Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning

Sangdoo Yun[1] Jongwon Choi[1] Youngjoon Yoo[2] Kimin Yun[3] and Jin Young Choi[1]
[1]ASRI, Dept. of Electrical and Computer Eng., Seoul National University, South Korea
[2]Graduate School of Convergence Science and Technology, Seoul National University, South Korea
[3]Electronics and Telecommunications Research Institute (ETRI), South Korea
{yunsd101, i0you200, jychoi}@snu.ac.kr, jwchoi.pil@gmail.com, kimin.yun@etri.re.kr

A. Additional Experimental Results

We evaluated the proposed method on OTB-50 [14] dataset for various attributes with state-of-the-art trackers including MDNet [9], C-COT [4], GOTURN [5], HDT [10], DeepSRDCF [3], SINT [11], FCNT [12], SCT [1], MUSTer [8], CNN-SVM [7], MEEM [15], DSST [2], and KCF [6]. The detailed results of the proposed method are illustrated in Figure 1, Figure 2, and Figure 3.

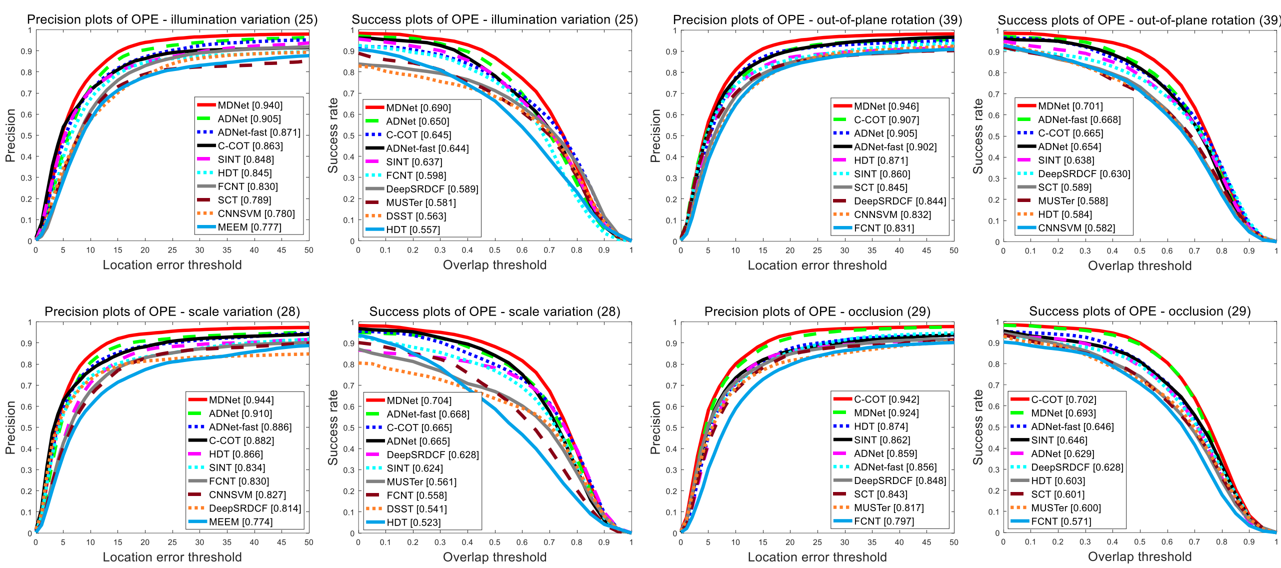

Figure 1. **Precision and success plots on OTB-50** for the subset of *illumination variation, out-of-plane rotation, scale variance, and occlusion*. The plotted curves are based on the center location error threshold and the overlap threshold. The scores in the legend indicate the mean precisions when the location error threshold is 20 pixels for the precision plots and area-under-curve (AUC) for the success plots. Only the top 10 trackers are presented.

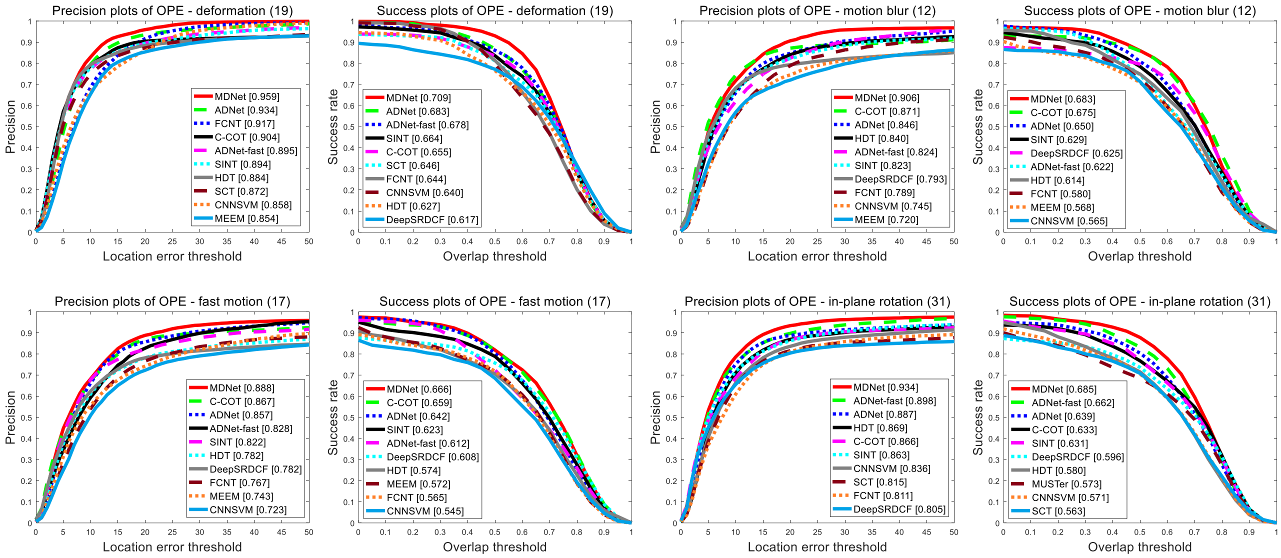

Figure 2. **Precision and success plots on OTB-50** for the subset of *deformation, motion blur, fast motion, and in-plane rotation*. The plotted curves are based on the center location error threshold and the overlap threshold. The scores in the legend indicate the mean precisions when the location error threshold is 20 pixels for the precision plots and area-under-curve (AUC) for the success plots. Only the top 10 trackers are presented.

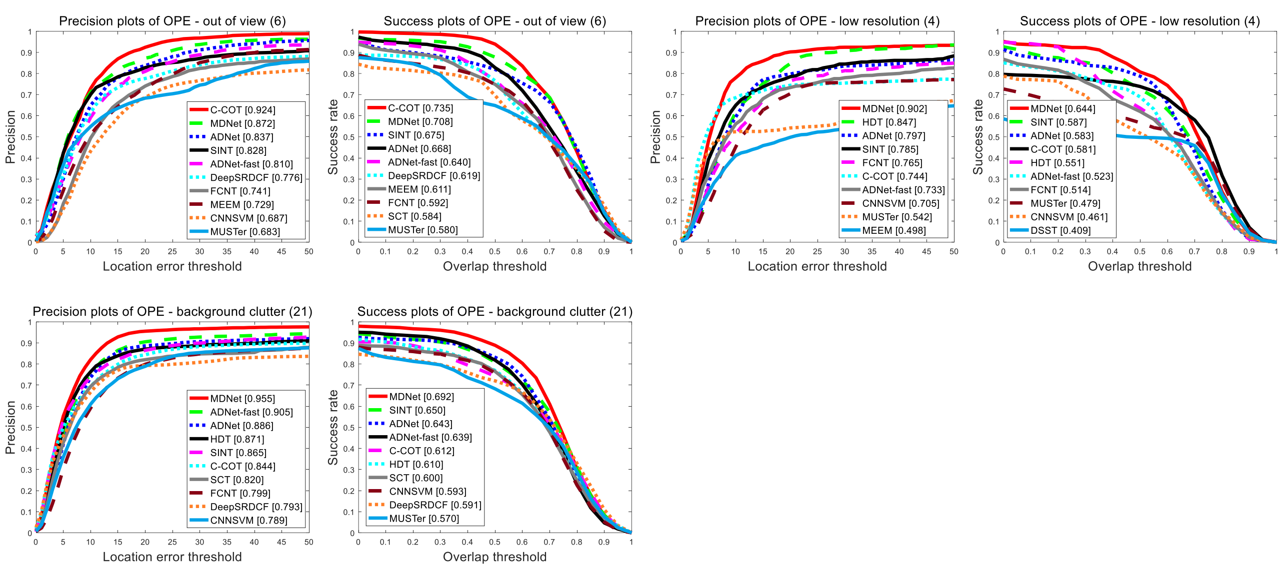

Figure 3. **Precision and success plots on OTB-50** for the subset of *out of view, low resolution, and background clutter*. The plotted curves are based on the center location error threshold and the overlap threshold. The scores in the legend indicate the mean precisions when the location error threshold is 20 pixels for the precision plots and area-under-curve (AUC) for the success plots. Only the top 10 trackers are presented.

B. Algorithms of Training ADNet

B.1. Training ADNet with Reinforcement Learning (Section 4.2)

The detailed algorithm to train ADNet with reinforcement learning is described in Algorithm 1. The initial network parameter W_{SL} is the same as W_{SL} (line 1). In the training iteration, we randomly select a piece of training sequences $\{F_l\}_{l=1}^{\mathcal{L}}$ and the ground truth $\{G_l\}_{l=1}^{\mathcal{L}}$ of length $\mathcal{L}$ from the training data (line 3). Then, the tracking simulation is performed using TRACKING_PROCEDURE (line 4-9). In frame l, the TRACKING_PROCEDURE iteratively pursues the position of the target by selecting actions and updating the states from the initial state $b_{T_{l-1},l-1}$ and $d_{T_{l-1},l-1}$ as shown in Algorithm 2. In line 10, the tracking scores $z_{t,l}$ for $t = 1, ..., T_l$ and $l = 1, ..., \mathcal{L}$ are computed using $b_{t,l}$ and $\{G_l\}_{l=1}^{\mathcal{L}}$. Using the tracking scores, the network parameter W_{RL} is updated by REINFORCE method [13] (line 11-12).

Algorithm 1 Training ADNet with reinforcement learning (RL).

Input: Pre-trained ADNet (W_{SL}), Training sequences $\{F_l\}$ and ground truths $\{G_l\}$
Output: Trained ADNet weights W_{RL}

1: Initialize W_{RL} with W_{SL}
2: **repeat**
3: Randomly select $\{F_l\}_{l=1}^{\mathcal{L}}$ and $\{G_l\}_{l=1}^{\mathcal{L}}$
4: Set initial $b_{1,1} \leftarrow G_1$
5: Set initial $d_{1,1}$ as zero vector
6: $T_1 \leftarrow 1$
7: **for** $l = 2$ to $\mathcal{L}$ **do**
8: $\{a_{t,l}\}, \{b_{t,l}\}, \{d_{t,l}\}, T_l \leftarrow$ TRACKING_PROCEDURE$(b_{T_{l-1},l-1}, d_{T_{l-1},l-1}, F_l)$ in Algorithm 2
9: **end for**
10: Compute tracking scores $\{z_{t,l}\}$ with $\{b_{t,l}\}$ and $\{G_l\}_{l=1}^{\mathcal{L}}$
11: $\Delta W_{RL} \propto \sum_{l=1}^{\mathcal{L}} \sum_{t=1}^{T_l} \frac{\partial \log p(a_{t,l}|s_{t,l}; W_{RL})}{\partial W_{RL}} z_{t,l}$ [13]
12: Update W_{RL} using ΔW_{RL}.
13: **until** W_{RL} converges

Algorithm 2 Tracking Procedure of ADNet.

procedure TRACKING_PROCEDURE$(b_{T_{l-1},l-1}, d_{T_{l-1},l-1}, F_l)$
$t \leftarrow 1$
$p_{t,l} \leftarrow \phi(b_{T_{l-1},l-1}, F_l)$
$d_{t,l} \leftarrow d_{T_{l-1},l-1}$
$s_{t,l} \leftarrow (p_{t,l}, d_{t,l})$
repeat
$a_{t,l} \leftarrow arg\max_a p(a|s_{t,l}; W)$
$b_{t+1,l} \leftarrow f_p(b_{t,l}, a_{t,l})$
$p_{t+1,l} \leftarrow \phi(b_{t+1,l}, F_l)$
$d_{t+1,l} \leftarrow f_d(d_{t,l}, a_{t,l})$
$s_{t+1,l} \leftarrow (p_{t+1,l}, d_{t+1,l})$
$t \leftarrow t + 1$
until $s_{t,l}$ is a terminal state
Set termination step $T_l \leftarrow t$
Return $\{a_{t,l}\}, \{b_{t,l}\}, \{d_{t,l}\}, T_l$
end procedure

B.2. Online Adaptation in Tracking (Section 4.3)

The detailed procedure of the proposed tracking and online adaptation is described in Algorithm 3. At the first frame, ADNet is finetuned using the initial samples $\mathbb{S}_1$ (line 1-2). At frame $l(\geq 2)$, the sequential actions and the states are obtained by the TRACKING_PROCEDURE of Algorithm 2 (line 6). Then we check the confidence score $c(s_{T_l,l})$ of the estimated target to determine the success and failure of the current tracking result. If the confidence score is above 0.5, that is, success case, we generate samples $\mathbb{S}_l$ for online adaptation (line 9). If the confidence score is below -0.5, which means the tracker miss the target, we conduct re-detection by generating target candidates (line 11-13). For every $\mathcal{I}$-th frame, the online adaptation is conducted using the samples $\{\mathbb{S}_k\}_{k=l-\mathcal{J}+1}^{l}$ (line 16).

Algorithm 3 Online adaptation of ADNet in tracking.

Input: Trained ADNet (W), Test sequences $\{F_l\}_{l=1}^{L}$ and initial target position $b_{1,1}$
Output: Estimated target positions $\{b_{T_l,l}\}_{l=2}^{L}$
1: Generate samples $\mathbb{S}_1$ from F_1 and G_1
2: Update W using $\mathbb{S}_1$
3: Set initial $d_{1,1}$ as zero vector
4: $T_1 \leftarrow 1$
5: **for** $l = 2$ to L **do**
6: $\quad \{a_{t,l}\}, \{b_{t,l}\}, \{d_{t,l}\}, T_l \leftarrow$ TRACKING_PROCEDURE($b_{T_{l-1},l-1}, d_{T_{l-1},l-1}, F_l$) in Algorithm 2
7: $\quad s_{T_l,l} \leftarrow (\phi(b_{T_l,l}, F_l), d_{T_l,l})$
8: $\quad$ **if** $c(s_{T_l,l}) > 0.5$ **then**
9: $\qquad$ Generate samples $\mathbb{S}_l$ from F_l and $b_{T_l,l}$
10: $\quad$ **else if** $c(s_{T_l,l}) < -0.5$ **then**
11: $\qquad$ Draw $\mathcal{N}_{\text{det}}$ target position candidates $\{\tilde{b}_i\}$
12: $\qquad$ Re-detect the target position $b^* \leftarrow arg\max_{\tilde{b}_i} c(\tilde{b}_i)$
13: $\qquad s_{T_l,l} \leftarrow (\phi(b^*, F_l), d_{T_l,l})$
14: $\quad$ **end if**
15: $\quad$ **if** $\text{mod}(l, \mathcal{I}) = 0$ **then**
16: $\qquad$ Update W using the samples $\{\mathbb{S}_k\}_{k=l-\mathcal{J}+1}^{l}$
17: $\quad$ **end if**
18: **end for**

References

[1] J. Choi, H. Jin Chang, J. Jeong, Y. Demiris, and J. Young Choi. Visual tracking using attention-modulated disintegration and integration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4321–4330, 2016. 1

[2] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014. 1

[3] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Convolutional features for correlation filter based visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 58–66, 2015. 1

[4] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, 2016. 1

[5] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. *arXiv preprint arXiv:1604.01802*, 2016. 1

[6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015. 1

[7] S. Hong, T. You, S. Kwak, and B. Han. Online tracking by learning discriminative saliency map with convolutional neural network. *arXiv preprint arXiv:1502.06796*, 2015. 1

[8] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 749–758, 2015. 1

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

[10] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, and J. L. M.-H. Yang. Hedged deep tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1

[11] R. Tao, E. Gavves, and A. W. Smeulders. Siamese instance search for tracking. *arXiv preprint arXiv:1605.05863*, 2016. 1

[12] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3119–3127, 2015. 1

[13] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992. 3

[14] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2411–2418, 2013. 1

[15] J. Zhang, S. Ma, and S. Sclaroff. Meem: robust tracking via multiple experts using entropy minimization. In *European Conference on Computer Vision*, pages 188–203. Springer, 2014. 1