COMPLEMENTARY SIAMESE NETWORKS FOR ROBUST VISUAL TRACKING

Heng Fan[†], *Lu Xu*[‡], *and Jinhai Xiang*[‡,*]

[†]Computer & Information Sciences Department, Temple University, Philadelphia 19122, USA
[‡]College of Informatics, Huazhong Agricultural University, Wuhan, 430070, China

ABSTRACT

In this paper, we propose the novel complementary Siamese networks (CoSNet) for visual tracking by exploiting complementary global and local representations to learn a matching function. In specific, the proposed CoSNet is two-fold: a global Siamese network (GSNet) and a local Siamese network (LSNet). The GSNet aims to match the target with candidates using holistic representation. By contrast, the LSNet explores partial object representation for matching. Instead of simply decomposing the object into regular patches in LSNet, we propose a novel attentional local part network, which automatically generates salient object parts for local representation and adaptively weights each part according to its importance in matching. In CoSNet, the GSNet and LSNet are jointly trained in an end-to-end manner. By coupling two complementary Siamese networks, our CoSNet learns a robust matching function which can effectively handle various appearance changes in visual tracking. Extensive experiments on a large-scale dataset with 100 sequences show that CoSNet outperforms other state-of-the-art trackers.

Index Terms— Visual tracking, complementary Siamese network, attention model, local and global model

1. INTRODUCTION

Visual tracking plays a crucial role in computer vision with many applications such as video surveillance, intelligent vehicles, etc. Despite great advances, visual tracking remains challenging due to many factors including rotation, deformation and so on. To handle these issues, numerous visual trackers have been proposed [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

By formulating tracking as matching the target object in the first frame with candidates in a new frame, Siamese network has recently drawn increasing interest in visual tracking due to its ability to learn a powerful *generic* matching function, and many trackers have been proposed. Tao *et al.* [13] learn a matching function off-line from a great deal of videos and directly apply it to tracking without any model update. Bertinetto *et al.* [12] introduce a fully-convolutional Siamese architecture to learn a discriminative similarity measurement for tracking, achieving robust performance while running in real-time. Guo *et al.* [14] propose a dynamic Siamese network for tracking by adding extra transformations to update the target template for tracking. In [15], He *et al.* propose a two-fold Siamese network to learn a both discriminative and generative similarity measurement for tracking.

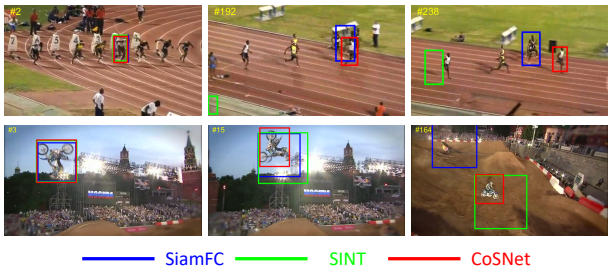

Fig. 1: Comparisons of CoSNet to other Siamese trackers on sequences *Bolt2* with deformation and *MotorRolling* with heavy rotation. Our CoSNet can well handle deformation and rotation and performs better than SiamFC [12] and SINT [13].

Despite good performance, existing Siamese trackers still fail in challenging situations with heavy rotation and deformation (see Fig. 1), because these methods only consider the holistic representation of object for matching. When heavy rotation or deformation happen, the global representation learned by Siamese network changes drastically. Therefore, it is difficult to use the matching function in these cases to correctly measure the similarities between the target object and candidates in a new frame.

To address the above issues, we propose the novel Complementary Siamese Network (CoSNet) by considering both global and local representation of object for matching. Specifically, CoSNet consists of two components: a Global Siamese network (GSNet) and a Local Siamese network (LSNet). As in existing approaches, the GSNet is utilized to capture the holistic representation of object for matching. To deal with the issues of deformation and rotation, we propose the LSNet to explore local representation of object for matching, which is complementary to global representation of GSNet. Instead of simply dividing object into regular patches for local representation, we propose an attentional local part network to automatically generate salient object parts in LSNet. Our local part network is able to extract more accurate yet meaningful salient parts from object, and semantically aligns these parts,

*Corresponding author.

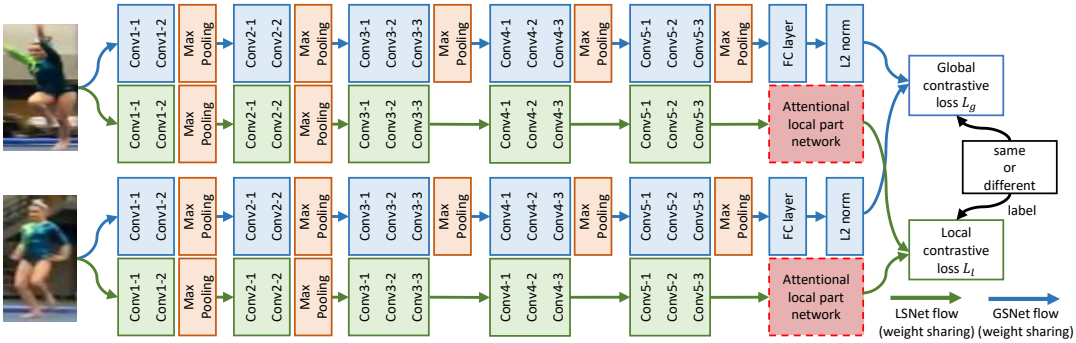

Fig. 2: Illustration of the architecture of our CoSNet, which is composed of complementary GSNet (the blue flow) and LSNet (the green flow). The two CNN branches of GSNet (or LSNet) share parameters with each other. Best viewed in color.

resulting in better local representation for matching. In addition, considering that object parts of different objects function differently in matching, we introduce an attention model into the local part network to adaptively weight each part based on its importance in the local representation, further improving the robustness of matching. Extensive experiments on a large-scale dataset evidence the effectiveness of our method.

In summary, our contributions are three-fold: (1) We propose a novel CosNet for object tracking by taking into account complementary global and local object representation to learn a robust matching function. (2) We propose an attentional local part network to generate salient object parts for local representation and adaptively weight each one according to its importance for matching. (3) Experiments on a large-scale dataset [16] show that CoSNet outperforms other state-of-the-art trackers.

2. THE PROPOSED ALGORITHM

2.1. Complementary Siamese Network (CoSNet)

We consider visual tracking as matching target object in the first frame with candidates in a new frames. To this end, we leverage the powerful deep Siamese network [17] to *off-line* learn a *generic* matching function from a set of sequences. Unlike existing approaches [13, 12, 15, 18], we take into account both holistic template and local representation to learn a robust matching function, as shown in Fig. 2.

The CoSNet comprises two complementary Siamese networks, i.e., GSNet and LSNet, which are jointly end-to-end trained. The loss L of the CoSNet can be expressed as the sum of losses of two sub-networks as follows

$$L = L_g + L_l \tag{1}$$

where L_g and L_l are the losses of GSNet and LSNet (as described later). Once trained, we can use the learned matching function as it is for tracking, without any further update.

Global Siamese network (GSNet): The GSNet aims to capture the global representation of object for matching. In specific, GSNet contains two identical CNN branches borrowed from VGGNet [19], as shown in Fig. 2 (the blue flow). Specifically, we employ the truncated VGGNet pre-trained on ImageNet [20], and discard all layers after the 5^{th} pooling layer. A new fully connected layer is added after the last pooling layer, followed by a ℓ_2 normalization layer. In the end, the two CNN branches are connected by a single contrastive loss layer, as shown in Fig. 2. The GSNet takes as inputs two images $\mathbf{x}_j$ and $\mathbf{x}_k$, and the contrastive loss L_g is expressed as

$$\begin{aligned} L_g(\mathbf{x}_j, \mathbf{x}_k, r_{jk}) = \frac{1}{2} r_{jk} \|\phi(\mathbf{x}_j) - \phi(\mathbf{x}_k)\|_2^2 + \\ \frac{1}{2}(1 - r_{jk}) \max(0, \epsilon - \|\phi(\mathbf{x}_j) - \phi(\mathbf{x}_k)\|_2^2) \end{aligned} \tag{2}$$

where $\phi(\cdot)$ represents the feature transformation via GSNet, $r_{jk} \in \{0, 1\}$ indicates that x_j and x_k are the same object or not, and ϵ denotes the minimum distance margin.

Local Siamese network (LSNet): The LSNet is used to explore local representation for matching. As shown in Fig. 2 (the green flow), LSNet comprises two CNN branches borrowed from VGGNet [19], except for layers after the 5^{th} pooling layer. Considering that local representation is susceptive to feature resolution, we designate LSNet with fewer pooling layers. Specifically, we only utilize two max pooling layers after conv1-2 and conv2-2. The features obtained from the conv5-3 layers are then fed into attentional local part net to generate salient object parts, which are concatenated to form the local representation. Finally, a contrastive loss layer is added to receive the local representations from LSNet. The LSNet receives two same images $\mathbf{x}_j$ and $\mathbf{x}_k$ as in the GSNet, and its contrastive loss L_l is computed as

$$\begin{aligned} L_l(\mathbf{x}_j, \mathbf{x}_k, r_{jk}) = \frac{1}{2} r_{jk} \|\varphi(\mathbf{x}_j) - \varphi(\mathbf{x}_k)\|_2^2 + \\ \frac{1}{2}(1 - r_{jk}) \max(0, \epsilon - \|\varphi(\mathbf{x}_j) - \varphi(\mathbf{x}_k)\|_2^2) \end{aligned} \tag{3}$$

where $\varphi(\cdot)$ is the local representation obtained by the attentional local part net in LSNet, as detailed in Section 2.2.

2.2. Attentional Local Part Net in LSNet

The core of LSNet is the proposed attentional local part net, which aims to detect salient part maps, output the part feature

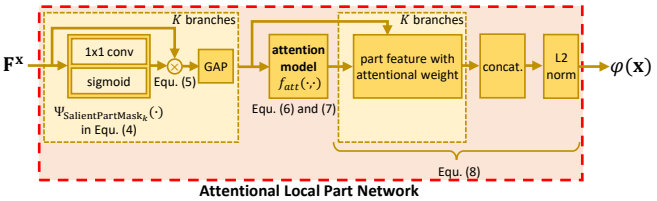

Fig. 3: Illustration of the architecture of attentional local part network. Best viewed in color and by zooming in.

of each part, and adaptively concatenate them with attention to form the local representation.

As illustrated in Fig. 3, the attentional local part net contains K branches, with each containing a convolutional layer followed by a non-linear sigmoid layer. Each branch receives feature from the conv5-3 layer in LSNet, and extracts a salient object region as the output. Afterwards, we can generate part features based on salient regions. Considering that for each object, the local parts function differently in matching, we propose an attention model to adaptively combine them.

For image $\mathbf{x}$, let a 3-dimension tensor $\mathbf{F}^{\mathbf{x}} \in \mathbb{R}^{H\times W\times C}$ represent the feature obtained after conv5-3 in LSNet, which is then fed to the part net. We can estimate the 2-dimension salient part masks $\mathbf{M}_k^{\mathbf{x}} \in \mathbb{R}^{H\times W}$ for $\mathbf{x}$ as

$$\mathbf{M}_k^{\mathbf{x}} = \Psi_{\text{SalientPartMask}_k}(\mathbf{F}^{\mathbf{x}}), \quad k = 1, 2, \cdots, K \tag{4}$$

where $\Psi_{\text{SalientPartMask}_k}(\cdot)$ is the k^{th} salient part region extractor. Fig. 4 demonstrates the detected salient part masks for objects in testing dataset.

With salient object map $\mathbf{M}_k^{\mathbf{x}}$, the corresponding part feature $\mathbf{F}_k^{\mathbf{x}} \in \mathbb{R}^{H\times W\times C}$ for the k^{th} region of $\mathbf{x}$ is computed as

$$\mathbf{F}_k^{\mathbf{x}}(x, y, c) = \mathbf{F}^{\mathbf{x}}(x, y, c) \times \mathbf{M}_k^{\mathbf{x}}(x, y) \tag{5}$$

where $\mathbf{F}^{\mathbf{x}}(x, y, c)$ is the element in the c^{th} channel over the location (x, y) in $\mathbf{F}^{\mathbf{x}}$, and $\mathbf{M}_k^{\mathbf{x}}(x, y)$ represents the element on the location (x, y) in $\mathbf{M}_k^{\mathbf{x}}$. After obtaining the part feature, we apply a global average pooling (GAP) operator on each $\mathbf{F}_k^{\mathbf{x}}$ to derive the final part feature $\mathbf{f}_k^{\mathbf{x}} = \text{GAP}(\mathbf{F}_k^{\mathbf{x}}) \in \mathbb{R}^{1\times 1\times C}$.

Taking into account that each object part plays a different role in matching, we use an attention model [21] to adaptively assign weight to each part. Mathematically, the attention model $f_{att}(\cdot,\cdot)$ is formulated as follows

$$e_k^{\mathbf{x}} = f_{att}(\mathbf{W}_{att}, \mathbf{f}_k^{\mathbf{x}}) = \mathbf{W}_{att}(\mathbf{f}_k^{\mathbf{x}})^{\mathrm{T}} \tag{6}$$

$$\alpha_k^{\mathbf{x}} = \frac{\exp(e_k^{\mathbf{x}})}{\sum_{k=1}^{K}\exp(e_k^{\mathbf{x}})} \tag{7}$$

where $\mathbf{W}_{att} \in \mathbb{R}^{1\times C}$ denotes the parameters of the attention model and can be learned end-to-end, and $\alpha_k^{\mathbf{x}}$ represents the weight of the k^{th} salient region of image $\mathbf{x}$.

Once $\alpha_k^{\mathbf{x}}$ are computed, we concatenate all the part features to derive $\varphi(\mathbf{x})$ for $\mathbf{x}$, followed by ℓ_2 normalization

$$\varphi(\mathbf{x}) = \left\| [\alpha_1^{\mathbf{x}}(\mathbf{f}_1^{\mathbf{x}})^{\mathrm{T}}, \alpha_2^{\mathbf{x}}(\mathbf{f}_2^{\mathbf{x}})^{\mathrm{T}}, \cdots, \alpha_K^{\mathbf{x}}(\mathbf{f}_K^{\mathbf{x}})^{\mathrm{T}}]^{\mathrm{T}} \right\|_2 \tag{8}$$

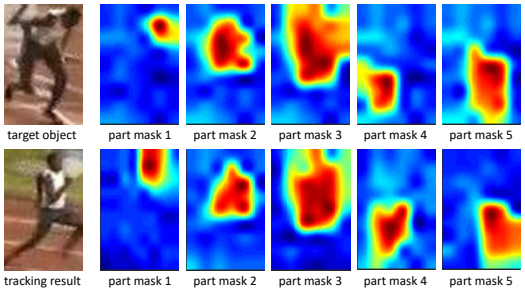

Fig. 4: The extracted local salient part masks in our method. The first column shows the target object (top row) in the first frame and the tracking result (bottom row) in a subsequent frame, followed by the learned local salient part masks.

With Equ. (8), we can calculate local representations $\varphi(\mathbf{x}_j)$ and $\varphi(\mathbf{x}_k)$ for images $\mathbf{x}_j$ and $\mathbf{x}_k$ in the Equ. (3).

2.3. Matching based Visual Tracking

Once the training of CoSNet is completed, we use the learned matching function $f_{match}(\cdot,\cdot)$ as it is for tracking, without any further update. To this end, we adopt the simple particle filter framework. In the $t^{th}(t \geq 2)$ frame, we sample n candidates $\{\mathbf{x}_i^t\}_{i=1}^n$ around the estimated position of the target in the last frame, and compare these candidates with template $\mathbf{x}_{obj}^1$ in the first frame using $f_{match}(\cdot,\cdot)$. The tracking result $\hat{\mathbf{x}}^t$ is determined by the candidate which best matches $\mathbf{x}_{obj}^1$,

$$\hat{\mathbf{x}}^t = \arg\max_{\mathbf{x}_i^t} \; f_{match}(\mathbf{x}_{obj}^1, \mathbf{x}_i^t) \tag{9}$$

where the matching function is mathematically computed as

$$f_{match}(\mathbf{a}, \mathbf{b}) = w_g \underbrace{\phi(\mathbf{a})^{\mathrm{T}}\phi(\mathbf{b})}_{\text{global template}} + w_l \underbrace{\varphi(\mathbf{a})^{\mathrm{T}}\varphi(\mathbf{b})}_{\text{local representa.}} \tag{10}$$

where $\phi(\cdot)$ and $\varphi(\cdot)$ are feature transformations via GSNet and LSNet, respectively, and w_g and w_l denote the weights of global template and local representation in final matching which are adaptively computed as

$$w_g = \frac{\phi(\mathbf{a})^{\mathrm{T}}\phi(\mathbf{b})}{\phi(\mathbf{a})^{\mathrm{T}}\phi(\mathbf{b}) + \varphi(\mathbf{a})^{\mathrm{T}}\varphi(\mathbf{b})}, \quad w_l = 1 - w_g \tag{11}$$

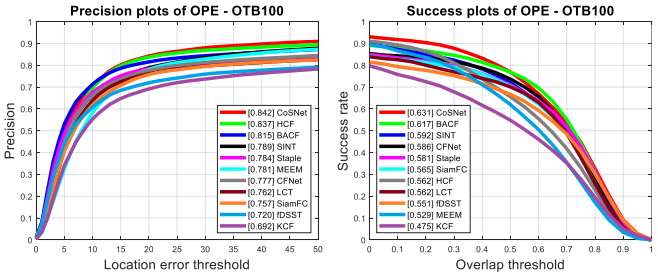

Fig. 5: Comparisons on OTB-100 using DPR and OSR. Our CoSNet outperforms other state-of-the-art trackers.

Table 1: Average DPR of 11 attributes. The best three results are shown in **red**, **blue** and **green** fonts, respectively.

Att.	SiamFC	CFNet	MEEM	Staple	SINT	BACF	HCF	CoSNet
IV	0.735	0.757	0.740	0.791	**0.816**	0.808	**0.817**	**0.858**
OPR	0.745	0.753	0.798	0.742	**0.814**	0.777	**0.810**	**0.827**
SV	0.743	0.748	0.740	0.731	0.750	**0.778**	**0.802**	**0.803**
OCC	0.696	0.713	0.741	0.726	**0.756**	0.728	**0.767**	**0.830**
DEF	0.676	0.669	0.754	0.748	0.745	**0.759**	**0.791**	**0.812**
MB	0.698	**0.761**	0.721	0.726	0.747	0.753	**0.797**	**0.798**
FM	0.730	0.741	0.734	0.703	0.739	**0.784**	**0.797**	**0.763**
IPR	0.748	0.803	0.793	0.770	**0.830**	0.778	**0.854**	**0.830**
OV	0.678	0.650	0.683	0.661	0.720	**0.765**	**0.677**	**0.736**
BC	0.694	0.737	0.751	0.770	0.782	**0.833**	**0.847**	**0.877**
LR	**0.834**	**0.861**	0.605	0.609	0.786	0.707	0.787	**0.789**
Overall	0.757	0.777	0.781	0.784	0.789	**0.815**	**0.837**	**0.842**

Table 2: Average OSR of 11 attributes. The best three results are shown in **red**, **blue** and **green** fonts, respectively.

Att.	SiamFC	CFNet	MEEM	Staple	SINT	BACF	HCF	CoSNet
IV	0.549	0.574	0.517	0.598	**0.625**	**0.630**	0.540	**0.644**
OPR	0.544	0.553	0.528	0.538	**0.600**	**0.581**	0.537	**0.611**
SV	0.555	0.555	0.473	0.529	**0.564**	**0.579**	0.488	**0.590**
OCC	0.523	0.536	0.503	0.548	**0.574**	**0.567**	0.525	**0.624**
DEF	0.490	0.492	0.489	**0.554**	0.550	**0.572**	0.530	**0.598**
MB	0.555	**0.593**	0.543	0.558	**0.588**	0.582	0.573	**0.614**
FM	0.564	**0.570**	0.528	0.541	0.567	**0.596**	0.555	**0.598**
IPR	0.557	**0.590**	0.528	0.552	**0.599**	0.575	0.559	**0.607**
OV	0.511	0.480	0.484	0.481	**0.553**	**0.552**	0.474	**0.570**
BC	0.504	0.545	0.521	0.574	**0.591**	**0.623**	0.587	**0.647**
LR	**0.604**	**0.619**	0.355	0.411	0.543	0.512	0.424	**0.563**
Overall	0.565	0.586	0.529	0.581	**0.592**	**0.617**	0.562	**0.631**

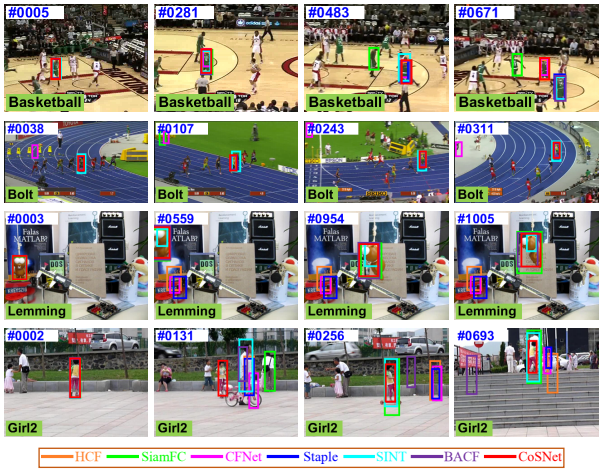

Fig. 6: Qualitative evaluation of CoSNet and other six state-of-the-art trackers on several challenging sequences.

3. EXPERIMENTS

3.1. Implementation details

We train CosNet using ALOV dataset [22] as in [13]. It is worth noting that *we exclude all videos in ALOV that are also in our evaluation benchmark OTB-100*. We utilize the parameters of VGGNet to initialize both GANet and LSNet. After that, we fine tune the CoSNet using Caffe [23] for 80 epochs. The initial learning rate for fine tuning is 0.001, and decreased by a factor of 10 after every 5 epochs. The weight decay parameter is set to 0.001, and batch size is 1.

Our CoSNet is implemented in Matlab using Caffe wrapper, and runs at around 3 frames per second (fps) on a PC with an i7 Core CPU and a NVIDIA GTX 1080 GPU. In each frame, we sample 200 ($n = 200$) candidates in translation and scale dimension from a Gaussian distribution for matching. The number K of object parts is empirically set to 5.

3.2. Experiments on OTB-100

We evaluate CoSNet on OTB-100 [16] and compare it to ten algorithms, including CFNet [18], BACF [24], fDSST [25], SINT [13], HCF [7], SiamFC [12], LCT [26], MEEM [27], Staple [4] and KCF [3]. As in [16], we use distance precision rate (DPR) and overlap success rate (OSR) for evaluation.

We report the comparison results in *one-pass evaluation* as shown in Fig 5. Overall, CoSNet achieves the best performance with a DPR of 84.2% and an OSR of 63.1%. Compared with the second best trackers HCF with a DPR of 83.7% and BACF with an OSR of 61.7%, CoSNet obtains the improvements of 0.5% and 1.4% in DPR and OSR, respectively. Compared to SINT with a 78.9% DPR and a 59.2% OSR, CoSNet obtains the gains of 6.2% in DPR and 3.9% in OSR, showing the power of local representation.

In order to further analyze the performance of CoSNet, we conduct attribute-based evaluation on OTB-100 [16], and compare CoSNet to seven tracking approaches in DPR and OSR as shown in Tab. 1 and 2. For DPR, CoSNet achieves the best results under 7 out of 11 attributes including IV, OPR, SV, OCC, DEF, MB and BC. For IPR, OV, FM and LR, the proposed CoSNet ranks in top three, showing competitive performance. For OSR, CoSNet achieves the best results under 10 out of 11 attributes including DEF, OPR, SV, OCC, IPR, IV, MB, FM, OV and BC. In the challenge of LR, our CoSNet still obtains the third ranking score.

Fig. 6 shows the qualitative results of CoSNet on four sequences. We observe that our tracker is able to deal with various challenges such as occlusion, deformation and rotation, while other tracker can only handle several of them.

4. CONCLUSION

This paper proposes the CoSNet for tracking. The CoSNet comprises two complementary networks, GSNet and LSNet, to exploit both global and local representations of target object for tracking. By coupling two complementary cues, we learn a robust matching function from a large set of videos and deploy it for tracking without adaption. Experiments on a large-scale dataset evidence the effectiveness of CoSNet.

Acknowledgments: This work was supported by Foundation Research Funds for Central Universities (No. 2662017JC049), State Scholarship Fund (NO.261606765054) and the National Key R&D Program of China (NO.2018YFC1604000).

5. REFERENCES

[1] Heng Fan and Haibin Ling, "Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking," in *ICCV*, 2017, pp. 5487–5495.

[2] Hyeonseob Nam and Bohyung Han, "Learning multi-domain convolutional neural networks for visual tracking," in *CVPR*, 2016, pp. 4293–4302.

[3] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "High-speed tracking with kernelized correlation filters," *TPAMI*, vol. 37, no. 3, pp. 583–596, 2015.

[4] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr, "Staple: Complementary learners for real-time tracking," in *CVPR*, 2016, pp. 1401–1409.

[5] Heng Fan and Jinhai Xiang, "Robust visual tracking with multitask joint dictionary learning.," *TCSVT*, vol. 27, no. 5, pp. 1018–1030, 2017.

[6] Heng Fan and Haibin Ling, "Sanet: Structure-aware network for visual tracking," in *CVPRW*, 2017.

[7] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang, "Hierarchical convolutional features for visual tracking," in *ICCV*, 2015, pp. 3074–3082.

[8] Heng Fan, Jinhai Xiang, and Liang Zhao, "Robust visual tracking via bag of superpixels," *MTAP*, vol. 75, no. 14, pp. 8781–8798, 2016.

[9] Chenglong Bao, Yi Wu, Haibin Ling, and Hui Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *CVPR*, 2012, pp. 1830–1837.

[10] Heng Fan, Jinhai Xiang, Honghong Liao, and Xiaoping Du, "Robust tracking based on local structural cell graph," *JVCIR*, vol. 31, pp. 54–63, 2015.

[11] Heng Fan and Jinhai Xiang, "Robust visual tracking via local-global correlation filter.," in *AAAI*, 2017, pp. 4025–4031.

[12] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr, "Fully-convolutional siamese networks for object tracking," in *ECCVW*, 2016, pp. 850–865.

[13] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders, "Siamese instance search for tracking," in *CVPR*, 2016, pp. 1420–1429.

[14] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang, "Learning dynamic siamese network for visual object tracking," in *ICCV*, 2017, pp. 1763–1771.

[15] Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng, "A twofold siamese network for real-time object tracking," in *CVPR*, 2018, pp. 4834–4843.

[16] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Object tracking benchmark," *TPAMI*, vol. 37, no. 9, pp. 1834–1848, 2015.

[17] Sumit Chopra, Raia Hadsell, and Yann LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *CVPR*, 2005, pp. 539–546.

[18] Jack Valmadre, Luca Bertinetto, João Henriques, Andrea Vedaldi, and Philip HS Torr, "End-to-end representation learning for correlation filter based tracking," in *CVPR*, 2017, pp. 5000–5008.

[19] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.

[21] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015.

[22] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah, "Visual tracking: An experimental survey," *TPAMI*, vol. 36, no. 7, pp. 1442–1468, 2014.

[23] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM MM*, 2014, pp. 675–678.

[24] H Kiani Galoogahi, Ashton Fagg, and Simon Lucey, "Learning background-aware correlation filters for visual tracking," in *ICCV*, 2017, pp. 1135–1143.

[25] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg, "Discriminative scale space tracking," *TPAMI*, vol. 39, no. 8, pp. 1561–1575, 2017.

[26] Chao Ma, Xiaokang Yang, Chongyang Zhang, and Ming-Hsuan Yang, "Long-term correlation tracking," in *CVPR*, 2015, pp. 5388–5396.

[27] Jianming Zhang, Shugao Ma, and Stan Sclaroff, "Meem: robust tracking via multiple experts using entropy minimization," in *ECCV*, 2014, pp. 188–203.