



Visual Tracking with Attentional Convolutional Siamese Networks

Ke Tan^{1,2} and Zhenzhong Wei^{1,2}(✉)

¹ School of Instrumentation and Optoelectronic Engineering,
Beihang University, Beijing, China
zhenzhongwei@buaa.edu.cn

² The Key Laboratory of Precision Opto-mechatronics Technology,
Ministry of Education, Beijing, China

Abstract. Recently Siamese trackers have drawn great attention due to their considerable accuracy and speed. To further improve the discriminability of Siamese networks for visual tracking, some deeper networks, such as VGG and ResNet, are exploited as backbone. However, high-level semantic information reduces the location discrimination. In this paper, we propose a novel Attentional Convolutional Siamese Networks for visual tracking (ACST), to improve the classical AlexNet by fusing spatial and channel attentions during feature learning. Moreover, a response-based weighted sampling strategy during training is proposed to strengthen the discrimination power to distinguish two objects with the similar attributes. With the efficiency of cross-correlation operator, our tracker can be trained end-to-end while running in real-time at inference phase. We validate our tracker through extensive experiments on OTB2013 and OTB2015, and results show that the proposed tracker obtains great improvements over the other Siamese trackers.

Keywords: Visual tracking · Siamese networks · Visual attentions

1 Introduction

Visual tracking is a one of the most important problems in computer vision due to its applications in diverse fields such as video monitoring, human-computer interactions, and industrial automation. Given only a bounding box of an arbitrary target in the first frame, the objective is to obtain the target region in the subsequent frames. Although great progress has been achieved in the past decades, it is still a herculean task to design a real-time high-performance tracker which can overcome all challenges including illumination, deformation, fast motion, occlusion and so on.

Recently, trackers based on Siamese networks [1, 6, 11, 15, 16, 27, 29, 35, 36] have achieved outstanding performances on various tracking datasets. In these

Supported by “the National Science Fund for Distinguished Young Scholars of China under Grant No. 51625501”.

methods, tracking an arbitrary object is formulated as a similarity learning problem. To ensure tracking efficiency, the similarity function is learned in an initial offline phase and often fixed during online tracking. One of the representative works is the SiamFC [1], which feeds a candidate image and a much larger search image into a fully convolutional network with a cross correlation layer. Because of the translation invariance of the embedding function, it can compute the similarity at all translated sub-windows on a dense grid in a single evaluation. However, to satisfy the strict spatial translation invariance, SiamFC is restricted to use the no-padding AlexNet [14] as the backbone which is not powerful enough. To address this issue, some modern networks, such as VGG [22] and ResNet [10], are embedded into the Siamese framework. However, deeper networks are hard to train from limited data and high-level semantic information reduces the location discrimination. In RASNet [29], a spatial residual attention and a channel favored attention are exploited to enhance the discriminative capacity of cross-correlation layer, but it does not strength the ability to extract the underlying semantic features for the targeted objects. Inspired by [30], we improve the classical AlexNet by fusing spatial and channel attentions during feature learning.

Another issue is the imbalance of positive and negative samples caused by the dense spatial sampling. Although average weighted logistic loss is used in the SiamFC, it is unreasonable to assign the same weight to all negative samples. Zheng et al. [36] introduced an effective sampling strategy to control the sample distribution and make the model focus on the semantic distractors. Rather than a distractor-aware incremental learning phase in [36], we use the feed-forward response map to penalize the effects of simple and hard negative samples simultaneously, which can be seen as a spatial regularization (Fig. 1).

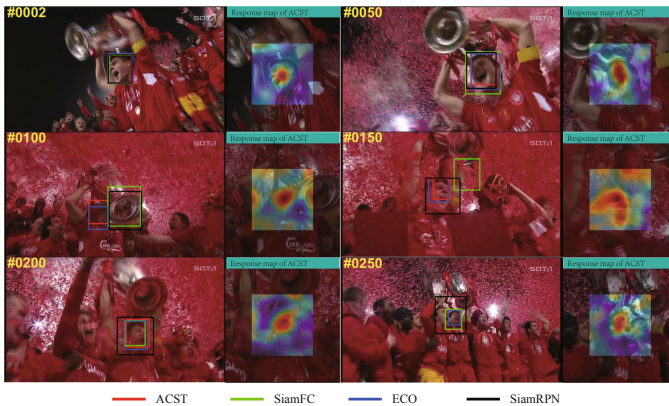


Fig. 1. Snapshots of the proposed tracker on sequence Soccer with comparison to SiamFC [1], ECO [5] and SiamRPN [16]. The right of each frame shows the response map produced by our tracker, which indicates the probability of target appears at every location.

Based on the above, we have developed an effective and efficient tracker, referred as ACST, where an improved attentional AlexNet is used as backbone and the learning phase is regularized by the response-based weighted sampling strategy. The model is trained with VID [21] and TrackingNet [19] and fine-tuned by ALOV [23] in an end-to-end manner. We evaluate our tracker on two benchmark datasets: OTB2013 [31] with 50 videos and OTB2015 [32] with 100 videos. Results show that our tracker performs favorably against a number of state-of-the-art trackers with the running speed over 30 frames per seconds.

To summarize, the main contributions of this work are listed below in three-fold:

- We improve the classical AlexNet by fusing spatial and channel attentions, which is embedded into the Siamese Network structure.
- We propose a response-based weighted sampling strategy to balance the training samples, which improve the discrimination ability to distinguish two objects with the similar attributes.
- A novel ACST tracker is proposed, which can be trained end-to-end and runs in real-time while achieves good performances.

The rest of the paper is organized as follows. Section 2 introduces the related works. The proposed architecture and training and tracking details are introduced in Sect. 3. While Sect. 4 presents the experimental results and Sect. 5 concludes this paper.

2 Related Work

In this section, we provide a brief review on methods closely related to this work.

2.1 CNNs in Visual Tracking

Recently, convolutional neural networks (CNNs) have made great breakthrough in many tasks of computer vision including visual tracking. A CNN is made up of several layers of convolution, activation, normalization and pooling operations. Existing trackers with CNNs can be roughly classified into two categories. One is that a CNN is used as a feature extractor which is combined with traditional machine learning methods [5, 12, 18, 34]. [12] combines a pretrained CNN and online SVMs to find the target location directly from saliency map. In [18], features extracted from a pretrained deep CNN are exploited to train adaptive correlation filters (CF), which are then widely used in subsequent CF-based trackers [5, 34]. However, these trackers suffer from slow speed and some are not particularly good compared to well-designed handcrafted trackers [25, 26] in accuracy. The other is that CNNs are used for feature embedding in deep tracking networks [3, 4, 7, 24, 33]. [20] proposes a multi-domain CNN to identify target in each video domain, which is then improved by [13]. And recently Siamese networks-based trackers [1, 6, 11, 15, 16, 27, 29, 35, 36] have drawn great attentions due to the balance in speed and accuracy.

2.2 Siamese Trackers

Tao et al. [27] first apply Siamese networks into visual tracking where ROI pooling is used to evaluate the similarity of two regions. Then [11] trains a Siamese network to regress directly from two images to the location in the second image of the object shown in the first image. [1] exploits fully-convolutional Siamese networks with cross correlation operations to directly produce a response map. Although [1] achieves a good performance in both speed and accuracy, it still has a gap to the best online tracker. To further improve the performance of the Siamese tracking framework, a great deal of work has been done such as combing with RPN [15, 16], using deeper networks [15, 17, 35], online learning [2, 8, 28], augmented loss function [6, 36] and so on.

3 The Proposed Tracker

In this section, we firstly introduce the architecture of proposed attentional convolutional Siamese network. Then a response-based weighted sampling strategy is developed to solve the unbalance of simple and hard samples. Finally, we present the online tracking pipeline of our ACST tracker.

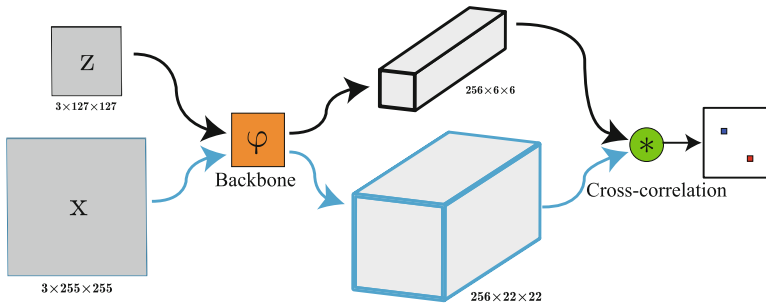


Fig. 2. The structure of fully-convolutional Siamese networks for visual tracking. The backbone extracts features from candidate and search image, which are then used to produce a response map with a cross-correlation layer.

3.1 Attentional Convolutional Siamese Network

We first review the Siamese network with fully-convolutional operations. A fully-convolutional network with integer stride k can be seen an embedding function φ satisfy

$$\varphi(L_{k\tau}(x)) = L_{\tau}(\varphi(x)) \quad (1)$$

where L_{τ} is the translation operator $(L_{\tau}x)[u] = x[u - \tau]$ for any translation τ . The advantage of fully-convolutional Siamese networks is that it can compute

the similarity at all translated sub-windows on a dense grid in a single evaluation by a cross-correlation operator f by

$$f(z, x) = \varphi(z) * \varphi(x) + b\mathbb{1} \tag{2}$$

where z and x are a candidate image and a much larger search image respectively, $b\mathbb{1}$ denotes a bias term at every location. Figure 2 shows the structure of Siamese network for visual tracking.

The discriminative ability of the above model depends largely on the backbone network φ which extracts features for the image. Inspired by [30], we exploit the attentional convolutional block to improve the classical AlexNet.

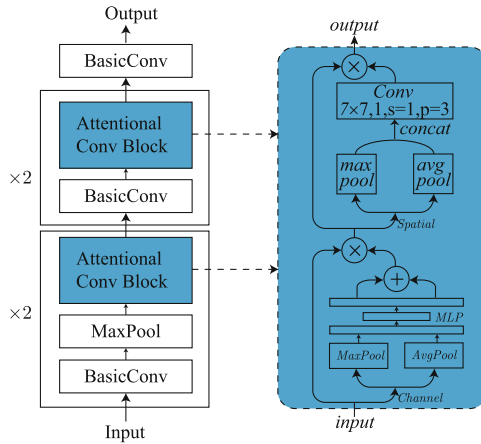


Fig. 3. The architecture of the proposed attentional AlexNet which is used as the backbone. The attentional convolutional block is illustrated in the right, where channel and spatial attention are applied to refine the input feature. Note that both max-pooling and average-pooling outputs are utilized in the channel attention with a shared Multi-Layer Perception (MLP), while the spatial attention concatenates two outputs that are pooled along the channel axis and forward them to a convolution layer.

As is shown in Fig. 3, the attentional convolutional block is made up of a basic convolution block and a dual attention module which operates along two separate dimensions, channel and spatial. Given an intermediate feature map, the output is obtained by sequentially multiplying the attention maps to it for adaptive feature refinement. Specifically, channel attention focuses on ‘What’ is meaningful while spatial attention focuses on ‘Where’ is an informative part.

We illustrate the Attentional Convolutional Siamese Network configuration in Table 1. In particularly, we modify the first 4 conv layers of no-padding AlexNet in [1] by the attentional convolutional block. Then we use a combination of cross correlation layers for calculating score maps. The generated maps indicate the similarity information between the candidate image and search image.

Table 1. Architecture of Attentional Convolutional Siamese Network which is modified from the no-padding AlexNet.

Layer	<i>Exemplar</i> output size	<i>Search</i> output size
<i>Input</i>	127 × 127	255 × 255
<i>Conv1</i> 3, 96, 11 × 11, <i>stride</i> = 3	59 × 59	123 × 123
<i>Pool1</i> 3 × 3, <i>stride</i> = 2	29 × 29	61 × 61
<i>AttConvBlock1</i>		
<i>Conv2</i> 96, 256, 5 × 5, <i>stride</i> = 1	25 × 25	57 × 57
<i>Pool2</i> 3 × 3, <i>stride</i> = 2	12 × 12	28 × 28
<i>AttConvBlock2</i>		
<i>Conv3</i> 256, 384, 3 × 3, <i>stride</i> = 1	10 × 10	26 × 26
<i>AttConvBlock3</i>		
<i>Conv4</i> 384, 384, 3 × 3, <i>stride</i> = 1	8 × 8	24 × 24
<i>AttConvBlock4</i>		
<i>Conv5</i> 384, 256, 3 × 3, <i>stride</i> = 1	6 × 6	22 × 22
<i>CrossCorrelation</i>	17 × 17	

3.2 Weighted Sampling Training

We perform the following steps one by one to generate sample pairs and corresponding labels.

- Randomly choose two frames with a max interval N from a video and decode into 32-bit floating point raw pixel values in $[0, 255]$.
- Extract the exemplar image centered on the target following the way in [1].
- Extract the instance image which is translated within T pixels as well as scaled by a random stretch factor in $[1 - \alpha, 1 + \alpha]$.
- Calculate the labels for the training pair, which are belong to positive if they are within radius R of the target center c (accounting for the stride k)

$$y[u] = \begin{cases} 1, & \text{if } k \|u - c\| \leq R \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

In [1], the average weighted logistic loss is adopted to train the network. However, assigning the same weight to all negative samples will reduce the ability to distinguish two objects with similar semantic information. Here we weight the training sample on the dense grid by the feed-forward response map, shown in Fig. 4. Given a real-valued score of a single exemplar-candidate pair v , we first calculate the sigmoid output $s[u] = \text{sigmoid}(v[u])$, which indicates the similarity of the training pair. Then the weight for each position is

$$w[u] = \begin{cases} \frac{s[u]}{2 \sum_{u \in y^+} s[u]}, & \text{if } k \|u - c\| \leq R \\ \frac{s[u]}{2 \sum_{u \in y^-} s[u]}, & \text{otherwise} \end{cases} \tag{4}$$

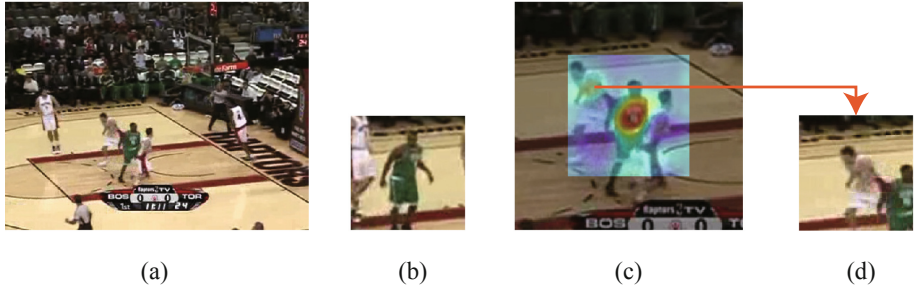


Fig. 4. Illustration of the weighted sampling strategy. (a) is the tracking frame. (b) is the candidate image extracted from the first frame. (c) is the search image with a response map added on it, which indicates the similarity between dense grid samples and candidate image. (d) is a hard sample with a similar attributes with candidate image, which is assigned more weight.

We choose the cross-entropy loss as the loss function, so the final weighted average loss is

$$loss(v, y) = - \sum_u w[u] (y[u] \ln(s[u]) + (1 - y[u]) \ln(1 - s[u])) \quad (5)$$

Furthermore, Stochastic Gradient Descent with momentum is applied to obtain the network parameters.

3.3 Tracking

Given the first frame with the target annotated, we extract the exemplar image and feed it into the attentional convolutional network to obtain the filter kernel. Note that we do not update the kernel for speed consideration. When there comes a new frame, the target is searched around the previous position. To handle the scale variances, multi-scaled search patches are extracted as a mini-batch, which is used to input into the network and then calculate the cross-correlation with the kernel. The obtained response maps are up-sampled by bi-cubic interpolation and penalized by the scale factor. Meanwhile, a cosine window is added to the response maps to penalize large displacements. The peak of response maps indicates the position of the target and the scale is updated with a decay.

4 Experiments

In this section, we firstly provide details of training and tracking implementation. Then experiments on OTB2013 [31] and OTB2015 [32] are conducted to evaluate the proposed tracker and the results are compared with some state-of-the-arts. Finally, we show qualitative results on some challenging videos.

4.1 Implementation Details

During training phase, the weights of convolutional layers are initialized with Kaiming algorithm [9] and all bias are initialized to zero. For batch normalization layers, γ and β vectors are initialized to one and zero respectively. We first use VID [21] and TrackingNet [21] to train the model with a batch size 32. And SGD is applied with a warm-up learning rate increasing linearly for the first 5 epochs from 5×10^{-3} to 2.5×10^{-2} and decayed by 0.8576 for 15 epochs. Then the last two convolution layers of backbone are fine-tuned using ALOV [23] with a batch size 8. The learning rate of SGD is decayed by 0.8576 for 30 epochs. Moreover, the preprocessing steps are described in Sect. 3.2 with $N = 100$, $T = 64$, and $\alpha = 0.1$.

During tracking, the weight of cosine window is set to 0.31 and the up-sampled factor of response map is set to 16. We search for the object over 3 scales $1.0375^{\{-1,0,1\}}$ and update the scale by a liner interpolation with a factor of 0.59. The proposed model is implemented in PyTorch 1.0 on a workstation with an Intel E5-1650 v4 CPU, 32G RAM, NVIDIA TITAN Xp GPU. Our tracker can run over 30 fps on OTB benchmark.

4.2 Experimental Validations

We evaluate the proposed tracker, referred as ACST, on OTB2013 and OTB2015 datasets which contain 50 and 100 videos respectively, with comparison to some state-of-the-art methods. We quantitatively evaluate trackers using center location error and overlap ratio. And for completeness, we also report qualitative results on some challenging videos.

Quantitative Analysis. We provide a comparison of ACST with 14 state-of-the-art trackers: MDNet [20], SCT [4], SiamFC [1], CREST [24], ECO [5], ADNet [33], SANet [7], SiamRPN [16], SiamTri [6], DaSiamRPN [36], ACT [3], SiamDW-FC [35], SiamDW-RPN [35] and SiamRPN++ [15]. Figure 5 shows the precision plots and success plots under one-pass evaluation (OPE) on OTB2013 and OTB2015 for all the 15 trackers which are ranked using the AUC (area under the curve) displayed in the legend.

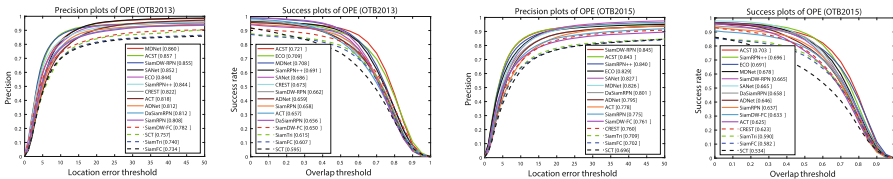


Fig. 5. Precision plots and Success plots under one-pass evaluation (OPE) on OTB2013 and OTB2015.

Qualitative Analysis. Figure 6 shows the tracking results of 4 methods: ECO [5], SiamFC [1], SiamRPN [16] and the proposed ACST on 6 challenging sequences. The ECO tracker performs well in sequences with illumination and fast motion (Singer2, DragonBaby) but fails when occlusion and rotation (MotoRolling, Freeman4). SiamRPN performs well in sequences with deformation, but fails when back-ground clutter (Tiger1). Our tracker performs well on all these videos, which validates the effectiveness.



Fig. 6. Qualitative results of our ACST and 3 compared trackers: SiamFC [1], ECO [5] and SiamRPN [16].

5 Conclusion

In this paper, we improve the classical AlexNet with attention mechanism, which is used as the backbone of Siamese Networks for visual tracking. Furthermore, we enhance the ability of the proposed network to distinguish targets with similar attributes. Based on the above, we propose an ACST tracker, which achieves great performance on OTB2013 and OTB2015 datasets while running in real-time. In the future, we will make efforts to improve our tracker with RPN.

References

1. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-convolutional siamese networks for object tracking. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 850–865. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_56
2. Bhat, G., Danelljan, M., Van Gool, L., Timofte, R.: Learning discriminative model prediction for tracking. arXiv preprint [arXiv:1904.07220](https://arxiv.org/abs/1904.07220) (2019)
3. Chen, B., Wang, D., Li, P., Wang, S., Lu, H.: Real-time ‘Actor-Critic’ tracking. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 328–345. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_20
4. Choi, J., Jin Chang, H., Jeong, J., Demiris, Y., Young Choi, J.: Visual tracking using attention-modulated disintegration and integration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4321–4330 (2016)
5. Danelljan, M., Bhat, G., Shahbaz Khan, F., Felsberg, M.: ECO: efficient convolution operators for tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6638–6646 (2017)
6. Dong, X., Shen, J.: Triplet loss in siamese network for object tracking. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11217, pp. 472–488. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01261-8_28
7. Fan, H., Ling, H.: SaNet: structure-aware network for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 42–49 (2017)
8. Guo, Q., Feng, W., Zhou, C., Huang, R., Wan, L., Wang, S.: Learning dynamic siamese network for visual object tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1763–1771 (2017)
9. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034 (2015)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
11. Held, D., Thrun, S., Savarese, S.: Learning to track at 100 FPS with deep regression networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 749–765. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_45
12. Hong, S., You, T., Kwak, S., Han, B.: Online tracking by learning discriminative saliency map with convolutional neural network. In: International Conference on Machine Learning, pp. 597–606 (2015)
13. Jung, I., Son, J., Baek, M., Han, B.: Real-time MDNet. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11208, pp. 89–104. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01225-0_6
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, pp. 1097–1105 (2012)
15. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: SiamRPN++: evolution of siamese visual tracking with very deep networks. arXiv preprint [arXiv:1812.11703](https://arxiv.org/abs/1812.11703) (2018)

16. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8971–8980 (2018)
17. Li, Y., Zhang, X.: SiamVGG: visual tracking using deeper siamese networks. arXiv preprint [arXiv:1902.02804](https://arxiv.org/abs/1902.02804) (2019)
18. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical convolutional features for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3074–3082 (2015)
19. Müller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B.: TrackingNet: a large-scale dataset and benchmark for object tracking in the wild. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11205, pp. 310–327. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01246-5_19
20. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4293–4302 (2016)
21. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**(3), 211–252 (2015)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
23. Smeulders, A.W.M., Chu, D.M., Rita, C., Simone, C., Afshin, D., Mubarak, S.: Visual tracking: an experimental survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(7), 1442–1468 (2014)
24. Song, Y., Ma, C., Gong, L., Zhang, J., Lau, R.W., Yang, M.H.: CREST: convolutional residual learning for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2555–2564 (2017)
25. Tan, K., Wei, Z.: Learning an orientation and scale adaptive tracker with regularized correlation filters. *IEEE Access* **7**, 53476–53486 (2019)
26. Tang, M., Yu, B., Zhang, F., Wang, J.: High-speed tracking with multi-kernel correlation filters. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4874–4883 (2018)
27. Tao, R., Gavves, E., Smeulders, A.W.: Siamese instance search for tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1420–1429 (2016)
28. Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., Torr, P.H.: End-to-end representation learning for correlation filter based tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2805–2813 (2017)
29. Wang, Q., Teng, Z., Xing, J., Gao, J., Hu, W., Maybank, S.J.: Learning attentions: residual attentional siamese network for high performance online visual tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
30. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: CBAM: convolutional block attention module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 3–19. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_1
31. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: a benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2411–2418 (2013)
32. Yi, W., Jongwoo, L., Ming-Hsuan, Y.: Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(9), 1834–1848 (2015)

33. Yun, S., Choi, J., Yoo, Y., Yun, K., Young Choi, J.: Action-decision networks for visual tracking with deep reinforcement learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2711–2720 (2017)
34. Zhang, T., Xu, C., Yang, M.H.: Learning multi-task correlation particle filters for visual tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **41**(2), 365–378 (2019)
35. Zhipeng, Z., Houwen, P., Qiang, W.: Deeper and wider siamese networks for real-time visual tracking. arXiv preprint [arXiv:1901.01660](https://arxiv.org/abs/1901.01660) (2019)
36. Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware siamese networks for visual object tracking. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) *ECCV 2018*. LNCS, vol. 11213, pp. 103–119. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01240-3_7