Combined Correlation Filters with Siamese Region Proposal Network for Visual Tracking

Shugang Cui, Shu Tian(✉), and Xucheng Yin

School of Computer and Communication Engineering,
University of Science and Technology Beijing, Beijing 100083, China
csg_charles@foxmail.com, {shutian,xuchengyin}@ustb.edu.cn

Abstract. Siamese network based trackers have received extensive attention with their trade-off between accuracy and speed. In particular, Siamese Region Proposal Network (SiamRPN) tracker can get more accurate bounding box with proposal refinement, yet, most siamese trackers are lack of discrimination without target classification and robustness without online learning module. To tackle the problem, in this paper, we propose an ensemble tracking framework based on SiamRPN tracker, consisting of two components: (1) Correlation Filter module with hierarchical features fusion; and (2) SiamRPN module. The Correlation Filter module fully exploits both the semantic features for classification and the lower-level features for precise localization through online learning process. By cascading the Correlation Filter to SiamRPN tracker, which can equip with discrimination power. The entire network based on multitask learning strategy is trained in an end-to-end manner, which enhances both robustness and module adaptability effect. In extensive experiments evaluations on GOT-10K test dataset, OTB2015 and VOT2016 benchmarks, our tracking approach obtains better performance than other trackers, including SiamRPN tracker, by a notable margin.

Keywords: Correlation Filters · Siamese Region Proposal Network · Visual tracking

1 Introduction

Visual object tracking is one of the most challenging tasks in computer vision, which has been widely applied in areas such as video surveillance and intelligent transportation systems. Due to the object appearance changes and complex motion scene, it still suffer many challenges including occlusion, scale variation, etc.

Correlation Filter (CF) based trackers regard object tracking as a two-class problem between target and background information. With its efficient performing operations in frequency domain, it has been attracted wide attention and

T. Gedeon et al. (Eds.): ICONIP 2019, LNCS 11954, pp. 128–138, 2019.
https://doi.org/10.1007/978-3-030-36711-4_12

achieved rapid progress. Since deep neural network can extract image features more accurately, correlation filter based trackers started using deep features to improve the accuracy, however, these features are usually pre-trained and mainly from classification and detection tasks, which can not better applied to object tracking task.

Recently, siamese network based trackers have made significant progress. Most siamese trackers achieved remarkable tracking performance by reducing the heavy burden of online learning in offline learning way. However, they can't effectively distinguish object from background information in clutter scene, besides, since no online learning is performed, most siamese trackers are lack of good robustness. To learn the convolutional features and perform the correlation tracking process simultaneously, Wang et al. [1] treated CF as a differentiable network layer added in siamese network, which enable get a significant accuracy compared with traditional CF tracking algorithms. However, they proposed tracker only utilized low-level features representations which lacked rich high-level semantic information and can not effectively distinguish different types of objects. Additionally, with traditional multi-scale test, the DCFNet [1] tracker only can obtain the central position of the target, but not the specific target size.

In this paper, we propose a tracking method of hierarchical features fusion based on Siamese Region Proposal Network (SiamRPN) [2]. Our tracker consists of two components: (1) Correlation Filter module with hierarchical features fusion; and (2) SiamRPN module. The CF module of our tracker that utilize the characteristics of convolutional networks at different levels can learn strong semantic and localization information. The entire network based on multi-task learning strategy is trained in an end-to-end manner, which enhances both discrimination and generation effect.

2 Related Work

Visual object tracking has been extensively studied in recent decades, in this section, we discuss tracking methods closely related to this work.

Correlation Filter Based Trackers (CF). The basic idea of CF trackers is to learn a filter template and let the image of the next frame correlates with the filter template, where the greatest response is the target of prediction. These trackers regress the input circulant matrix features to a target Gaussian function by minimizing the mean square error. Since Henriques et al. [3] diagonalized circulant matrix with the discrete fourier transform to propose a kernelized correlation filter (KCF) tracker, many extensions has been proposed to improve the tracking accuracy. Besides handcrafted features, a series of CF trackers utilized deep convolutional features to train classifier, e.g. [4] exploited three convolutional layers features from pretrained VGG-19 net to learn three independent CF, and then the final target position is obtained by weighted fusion of the three score maps. However, the pretrained feature representations from other computer vision tasks cant't suit object tracking well. In this work, we combine the online learning CF with the off-line trained SiamRPN [2] tracker, improving the tracking performance.

Siamese Network Based Trackers. Siamese networks have drawn increasing interest in the tracking community because of its super tracking speed and higher accuracy. SiamFC [5] used siamese neural network to learn a kind of similarity measure function between target patch and a test patch, which designed an end-to-end network and exceeded the real-time requirement. Since then, many siamese network based trackers have emerged. The challenge of the siamFC-based approaches are lack of robustness and discrimination. He et al. [6] used complementary semantic features and appearance features to present two fold siamese network named SA-Siam with real-time tracking effect, and it's performance largely exceeded all other real-time trackers at the time. Another main challenges of siamFC-based trackers are how to handle target scale and shape changes, Li et al. [2] exploited Region Proposal Network (RPN) [7] to get more accurate bounding boxes by using box refinement process. Different from them, we enhance discriminability of SiamRPN tracker by exploiting CF trackers with hierarchical features fusion.

Ensemble Trackers. The ensemble framework usually contains multiple models rather than using a single one, which makes the ensemble tracker have stronger generalization ability. e.g. Qi [8] presented an typical ensemble tracking framework, which contained different convolution layers and used adaptive hedging method to hedge several CNN-based trackers into a stronger tracker. BranchOut [9] as an online ensemble tracker had multi-level target representation, which can learn robust target appearance models with diversity and handle various challenges effectively. Similarly, In our designed tracker, the adaptive Correlation Filter with hierarchical features fusion complement with the SiamRPN [2] module, and they are jointly trained in an end-to-end manner.

3 Proposed Method

3.1 Framework

Our proposed framework that is integrated in a unified multi-task network architecture is shown in Fig. 1. We cascade a differentiable Correlation Filter layer to SiamRPN [2] tracker. Our proposed tracker contains Correlation Filter and SiamRPN two parts, which can enable correlation filter tightly to couple to the deep features. More details refer to Sect. 3.4.

3.2 Hierarchical Features Fusion

In order to fully exploit the multi-level features, we learn the Correlation Filter by building hierarchical features at various scales in top-down pathway and lateral connections manner [10]. The top-down process enlarges higher resolution

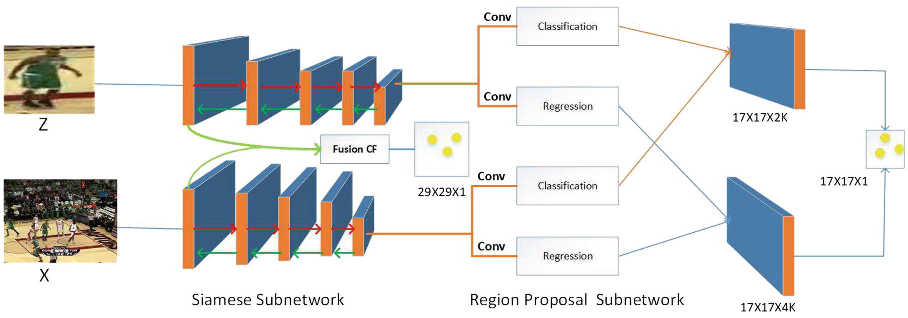


Fig. 1. The proposed ensemble tracking framework. The framework contains Correlation Filter module with hierarchical features fusion and SiamRPN [2] module. The network inputs target patch Z and search patch X, the extracted features are fused for correlation filter tracker and exploited by SiamRPN, respectively. Where → denotes features extraction, ← denotes features fusion in top-down pathway and lateral connections way, Conv denotes one convolutional operation and ● denotes predicted target location.

features to the same size as the former lower feature by up-sampling, to take advantage of the underlying location details, the lateral connection fuses the features of the high-level layer after the upsampling and the current layer by addition method. Therefore, the fused features contains the semantic features for classification and the lower-level for precise localization.

Figure 2 shows the detailed process of fusion. Conv1, Conv2, Conv3, Conv4, Conv5 denotes each layer of network, individually. The Conv5 layer undergoes a 1×1 convolutional layer to adjust channel dimensions and obtains feature map M5. After M5 feature is upsampled, it is then merged with the Conv4 layer (which also needs a 1×1 convolutional operation) in element-wise addition way to obtain feature map M4. This process is iterated until the finest resolution feature map M1 is generated. Finally, The merged feature map M1 is appended a 3×3 convolution to generate the final feature map P, which is to reduce the aliasing effect of upsampling. Different from detection task, visual tracking is more dependent on features that contain object location information, we only add a differentiable CF lay with extracting fusion features in lowest level feature map P, which can complement SiamRPN [2] discrimination effect.

3.3 Correlation Filters

Standard correlation filters learn a discriminative classifier by ridge regression, which can get a simple closed-form solution. By successfully utilizing the diagonalization property of cyclic matrix with fast fourier transforms, correlation filters greatly reduce the amount of operation and improves the speed of operation. The goal of training is to find a function f(Z) that minimizes the squared error over samples Z and response values y that is a 2D gaussian function:

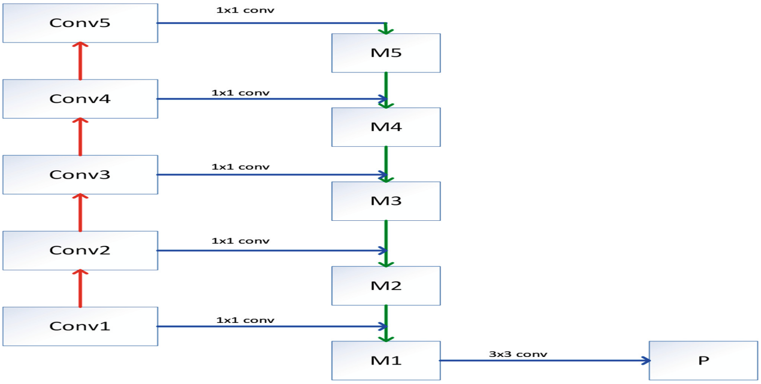


Fig. 2. The top-down pathway and lateral connections manner. Where ↑ denotes features extraction, ↓ denotes features fusion in top-down pathway and → denotes lateral connections.

$$\min_{w} \parallel Zw - y \parallel_2^2 + \lambda \parallel w \parallel_2^2 \tag{1}$$

where w refers to learned correlation filter, the features of target patch Z is extracted by certain circulant shift operation, and λ is a regularization parameter that controls overfitting. With diagonalization of fourier frequency domain of cyclic matrix, the closed-form solution can be obtained:

$$\hat{w} = \frac{\hat{Z}^* \odot \hat{y}}{\hat{Z}^* \odot \hat{Z} + \lambda} \tag{2}$$

where $\hat{Z}^*$ is the complex-conjugate of Z, $\hat{Z}$ denotes the Discrete Fourier Transform of the generating vector and $\odot$ denotes the Hadamard product.

In the tracking process, we extract features in the next search region X, and the response can be obtained according to the trained filter template w and the sample based on the target position of the previous frame Z.

3.4 Refined Siamese Region Proposal Network

In this section, we begin with an overview of the general Siamese Region Proposal Network (SiamRPN) [2] and discuss how to combine it with Correlation Filter module.

SiamRPN [2] consists of Siamese Subnetwork (left side of Fig. 1) for features extraction and Region Proposal Subnetwork (right side of Fig. 1) for classification and regression task. Siamese Subnetwork consists of target template branch and detection branch, the former branch takes cropped target patch Z as input, the latter branch takes cropped search region patch X in the next frame as input. To ensure classification and regression for each anchor, convolutional operation is needed to adjust the channels into suitable forms. Therefore, after Siamese

Subnetwork, the extracted feature $\varphi(z)$ of target template branch perform two convolutional operations individually, hence, $\varphi(z)$ is split into two parts $[\varphi(z)]_{cls}$ and $[\varphi(z)]_{reg}$. Similarly, the extracted features $\varphi(x)$ of detection branch also perform two convolutional operations individually, and then $\varphi(x)$ become $[\varphi(x)]_{cls}$ and $[\varphi(x)]_{reg}$ two parts. Therefore, the final classification scores cls_{2k} and regression offsets reg_{2k} of output can be obtained as follows:

$$\begin{aligned}\{cls_{2k}\} &= corr([\varphi(z)]_{cls}, [\varphi(x)]_{cls}) \\ \{reg_{2k}\} &= corr([\varphi(z)]_{reg}, [\varphi(x)]_{reg})\end{aligned} \tag{3}$$

where corr(a, b) denotes convolution between a and b, the feature maps $[\varphi(z)]_{cls}$ and $[\varphi(z)]_{reg}$ are used as kernels, k denotes the number of anchors. The SiamRPN [2] is trained in end to end manner, which consists of cross-entropy loss for classification and smooth L1 loss for regression. The multi-task loss is as follows and detailed loss information refer to [2].

$$L_{SiamRPN} = L_{cls} + L_{reg} \tag{4}$$

where L_{cls} and L_{reg} denote classification loss and regression loss, respectively.

Compared with traditional SiamRPN [2], our proposed tracker is shown in Fig. 1, after Siamese Subnetwork extract features of target template Z and the search region X, we cascade a correlation filter module with multi-level fusion for discriminative tracking between Siamese Subnetwork and Region Proposal Subnetwork. After template branch feature $\varphi(z)$ is fused by the way described in Sect. 3.2, we obtain final fused features $z = P(z;\theta)$, likewise, the fused detection branch features denoted as $x = P(x;\theta)$. θ represent the parameters of these convolutional layers. The specific cascaded CF loss function is as follows:

$$L_{CF} = \| g(x) - y \|_2^2 + \gamma \| \theta \|^2 \tag{5}$$

$$\mathrm{g}(x) = F^{-1}(\hat{\varphi}(x;\theta) \odot \hat{w}) \tag{6}$$

where $\widehat{x}$ is the Discrete Fourier Transform of x and $\widehat{w}$ is the learned CF based on fused target feature z, γ is a regularization parameter. The derivatives of L_{CF} can be obtained as follows:

$$\frac{\delta L_{CF}}{\delta \hat{g}^*} = 2(\hat{g}(x) - \hat{y}) \tag{7}$$

$$\frac{\delta L_{CF}}{\delta x} = F^{-1}(\frac{\delta L_{CF}}{\delta \hat{g}^*} \odot \hat{w}^*) \tag{8}$$

$$\frac{\delta L_{CF}}{\delta \hat{w}} = \frac{\delta L_{CF}}{\delta \hat{g}^*} \odot \hat{x}^* \tag{9}$$

$$\frac{\delta L_{CF}}{\delta z} = F^{-1}(\frac{\delta L_{CF}}{\delta \hat{w}} \odot \frac{\hat{y}^* - 2\mathrm{Re}(\hat{z}^* \odot \hat{w})}{\hat{z}^* \odot \hat{z}}) \tag{10}$$

where Re(·) is the real part of a complex-valued matrix.

Due to the correlation filters and siamRPN [2] modules complement each other in multi-scale regression and recognition tracking based on multi-resolution representation. We adopt multi-task learning strategy to end-to-end train the network, the overall loss function can be written as:

$$L_{ALL} = L_{CF} + \mu L_{SiamRPN} \tag{11}$$

where L_{CF} denotes the correlation filters module loss and $L_{SiamRPN}$ denotes the SiameseRPN module loss, μ is hyper-parameter to balance the two parts.

In the tracking process, we feed target patch Z and search region X centered at the previous target position into the network, then we can get their corresponding feature representations of target template branch and detection branch through Siamese Subnetwork. On one hand, The two branch features with hierarchical features fusion are exploited by CF module, on the other hand, these features are further fed into Region Proposal Subnetwork (RPN) module for classification and localization. The target state is obtained in Eq. 12, which is estimated by finding the maximum of the fused CF module scores given in Eq. 6 and classification scores given in Eq. 3.

$$\arg\max_{m,n} = \{cls_{2k}\}_{m,n} + g_{m,n}(X) \tag{12}$$

Then the final target bounding box can be obtained with the max target state give in Eq. 12 and the regression offsets given in Eq. 3 by non-maximum-suppression (NMS). Note that in Eq. 12, we adopt the bilinear interpolation method to fuse $\{cls_{2k}\}$ and g(x) to have consistent resolution.

In order to make our tracker adaptive to continuous changes in the appearance of the object, we adopt incremental update strategy. The training goal in Eq. 1 is changed as follows:

$$\min_{w_p} = \sum_{t=1}^{p} \beta_t (\| Zw_p - y \|_2^2 + \lambda \| w_p \|_2^2) \tag{13}$$

where β_t is a hyper parameter, the advantage is that we do not have to maintain a large exemplar set and only need small memory footprint. The solution can be gained as:

$$\hat{w}_p = \frac{\sum_{t=1}^{p} \beta_t \hat{y} \odot \hat{Z}^*}{\sum_{t=1}^{p} \beta_t (\hat{Z}^* \odot \hat{Z} + \lambda)} \tag{14}$$

4 Experiments

4.1 Datasets

Our network is end-to-end trained on the GOT-10K train dataset. We evaluate tracking performance on GOT-10K [11] test dataset, OTB2015 [12], and VOT2016 [13] benchmarks. Notice that different from SiamRPN tracker [2], We need end-to-end train our proposed framework, therefore, for fair comparison, we also retrain SiamRPN [2] tracker on GOT-10K [11] train dataset as our baseline.

GOT-10K [11] is a large high-diversity database for generic object tracking, which owns more than 10 thousand video segments and 1.5 million bounding boxes. It is also a generic evaluating benchmark including three subsets:train, validation and test datasets. It uses average overlap (AO) and success rate (SR) as evaluation indicator. The AO denotes the average of overlaps between the tracked and groundtruth bounding boxes, while the SR measures the percentage of successfully tracked frames where the overlaps exceed 0.5.

OTB2015 [12] benchmark with 100 videos is a fair testbed. It adopts the precision and success for evaluation, the success plot shows the ratios of successful frames when the threshold varies from 0 to 1, the precision plots show the percentage of frames where the center location error is within a threshold 20. The area under curve (AUC) of success plot is used to rank tracking algorithm.

The VOT2016 [13] benchmark has 60 sequences to evaluate a tracker with applying a reset-based methodology. it exploited accuracy (A), robustness (R) and expected average overlap (EAO) to compare different trackers.

4.2 Implementation Details

Our experiment is implemented in python using Pytorch of deep learning framework on two Nvidia GTX 1080 with 20 GB memory. Following SiamRPN [2], we use first five layers of pre-trained classification model on ImageNet dataset as our backbone and train the network on GOT-10K train dataset. The target patch has a size of $127 \times 127 \times 3$, and search region patch has a size of $255 \times 255 \times 3$. After features extraction by Siamese Subnetwork, its output is fused with hierarchical features to train CF classifier, at the same time, the output features are directly fed to the RPN [7] layer for further classification and localization. During training process, We apply stochastic gradient descent (SGD) to train the network and the regularization parameters μ is set 0.8, the learning rate exponentially decays from 0.01 to 0.0005. The model is trained for 10 epoches with a mini-batch size of 20.

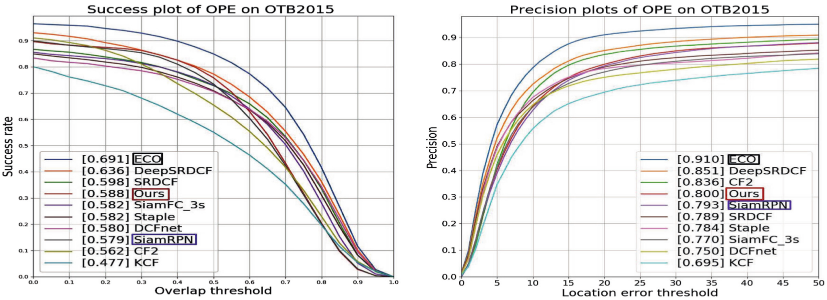
4.3 Results and Conclusions

GOT-10K Test Datasest. Table 1 shows the comparisons between on GOT-10K test datasest, by introducing a discriminative correlation filter with hierarchical features fusion, it proves that the AO, SR and speed metrics in our tracker perform better than baseline SiamRPN [2] tracker, Compared to the SRDCF tracker that is a general correlation filter, it also proves that our method is much more efficient with ensemble strategy.

OTB2015 Benchmark. We compare our tracker with KCF [3], ECO [14], SiamFC [5], Staple [15], SiamRPN [7], DCFNet [1] et al. trackers on OTB2015 benchmark. The precision plots and success plots of one path evaluation (OPE)

Table 1. Results on GOT-10K test dataset with average overlap (AO), success rate (SR) metrics at the threshold of 0.5 and Speed.

Tracker	AO	SR	Speed
SiamRPN	40.1	45.0	70
CF2	31.5	30.0	5.57
CCOT	32.5	32.8	0.68
MDNet	29.9	30.3	1.52
CFNetc2	29.3	26.5	35.62
ECOhc	28.6	27.6	34.5
Staple	24.6	23.9	28.87
SRDCF	23.6	22.3	5.58
KCF	20.3	17.7	94.66
Ours	40.5	45.9	55



(a) success plot on OTB2015 (b) Precision plot on OTB2015

Fig. 3. Success and precision comparisons on OTB2015.

are shown in Fig. 3. We obtain the 0.588 AUC and 0.800 precision scores on success and precision metrics, respectively. Compared with the baseline SiamRPN [2] tracker, our ensemble method performs slightly better with online update strategy. Due to the ECO tracker [14] has more diversity data and its correlation filter method is optimized, there still is a gap between our tracker and ECO [14].

VOT2016 Benchmark. We compare our tracker with SiamRPN [2] and other trackers on VOT2016 benchmark. Table 2 shows our tracker ranks 3rd, 4th and 5th in the overall performance evaluations based on the accuracy (A), robustness (R) and expected average overlap (EAO) metrics, respectively. Compared with the baseline SiamRPN [2], Our tracker achieves gain of 2.9% on EAO, 2.8% on R and 1.1% on A. Even if our method adds CF, the speed (fps) still shows better performance. In addition, our tracker outperforms many correlation filter based

Table 2. Comparison with trackers on VOT2016 benchmark. A, R and EAO denote accuracy, robustness and expected average overlap, respectively. The larger values of ECO and A, they represent better performance, however, the larger value of R, it represents worse performance. The best results are highlighted in **black thick fonts** on the three metrics, respectively.

Tracker	A	R	EAO	Speed
SiamRPN	0.557	0.317	0.280	160
SiamFC	0.549	0.382	0.277	86
C-RPN	**0.594**	0.95	**0.363**	1
MDNet	0.54	0.34	0.257	1
ECO	0.55	0.20	0.375	8
C-COT	0.539	0.238	0.331	0.3
Staple	0.544	0.378	0.295	80
SPM	0.620	**0.210**	0.434	120
SSKCF	0.547	0.373	0.277	150
Ours	0.568	0.289	0.309	150

trackers. However, due to C-RPN [16] owns larger train dataset, therefore, it performs better than our tracker.

Conclusions. In this paper, we present an ensemble tracker with SiamRPN [2] module and Correlation Filter module. The Correlation Filter of our tracker is learned with hierarchical features fusion to localize and online update for adaptive tracking. Our tracker is evaluated on GOT-10K test dataset, OTB2015 and VOT2016 benchmarks, they show our method can achieve more significant performance than baseline SiamRPN [2] tracker and most correlation filter based trackers.

Acknowledgments. The research is partly supported by National Natural Science Foundation of China (61806017).

References

1. Wang, Q., Gao, J., Xing, J., et al.: DCFNet: discriminant correlation filters network for visual tracking. arXiv:1704.04057 (2017)
2. Li, B., Yan, J., Wu, W., et al.: High performance visual tracking with siamese region proposal network. In: 2018 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, 18–23 June 2018, pp. 8971–8980 (2018)
3. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. 2015 IEEE Trans. Pattern Anal. Mach. Intell. J. **37**(3), 583–596 (2015)

4. Ma, C., Huang, J.-B., Yang, X., Yang, M.-H.: Hierarchical convolutional features for visual tracking. In: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, 11–18 December 2015, pp. 3074–3082 (2015)
5. 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
6. He, A., Luo, C., Tian, X., Zeng, W.: A twofold Siamese network for real-time object tracking. In:2018 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, 18–23 June 2018, pp. 4834–4843 (2018)
7. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In:2015 29th Annual Conference on Neural Information Processing Systems, NIPS 2015, Montreal, Canada, 07–12 December 2015, pp. 91–99 (2015)
8. Qi, Y., Zhang, S., Qin, L., Yao, H., et al.: Hedged deep tracking. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Seattle, WA, 27–30 June 2016, pp. 4303–4311 (2016)
9. Han, B., Sim, J., Adam, H.: BranchOut: regularization for online ensemble tracking with convolutional neural networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, 21–26 July 2017, pp. 521–530 (2017)
10. Lin, T.-Y., Dollar, P., Girshick, R.: Feature pyramid networks for object detection. In: 2017 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, 21–26 July 2017, pp. 936–944 (2017)
11. Huang, L., Zhao, X., Huang, K.: GOT-10k: a large high-diversity benchmark for generic object tracking in the wild. IEEE Trans. Pattern Anal. Mach. Intell. J. **37**(9) (2018)
12. Wu, Y., Lim, J., Yang, M.-H.: Object tracking benchmark. 2015 IEEE Trans. Pattern Anal. Mach. Intell. J. **37**(9), 1834–1848 (2015)
13. Kristan, M., et al.: The visual object tracking VOT2016 challenge results. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 777–823. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_54
14. Danelljan, M., Bhat, G., Shahbaz Khan, F., et al.: ECO: efficient convolution operators for tracking. In: 2017 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, 21–26 July (2017)
15. Bertinetto, L., Valmadre, J., Golodetz, S., et al.: Complementary learners for real-time tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Seattle, WA, 27–30 June 2016, pp. 1401–1409 (2016)
16. Fan, H., Ling, H.: Siamese cascaded region proposal networks for real-time visual tracking. In: 2019 32st IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, 16–20 June (2019)