Siamese Network Based Features Fusion for Adapative Visual Tracking

Dongyan Guo[1], Weixuan Zhao[1], Ying Cui[1,2], Zhenhua Wang[1], Shengyong Chen[1], and Jian Zhang[3]

[1] College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou, 310023, China
[2] Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, Nanjing University of Science and Technology, Nanjing, 210094, China
[3] Global Big Data Technologies Centre, University of Technology Sydney, Sydney, 2007, Australia
cuiying@zjut.edu.cn

Abstract. Visual object tracking is a popular but challenging problem in computer vision. The main challenge is the lack of priori knowledge of the tracking target, which may be only supervised of a bounding box given in the first frame. Besides, the tracking suffers from many influences as scale variations, deformations, partial occlusions and motion blur, *etc.*. To solve such a challenging problem, a suitable tracking framework is demanded to adopt different tracking scenes. This paper presents a novel approach for robust visual object tracking by multiple features fusion in the Siamese Network. Hand-crafted appearance features and CNN features are combined to mutually compensate for their shortages and enhance the advantages. The proposed network is processed as follows. Firstly, different features are extracted from the tracking frames. Secondly, the extracted features are employed via Correlation Filter respectively to learn corresponding templates, which are used to generate response maps respectively. And finally, the multiple response maps are fused to get a better response map, which can help to locate the target location more accurately. Comprehensive experiments are conducted on three benchmarks: Temple-Color, OTB50 and UAV123. Experimental results demonstrate that the proposed approach achieves state-of-the-art performance on these benchmarks.

Keywords: Deep Learning · Siamese Network · Object Tracking · Feature Fusion

1 Introduction

Visual object tracking is one of the hotspots in computer vision. Object tracking is widely employed in many real-world visual applications, such as autonomous driving, video surveillance, human-computer interaction, *etc.*. The task of object tracking is estimating the trajectory of an object in an image sequence. However,

the only knowledge about the object is that the target location in the first frame. The lack of priori knowledge makes the task challenging. Besides, the problem is challenged from many influences such as illumination variations, scale variations, non-rigid deformations, fast motion, background clutters, motion blur and occlusions.

In recent years, correlation filter based methods have shown excellent performance on object tracking benchmarks [30]. However, most of these methods only use hand-crafted appearance features to present the tracking target, which cannot get satisfactory performance in some scene applications like occlusions [6, 7, 31], background clutters [2, 19, 31], *etc.*. In the process of object tracking, most of the existing adaptive model based approaches which update the model continuously through the tracking process can achieve better performance [2, 6, 11, 24]. The target information in later frames can make the adaptive model become more accurate. However, in the other side, the model may be updated with some negative information such as target losing. With the accumulation of these small negative errors, the performance of the model become worse. Finally, these small errors may lead to model drift and target lost.

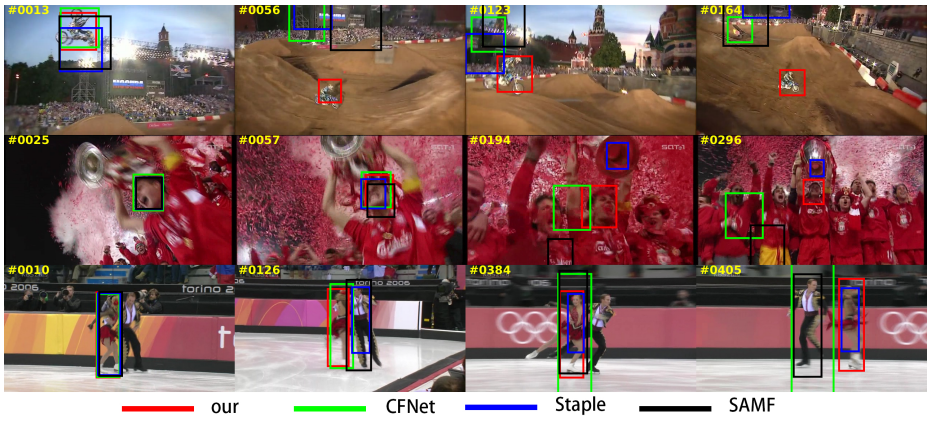


Fig. 1. A qualitative comparison of our approach with other three state-of-the-art approaches on three example sequences. It is shown the three sequences results: Motor Rolling (top row), Soccer (middle row) and skating2 (bottom row). These example sequences include these cases: scale variation, occlusion, deformation, fast motion, out-of-plane rotation, in-plane rotation and background clutters. Our approach achieves superior results in these scenarios.

Deep neural networks can train powerful models with large numbers of labeled training samples. When enough priori knowledge of the target is obtained, the deep neural networks can achieve excellent performance in many application scenes. However, the lack of priori knowledge about the target is the main challenge for training deep neural networks in object tracking task. Moreover, the

training of deep neural networks is time consuming for real time on-line training and tracking.

A possible way to solve the above problems is to train the deep neural networks model offline. Some existing works adapt a pre-trained model for the target to get CNN (Convolutional Neural Network) features [8, 10, 22]. Though the pre-trained model bypass the online learning problem, its fixed metric prevents the learning strategy from exploiting the sense-specific cues which is important for discrimination. Some approaches use Siamese CNN architecture, which is a non-online adaptation network [3, 5, 14, 18, 26]. Siamese CNN is trained offline but have excellent performance in discriminating whether or not the same object in two image patches. According to some research works, combining on-line learning method with pre-trained CNN features has obtained successful improvement. For example, with deep integration of CNN and correlation filter, some investigators take the correlation filter as a network layer [27].

However, it is difficult to achieve satisfactory tracking results with single feature for both on-line and off-line learning strategy. Each feature has its own disadvantages. To overcome the shortcoming, combining different features is a good way to apply in the object tracking task. For example, HOG (Histogram of Orientation Gradient) presents the oriented gradients histograms for an image and it is also a general feature which has employed in many state-of-the-art methods [2, 6, 7, 16]. Those trackers achieve excellent performance in scenarios with little deformation and occlusions. But the HOG features based method has its drawbacks, for example, it is sensitive to large deformation. The trackers perform poorly when the object change rapidly. However, the CNN features are powerful in image representations, it is not sensitive to deformation. It turns out that as long as there are enough diverse training samples, the CNN features can achieve excellent performance even in scenarios with large object variations and background clutters. The one shortcoming is that if the training samples are not enough and lacking some kinds of scenes, the performance of CNN will drop very fast.

A key issue in general object tracking is designing general object descriptors to describe object discriminatively with any class. In this paper, we propose a Siamese Network framework that combines the CNN features with hand-crafted appearance feature for adaptive robust object tracking and achieve excellent performance (see figure 1). The features fused in this paper are CNNs and HOGs. Moreover, our network can not only fuse CNN and HOG features, but also integrate CNN features with more features. In the network, we apply the Correlation Filter to generate a discriminative template for CNN and HOG features respectively, which can be used to get CNN and HOG response maps. Thousands of parameters have been trained through the Siamese Network framework to improve the CNN and HOG features fusion results. The improvement is beneficial to the tracking performance. The architecture of our network is shown in Figure 2. In general, the network architecture we proposed can be divided into three parts. The feature extraction layer are utilized to extract different features from the training samples and testing samples. The training image is an image patch

of the previous frame that contains the tracking target. The test image is the current frame to be searched of. The template generation layer utilize the features extracted from the traing image to generate the corresponding template. The corresponding response maps can be obtained by the convolution of the feature maps and the discriminative templates. And finally, the multiple response maps are fused by fusion layer to generate the final response. The final response map help to locate the target location more accurately. Experimental results show that our approach is a robust general tracker, and achieves state-of-the-art performance on multiple benchmarks. Code is avaliable online[4].

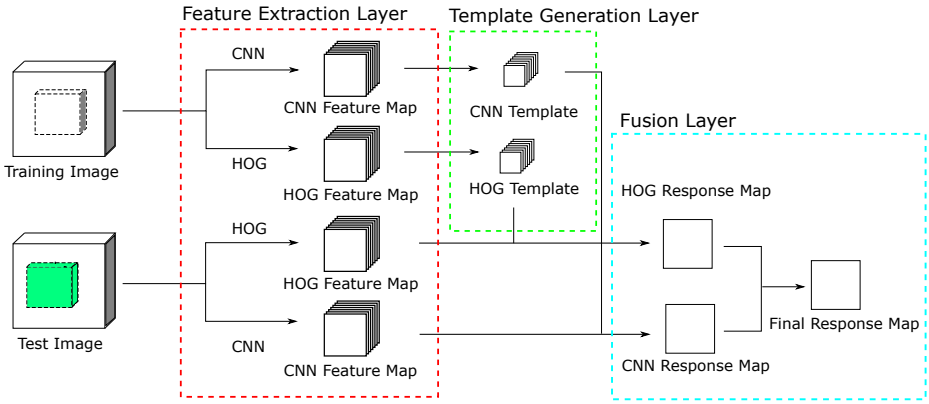


Fig. 2. The architecture of our proposed network.

2 Related Work

The mainstream object tracking methods can be divided into two categories — generative [1, 21, 25] or discriminative [13, 15, 32] approaches. Generative model approaches use the statistical models or templates to describe the object. The generative model approaches consist of Kalman filtering, Particle filter, mean-shift and so on. Discriminative model approaches use the machine learning to train classifier by taking the object as the positive samples and the background as the negative samples. And then, use the classifier to find the optimal region of target frame by frame. The Support Vector Machine (SVM) is a classical machine learning algorithm used in discriminative model approaches. Many approaches like Struck tracker [13] employ haar features and structured SVM to achieve the tracking task.

The Discriminative Correlation Filters (CF) is an outstanding method of discriminative model approaches. In recent years, these trackers have been employed in the CF and achieved excellent performance on tracking bechmarks.

[4] https://github.com/needniming/SNBFF

The MOOSE tracker [4] is the work of Bolme et al. which is the first tracker that used the CF. These trackers like CSK [15] and MOOSE use the raw pixels combined with CF to estimae the trajectory of an object. However, these methods could not take advantage of image features, and tracking performance is extremely limited. With the help of HOG features, KCF/DCF [16] improve the tracking results. But the HOG is sensitive to deformation while the tracker CN [11] demonstrates that the color feature is robustness to deformation. In any case, it is difficult to achieve satisfactory tracking results with single feature. With the development of deep learning, the method based on deep learning is developing rapidly in the visual object tracking. Recent works have focused on learning universal object descriptors to achieve tracking task. These methods [3, 5, 14, 18, 26, 27] are based on the Siamese CNN architecture. The network is trained offline, so it can take advantage of information present in numerous training images. The GOTURN tracker [14] based on this architecture can run 100FPS in GPU mode. However, the performance of this method is not satisfactory. End-to-end representation learning method SiameseFC [3] and CFNet [27] get excellent performance in aspects of speed and results. The network we designed combines different features to improve the generality of the tracker, and achieves state-of-the-art performance on multiple benchmarks.

The method of combining multiple estimates can improve tracking results. The Staple [2] tracker combines HOG and color histogram together to make up for the defect of the two features, so it can make the tracker robust to deformation. The tracker in [28] use a factorial HMM to combine the results of five independent trackers. The MEEM [31] tracker stores a collection of past models. For each frame, the tracker can obtain an evaluation result equal to the number of storage models. Using the loss entropy function, an optimal one is selected from these results. Our approach differs from these approaches in that a) our approach based on a deep neural network architecture is a end-to-end method. b) A deep intergation of different features are achieved in our network by training correlation weights, and these features can describe the object more elaborate. c) We add a fusion layer in the network to fuse differrnt response maps. The output of the fusion layer can help to locate the target more accurate.

3 The Proposed Approach

We briefly introduce our proposed network framework in section 3.1. And then, the usage of the Correlation Filter to generate CNN and HOG template is explained in section 3.2 and the fusion approach is presented in section 3.3. In the last, we illustrate the use of the fusion model for object tracking in section 3.4.

3.1 Siamese Network framework

The starting point of this paper is to design a network to combine different features more compact. The CFNet [27] uses the CNN features for visual object tracking and get state-of-the-art performance in the OTB benchmarks [29,

30]. However, using the CNN features should face a key problem, if the training samples are not enough and lack some scenes, the CNN features may not achieve satisfactory tracking results in some scenes which are not contained in the training samples. We find that combining the CNN with HOG can improve the universal property of the tracker.

The network we proposed is used to fuse the response map of features. The input of the network is pairs of image patches (x', y'). The image x' represents the object of interest in the x_{th} frame of an image sequence and the object is in the middle of the image x'. Moreover, the image y' represents the object search area in the $x + 1_{th}$ frame of an image sequence and the size of y' is larger than x'. The y' is extracted from $x + 1_{th}$ frame based on the object location in the x_{th} frame.

HOG and CNN features are extracted from the two inputs respectively. Here we utilize the fuction f_c and f_h to extract CNN and HOG features from an image respectively. The parameters used in feature extraction function f are trained by our proposed network. A pair of image patches can yield four feature maps (two CNN feature maps $f_c(x')$, $f_c(y')$ and two HOG feature maps $f_h(x')$, $f_h(y')$) which can get two response maps after cross-correlated operation:

$$g_c(x', y') = f_c(x') \star f_c(y') \tag{1}$$

$$g_h(x', y') = f_h(x') \star f_h(y') \tag{2}$$

Eq. 1 gets the CNN feature response map while eq. 2 gets the HOG feature response map. In order to get a better response map, it is necessary to fuse the CNN response map and HOG response map.

$$g(x', y') = M_\rho(g_c, g_h) \tag{3}$$

Here, the maximum value of the response map $g(x', y')$ is related to the center of the target. The function M is used to represent the fusion approach and the ρ is the learnable parameters. More details about the function M will be explained in section 3.3.

To obtain the model, the network is trained offline. The training image samples of the network are millions of random pair (x_i', y_i'). Each image pair has a spatial map of label information which is composed of $\{-1, 1\}$. The label is represented whether the pixel point is belonging to the ground truth or not.

$$L_i(r, c) = \begin{cases} -1, & \textit{not belong to ground truth} \\ 1, & \textit{belong to ground truth} \end{cases} \tag{4}$$

Here, r and c represent the row number and col number of the spatial map. The purpose of the network training is minimizing the element-wise logistic loss function ℓ:

$$\arg\min \sum_i \ell(g(x', y'), L_i) \tag{5}$$

3.2 Correlation Filter

The Correlation Filter is an algorithm to train a liner template for discriminating the relationship of image and image transformation. The problem of solving the correlation filter template is equivalent to solving the ridge regression problem. In the following, the correlation filter template is denoted as w, $x \in \mathbb{R}^{m \times m \times K}$ is a K-channel feature image, $y \in \mathbb{R}^{m \times m}$ is the desired response map. In our network, the CNN and HOG feature maps of training image x' are all belong to feature image x. Under a least-squares Correlation Filter formulation, the problem can be represented as:

$$\arg\min_{w} \|w \star x - y\|^2 \tag{6}$$

where symbol $\star$ denotes the circular cross-corrleation. To avoid overfitting, we should add the quadratic regularization into eq. 6 and get:

$$\arg\min_{w} \|w \star x - y\|^2 + \lambda \|w\|^2 \tag{7}$$

To solve the problem, and obtain the optimal template w, we set and expand $F(w)$

$$\begin{aligned} F(w) &= \|w \star x - y\|^2 + \lambda \|w\|^2 \\ &= (w \star x - y)^T (w \star x - y) + \lambda w^T w \end{aligned} \tag{8}$$

The optimal template w can then obtained by solving the equation $\frac{d(F(w))}{dw} = 0$,

$$w = \frac{y}{x \star x + \lambda} \star x \tag{9}$$

It is time-consuming to solve eq. 9 in time domain. To avoid the problem, we can make fast Fourier transform for eq. 9,

$$\hat{w} = \frac{\hat{y}^* \circ \hat{x}}{(\hat{x}^* \circ \hat{x}) + \lambda} \tag{10}$$

where $\hat{w}$ represents the value of w in the frequency domain, the x^* denotes conjugation and the symbol $\circ$ denotes the element-wise multiplication. And introducing inverse fast Fourier transform $\hat{w}$ can get the optimal template w.

3.3 Feature Fusion

In section 3.1, we briefly introduce our proposed network framework, the feature extraction layer of the network can extract features from pairs of image patches (x', y'). These features can obtain two response maps by correlation filter operation. In order to make better use of these response maps, set different weights, and fuse these together to obtain a new response map. The fusion approach can make full use of the advantages of the two features and make up for the deficiency between the two features, therefore the tracking performance could be

improved. In section 3.2, the feature templates are obtained through correlation filter, therefore eq. 1 and eq. 2 can be represented as:

$$g_c(x', y') = w(f_c(x')) \star f_c(y') \tag{11}$$

$$g_h(x', y') = w(f_h(x')) \star f_h(y') \tag{12}$$

where the function $w(x)$ is represented to get the optimal template w. The fusion approach can use the eq. 13 to represent,

$$m(x', y') = \sum_{d=1}^{D} g_d(x', y') * k_d \tag{13}$$

where D is the amount of features, k_d is the fusion kernal which trained by our network.

In the last, in order to make the response map more suitable for logistic regression, the scale and bias are added into $m(x', y')$ to get the function $M(x', y')$,

$$M(x', y') = sm(x', y') + b \tag{14}$$

In order to make the fusion result become better, all the parameters are trained through the network .

3.4 Visual Object Tracking Algorithm

The network needs a pair of image patches as input. The input of the model consists of the target region in the previous frame and the search region in the current region. The search region is extracted as a sub-window centred at the previously estimated position which size is four times of the object. The output of the model is the fusion response map. The maximum value of the response is corresponding to the center of the object.

Although the model is trained offline, we find that the updating strategies using online learning can improve the experimental results. When a pair of image patches is inputted to the model, two new template $w(f_c(x'))$ and $w(f_h(x'))$ are obtained. The approach fuses new feature template with old feature template is shown in eq. 15:

$$\begin{aligned} Temp_{c,new} &= (1 - \eta_c) Temp_{c,pre} + \eta_c w(f_c(x')), \\ Temp_{h,new} &= (1 - \eta_h) Temp_{h,pre} + \eta_c w(f_h(x')). \end{aligned} \tag{15}$$

where the parameter η represents the learning rate of the template in online tracking.

4 Experiments

We evaluate our proposed network by performing contrast experiments on three benchmarks: Temple-Color [20], UAV123 [23], and OTB50 [29, 30]. The fundamental purpose of our experiments is to evaluate the effect of using our network

to train parameters for feature fusion during training. First, we compare the effects of different convolutional layer depths on the tracking performance. And then, we compare our approach with some state-of-the-art trackers on benchmarks. Bounding box overlap ratio and center location error are two metrics to evaluate the trackers. The bounding box overlap ratio is defined to measure the bounding boxes overlap of ground truth R_{gt} and the tracker's predict result R_t.

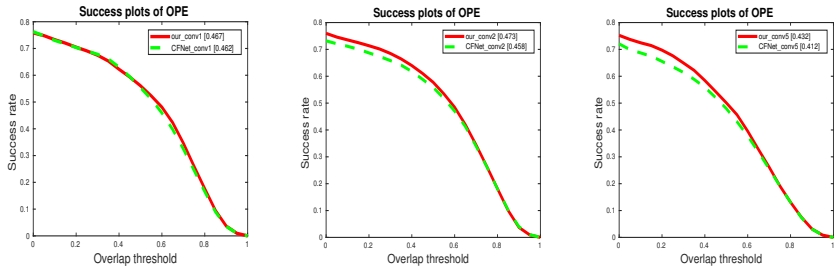
$$S(\sigma_{over}) = \frac{R_{gt} \cap R_t}{R_{gt} \cup R_t} \geq \sigma_{over} \tag{16}$$

The center location error is defined as the bounding box center Euclidean distance between ground truth P_{gt} and the trackers predict result P_t.

$$P(\sigma_{succ}) = \|P_{gt} - Pt\| \leq \sigma_{succ} \tag{17}$$

4.1 Evaluation of different convolutional layer depths

In this part, we use the bounding box overlap ratio to evaluate the trackers. The success plot is calculated as the percentage of frames with an intersection-over-union (IOU) overlap exceeding a threshold. The Temple-Color is the validation dataset in this part. Since we can only get the model of CFNet [27] using Conv-1, Conv-2 and Conv-5, our approach uses the same convolutional layers to do comparison. The results are shown in Figure 3. In Figure 3, we can find that the result of our approach is better than CFNet. It proves that the combination of the CNN and HOG features improves the performance of the tracker. To show the tracking results of HOG fused with varying convolutional layers, we choose the results when the overlap threshold is 0.5. The results are shown in Figure 4. In Figure 4, we find that the Conv-2 can achieve better results, when more convolutional layers are added it seems to be redundant.



(a) conv1 (b) conv2 (c) conv5

Fig. 3. Success rates of rectangle overlap for different convolutional layers on the validation dataset Temple-Color.

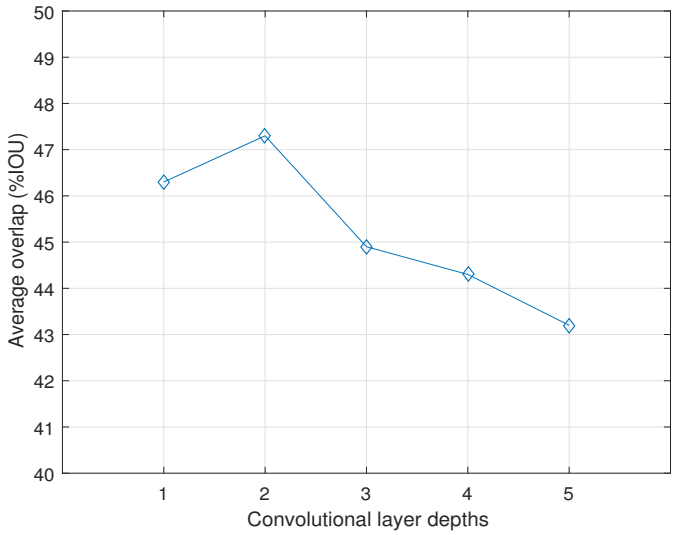


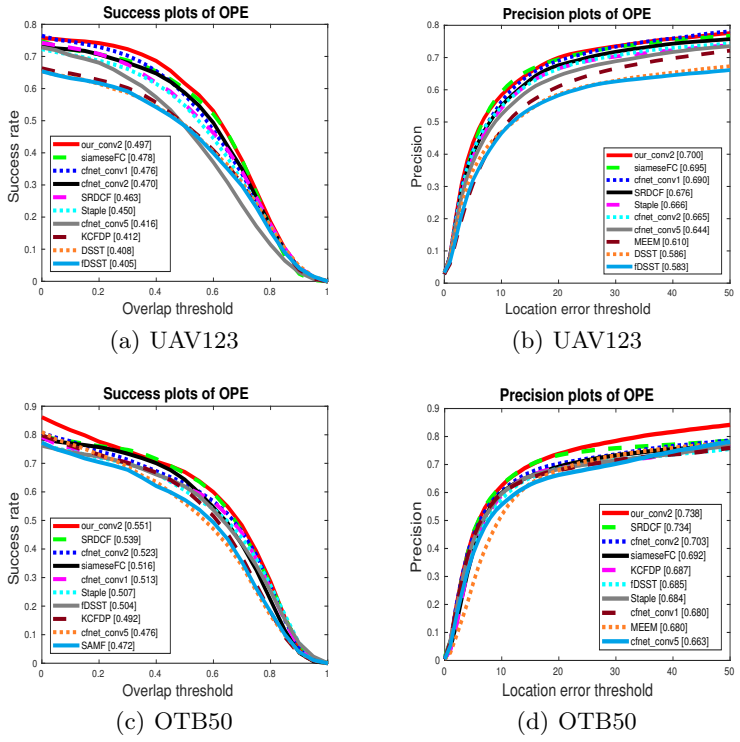
Fig. 4. The accuracy with different convolutional layer depths of our approach.

4.2 Comparisons with state-of-the-art methods

We compare our proposed approach with 13 state-of-the-art trackers: KCF [16], Staple [2], SAMF [19], SiameFC [3], CFNet [27], MEEM [31], SRDCF [9], DSST [6], DAT [24], ACT [11], TGPR [12], KCFDP [17] and fDSST [7]. Our experiments are using success plot and precision plot. The comparisons are done on the benchmarks UVA123 and OTB50, detailed as follows.

UAV123: UAV123 is a very large dataset which is captured from low-altitude UAVs. The dataset consists of sequences from an aerial viewpoint, containing a total of 123 video sequences and more than 110K frames. Figures 5 (a), (b) show the results of precision and success rate respectively. Among the comparison with the Siamese Network based approach, SiameseFC and CFNet provide the best results with AUC scores of 47.8% and 47.6% respectively. Our approach provides a better performance with an AUC score of 49.7%.

OTB50: OTB2013, OTB50 and OTB100 are commonly used OTB datasets in comparative experiments. OTB50 is the most challenging dataset of these OTB datasets, so the experiment only compare in the OTB50. The dataset contains 50 video sequences. Figures 5 (c), (d) show the results of precision and success rate respectively. Figure 5 show that our approach achieves state-of-the-art results on UAV123 and OTB50 datasets. Compared with SRDCF, employing hand-crafted features, our approach achieve a better performance with an AUC score of 55.1%. Compared with the deep features trackers SiameseFC and CFNet, our approach also achieves a better performance.



(a) UAV123 (b) UAV123

(c) OTB50 (d) OTB50

Fig. 5. Success plots on the UAV-123 (a) and OTB50 (c) datasets. Precision plots on the UAV123 (b) and OTB50 (d) datasets. The score of each tracker is shown in the legend. Our approach achieves state-of-the-art performance in all datasets. For clarity, only the results of top 10 trackers are shown in the legend.

5 Conclusion

In this paper, we propose a novel approach based on Siamese Network for robust visual object tracking. The training of the network model makes up the defect of different features in the tracking effect. Our feature fusion network improves the generality of the tracker, achieves excellent performance in scenes with fast motion, motion blur, background clutters and so on. Furthermore, our approach achieves the state-of-the-art performance on UAV123, OTB50 and Temple-Color. It also shows that the deep network model is trained with a large amount of data has a good application prospect in the object tracking, and the work based on Siamese Network is worthy for further study.

Acknowledgments. This work was supported in part by Natural Science Foundation of Zhejiang Province (LQ18F030013, LQ18F030014, LQ16F030007) and Innovation Foundation from Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education (JYB201706).

References

1. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust l1 tracker using accelerated proximal gradient approach. In: CVPR (2012)
2. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.H.: Staple: Complementary learners for real-time tracking. In: CVPR (2016)
3. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: ECCV (2016)
4. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: CVPR (2010)
5. Chen, K., Tao, W.: Once for all: a two-flow convolutional neural network for visual tracking. TCSVT **PP**(99), 1–1 (2017)
6. Danelljan, M., Häger, G., Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: BMVC (2014)
7. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Discriminative scale space tracking. PAMI **39**(8), 1561–1575 (2017)
8. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Convolutional features for correlation filter based visual tracking. In: ICCV Workshops (2015)
9. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: CVPR (2015)
10. Danelljan, M., Robinson, A., Khan, F.S., Felsberg, M.: Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: ECCV (2016)
11. Danelljan, M., Shahbaz Khan, F., Felsberg, M., Van de Weijer, J.: Adaptive color attributes for real-time visual tracking. In: CVPR (2014)
12. Gao, J., Ling, H., Hu, W., Xing, J.: Transfer learning based visual tracking with gaussian processes regression. In: ECCV (2014)
13. Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.M., Hicks, S.L., Torr, P.H.: Struck: Structured output tracking with kernels. PAMI **38**(10), 2096–2109 (2016)
14. Held, D., Thrun, S., Savarese, S.: Learning to track at 100 fps with deep regression networks. In: ECCV (2016)
15. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: ECCV (2012)
16. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. PAMI **37**(3), 583–596 (2015)
17. Huang, D., Luo, L., Wen, M., Chen, Z., Zhang, C.: Enable scale and aspect ratio adaptability in visual tracking with detection proposals. In: BMVC (2015)
18. Leal-Taixé, L., Canton-Ferrer, C., Schindler, K.: Learning by tracking: Siamese cnn for robust target association. In: CVPR Workshops (2016)
19. Li, Y., Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. In: ECCV Workshops (2014)
20. Liang, P., Blasch, E., Ling, H.: Encoding color information for visual tracking: Algorithms and benchmark. TIP **24**(12), 5630–5644 (2015)
21. Liu, B., Huang, J., Kulikowski, C., Yang, L.: Robust visual tracking using local sparse appearance model and k-selection. PAMI **35**(12), 2968–2981 (2013)
22. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical convolutional features for visual tracking. In: ICCV (2015)
23. Mueller, M., Smith, N., Ghanem, B.: A benchmark and simulator for uav tracking. In: ECCV (2016)
24. Possegger, H., Mauthner, T., Bischof, H.: In defense of color-based model-free tracking. In: CVPR (2015)

25. Sevilla-Lara, L., Learned-Miller, E.: Distribution fields for tracking. In: CVPR (2012)
26. Tao, R., Gavves, E., Smeulders, A.W.: Siamese instance search for tracking. In: CVPR (2016)
27. Valmadre, J., Bertinetto, L., Henriques, J.F., Vedaldi, A., Torr, P.H.: End-to-end representation learning for correlation filter based tracking. In: CVPR (2017)
28. Wang, N., Yeung, D.Y.: Ensemble-based tracking: Aggregating crowdsourced structured time series data. In: ICML (2014)
29. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: CVPR (2013)
30. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. PAMI **37**(9), 1834–1848 (2015)
31. Zhang, J., Ma, S., Sclaroff, S.: Meem: robust tracking via multiple experts using entropy minimization. In: ECCV (2014)
32. Zhang, K., Zhang, L., Yang, M.H.: Real-time compressive tracking. In: ECCV (2012)