Multi-branch Siamese Network for High Performance Online Visual Tracking

Junfei Zhuang zjf1@bupt.edu.cn
Beijing University of Posts and Telecommunications

Yuan Dong yuandong@bupt.edu.cn
Beijing University of Posts and Telecommunications

Hongliang Bai hongliang.bai@faceall.cn
Beijing Faceall Technology Co.,Ltd

Gang Wang gang.wang@srcb.ricoh.com
Ricoh software research center (Beijing) co. Ltd

Editors: Wee Sun Lee and Taiji Suzuki

Abstract

Recently, Siamese networks have drawn great attention in the visual tracking community because of their balanced accuracy and speed. However, most existing Siamese frameworks describe the target appearance using a global pattern from the last layer, leading to high sensitivity to similar distractors, non-rigid appearance change, and partial occlusion. Addressing these issues, we propose a Multi-branch Siamese network (MSiam) for high-performance object tracking. The MSiam performs layer-wise feature aggregations and simultaneously considers the global-local patterns for more accurate target tracking. In particular, we propose a feature aggregation module (FAM) keeping the heterogeneity of the three types of features, further improving the discriminability of MSiam using both high-level semantic and low-level spatial information. To enhance the adaptability to non-rigid appearance change and partial occlusion, a multi-scale local pattern detection module (LPDM) is designed to identify discriminative regions of the target objects. By considering various combinations of the local structures, our tracker can form various types of structure patterns. Extensive evaluations on five benchmarks demonstrate that the proposed tracking algorithm performs favorably against state-of-the-art methods while running beyond real-time.

Keywords: Visual tracking, deep learning, Siamese network

1. Introduction

Visual object tracking is an essential field of computer vision with many applications, such as automated surveillance, human-computer interaction, autonomous driving, and vehicle navigation. The core task for a single object tracking is to locate an arbitrary target in constantly video sequences. It still remains challenging due to practical factors like scale variation, fast motion, occlusions, deformation, background clutter, and other variations.

In recent years, deep convolutional neural networks (CNNs) demonstrated their superior capabilities in various vision tasks. CNNs have also significantly shown the state-of-the-art performance of object tracking. Some trackers Danelljan et al. (2015); Qi et al. (2016);

?); Sun et al. (2018) integrate deep features into conventional correlation filters tracking approaches and benefit from the expressive power of CNN features. However, these trackers cannot train a deep architecture from end to end leading to insufficient data-driven utilization and low efficiency. Some trackers Nam and Han (2016); Fan and Ling (2017); Wang et al. (2016) directly use CNNs as classifiers and take full advantage of end-to-end training. With a high volume of CNN features and the online model update application, it is computationally expensive to perform online tracking.

Recently, the Siamese architectures show their great potential in single object tracking owing to its balanced accuracy and speed. Bertinetto et al. (2016a) design a fully offline convolutional network without a model update to improve the speed of the tracking process. Despite having achieved the promising result, the model drift may occur due to two reasons: (1)**Low-level spatial features are not fully explored.** Only features from the last layer, which contain more semantic information, are employed to locate the target object. Nevertheless, background distractors and the target may have similar semantic features Zhang and Vela (2015), in such case, the high-level semantic features are less discriminative in distinguishing target or background. (2)**The entire object is described in a single global pattern.** The first frame determines the template feature maps of the Siamese network, and the target in the first frame is always clear without any occlusion. Thus, the partial occlusion and non-rigid appearance change may lead to model drift if we describe the entire object in a single global pattern.

Addressing the above issues, we propose a Multi-branch Siamese network (MSiam) for improving tracking performance. Compared to previous solutions, MSiam has two advantages: (1) Leverage features from different layers in the neural networks have been proven to be beneficial for model discriminability Lin et al. (2017); Long et al. (2015). To fully explore both the high-level semantic and the low-level spatial features for the Siamese network, we design a novel feature aggregation module (FAM) to combine different level information. Instead of using features from the last single layer in Siamese architecture, FAM enables us to fuse the high-level features into low-level features, which further improves its discriminative power to deal with the complex background, resulting in better performance. (2) To adapt to non-rigid appearance change and partial occlusion circumstance, we identify object parts with discriminative patterns using a local pattern detection module (LPDM). During the off-line training phase, we train different scale local pattern detection module with multi-level supervision, separately. During the online tracking phase, we integrate different scale local pattern score maps into the final score map to locate the target. Besides, we investigate the best scale local pattern combination to improve tracking performance. To guarantee high tracking efficiency, all these learning processes are performed during the offline training stage. Extensive analyses and evaluations on the latest tracking benchmarks Wu et al. (2013, 2015); Huang et al. (2018) and challenges Kristan et al. (2015, 2017) verify the effectiveness and efficiency of the proposed model.

To summarize, the main contributions of this work are three-fold.

- We propose a novel feature aggregation module to combine the high-level and the low-level layers information for the Siamese network, leading to an enhancement in discriminability between target and distractor.

- We propose a local pattern detection module, which can identify discriminative local parts of target objects, and we investigate the best scale local pattern combination to overcome the model drift problem caused by partial occlusion and non-rigid appearance change.
- We perform the proposed algorithm on multiple benchmark datasets and demonstrate outstanding performance with real-time tracking speed.

The rest of the paper is organized as follows. We first discuss related work in Section 2. Section 3 discusses our main contribution for target representation via the feature aggregation module and the multi-scale local pattern detection module, and we also present the overview of the proposed algorithm in this section. In Section 4, we provide experimental results. Finally, we perform the summarized conclusion of this paper in Section 5.

2. Related Works

2.1. Deep learning for visual tracking.

Deep convolutional neural network (CNN) showed the great successes in classification task Krizhevsky et al. (2012), the CNN had also been introduced into visual tracking and demonstrated excellent performances Danelljan et al. (2015, 2016a, 2017); Nam and Han (2016); Fan and Ling (2017); Song et al. (2018). Danelljan et al. (2015, 2016a, 2017) combined deep CNN features with the hand-craft features in the traditional Correlation Filter (CF) tracking model, achieving remarkable gains. Nam and Han (2016) proposed a multi-domain branch architecture with online fine-tuning. The light architecture was used to learn generic feature for tracking target. Fan and Ling (2017) introduced Recurrent Neural Networks (RNN) to learn different directional features, leading to more powerful features for locating the target object.

2.2. Tracking by Siamese Network.

Siamese network based trackers Held et al. (2016); Tao et al. (2016); Bertinetto et al. (2016b); Li et al. (2018); Wang et al. (2018) consist of two branches. One is an exemplar branch for selecting target patches, the other is a template branch. The goal of Siamese trackers is locating the target object in subsequent frames using the features from the exemplar branch and template branch. Held et al. (2016) learned a regression model to predict the location using concatenated pairs of consecutive frames. Tao et al. (2016) trained a Siamese architecture to learn a metric for online target matching and formu- lated visual tracking as a verification problem. Bertinetto et al. (2016b) learned to measure the feature similarity between the template and candidates. Owing to its light structure and without the model update, SiamFC runs faster than the real-time speed at 80 frames per second. Wang et al. (2018) introduced multiple attention mechanisms into Bertinetto et al. (2016b) to produce effective deep feature learning for visual tracking. Li et al. (2018) combined Siamese network with Region proposal Network (RPN), achieving excellent performance.

Despite all these significant progress, these trackers still suffer from two problems. First, only the feature information from the last layer is employed to predict the location of the target. These features contain semantic information which is easily distributed by distractor belonging to the same category with the target. Second, these trackers employ the global pattern to describe the target but ignore part information. Thus, partial occlusion and non-rigid appearance change may lead to model drift.

2.3. Multi-level features for tracking.

The features from different layers in the neural network contain different information. The high-level feature consists of more abstract semantic cues, while the low-level layers contain more detailed spatial information Long et al. (2015). In visual tracking, Ma et al. (2015) employed features from three different layers to obtain score maps and fused these score maps into the final output score map to locate the target. Danelljan et al. (2016c) merged different level deep features with hand-craft features to enhance the robustness, and achieved the-state-of-art results on multi-benchmarks. Wang et al. (2015) developed a regression model with two-layer features to distinguish similar semantic distractors. However, these tracking methods can not develop an end to end model.

2.4. Part-based Trackers.

Nowadays, most existing trackers can barely deal with extreme deformations. Some trackers try to solve the problem by exploiting part information. Son et al. (2015) proposed an online gradient boosting decision tree to integrate individual patches into the merged patch. Liu et al. (2015) tracked objects based on parts with multiple correlation filters in real-time speed. In Yang et al. (2015) a trackable confidence function was proposed to compute and select the reliable patches, which is capable of capturing the underlying object geometry. However, these methods are hard to design, and the patches lack the cues combination between the local and global view leading to insufficient semantic information.

3. Multi-branch Siamese Network

In our observation, some tracking failures are related to similar distractors and the lack of partial detection. We propose a deep network named Multi-branch Siamese network (MSiam) for high-performance object tracking. Figure. 1 shows the pipeline of our proposed framework. In contrast to the basic framework (SiameseFC, Bertinetto et al. (2016b)), we propose the feature aggregation module (FAM) and the multi-scale local pattern detection module (LPDM) to solve the above problems separately. In the rest of this section, we will show the overview of MSiam in Section 3.1 Then, FAM and LPDM will be shown in Section 3.2 and 3.3.

3.1. Overview

In this work, we propose a Multi-branch Siamese network, which simultaneously performs discriminative pattern detection and feature integration in an end-to-end manner. Figure. 1 shows the pipeline of our tracking algorithm. Inputs of the proposed network are a 127×127 template image z within a larger 255×255 search image x. We prepare z and x by the same way with SiameseFC Bertinetto et al. (2016b). The model learns a similarity function to densely compare the all translated sub-windows within the search image in one evaluation.

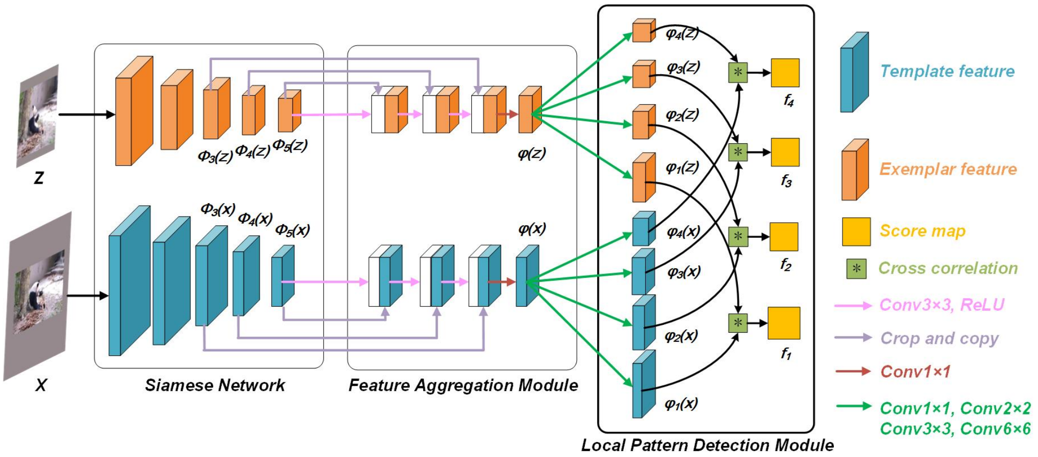

Figure 1: Illustration of the architecture of the MSiam tracking method. MSiam includes a Siamese network for feature extraction, the feature aggregation module for multi-level feature integration, and the local pattern detection module for the local pattern description. Different level feature maps are sensitive to detect different objects. We integrate three level feature maps using FAM to enhance discriminative ability between the target and background. Besides, the SiameseFC Bertinetto et al. (2016b) method exploits the holistic model for target representation and ignore detailed information. To resolve the problem, LPDM is employed to match partial feature maps between the template Z and the search region X. Best viewed in color.

To achieve this, the cross-correlation layer is proposed

$$f_i(z,x) = \varphi_i(z) * \varphi_i(x) + b_i \tag{1}$$

where φ_i is an identical transformation generated by each network stream; $b_i \in \mathbb{R}$ denotes the bias for each location; $f_i(z,x)$ represents the predicted confidence score map that highlights the 17×17 target region; and $i \in \{1,2,3,6\}$ denotes the scale of local pattern detection. To obtain $\varphi_i(z)$ and $\varphi_i(x)$, we combine the feature maps from the Siamese Network

$$\begin{aligned} \varphi(z) &= FAM(\phi_3\left(z\right), \phi_4\left(z\right), \phi_5\left(z\right)) \\ \varphi(x) &= FAM(\phi_3\left(x\right), \phi_4\left(x\right), \phi_5\left(x\right)) \end{aligned} \tag{2}$$

where FAM represents feature aggregation transformation. Then, we employ convolutional layers with different scale kernel size to tansfer $\varphi(z)$, $\varphi(x)$ to $\varphi_i(z)$ and $\varphi_i(x)$, respectively. We take the $\varphi(z)$ as an example.

$$\begin{aligned} \varphi_1(z) &= Conv1 \times 1\left(\varphi(z)\right)) \\ \varphi_2(z) &= Conv2 \times 2\left(\varphi(z)\right)) \\ \varphi_3(z) &= Conv3 \times 3\left(\varphi(z)\right)) \\ \varphi_4(z) &= Conv6 \times 6\left(\varphi(z)\right)) \end{aligned} \tag{3}$$

Each $f_i\,(z, x))$ is independently supervised by the same ground-truth label $y \in \{+1, -1\}$ as Bertinetto et al. (2016b). The final score map is the sum of the four independent score maps

$$f_{out} = \sum_{i}^{\{1,2,3,4\}} (f_i + b_i) \tag{4}$$

The two streams of the network share the same architecture and parameters, consisting of three components: Siamese network for feature extraction, the feature aggregation module for multi-level feature integration, and the local pattern detection module for the local pattern detection. The details of these components are presented in the following sections. Finally, we combine multiple level feature maps using cross-correlation and evaluate the network once on the larger search image is mathematically equivalent to combining feature maps using the inner product and evaluating the network on each translated sub-window independently.

The loss of each branch is defined as

$$L_i = \frac{1}{|D|} \sum_{u \in D} l_i(y, v) \tag{5}$$

where $L_i(y, v)$ is ith branch loss; $D \to R$ denotes the map of scores; and $l_i(y, v)$ represents the logistic loss on the one position defined as

$$L_i(y, v) = log(1 + exp(-yv)) \tag{6}$$

where v is the real-valued score of a single exemplar-candidate pair and y is its ground-truth label. The final loss is a combination of the loss from four branches

$$L = \sum_{i}^{\{1,2,3,4\}} \lambda \times L_i \tag{7}$$

where λ_i the weight parameter is a constant value equal to 0.25 in our algorithm.

3.2. Feature Aggregation Module

Different layers encode different types of features that are sensitive to different objects. As shown in Figure. 2, the $Conv3$ has a high response on the face, while it has a low response on the dog and head. Besides, the $Conv4$ is sensitive to the head, and the $Conv5$ is sensitive to the dog. To effectively leverage multi-level features, we introduce Feature Aggregation Module (FAM) to fuse information from different layers, so that our model is capable of sharing low-level features with high-level features to improve the discriminability.

The FAM is illustrated in Figure. 1. We gradually integrate features from different layers. The aggregation strategy is enlightened by Ronneberger et al. (2015), the FAM consists of four convolutional layers. Three of them are employed to integrate features from consecutive layers, and the kernel sizes of these layers are 3 × 3 with padding 1, each of the three convolutional layers is followed by a ReLU layer. To improve the efficiency of our method, the other convolutional layer with the kernel sizes 1 × 1 and, no padding is used to compress the number of channels.

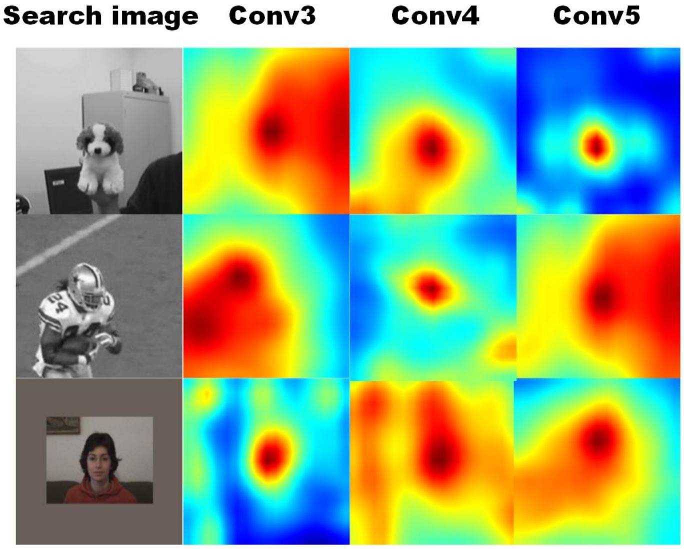

Figure 2: Response maps from different layers. The first column contains three corresponding search regions from OTB-2015 Wu et al. (2015) dataset. The second, third, and fourth column are response maps from the *Conv*3, *Conv*4, and *Conv*5 layer, respectively.

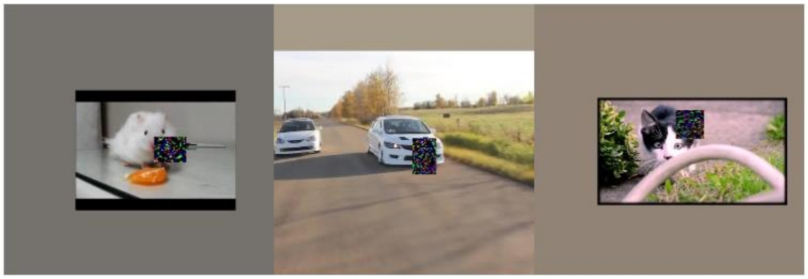

Figure 3: Random occlusion for data augmentation.

For a brief description, we define the last, second-to-last and third-to-last feature maps from Siamese Network as $\phi_5(x)$, $\phi_4(x)$, and $\phi_3(x)$, respectively. The typical workflow of feature aggregation is described as follows. Firstly, we transfer $\phi_5(x)$ with convolutional layer and concatenate the transferred features with $\phi_5(x)$; Then, we transfer the concatenated features and concatenate these features with the center cropped $\phi_4(x)$; Repeat this process for $\phi_3(x)$; Finally, the output feature maps are thrown into 1×1 convolutional layers to obtain the final aggregated features. The final output feature maps has the same size as $\phi_5(x)$.

3.3. Multi-scale Local Pattern Detection Module

Informative local patterns are crucial cues to characterize target appearance. We design the multi-scale local pattern detection module to identify discriminative patterns through end-to-end training. The LPDM contains four scale convolutional layers with a kernel size of 1 × 1, 2 × 2, 3 × 3, and 6 × 6, respectively.

Each of four scale convolutional layers corresponds to a specific local pattern. For instance, the size of $\varphi(z)$ is 6 × 6 and 6 × 6 kernel size convolutional layer is applied to detect the target from the global view, 1 × 1 kernel size convolutional layer is applied to identify the target from the pixel view. It is only a pixel in feature maps, but it is a part of the object in the template image due to the receptive field. Thus, we accomplish local pattern detection using convolutional layers with different kernel scales. The reason why we choose these four scales is our experimental results that are shown in Table. 2. The LPDM can better focus on local regions of the target and preserve more detailed information. This design is also consistent with recent findings Danelljan et al. (2016c) in visual tracking that detailed low-level features are more discriminative and suitable for target matching.

One of the primary purposes of LPDM is to improve the adaptive ability for partial occlusions. To further achieve this purpose, the random occlusion is used for data augmentation. The area of occlusion is a constant value 40 with arbitrary height and width, as shown in Figure. 3. In consideration of occlusive frames are only a tiny proportion in videos, we add on five percent occlusion frames for all training videos.

4. Experiments

To validate the proposed approach, we conducted experiments on the popular Object Tracking Benchmark 2013 (OTB-2013) Wu et al. (2013), Object Tracking Benchmark 2015 (OTB-2015) Wu et al. (2015), Visual Object Tracking 2015 (VOT2015) Kristan et al. (2015), Visual Object Tracking 2017 (VOT2017) Kristan et al. (2017) and GOT-10K Huang et al. (2018), compared with state-of-the-art trackers, and analyzed performance of our tracker by ablation studies. All benchmark details can be found from the corresponding reference Wu et al. (2013, 2015); Kristan et al. (2015, 2017); Huang et al. (2018), respectively.

4.1. Implementation Details

MSiam is implemented using PyTorch on a PC with an Intel(R) Xeon(R) 2.60GHz CPU and a single Nvidia GTX1080Ti with 12GB memory. To avoid over-fitting, our MSiam is trained on the video object detection dataset of ImageNet Large Scale Visual Recognition Challenge (ILSVRC15) Russakovsky et al. (2015). The backbone Siamese Network adopts the modified AlexNet Bertinetto et al. (2016b). The parameters of all convolution layers are randomly generated. We apply stochastic gradient descent (SGD) with the momentum of 0.9 to train the network, and the weight decay is set to 0.0005. The learning rate exponentially decays from 10^{-2} to 10^{-5}. The model is trained for 50 epochs with a mini-batch size of 32. To adapt to the scale variations, we search for the object over three scales $1.025^{\{-1,0,1\}}$.

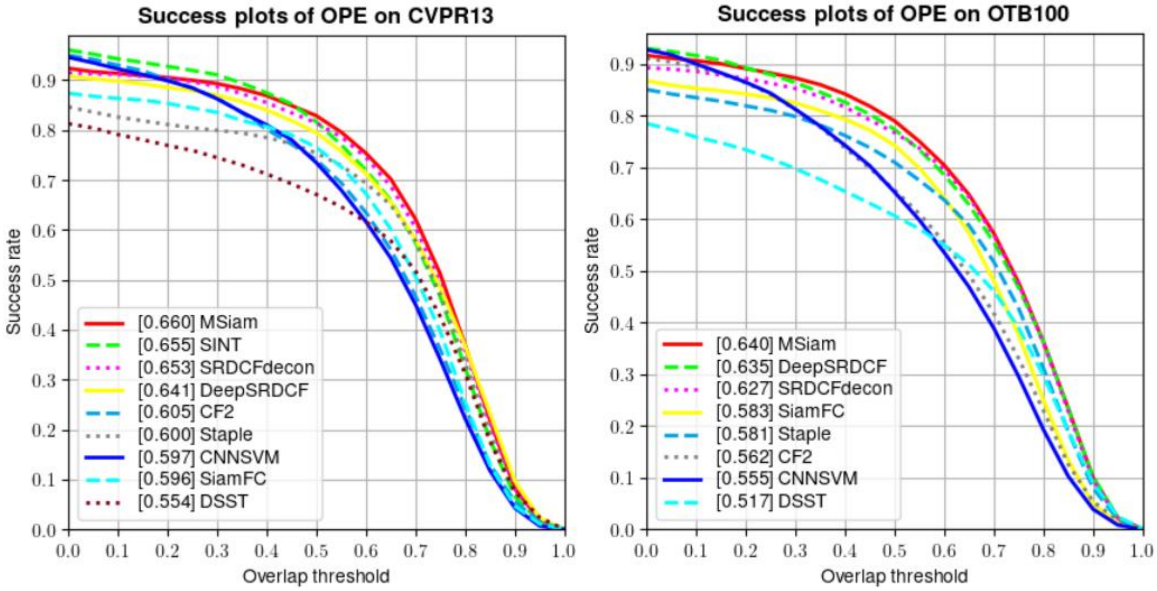

Figure 4: Overall performance on the OTB-2013 and OTB-2015.

4.2. State-of-the-art Comparison

OTB benchmark OTB-2013 Wu et al. (2013) is a widely used public tracking benchmark. The OTB-2013 and OTB-2015 dataset respectively include 50 and 100 sequences tagged with 11 attributes, and all sequences are fully annotated. We evaluate the proposed algorithms with comparisons to state-of-the-art trackers including DSST Danelljan et al. (2016b), SiamFC Bertinetto et al. (2016b), CNNSVM Hong et al. (2015), Staple Bertinetto et al. (2016a), CF2 Ma et al. (2015), DeepSRDCF Lukezic et al. (2017), SRDCFdecon Lukezic et al. (2017), SINT Tao et al. (2016). All the trackers were initialized with ground-truth object states in the first frame and average success plots were reported. The OPE criteria for OTB is applied to evaluate our MSiam. Figure. 4 hows the success plots in AUC, which is the bounding box overlapped ratio measures the Intersection-over-Union (IOU) ratio between the tracked bounding box and the ground truth. The left column is the success plots on OTB-2013, and the right column is success plots on OTB-2015. According to Figure. 4, our MSiam achieves the best performance among the state-of-the-art trackers on both datasets. The success plots are 0.660 on OTB-2013 and 0.640 OTB-2015, respectively.

VOT benchmark The VOT2015 Kristan et al. (2015) dataset consists of 60 sequences, aiming at assessing the short-term performance of trackers. The toolkit applies a reset-based methodology. The overall performance is evaluated using the Expected Average Overlap (EAO), which takes account of both accuracy and robustness.

Figure. 5 illustrates the EAO score evaluated on VOT2015, and 62 other state-of-the-art trackers are compared with our tracker. Although, our MSiam ranks second in terms of EAO score, MSiam can conduct at 37 FPS, which is more than 37 times of MDNet (first rank).

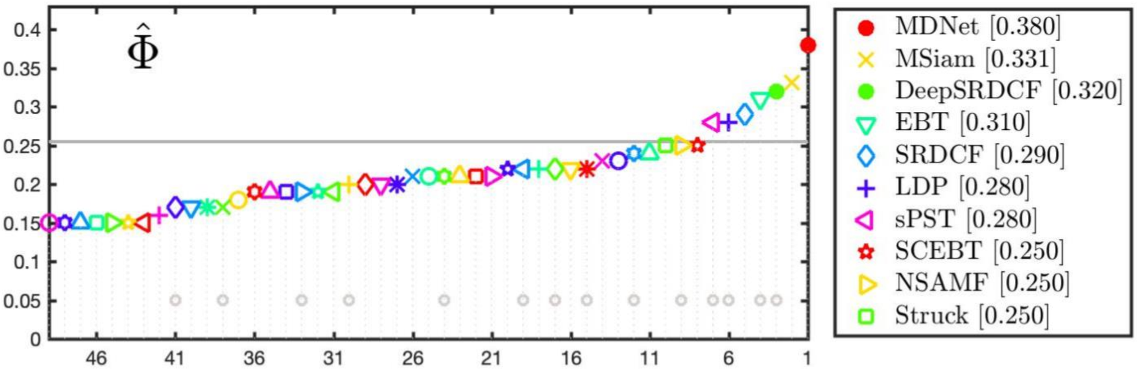

Figure 5: Overall performance on the VOT2015 bechmark.

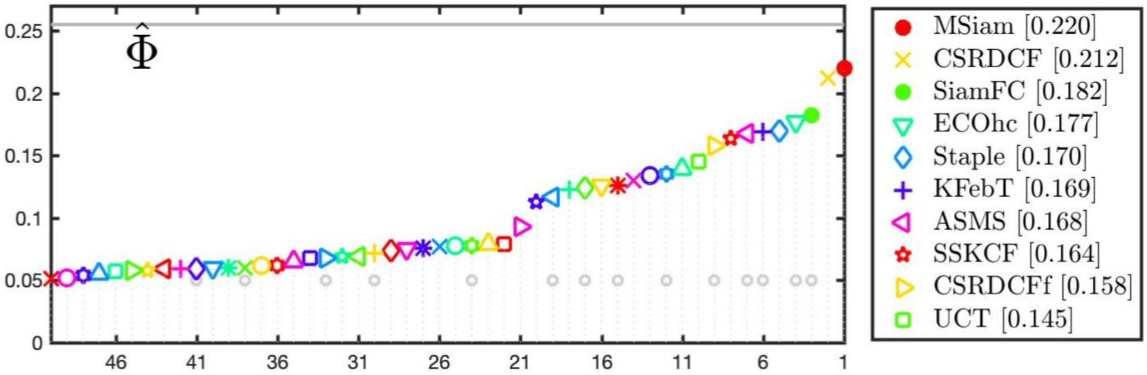

Figure 6: Overall performance on the VOT2017 real-time benchmark.

Compared with VOT2015, VOT2017 Kristan et al. (2017) replaces with 10 challenging sequences. Besides, a new real-time experiment is conducted. The real-time experiment requires trackers to deal with real-time video stream at least 25 FPS if the tracker fails to submit the tracking result in 40ms, the bounding box of the last frame will be reused as the result in the current frame.

Figure. 6 reports the results of MSiam against 51 other state-of-the-art trackers concerning the EAO score. MSiam achieves the first rank according to EAO score. Specifically, MSiam surpasses the original SiamFC by 21%. **GOT-10K benchmark** The GOT-10k benchmark test set embodies 84 object classes and 32 motion classes with only 180 video segments. The success rate (SR) is used for the evaluation of trackers, and it measures the percentage of successfully tracked frames where the overlaps exceed 0.5.

The success curves on GOT-10k benchmark is shown in Figure. 7. In this experiment, we compare our method with several representative trackers, including SiamFC Bertinetto et al. (2016b), GOTURN Held et al. (2016), CCOT Danelljan et al. (2016a), ECO Danelljan et al. (2017), MDNet Nam and Han (2016), BACF Kiani Galoogahi et al. (2017), and

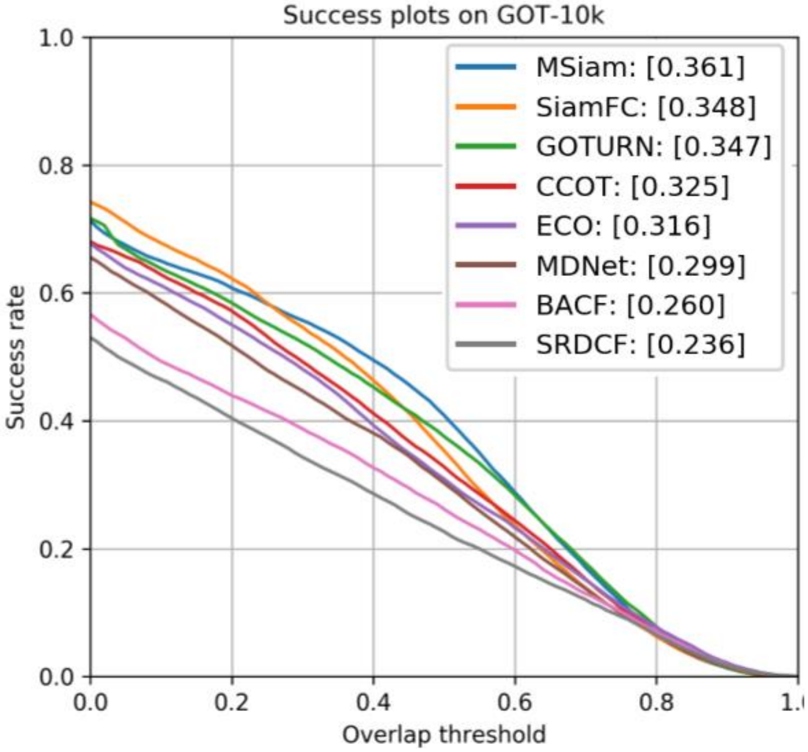

Figure 7: Overall performance on the GOT-10K bechmark.

SRDCF Lukezic et al. (2017). MSiam obtains 0.361 success score and achieves the best performance on this benchmark.

4.3. Comparison with baseline SiameseFC

As shown in Figure. 8, all compared sequences come from OTB-2015 benchmark, we compare our MSiam with SiameseFC Bertinetto et al. (2016b) on four challenging series: *MotorRolling* (the first column) with non-rigid appearance change, *Bolt2* (the second column) with similar distractors, *Bird1* (the third column) with full occlusion, and *Girl2* (the fourth column) with partial occlusion.

We observe that MSiam can distinguish the target from distractors, while SiameseFC drifts to the background in *Bolt2*. This fact proves the FAM is effective in recognizing distractors. In addition, compared to SiameseFC, our MSiam with multi-brach is able to deal with non-rigid appearance change, full occlusion, and partial occlusion in *MotorRolling*, *Bird1*, and *Girl2*. The results show that multi-branch bring local patterns detection ability to our model. Comparison between the green line (MSiam) and yellow line (MSiam without data augmentation) in sequence *Bird1* indicates that Msiam without random occlusion data augmentation can not tackle full occlusion problems. However, our model achieves a better way to tackle partial occlusion and non-rigid appearance change circumstances.

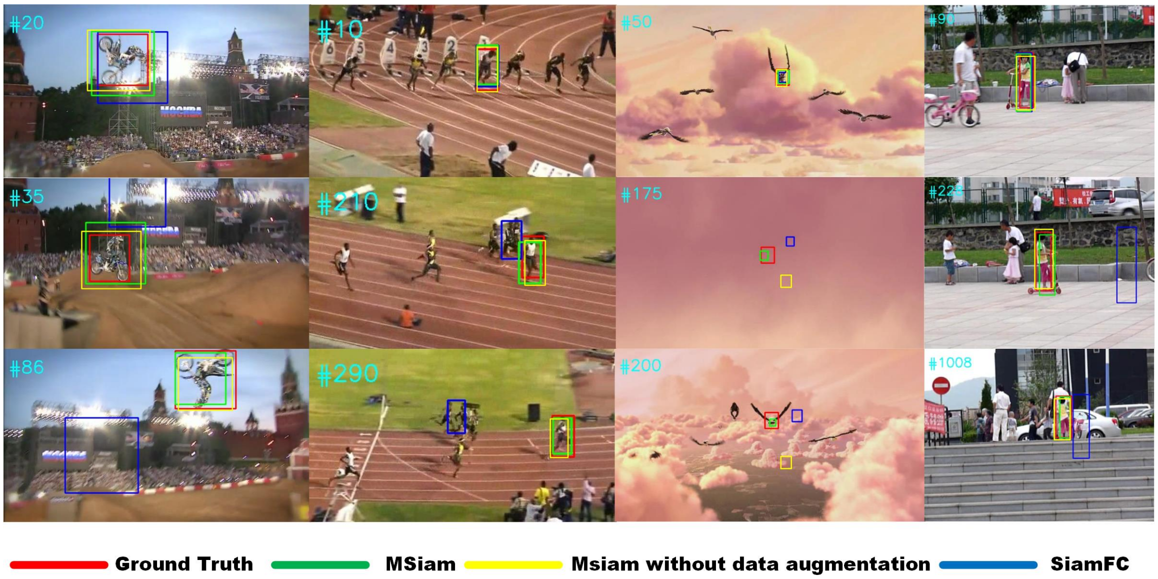

Figure 8: Comparisons between MSiam and SiameseFC.

4.4. Comparison with other variations of SiameseFC

In this section, we compare our MSiam with other variations of SiameseFC including StructSiam Zhang et al. (2018), Siam-tri Dong and Shen (2018), CIResNet-16 Zhipeng et al. (2019), and CIResNet-43 Zhipeng et al. (2019) on OTB-2015 benchmark. Table. 1 shows that our MSiam achieves best the best performance among the other variations of SiameseFC. It is worth mentioning that StructSiam also proposes a local structure learning method, but our method shows a better performance. Besides, CIResNet-16 and CIResNet-43 are ResNet-driven Siamese network while our method is AlexNet-driven. This fact proves our method is more effective even if the backbone is a shallow network.

Table 1: The comparisons of different variations of SiameseFC.

Tracker	SiamFC	StructSiam	Siam-tri	CIResNet-16	CIResNet-43	MSiam
AUC	0.583	0.621	0.592	0.632	0.638	0.640

4.5. Ablation Study

To show the impacts of different components of our tracker, we perform six variants of our tracker by employing the different combination of branches and evaluate them on OTB-2015 dataset. In this section, all training parameters and the dataset is the same for the variants. We test six variants of multi-branch MSiam in this section: ① represents only one branch $Conv1 \times 1$ is applied; ② represents two branches $Conv1 \times 1$ and $Conv6 \times 6$ are applied; ③ represents three branches $Conv1 \times 1$, $Conv2 \times 2$, and $Conv6 \times 6$ are applied; ④ represents three branches $Conv1 \times 1$, $Conv3 \times 3$, and $Conv6 \times 6$ are applied; ⑤ represents four branches $Conv1 \times 1$, $Conv2 \times 2$, $Conv3 \times 3$, and $Conv6 \times 6$ are applied; ⑥ represents all six branches $Conv1 \times 1$, $Conv2 \times 2$, $Conv3 \times 3$, $Conv4 \times 4$, $Conv5 \times 5$, and $Conv6 \times 6$ are applied. We list the six types of MSiam with AUC, Params, FLOPs, and Speed in Table. 2. Compared Single last layer features with Multi-level features, we can conclude that FAM, which enables our model to integrate multi-level features, improves the tracking performance for six variants. Moreover, compared ⑤ and ⑥ more branches are not always providing good effects to our method. ⑤ with multi-level features is the best variant in all six variants.

Table 2: Ablation study of our proposed method on OTB-2015.

		Multi-branch type					
		1	2	3	4	5	6
Single last layer features	AUC	0.591	0.599	0.605	0.611	0.623	0.617
	Params	2.402M	4.762M	5.024M	5.352M	6.401M	8.302M
	FLOPs	2.771G	3.424G	3.559G	3.668G	4.043G	4.691G
	Speed	96	84	79	77	64	32
Multi-level features	AUC	0.598	0.606	0.616	0.625	0.640	0.632
	Params	8.727M	11.087M	11.673M	12.136M	12.362M	14.623M
	FLOPs	5.991G	6.644G	6.888G	7.019G	7.023G	7.911G
	Speed	58	48	44	43	37	18

5. Conclusion

In this paper, we propose a novel multi-branch framework MSiam for visual tracking. Compared with previous arts, MSiam demonstrates more robust performance in handling complex backgrounds such as similar distractors by aggregating multi-level features, non-rigid appearance change, and partial occlusion by providing multi-scale local pattern detection. In addition, the proposed FAM enables effective feature leverage across layers for more discriminative representation, LPDM is able to locate the target using local clue. Extensive experiments on 5 public datasets have validated the advantages of tracking robustness and efficiency of the proposed method, and our model runs in real-time.

Acknowledgments

This work is supported by Chinese National Natural Science Foundation under Grants 61532018.

References

Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr. Staple: Complementary learners for real-time tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1401–1409, 2016a.

Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016b.

M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, 2016a.

Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 4310–4318, 2015.

Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Discriminative scale space tracking. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1561–1575, 2016b.

Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *European Conference on Computer Vision*, pages 472–488. Springer, 2016c.

Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: efficient convolution operators for tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6638–6646, 2017.

Xingping Dong and Jianbing Shen. Triplet loss in siamese network for object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 459–474, 2018.

Heng Fan and Haibin Ling. Sanet: Structure-aware network for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 42–49, 2017.

David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *European Conference on Computer Vision*, pages 749–765. Springer, 2016.

Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. Online tracking by learning discriminative saliency map with convolutional neural network. In *International conference on machine learning*, pages 597–606, 2015.

Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *arXiv preprint arXiv:1810.11981*, 2018.

Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1135–1143, 2017.

Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Luka Cehovin, Gustavo Fernandez, Tomas Vojir, Gustav Hager, Georg Nebehay, and Roman Pflugfelder. The visual object tracking vot2015 challenge results. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 1–23, 2015.

Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Cehovin Zajc, Tomas Vojir, Gustav Hager, Alan Lukezic, Abdelrahman Eldesokey, et al. The visual object tracking vot2017 challenge results. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1949–1972, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8971–8980, 2018.

Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017.

Ting Liu, Gang Wang, and Qingxiong Yang. Real-time part-based visual tracking via adaptive correlation filters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4902–4912, 2015.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

Alan Lukezic, Tomás Vojír, L Cehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 4847–4856, 2017.

Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 3074–3082, 2015.

Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4293–4302, 2016.

Yuankai Qi, Shengping Zhang, Lei Qin, Hongxun Yao, Qingming Huang, Jongwoo Lim, and Ming-Hsuan Yang. Hedged deep tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4303–4311, 2016.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing & Computer-assisted Intervention*, 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Jeany Son, Ilchae Jung, Kayoung Park, and Bohyung Han. Tracking-by-segmentation with online gradient boosting decision tree. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3056–3064, 2015.

Yibing Song, Ma Chao, Xiaohe Wu, Lijun Gong, and Ming Hsuan Yang. Vital: Visual tracking via adversarial learning. 2018.

Chong Sun, Dong Wang, Huchuan Lu, and Ming-Hsuan Yang. Learning spatial-aware regressions for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8962–8970, 2018.

Ran Tao, Efstratios Gavves, and Arnold WM Smeulders. Siamese instance search for tracking. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 1420–1429. IEEE, 2016.

Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu. Visual tracking with fully convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 3119–3127, 2015.

Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu. Stct: Sequentially training convolutional networks for visual tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Qiang Wang, Zhu Teng, Junliang Xing, Jin Gao, Weiming Hu, and Stephen Maybank. Learning attentions: residual attentional siamese network for high performance online visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4854–4863, 2018.

Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2411–2418, 2013.

Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015.

Li Yang, Jianke Zhu, and Steven C. H. Hoi. Reliable patch trackers: Robust visual tracking by exploiting reliable patches. In *Computer Vision & Pattern Recognition*, 2015.

Guangcong Zhang and Patricio A Vela. Good features to track for visual slam. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1373–1382, 2015.

Yunhua Zhang, Lijun Wang, Jinqing Qi, Dong Wang, Mengyang Feng, and Huchuan Lu. Structured siamese network for real-time visual tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 351–366, 2018.

Zhang Zhipeng, Peng Houwen, and Wang Qiang. Deeper and wider siamese networks for real-time visual tracking. *arXiv preprint arXiv:1901.01660*, 2019.