IEICE TRANS. INF. & SYST., VOL.E102–D, NO.3 MARCH 2019

LETTER

Faster-ADNet for Visual Tracking*

Tiansa ZHANG[†,††], ***Nonmember***, **Chunlei HUO**[††], ***Member***, **Zhiqiang ZHOU**[†a)], ***and*** **Bo WANG**[†], ***Nonmembers***

SUMMARY By taking advantages of deep learning and reinforcement learning, ADNet (Action Decision Network) outperforms other approaches. However, its speed and performance are still limited by factors such as unreliable confidence score estimation and redundant historical actions. To address the above limitations, a faster and more accurate approach named Faster-ADNet is proposed in this paper. By optimizing the tracking process via a status re-identification network, the proposed approach is more efficient and 6 times faster than ADNet. At the same time, the accuracy and stability are enhanced by historical actions removal. Experiments demonstrate the advantages of Faster-ADNet.

key words: *visual tracking, deep learning, status re-identification*

1. Introduction

The aim of visual tracking is to track the target robustly and fast. In recent years, a variety of new approaches [1]–[6] have been proposed. Hong [1] utilized CNN features and an online SVM to distinguish the target and background. Danelljan [2] improved the accuracy by adaptively decontaminating the training set. Fan [3] suggested verifying the results at intervals for balancing tracking performance and efficiency. Qi [4] constructed a stronger tracker for precise target localization by combining weak trackers at different CNN layers. Nam [5] proposed the multi-domain structure to extract more discriminative features. Despite the improved accuracy, the above approaches are limited in the efficiency since they ignored adjusting the tracking strategy to different tracking difficulties at different frames. On the basis of [5], ADNet [6] adopted the reinforcement learning and improved searching strategy for improving the tracking speed (i.e., 2.9 fps). However, ADNet is still far from meeting the real-time requirement, and it is urgent to develop efficient tracking strategy.

As illustrated in Fig. 1, ADNet is overloaded by frequent re-detection, online fine-tuning and samples generation. Firstly, in testing, ADNet is impacted by the time-consuming target re-detection and network fine-tuning caused by the unreliable estimation of confidence score $S1$. In fact, $S1$ evaluates the input image patch cropped at the bounding box of the previous state, which has no strong relation to the success of the predicted result. And $S1$ fails to capture the appearance changes without frequent online fine-tuning. For this reason, the confidence score $S1$ will be unreliable if the target appearance varies across frames even if tracking results are correct, and it will take ADNet many time for frequent re-detection and online fine-tuning. Secondly, in visual tracking, only the target appearance at the first frame is available. To address this problem, at every frame in testing, a large number of samples are generated by ADNet to collect the target appearances. However, many repetitive appearances are collected by generating samples too frequently since the appearance change is small at most time, which is inefficient. Samples generation is the most time-consuming step (10 times slower than the prediction step). In addition, online fine-tuning based on duplicated samples will cause the network overfitting. With 3% decrease in accuracy, ADNet-fast [6] generates a small number of samples per frame, which can not solve the problem fundamentally.

To address the above limitations, a new approach named **Faster-ADNet** is proposed. Compared with traditional methods, our algorithm is faster and more accurate. Specifically, Faster-ADNet is 6 times faster than ADNet with higher accuracy.

2. The Proposed Approach

As shown in Fig. 2, the main rationale of Faster-ADNet is to optimize the testing procedure by reliable status re-identification and efficient hierarchical decision. Specifically, a status re-identification network is added to provide a reliable confidence for the latter decision, and a hierarchical decision procedure is presented to reduce redundant operators and accelerate the tracking speed. Below, we elaborate our improvements in detail.

2.1 Status Re-Identification Network

To evaluate the predicted result fast and reliably, the status re-identification network is implemented by Siamese network [7], which is trained using the multi-domain strategy [5] with tracking datasets [8]–[10]. The input of Siamese network is the image pair, i.e., the intra-class pair and the inter-class pair. The intra-class pair is the image pair

Manuscript received October 12, 2018.
Manuscript revised November 28, 2018.
Manuscript publicized December 12, 2018.

[†]The authors are with School of Automation, Beijing Institute of Technology, China.
[††]The authors are with National Lab. Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China.

*This work was supported by Beijing Natural Science Foundation under Grants No. L172053.

a) E-mail: zhzhzhou@bit.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2018EDL8214

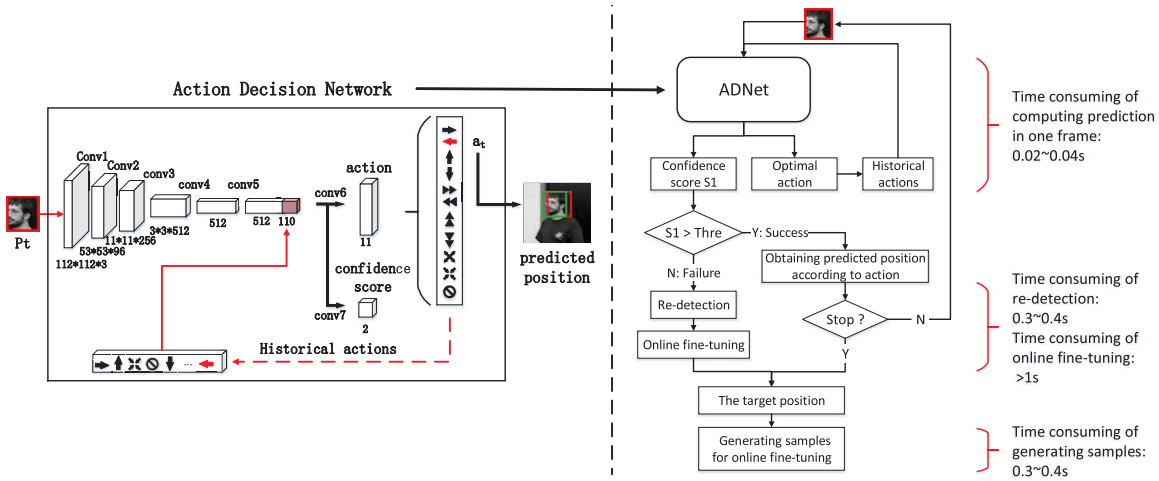

Fig. 1 Network structure, flow diagram and time consuming of ADNet [6]. Generating samples is performed in every frame. Re-detection and online fine-tuning are performed when tracking fails. ADNet captures the target by sequential actions. Re-detection is done by generating N_{re} (= 256) target position candidates and choosing the one with the highest confidence score. The stop condition is reached by selecting the stop action or falling in the oscillation case (i.e., the sequential actions are obtained as $\{right, left, right\}$).

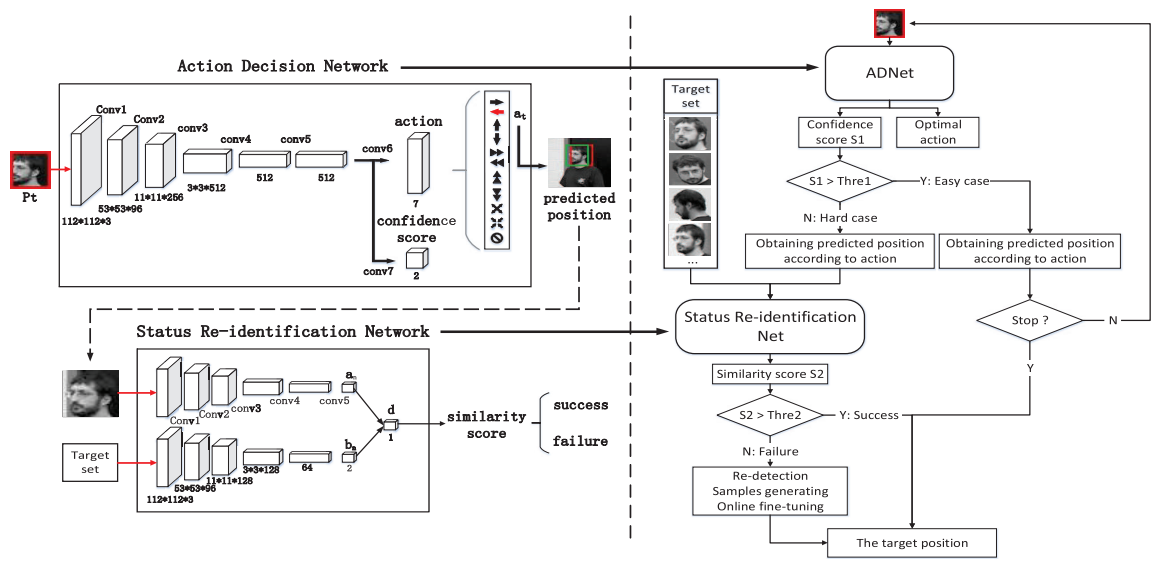

Fig. 2 Network structure and flow diagram of Faster-ADNet. The role of status re-identification network is utilizing the similarity score to determine the latter re-detection, samples generation and online fine-tuning operations.

with the same semantic label (i.e., both are the backgrounds or the target), and the inter-class pair is the image pair with different semantic labels. Two images x_i and z_i can be expressed as (x_i, z_i, y_i), where $y_i = 1$ denotes the intra-class pair, and $y_i = 0$ denotes the inter-class pair. Compared with individual images, the Siamese network is more promising for capturing the intra-class similarities and the inter-class differences. In detail, the appearances of the same target under different perspectives, x_i and z_i, may have a large distance in the image feature space, but they should have a smaller distance than the inter-class pair after training since they have the same semantical label. In this context, the Siamese network can reliably decide whether the predicted result is the target.

Siamese network achieved the above goals by the following contrastive loss [7],

$$L = \frac{1}{2N}\sum_{i=1}^{N} y d_i^2 + (1 - y_i) max(m - d_i, 0)^2 \quad (1)$$

Where $d_i = \|a_i - b_i\|_2$ denotes the Euclidean distance of two image features a_i and b_i, $a_i = P(x_i)$ and $b_i = P(z_i)$ are the outputs at the final layer, m is the threshold and we set it to 1. The image pairs are obtained by cropping patches around the ground truth. These patches are divided into target and background categories according to the IOU between the

patch and the ground truth (i.e. the patch with IOU > 0.7 belongs to the target class, otherwise the background class).

In testing, for the predicted result x_j and the patch z_j from the target set, the similarity score is obtained as,

$$S2 = m - d_j = m - \left\| P(x_j) - P(z_j) \right\|_2 \quad (2)$$

There may be more than one patch in the target set, and the highest score is used as the similarity score. Another advantage of Siamese network is when the target appearance changes, only the inputs need updating without the demand of fine-tuning the network, which saves a lot of time in tracking. Each re-identification operator only takes less than 0.01 seconds.

2.2 Faster Testing Based on Hierarchical Decision

As shown in Fig. 2, hierarchical decision procedure bases on the confidence score $S1$ and the similarity score $S2$, which can reduce unnecessary calculations in testing. For easy frames, the predicted position will be obtained according to the action directly. For hard frames, the predicted position will be re-identified with Siamese network. The time-consuming steps (e.g., re-detection, samples generation and online fine-tuning) are only required when the result of re-identification is failed.

Different target appearances are stored in the target set for identifying the predicted result. At first, only the ground truth was stored. In tracking, the target set is updated adaptively. The representative sample sampled at every N_l (= 25) frames will be added if the sample number is less than N_s (= 8) and no overly similar samples have been stored. The bad sample will be deleted, which has the most similar appearance with the current failing position or produces the failure tracking but with continuous low confidence score $S1$ and high similarity score $S2$. The profit of the hierarchical decision strategy is that the following operators were reduced, 85% of generated samples, 40% of re-detection, 80% of online fine-tuning and the speed is improved by 6 times.

Another improvement is the removal of redundant historical actions. In fact, in many cases such as the *Human2* sequences [11], the target is in the view of a moving camera. Since the camera does not move linearly and uniformly, the historical actions are stochastic. As shown in Fig. 3, the tracker will be disturbed and make the wrong decision due to the hysteretic historical actions. For this reason, the historical actions are not utilized.

3. Experiments

We evaluated our method on the Object Tracking Benchmark (OTB) [11], [12]. To demonstrate the effectiveness of the proposed approach, the following 6 state-of-the-art trackers are used for comparison: HDT [4], SRDCFdecon [2], CNN-SVM [1], DeepSRDCF [13], ADNet [6] and ADNet-fast [6]

We used the same training datasets as ADNet [6] from

Fig. 3 Illustration of historical actions disturbance on *Human2* [11], the target has been moving to the left, but the camera starts moving since the 90th frame, so the person moves quickly to the right.

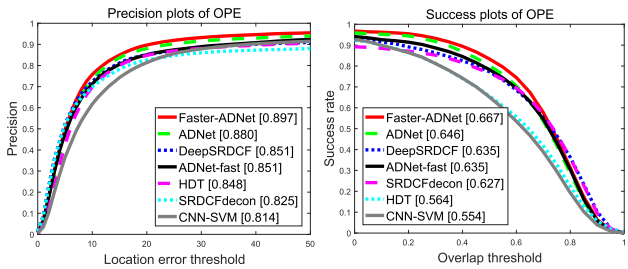

Fig. 4 Performance comparison of different trackers on OTB-100.

Table 1 Performance comparison on OTB-50 [12].

Algorithm	Prec (20px)	AUC	FPS	GPU
Faster-ADNet	0.924	0.687	16.7	O
ADNet	0.903	0.659	2.9	O
ADNet-fast	0.898	0.670	15.0	O
HDT	0.889	0.603	5.8	O
SRDCFdecon	0.870	0.653	1.7	X
CNN-SVM	0.852	0.579	< 1	O
DeepSRDCF	0.849	0.641	< 1	O

VOT2013, 2014, 2015 [8]–[10] and ALOV300 [14]. The action decision network and status re-identification network are fine-tuned T (= 300) iterations at the initial frame. In the online adaptation, N_1 (= 3000) samples are obtained in the initial frame, and N_2 (= 250) samples are obtained when tracking fails. The tracking performance was measured based on the following metrics: FPS, overlap ratio and center location error [12]. Performances of different approaches are shown in Fig. 4 and Table 1. The advantages of Faster-ADNet can be validated by visually comparing precisions and success rates of different methods.

To understand how our modifications work, Faster-ADNet is compared with two variants by considering ADNet as the baseline:

1) **ADNet+re-identification**. "ADNet+re-identification" aims to improve ADNet by adding status re-identification in testing, which achieves the speed of 16 fps, and it does not remove the historical actions.

2) **ADNet-HA**. "ADNet-HA" aims to improved ADNet by removing historical actions, and it does not have the status re-identification procedure.

Performances of various variants are shown in Fig. 5. Owe to the status re-identification network, the speed of "ADNet+re-identification" can reach 16 fps and precision is also improved by 0.7%. "ADNet-HA" achieves 1.7% im-

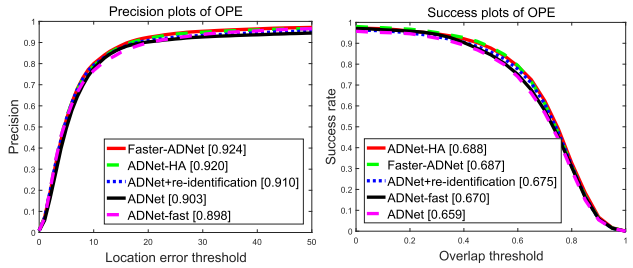

Fig. 5 Performance comparison of different variants on OTB-50.

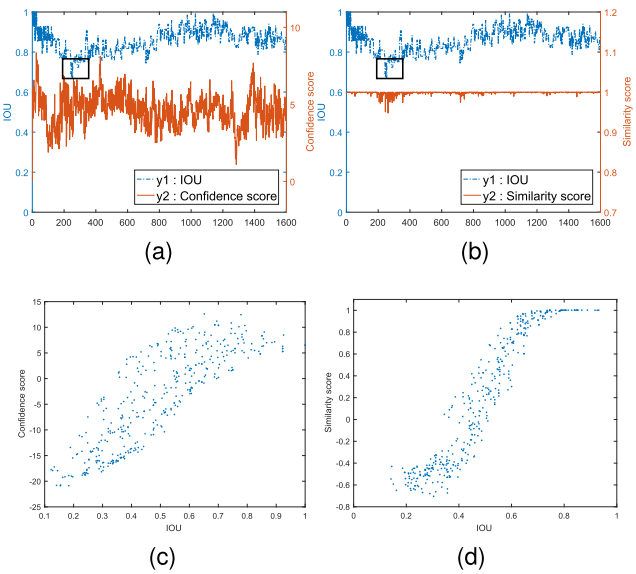

Fig. 6 Comparison of confidence score and similarity score with respect to IOU on *Car24* [12]. (a) and (c): IOU and confidence score. (b) and (d): IOU and similarity score.

provement, which is the new model without historical actions in both training and testing. The performance differences illustrate the effectiveness of status re-identification and historical actions removal. By combing the above two modifications, the precision of Faster-ADNet is 2.1% higher than ADNet, and the speed is 6 times faster. To validate the effectiveness of status re-identification network, the confidence score and the similarity score are compared with respect to IOU in Fig. 6. In (a) and (b), each predicted position was evaluated and obtained the confidence score and the similarity score during testing on *Car24* [12]. Results show that the similarity score is more stable than the confidence score while the IOU remains quite well in testing, which means the status re-identification process can reduce many misjudgments. When the tracking result is poor, the similarity score can also accurately identify it (e.g., the black box in (a) and (b)). In (c) and (d), each point represents a sample generated around the target while testing on *Car24* [12]. The X axis represents the IOU between the sample and the ground truth, the Y axis represents the confidence score or the similarity score of the sample. The distribution between IOU and confidence score is much more scattered. In short, the similarity score is more discriminative and robust for evaluating the tracking status.

4. Conclusion

In this paper, Faster-ADNet is proposed, which reduces tracking status mis-classification by status re-identification network and avoid time-consuming processes by hierarchical decision and redundant historical action removal. Compared with ADNet, the proposed approach is 6 times faster and more accurate.

References

[1] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," International Conference on Machine Learning, pp.597–606, 2015.
[2] M. Danelljan, G. Häger, F.S. Khan, and M. Felsberg, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp.1430–1438, 2016.
[3] H. Fan and H. Ling, "Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking," Proc. IEEE Int. Conf. Computer Vision, Venice, Italy, pp.5487–5495, 2017.
[4] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, "Hedged deep tracking," 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp.4303–4311, 2016.
[5] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp.4293–4302, 2016.
[6] S. Yun, J. Choi, Y. Yoo, K. Yun, and J.Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," 2017 IEEE Conference on Computer Vision and Pattern Recognition, pp.1349–1358, 2017.
[7] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.1735–1742, 2006.
[8] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Cehovin, G. Nebehay, G. Fernandez, T. Vojir, et al., "The visual object tracking VOT2013 challenge results," 2013 IEEE International Conference on Computer Vision Workshops, pp.98–111, 2013.
[9] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Čehovin, G. Nebehay, T. Vojíř, G. Fernández, A. Lukežič, et al., "The visual object tracking VOT2014 challenge results," International Conference on Computer Vision, pp.191–217, 2014.
[10] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, R. Pflugfelder, A. Gupta, A. Bibi, A. Lukezic, A. Garcia-Martin, A. Saffari, A. Petrosino, and A.S. Montero, "The visual object tracking VOT2015 challenge results," International Conference on Computer Vision Workshop, pp.564–586, 2015.
[11] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," IEEE Trans. Pattern Anal. Mach. Intell., vol.37, no.9, pp.1834–1848, 2015.
[12] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp.2411–2418, 2013.
[13] M. Danelljan, G. Häger, F.S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," International Conference on Computer Vision, pp.621–629, 2016.
[14] A.W.M. Smeulders, D.M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," IEEE Trans. Pattern Anal. Mach. Intell., vol.36, no.7, pp.1442–1468, 2014.