REGION-BASED FULLY CONVOLUTIONAL SIAMESE NETWORKS FOR ROBUST REAL-TIME VISUAL TRACKING

*Longchao Yang, Peilin Jiang, Fei Wang, Xuan Wang**

Institute of Artificial Intelligence and Robotics
Xi'an Jiaotong University
710049, 28 Xianning Road, Xi'an, China

ABSTRACT

Partial occlusions and deformations in visual object tracking are still very challenging. Existing Convolutional Neural Networks (CNNs) trackers either fail to handle these issues or can just run in low speed. In this paper, we present a real-time tracker which is robust to occlusions and deformations based on a Region-based, Fully Convolutional Siamese Network (R-FCSN). In the proposed R-FCSN, the information of regions is extracted separately by the proposition of position-sensitive score maps. Combining these score maps via adaptive weights leads to accurate location of the target on a new frame. The experiments illustrate that our method outperforms state-of-the-art approaches, and can handle the cases of object deformation and occlusion at about 51 FPS.

Index Terms— visual tracking, region-based, adaptive weights, Siamese-network, deep learning

1. INTRODUCTION

Visual tracking is an important computer vision task with various applications. Since it has been actively studied for decades, most of the tracking tasks in simple environment with slow motion and slight occlusion can be addressed effectively by current algorithms. The trackers based on CNNs show great potential in handling more challenging situations such as heavy occlusion, illumination variation, abrupt motion and deformation (as sequences in Fig.1), however encounter a dilemma between accuracy and efficiency. In this paper, we focus on more robust method which can realize accurate and real-time tracking in challenging situations.

Current visual tracking methods can be categorized as either generative or discriminative[1]. Generative methods use appearance models to represent the target object and search for the most similar regions to generative models, such as incremental tracker[2], sparser tracker[3]. Discriminative methods formulate object as a binary classification that distinguish target from background, including TLD[4], struck tracker[5], etc. Most of these methods, however, delineate the entire tracked target by a single regular bounding box, which renders them sensitive to partial occlusion, deformation, and other issues. Part-based methods[6, 7] divide the entire target into parts and track them separately to resist occlusion. However, these methods are difficult to achieve real-time tracking since the tracker has to repeat several times for multiple parts.

CNNs have demonstrated their outstanding representation capacity in a wide range of computer vision applications, including image classification[12], object detection[13], segmentation[14], etc. Due to such a huge success, a large number of tracking algorithms using the representations from CNNs have been proposed recently, such as MDNet[1], TCNN[15], C-COT[16]. TCNN and C-COT which utilize

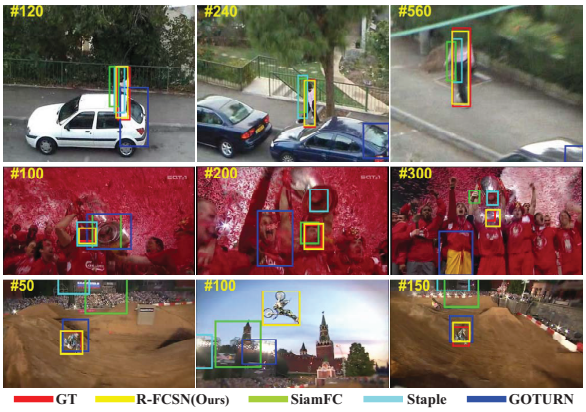


Fig. 1. A comparison of our tracker R-FCSN with several state-of-the-art real-time tracking methods, SiamFC[8], Staple[9], GOTURN[10], and GT (Ground-Truth) in three challenging sequences of OTB100 dataset[11]: Woman (top row, *occlusion and deformation*), Soccer (middle row, *occlusion and motion-blur*), MotorRolling (bottom row, *deformation and rotation*). R-FCSN outperforms all these methods in tackling above challenging tracking situations.

This work was supported in part by Natural Science Foundation of China (No.61231018), National Science and Technology Support Program (2015BAH31F01) and Program of Introducing Talents of Discipline to University under grant B13043.

*Corresponding author.

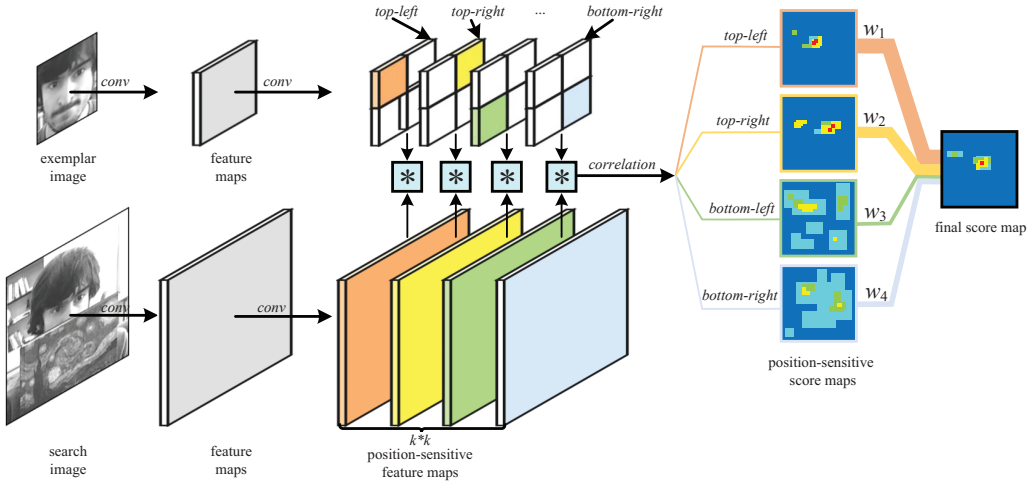


Fig. 2. Region-based Fully Convolutional Siamese architecture. We build $k \times k$ (2×2) position-sensitive feature maps by adding an 1 x 1 convolutional layer to feature map of search image. Then cross-correlation is used to compute the similarity between each position of exemplar image's feature maps and correspondent search image's feature maps. By our designed adaptive weights, the position-sensitive scores of regions not being occluded or deformed (*bottom-left* and *bottom-right* in above examples) contribute more on the final joint score map (the line width indicates the portion of contribution).

multiple features from deep convolutional networks have achieved best results in VOT challenge[17]. Unfortunately, these methods could not run in real time for the complexity of computation and the depth of the networks, which brings a lot of inconvenience in their applications. Siamese networks encode the target image and search image by two branches of CNNs separately, and then combine them to locate the target on the search image. [8, 10] introduce the Siamese architectures which use representations from a shallow network, i.e., AlexNet[12], they can be implemented at 80-170FPS but lack robustness to the challenging situations.

Region-based Fully Convolutional Networks (R-FCN)[18] which encode information of different regions independently show a superior performance in object detection[18], instance segmentation[19] compared to fully convolutional networks (FCN). Motivated by these facts, we equip the Siamese networks with R-FCN so that different regions features are extracted. As illustrate in Fig.2, we build a set of position-sensitive score maps by using a bank of specialized convolutional layers correlate with different parts of the template features. Each of these score maps encodes the position information with respect to a relative spatial position (e.g., left of an object). Then combining these score maps through adaptive weights donates the final score map for each candidate sub-window. Unlike traditional part-based trackers[6, 7], all learnable layers are shared on the entire image without repetitions. Moreover the entire architecture is learned end-to-end.

In summary, with our proposed region-based fully convolutional Siamese architecture and adaptive weights, our method can track an arbitrary object in video sequences in real time even when it is occluded or deformed.

2. PROPOSED ALGORITHM

2.1. Region-based Siamese Network

Arbitrary object tracking can be treated as similarity measurement between target image z and candidate image x of the same size by a function $f(x, z)$. We apply a CNN, namely fully-convolutional Siamese network as show in Fig.2, as the function f due to its widespread success in computer vision. To achieve this, we use an embedding function φ representing the feature maps of fully convolutional layer and combine them using a cross-correlation layer

$$f(x, z) = \varphi(x) * \varphi(z) + b\mathbf{1} \tag{1}$$

where $b\mathbf{1}$ denotes a signal which takes value $b \subset \mathbb{R}$ in every evaluation. The output of this network is not a single score but rather a score map defined on a finite grid $\mathcal{D} \subset \mathbb{Z}^2$ as illustrate in Fig.2. During tracking, we pick a search image centered at the previous position of the target. The position of the maximum score relative to the center of the score map is multiplied by the stride of the network, which then gives the displacement of the target from frame to frame.

Apart from traditional fully-convolutional layers, R-FCN[18] constructs a set of position-sensitive score maps by using a bank of specialized convolutional layers as the FCN output for object detection. This could be beneficial to visual tracking since position-sensitive information can still provide reliable cues of regions even when the target is

partially occluded or deformed. We build $k \times k$ position-sensitive convolutional layers on top of the feature maps so that former layers are shared on the entire image and the framework doesn't introduce much extra computation as part-based trackers[6, 7]. As illustrate in Fig.2, the cross-correlation between each of the $k \times k$ feature maps and the correspondent region of the target's feature map are calculated. Then k^2 score maps describe similarity of the relative positions between all sub-windows of search image and target image. For instance, with $k \times k = 2 \times 2$, the 4 score maps encode the similarity of the spatial positions {*top-left, top-right, bottom-left, bottom-right*} between target and candidates.

The parameter k will obviously influence the performance of our tracker. With a small value, the extracted position-sensitive information is too general to deal with occlusion and deformation, e.g., 2×1 (top-bottom, *Accuracy: 0.54, Speed: 58FPS*). On the contrary, with a big value, it brings extra fluctuation and computation, e.g., 4×4 (*Accuracy: 0.48, Speed: 36FPS*). Finally, we choose $k \times k = 2 \times 2$ (*Accuracy: 0.58, Speed: 51FPS*) in this paper to keep a balance between accuracy and efficiency.

2.2. Adaptive weights for position-sensitive score maps

Given the score map of each position, combining them becomes another important step to finally locate the target. In different frames, different regions of targets may undergo different appearance changes, illumination variation, occlusion and deformation. If we simply combine these score maps with the same weight, the response of the falsely tracked regions may be unfairly emphasized. By adaptively weighting each region response, the joint confidence map puts more emphasis on reliable regions and eliminates the clutters caused by drifting regions. In part-based tracker[6], the peak-to-sidelobe ratio (PSR)(Eq.4) is used to quantify the sharpness of the correlation peak. In order to expand the contribution of the more confidential regions, we take the exponential of PSR as the weight of each position-sensitive score map in Eq.3.

The joint score map at *t-th* frame is finally defined as:

$$S^t = \sum_{i=1}^{k^2} w_i^t \cdot \hat{f}_{r(i)}^t \tag{2}$$

where $\hat{f}_{r(i)}^t$ is the score map (Eq.1) of the *i-th* region at time *t*. $r(i)$ donates the relative position of the regions response in the joint score map S^t; k^2 is the number of the regions used to divide the target. w_i^t is the weight parameter of the correspondent region.

$$w_i^t = \frac{e^{PSR_i^t}}{\sum_{j=1}^{k^2} e^{PSR_j^t}} \tag{3}$$

$$PSR_i^t = \frac{\max(\hat{f}_{r(i)}^t) - \mu_i^t}{\sigma_i^t} \tag{4}$$

where μ_i^t and σ_i^t are the mean and standard deviation of the *i-th* position-score map at time *t* respectively. As shown in Fig.2, the *bottom-left, bottom-right* regions which are occluded have lower value of PSR, therefore, have smaller weights to the joint score map.

2.3. Implementation details

We adopt the network of AlexNet[12] as the basis of the embedding function φ. The last fully connected layer is removed and ReLU non-linearities is used as the activation function. During training, batch normalization follows every linear layer. No padding is applied in the network to avoid violating the fully convolutional property. The stride of the final representation (*conv5*) is eight.

Training. We train the network on positive and negative pairs by using the logistic loss ℓ. Our network produces a map of the score $v : \mathcal{D} \to \mathbb{R}$ for each target-search image pair. The final loss of the score map is the mean of the individual losses

$$L(y, v) = \frac{1}{|\mathcal{D}|} \sum_{u \in \mathcal{D}} \ell(y[u], v[u]) \tag{5}$$

requiring the true label $y[u] \in \{+1, -1\}$ and the predicted label $v[u] \in \{+1, -1\}$ for each position $u \in \mathcal{D}$ in the score map. The parameters of the networks are obtained by applying Stochastic Gradient Descent (SGD) to minimize the loss function Eq.5. The elements of the score map are considered to belong to positive examples if they are within radius R of the ground-truth centre. We used the ImageNet Video dataset[20] from the 2015 ImageNet Large Scale Visual Recognition Challenge(ILSVRC) which consists almost 4500 videos and more than one million annotated frames. Our deep model can be trained without over-fitting not only for its vast size, but also various objects and scenes it involves. We adopt exemplar images that are 127×127 and search images that are 255×255. A margin for context on the bounding box is also added so that the network is robust to noise. For the experiments of this paper, more than 2 millions labeled bounding-boxes of all videos are covered for training.

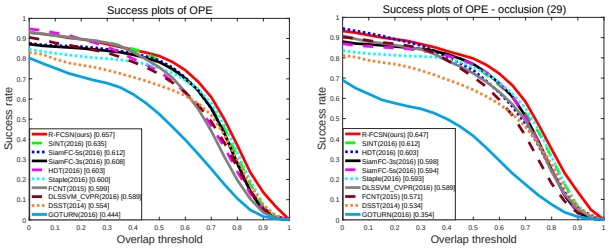
Tracking. Thanks to the great capacity of this network, we use an extremely simplistic algorithm to perform tracking. Unlike more sophisticated trackers, we do not use skills such as model updating, integrating cues of optical flow or colour histogram, refining the predicated bounding box, which may enhance the performance of tracker. Yet, despite its simplicity, our tracking algorithm produces surprisingly good results. During the online tracking, we use the first bounding box as target and search for it within a region of approximately four times its previous size, and a cosine window is added to score map to penalize large displacements. To cope with the size change of the target, we process 3 scaled versions of the search image.

3. EXPERIMENTS

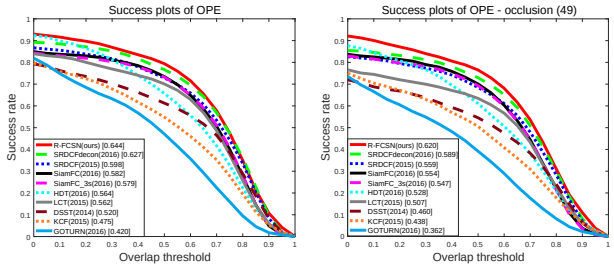
Our algorithm is implemented in MATLAB using MatConvNet library. The average speed is approximately 51FPS using a single NVIDIA GeForce GTX Titan X. To evaluate the proposed tracker, we perform experiments on two public benchmarks: OTB[21, 11], and VOT2015[17].

3.1. The OTB benchmark

We evaluate our algorithm by following the evaluation protocol of benchmark[21, 11]. The success plots is generated by the rates of the successfully tracked frames at many different thresholds in bounding box overlap ratio. The ranks of trackers are determined by the accuracy at 20 pixels threshold Area Under Curve (AUC) score in the success plot.



(a) OTB-2013[21]



(b) OTB-2015[11]

Fig. 3. Quantitative results on OTB benchmark (left) and on sequences with attribute of occlusion (right).

The proposed algorithm is compared with nine state-of-the-art trackers including Siamese-network based trackers SiamFC[8], fully convolutional based trackers FCNT[22], real-time tracker GOTURN[10], SRDCFdecon[23], DSST[24], Staple[9], LCT[25], KCF[26], etc. The results on the two OTB datasets in Fig.3 show that R-FCSN outperforms all other trackers in the success precision plot of whole sequences, especially for occlusion attribute. Compared with other two real-time tracking method, SiamFC, GOTURN, our method has promoted AUC by 12% and 80% respectively. The better accuracy of our method implies that our position-sensitive score maps and adaptive weights are helpful to dealing with various challenges.

3.2. The VOT2015 benchmark

We also tested the proposed tracker in the recent dataset VOT2015[17]. The performance metrics are defined based on accuracy and robustness, which are computed with the bounding box overlap ratio and the number of the tracking failures. The expected average overlap which estimates the accuracy of the estimated bounding box is used to rank tracking algorithms.

Table 1. The accuracy (A), robustness (R), expected average overlap (EAO) and speed on the experiments in VOT2015. The results of trackers marked with * are from benchmark[17].

Trackers	A	R	EAO	Speed(FPS)
MDNet*	0.60	0.69	0.38	0.87
R-FCSN(ours)	**0.58**	**0.64**	**0.36**	**50.25**
DeepSRDCF*	0.56	1.05	0.32	0.38
EBT*	0.47	1.02	0.31	1.76
SiamFC-3s[8]	0.53	1.83	0.29	86
LDP*	0.51	1.84	0.28	4.36
sPST*	0.55	1.48	0.28	1.01
SiamFC[8]	0.52	1.29	0.27	58
SC-EBT*	0.55	1.86	0.25	0.80
NSAMF*	0.53	1.29	0.25	5.47
Struck*	0.47	1.61	0.25	2.44
RAJSSC*	0.57	1.63	0.24	2.12
S3Tracker*	0.52	1.77	0.24	14.27
SumShift*	0.52	1.68	0.23	16.78
SODLT*	0.56	1.78	0.23	0.83

Table 1 illustrates the strength of R-FCSN in VOT2015 dataset compared to other trackers. R-FCSN outperforms all the compared algorithms in *Robustness* while providing comparable results in other metrics of evaluation. MDNet[1] which tracks objects into different branches of CNNs layers according to their domains has achieved optimal result in terms of *Accuracy* and *EAO*. However, R-FCSN is about 58 times faster than MDNet due to its simple architecture, which makes it easier to put into practice. The design of the position-sensitive score maps leads to better results than other Siamese networks methods, i.e., SiamFC[8], while keeping the advantage in speed.

4. CONCLUSION

In this paper, we developed a region-based fully convolutional Siamese-network which enables online real-time visual tracking. Our algorithm accomplished outstanding performance in two large public tracking benchmark, OTB and VOT2015, compared to the state-of-the-art tracking algorithms. By using the position-sensitive score maps and adaptive weights, our tracker exhibits better robustness to occlusion, deformation and appearances changes.

5. REFERENCES

[1] Hyeonseob Nam and Bohyung Han, "Learning multi-domain convolutional neural networks for visual tracking," *arXiv preprint arXiv:1510.07945*, 2015.

[2] David A. Ross, Jongwoo Lim, Ruei Sung Lin, and Ming Hsuan Yang, "Incremental learning for robust visual tracking," *IJCV*, vol. 77, no. 1, pp. 125–141, 2008.

[3] N. Ahuja, Si Liu, B. Ghanem, and Tianzhu Zhang, "Robust visual tracking via multi-task sparse learning," in *CVPR*, 2012, pp. 2042–2049.

[4] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, "Tracking-learning-detection," *TPAMI*, vol. 34, no. 7, pp. 1409–22, 2012.

[5] Sam Hare, Amir Saffari, and Philip H. S. Torr, "Struck: Structured output tracking with kernels," *TPAMI*, vol. 38, no. 10, pp. 263–270, 2016.

[6] Ting Liu, Gang Wang, and Qingxiong Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *CVPR*, 2015, pp. 4902–4912.

[7] Rui Yao, Qinfeng Shi, Chunhua Shen, and Yanning Zhang, "Part-based visual tracking with online latent structural learning," in *CVPR*, 2013, pp. 2363–2370.

[8] Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip Torr, "Fully-convolutional siamese networks for object tracking," *arXiv preprint arXiv:1606.09549*, 2016.

[9] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip Torr, "Staple: Complementary learners for real-time tracking," *Computer Science*, vol. 38, no. 2, pp. 311C323, 2016.

[10] David Held, Sebastian Thrun, and Silvio Savarese, "Learning to track at 100 fps with deep regression networks," in *ECCV*, 2016.

[11] Yi Wu, Jongwoo Lim, and Ming Hsuan Yang, "Object tracking benchmark," *TPAMI*, vol. 37, no. 9, pp. 1–1, 2015.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, no. 2, pp. 2012, 2012.

[13] S. Ren, K. He, R Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks.," *TPAMI*, pp. 1–1, 2016.

[14] Pedro O. Pinheiro, Ronan Collobert, and Piotr Dollar, "Learning to segment object candidates," *Computer Science*, 2015.

[15] Hyeonseob Nam, Mooyeol Baek, and Bohyung Han, "Modeling and propagating cnns in a tree structure for visual tracking," *arXiv preprint arXiv:1608.07242*, 2016.

[16] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *ECCV*, 2016.

[17] Matej Kristan, Jiri Matas, Ale Leonardis, Michael Felsberg, Luka Cehovin, Gustavo Fernandez, Toma Vojir, Gustav Hager, Georg Nebehay, and Roman Pflugfelder, "The visual object tracking vot2015 challenge results," in *ICCV*, 2016, pp. 564–586.

[18] Dai Jifeng, Li Yi, He Kaiming, and Sun Jian, "R-FCN: Object detection via region-based fully convolutional networks," *arXiv preprint arXiv:1605.06409*, 2016.

[19] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei, "Fully convolutional instance-aware semantic segmentation," *arXiv preprint arXiv:1611.07709*, 2016.

[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein, "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[21] Yi Wu, Jongwoo Lim, and Ming Hsuan Yang, "Online object tracking: A benchmark," in *CVPR*, 2013, pp. 2411–2418.

[22] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu, "Visual tracking with fully convolutional networks," in *ICCV*, 2016, pp. 3119–3127.

[23] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *CVPR*, 2016.

[24] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg, "Accurate scale estimation for robust visual tracking," in *BMVC*, 2014.

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

[26] Rui Caseiro, Pedro Martins, and Jorge Batista, "High-speed tracking with kernelized correlation filters," *TPAMI*, 2015.