IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 6, 2020, accepted September 26, 2020, date of publication October 2, 2020, date of current version October 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3028308

Visual Tracking Jointly With Online and Offline Learning

AIHONG SHEN[1], SHENGJING TIAN[2], GUOQIANG TIAN[2], JIE ZHANG[3], AND XIUPING LIU[2]

[1]Department of Basic Courses, Criminal Investigation Police University of China, Shenyang 110854, China
[2]School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China
[3]School of Mathematics, Liaoning Normal University, Dalian 116029, China

Corresponding author: Jie Zhang (jzhang@lnnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61976040 and Grant 61702245.

ABSTRACT While recent approaches based on offline learning perform well in balancing the accuracy and speed of tracking, it is still non-trivial to accommodate a pre-trained model to an unseen target. In fact, online learning, which requires to capture specified target characteristics from one shot, is a unique attribute of single object tracking. How to favorably bridge the gap between the offline and online learning is of importance for tracking any unseen targets. In this work, in order to obtain sequence-specific information, we propose an online lightweight network consisting of feature adapting layer and ridge regression layer. Its key innovation is to interpret the ridge regression as one layer of the network. Furthermore, we integrate cross-similarity into the Siamese network and train it offline in an end-to-end manner to acquire the fine-grained local pattern of the target object. Through our effective fusion scheme for the offline and online procedures, our method can achieve considerable improvements on prevalent benchmarks.

INDEX TERMS Cross-similarity, few-shot learning, online learning, Siamese network, visual tracking.

I. INTRODUCTION

Visual tracking aims to estimate the state of the target continuously in the subsequent video frames with the solely prior knowledge from the first frame. It has been widely studied due to its applications in autonomous driving [1], [2], mobile robot system [3]–[5], human-computer interaction [6], and intelligent transportation [7]. The recent trend is to train convolutional neural networks (CNN) offline on some large-scale datasets [8]–[10], which cover a variety of changes in target and background. However, it is truly challenging to obtain generic feature representations due to inconsistencies among sequences. For example, objects of the same class can be foreground in one sequence or background in another sequence. On the other hand, online learning algorithms can capture sequence-specific characteristics, but limited by the balance of accuracy and speed. Therefore, it is meaningful and significant to bridge the gap between offline and online learning.

Over the past years, many trackers based on online learning have been widely explored, among which the representative approaches include sparse representing [11], dictionary learning [12], support vector machine [13], correlation filter (CF) [14], [15] and so on. These methods left such an indelible mark on the tracking research community. Here, it is particularly worth mentioning that, CF-based algorithms have attracted a lot of attention due to their competitive performance. The essence of them is to online learn a template filter in a Fourier domain utilizing the properties of the circular matrix with low computational load, and finally obtains the response map by convolving with the search region. Because of the mechanism of the online updating model, such methods can effectively capture variations of target and background during the tracking process. However, owing to the cyclical shift of the central image patch, all samples except the central one are born with undesired boundary effects. Afterward, Danelljan *et al.* [16] proposed the SRDCF to alleviate this problem using spatial regularization component. Based on this effective constrain, many improved trackers [15], [17] achieved more accurate performance but sacrifice tracking speed. In addition, other trackers taking advantage of reliability concept [18] and surrounding context [19] also obtain comparable results. However, all of the above methods focus on learning an observation model online merely resorting to the traditional methods, which still suffer from inevitable challenges including occlusion, scale, deformation, etc.

The associate editor coordinating the review of this manuscript and approving it for publication was Huiling Chen.

Contrary to the aforementioned approaches, many recent methods based on training a CNN offline are flourishing in the visual tracking realm. In particular, the Siamese network based trackers [20] have attracted a lot of attention from researchers with its ultra-fast tracking speed and promising performance. This kind of method tracks the target from the coming frame by offline learning a generically matching function that is designed to cope with various challenges. Its architecture is composed of two shared fully convolution branches, which takes the template and search region as inputs and predicts a response map to locate the target. For instance, He *et al.* [21] used a twofold Siamese network to learn the semantic and appearance features simultaneously, which can complement each other in a mutually beneficial way. In order to enhance the discrimination capacity and adaptability, Wang *et al.* [8] proposed residual attentional Siamese network by making extensive use of different attention mechanisms: residual attention, channel attention, and general attention. In the light of the outstanding performance of the correlation filter, Valmadre *et al.* [22] integrated it into the Siamese network and trained their CFNet in an end-to-end manner. Based on the Siamese network, Lee [23] improved the speed and accuracy of the tracking model by introducing the estimation of scale and angle estimation, and Li *et al.* [9] replaced AlexNet with ResNet and successfully trained a ResNet-driven Siamese tracker with significant performance gain. However, these trackers ignore the temporal information and merely rely on feature representation learned offline to conduct tracking. To address this problem, Yang and Chan [24] trained LSTM offline to enable the Siamese network to adaptively update the template. Zhu *et al.* [25] made use of a pre-trained optical flow network to warp feature representation in the previous frames, dynamically obtaining a new template. And Guo *et al.* [26] thoughtfully suppressed background distractors and learned target appearance variation via a fast transformation module. Nevertheless, these trackers are committed to learning a fixed pattern of updating template offline through abundantly annotated sequences. When the tracking scenario is quite different from the training domain, it will show unexpected performance since lacks the ability to capture sequence-specific information. Therefore, Danelljan *et al.* [27] introduced an online classification module solved by an optimization strategy based on conjugate gradient and Gauss-Newton. But it still struggles with some challenging factors without the rectification of IoUNet [28].

Motivated by the above observations, in this paper, we propose a tracking approach that smoothly combines the merits of online learning and offline learning. Our contributions contain three aspects. Firstly, different from the existing online learning based trackers, which resort to traditional machine learning, we incorporate the ridge regression into the convolution network for capturing the sequence-specific information. Actually, our purpose is to treat standard ridge regression as part of the convolution network and enable a quick adaptation by means of its fast convergence [29]. In particular, we reformulate the online tracking as a few-shot problem and interpret the closed-form solution of the ridge regression as one differentiable layer of the online network. Secondly, in order to discriminate the target suffering from multiple interferences in one sequence, we also integrate the calculation of the cross similarity among patches into the Siamese network to acquire some prior knowledge offline. Our goal is to accomplish the procedure of the kernelized ridge regression (KRR) mentioned in [30], and construct a learnable kernel function based on local information. As opposed to traditional patch based trackers, our approach trains the whole network from end to end and considers the structural relationship among all local patches. Finally, to harness their combined strengths of online and offline learning, we introduce an effective fusion scheme to narrow the gap between them. Extensive experiments on prevalent benchmarks show that bridging the gap between online and offline learning is beneficial to generic visual tracking.

II. METHODOLOGY

This section is organized as follows. Section II-A details the online learning model consisting of feature adapting layer and ridge regression layer. Section II-B introduces the offline learning model with the dense cross-similarity. Section II-C presents the effective fusion scheme.

A. ONLINE LEARNING

Considering that generic visual tracking requires the model to possess the generalization ability to unseen objects, we adopt an online learning algorithm to capture the sequence-specific information when tracking a specified target. In contrast to those methods that learn an observation model online based on advanced machine learning algorithms [11]–[15], we aim to realize a strong alliance between traditional classifier and convolution network. Taking inspiration from few-shot learning [29], we interpret the ridge regression as one layer of the online network to quickly adapt to any unseen categories online.

In particular, as shown in Figure 1, our online learning model Ψ is composed of two layers: feature adapting layer (FAL) and ridge regression layer (RRL), which are parameterized by ω_1 and ω_2 respectively. As for the ω_1 of the FAL, we use the standard 1×1 convolution kernel followed by batch normalization (BN) and ReLU. Regarding the ω_2 of the RRL, we instead leverage the closed-form solution of the ridge regression to learn it directly. More specifically, the ridge regression is formulated as follows

$$\omega_2 = \arg\min_{\omega_2} \|X\omega_2 - Y\| + \lambda\|\omega_2\|^2. \tag{1}$$

Its closed-form solution can be written as

$$\omega_2 = (X^\top X + \lambda E)^{-1} X^\top Y, \tag{2}$$

where each row of $X \in R^{M \times K}$ is a sample represented by a K-dimension feature vector, λ is the factor for controlling the regularization term, E is the identity matrix, and $Y \in R^{M \times 1}$ is the ground-truth label. To serve as a convolution kernel,

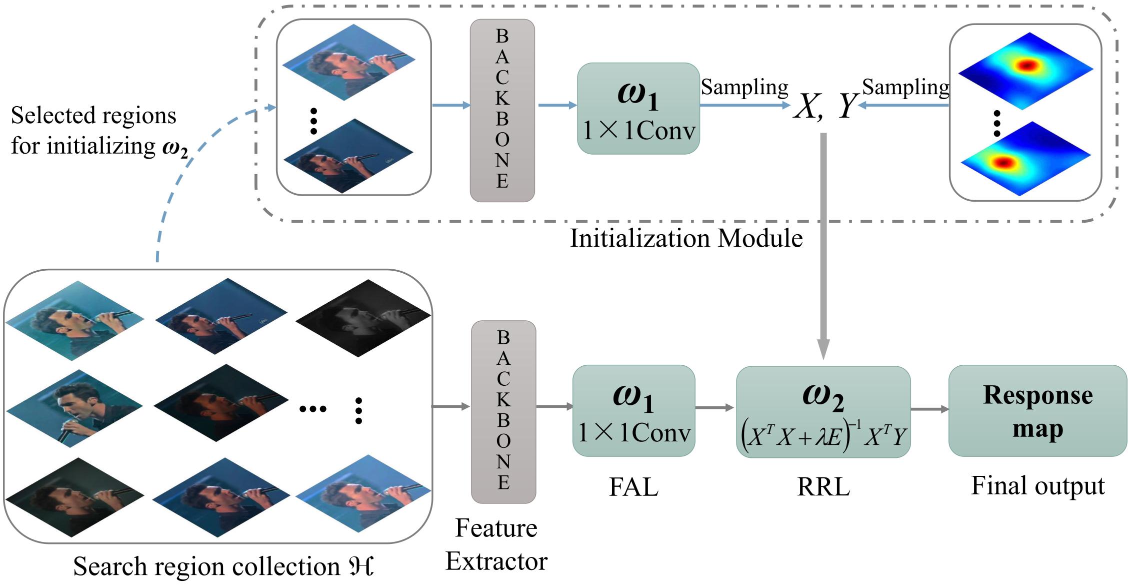


FIGURE 1. Online learning framework. Firstly, the search region collection $\mathcal{H}$ is obtained by some augmentation operations, and we extract their feature vectors via pre-trained backbone. The details about the search region are introduced in Section III-A. Next, some search regions are selected to initialize the RRL with the formulation of $\omega_2 = \texttt{Reshape}((X^\top X + \lambda E)^{-1}X^\top Y)$. This step corresponds to the initialization module in dashed box of the above figure. Finally, the rest elements of $\mathcal{H}$ are utilized for online learning by Eq. (4).

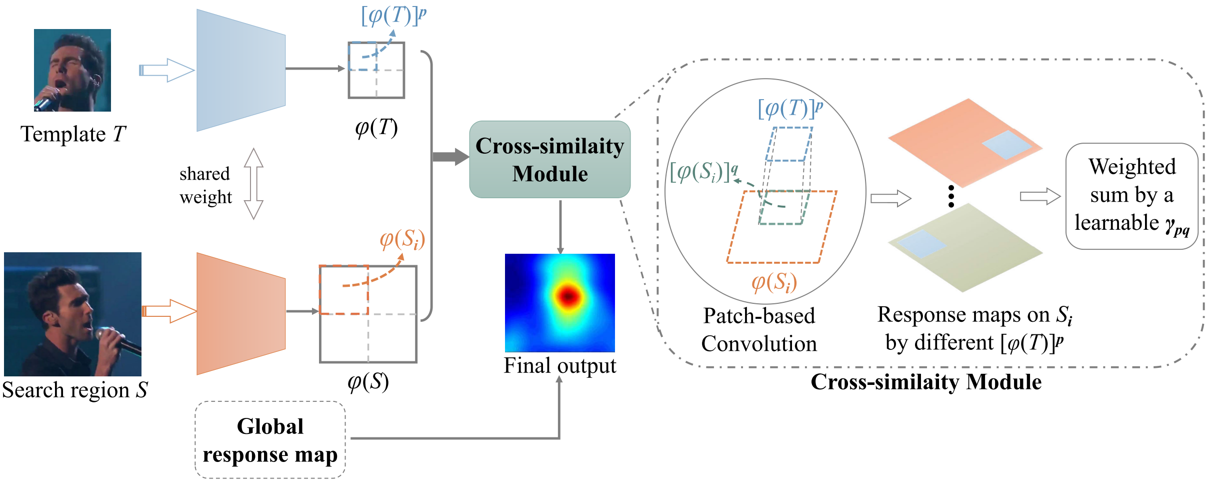


FIGURE 2. Offline learning architecture. The cross-similarity module first consumes a set of paired samples $(\varphi(T), \varphi(S_i))$, and produces a response map with a learnable weight γ_{pq}. To leverage the local pattern, the final output is the weighted sum of the patch-based and global response map. The global response map is generated directly using global feature.

we further reshape ω_2 by the $\texttt{Reshape}$ operation, *i. e.*, $\omega_2 = \texttt{Reshape}((X^\top X + \lambda E)^{-1}X^\top Y)$. Here, the operation $\texttt{Reshape} : R^{K\times 1} \longrightarrow R^{w\times h\times c}$ projects a flattened vector to a desired tensor ($K = w \times h \times c$). And X is obtained by sliding window on search regions, Y is its corresponding label generated by Gaussian distribution. Mathematically, our online model can be written as

$$\Psi(H;\omega) = \sigma_2(\omega_2 \star \sigma_1(\omega_1 \star H)), \tag{3}$$

where $\omega = \{\omega_1, \omega_2\}$, σ_1 and σ_2 are activation functions, and H represents a search region's feature extracted from a CNN backbone. The output of Ψ is a response map where the location that the target lies in should have the highest value. Because the proposed model needs updating parameters by back-propagation, we adopt the weighted square error as online loss, *i. e.*,

$$L_{on} = \sum_{i=1}^{|\mathcal{H}|} \beta_i \|\Psi(H_i;\omega) - G_i\|_2^2, \tag{4}$$

where $|\mathcal{H}|$ is the element number of the search region collection $\mathcal{H}$, G_i is the ground-truth response map of the i-th sample. In the first frame, we need to split the collection $\mathcal{H}$ to

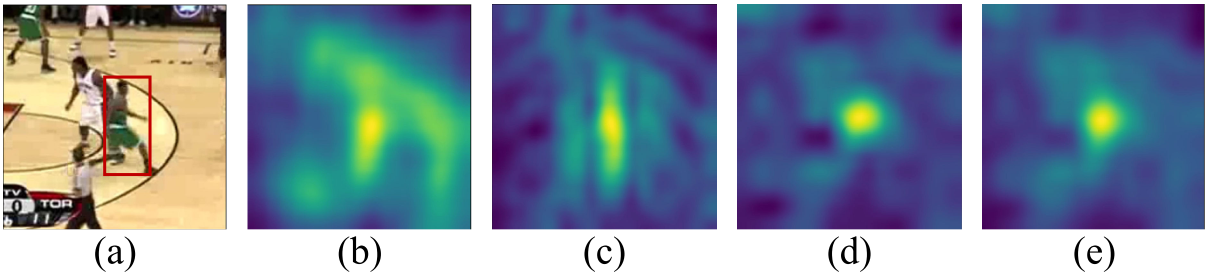


FIGURE 3. Visualization of response maps. (a) is the search region. (b)-(e) are generated by a global descriptor [20], our cross-similarity based offline learning, our online learning and our final fusion, respectively.

two sets: one for initialization with Eq. (2) and the other one for training model with Eq. (4). Subsequently, the collection $\mathcal{H}$ serves as a memory bank and its elements are dynamically replaced with a new one cropped from the current frame using first-in first-out rule. Based on this dynamical $\mathcal{H}$, we update our online network every ten frames. In this way, we can incorporate the ridge regression into the proposed network. And it can be trained in an end-to-end manner when the back-propagation of a batch of inverse matrices is implemented.

B. OFFLINE LEARNING

Generally, individual sequence involves different challenges including occlusion, blurring, cluster, deformation, illumination, and scale variation, etc. It is truly challenging for online learning to quickly accommodate with various scenarios within limited samples cropped from the first frame. Therefore, it is necessary to learn some common information offline from abundant sequences as the result of its indeed benefits to the general representation of any unseen scenes. Recently, most trackers using deep networks focus on extracting a global appearance descriptor and are then combined with other techniques such as attention [8], correlation filter [22], and LSTM [24]. However, for generic visual tracking, a global descriptor from the deep network is easily subject to non-rigid deformation and partial occlusion, as shown in Figure 3 (b). In this work, we consider the local pattern similarities between the template and candidate instance utilizing the Siamese network. Moreover, different from those traditional patch-based methods [31], [32], which rely more on pre-trained or hand-crafted features and ignore the relationship between local patches, we aim to incorporate the philosophy of patch-based trackers to a deep network and accomplish this whole procedure in an end-to-end manner.

At first, we review fully convolutional Siamese network (SiamFC [20]) briefly. SiamFC aims to learn a generically applicable matching function Φ and finally output a response map. Its model is as follows

$$\Phi(T, S) = \phi(T) \star \phi(S) + b\dagger, \tag{5}$$

where T and S are tracked target patch (template) and search region respectively, ϕ is the feature extractor, $\star$ represents the standard convolution operation, every elements of the matrix $\dagger$ is 1, and b is an adjusting bias. Supposing that S_i is one instance cropped from search region S, SiamFC actually calculates the similarity between T and S_i via learning a kernel function $\mathcal{K}(T, S_i) = (\phi(T) \odot \phi(S_i) + b)$. Here $\odot$ denotes the element-wise multiplication. Therefore, its training loss can be rewritten as

$$\theta = \arg\min_{\theta} \frac{1}{2} \sum_{i=1}^{M} \mathcal{L}(y_i, k_i), \tag{6}$$

where $y_i \in \{-1, 1\}$ is the label, k_i is the short form of $\mathcal{K}(T, S_i)$, M is the number of instances, $\mathcal{L}$ is the logistic loss, and θ is the parameters of ϕ. In practice, for generic visual tracking, a learnable kernel function can project samples into a more discriminatory space when compared with the hand-crafted kernel function such as Gaussian kernel and polynomial kernel. However, this kernel function $\mathcal{K}$ of SiamFC starts with a holistic view which cannot capture some local pattern.

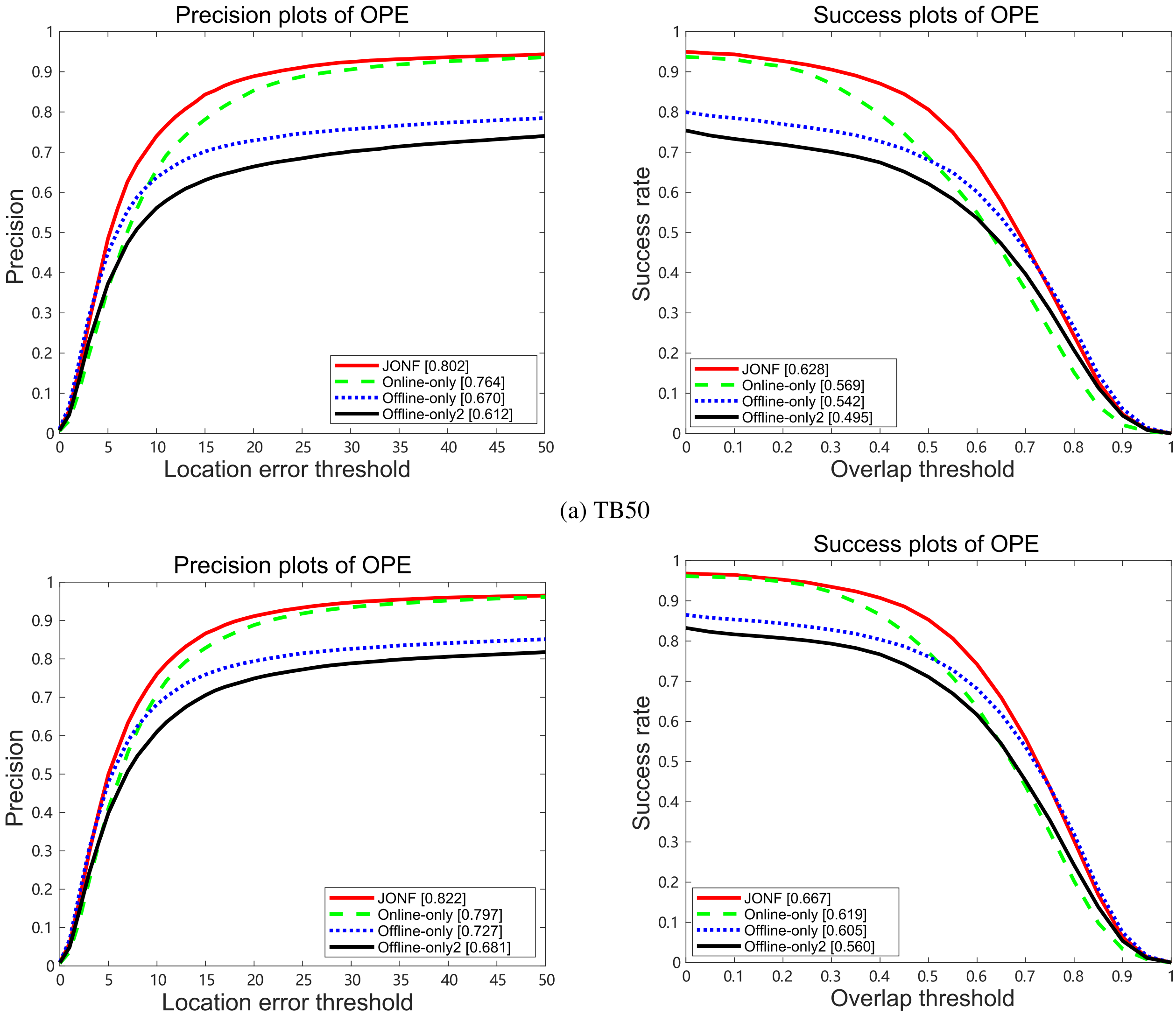
In this work, inspired by the consideration of spatial layouts [30], we propose a Siamese architecture for calculating the dense cross-similarity, which can be also thought of as a novel kernel function exploiting detailed information when compared with the above SiamFC. Specially, in cross-similarity module, we divide the template T into P patches and each instance S_i into Q patches. Given the respective feature representation $\varphi(T)$ and $\varphi(S_i)$, our kernel value between the template T and instance S_i can be calculated by

$$\hat{k}_i = \sum_{p=1}^{P} \sum_{q=1}^{Q} \gamma_{pq} [\varphi(T)]^p \odot [\varphi(S_i)]^q, \tag{7}$$

where γ_{pq} represents the weight of a paired patches consisting of the p-th patch of T and the q-th patch of S_i. More concretely, for each paired $\varphi(T)$ and $\varphi(S_i)$, our cross-similarity module conducts patch-based convolution, yielding response maps. Then, with the learnable parameter γ_{pq}, we calculate the weighted sum to mine the relationship among all local patches. The proposed pipeline is shown in Figure 2.

C. FUSION SCHEME

In order to make up for the difference between online and offline learning, we present an effective fusion scheme in this section. Our goal is to make full use of the consistency



(b) OTB100

FIGURE 4. Self-contrast comparisons on the TB50 and OTB100. The results on TB50 are shown in the first row, and OTB100 in the last row. The trackers in the precision plots (first column) are ranked by DP under a given threshold (20), and in the success plot (second column) are ranked by AUC.

knowledge that offline learning and online learning have mastered. This can be reflected in the fact that the response map generated by fusing their respective one has an unimodal characteristic. Furthermore, its essence is to measure the quality of fusion response map. To do this, we model the following objective

$$\begin{aligned} \min_{\beta_1,\beta_2} \quad & \Delta = \xi_1 \mathcal{M}(\beta_1 R^{on} + \beta_2 R^{off}) + \xi_2(\beta_1^2 + \beta_2^2), \\ s.t.\ & \beta_1 + \beta_2 = 1; \quad a \leq \beta_1 \leq b \end{aligned} \tag{8}$$

where R^{on} and R^{off} are response maps produced by online and offline respectively, β_1 and β_2 are weight coefficients, ξ_1 and ξ_2 are balance factors. Here, $\mathcal{M}$ is the confidence margin [33], $\mathcal{M}(R) = -(R(t_\star))^\mu \sum_{t_\star \neq t} R(t)\log(R(t))$, where $R(t_\star)$ is the peak value, μ is the power. The lower the value of this objective function Δ, the better the quality of its corresponding response map. The regularization term of $\xi_2(\beta_1^2 + \beta_2^2)$ is to balance the weights of R^{on} and R^{off}. For fast speed, we jointly consider scale variation and solve it in a discrete manner. Algorithm 1 shows its details.

III. EXPERIMENTS

A. IMPLEMENT DETAILS

We conducted experiments with PyTorch and Python2.7 on a PC equipped with a GTX 1080Ti, 32GB RAM, and 4.00 GHz Intel Core i7-4790K CPU.

To train the parameters ω_1 and ω_2 of the online network, we first generated the collection $\mathcal{H}$ in the first frame with the help of some augmentation operations, such as shifting, scaling, rotating, blurring, etc. This collection initially includes 20 search regions and appends new one periodically. The size of each region is 255 × 255. For initializing

Algorithm 1 Fusion Scheme

```
Require: Online and offline response maps in different scale: $\{R_s^{off}\}$, $\{R_s^{on}\}$, $s = 1, 2, 3$
Ensure: Final response map: $R$
$\hat{\Delta} \leftarrow \inf$; $R \leftarrow \mathbf{0}$;
for $\beta_1 \in [a, b]$ do
    for $s \in \{1, 2, 3\}$ do
        $\beta_2 \leftarrow 1 - \beta_1$;
        $R_s \leftarrow \beta_1 R_s^{on} + \beta_2 R_s^{off}$;
        $\Delta \leftarrow$ calculate by $\mathcal{M}(R_s)$;
        $\Delta \leftarrow \xi_1 \Delta + \xi_2(\beta_1^2 + \beta_2^2)$;
        if $\Delta < \hat{\Delta}$ then
            $\hat{\Delta} \leftarrow \Delta$;
            $R \leftarrow R_s$;
        end if
    end for
end for
```

TABLE 1. Self-contrast of online learning. We report the value of AUC/DP@20 [38]. Here, DP@20 is the location precision at threshold 20. The best results are bold.

Backbones	TB-50		OTB-100	
	Online-RRL	Online-CONV	Online-RRL	Online-CONV
AlexNet	54.0/69.8%	30.0/41.3%	59.6/74.8%	35.8/47.7%
ResNet-18	56.9/74.6%	39.9/54.6%	60.9/77.5%	41.7/55.5%
ResNet-50	**57.0/76.4%**	45.9/61.3%	**61.9/79.7%**	49.2/63.6%

ω_2, we fed 3 search regions which have same content but different scale into Eq. (2). For initializing ω_1, we adopted Xavier initialization [34]. And the online model utilizes ResNet [35] pre-trained on ImageNet as its backbone to extract features. Finally, the online learning problem is solved by ADAM optimizer [36] with learning rate $= 10^{-3}$, batch size $=$ 17 and epoch $=$ 100. During tracking, $\mathcal{H}$ is treated as the memory bank and dynamically updated using first-in first-out rule when the number of its elements reaches 64. And we fine-tuned ω_1 and ω_2 every ten frames.

As for our offline network, we used Siamese network whose backbone is AlexNet to learn patch-based prior knowledge. Its architecture is the same as SiamFC, but we insert the proposed cross-similarity module into the top of SiamFC. The sizes of the template and search region are 127×127 and 255×255 respectively. The training set is ILSVRC2015-VID [37]. And we trained this offline network using SGD optimizer with learning rate $=$ 10^{-3}, batch size $=$ 8 and epoch $= 30$.

B. DATASETS AND METRICS

We comprehensively tested tracking performance on multiple datasets including VOT2018 [39], UAV123 [40], GOT-10k [41], TB50 and OTB100 [38]. On VOT2018, the expected average overlap (EAO), accuracy (A) and Robust (R) are utilized to measure tracking results. Please refer to [39] for the details of these three metrics. On TB50 and OTB100, the tracking results are reported on two plots: success plot and precision plot. For success plot, the area under curve (AUC) is used to rank the evaluated trackers. For precision plot, the distance precision (DP) rate at a certain threshold (20 pixels) is utilized to rank the evaluated trackers. On GOT-10k, the metrics are AO and SR. Here, AO is the average value of overlap rates between tracking results and ground-truths overall frames, and SR is the percentage of successfully tracked frames where overlap rates are above a threshold. On UAV123, we use the AUC as its metric.

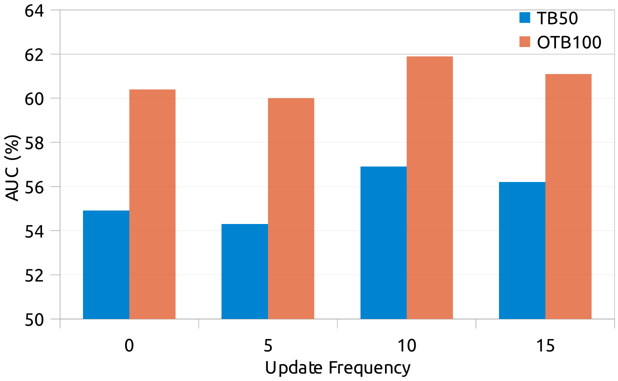


FIGURE 5. The AUC of different frequency of updating online model. “0” means no update for online model.

C. ABLATION STUDIES

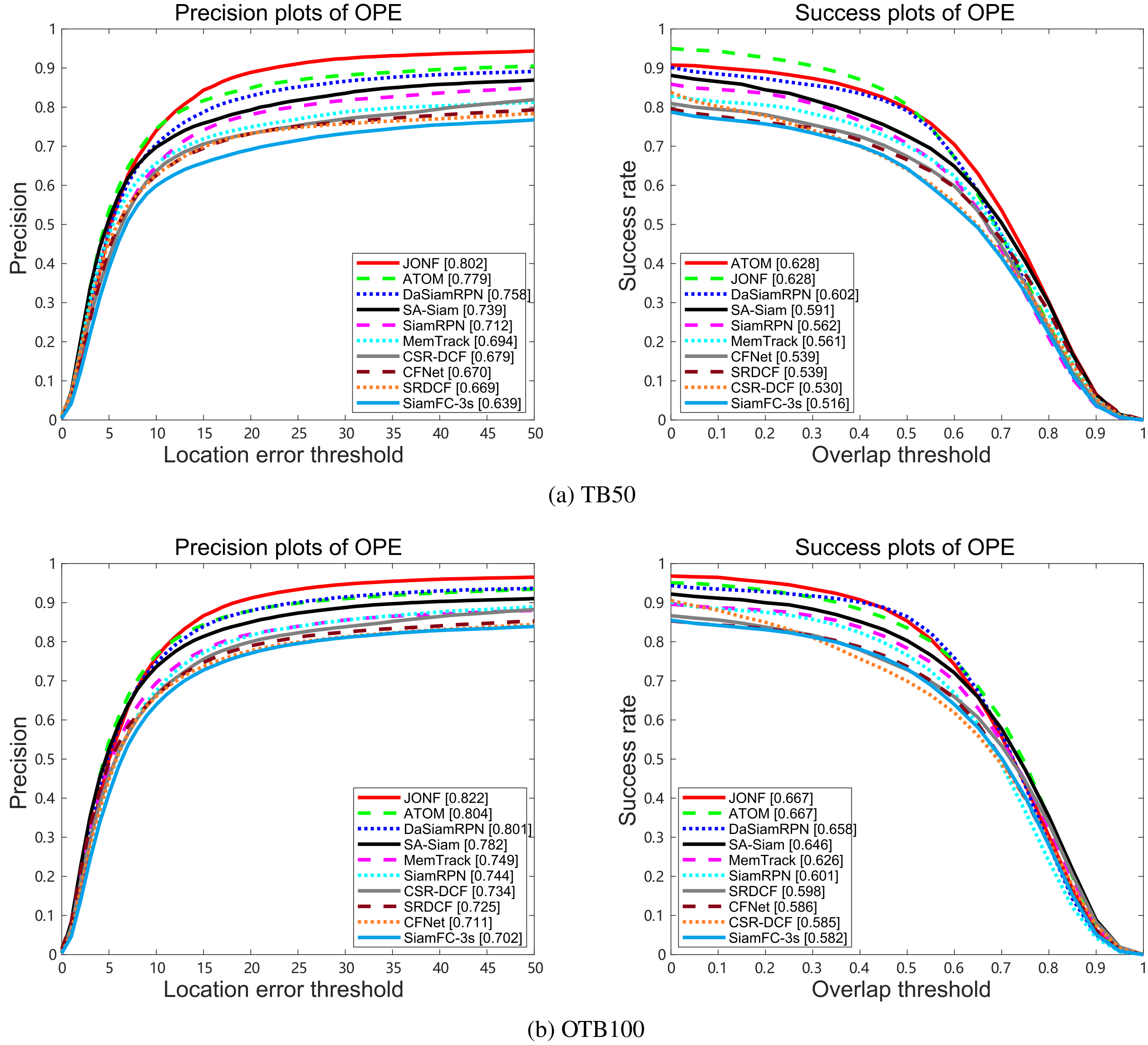
To demonstrate the effectiveness of different modules, we carefully designed some self-contrast experiments and tested them on both TB50 and OTB100 benchmarks.

1) JOINTLY OR NOT

We investigated the impact of bridging the gap between online and offline learning. In Figure 4, Online-only represents a variant that only uses the online model mentioned in Section 2.1. Offline-only conducts tracking only by the offline model mentioned in Section 2.2. And JONF is the proposed tracker that combines offline learning and online learning. As we can see, according to the AUC on TB50 and OTB100, JONF achieves superior performances when compared with Online-only and Offline-only. The main reasons are as follows. Offline model has learned some common representation capabilities under various scenario but lacks the discrimination of a specified instance, so it is easily disturbed by similar objects and obstructions. Online model focuses on classifying samples which are acquired from the tracked sequence but leads to poor performance when tracking drift happens. Eventually, benefiting from our fusion strategy, these two models can achieve complementary advantages, which highlights the necessity of making up the difference between online and offline learning.

2) RRL OR NOT

In order to verify the validation of our ridge regression layer proposed in Section 2.1, we test their performances with or



(a) TB50

(b) OTB100

FIGURE 6. State-of-the-art comparisons on the TB50 and OTB100. The results on TB50 are shown in the first row, and OTB100 in the last row. The trackers in the precision plots (first column) are ranked by DP under a given threshold (20), and in the success plot (second column) are ranked by AUC.

without RRL. In Table 1, Online-RRL represents our tracker with RRL. Online-CONV is the variant that replaces the RRL with standard 2D convolution. The results in Table 1 show that it is effective to interpret the closed-form of ridge regression as part of the online network.

3) BACKBONE

We also studied the effect of different backbones used in online learning. In particular, three pre-trianed CNN backbones were used to extract features: AlexNet, ResNet-18 and ResNet50. As shown in Table 1, ResNet-50 gets the best performances and ResNet-18 is at the second place. However, their relative improvements are slight. For instance, JONF-RRL using ResNet-50 only obtains 0.01 and 0.02 improvements when compared to ResNet-18 and AlexNet. This illustrates that, the depth increment of backbone is benefit to extract more discriminative feature, but has a slight influence on our online model.

4) RELATIONSHIP γ_{pq}

We designed two solutions to consider the relationship γ_{pq} among local patches in Eq. 7. Offline-only assigns every local patch of T equal confidence while treating local patches of S_i differently. That is, $\gamma_{1q} = \gamma_{2q} = \ldots = \gamma_{Pq}$ but $\gamma_{p1} \neq \gamma_{p2} \neq \ldots \neq \gamma_{pQ}$. Offline-only2 directly learns γ_{pq} using 2D convolution kernel without any constrains. As shown in Figure 4, Offline-only is superior to Offline-only2. The reason is that the template T is a reliable cue, and equal confidence can ensure that every local pattern of the template gets the attention it deserves.

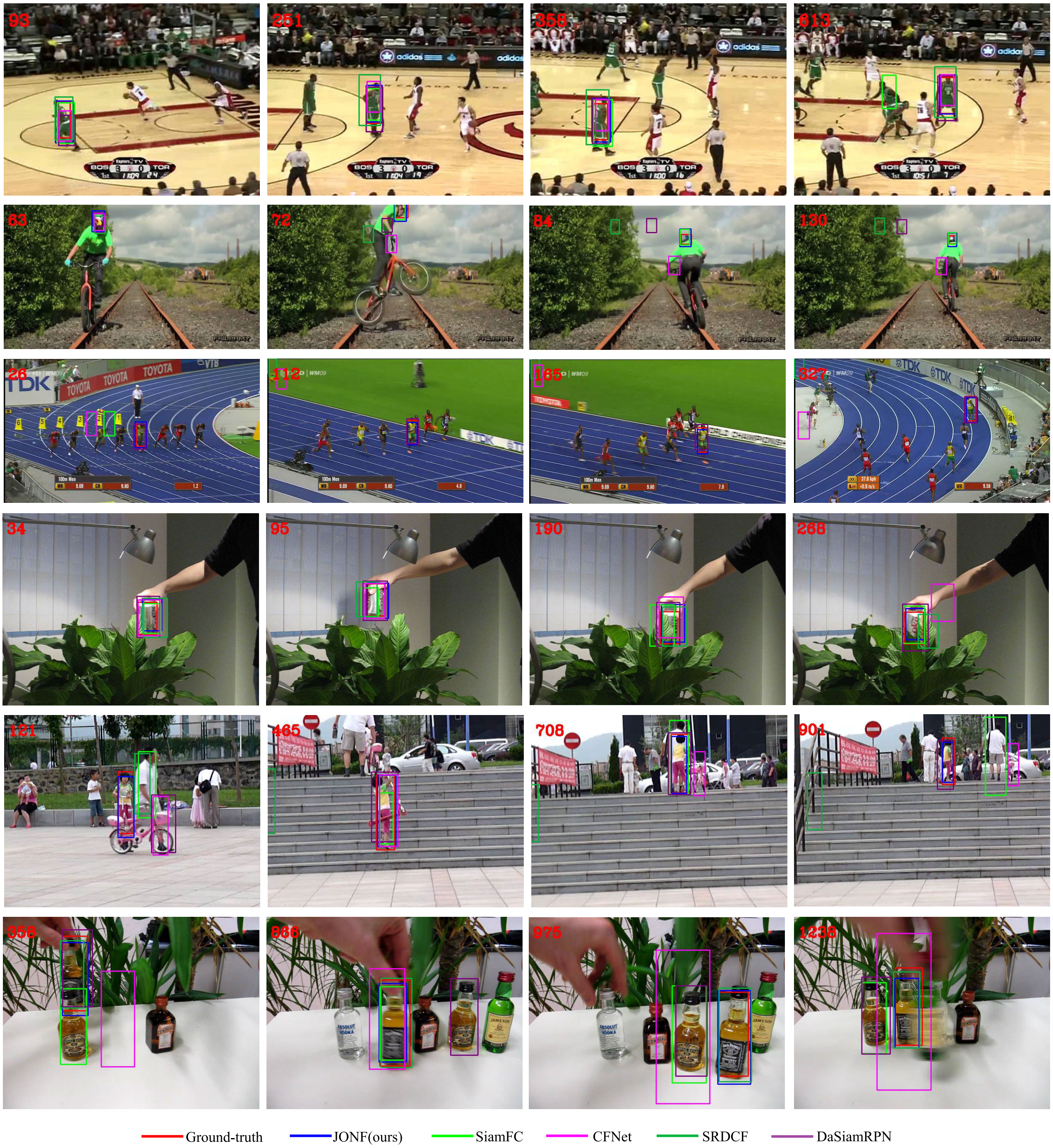


FIGURE 7. Visual tracking results on OTB benchmark. We show some representative sequences that involve the challenges of occlusion, background clutters, blur, out-of-plane rotation, and fast motion. The bounding boxes with different color correspond to different trackers. Frame number is shown on the left-top of each image.

5) UPDATE FREQUENCY

Considering the update frequency factor affects catching the information of the target from different video frames, we design different experiments in order to research the update frequency of the online model. Firstly, we only initialize ω_1 from FAL and ω_2 from RRL by the first frame. Secondly we update online model every 5, 10, 15 frames with ADAM optimizer and learning rate $= 10^{-3}$, epoch $=$ 2, batch size $=$ 8. Figure 5 shows that updating model is very important for improving tracking performance. Frequent updating make online model unstable due to interference terms, while slow updating maybe lead to drifting because of appe-arance variation and deformation.

TABLE 2. The performance comparison between the proposed online (and offline) algorithm and its corresponding recent online (and offline) approaches. The AUC metric on OTB100 is reported tersely. Our results are highlighted in bold.

Online methods	Online-only ours	KCF [14]	SRDCF [16]	CSR-DCF [18]	Staple [42]	ACFN [13]	LMCF [43]
	61.9%	47.6%	59.8%	58.5%	57.8%	57.1%	58.0%
Offline methods	Offline-ony ours	SiamFC [20]	SiamFC-tri [44]	DSiam [26]	CFNet [22]	SINT [45]	StructSiam [46]
	60.5%	58.2%	59.0%	56.1%	58.6%	58.2%	62.1%

TABLE 3. State-of-the-art comparison on the VOT2018 benchmark in terms of EAO, A and R. Here, A and R are accuracy and robustness, respectively. Our results are highlighted in bold.

	JONF	ECO [15]	CCOT [17]	SiamRPN [47]	DSiam [26]	SiamFC [20]	CSR-DCF [18]	SA-Siam [21]
EAO ↑	**0.319**	0.280	0.267	0.383	0.196	0.188	0.256	0.286
A ↑	**0.467**	0.484	0.494	0.586	0.512	0.503	0.491	0.533
R ↓	**0.297**	0.276	0.318	0.276	0.646	0.585	0.356	0.337

TABLE 4. State-of-the-art comparison on the UAV123 dataset in terms of AUC score.

	JONF	ECO [15]	CCOT [17]	MDNet [48]	SRDCF [16]	Struck [13]	SiamRPN [47]
AUC(%)↑	**54.0**	53.7	51.7	52.8	47.3	38.1	57.1

D. STATE-OF-THE-ART COMPARISON

In this section, we compared our JONF with state-of-the-art trackers on some public benchmarks.

1) OTB [38]

Figure 6 shows the comparison results with some state-of-the-art methods on TB50 and OTB100. Overall, the proposed JONF obtains competitive performance. On OTB100, our JONF achieves the best performance (82.2/66.7%). Specifically, it is superior to SA-Siam [21], DaSiamRPN [26], and MemTrack [24]. Moreover, it surpasses the ATOM (80.4/66.7%) in DP@20 while achieving competitive results in AUC.

For comprehensive evaluation, the visualization tracking results from some representative sequences are shown in Figure 7, where the target objects suffer from occlusion, background clutters, blur, out-of-plane rotation, and fast motion. As we can see, the proposed JONF obtains considerably stable results on these sequences when compared with SiamFC [20], SRDCF [16], CFNet [22], and DaSiamRPN [49]. Specifically, as shown in the second row of Figure 7, DaSiamRPN drifts to the background due to out-of-plane rotation, whereas JONF can track the target object successfully.

We also compared the proposed online (and offline) algorithm with its corresponding recent online (and offline) approaches. For online learning methods, we took into consideration some representative CF-based trackers including KCF [14], SRDCF [16], CSR-DCF [18], Staple [50], ACFN [42], and LMCF [43]. As shown in Table 2, Our online learning method (denoted as online-only) achieves 61.7% according to AUC, which surpasses all of aforementioned CF-based trackers. On the other hand, because our offline learning method (denoted as offline-only) takes SiamFC [20] as the baseline, we thus compared it with some representative approaches based on the Siamese network including CFNet [22], SiamFC-tri [44], DSiam [26], SINT [45], and StructSiam [13]. Table 2 shows that our offline-only is better than these Siamese network based trackers except StructSiam. Note that, this work concentrates on bridging the gap between offline and online learning, our offline learning model is trained without the adjustment of hyper-parameters and extra dataset like Youtube-BB.

2) VOT2018 [39]

Table 3 tersely reports the expected average overlap (EAO), accuracy (A), and Robust (R) of some advanced trackers on VOT2018. Overall, the proposed JONF ranks second. In particular, according to the EAO, our JONF obtains comparable performance when compared with SiamRPN. Note that SiamRPN contains a regression branch for the resulting bounding boxes which is more advantageous for adjusting the target state in such a scale-varying dataset.

TABLE 5. State-of-the-art comparison on the GOT-10k benchmark in terms of average overlap (AO), and success rates (SR) at overlap thresholds of 0.5 and 0.75. Our results are highlighted in bold.

	JONF	ECO [15]	CCOT [17]	MDNet [48]	SRDCF [16]	SiamFC [20]	BACF [19]	CFNet [22]
AO(%) ↑	**39.1**	31.6	32.5	29.9	23.6	34.8	26.0	27.0
$\mathrm{SR}_{0.5}$(%) ↑	**40.8**	30.9	32.8	30.3	22.7	35.3	26.2	22.5
$\mathrm{SR}_{0.75}$(%) ↑	**11.4**	11.1	10.7	9.9	9.4	9.8	10.1	7.2

Moreover, the JONF has a considerable improvement against ECO [15], DSiam [26], and CSR-DCF [18].

3) GOT-10k [41]

This benchmark provides wide coverage of moving objects in the wild. It does not publish the ground-truth of the test set, and requires researchers to upload the test results to the server to avoid excessive hyper-parameter adjustment. Table 5 shows the comparison results on GOT-10k. According to the AO, $SR_{0.5}$ and $SR_{0.75}$, JONF (39.1/40.8/11.4) considerably outperforms other online learning based methods, such as CCOT [17] and MDNet [48].

4) UAV123 [40]

UAV123 is designed for a specific tracking task whose sequences are sampled from the view of unmanned aerial vehicles. We also evaluated the proposed method to demonstrate its generalization ability. Table 4 presents the AUC score on the UAV123. SiamRPN has the best performance and our JONF ranks in the second place. The main reasons are that 1) many targets in the wild suffer from scale changing, and the trackers with the state estimation branch like SiamRPN have an obvious advantage over one determined only by response map; 2) more annotated dataset (*i. e.*, Youtube-BB) was utilized in addition to ILSVRC2015-VID during training.

IV. CONCLUSION

We proposed a novel method for making up for the difference between online and offline learning. For online learning, we interpreted the closed-form solution of the ridge regression as part of our lightweight network to enable quick adaption for unseen objects. For offline learning, we applied the philosophy of KRR based on cross-similarity into the Siamese network and accomplished this procedure in an end-to-end manner. And an effective fusion scheme was presented to bridge their gap. The ablation studies prove the validation of each component, and comprehensive experiments show that our approach achieved considerable improvements on multiple public benchmarks.

REFERENCES

[1] J. Cao, C. Song, S. Peng, S. Song, X. Zhang, and F. Xiao, "Trajectory tracking control algorithm for autonomous vehicle considering cornering characteristics," *IEEE Access*, vol. 8, pp. 59470–59484, 2020.

[2] M. Li, Y. Xu, M. Lei, and B. Zhou, "Velocity tracking control based on Throttle-Pedal-Moving data mapping for the autonomous vehicle," *IEEE Access*, vol. 7, pp. 176712–176718, 2019.

[3] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Access*, vol. 24, pp. 237–267, 2002.

[4] L.-H. Juang and J.-S. Zhang, "Visual tracking control of humanoid robot," *IEEE Access*, vol. 7, pp. 29213–29222, 2019.

[5] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 278–288, Jun. 1991.

[6] U. Sambrekar and D. Ramdasi, "Human computer interaction for disabled using eye motion tracking," in *Proc. Int. Conf. Inf. Process. (ICIP)*, Dec. 2015, pp. 745–750.

[7] H. Sabirin and M. Kim, "Moving object detection and tracking using a spatio-temporal graph in H.264/AVC bitstreams for video surveillance," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 657–668, Jun. 2012.

[8] Q. Wang, Z. Teng, J. Xing, J. Gao, W. Hu, and S. Maybank, "Learning attentions: Residual attentional siamese network for high performance online visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4854–4863.

[9] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4282–4291.

[10] S. Tian, X. Liu, M. Liu, S. Li, and B. Yin, "Siamese tracking network with informative enhanced loss," *IEEE Trans. Multimedia*, early access, Mar. 6, 2020, doi: 10.1109/TMM.2020.2978636.

[11] X. Mei and H. Ling, "Robust visual tracking using ℓ_1 minimization," in *Proc. IEEE Conf. Int. Conf. Comput. Vis.*, Oct. 2009, pp. 1436–1443.

[12] N. Wang, J. Wang, and D.-Y. Yeung, "Online robust non-negative dictionary learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 657–664.

[13] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. S. Torr, "Struck: Structured output tracking with kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2096–2109, Oct. 2016.

[14] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[15] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6638–6646.

[16] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4310–4318.

[17] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 472–488.

[18] A. Lukezic, T. Vojir, L. C. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6309–6318.

[19] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1135–1143.

[20] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 850–865.

[21] A. He, C. Luo, X. Tian, and W. Zeng, "A twofold siamese network for real-time object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4834–4843.

[22] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2805–2813.

[23] D.-H. Lee, "One-shot scale and angle estimation for fast visual object tracking," *IEEE Access*, vol. 7, pp. 55477–55484, 2019.

[24] T. Yang and A. B. Chan, "Learning dynamic memory networks for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 152–167.

[25] Z. Zhu, W. Wu, W. Zou, and J. Yan, "End-to-end flow correlation tracking with spatial-temporal attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 548–557.

[26] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic siamese network for visual object tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1763–1771.

[27] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4660–4669.

[28] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 784–799.

[29] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10657–10665.

[30] C. Sun, D. Wang, H. Lu, and M.-H. Yang, "Learning spatial-aware regressions for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8962–8970.

[31] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4902–4912.

[32] Y. Li, J. Zhu, and S. C. H. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 353–361.

[33] H.-Y. Chen, J.-H. Liang, S.-C. Chang, J.-Y. Pan, Y.-T. Chen, W. Wei, and D.-C. Juan, "Improving adversarial robustness via guided complement entropy," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4881–4889.

[34] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J. Mach. Learn. Res.*, vol. 9, pp. 249–256, May 2010.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[36] K. Diederik and B. Jimmy, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–15.

[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[38] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[39] M. Kristan, A. Leonardis, J. Matas, and, "The sixth visual object tracking vot2018 challenge results," in *Proc. Eur. Conf. Comput. Vis. Workshop*, Sep. 2018, pp. 3–53.

[40] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 445–461.

[41] L. Huang, X. Zhao, and K. Huang, "GOT-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Dec. 4, 2019, doi: 10.1109/TPAMI.2019.2957464.

[42] J. Choi, H. J. Chang, S. Yun, T. Fischer, Y. Demiris, and J. Y. Choi, "Attentional correlation filter network for adaptive visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4807–4816.

[43] M. Wang, Y. Liu, and Z. Huang, "Large margin object tracking with circulant feature maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4021–4029.

[44] X. Dong and J. Shen, "Triplet loss in siamese network for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 459–474.

[45] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1420–1429.

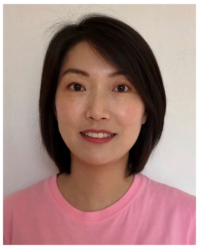
[46] Y. Zhang, L. Wang, J. Qi, D. Wang, M. Feng, and H. Lu, "Structured siamese network for real-time visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 351–366.

[47] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8971–8980.

[48] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4293–4302.

[49] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 101–117.

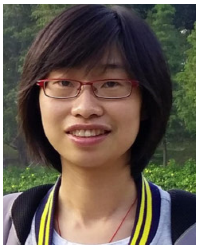
[50] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1401–1409.



AIHONG SHEN received the Ph.D. degree in computational mathematics from the Dalian University of Technology, Dalian, China. She is currently a Lecturer with the Department of Basic Courses, Criminal Investigation Police University of China. Her current research interest includes computer vision.

SHENGJING TIAN is currently pursuing the Ph.D. degree with the School of Mathematical Sciences, Dalian University of Technology. His research interests include visual tracking and 3D point cloud processing.

GUOQIANG TIAN received the M.S. degree from the School of Mathematical Sciences, Dalian University of Technology. His research interest includes visual tracking.



JIE ZHANG received the Ph.D. degree in computational mathematics from the Dalian University of Technology, Dalian, China, in 2015. She is currently a Lecturer with the School of Mathematics, Liaoning Normal University, Dalian. She has published over 20 articles in reputed journals/conferences. Her current research interests include geometric processing, machine learning, and deep learning.

XIUPING LIU received the M.S. degree from Jilin University and the Ph.D. degree from the Dalian University of Technology. She is currently a Professor with the School of Mathematical Sciences, Dalian University of Technology. She has published more than 70 articles in international journals and conferences, including *ACM Transactions on Graphics*, the IEEE Transactions on Visualization and Computer Graphics, the IEEE Transactions on Image Processing, ECCV, and ICCV. Her research interests include computer vision, computer graphics, and machine learning.