Robust Visual Tracking via Collaborative and Reinforced Convolutional Feature Learning

Dongdong Li[1], Yangliu Kuai[1], Gongjian Wen[1], and Li Liu[2,3]
[1]College of Electronic Science and Technology, National University of Defense Technology, China
[2]College of System Engineering, National University of Defense Technology, China
[3]Center for Machine Vision and Signal Analysis, University of Oulu, Finland
moqimubai@sina.cn,kuaiyangliunudt@163.com,wengongjian@sina.com, dreamliu2010@gmail.com

Abstract

Convolutional neural networks are potent models that yield hierarchies of features and have drawn increasing interest in the visual tracking field. In the paper, we design an end-to-end trainable tracking framework based on Siamese network, which proposes to learn the low-level fine-grained and high-level semantic representations simultaneously with the aim of mutual benefit. Due to the distinct and complementary characteristics of the feature hierarchies, different tracking mechanisms are adopted for different feature layers. The low-level features are exploited and updated with a correlation filter layer for adaptive tracking and the high-level features are compared through cross-correlation directly for robust tracking. The two-level features are jointly trained with a multi-task loss function end-to-end. The proposed tracker takes full advantage of the adaptability of the low-level features and the generalization ability of the high-level features. Extensive experimental tracking results on the widely used OTB and TC128 benchmarks demonstrate the superiority of our tracker. Meanwhile, our proposed tracker can achieve a real-time tracking speed.

1. Introduction

Visual object tracking is an established yet rapidly evolving research area in computer vision. In general, it aims to estimate the spatial trajectory of a target object in an image sequence, given its initial state, i.e., location and underlying area. It provides a fundamental component for high-level visual understanding problems such as motion analysis, event detection, situational awareness, and activity recognition. Despite significant process in recent years, finding the corresponding object regions across multiple frames is still a challenging problem due to factors such as occlusion, deformation, illumination change, fast motion, and background clutter. In this paper, we only focus on single camera, single-target, short-term and model-free tracking and refer the interested readers to [1] and [2] for a thorough review of the existing tracking algorithms.

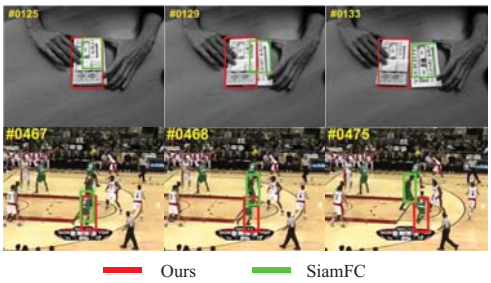

Figure 1: Tracking snapshots of SiamFC and our tracker in the presence of similar distractors on the *Coupon* and *Basketball* sequences [3]. The SiamFC tracker is based on high-level semantic embeddings and drifts to surrounding distractors in the given sequences while our tracker performs well.

Accuracy has always been the pursuit of all tracking algorithms. In recent years, Convolutional Neural Networks (CNNs) have been pervasively adopted in various computer vision tasks such as image classification [4], object detection [5] and semantic segmentation [6], due to the excellent performance in representing visual data. However, the scarcity of training data and real-time demand greatly limit the applications of CNNs in the tracking field. Some researchers propose to utilize a pre-trained CNN that was learned for a different but related task to extract deep features in the existing "shallow" tracking methods [7, 8, 9]. However, this kind of practice does not take advantage of the benefits of end-to-end training. Later, end-to-end methods are proposed to fine-tune the last or last few layers of the pre-trained network [10] but at the cost of tracking speed. Given this situation, the Siamese network

based trackers are introduced [11, 12, 13, 14]. Generally, these trackers learn embedded semantic features for classification with massive training data, i.e., ILSVRC2015 video detection dataset [4] offline, and then estimate the target position through one network forward-propagation online. As the embedded features mainly capture category-aware semantics and are robust to significant and dramatic appearance variations, these trackers perform well in distinguishing targets of different categories. However, an issue ensues with such an approach. The semantic feature representation neglects the low-level fine-grained details and blurs the intra-class difference. In specific, the trackers can easily distinguish a dog from a cat, but cannot differentiate person in red clothes with the other in blue. When faced with distractors with the same category, the tracker may drift as shown in Figure 1. As pointed in [15], the different layers of a CNN provide multiple levels and different perspective characteristics of a target in a feature hierarchy. The earlier feature layers mainly provide fine-grained features, which is beneficial to separate target from similar distractors and retains high spatial resolution for precise localization. Considering that the low-level features have been computed through the network forward propagation, a naïve solution to the issue mentioned above is to combine the feature representations from different layers directly. In our previous work, we adopt a skip-layer connection to constitute hyper-feature representations of the target [14], however, a limited improvement is achieved on the tracking performance. We attribute this to the fixed feature representations during tracking. When the high-level feature is adopted to discriminate the target from the background, fixing the representation does not affect the tracking performance much, as verified in [11]. The authors found that updating the feature representation of the exemplar online through linear interpolation does not gain much performance and thus they keep it fixed. However, the low-level representation mainly focuses on fine-grained spatial details and needs to be updated to adapt to the video-specific appearance variations of the target. Correlation filter is an efficient online learning method and has been integrated into CNN as a differential layer in the previous work [16, 17]. Through end-to-end training, the feature can be updated easily and tightly coupled to the correlation filter.

From the above two paragraphs, we have the following observations. First, Siamese network based trackers are few real-time trackers that perform end-to-end tracking based on CNNs, which is a crucial factor that we build our work on them. Second, the low-level fine-grained and high-level semantic representations from CNNs provide complementary characteristics of the target and can function jointly to reinforce the representation. Finally, the low-level features need to be updated to capture the video-specific variation of the target and this can be achieved through embedding a correlation layer in the network. In the paper, we design an end-to-end multi-task learning based tracker on the basis of the Siamese network, which learns two tasks simultaneously with the aim of mutual benefit. The low-level features are exploited and updated with a correlation filter layer for precise tracking; the high-level features are utilized through cross-correlation for robust tracking. The main contributions of our work are three-fold:

(1) We design an end-to-end tracking framework based on Siamese network, which utilizes different tracking mechanisms for different feature layers according to their specific characteristics.

(2) We adopt a multi-task strategy to train the network and learn a collaborative and reinforced representation for the target. The proposed tracker takes full advantage of the adaptability of the low-level features and the generalization ability of the high-level features.

(3) Extensive experimental results on the widely used tracking benchmarks demonstrate that our method can achieve state-of-the-art tracking performance and real-time tracking speed.

2. Related Work

There have been many advances in the object tracking literature in the recent years. Due to space limitations, here we focus on those that are most relevant to our work.

2.1. Correlation Filter based Tracking

Correlation filters have attracted considerable attention in the tracking field due to the fair robustness and extreme efficiency. Bolme et al. pioneer the work with a minimum output sum of squared error correlation filter [18]. Henriques et al. introduce a kernelized correlation filter [19] and extend the feature representation to multi-channel. Later, based on the standard DCF formulation, different variants of correlation filters have been proposed to boost tracking performance using scale estimation [20, 21], boundary effect alleviation [22, 23, 24], context learning [25], complementary cues [26], target adaptation and feature integration [27, 28]. Recently, driven by the popular trend of CNNs in other fields, researchers in the tracking community have started to combine DCFs with CNNs. The conventional approach is to integrate CNN features to the DCF framework. DeepSRDCF substitutes hand-crafted features with shallow CNN features in a spatially regularized DCF framework and achieves superior tracking performance [29]. CF2 employs CNN features extracted from multiple convolutional layers to encode both spatial details and high-level semantics [7]. Despite significant performance improvements, these methods extract CNN features from a pre-trained classification network and the feature extraction process is separate

from the filter training, the tracking results may be suboptimal. Thus CFnet [16] proposed by Jack Valmadre et al. and DCFNet [17] put forward by QiangWang et al. interpret the correlation filter as a differentiable layer in a deep neural network and train the network end-to-end to find the features most suitable for the correlation filter. Inspired by these trackers, in the paper we incorporate the correlation filter as a differential layer in our network to update the low-level features and further implement precise target location.

2.2. CNN based Tracking

Simply regarding CNNs as a feature extractor does not take full advantage of the benefits of CNNs. To fully exploit the representation power of CNNs in visual tracking, it is desirable to train them on large-scale dataset specialized for visual tracking. MDNet [10] trains a multi-domain lightweight network offline with massive data and perform SGD to fine-tune the last few layers of the network during online tracking. MDNet achieves state-of-the-art results but fails to operate in real-time. As the essence of tracking is to find the region in a search image most similar to the given target bounding box, the Siamese architecture has been exploited in the tracking field and shows impressive performance. Held et al. introduce GOTURN [12], in which the motion between successive frames is predicted using a deep regression network. Tao et al. propose to train a Siamese network to identify candidate image locations that match the initial object appearance and term their method as Siamese Instance search Tracker [13]. Bertinetto et al. [11] put forward a novel fully-convolutional Siamese network (SiamFC) to measure the similarity between two images and locate the target in the current frame. The fully-convolutional architecture enables dense and efficient sliding-window evaluations with a bilinear layer and makes the tracker real-time. However, these Siamese network based methods share a common problem that the feature representations are based on the high-level semantic layers and no fine-tuning is performed, which may drift in the presence of same-category distractors. Based on SiamFC, Y. Kuai et al. propose to combine different layers of the network in SiamFC to constitute a more abundant representation of the target [14], but a limited performance improvement is achieved due to the absence of model update in the low-level features. In the paper, we propose to combine the low-level and high-level features to reinforce the feature representations of the target, and according to their specific characteristics, we select different tracking mechanisms for different feature layers.

2.3. Collaborative Tracking

Due to the different and complementary characteristics of different features and trackers, methods based on ensemble deep features and trackers are proposed. In terms of deep feature ensemble, J. Li et al. [15] build a general network (GNet) and a specific network (SNet) on top of the conv5 and conv4 feature layers respectively. The two networks are used interchangeably based on distractor detection scheme. Chao et al. [7] learn the correlation filters from the features in the third, fourth and fifth convolutional layers successively and then determine the target location according to the maximum response of each filter comprehensively. Feature ensemble helps complement different representations, similarly, the tracker ensemble is proposed to combine the advantages of different trackers. PTAV [30] proposed by H. Fan et al. combines short-term correlation filter based tracking with long-term re-detections based on Siamese network with a switching mechanism. Instead of adaptive selection, Y. Kuai et al. utilize correlation filter trackers to refine the tracking results based on Siamese network [31]. Our tracker proposed in this paper is a combination of both deep features (low-level and high-level) and trackers (correlation filter based deep tracker and Siamese network based tracker).

3. Reinforced Convolutional Feature Learning for Visual Tracking

In the paper, we design an end-to-end multi-task learning based tracker on the basis of Siamese network, which learns two tasks simultaneously with the aim of mutual benefit. In this section, the overall network architecture of the proposed tracker is given at first. Then we introduce two primary components of our tracker, namely the high-level robust semantic tracking and the low-level fine-grained adaptive tracking. At last, the collaborative training and tracking procedures are described.

3.1. Overall Network Architecture

The overall network architecture is shown in Figure 2. The main part of our network architecture resembles the two-branch network used in SiamFC, and each branch is composed of five convolutional layers and two max pooling layers. A correlation filter layer is imposed on the low-level features (after *pool2* layer) for fine-grained target location and efficient model update, and the high-level semantic features (after *conv5* layer) of exemplar and search branches are compared directly through cross-correlation for robust tracking. Although the feature after *pool1* layer is more fine-grained and beneficial for precise target location, adopting such feature brings much burden to the tracking efficiency. To make our tracker real-time, we select the feature after *pool2* layer to balance the tracking efficiency and accuracy. The detailed dimensions of network parameters and output activations in different layers are given in Table 1. As the search image shares the same size with the exemplar image, here we only provide the parameter dimensions in the exemplar branch.

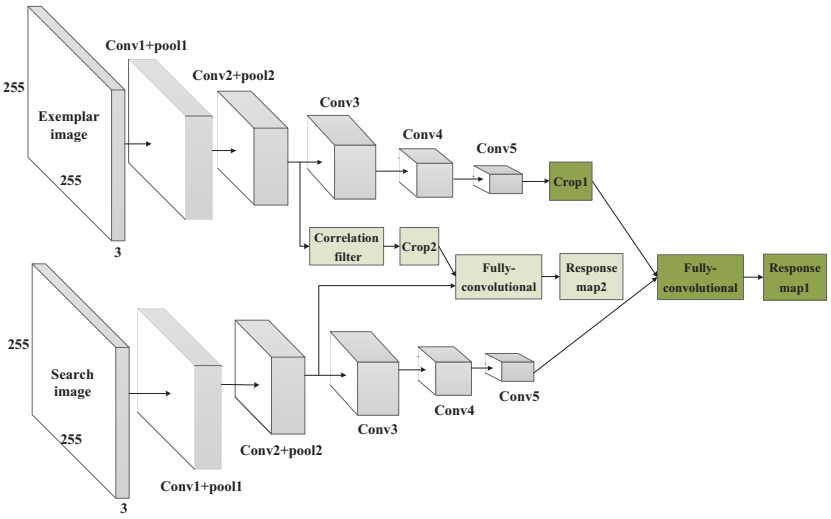

Figure 2: The network architecture of our proposed tracker. The main part of the network is a two-branch Siamese network. And to comprehensively utilize the features from different CNN layers, a correlation filter layer is imposed on the low-level feature (after *pool2* layer) for fine-grained target location and efficient model update, and the high-level semantic feature (after *conv5* layer) is compared directly through cross-correlation for robust tracking.

Table 1: Dimensions of network parameters and output activations in the designed network. In the table, KS is the kernel size of the filters. IC(input channel) represents the dimensions and number of filters used. EAS stands for activation size for exemplar image. OC(output channel) means the dimension of each output activation.

Layer	KS	IC	Stride	EAS	OC
Input				255×255	3
Conv1	11×11	3×16	2	123×123	16
Pool1	5×5		2	61×61	16
Conv2	5×5	16×32	1	57×57	32
Pool2	3×3		1	55×55	32
Conv3	3×3	32×64	1	53×53	64
Conv4	3×3	64×128	1	51×51	128
Conv5	3×3	128×32	1	49×49	32
CF				55×55	32
Crop1				23×23	32
Crop2				17×17	32

3.2. Generic Semantic Learning for Robust Tracking

For each pair of exemplar and search images (z, x), the network applies the embedding transformation f to them and generates the feature representations $f(z)$, $f(x)$ after *conv5* layer. As the high-level feature representations mainly focus on object category and are robust to target appearance variations, we compare the similarity between $f(z)$ and $f(x)$ directly through cross-correlation g and obtain the response map D_1. The computed value is denoted as $v_1 = g(f(z), f(x))$. The pixels y_1 on the response map is labeled {+1,-1} according to their distances to the center. And we adopt the logistic loss function to measure the difference between the computed value v_1 and labeled ground truth value y_1 on the response map, as shown in Equation 1.

$$L_{high} = \frac{1}{|D_1|} \sum_{u \in D_1} log(1 + exp(-y_1[u] \cdot v_1[u])) \quad (1)$$

Where D_1 represents the response map and $|D_1|$ denotes the number of pixels on the response map. The generic semantic learning is optimized by minimizing the above loss function with SGD.

3.3. Correlation Filter Learning for Adaptive Tracking

Different from the robust high-level feature representations, the low-level features mainly capture fine-grained details, such as edge, texture, contour, and need updating to adapt to variations of the target. A correlation filter layer is imposed on the low-level features to achieve this. We select the feature after *pool2* layer to balance the tracking efficiency and accuracy.

Discriminate correlation filter formulation: Given a scalar-valued image x and the corresponding Gaussian label y, the correlation filter template w can be obtained by

regressing all the circular shifted version of x to the label y. In specific, w is solved using the following equation:

$$arg\min_{w} \frac{1}{2n}\|w * x - y\|^2 + \frac{\lambda}{2}\|w\|^2 \tag{2}$$

Where n is the effective number of samples, $*$ denotes circular cross-correlation. By the use of the Lagrange multiplier method and the property of circulant matrix in the Fourier domain, the solution of Equation 2 is illustrated in Equation 3.

$$\begin{cases} \hat{k} = \frac{1}{n}(\hat{x}^* \cdot \hat{x}) + \lambda \mathbb{1} \\ \hat{\alpha} = \frac{1}{n}\hat{k}^{-1} \cdot \hat{y} \\ \hat{w} = \hat{\alpha}^* \cdot \hat{x} \end{cases} \tag{3}$$

Here, $\hat{x}$ represents the Fourier transform of the variable x, $\hat{x}^*$ represents the complex conjugation of $\hat{x}$ and $\mathbb{1}$ is a signal of ones. The product and division in Equation 3 are point-wise operations.

Back-propagation: Given the image sample and the corresponding label, we can figure out the coefficients of correlation filter and accomplish the network forward propagation. And to integrate the correlation filter as a differential layer in the network, we also need to derive the back-propagation equation. Given the output scalar loss l and the derivative of l on w, namely $\nabla_w l$, the derivation propagation from $\nabla_w l$ to $\nabla_x l$ and $\nabla_y l$ can be solved as Equation 4. The detailed derivation procedures of Equation 3 and Equation 4 can be found in [16].

$$\begin{cases} \widehat{\nabla_\alpha l} = \hat{x} \cdot (\widehat{\nabla_w l})^* \\ \widehat{\nabla_y l} = \frac{1}{n}\hat{k}^{-*} \cdot \widehat{\nabla_\alpha l} \\ \widehat{\nabla_k l} = -\hat{k}^{-*} \cdot \hat{\alpha}^* \cdot \widehat{\nabla_\alpha l} \\ \widehat{\nabla_x l} = \hat{\alpha} \cdot \widehat{\nabla_w l} + \frac{2}{n}\hat{x} \cdot Re\left\{\widehat{\nabla_k l}\right\} \end{cases} \tag{4}$$

After derivations of the forward and backward propagation of the correlation filter layer, we construct the loss function L_{low} of the low-level feature representations as shown in Equation 5, which is similar to the high-level feature branch.

$$L_{low} = \frac{1}{|D_2|}\sum_{u \in D_2} log(1 + exp(-y_2[u] \cdot v_2[u])) \tag{5}$$

Where D_2 represents the response map for low-level features, $|D_2|$ denotes the number of pixels on the response map, y_2 and v_2 respectively denotes the ground truth and computed value on the response map.

3.4. Collaborative Training and Tracking

In the training stage, to collaboratively learn the fine-grained and robust semantic features, we propose to train the network with a multi-task loss function in a unified way. The detailed loss function is as follows:

$$L = L_{high} + L_{low} + P(\theta) \tag{6}$$

Where $P(\theta)$ is a l_2-norm punishment item of the network parameters θ for better regularization. The network parameter θ is obtained by minimizing the multi-task loss above with SGD.

In the tracking stage, for frame t, the search image area x_s is cropped around the target location in frame $t-1$ with multiple scale variations s. Through network forward propagation, we acquire response maps for the low-level and high-level branches, respectively expressed by q and h. The two response maps are linearly added. And the target location is estimated by finding the maximum on the fused response map as shown in Equation 7:

$$arg\max_{u,v,s} q_{u,v}(x_s) + h_{u,v}(x_s) \tag{7}$$

And in order to adapt to the target appearance variations during tracking, the correlation filter coefficients w in the low-level branch is updated via a simple linear interpolation as follows:

$$w_t = \alpha w + (1-\alpha)w_{t-1} \tag{8}$$

Where α is the linear weighting coefficient, w_t and w_{t-1} respectively denotes the correlation filter coefficients in frame t and $t-1$, w is the computed coefficients in the current frame. On the contrary, the semantic representation in the high-level branch is fixed after being computed in the first frame, which avoids the background contamination caused by tracking drift and reserves the original accurate target information. Through combinations of per-frame updating in the low-level branch and the fixed accurate representation in the high-level branch, our tracker captures the fixed and variant information of the target simultaneously and performs well.

4. Experiments

4.1. Implementation Details

Data and parameters: Our algorithm is implemented in Matlab using MatConvNet toolbox [32]. The initial network parameters follow a Gaussian distribution, which is further updated by minimizing Equation 2. The total training number is set to be 50, each consisting of 53200 pairs of images from the ILSVRC2015 object detection dataset. The weight decay is 10^{-3} and learning rate is annealed geometrically at each epoch from 10^{-2} to 10^{-5}. To handle scale variation, the tracking target is searched over three scales $1.025^{\{-1,0,1\}}$. The linear weighting coefficient α in Equation 8 is set to be 0.41, which follows the default parameter setting in CFnet [16]. All the parameters are fixed for all experiments. And all the trackers are run on a machine equipped with a single NVIDIA GeForce GTX Titan GPU and an Intel Core i7-6700K at 4.0GHz.

Evaluation methodology: On OTB and TC128 benchmarks, the evaluation is based on two metrics: precision

plot and success plot in a one-pass evaluation. The precision plot is computed as the percentage of frames in the sequences where Euclidean distance between the ground-truth and the estimated target position is smaller than a certain threshold. The success plot is plotted over the range of intersection over union (IOU) thresholds on all videos. We use the distance precision (DP) rate at 20 pixels to rank trackers in the precision plot and the area under curve, also called overlap success (OS) rate to rank trackers in the success plot. Notably, OS rate is used as the primary metric for ranking methods.

4.2. Experiments on the OTB Dataset

OTB50 [33] is a popular tracking dataset containing 50 fully annotated videos. OTB100 [3] is the extension of OTB50 and contains 100 sequences. Compared to OTB50, 50 more challenging sequences are included in OTB100. In this subsection, we first conduct an ablation study between our tracker and two baseline trackers, and then an experiment is performed to compare our tracker with other state-of-the-art trackers.

Ablation study: To demonstrate the effectiveness of the reinforced representation learning proposed in our tracker, we utilize the success plot to compare our method with two baseline trackers (SiamFC [11] and CFnet [16]) on the OTB50 and OTB100 datasets respectively. SiamFC utilizes the high-level semantic embeddings while CFnet is based on the low-level fine-grained target localization. The comparison result is shown in Figure 3. And it can be seen that our tracker respectively achieves an absolute gain of 1.8% and 1.9% in the OS rate on the OTB50 and OTB100 datasets when compared to SiamFC. And the performance improvements are respectively 2.7% and 1.3% when compared to CFnet. Therefore, fusing the high-level semantic and low-level fine-grained representations is more effective for tracking than using the two components independently.

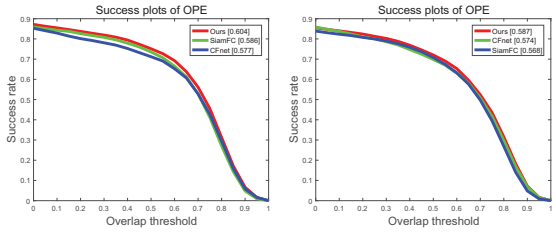

Figure 3: Success plot of our proposed tracker and two baseline trackers on the OTB50 (left) and OTB100 (right) benchmarks.

Comparison with state-of-the-art trackers: To further validate the effectiveness of our proposed algorithm, an experiment is carried out to compare our algorithm with seven other state-of-the-art trackers on the OTB100 dataset. These trackers cover the two mainstream methods in the tracking field, namely deep learning trackers (DCFNet [17], SiamFC [11] and CFnet [16]) and correlation filter trackers (DSST [21], KCF [19], LCT [34] and SAMF [20]. Figure 4 shows the results of all the trackers in comparison mentioned above. We also present the quantitative results of OS rate at 0.5, DP rate at 20 pixels and tracking speed in fps (frames per second) in Table 2. In general, our proposed method performs favorably against the trackers in comparison in both OS rate and DP rate. And apart from the superior performance, our tracker also achieves real-time tracking speed.

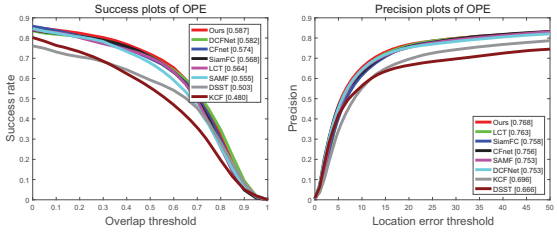

Figure 4: Success plot (left) and Precision plot (right) of our proposed tracker and seven other state-of-the-art trackers in comparison on the OTB100 benchmark.

Attribute-based evaluation: For detailed performance analysis, we provide an attribute-based analysis on OTB100 dataset. The sequences in the dataset are annotated with 11 different attributes: illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutter and low-resolution. Figure 5 shows OS plots under six different attributes. From the curves, we have the following observations. First, our tracker generally performs better than the correlation filter based trackers, such as KCF, DSST, LCT and SAMF, due to the advantage of CNN features over the hand-crafted features. Second, our tracker combines the merits of SiamFC and CFnet and performs well in the majority of sequences with challenging attributes.

Qualitative Analysis: Figure 6 intuitively illustrates the tracking results of our proposed algorithm and its baseline trackers (SiamFC [11] and CFnet [16]) on seven challenging sequences from OTB100 dataset. These selected sequences cover many difficulties faced in the visual tracking task, such as heavy occlusions, fast motion, severe deformation, similar distractor, rotations, illumination variation and et,al. SiamFC utilizes the robust convolutional features and performs well in sequences with occlusion (*jogging1*), fast motion (*jumping*) and deformation (*sylvester, skiing*), but fails when similar distractors occur around (*walking2,skating2*) due to the semantic feature representation and absence of model updating. CFnet is based on

Table 2: Quantitative comparison results with state-of-the-art trackers on the OTB100 benchmark in terms of OS rate at 0.5, DP rate at 20 pixels and tracking speed in fps.

Trackers	Ours	SiamFC[11]	CFnet[16]	DCFNet[17]
OS rate (%)	72.1	70.8	69.8	71.0
DP rate (%)	76.8	75.8	75.6	75.3
Speed (fps)	25	47	43	41
Trackers	DSST[21]	KCF[19]	LCT[34]	SAMF[20]
OS rate (%)	59.2	55.6	70.4	67.7
DP rate (%)	66.6	69.6	76.3	75.3
Speed (fps)	61	413	20	17

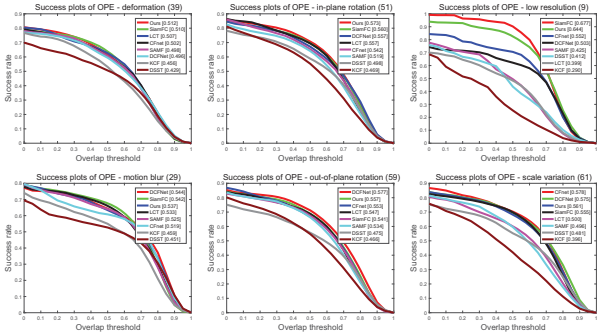

Figure 5: The success plots of trackers in comparison under six challenging attributes: deformation, in-plane rotation, low resolution, motion blur, out-of-plane rotation and scale variation.

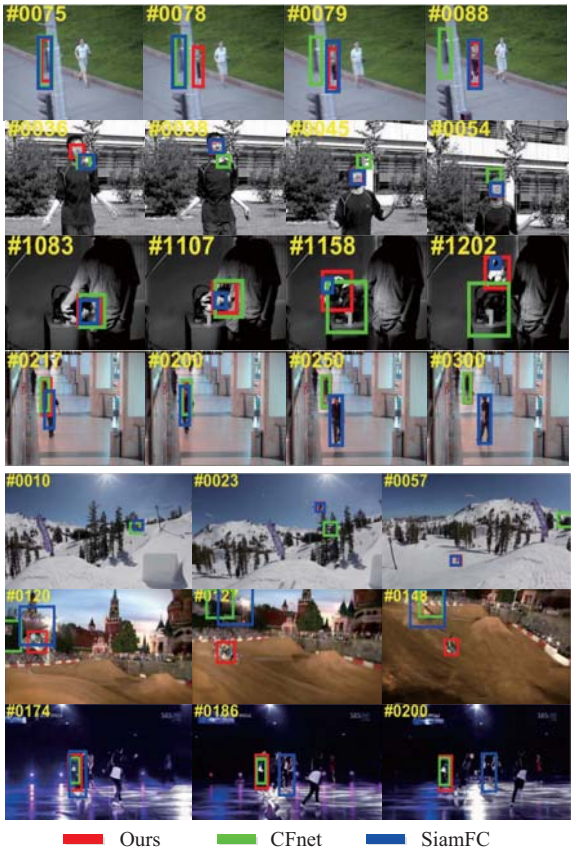

Figure 6: Qualitative snapshots of proposed tracker, SiamFC [11] and CFnet [16] on seven challenging sequences from OTB100 [3] (from top to down are *jogging1*, *jumping*, *sylvester*, *walking2*, *skiing*, *motorRolling* and*skating1*).

the low-level features and updated with a correlation filter layer. It performs well in sequences with similar distractions (*walking2,skating2*), but drifts when targets undergo heavy occlusion (*jogging1*), deformation (*sylvester, skiing*) and fast motion (*jumping*), as a result of the low-level feature representation and the boundary effect. Due to the complementary characteristics of SiamFC and CFnet, our proposed method proposes to learn collaborative feature representations simultaneously with a multi-task strategy. The proposed algorithm combines the advantages of the adaptivity of CFnet and robustness of SiamFC and tracks the target accurately over all the sequences, including sequences difficult for both trackers (*motorRolling*).

4.3. Experiments on the TC128 Benchmark

The TC128 dataset [35] contains 128 color sequences and is specially designed to evaluate the tracking performance in color sequences. We perform comparative experiments between our proposed tracker with the existing top-ranking trackers on the dataset. As shown in Figure 7, our proposed tracker achieves an OS rate of 51.33% and a DP rate of 69.75%, which perform better than the majority trackers. In specific, our tracker performs much better than CFnet in both the OS and DP rate, and has similar performance with SiamFC.

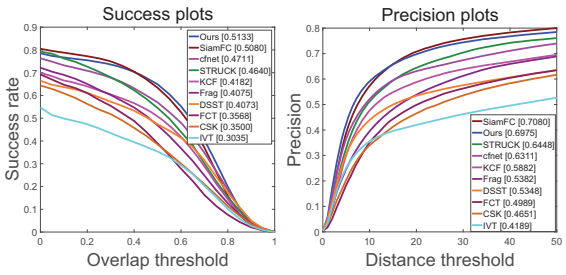

Figure 7: Success plot (left) and Precision plot (right) of our proposed tracker and seven other state-of-the-art trackers in comparison on the OTB100 benchmark..

5. Conclusion

In this paper, we propose an end-to-end tracking framework to comprehensively utilize the advantages of feature representations from different CNN layers (adaptability and generalization). A correlation filter layer is imposed on the low-level features to implement model update and adaptive tracking. The high-level features are cross-correlated directly for robust tracking. And the two complementary components are jointly trained through a multi-task loss strategy to learn a reinforced feature representation. Experimental results on the widely used tracking benchmarks OTB and TC128 demonstrate performance improvements and limited burden on the tracking efficiency when compared to the existing Siamese network based trackers.

References

[1] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys.*, 38:1–45, 2006.

[2] A. Smeulders, D. Chu, R. Cucchiara, and S. Calderara. Visual tracking: an experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 36:1442–1468, 2014.

[3] Y. Wu and J. Lim. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. IEEE Conference on Neural Information Processing Systems, pages 1097–1105, 2012.

[5] L. Wang, L. Wang, H. Lu, and et al. Saliency detection with recurrent fully convolutional networks. European Conference on Computer Vision, pages 825–841, 2016.

[6] J. Long, E. Shelhamer, and Darrell T. Fully convolutional networks for semantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition, pages 3431–3440, 2015.

[7] C. Ma, J.B. Huang, X. Yang, and M. H. Yang. Hierarchical convolutional features for visual tracking. IEEE Conference on Computer Vision, pages 3074–3082, 2015.

[8] M. Danelljan, G. Hager, F. Khan, and M. Felsberg. Convolutional features for correlation filter based visual tracking. IEEE Conference on Computer Vision Workshop, pages 621–629, 2015.

[9] N. Wang, S. Li, A. Gupta, and D.Y. Yeung. Transferring rich feature hierarchies for robust visual tracking. *Computer Science.*, 2015.

[10] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. *Computer Science.*, 2015.

[11] L. Bertinetto, J. Valmadre, J. F. Henriques, and et al. Fully-convolutional siamese networks for object tracking. IEEE Conference on Computer Vision, pages 3119–3127, 2015.

[12] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. European Conference on Computer Vision, 2015.

[13] R. Tao, E. Gavves, and A. W. M. Smeulders. Siamese instance search for tracking. IEEE Conference on Computer Vision and Pattern Recognition, pages 1420–1429, 2016.

[14] Y. Kuai, G. Wen, and D. Li. Hyper-feature based tracking with the fully-convolutional siamese network. 2017 International Conference on Digital Image Computing: Techniques and Applications, pages 1–7, 2017.

[15] L. Wang, W. Ouyang, X. Wang, and et al. Visual tracking with fully convolutional networks. IEEE Conference on Computer Vision and Pattern Recognition, pages 3119–3127, 2015.

[16] V. Jack, B. Luca, H. João F, V. Andrea, and T. Philip HS. End-to-end representation learning for correlation filter based tracking. *arXiv preprint arXiv:1704.06036*, 2017.

[17] Q. Wang, J. Gao, J. Xing, M. Zhang, and W. Hu. Dcfnet: Discriminant correlation filters network for visual tracking. *arXiv preprint arXiv:1704.04057*, 2017.

[18] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2544–2550, 2010.

[19] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.

[20] Y. Li and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. European Conference on Computer Vision, pages 254–265, 2014.

[21] M. Danelljan, G. Häger, and F. Khan. Accurate scale estimation for robust visual tracking. British Machine Vision Conference, pages 1–11, 2014.

[22] M. Danelljan, G. Häger, and F. S. Khan. Learning spatially regularized correlation filters for visual tracking. In *IEEE Conference on Computer Vision*, pages 4310–4318, 2015.

[23] H. K. Galoogahi, T. Sim, and S. Lucey. Correlation filters with limited boundaries. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 4630–4638, 2015.

[24] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. *CoRR*, abs/1703.04590, 2017.

[25] M. Mueller, N. Smith, and B. Ghanem. Context-aware correlation filter tracking. IEEE Conference on Computer Vision and Pattern Recognition, pages 1387–1395, 2017.

[26] L. Bertinetto, J. Valmadre, and S. Golodetz. Staple: Complementary learners for real-time tracking. IEEE Conference on Computer Vision and Pattern Recognition, pages 1401–1409, 2015.

[27] M. Danelljan, A. Robinson, F.K. Shahbaz, and M. Felsberg. Beyond correlation flters: Learning continuous convolution operators for visual tracking. European Conference on Computer Vision, 2016.

[28] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg. Eco: Efficient convolution operators for tracking. IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[29] M. Danelljan, G. Häger, F. S. Khan, and et al. Convolutional features for correlation filter based visual tracking. IEEE Conference on Computer Vision Workshop, pages 621–629, 2016.

[30] H. Fan and H. Ling. Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking. IEEE Conference on Computer Vision and Pattern Recognition, pages 5487–5495, 2017.

[31] Y. Kuai, G. Wen, and D. Li. When correlation filters meet fully-convolutional siamese networks for distractor-aware tracking. *Signal Processing: Image Communication*, 64:107–117, 2018.

[32] A. Vedaldi and K. Lenc. Matconvnet: Convolutional neural networks for matlab. In *Association for Computing Machinery*, pages 689–692, 2015.

[33] Y. Wu, J. Lim, and M. H. Yang. Online object tracking: a benchmark. IEEE Conference on Computer Vision and Pattern Recognition, pages 2411–2418, 2013.

[34] C. Ma, X. Yang, and C. Zhang. Long-term correlation tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5388–5396, 2015.

[35] P. Liang, E. Blasch, and H. Ling. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Transactions on Image Processing*, 24(12):5630–5644, 2015.