2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

A Selective Tracking and Detection Framework with Target Enhanced Feature

Xinyao Ding, Lian Li, Xin Zhang*
School of Electronic and Information Engineering
South China University of Technology
Guangzhou, China 510640
Email: eexinzhang@scut.edu.cn

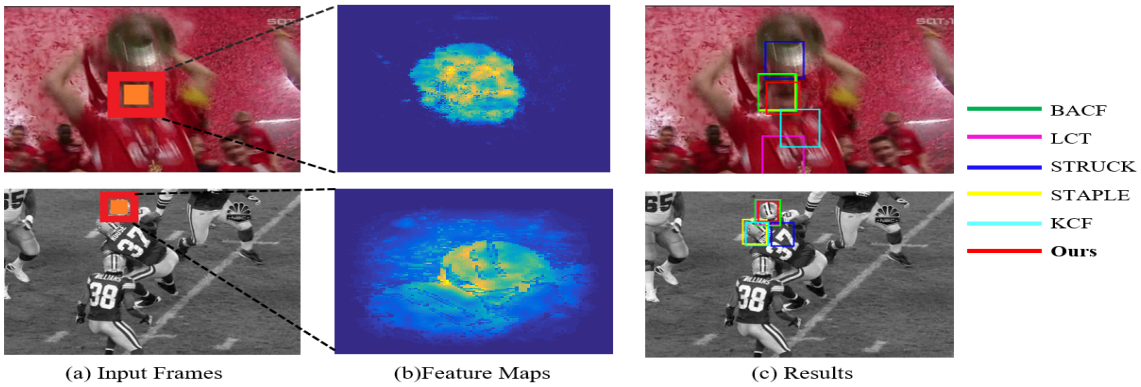

Fig. 1: Demonstration of one major contribution and results comparison. (a) Two sample frames with superimposed the foreground (central orange area) and background (surrounding red area); (b) Foreground probability feature map (FP map) to enhance the foreground target and suppress the surrounding background; (c)Comparison with state-of-the-art methods. Our proposed method shows the best performance.

***Abstract*—In the long time tracking, object representation and occlusion handling are two important challenges. We propose a novel selective tracking and detection framework in which a new probabilistic object-enhanced feature is integrated. Firstly, besides precise object appearance feature, we believe the neighboring foreground-background contrast is another key factor in the tracking. Hence we propose a foreground probability map to enhance the target and weaken the surrounding background. It is computed based on the object color distribution and its comparison with the surrounding background. Secondly, we introduce the selective tracking and detection framework that has two sets of conditions to control the detector activation and final result selection. The detector will only be activated when the tracker is not trustable, which is determined by the tracking confidence and foreground parochiality value. Then, given the tracking and detection results, the final output is selected in terms of their individual correspondence values. We have evaluated our methods on two popular benchmark datasets. Extensive experiments demonstrate that our algorithm performs favorably comparing with state-of-the-art methods.**

Keywords*—object tracking; target enhancement; tracking and detection; selective mechanism

I. Introduction

Object tracking is an important research topic in the computer vision field with extensive applications [2]. To infer the accurate object location, we face several challenging issues such as object variation and deformation, illumination changes, partial occlusions, background clutters and so on.

Recently, the discriminative correlation filter (DCF) based trackers [3], [6], [7], [8], [11] and deep convolutional neural networks (CNN) based trackers [4], [20], [21], [22], [23] are leading tracking algorithms. CNNs based trackers exploit the rich features to achieve robust tracking results but they usually require large data set for training. DCF based approaches present conspicuous performance in terms of accuracy and speed. DCF mainly relies on hand-crafted features like HOG [1] and color features [10]. The HOG feature has better performance in light and motion blurred scenes, while color features are better for deformed or rotating scenes. In [12], these two features are combined by the weighted linear summation and nice results are achieved. Object appearance features are subject to change when the illumination varies and the object rotates. Hence, online feature updating and surrounding contrast feature are essential. Additionally, the boundary effect is the critical problem of DCF framework, which is caused by the circular shifted examples. The application of the cyclic matrix in DCF can speed up the calculation. However, the negative samples produced by cyclic shift are not reasonable in real-world scenes, which can degrade the robustness of the tracking model [15]. Spatially regularized DCFs (SRDCF) [13], correlation filter with limited

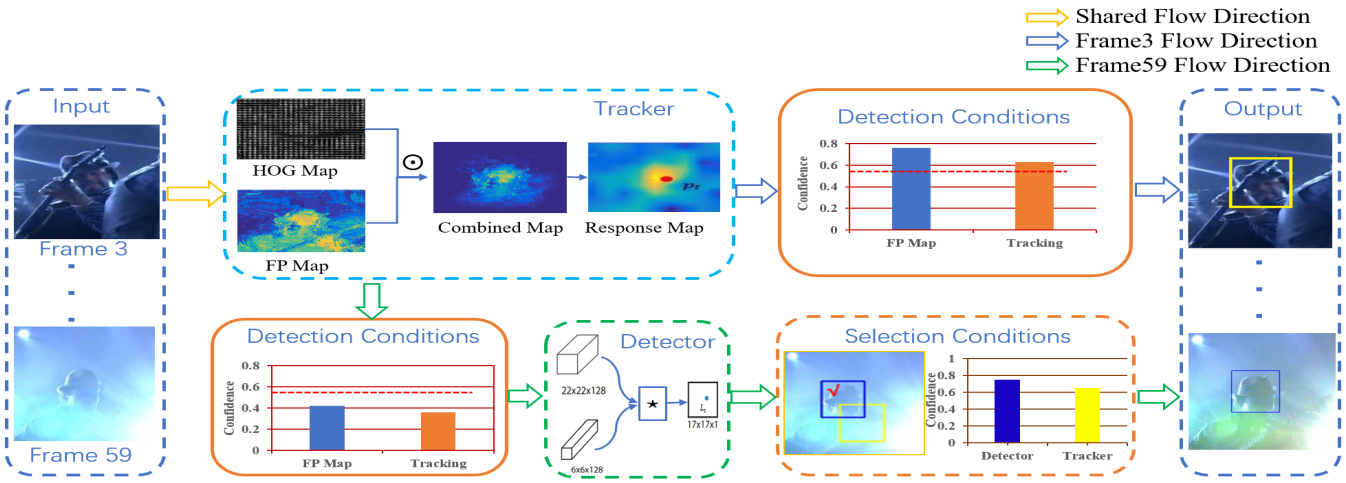

Fig. 2: The flow chart of our proposed algorithm. There are four main modules, i.e., tracker, detector, detection conditions and selection conditions. *Tracker* and *detection conditions* modules are basic blocks for every frame. If *detection conditions* are true, the *detector* is waked up, as Frame59 shown(green arrow). The final output is chosen between the *tracker* and *detector* result according to the *selection conditions*.

boundaries (CFLB) [14], background-aware correlation filters (BACF) [15] were proposed to improve DCF. We adopt the BACF framework to avoid the bounding effect in the long time tracking. There are some long-term tracking approaches [16], [17], [18], [24] proposed to alleviate the tracking drift. For example, TLD [19] proposes to use the combination of tracking, learning and detection modules for long term object tracking. Inspired by human brain memory, MUSTer [24] utilized the combination of long-term store and short-term store to achieve a robust tracking. Introducing detection can deal with the tracker failure but when to switch and which result to trust are key factors.

In this paper, we make the following two main contributions. First, it is believed that the surrounding background interference and similarity may lead to the tracking failure. Hence, we propose to increase the target-background contrast by the foreground probability map (FP map) feature, which is computed based on the pixel-wise foreground-background color distribution statistics. Then, FP map is combined with HOG feature by dot product to obtain a robust target representation. Second, to solve the tracking drift and occlusion issues, we propose the conditional detection mechanism which controls when to activate the detector and which result to choose. The mechanism has two decision modules, i.e., detection conditions and selective conditions. The detection conditions decide whether to activate the detector according to the FP map feature confidence and the basic tracking result confidence. The selective condition can help us determine whether the detector result is better than the tracking one. The selection criterion is based on their individual correspondence. These two sets of conditions make our algorithm suitable for the long-term tracking. Comparing with state-of-the-art methods, experimental results on VOT2015 and OTB two benchmarks validate the proposed framework has very promising performance and its speed is relative fast.

II. Proposed Algorithm

We discuss our method as three parts: basic tracking framework, combined feature with foreground probability (FP) map and conditional detection mechanism.

A. Basic Tracking Framework

As mentioned above, our method has four modules shown in Fig. 2. Every frame has to pass through *tracker* and *detection conditions* two modules. We employ BACF [15] as the baseline tracker with the new proposed feature, foreground probability map, to emphasize the object. Then in the *detection conditions* module, two confidence values are both higher than the red line, which means the tracking result can be trusted, as Frame 3 shown in Fig. 2. The red line is the reference criteria of the sequence. When there is any potential disturbance causing any one of the two confidence values is lower than threshold, as Frame59 shown in Fig. 2, the *detector* is activated. In the last *selection conditions* module, the tracking and detection results are inputs, and the final output is chosen between them according to their corresponding confidence coefficients.

B. Combined Feature with Foreground Probability (FP) Map

To obtain a robust representation, we propose a foreground probability (FP) map that enhances target and suppresses the background significantly. FP map can be further combined with HOG feature to help DCF infer the target location from the searching area.

Given the tracking area of previous frame $\mathbf{Z}$, every pixel in the area has R, G and B three color channels and we quantize their value range to [1,32]. That is, each color channel has 32 possibilities and then every pixel value will have $32 \times 32 \times 32$ possibilities. Supposed that the target area size is $w \times h$ at frame $t-1$. We denote that the box of $(w-k)\times(h-k)$ selected

area around the target center is the foreground area $\mathbf{Z}_f$. The area between the target area and expanded area of $(w+k) \times (h+k)$ denotes the background area $\mathbf{Z}_b$. k equals $\alpha*(w+h)/2$, where α is a parameter to decide appropriate foreground and background area. To compute FP map, we need to compute foreground and background dictionary models $\beta_{t-1}(F)$ and $\beta_{t-1}(B)$ from previous frame $t-1$ respectively. The dictionary models are computed as,

$$\beta^j_{t-1}(F) = \frac{\sum_{i=1}^{N_F} \phi(z_i)}{N_F}, z_i \in \mathbf{Z}_f \tag{1}$$

$$\beta^j_{t-1}(B) = \frac{\sum_{i=1}^{N_B} \phi(z_i)}{N_B}, z_i \in \mathbf{Z}_b \tag{2}$$

where z_i denotes a vector with the pixel values of three channels and j is a three-dimensional index vector. N_F and N_B denotes the number of pixel in foreground and the background area respectively. The function ϕ is used to judge whether the pixel value z_i is the same as the j dimension index value in the dictionary. If they are same, ϕ return 1. Otherwise, ϕ return 0. Then we can get the probability value in every dimension of the two dictionary models. So both dictionary models are matrices of $32 \times 32 \times 32$ dimensions and they are updated every frame with a fixed rate.

Given the current frame I_t, the input searching patch $\mathbf{Z}_s$ is $(w+k) \times (h+k)$ according to the tracking result of previous frame. With the foreground and background two dictionary models at frame $t-1$, we can compute two probability models $\omega(F)$ and $\omega(B)$. $\omega(F)$, at frame t, denotes the probability map of the searching area pixels being in the foreground area according to $\beta_{t-1}(F)$. And $\omega(B)$, at frame t, denotes the probability map of the searching area pixels being in background area according to $\beta_{t-1}(B)$. According to the value of each pixel in $\mathbf{Z}_s$, we search the corresponding index vector in $\beta_{t-1}(F)$ and take the corresponding probability value in the dictionary as the probability of the pixel in the foreground area in $\omega(F)$. It's the same for $\omega(B)$ with $\beta_{t-1}(B)$ model. The FP map is obtained as

$$f_{FPM} = \frac{\omega(F)}{\omega(F) + \omega(B)} \tag{3}$$

where $\frac{\bullet}{\bullet}$ denotes pixel-wise division and $\bullet + \bullet$ denotes pixel-wise addition. The FP map f_{FPM} based on the color feature can have a clear description of the foreground-background probability. HOG map can clearly describe edge features. So we employ the **dot product** between the FP map and the HOG map as

$$f_{FINAL} = f_{FPM} \bigodot f_{HOG} \tag{4}$$

where f_{FPM} is the FP map and f_{HOG} denotes the HOG map in Fig.2. $\bigodot$ denotes pixel-wise multiplication.

C. Conditional Detection Mechanism

In this section, we employ the Siamese Network [25] as our detector. To wake up the detector at the right moment, we propose a standard regulation to estimate if there is a non-trustable tracking result. There are two conditions used to decide whether to wake up the detector. They are FP map confidence and tracking confidence respectively. Because the FP map feature is robust for illumination change, occlusion and some other problems, we use it to calculate the confidence to measure the tracking state. The confidence S_{FPM} according to FP map in Eq. 3 can be calculated by

$$S_{FPM} = max f_{FPM} \tag{5}$$

If $S_{FPM} < T_c S^0_{FPM}$, it indicates that the target could be out of tracking. T_c means the color threshold, S^0_{FPM} denotes the FP map confidence calculated in the first frame. In addition to the FP map confidence, we also make use of the tracking confidence S_T as a complement to it. The tracking confidence at frame t can be calculated as:

$$R_t = \mathcal{F}^{-1}(\hat{g}_{t-1} \bigodot \hat{f}_{FINAL}) \tag{6}$$

$$S_T = R_t(x_T, y_T) \tag{7}$$

where $\hat{g}_{t-1}$ is a BACF [15] correlation filter model at frame t-1 in frequency domain to catch the target response map, which get from an efficient Alternating Direction Method of Multipliers based approach. The detail analysis can be seen in [15]. $\hat{f}_{FINAL}$ is the final feature at frame t in frequency domain. R_t is the final confidence map. (x_T, y_T) is the target center position getting by tracker module in Fig. 2. S_T is the confidence value at (x_T, y_T). The detector will be activated while $S_T < T_t S^0_T$, where T_t means the tracking threshold, S^0_T denotes the tracking confidence calculated in the first frame.

It is highly possible that the detector result is not better than the tracking one. Hence, we suggest to compare them before making final decision. We compare the tracking confidence between the position (x_T, y_T) obtained from the tracker and the center position (x_D, y_D) obtained from the detector. According to Eq. 9, we can select which one to be used as the final tracking results (x_F, y_F).

$$M_t = \mathcal{F}^{-1}(\hat{g}_{t-m} \bigodot \hat{f}_{FINAL}) \tag{8}$$

$$(x_F, y_F) = \begin{cases} (x_D, y_D), & M_t(x_D, y_D) > M_t(x_T, y_T) \\ (x_T, y_T), & otherwise \end{cases} \tag{9}$$

The M_t is the original from BACF [15] model that is generated by dense bounding box without cosine windows. The difference between the M_t in Eq. 8 and the R_t in Eq. 6 is the update frequency. The R_t is updated every frame. However, only if it satisfies $M_t(x, y) > T_B$, where T_B is a model threshold, we free and update M_t. The variable m means there is continuous m-frame that don't satisfy the condition. Comparing to R_t, the model obtained by this method is not conducive to the short-term tracking because the update is not timely. However, it is useful for the long-term tracking due to its purity. Besides, the experiments in Fig. 3 can prove the validity of the model. If there is no challenging issues like Fig. 3(a1), the detector should not be activated. In Fig. 3(a2), there is a severe occlusion leading to the failure of the tracker. Then the FP map and tracking two confidence values are lower than the red line. The detector is activated and bring a detection

result. We need to make a selection between the detection result and the tracker result. From the bottom left bar in Fig. 3, we can see the detector confidence is higher than the tracker confidence which means the blue box should be selected as the final result. It is the opposite situation for Fig. 3(b2).

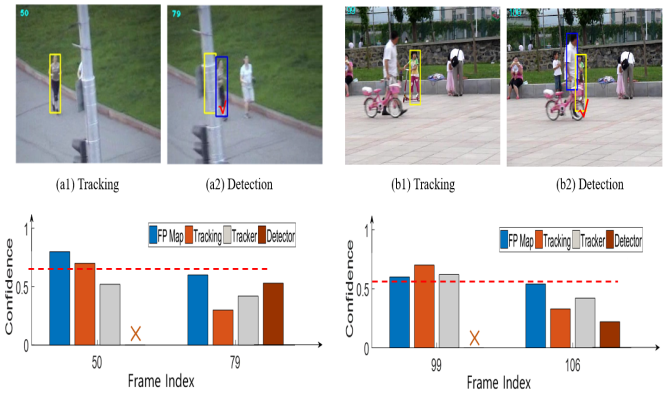

Fig. 3: The yellow box corresponds to the result of the tracker. The blue box corresponds to the result of the detector. The red line denotes the reference standard which is not constant according to different sequences. The symbol $\times$ denotes the detector is not be used.

III. Experiment Results

In this section, we evaluate our tracking mechanism on two recent popular standard benchmarks, OTB [26] and VOT2015 [27], and compare it with several state-of-the-art methods.

A. Analysis of Algorithm Effectiveness

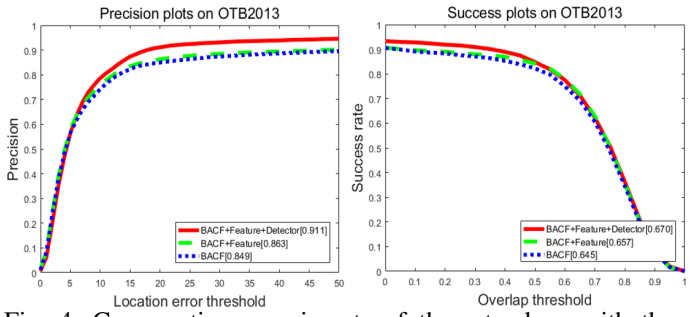

Fig. 4: Comparative experiments of three trackers with the basic tracker BACF on OTB2013.

We design three tracking models, i.e., BACF, BACF + FP map Feature and final method, to verify the effectiveness of proposed techniques. In Fig.4, BACF + Feature is higher than BACF in the precision score and success score. That can explain the robust features we get through the FP map in Eq. 3 play a useful role. Additionally, the result of final method verifies the effectiveness of the conditional detection mechanism.

B. OTB Benchmark

The OTB 2013 contains 51 test sequences and TB50 contains another 50 sequences. This benchmark is suitable for long-term tracking. We compare our method with 9 state-of-the-art algorithms, i.e., DeepSRDCF [9], SRDCF [13], BACF [15], MEEM [28], SAMF [29], LCT [30], KCF [8], Staple [12] and SiamFC [25]. Fig. 5 and Fig. 6 show the comparison of ten robust trackers. We use the one-pass evaluation on the OTB benchmark. Criterias for precision score and the success score are the average distance precision score at 20 pixels and the area under the curve plot, respectively. Our model reaches the best results. Comparing to baseline tracker BACF, our method has a clear improvement showing in the plots.

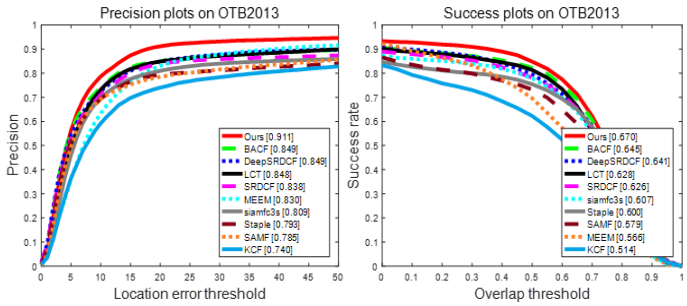

Fig. 5: Comparison of precision and success plots on OTB 2013 dataset.

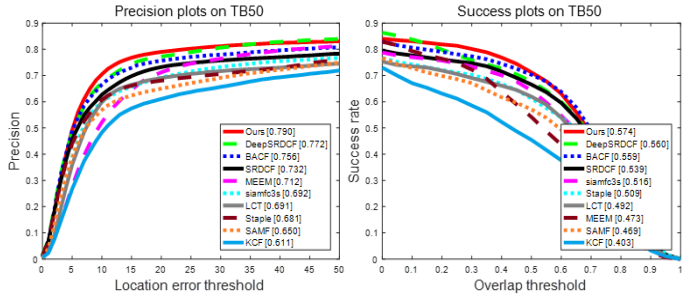

Fig. 6: Comparison of precision and success plots on TB50.

C. VOT Benchmark

The VOT benchmark is another popular evaluation data set. The VOT2015 set contains 60 test sequences and common evaluation platform. We present the comparison of our method with the other five outstanding trackers tested on VOT2015 in Table I. Our method also has the best results in terms of both accuracy and robustness.

TABLE I: Results on VOT2015

Trackers	Accuracy	Robustness
LDP	0.51	1.84
NSAMF	0.53	1.29
MUSTer	0.52	2.00
S3Tracker	0.52	1.77
BACF	0.50	2.03
BACF+Feature	0.52	1.62
Ours	**0.54**	**1.28**

IV. Conclusions And Future Works

In this paper, we proposed a novel selective tracking and detection framework combined a new probabilistic object-enhanced feature. The enhanced feature with FP map can weaken the surrounding feature of target and strengthen the feature of the target. Further, we introduce a conditional detection mechanism to assist our tracker in the long-term tracking.

Extensive experiments illustrate that our tracking method is effective and robust. Our future work will incorporate deep learning with the enhanced feature and do more explorations on the whole framework.

REFERENCES

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition*(Vol.1, pp.886-893) , 2005.

[2] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," in *TPAMI*(pp.14421468), 2014.

[3] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan,"Foreground Detection with Discriminatively Trained Part-Based Models," in *IEEE transactions on pattern analysis and machine intelligence*, 32(9), pp.1627-1645, 2010.

[4] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *International Conference on Machine Learning* (pp. 597-606), 2015.

[5] M. Danelljan, G. Hager, F. S. Khan and M. Felsberg, "Accurate Scale Estimation for Robust Visual Tracking," in *Proceedings of the British Machine Vision Conference (BMVC)* , 2015.

[6] D. S.Bolme, J. R.Beveridge, B. A.Draper, and Y. M.Lui,"Visual foreground tracking using adaptive correlation filters," in *Computer Vision and Pattern Recognition* (Vol.119, pp.2544-2550). IEEE, 2010.

[7] J. F. Henriques, C.Rui, P. Martins, and J. Batista, "Exploiting the Circulant Structure of Tracking-by-Detection with Kernels," in *European Conference on Computer Vision* (Vol.7575, pp.702-715). Springer-Verlag, 2012.

[8] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-Speed Tracking with Kernelized Correlation Filters," in *IEEE Transactions on Pattern Analysis and Machine Intelligence* , 37(3), pp. 583-596, 2015.

[9] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision Workshops* (pp. 58-66), 2015.

[10] M. Danelljan, F. S.Khan, M. Felsberg, and J. V. D.Weijer, "Adaptive Color Attributes for Real-Time Visual Tracking," in *IEEE Conference on Computer Vision and Pattern Recognition* (pp.1090-1097), 2014.

[11] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *European Conference on Computer Vision* (pp. 472-488). Springer International Publishing, 2016.

[12] L.Bertinetto, J.Valmadre, S.Golodetz, O.Miksik, P.H.S.Torr, "Staple: Complementary learners for real-time tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1401-1409) ,2016.

[13] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision* (pp. 4310-4318), 2015.

[14] H. Kiani Galoogahi, T. Sim, and S. Lucey, "Correlation filters with limited boundaries," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4630-4638), 2015.

[15] Galoogahi, Hamed Kiani, A. Fagg, and S. Lucey, "Learning Background-Aware Correlation Filters for Visual Tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[16] K. Lebeda, S. Hadfield, J. Matas, and R. Bowden, "Long-term tracking through failure cases," in *Proceedings of the IEEE International Conference on Computer Vision Workshops* (pp. 153-160), 2013.

[17] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Computer Vision and Pattern Recognition (CVPR)* (pp. 1177-1184). IEEE, 2011.

[18] F. Pernici and A. Del Bimbo, "Foreground tracking by oversampling local features," in *IEEE transactions on pattern analysis and machine intelligence*, 36(12):25382551, 2014.

[19] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning detection," in *IEEE transactions on pattern analysis and machine intelligence*, 34(7):14091422, 2012.

[20] Y .Song , C.Ma , L.Gong, J. Zhang, R. Lau and M. H. Yang, "CREST: Convolutional Residual Learning for Visual Tracking," in *IEEE International Conference on Computer Vision* (Vol. 2), 2017.

[21] B .Han, J .Sim, H.Adam, "BranchOut: Regularization for Online Ensemble Tracking with Convolutional Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2217-2224), 2017.

[22] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in neural information processing systems* (pp. 809-817), 2013.

[23] Wang L, Ouyang W, Wang X and H. LU, "Visual Tracking with Fully Convolutional Networks," in *Proceedings of the IEEE International Conference on Computer Vision* (pp. 3119-3127), 2015.

[24] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov and D. Tao, "MUlti-Store Tracker (MUSTer): A cognitive psychology inspired approach to foreground tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 749-758), 2015.

[25] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi and P. H. S. Torr, "Fully-convolutional Siamese networks for foreground tracking," in *European Conference on Computer Vision* (pp. 850-865). Springer International Publishing, 2016.

[26] Y. Wu, J. Lim, and M.-H. Yang, "Online foreground tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2411-2418), 2013.

[27] M. Kristan et al, "The Visual Foreground Tracking VOT2015 Challenge Results," in *Proceedings of the IEEE international conference on computer vision workshops* (pp. 1-23), 2015.

[28] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: robust tracking via multiple experts using entropy minimization," in *European Conference on Computer Vision* (pp. 188-203). Springer, Cham, 2014.

[29] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *ECCV Workshops* (2) (pp. 254-265), 2014.

[30] C. Ma, X. Yang, C. Zhang, and M. Yang, "Long-term correlation tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5388-5396), 2015.