Neural Networks 129 (2020) 334–343

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Appearance variation adaptation tracker using adversarial network

Mohammadreza Javanmardi *, Xiaojun Qi

Utah State University, Logan, UT, United States

ARTICLE INFO

Article history:
Available online 17 June 2020

Keywords:
Visual tracking
Convolutional neural network
Adversarial learning

ABSTRACT

Visual trackers using deep neural networks have demonstrated favorable performance in object tracking. However, training a deep classification network using overlapped initial target regions may lead an overfitted model. To increase the model generalization, we propose an appearance variation adaptation (AVA) tracker that aligns the feature distributions of target regions over time by learning an adaptation mask in an adversarial network. The proposed adversarial network consists of a generator and a discriminator network that compete with each other over optimizing a discriminator loss in a mini-max optimization problem. Specifically, the discriminator network aims to distinguish recent target regions from earlier ones by minimizing the discriminator loss, while the generator network aims to produce an adaptation mask to maximize the discriminator loss. We incorporate a gradient reverse layer in the adversarial network to solve the aforementioned mini-max optimization in an end-to-end manner. We compare the performance of the proposed AVA tracker with the most recent state-of-the-art trackers by doing extensive experiments on OTB50, OTB100, and VOT2016 tracking benchmarks. Among the compared methods, AVA yields the highest area under curve (AUC) score of 0.712 and the highest average precision score of 0.951 on the OTB50 tracking benchmark. It achieves the second best AUC score of 0.688 and the best precision score of 0.924 on the OTB100 tracking benchmark. AVA also achieves the second best expected average overlap (EAO) score of 0.366, the best failure rate of 0.68, and the second best accuracy of 0.53 on the VOT2016 tracking benchmark.

1. Introduction

Visual tracking aims to estimate states of a moving object in a dynamic frame sequence. Numerous methods have been introduced (Bertinetto, Valmadre, Golodetz, Miksik, & Torr, 2016; Dai, Wang, Lu, Sun, & Li, 2019; Danelljan, Bhat, Shahbaz Khan, & Felsberg, 2017; Danelljan, Robinson, Khan, & Felsberg, 2016; Hare et al., 2016; Henriques, Caseiro, Martins, & Batista, 2015; Zhang, Ma, & Sclaroff, 2014) to track targets under various challenges such as deformation, occlusion, illumination variation, scale variation, and fast motion. However, developing a robust algorithm that can handle different challenges still remains unsolved.

Recently, convolutional neural network (CNN) based trackers (Li et al., 2019; Nam & Han, 2016; Pu, Song, Ma, Zhang, & Yang, 2018; Song et al., 2018; Zhang & Peng, 2019; Zhang, Xu, & Yang, 2018) have shown state-of-the-art performance in terms of accuracy and robustness. They cast tracking as a deep binary classification problem and categorize the candidates into target or background classes. As one of the pioneer works, Wang and Yeung (2013) propose a multi-layer denoising auto-encoder network to learn a generic object representation. Various CNNs-based trackers utilize pretrained neural networks on a large-scale classification dataset (Simonyan & Zisserman) to extract deep features of target candidates. These features are then separately integrated in correlation filter-based trackers and sparse trackers (Henriques et al., 2015; Javanmardi, Farzaneh, & Qi, 2020; Qi et al., 2016; Zhang et al., 2018) to achieve better performance. On the other hand, some tracking methods (Nam & Han, 2016; Song et al., 2018) directly use external videos to pretrained CNNs for the classification purposes. As one of the representative works, Nam and Han (2016) introduced the MDNet tracker, which pretrains a discriminative CNN using auxiliary sequences with tracking ground truths to obtain a generic object representation. Various trackers have then been proposed to improve the performance of MDNet by using a tree structure to manage multiple target appearance models (Nam, Baek, & Han, 2016), using adversarial learning to identify the mask that maintains the most robust features of the target objects over a long temporal span (Song et al., 2018), and using reciprocative learning to exploit visual attention for training deep classifiers (Pu et al., 2018).

A major drawback of CNN-based trackers is lack of temporal generalization of their model, which is caused by over-fitting on the overlapped initial target regions. Therefore, CNN-based trackers such as MDNet (Nam & Han, 2016) fail to maintain the similarities between the discriminative features of targets over time. To increase the generalization of the classification network,

* Corresponding author.
E-mail address: javanmardi@aggiemail.usu.edu (M. Javanmardi).

https://doi.org/10.1016/j.neunet.2020.06.011
0893-6080/

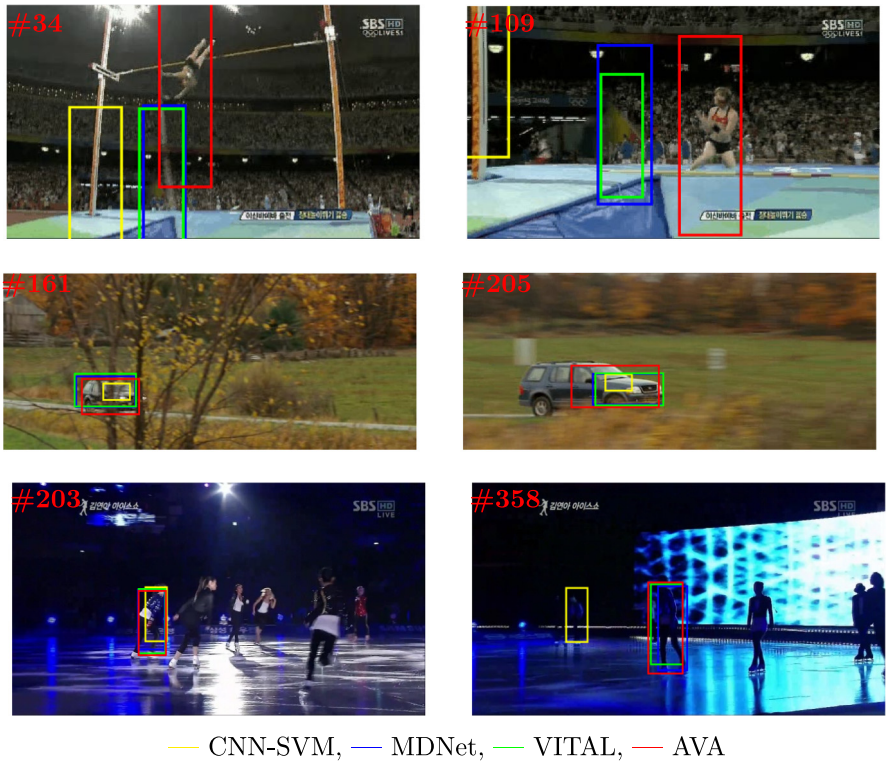


Fig. 1. Comparison of the proposed AVA tracker with CNN-SVM (Hong, You, Kwak, & Han, 2015b), MDNet (Nam & Han, 2016), and VITAL (Song et al., 2018) on three OTB100 sequences including *Jump* (Top), *CarScale* (Middle), and *Skating1* (Bottom).

VITAL tracker (Song et al., 2018) utilizes adversarial learning to generate a mask that dropouts convolutional features of target candidates. In each training iteration, VITAL prepares 9 random masks, where each mask covers one of 9 locations in the 3×3 feature map, to learn the optimal mask in a least square optimization problem. Therefore, this optimal mask is updated to cover only one part of local features in each iteration. This can lead to the loss of the informative local features during training.

In this paper, we propose an appearance variation adaptation (AVA) tracker to not only improve the model generalization but also maintain the informative local features. The AVA tracker aligns the feature distributions of target regions over time by learning an adaptation mask in an adversarial network. This adversarial network works with the classification network to learn robust discriminative features of targets. The proposed adversarial network consists of a generator and a discriminator network that compete with each other over optimizing a discriminator loss in a mini-max optimization problem. Specifically, the discriminator network aims to distinguish recent target regions from earlier ones by minimizing the discriminator loss, while the generator network aims to produce an adaptation mask to maximize the discriminator loss. This leads to alignment of informative features of recent and earlier target regions during tracking, while maintaining accuracy of the classification network to distinguish targets and backgrounds. We incorporate a gradient reverse layer (Ganin & Lempitsky, 2014) in the adversarial network to solve the aforementioned mini-max optimization in an end-to-end manner. Unlike the VITAL tracker, the learned mask in AVA incorporates a weighted combination of multiple parts of target features in each training iteration. The proposed AVA tracker is evaluated on multiple tracking benchmarks (Kristan et al., 2016; Wu, Lim, & Yang, 2013, 2015) and achieves a favorable performance against state-of-art trackers. Sample qualitative results are shown in Fig. 1.

The major contributions of the proposed tracker are:

- Employing adversarial learning to improve the model generalization and learn more robust discriminative features of target regions over a long time span.
- Designing an adversarial network including both generator and discriminator networks that compete with each other to learn an adaptation mask, which aligns feature distributions of target regions that may undergo various changes.
- Incorporating a gradient reverse layer in the adversarial network to solve the mini-max optimization problem in an end-to-end manner.
- Performing extensive experiments on challenging benchmarks to evaluate the performance of the AVA tracker against state-of-the-art trackers.

The remainder of this paper is organized as follows: Section 2 presents the proposed AVA tracker method together with its CNN in detail. Section 3 presents the experimental setup and the results on OTB50 (Wu et al., 2013), OTB100 (Wu et al., 2015), and VOT2016 tracking benchmarks (Kristan et al., 2016). Section 4 draws the conclusion and discuss the future work.

2. Proposed method

In this section, we detail the proposed appearance variation adaptation (AVA) tracker, which aligns the feature distributions of target regions over a long time span by learning an adversarial network (Goodfellow et al., 2014). Fig. 2 shows the network diagram of the proposed AVA tracker. The feature map extracted from convolutional layers is fed to the adversarial network to learn an adaptation mask. The mask highlights the regions with higher similarities with both recent and earlier target regions. These aligned features are passed to the classification network for label prediction (Nam & Han, 2016).

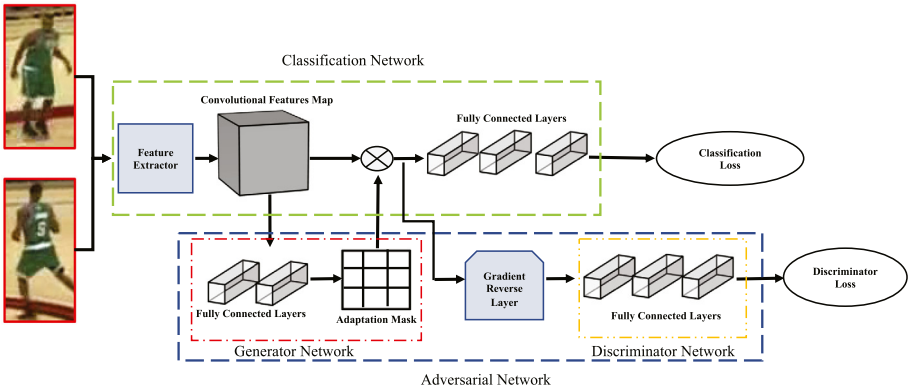


Fig. 2. The architecture of the proposed AVA tracker. The adversarial network consists of two networks: Generator and Discriminator. These two networks work hand-in-hand to learn an adaptation mask in a mini-max optimization problem to align the feature distributions of recent and earlier target regions up to the current frame. The classification network, including the feature extractor and three fully connected layers, is the same as the MDNet tracker (Nam & Han, 2016).

2.1. AVA tracker model

Taking advantage of adversarial learning (Goodfellow et al., 2014; Schmidhuber, 2020) and domain adaptation (Ganin & Lempitsky, 2014), we adversarially learn an adaptation mask to align features of recent and earlier target regions to make the AVA tracker model more generalized. In domain adaptation using an adversarial network (Ganin & Lempitsky, 2014), the training set consists of two domains (subsets) coming from different distributions. Images in one domain have classification labels, while images in the other domain may belong to different classes without labels. Existing domain adaptation methods cast the convolutional layers as a generator network and utilize a discriminator network to adapt the feature distributions of both domains. The same domain adaptation concept cannot be directly adopted in visual tracking due to the following reasons: (1) The training set exclusively contains object candidates and does not include two domains coming from different distributions. (2) The convolutional layers are expensive to learn during online tracking.

In visual tracking, a target may undergo various changes in a frame sequence due to in-plane and out-plane rotations, deformation, partial occlusion, and scale variation. However, its identity remains unchanged even with various appearance changes. To this end, we can safely assume that recent and earlier target regions in a sequence come from two domains with different distributions. Therefore, we propose to utilize adversarial learning, which adapts the feature distributions of recent and earlier target regions to maintain the similarities between the discriminative features of targets over time. Instead of learning convolutional layers directly during online tracking, we propose to learn an adaptation mask in a generator network and apply it to the feature map coming from convolution layers to produce a more robust feature representation of target candidates over time.

Here we formulate the proposed AVA tracker mathematically. Suppose that the training set of target regions up to the current frame is $\{\mathbf{x}_j\}_{j=1}^{N_t}$, where $\mathbf{x}_j \in \mathbb{R}^{1\times m}$ is the convolutional features of the jth target region, N_t is the number of target samples up to the current frame, and $m = w \times h \times d$ with w, h, and d respectively being the width, the height, and the depth of the feature map in the last convolutional layer. Suppose $\mathbf{x}_i$ is a recent target region with a distribution of $\mathcal{S}(x)$ and $\mathbf{x}_k$ is an earlier target region with a distribution of $\mathcal{O}(x)$. The goal of the proposed tracker is to align the feature distributions corresponding to $\mathbf{x}_i$ and $\mathbf{x}_k$ to increase the network generalization. The alignment is performed by an adaptation mask that is adversarially learned in the proposed adversarial network, which consists of a generator network G_f and a discriminator network G_d. It should be emphasized that the adaptation mask, which is included in G_f, responds to the input features since the proposed AVA tracker model seamlessly integrates both G_f and G_d. These two networks (G_f and G_d) compete with each other over the optimization of the proposed discriminator loss $\mathcal{L}_d$ between $\mathbf{x}_i$ and $\mathbf{x}_k$. Specifically, the discriminator network G_d tries to adjust its parameters $\mathbf{w}_d$ to minimize $\mathcal{L}_d$, while the generator network G_f attempts to learn its parameters $\mathbf{w}_f$ to deceive G_d and maximize $\mathcal{L}_d$. The aligned feature map pair $(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_k)$ is then fed to the classification network G_c with its parameters $\mathbf{w}_c$ to minimize the classification loss $\mathcal{L}_c$. Finally, the proposed AVA tracker model is formulated as below:

$$\begin{aligned}\mathcal{L}(\mathbf{w}_c, \mathbf{w}_f, \mathbf{w}_d) &= \mathcal{L}_c(\cdot) - \lambda\mathcal{L}_d(\cdot) \\ &= \sum_{l\in\{1:N\}} \mathcal{L}_c(G_c(G_f(\mathbf{z}_l)), y_l) \\ &\quad - \lambda \sum_{j\in\{1:N_t\}} \mathcal{L}_d(G_d(G_f(\mathbf{x}_j)), v_j) \end{aligned} \tag{1}$$

where $\mathcal{L}_c(\cdot)$ is the cross-entropy classification loss between target and background (Nam & Han, 2016), $\mathcal{L}_d(\cdot)$ is a cross-entropy loss between the feature maps of recent and earlier target regions, and λ is a hyper-parameter to control the balance between $\mathcal{L}_c$ and $\mathcal{L}_d$. In both loss terms, the operator $G(\cdot)$ generates the output feature map of network G. In $\mathcal{L}_c$, $\mathbf{z}_l$ is the feature map of a target or background candidate, N is the total number of target and background samples up to the current frame, and y_l is the class label (e.g., target and background). In $\mathcal{L}_d$, N_t is the number of target samples up to the current frame, and v_j is the binary label of the jth target region defined as follows:

$$v_j = \begin{cases} 0 & x_j \sim \mathcal{S}(x) \\ 1 & x_j \sim \mathcal{O}(x) \end{cases} \tag{2}$$

The optimal parameters $(\hat{\mathbf{w}}_c, \hat{\mathbf{w}}_f, \hat{\mathbf{w}}_d)$ for the proposed model in Eq. (1) are learned in the following mini-max optimization problem (Ganin & Lempitsky, 2014):

$$(\hat{\mathbf{w}}_c, \hat{\mathbf{w}}_f) = \arg\min_{\mathbf{w}_c, \mathbf{w}_f} \mathcal{L}(\mathbf{w}_c, \mathbf{w}_f, \hat{\mathbf{w}}_d) \tag{3}$$

$$(\hat{\mathbf{w}}_d) = \arg\max_{\mathbf{w}_d} \mathcal{L}(\hat{\mathbf{w}}_c, \hat{\mathbf{w}}_f, \mathbf{w}_d) \tag{4}$$

The solution to (3) and (4) can be found by using the following stochastic updates:

$$\mathbf{w}_c^{(i+1)} = \mathbf{w}_c^{(i)} - \mu\nabla\mathcal{L}_c(\mathbf{w}_c^{(i)}) \tag{5}$$

$$\mathbf{w}_f^{(i+1)} = \mathbf{w}_f^{(i)} - \mu\left(\nabla\mathcal{L}_c(\mathbf{w}_f^{(i)}) - \lambda\nabla\mathcal{L}_d(\mathbf{w}_f^{(i)})\right) \tag{6}$$

$$\mathbf{w}_d^{(i+1)} = \mathbf{w}_d^{(i)} - \mu\nabla\mathcal{L}_d(\mathbf{w}_d^{(i)}) \tag{7}$$

where $\nabla f(x)$ is the gradient of f with respect to x, μ is the learning rate, and (i) is the iteration number. The only difference between the update (5)–(7) and stochastic gradient descent (SGD) update is the $-\lambda$ factor in (6). Without this factor, SGD aims to make features of target regions over time dissimilar in order to minimize the discriminator loss. Therefore, $-\lambda$ factor is important to adjust the weight of generator network, $\mathbf{w}_f$, for producing similar features for recent and earlier target regions. On the other hand, the weights of the discriminator network, $\mathbf{w}_d$, are adjusted in (7) to discriminate between the features of recent and earlier target regions. This competition results in achieving the solution of the mini-max optimization problem in (3) and (4). We incorporate a gradient reverse layer (Ganin & Lempitsky, 2014) in the adversarial network to make the updates in (5)–(7) in align with the updates of the SGD method and find the solution in an end-to-end manner. In the feed-forward, the gradient reverse layer is an identity transform, while in the back-propagation, the gradient reverse layer multiplies the gradient by $-\lambda$ and passes it to the preceding layer. As shown in Fig. 2, the gradient reverse layer is inserted between G_f and G_d to make them competing with each other over optimization of $\mathcal{L}_d$.

2.2. Network architecture

As mentioned in Section 2.1, the proposed AVA tracker consists of three networks, whose architectures are presented in Fig. 2. In this subsection, we provide detailed information about the dimension of input and output features of each network layer.

The classification network, G_c, is the main network used in trackers such as MDNet (Nam & Han, 2016), DAT (Pu et al., 2018), and VITAL (Song et al., 2018). This network has a simple architecture that is suitable for visual tracking and has shown to achieve superior tracking performance in multiple trackers. It has three convolutional layers as shown in the feature extractor block in Fig. 2 followed by three fully connected layers. The input image to the feature extractor block is resized to the dimension of $107 \times 107 \times 3$ and the output feature map of this block has the dimension of $3 \times 3 \times 512$. This feature map is vectorized to a 4608-dimensional array and passed to the fully connected layers. The output of the first, second, and the third fully connected layers is 512, 512, and 2, respectively. The last 2 output values correspond to the background and target scores. More information can be found in Nam and Han (2016).

The generator network, G_f, in the proposed AVA tracker aims to learn an adaptation mask for feature alignment by maximizing the discriminator loss $\mathcal{L}_d$. This network consists of two fully connected layers combined with dropout and activation functions. Specifically, after applying a dropout operation with a dropout rate of 0.5, the first fully connected layer takes the input feature vector with the dimension of 4608 and outputs a 256-dimensional feature vector. The resultant feature vector is then activated using the ReLU activation function. A similar dropout operation is performed before the second fully connected layer to produce a 3×3 adaptation mask, which is activated using the sigmoid activation function. The dimension of the adaptation mask is the same as the spatial dimension of the output from the feature extractor block in G_c. This mask is multiplied with the feature map generated from the feature extractor block to highlight the representation of regions with higher feature similarity.

The discriminator network, G_d, in the proposed AVA tracker aims to distinguish recent target regions from earlier target regions by minimizing the discriminator loss $\mathcal{L}_d$. This network has three fully connected layers combined with dropout and activation functions. The input for the first fully connected layer is the vectorized aligned features with the dimension of 4608. The output of the first, second, and third fully connected layers are 512, 512, and 1, respectively. The output of the first and second fully connected layers are ReLU activated and the output of the third fully connected layer is sigmoid activated. The output value from the last fully connected layer is a binary prediction value for recent and earlier target regions, which is further used to optimize the cross-entropy loss (i.e., $\mathcal{L}_d$) in the aforementioned mini-max optimization problem.

2.3. Online AVA tracking

In this subsection, we present detailed information regarding initializing the tracker, obtaining the tracking result for each frame, updating the model, and setting parameters to run the AVA tracker on a sequence.

Model Initialization: Following the initial parameters set in the MDNet tracker (Nam & Han, 2016), we pre-train the classification network G_c on auxiliary frame sequences. At the first frame of each testing sequence, we load the pre-trained model of G_c, freeze the convolutional layer parameters, and fine-tune the parameters of the fully connected layers (Nam & Han, 2016). Particularly, we randomly draw target and background samples around the initial location of the target region and re-train the fully connected layers of G_c to be adapted to the current frame sequence. However, we do not perform any training on the adversarial network for the first frame in the sequence.

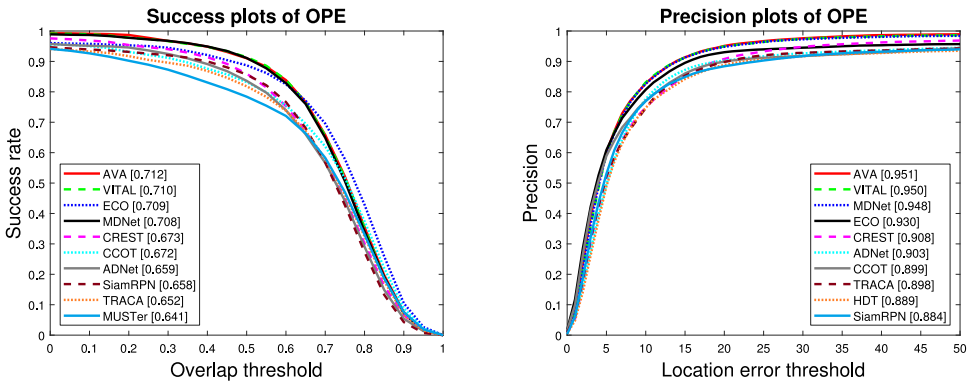
Tracking: We produce n_s number of samples around the target identified in the previous frame. These candidates are passed through the classification network G_c. The candidate that yields the highest target score is considered as the tracking result in the current frame. The adversarial network (i.e., G_f and G_d) is disabled in this testing step (Cao, Ma, Long, & Wang, 2018; Song et al., 2018).

Model Update: The model is automatically updated every other 10 frames. In order to capture the latest appearance variations of targets over time, we update the classification network G_c, the generator network G_f, and the discriminator network G_d in the adversarial network in an end-to-end manner. The adversarial network attempts to align feature distributions of both recent and earlier target regions. When a tracking result has a negative target score, we update the classification network G_c using the tracking results up to the current frame. It should be noted that the adversarial network is not updated to avoid error propagation.

Parameters Setup: The following parameter settings are used to run the proposed AVA tracker on each frame sequence. n_s target candidates are generated similar to MDNet tracker, where $n_s = 256$. We use the same parameters for MDNet to pre-train G_c since they have shown to be effective to achieve good tracking performance. The parameters for the adversarial network are empirically determined to be optimal to achieve good tracking performance and fast convergence. They are summarized as follows: The learning rate of G_f is 0.1 and the learning rate of G_d is 0.0001. These learning rates are empirically determined for fast convergence. The number of training iterations for the adversarial network is set to be 50. The momentum and weight decay for all three networks G_c, G_f, and G_d are set to be 0.9 and 0.0005, respectively. The number of target samples in each mini-batch is 64, where 32 of them are randomly selected from recent target subset and 32 of them are randomly selected from earlier target subset. The recent target subset is defined as the second half of the target samples up to the current frame. The earlier target subset is the first half of the target samples up to the current frame. The number of background samples in each mini-batch is 96. We set $\lambda = 0.01$ for the discriminator loss.

3. Experimental results

We perform extensive experiments to evaluate the performance of the proposed AVA tracker in terms of accuracy and



(a) Success plot of OPE (b) Precision plot of OPE

Fig. 3. The overall OPE plots for OTB50 benchmark (Wu et al., 2013).

robustness. We compare the AVA tracker with state-of-the-art trackers on OTB50 (Wu et al., 2013), OTB100 (Wu et al., 2015), and VOT2016 (Kristan et al., 2016) tracking benchmarks. For OTB experiments, we pre-train the network using 58 VOT2016 sequences, which do not include the common sequences in the OTB100 dataset. For VOT experiments, we pre-train the network using 89 OTB100 sequences, which do not include the common sequences in the VOT2016 dataset. We implement the AVA tracker in Python with PyTorch deep learning framework on a machine with a 3.60 GHz CPU, 32 GB RAM, and a 1080Ti 11GB Nvidia GPU.

3.1. Evaluation metrics

We follow standard protocols introduced in popular tracking benchmarks (Kristan et al., 2016; Wu et al., 2013, 2015) to compare the performance of different trackers. For OTB50 and OTB100 datasets, we perform one pass evaluation (OPE) experiments and display success and precision plots. OPE is conventionally used to evaluate trackers by initializing them using the ground truth location in the first frame. Success plots display success rates at different overlap thresholds for the bounding box overlap ratio. Precision plots display precision rates at different error thresholds for the center location error. To rank trackers using success plots, we calculate the area under curve (AUC) score for each compared tracker on all image sequences. To rank trackers using precision plots, we calculate the average precision score for each compared tracker on all image sequences at the location error threshold of 20 pixels (Wu et al., 2013, 2015). For the VOT2016 dataset (Kristan et al., 2016), we use accuracy, failure rate, and expected average overlap (EAO) to evaluate the tracker's performance. Accuracy is the average of overlap ratios between ground-truth and detected bounding boxes. Failure rate is a robustness measure and computed as the average of the number of times that trackers fail. EAO measures the expected no-reset overlap of a tracker run on a short-term sequence and combines the raw values of accuracy and failure per frame in a principled manner.

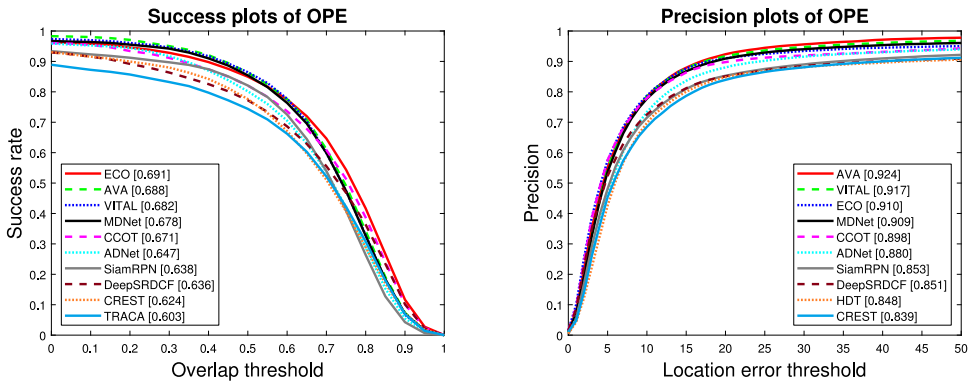
3.2. Experimental results on OTB50

This benchmark consists of 50 annotated sequences, where 49 sequences have one annotated target and one sequence (*jogging*) has two annotated targets. We compare AVA with 29 baseline trackers in Wu et al. (2013), and 25 recent trackers including DSST (Danelljan, Häger, Khan, & Felsberg, 2014), KCF (Henriques et al., 2015), TGPR (Gao, Ling, Hu, & Xing, 2014), MEEM (Zhang et al., 2014), MUSTer (Hong et al., 2015), LCT (Ma, Yang, Zhang, & Yang, 2015), RSST (Zhang et al., 2018), SRDCF (Danelljan, Hager, Shahbaz Khan, & Felsberg, 2015a), DeepSRDCF (Danelljan, Hager, Shahbaz Khan, & Felsberg, 2015b), SiamFC (Bertinetto, Valmadre, Henriques, Vedaldi, & Torr, 2016), ADNet (Yun, Choi, Yoo, Yun, & Young Choi, 2017), CFNet (Valmadre, Bertinetto, Henriques, Vedaldi, & Torr, 2017), SGLST (Javanmardi & Qi, 2019),SCT (Choi, Jin Chang, Jeong, Demiris, & Young Choi, 2016), CNN-SVM (Hong et al., 2015b), CCOT (Danelljan et al., 2016), ECO (Danelljan et al., 2017), MDNet (Nam & Han, 2016), VITAL (Song et al., 2018), CREST (Song et al., 2017), TRACA (Choi et al., 2018), SiamRPN (Li, Yan, Wu, Zhu, & Hu, 2018), STAPLE (Bertinetto, Valmadre, Golodetz, et al., 2016), CNT (Zhang, Liu, Wu, & Yang, 2016), and HDT (Qi et al., 2016). Adopting the protocol proposed in Wu et al. (2013), we use the same parameters for all sequences to obtain OPE results.

We present the overall OPE success and precision plots in Fig. 3. We include the top 10 of the 55 compared trackers in each plot to avoid clutter and increase the readability. The value within the parenthesis alongside each legend of success plots is the AUC score for its corresponding tracker. Similarly, the value within the parenthesis alongside each legend of precision plots is the precision score for its corresponding tracker. Fig. 3 clearly demonstrates that the proposed AVA tracker achieves the best tracking performance with the highest AUC score of 0.712 and the highest precision score of 0.951 when comparing with 55 state-of-the-art trackers. Among the 29 baseline trackers employed in Wu et al. (2013), SCM (Zhong, Lu, & Yang, 2012) achieves the best performance with an AUC score of 0.499 and a precision score of 0.649. The proposed AVA tracker significantly outperforms SCM by 42.69% and 46.53% in terms of AUC and precision scores, respectively. It also improves AUC scores of the top 9 trackers among the 25 additional recent trackers, namely, MUSTer, TRACA, SiamRPN, ADNet, CCOT, CREST, MDNet, ECO, and VITAL by 11.08%, 9.20%, 8.21%, 8.04%, 5.95%, 5.79%, 0.56%, 0.42%, 0.28%, respectively. It outperforms precision scores of the top 9 trackers among the 25 additional recent trackers, namely, SiamRPN, HDT, TRACA, CCOT, ADNet, CREST, ECO, MDNet, and VITAL by 7.58%, 6.97%, 5.90%, 5.78%, 5.32%, 4.74%, 2.26%, 0.32%, and 0.11%, respectively.

3.3. Experimental results on OTB100

OTB100 (Wu et al., 2015) extends OTB50 (Wu et al., 2013) by adding 48 additional annotated sequences. Two sequences, *jogging* and *Skating*, have two annotated targets. The rest of the sequences have one annotated target. Each of 100 sequences is also labeled with attributes specifying the presence of different challenges including illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast



(a) Success plot of OPE (b) Precision plot of OPE

Fig. 4. The overall OPE plots for OTB100 benchmark (Wu et al., 2015).

motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background clutter (BC), and low resolution (LR). The sequences are categorized based on the attributes and 11 challenge subsets are generated. These subsets are utilized to evaluate the performance of trackers in different challenge categories.

We evaluate the performance of the proposed AVA tracker against the same state-of-the-arts trackers presented in the OTB50 experiment. The MUSTer and CNT tracker are excluded from this experiment since they do not have any published results on OTB100. Similar to the experiments on OTB50, we follow the protocol proposed in Wu et al. (2013, 2015) and use the same parameters on all the sequences to obtain OPE results. To avoid clutter and increase the readability, we present the overall OPE success and precision plots for the top 10 of the 53 compared trackers in Fig. 4. Each tracker's AUC and precision scores are shown inside their corresponding parenthesis in the success and precision plots, respectively. It clearly shows that the proposed AVA tracker achieves a favorable performance against state-of-the-art trackers in terms of both AUC and precision scores. Among the 29 baseline trackers, Struck (Hare et al., 2016) is the best tracker yielding an AUC score of 0.463 and a precision score of 0.640. The proposed AVA tracker outperforms Struck by 48.60% in AUC score and 44.38% in precision score. When comparing with the 23 additional recent trackers, the proposed AVA tracker achieves the second highest AUC score of 0.688 and the highest precision score of 0.924. ECO achieves the best AUC score of 0.691, which improves the AUC score of AVA by 0.44%. Specifically, AVA improves the AUC scores of TRACA, CREST, DeepSRDCF, SiamRPN, ADNet, CCOT, MDNet, and VITAL by 14.10%, 10.26%, 8.18%, 7.84%, 6.34%, 2.53%, 1.47%, and 0.88%, respectively. It also outperforms CREST, HDT, DeepSRDCF, SiamRPN, ADNet, CCOT, MDNet, ECO, and VITAL in terms of precision score by 10.13%, 8.96%, 8.58%, 8.32%, 5.00%, 2.90%, 1.65%, 1.54%, and 0.76%, respectively.

In Fig. 5, we present success plots of the top 10 trackers for 8 challenge subsets containing large appearance changes of target regions. The number of sequences in each specific subset is shown in the parenthesis at the top of its plot. The AUC scores are shown in the parenthesis alongside the legend of the tracker. The results of the other trackers can be found in Wu et al. (2015).

It is clear from Fig. 5 that the proposed AVA tracker successfully handles significant appearance variations of targets due to deformation, scale variation, in-plate rotations, out-plane rotations and occlusions. It achieves the best performance in 5 of these 8 challenge subsets such as DEF, IPR, OPR, LR, and SV and achieves the third best performance in the remaining 3 challenge subsets. Compared to the base model MDNet, AVA achieves the best AUC scores for all the aforementioned challenge subsets. This mainly due to the adaptation mask learned in the generator and discriminator network. This adaptation mask highlights different variations of target over time. In addition, it aligns the discriminative features of target candidates to increase their similarities during frame sequences, while simultaneously maintaining their distinctive properties from the background. Compared to the improved base model VITAL, AVA tracker achieves better performance in all challenge subsets except for the OV subset. This is mainly due to the incorporation of a weighted combination of different parts of target features in each training iteration. Such an incorporation increases the temporal generalization capability of the model and therefore avoids the loss of informative local features over a long temporal span.

Table 1
Comparison of the state-of-the-art trackers in terms of EAO, failure rate, and accuracy on the VOT2016 dataset.

	CCOT	ECO	Staple	MDNet	VITAL	AVA
EAO	0.329	0.374	0.294	0.257	0.322	0.366
Failure rate	0.85	0.72	1.35	1.20	0.98	0.68
Accuracy	0.52	0.54	0.54	0.53	0.54	0.53

3.4. Experimental results on VOT2016

We conduct supervised evaluation on 60 VOT2016 sequences (Kristan et al., 2016). Based on the VOT challenge protocol, the target is re-initialized using the ground-truth whenever a tracker fails. A tracker is considered as failed in a frame, when the overlap ratio of the tracking result and the ground-truth is zero. The re-initialization happens 5 frames after the failure and the performance is re-evaluated after 10 frames to avoid the bias.

Table 1 compares the proposed AVA tracker with the baseline tracker Staple and the top 4 trackers (ECO, VITAL, MDNet, and CCOT) in OTB100 in terms of accuracy, failure rate, and EAO. Values in red indicate the best performance and values in blue indicate the second best performance. It shows that the proposed AVA tracker obtains a comparable EAO value (e.g., 0.366) with ECO and stands as the second best tracker in terms of EAO. It achieves the best robustness performance and yields the lowest failure rate of 0.68 among the compared trackers. It achieves the second best accuracy of 0.53, which is comparable to the best accuracy of 0.54 tied by ECO, Staple, and VITAL. It is interesting to observe that trackers (e.g., Staple, MDNet, and VITAL) with a higher failure rate (i.e., more re-initialization) still attain better accuracy despite the reduction of the re-initialization bias for accuracy calculation. As a result, the EAO measure, which

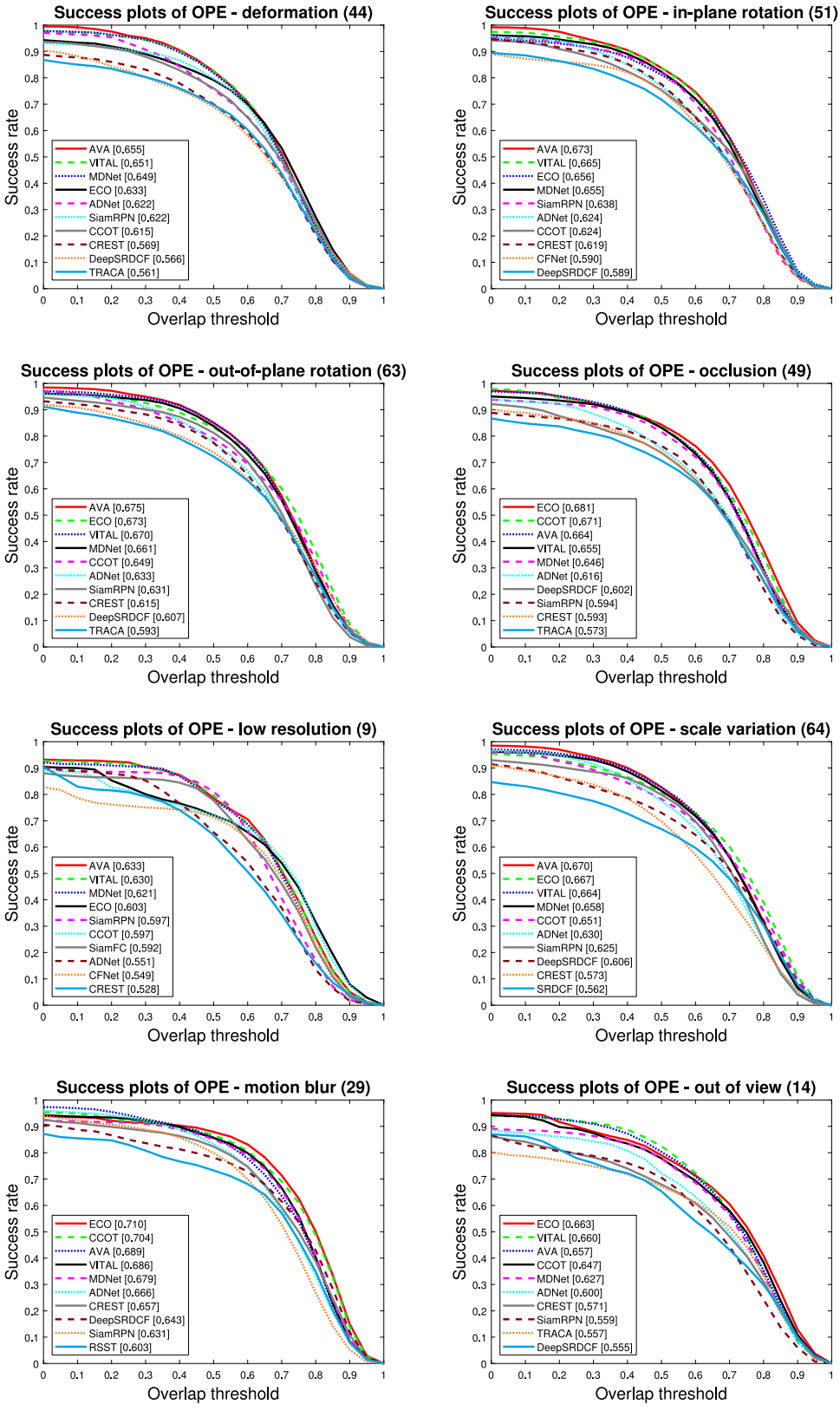


Fig. 5. OTB100 OPE success plot for DEF, IPR, OPR, OCC, LR, SV, MB, and OV challenge subsets.

simultaneously considers both accuracy and failure rate, is considered as the best evaluation metric for the VOT2016 benchmark. The VOT2016 report (Kristan et al., 2016) states that trackers with the EAO value exceeding a limit of 0.251 are considered as state-of-the-art. Table 1 clearly demonstrates that the proposed AVA tracker outperforms its peers in terms of its EAO score. It improves the EAO score of MDNet by 42.41% and the EAO score of VITAL by 13.66%. This improvement is mainly due to integration of the adversarial network and alignment of the discriminative features of target candidates over time.

Table 2
Comparison of the proposed AVA tracker with 11 state-of-the-art trackers on OTB50, OTB100, and VOT2016 challenging tracking benchmarks. Numbers in red, blue, green indicate the best, the second best, and the third best performance, respectively. The dash line (–) indicates no reported result.

Trackers	Year	Publisher	OTB50 (AUC)	OTB100 (AUC)	VOT2016 (EAO)
MDNet	2017	CVPR	0.708	0.678	0.257
ECO	2017	CVPR	0.709	0.691	0.374
DAT	2018	NIPS	0.704	0.673	0.320
SiamRPN	2018	CVPR	–	0.640	0.340
StructSiam	2018	ECCV	0.640	0.620	0.260
TriSiam	2018	ECCV	0.62	0.59	–
VITAL	2018	CVPR	0.710	0.682	0.322
TADT	2019	CVPR	0.680	0.660	0.299
UTD	2019	CVPR	–	0.632	0.301
LDES	2019	AAAI	0.677	0.643	–
SiamRPN+	2019	CVPR	0.670	0.670	0.380
AVA (ours)	2019	JNN	0.712	0.688	0.366

3.5. Comparison and discussion

We provide comprehensive comparison of the proposed AVA tracker, eight additional state-of-the-art trackers published in 2018 or 2019, and the top three trackers (ECO, MDNet, and VITAL) in previous subsections on three tracking benchmarks. The eight additional trackers include DAT (Pu et al., 2018), SiamRPN (Li et al., 2018), StructSiam (Zhang et al., 2018), TriSiam (Dong & Shen, 2018), TADT (Li, Ma, Wu, He, & Yang, 2019a), UDT (Wang et al., 2019), LDES (Li et al., 2019), and SiamRPN+ (Zhang & Peng, 2019). Table 2 summarizes the performance of these 12 compared trackers in terms of the AUC score on OTB50 and OTB100 benchmarks and in terms of the EAO score on the VOT2016 benchmark. The AUC and EAO scores of the 11 compared trackers are directly copied from the researchers' published work. We also include the year and the publication venue that each compared tracker was published. To facilitate comparison, we list the trackers in the chronological order. Table 2 clearly demonstrates that the proposed AVA tracker achieves the best AUC score of 0.712 on OTB50, which improves the second best tracker VITAL by 0.28% and the third best tracker ECO by 0.42%. AVA achieves the second best AUC score of 0.688 on OTB100 with a 0.44% decrease when compared to the best tracker ECO and a 0.88% improvement when compared to the third best tracker VITAL. AVA achieves the third best EAO score of 0.366 on VOT2016 with a 3.82% decrease when compared to the best tracker SiamRPN+ and a 2.19% decrease when compared to the second best tracker ECO. It is clear that none of these state-of-the art trackers consistently performs the best on three tracking benchmarks. AVA and ECO are the only two trackers that rank as the top 3 trackers on three tracking benchmarks.

We also summarize the performance comparison of the proposed AVA tracker, its model-based peer tracker MDNet (Nam & Han, 2016), and its adversarial learning-based peer tracker VITAL (Song et al., 2018) on OTB50, OTB100, and VOT2016 challenging tracking benchmarks. For the OTB50 benchmark (Fig. 3), AVA outperforms MDNet by 0.56% and VITAL by 0.28% in the AUC score and outperforms MDNet by 0.32% and VITAL by 0.11% in the precision score. For the OTB100 benchmark (Fig. 4), AVA improves the AUC and precision scores of MDNet by 1.47% and 1.65%, respectively. It improves VITAL by 0.88% in terms of the AUC score and by 0.76% in terms of the precision score. For the VOT2016 benchmark (Table 1), AVA attains comparable accuracy with both MDNet and VITAL. However, it drastically improves the failure rate of both MDNet and VITAL. This results in an EAO score improvement of 42.41% and 13.66% over MDNet and VITAL, respectively. For eight challenge subsets containing large appearance changes of target regions (Fig. 5), AVA achieves better AUC scores than both MDNet and VITAL when a target undergoes deformation, in-plane rotations, out-plane rotations, occlusions, low resolution, scale variation, and motion blur. It achieves a better AUC score than MDNet and a comparable AUC score as VITAL when target is out of view. Overall, the proposed AVA tracker uses the model of MDNet as a base network. Unlike MDNet, AVA aligns the feature distributions of target regions over time by learning an adaptation mask adversarially. This adaptation mask increases the model generalization by highlighting the informative features of target regions over time and dropping out some non-informative features. Therefore, the more generalized model tends to attain the similarity between the features distributions of recent and earlier target regions while maintaining distinctive properties from the background. The VITAL tracker also learns a mask during tracking. However, it prepares 9 random masks where each mask covers one of 9 locations in the 3×3 feature map in each training iteration and learns an optimal mask in a least square optimization problem. Therefore, this optimal mask is updated to cover only one part of local features in each iteration, which leads to the loss of the informative local features during training. Unlike the optimal mask learned in VITAL, the adaptation mask learned in AVA increases the temporal generalization capability and avoids the loss of informative local features over time by incorporating a weighted combination of multiple parts of target features in each training iteration via a gradient reverse layer.

All CNN-based trackers aim to construct a model to classify the candidates in each frame as a target or a background. They update the model during tracking to keep track of the latest changes of target regions. However, one major shortcoming is that the model may overfit to the initial target appearances, which leads to the failure to discriminate the similarity of the current target with its tracked targets in earlier frames when a target appearance has a drastic change. The proposed AVA tracker aims to address this shortcoming using adversarial learning. Our extensive experimental results show that the AVA tracker outperforms most state-of-the-art trackers in various challenges (e.g., occlusion, fast motion, scale variation, rotations, etc.) for OTB and VOT benchmarks. However, like all other trackers, its performance decreases for the most challenging sequences such as *soccer, bird1, fenando, rabbit* where a target suffers from heavy occlusion, *skiing, trans, motorRolling, matrix* where a target's scale changes with a fast rate, and *matrix, iroman, gymnastics* where a target's motion rate is high. To the best of our knowledge, none of the existing trackers is able to handle all the severe challenges that lead to significant appearance changes of the tracked targets. Designing a tracker that is able to produce a model to discriminate target regions from background in all frames of challenging sequences is still an active and open computer vision task.

4. Conclusions and future work

We propose an appearance variation adaptation (AVA) tracker that is capable of handling the significant appearance variations of targets. Our contributions are: (1) Aligning feature distributions of target regions over a long time span by adversarially learning an adaptation mask. This adaptation mask is applied on the discriminative features of target regions to increase the generalization of the classification network. (2) Designing an adversarial network, which consists of a generator network and a discriminator network competing with each other over optimization of a discriminator loss between recent and earlier target

regions. The discriminator network aims to distinguish recent target regions from earlier ones by minimizing the discriminator loss, while the generator network aims to produce an adaptation mask to maximize the discriminator loss. (3) Incorporating the adversarial network with the classification network to align informative features of recent and earlier target regions during tracking, while maintaining the network classification accuracy to distinguish targets and backgrounds. We add a gradient reverse layer to solve the aforementioned mini-max optimization in an end-to-end manner. Our extensive experiments on OTB and VOT challenge benchmarks show that the proposed AVA tracker achieves favorable performance against state-of-the-arts trackers.

In the future, we plan to use worst-case target appearance changes in auxiliary frame sequences to train a model in an adversarial manner before tracking takes place. We will update the model during tracking to be fine-tuned with the current target. Furthermore, we will investigate adversarial dropout (Park, Park, Shin, & Moon, 2018) in visual tracking and incorporate it on the channels of the feature map to keep the informative features and achieve better model generalization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., & Torr, P. H. (2016). Staple: Complementary learners for real-time tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1401–1409).

Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., & Torr, P. H. (2016). Fully-convolutional siamese networks for object tracking. In *European conference on computer vision* (pp. 850–865). Springer.

Cao, Z., Ma, L., Long, M., & Wang, J. (2018). Partial adversarial domain adaptation. In *Proceedings of the European conference on computer vision* (pp. 135–150).

Choi, J., Jin Chang, H., Fischer, T., Yun, S., Lee, K., Jeong, J., et al. (2018). Context-aware deep feature compression for high-speed visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 479–488).

Choi, J., Jin Chang, H., Jeong, J., Demiris, Y., & Young Choi, J. (2016). Visual tracking using attention-modulated disintegration and integration. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4321–4330).

Dai, K., Wang, D., Lu, H., Sun, C., & Li, J. (2019). Visual tracking via adaptive spatially-regularized correlation filters. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4670–4679).

Danelljan, M., Bhat, G., Shahbaz Khan, F., & Felsberg, M. (2017). Eco: Efficient convolution operators for tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6638–6646).

Danelljan, M., Häger, G., Khan, F., & Felsberg, M. (2014). Accurate scale estimation for robust visual tracking. In *British machine vision conference*. BMVA Press.

Danelljan, M., Hager, G., Shahbaz Khan, F., & Felsberg, M. (2015a). Learning spatially regularized correlation filters for visual tracking. In *Proceedings of the IEEE international conference on computer vision* (pp. 4310–4318).

Danelljan, M., Hager, G., Shahbaz Khan, F., & Felsberg, M. (2015b). Convolutional features for correlation filter based visual tracking. In *Proceedings of the IEEE international conference on computer vision workshops* (pp. 58–66).

Danelljan, M., Robinson, A., Khan, F. S., & Felsberg, M. (2016). Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *European conference on computer vision* (pp. 472–488). Springer.

Dong, X., & Shen, J. (2018). Triplet loss in siamese network for object tracking. In *Proceedings of the European conference on computer vision* (pp. 459–474).

Ganin, Y., & Lempitsky, V. (2014). Unsupervised domain adaptation by backpropagation. arXiv preprint arXiv:1409.7495.

Gao, J., Ling, H., Hu, W., & Xing, J. (2014). Transfer learning based visual tracking with gaussian processes regression. In *European conference on computer vision* (pp. 188–203). Springer.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.-M., Hicks, S. L., et al. (2016). Struck: Structured output tracking with kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(10), 2096–2109.

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

Hong, Z., Chen, Z., Wang, C., Mei, X., Prokhorov, D., & Tao, D. (2015). Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 749–758).

Hong, S., You, T., Kwak, S., & Han, B. (2015). Online tracking by learning discriminative saliency map with convolutional neural network. In *International conference on machine learning* (pp. 597–606).

Javanmardi, M., Farzaneh, A. H., & Qi, X. (2020). A robust structured tracker using local deep features. *Electronics*, *9*(5), 846.

Javanmardi, M., & Qi, X. (2019). Structured group local sparse tracker. *IET Image Processing*, *13*(8), 1391–1399.

Kristan, M., et al. (2016). The visual object tracking VOT2016 challenge results. In *Computer vision – ECCV 2016 workshops* (pp. 777–823). Cham: Springer International Publishing.

Li, X., Ma, C., Wu, B., He, Z., & Yang, M.-H. (2019). Target-aware deep tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1369–1378).

Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., & Yan, J. (2019). Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4282–4291).

Li, B., Yan, J., Wu, W., Zhu, Z., & Hu, X. (2018). High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8971–8980).

Li, Y., Zhu, J., Hoi, S. C., Song, W., Wang, Z., & Liu, H. (2019). Robust estimation of similarity transformation for visual object tracking. In *Proceedings of the AAAI conference on artificial intelligence (vol. 33)* (pp. 8666–8673).

Ma, C., Yang, X., Zhang, C., & Yang, M.-H. (2015). Long-term correlation tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5388–5396).

Nam, H., Baek, M., & Han, B. (2016). Modeling and propagating cnns in a tree structure for visual tracking. arXiv preprint arXiv:1608.07242.

Nam, H., & Han, B. (2016). Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4293–4302).

Park, S., Park, J., Shin, S.-J., & Moon, I.-C. (2018). Adversarial dropout for supervised and semi-supervised learning. In *Thirty-second AAAI conference on artificial intelligence*.

Pu, S., Song, Y., Ma, C., Zhang, H., & Yang, M.-H. (2018). Deep attentive tracking via reciprocative learning. In *Advances in neural information processing systems* (pp. 1931–1941).

Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., et al. (2016). Hedged deep tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4303–4311).

Schmidhuber, J. (2020). Generative adversarial networks are special cases of artificial curiosity (1990) and also closely related to predictability minimization (1991). *Neural Networks*.

Simonyan, K., & Zisserman, A. Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

Song, Y., Ma, C., Gong, L., Zhang, J., Lau, R. W., & Yang, M.-H. (2017). Crest: Convolutional residual learning for visual tracking. In *Proceedings of the IEEE international conference on computer vision* (pp. 2555–2564).

Song, Y., Ma, C., Wu, X., Gong, L., Bao, L., Zuo, W., et al. (2018). Vital: Visual tracking via adversarial learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8990–8999).

Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., & Torr, P. H. (2017). End-to-end representation learning for correlation filter based tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2805–2813).

Wang, N., Song, Y., Ma, C., Zhou, W., Liu, W., & Li, H. (2019). Unsupervised deep tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1308–1317).

Wang, N., & Yeung, D.-Y. (2013). Learning a deep compact image representation for visual tracking. In *Advances in neural information processing systems* (pp. 809–817).

Wu, Y., Lim, J., & Yang, M.-H. (2013). Online object tracking: A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2411–2418).

Wu, Y., Lim, J., & Yang, M.-H. (2015). Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(9), 1834–1848.

Yun, S., Choi, J., Yoo, Y., Yun, K., & Young Choi, J. (2017). Action-decision networks for visual tracking with deep reinforcement learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2711–2720).

Zhang, K., Liu, Q., Wu, Y., & Yang, M.-H. (2016). Robust visual tracking via convolutional networks without training. *IEEE Transactions on Image Processing*, *25*(4), 1779–1792.

Zhang, J., Ma, S., & Sclaroff, S. (2014) MEEM: Robust tracking via multiple experts using entropy minimization. In *Proc. of the European conference on computer vision*.

Zhang, Z., & Peng, H. (2019). Deeper and wider siamese networks for real-time visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4591–4600).

Zhang, Y., Wang, L., Qi, J., Wang, D., Feng, M., & Lu, H. (2018). Structured siamese network for real-time visual tracking. In *Proceedings of the European conference on computer vision* (pp. 351–366).

Zhang, T., Xu, C., & Yang, M.-H. (2018). Robust structural sparse tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhong, W., Lu, H., & Yang, M.-H. (2012). Robust object tracking via sparsity-based collaborative model. In *Computer vision and pattern recognition, 2012 IEEE conference on* (pp. 1838–1845). IEEE.