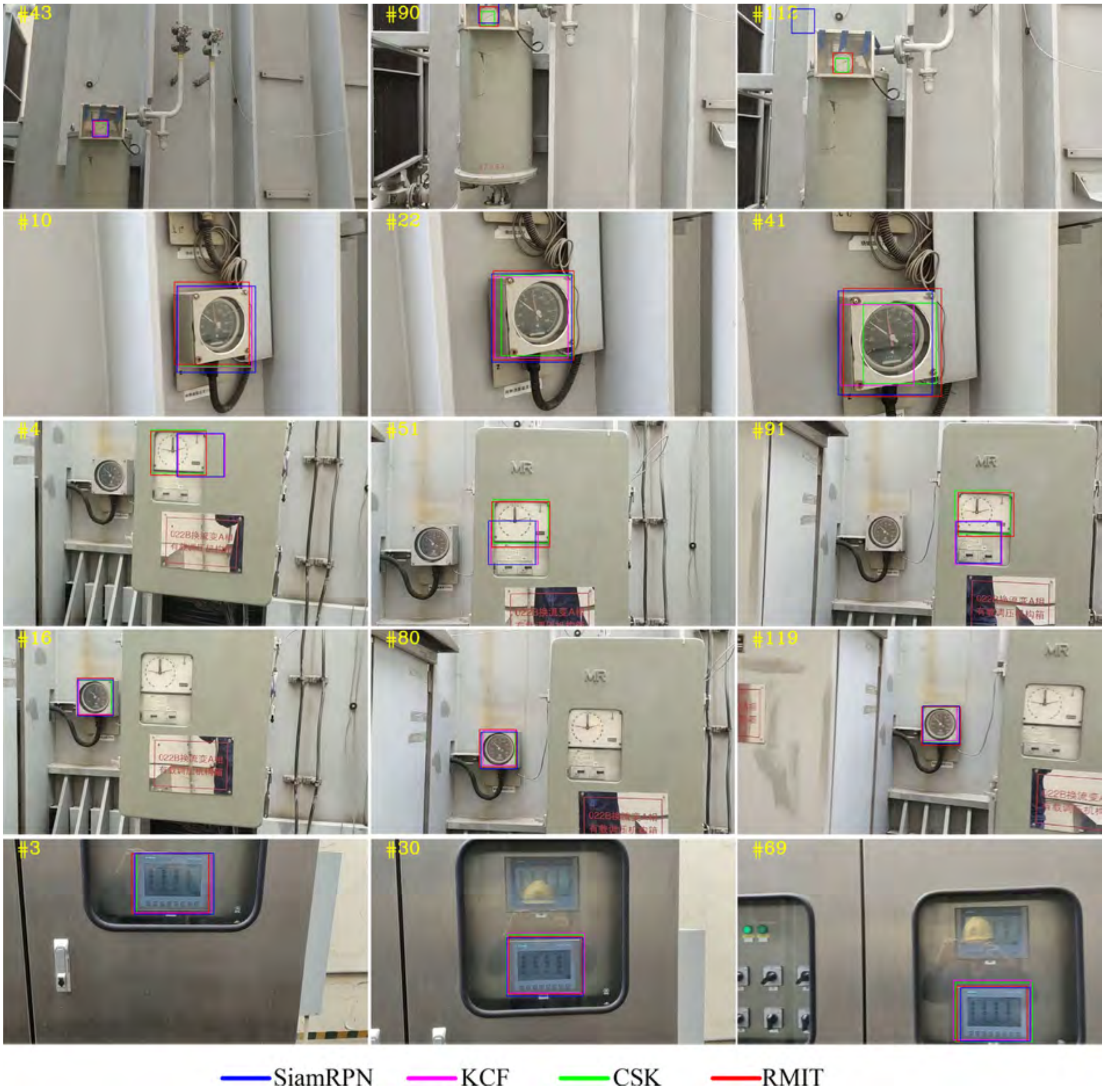


Fig. 12. Qualitative evaluation of our RMIT tracker, SiamRPN, KCF, and CSK on 5 electricity scenes on the actual electric domain scene videos. Our RMIT tracker performs favorably against the advanced trackers.

In the final CarDark and Deer sequences, the SiamRPN and HCFtstar drift to varying degrees due to headlights, water splashes, and similar objects interfering with them, respectively. Overall, the visual evaluation indicates that our RMIT tracker performs favorably against these trackers.

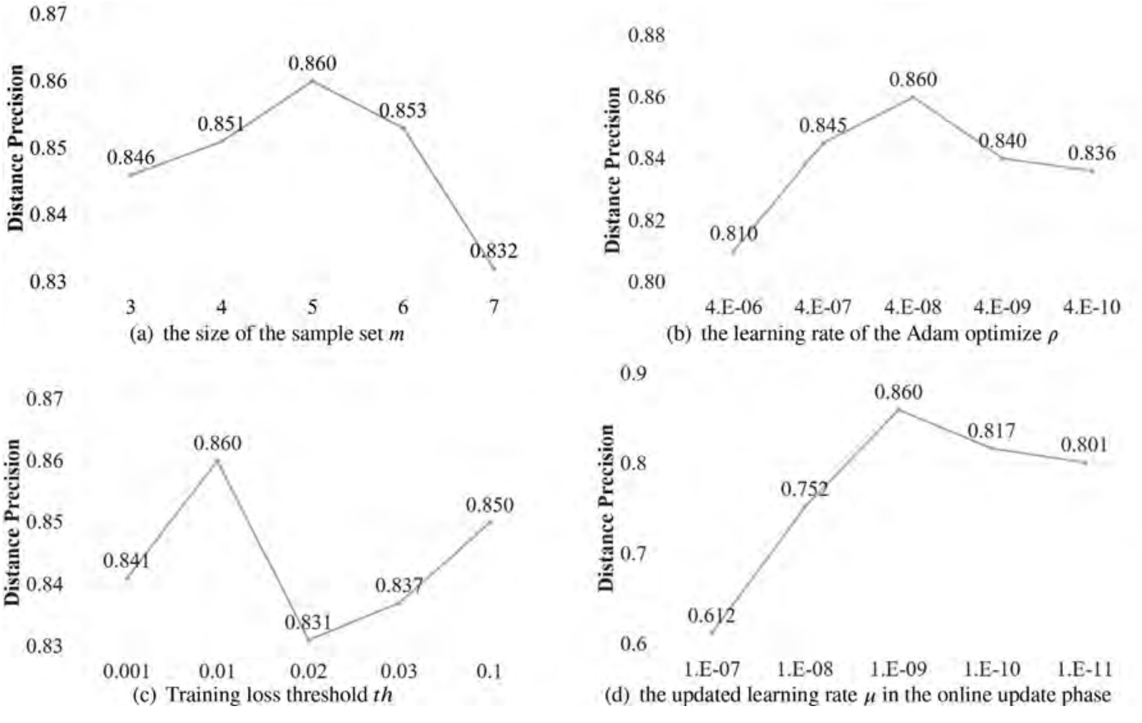
Nowadays, in the intelligent inspection of electric power, the application of vision-based inspection technology in electric power inspection has been widely concerned. Then, we qualitatively evaluate the proposed method in the real electric domain. We collected several electric domain scene videos and made a small dataset for the experiment. Fig. 12 shows the results of the top-performing trackers: SiamRPN [5], KCF [18], CSK [17], and our RMIT tracker on five actual electric sequences. It can be observed from the first row that SiamRPN based on the detection method does not apply to this power scenario. Our method keeps track of the electrical instrumentation in this real-world scenario. It is evident from the second line that KCF and CSK based on correlation filter are not robust enough to face the challenge of appearance change of the instrument due to the change in the camera view. RMIT integrates both short-term and long-term memory to remember the target's appearance so that it can adapt well to changes in the object's appearance. In the other power scenarios, RMIT still shows excellent performance. Therefore, we can conclude that our algorithm can cope with the changes in the appearance of electrical instruments caused by camera motion in a real power scene.

5.3.3. Parameter analysis

Hyperparameters exist more or less in the algorithm and define some rules of the model learning stage. It is essential to learn the suitable hyperparameters to minimize the model error [39]. In this Section, as shown in Fig. 13, We tune the values of some critical hyperparameters and use the DP scores to estimate the model error. The high DP score means the closer the center of the model prediction and the ground truth and the lower the model's errors. These parameters include the size of the sample set m, the learning rate of the Adam optimize ρ, the training loss threshold th, and the updated learning rate μ in the online update phase. All parameter experiments are conducted on the OTB-2015 dataset.

(1) **effect of m:** The parameter m is the size of the memory sample set, which is used to update the model online. An adapted sample set is used as a training batch containing time steps. A larger m means that more sample information can be used to update the proposed model. The experimental results are shown as Fig. 13a when we set m to 3, 4, 5, 6, and 7. We can observe from the figure that the tracker works best when m is set to 5.

(2) **effect of ρ and th:** The learning rate of the Adam optimize ρ is an important hyper-parameter in supervised learning and deep learning. It determines the learning progress of the network model and determines whether the network can succeed or how long it can find the global minimum. While the training loss threshold th is one of the terminate conditions for the training. Moreover, it is a method to avoid over-fitting. As shown in Fig. 13b and Fig. 13c, the proposed model achieves the best performance when the learning rate ρ is 4e-8, and the threshold value th is 0.01.



(a) the size of the sample set m

(b) the learning rate of the Adam optimize ρ

(c) Training loss threshold th

(d) the updated learning rate μ in the online update phase

Fig. 13. Influence of hyperparameter tuning on tracking performance.

(3) **effect of μ:** The parameter μ denotes the updated learning rate when the network is updated online, which affects the update speed of memory in the model and the convergence of the model. Fig. 13d shows tracker performance for the tracker with different μ. We can observe that the tracker performs more competitively when μ is taken as 1e-9.

6. Concluding remarks

This paper proposes a residual memory-based single-step regression inference tracking framework, namely RMIT, which takes advantage of the potential gains from long short-term memory. In contrast to other memory networks, this network can synergize long-and short-term memories in a single model to achieve Tracking. We update the weights of the single-layer convolutional network by gradient descent to maintain the ability to remember rapid changes of the target. Moreover, the feature flow through the recurrent unit of RNNs also carries rich information of previous inputs to maintain the long-term memory. We connect these two branches through a residual framework to facilitate the learning and updating of the memory network in an end-to-end manner. We propose an L2-based weighted gradient harmonized loss function for the sample imbalance problem in regression networks training to improve regression learning further. The proposed method is extensively validated on six benchmark datasets, including OTB-50/100, TC-128, UAV-123, and VOT-2016/2018. The results show that our RMIT tracker performs favorably against other trackers.

Although the proposed RMIT has achieved good performance, it takes some time for our tracker to obtain the initial parameters in the first frame. Thus, it can not achieve the effect of real-time Tracking. In future research, we plan to investigate the use of meta-learning methods to generate an optimal set of initialization parameters so that the network can be trained online in the first frame at a faster convergence speed. Moreover, we will also explore the combination of the invariance of the target and the memory model to achieve robust object tracking.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant (61873246, 62072416, 62102373, 61806181, 62006213), Program for Science & Technology Innovation Talents in Universities of Henan Province, China (21HASTIT028), Natural Science Foundation of Henan Province, China (202300410495) and Zhongyuan Science and Technology Innovation Leadership Program, China (214200510026).

References

[1] Mohamed H. Abdelpakey, Mohamed S. Shehata, Mostafa M. Mohamed, Denssiam: End-to-end densely-siamese network with self-attention model for object tracking, in: International Symposium on Visual Computing, Springer, 2018, pp. 463–473.

[2] Sungyong Baik, Junseok Kwon, Kyoung Mu Lee, Learning to remember past to predict future for visual tracking, in: 2019 IEEE International Conference on Image Processing (ICIP), IEEE, 2019, pp. 3068–3072.

[3] L. Bertinetto, J. Valmadre, Joo F. Henriques, A. Vedaldi, Phs Torr, Fully-convolutional siamese networks for object tracking, European Conference on Computer Vision (2016).

[4] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip H.S. Torr, Staple: Complementary learners for real-time tracking, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1401–1409..

[5] L. Bo, J. Yan, W. Wei, Z. Zheng, X. Hu, High performance visual tracking with siamese region proposal network, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[6] M. Che, R. Wang, L. Yan, L. Yan, Z. Hui, C. Xiong, Channel pruning for visual tracking, Springer, Cham, 2018.

[7] Kai Chen, Wenbing Tao, Convolutional regression for visual tracking, IEEE Trans. Image Process. 27 (7) (2018) 3611–3620.

[8] L. Chu, H. Li, Regressive scale estimation for visual tracking, in: 2019 IEEE International Conference on Industrial Technology (ICIT), 2019.

[9] M. Danelljan, G. Bhat, F.S. Khan, M. Felsberg, Eco: Efficient convolution operators for tracking, IEEE Comput. Soc. (2016).

[10] M. Danelljan, G. Hager, F.S. Khan, M. Felsberg, Learning spatially regularized correlation filters for visual tracking, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015.

[11] M. Danelljan, G. Hager, F.S. Khan, M. Felsberg, Convolutional features for correlation filter based visual tracking, in: 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), 2016.

[12] Martin Danelljan, Gustav Häger, Fa.had. Khan, Michael Felsberg, Accurate scale estimation for robust visual tracking, in: British Machine Vision Conference, Bmva Press, 2014.

[13] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, Joost Van de Weijer, Adaptive color attributes for real-time visual tracking, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 1090–1097.

[14] Peng Gao, Qiquan Zhang, Fei Wang, Liyi Xiao, Hamido Fujita, Yan Zhang, Learning reinforced attentional representation for end-to-end visual tracking, Inf. Sci. 517 (2020) 52–67.

[15] Q. Guo, F. Wei, C. Zhou, H. Rui, W. Song, Learning dynamic siamese network for visual object tracking, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017.

[16] Sam Hare, Stuart Golodetz, Amir Saffari, Vibhav Vineet, Ming-Ming Cheng, Stephen L. Hicks, and Philip H.S. Torr, Struck: Structured output tracking with kernels, IEEE Trans. Pattern Anal. Mach. Intell. 38(10) (2015) 2096–2109..

[17] Joao F. Henriques, Rui Caseiro, Pedro Martins, Jorge Batista, Exploiting the circulant structure of tracking-by-detection with kernels, in: European conference on computer vision, Springer, 2012, pp. 702–715.

[18] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, High-speed tracking with kernelized correlation filters, IEEE Trans. Pattern Anal. Mach. Intell. 37(3) (2014) 583–596..

[19] Seunghoon Hong, Tackgeun You, Su.ha. Kwak, Bohyung Han, Online tracking by learning discriminative saliency map with convolutional neural network, in: International conference on machine learning, PMLR, 2015, pp. 597–606.
[20] Zhibin Hong, Zhe Chen, Chaohui Wang, Xue Mei, Danil Prokhorov, Dacheng Tao, Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 749–758.
[21] Dafei Huang, Lei Luo, Mei Wen, Zhaoyun Chen, and Chunyuan Zhang, Enable scale and aspect ratio adaptability in visual tracking with detection proposals, 2015..
[22] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Cehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, et al., The sixth visual object tracking vot2018 challenge results, in: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, pages 0–0, 2018..
[23] Matjaz Kukar, Igor Kononenko, et al., Cost-sensitive learning with neural networks, in: ECAI, vol. 15, pages 88–94. Citeseer, 1998..
[24] Hyeonseok Lee, Sungchan Kim, Sspnet: Learning spatiotemporal saliency prediction networks for visual tracking, Inf. Sci. 575 (2021) 399–416.
[25] Buyu Li, Yu Liu, and Xiaogang Wang, Gradient harmonized single-stage detector, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pages 8577–8584, 2019..
[26] Guiji Li, Manman Peng, Ke Nai, Zhiyong Li, Keqin Li, Reliable correlation tracking via dual-memory selection model, Inf. Sci. 518 (2020) 238–255.
[27] Meihui Li, Lingbing Peng, Wu. Tianfu, Zhenming Peng, A bottom-up and top-down integration framework for online object tracking, IEEE Trans. Multimedia 23 (2020) 105–119.
[28] Peixia Li, Boyu Chen, Wanli Ouyang, Dong Wang, Xiaoyun Yang, Huchuan Lu, Gradnet: Gradient-guided network for visual object tracking, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6162–6171.
[29] X. Li, C. Ma, B. Wu, Z. He, M.H. Yang, Target-aware deep tracking, IEEE (2019).
[30] P. Liang, E. Blasch, H. Ling, Encoding color information for visual tracking: Algorithms and benchmark, IEEE Trans. Image Process. 24 (12) (2015) 5630–5644.
[31] T.Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017.
[32] Xiankai Lu, Chao Ma, Bingbing Ni, Xiaokang Yang, Ian Reid, and Ming-Hsuan Yang, Deep regression tracking with shrinkage loss, in: Proceedings of the European conference on computer vision (ECCV), pages 353–369, 2018..
[33] C. Ma, J.B. Huang, X. Yang, M.H. Yang, Robust visual tracking via hierarchical convolutional features, IEEE Trans. Pattern Anal. Mach. Intell. (2017).
[34] C. Ma, J.B. Huang, X. Yang, M.H. Yang, Adaptive correlation filters with long-term and short-term memory for object tracking, Int. J. Comput. Vis. (2018).
[35] Chao Ma, Jia-Bin Huang, Xiaokang Yang, Ming-Hsuan Yang, Hierarchical convolutional features for visual tracking, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 3074–3082.
[36] M. Mueller, N. Smith, and B. Ghanem, A benchmark and simulator for uav tracking, in: European Conference on Computer Vision (ECCV16), 2016..
[37] Matthias Mueller, Neil Smith, Bernard Ghanem, Context-aware correlation filter tracking, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1396–1404.
[38] Hyeonseob Nam, Bohyung Han, Learning multi-domain convolutional neural networks for visual tracking, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4293–4302.
[39] Luca Oneto, Model selection and error estimation in a nutshell, Springer, 2020.
[40] S. Oron, A. Bar-Hillel, L. Dan, S. Avidan, Locally orderless tracking, Int. J. Comput. Vis. (2015) 213–228.
[41] Shi Pu, Yibing Song, Chao Ma, Honggang Zhang, and Ming-Hsuan Yang, Deep attentive tracking via reciprocative learning. arXiv preprint arXiv:1810.03851, 2018..
[42] Y. Song, M. Chao, L. Gong, J. Zhang, and M.H. Yang, Crest: Convolutional residual learning for visual tracking, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017..
[43] Qiang Wang, Jin Gao, Junliang Xing, Mengdan Zhang, and Weiming Hu, Dcfnet: Discriminant correlation filters network for visual tracking. arXiv preprint arXiv:1704.04057, 2017..
[44] Y. Wu, J. Lim, Ming Hsuan Yang, Object tracking benchmark, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1834–1848.
[45] Yi Wu, Jongwoo Lim, Ming-Hsuan Yang, Online object tracking: A benchmark, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 2411–2418.
[46] Fei Xue, Xin Wang, Shunkai Li, Qiuyuan Wang, Junqiu Wang, and Hongbin Zha, Beyond tracking: Selecting memory and refining poses for deep visual odometry, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8575–8583, 2019..
[47] L. Yang, J. Zhu, A scale adaptive kernel correlation filter tracker with feature integration, European Conference on Computer Vision (2014).
[48] T. Yang and A.B. Chan, Visual tracking via dynamic memory networks, IEEE Trans. Pattern Anal. Mach. Intell. 1–1 (2019) 99..
[49] T. Yang and Antoni B. Chan, Learning dynamic memory networks for object tracking, in: Springer, Cham, 2018..
[50] J. Zhang, S. Ma, S. Sclaroff, Meem: Robust tracking via multiple experts using entropy minimization, European Conference on Computer Vision (2014).

Huanlong Zhang received the Ph.D. degree from the School of Aeronautics and Astronautics, Shanghai Jiao Tong University, China, in 2015. He is currently an Associate Professor with the College of Electric and Information Engineering, Zhengzhou University of Light Industry, Henan, Zhengzhou, China. His research has been funded by the National Natural Science Foundation of China (NSFC), the Key Science and Technology. Henan Province et al. He has published more than 40 technical articles in refereed journals and conference proceedings. His research interests include pattern recognition, machine learning, image processing, computer vision, and intelligent human-machine systems.

Jiapeng Zhang was born in Xinxiang, Henan, China, in 1998. He is currently pursuing the degree with the Zhengzhou University of Light Industry, Zhengzhou, China. His research interests include pattern recognition, machine learning, image processing, computer vision, and intelligent human–machine systems.

Guohao Nie was born in Zhengzhou, Henan, China, in 1996. He is currently pursuing a degree with College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China. His research interests include pattern recognition, machine learning, image processing, computer vision, and intelligent human-machine systems.

Jilin Hu received the PhD degree from Aalborg University, Denmark, in 2019. He is currently an Assistant Professor at Aalborg University, Denmark. He has published several papers in PVLDB, ICDE, VLDB Journal, etc. His research interests include spatio-temporal data management, traffic data analytics, and machine learning. He was a session chair for PVLDB'20. He has been reviewers for several top tier journals, e.g., IEEE TKDE, VLDB Journal, IEEE TNNLS, Neurocomputing, etc. He was also PC members for CVPR'21, AAAI'21, APWeb'20.

W.J. (Chris) Zhang (Senior Member, IEEE) received the Ph.D. degree from the Delft University of Technology, Delft, The Netherlands, in 1994. He is currently a Full Professor with the Department of Mechanical Engineering and the Division of Biomedical Engineering, University of Saskatchewan, Saskatoon, SK, Canada. His current research interests include informatics, computer vision, and control of micromotion systems, and modeling and management of large complex systems. He was one of the most highly cited authors by Elsevier (China) in IEEE for the multiple consecutive years from 2015 to 2018. He is a Fellow of Canadian Academy of Engineering owing to his outstanding work on resilience engineering.