Target-Aware Deep Tracking

Xin Li[1,2] Chao Ma[3] Baoyuan Wu[4] Zhenyu He[1*] Ming-Hsuan Yang[2,5]

[1]Harbin Institute of Technology, Shenzhen
[2]Electrical Engineering and Computer Science, University of California, Merced
[3]AI Institute, Shanghai Jiao Tong University [4]Tencent AI Lab [5] Google Cloud AI

Abstract

Existing deep trackers mainly use convolutional neural networks pre-trained for generic object recognition task for representations. Despite demonstrated successes for numerous vision tasks, the contributions of using pre-trained deep features for visual tracking are not as significant as that for object recognition. The key issue is that in visual tracking the targets of interest can be arbitrary object class with arbitrary forms. As such, pre-trained deep features are less effective in modeling these targets of arbitrary forms for distinguishing them from the background. In this paper, we propose a novel scheme to learn target-aware features, which can better recognize the targets undergoing significant appearance variations than pre-trained deep features. To this end, we develop a regression loss and a ranking loss to guide the generation of target-active and scale-sensitive features. We identify the importance of each convolutional filter according to the back-propagated gradients and select the target-aware features based on activations for representing the targets. The target-aware features are integrated with a Siamese matching network for visual tracking. Extensive experimental results show that the proposed algorithm performs favorably against the state-of-the-art methods in terms of accuracy and speed.

1. Introduction

Visual tracking is one of the fundamental computer vision problems with a wide range of applications. Given a target object specified by a bounding box in the first frame, visual tracking aims to locate the target object in the subsequent frames. This is challenging as target objects often undergo significant appearance changes over time and may temporally leave the field of the view. Conventional trackers prior to the advances of deep learning mainly consist of a feature extraction module and a decision-making mechanism. The recent state-of-the-art deep trackers often use deep models pre-trained for the object recognition task to extract features, while putting more emphasis on designing effective decision-making modules. While various decision models, such as correlation filters [15], regressors [14, 35, 38, 37], and classifiers [16, 29, 32], are extensively explored, considerably less attention is paid to learning more discriminative deep features.

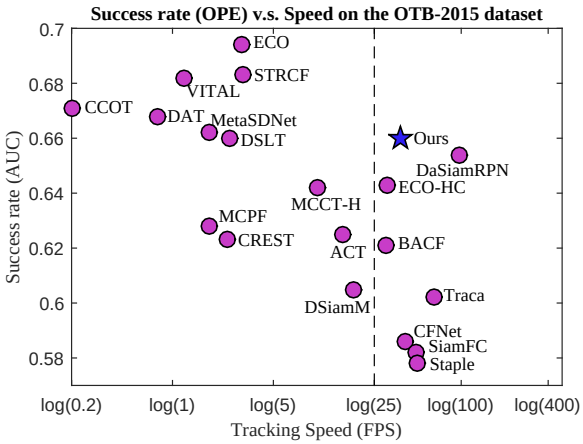


Figure 1. **Tracking accuracy vs. speed on the OTB-2015 dataset.** The horizontal and vertical coordinates correspond to tracking speed and AUC overlap ratio score, respectively. The proposed algorithm achieves a favorable performance against the state-of-the-art trackers.

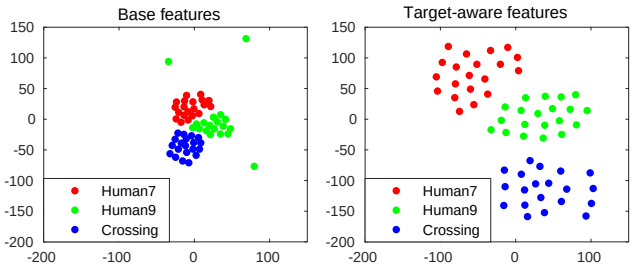
Despite the state-of-the-art performance of existing deep trackers, we note that the contributions of pre-trained deep features for visual tracking are not as significant as that for object recognition. Numerous issues may arise when using pre-trained deep features as target representation. First, a target in visual tracking can be of arbitrary forms, *e.g.*, an object unseen in the training sets for the pre-trained models or one specific part, which does not contain the object-

*Corresponding author.

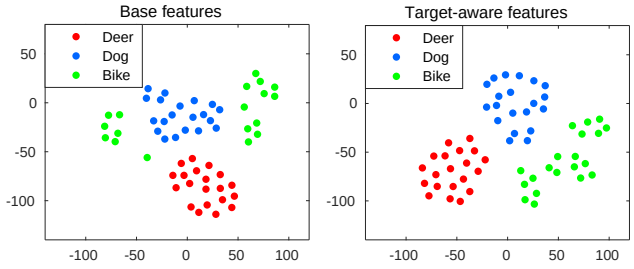
ness information exploited for the object recognition task. That is, pre-trained CNN models from generic images are agnostic of a target object of interest and less effective in separating them from the background for visual tracking. Second, even if target objects appear in the training set for pre-trained models, deep features taken from the last convolutional layers often retain only high-level visual information that is less effective for precise localization or scale estimation. Third, state-of-the-art deep trackers [29, 35, 36] require high computational loads as deep features from pre-trained models are high-dimensional (see Figure 1). To narrow this gap, it is of great importance to exploit deep features pertaining specifically to target objects for visual tracking.

To address the above-mentioned issues, we propose to learn target-aware deep features. Our work is motivated based on the following observations. The gradients obtained through back-propagating a classification neural network indicate class-specific saliency well [33]. With the use of global average pooling, the gradients generated by a convolutional filter can determine the importance of a filter for representing target objects. To select the most effective convolutional filters, we design two types of objective losses to perform back-propagation on top of a pre-trained deep model in the first frame. We use a hinge loss to regress pre-trained deep features to soft labels generated by a Gaussian function and use the gradients to select the target-active convolutional filters. We use a ranking loss with pair-wise distance to search for the scale-aware convolutional filters. The activations of the selected most important filters are the target-aware features in this work. Figure 2 shows the target-aware features and original deep features using the t-SNE method [27]. Note that the target-aware deep features are more effective in separating different target objects with a same semantic label than the pre-trained deep features, which are agnostic of the objectness of the targets. As we exploit a small set of convolutional filters to generate target-aware features, the feature number is significantly reduced, which can reduce computational loads.

We integrate the proposed target-aware features with a Siamese matching network [2] for visual tracking. The Siamese framework has recently received much attention due to its simplicity in implementation, where the ground-truth in the first frame is used as the template for locating the target location in the current frame. The Siamese framework typically does not perform an online update. Hence, the tracking accuracy of Siamese deep trackers hinges on designing discriminative features to separate target objects from the background or distractors in each frame. We extensively evaluate the proposed tracker on five large-scale benchmark datasets including OTB-2013 [45], OTB-2015 [46], VOT-2015 [19, 20], VOT-2016 [18], and Temple Color-128 [24]. Extensive experiments with ablation studies demonstrate that the proposed target-aware features are more effective than those from pre-trained models for the Siamese trackers to perform well against the state-of-the-art algorithms in terms of accuracy and tracking speed.



(a) Distributions of intra-class targets (pedestrian).



(b) Distributions of inter-class targets.

Figure 2. **Pre-trained classification CNNs features and target-aware features using the t-SNE method.** In this example, we randomly select 20 frames from each video. Each point in the figure denotes a target in one frame. (a) All points belong to the pedestrian class but in different videos. The target-aware features are more sensitive to intra-class differences for each video, which are crucial for distinguishing the target from distractors. (b) Points of different colors belong to different object classes. The target-aware features separate objects of different categories more effectively, which can be used to remove unrelated filters and retaining target-active filters.

The main contributions of this work are summarized as follows:

- We propose to learn target-aware deep features for visual tracking. We develop a regression loss and a ranking loss for selecting the most effective convolutional filters to generate target-aware features. We narrow the gap between the pre-trained deep models and target objects of arbitrary forms for visual tracking.
- We integrate the target-aware features with a Siamese matching network for visual tracking. The target-aware features with reduced number of features can accelerate Siamese trackers as well.
- We evaluate the proposed method extensively on five benchmark datasets. We show that the Siamese tracker with the proposed target-aware features performs well against the state-of-the-art methods in terms of effectiveness and efficiency.

2. Related Work

Visual tracking has been an active research topic in the literature. In the following, we mainly discuss the representative deep trackers and related issues on the gradient-based deep models.

Deep trackers. One notable issue of applying deep learning models to visual tracking is that there are limited training samples and only the ground truth visual appearance of the target object in the first frame is available. On one hand, most existing deep trackers use deep models pre-trained for the object classification task for feature representations. Several trackers [26, 42] exploit the complementary characteristics of shallow and deep layer features to enable the abilities of robustness and accuracy. Deep features from multiple layers have also integrated for visual tracking [10, 32, 7, 3]. However, the combination of pre-trained deep features may not always bring performance gains, due to issues of unseen targets, incompatible resolutions, and increasing dimensions, as demonstrated by Bhat *et al*. [3]. On the other hand, numerous trackers [16, 6, 28, 17, 35, 47, 12] are developed by improving the decision models including support vector machines, correlation filters, deep classifiers, and deep regressors. Nam and Han [29] propose a multi-domain deep classifier combined with the hard negative mining, bounding box regression, and online sample collection modules for visual tracking. The VITAL tracker [36] exploits adversarial learning to generate effective samples and leverages the class imbalance with a cost-sensitive loss. However, these models may drift from target object in the presence of noisy updates and require high computational loads, which is caused by the limited online training samples to a large extent.

To exploit datasets with general objects for tracking, numerous Siamese based trackers [2, 39, 11, 21, 14] cast tracking as a matching problem and learn a similarity measurement network. Tracking is carried out by comparing the features of the initial target template and search regions in the current frame. A number of trackers [44, 52, 13] have since been developed by introducing attention mechanisms for better matching between templates and search regions. Although these Siamese frameworks are pre-trained on large video datasets, the pair-wise training sample only tells whether the two samples belong to the same target or not without category information. That is, the Siamese trackers do not fully exploit semantic and objectness information pertaining to specific target objects. In this work, we select the most discriminative and scale-sensitive convolutional filters from a pre-trained CNN to generate target-aware deep features. The proposed features enhance the discriminative representation strength of the targets regarding semantics and objectness, which facilitate the Siamese tracking framework to perform well against the state-of-the-art methods in terms of robustness and accuracy.

Gradient-based deep models. Several gradient-based models [49, 33] are developed to determine the importance of each channel of CNN features in describing a specific object class. The GCAM model [49] generates a class-active map by computing a weighted sum along the feature channels based on the observation that the gradient at each input pixel indicates the corresponding importance belonging to given class labeling. The weight of a feature channel is computed by globally average pooling of all the gradients in this channel. Unlike these gradient-based models using classification losses, we specifically design a regression loss and a ranking loss for the tracking task to identify which convolutional filters are active to describe targets and sensitive to scale changes.

3. Target-Aware Features

In this section, we present how to learn target-aware features for visual tracking. We first analyze the gap between the features from pre-trained classification deep models and effective representations for visual tracking. Then, we present the target-aware feature model including a discriminative feature generation model and a scale-sensitive feature generation component based on the gradients of regression and ranking losses.

3.1. Features of pre-trained CNNs

The gap between the features effective for generic visual recognition and object-specific tracking is caused by the following issues. First, the pre-trained CNN features are agnostic of the semantic and objectness information of the target, which most likely does not appear in the offline training data. Different from other vision tasks (*e.g*., classification, detection, and segmentation), where the class categories for training and testing are pre-defined and consistent, online visual tracking needs to deal targets of any object labels. Second, the pre-trained CNNs focus on increasing inter-class differences and the extracted deep features are insensitive to intra-class variations. As such, these features are less effective for trackers to accurately estimate scale changes and distinguish the targets from distractors with the same class label. Third, the pre-trained deep features are sparsely activated by each category label (*i.e*., inter-class difference are mainly related to a few feature channels) especially in a deeper convolutional network. When applied to the tracking task, only a few convolutional filters are active in describing the target. A large portion of the convolutional filters contain redundancy and irrelevant information, which leads to high computational loads and overfitting. Figure 2 shows the distributions of the pre-trained deep features and the proposed target-aware features using the t-SNE method [27].

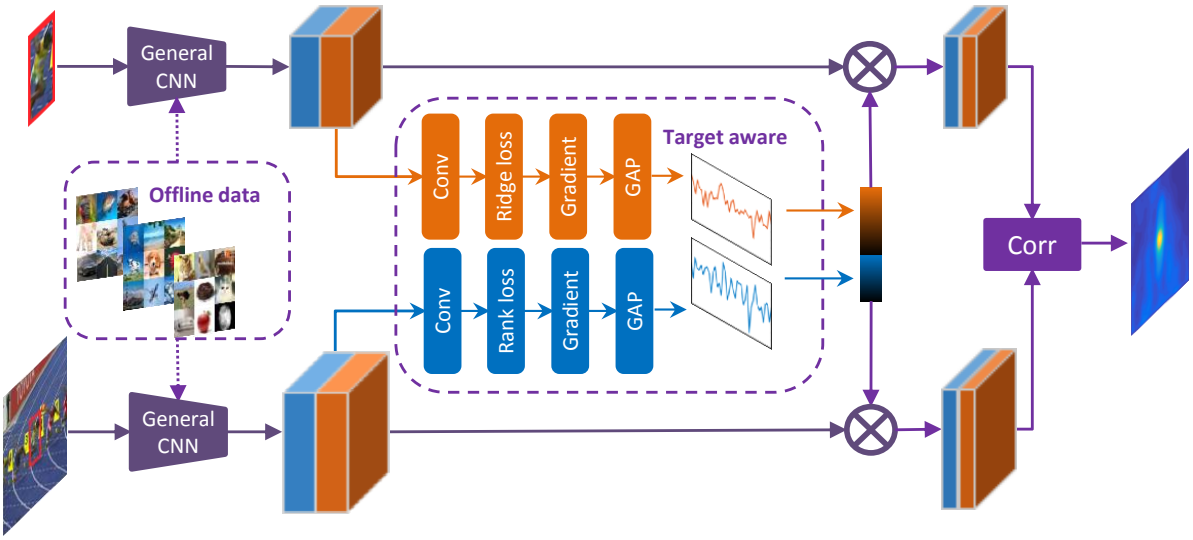


Figure 3. **Framework of the proposed algorithm.** This framework consists of a general CNN feature backbone network, a target-aware model, and a correlation matching module. The target-aware model, constructed with a regression loss part (*i.e.*, Ridge loss) and a ranking loss part, selects the target-aware filters with target-active and scale-sensitive information from the pre-trained CNNs for object recognition. The correlation matching module computes the similarity score between the template and the search region. The maximum of the score map indicates the target position.

Several methods on interpretation of neural networks demonstrate that the importance of convolutional filters on capturing the category-level object information can be computed through the corresponding gradients [49, 33]. Based on the gradient-based guidance, we construct a target-aware feature model with losses designed specifically for visual tracking. Given a pre-trained CNN feature extractor with the output feature space χ, a subspace χ' can be generated based on the channel importance Δ as

$$\chi' = \varphi(\chi; \Delta), \tag{1}$$

where φ is a mapping function selecting the most important channels. The importance of the i-th channel Δ_i is computed by

$$\Delta_i = G_{AP}(\frac{\partial L}{\partial z_i}), \tag{2}$$

where $G_{AP}(\cdot)$ denotes the global average pooling function, L is the designed loss, and z_i indicates the output feature of the i-th filter. For visual tracking, we exploit the gradients of a regression loss (Section 3.2) and a ranking loss (Section 3.3) to extract target-aware features.

3.2. Target-Active Features via Regression

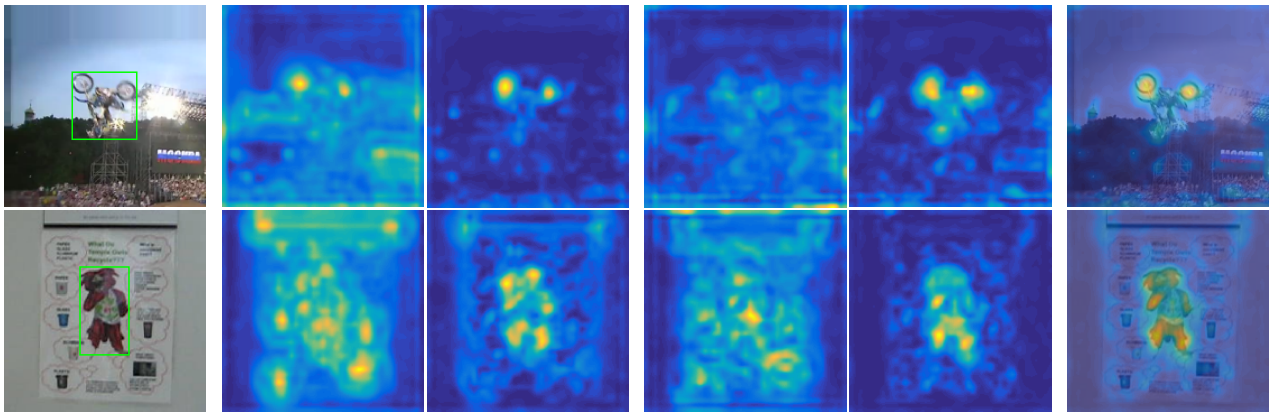
In a pre-trained classification network, each convolutional filter captures a specific feature pattern and all the filters construct a feature space containing different objectness priors. A trained network recognizes a specific object category mainly based on a subset of these filters. For the visual tracking task, we can obtain the filters with objectness information pertaining to the target by identifying those active to the target area while inactive to the backgrounds. To this end, we regress all the samples $X_{i,j}$ in an image patch aligned with the target center to a Gaussian label map $Y(i,j) = e^{-\frac{i^2+j^2}{2\sigma^2}}$, where (i,j) is the offset against the target and σ is the kernel width. For computational efficiency, we formulate the problem as the ridge regression loss,

$$L_{reg} = \|Y(i,j) - W * X_{i,j}\|^2 + \lambda\|W\|^2, \tag{3}$$

where $*$ denotes the convolution operation and W is the regressor weight. The importance of each filter can be computed based on its contribution to fitting the label map, *i.e.*, the derivation of L_{reg} with respect to the input feature X_{in}. With the chain rule and Eq. 3, the gradient of the regression loss is computed by

$$\begin{aligned} \frac{\partial L_{reg}}{\partial X_{in}} &= \sum_{i,j} \frac{\partial L_{reg}}{\partial X_o(i,j)} \times \frac{\partial X_o(i,j)}{\partial X_{in}(i,j)} \\ &= \sum_{i,j} 2(Y(i,j) - X_o(i,j)) \times W, \end{aligned} \tag{4}$$

where X_o is the output prediction. With the gradient of the regression loss and Eq. 2, we find the target-active filters that are able to discriminate the target from the background. The generated features have the following merits compared to the pre-trained deep features. We select a portion of target-specific filters to generate discriminative deep features. This not only alleviates the model over-fitting is-



(a) Input images (b) Conv4-1 w/o and w/ ranking+regression loss (c) Conv4-3 w/o and w/ regression loss (d) Target-aware

Figure 4. **Visualization of the original and the learned target-aware features.** The visualized images are generated by averaging all channels. From left to right on each row are the input images, pre-trained deep features (*Conv4-1*) without and with ranking and regression losses for learning scale-sensitive features, pre-trained deep features (*Conv4-3*) without and with a regression loss for learning objectness-sensitive features, and the overall target-aware deep features. Notice that the original pre-trained features are not effective in describing the targets, while the target-aware features can readily separate the targets from the background.

sue but also reduces the number of features. The target-aware features are effective for representing an arbitrary target or an unsee object in the training set. Figure 4(c) visually compares the deep features learned with and without regression-loss by averaging all channels.

3.3. Scale-Sensitive Features via Ranking

To generate scale-sensitive features, we need to find the filters that are most active to the target scale changes. The exact scale of the target is hard to compute as target presentation is not continuous, but we can get the closest scale with a model that tells which one has a closer size of a paired sample. As such, we formulate the problem as a ranking model and rank the training sample whose size is closer to the target size higher. The gradients of the ranking loss indicate the importance of the filters to be sensitive to scale changes. For ease of implementation, we exploit a smooth approximated ranking loss [23] defined by

$$L_{rank} = \log\big(1 + \sum_{(x_i,x_j)\in\Omega} \exp\left(f(x_i) - f(x_j)\right)\big), \quad (5)$$

where (x_i, x_j) is a pair-wise training sample and the size of x_i is closer to the target size comparing with x_j, and $f(x; w)$ is the prediction model. In addition, Ω is the set of training pairs. The derivation of L_{rank} with respect to $f(x)$ is computed as [23]:

$$\frac{\partial L_{rank}}{\partial f(x)} = -\frac{1}{L_{rank}} \sum_{\Omega} \Delta\mathbf{z}_{i,j} \exp(-f(x)\Delta\mathbf{z}_{i,j}), \quad (6)$$

where $\Delta\mathbf{z}_{i,j} = \mathbf{z}_i - \mathbf{z}_j$ and $\mathbf{z}_i$ is a one-hot vector with the i-th element being 1 while others being 0. By back-propagation, the gradients of ranking loss with respect to the features can be computed by

$$\frac{\partial L_{rank}}{\partial x_{in}} = \frac{\partial L_{rank}}{\partial x_o} \times \frac{\partial x_o}{\partial x_{in}} = \frac{\partial L_{rank}}{\partial f(x_{in})} \times W, \quad (7)$$

where W is the filter weights of the convolutional layer. With the above gradients of the ranking loss and Eq. 2, we find the filters that are sensitive to scale changes. Considering we only need the scale-sensitive features of the target object, we combine the regression and ranking losses to find the filters that are both active to the target and sensitive to scale changes. Figure 4(b) visually compares deep features generated with and without the proposed model by averaging all channels.

4. Tracking Process

Figure 3 shows the overall framework of the proposed tracker. We integrate the target-aware feature generation model with the Siamese framework due to the following two reasons. First, the Siamese framework is concise and efficient as it performs tracking by comparing the features of the target and the search region. Second, the Siamese framework can highlight the effectiveness of the proposed feature model, as its performance solely hinges on the effectiveness of the features. We briefly introduce the tracking process with the following modules.

Tracker initialization. The proposed tracking framework comprises a pre-trained feature extractor, the target-aware feature learning module, and a Siamese matching module. The pre-trained feature extractor is offline trained on the classification task and the target-aware part is only trained in the first frame. In initial training, the regression loss and

the ranking loss parts are trained separately and we compute the gradients from each loss once the networks are converged. With the gradients, the feature generation model selects a fixed number of the filters with the highest importance scores from the pre-trained CNNs. The final target-aware features are obtained by stacking these two types of feature filters. Considering the scalar difference, these two types of features are re-scaled by dividing their maximal channel summation (summation of all the values in one channel).

Online detection. At the inference stage, we directly compute the similarity scores between the initial target and the search region in the current frame using the target-aware features. This is achieved by a convolution operation (*i.e.*, the correlation layer in the Siamese framework) and outputs a response map. The value in the response map indicates the confidence of its corresponding position to be the real target. Given the initial target x_1, and the search region in the current frame z_t, the predicted target position in frame t is computed as

$$\hat{p} = \arg\max_{p} \ \chi'(x_1) * \chi'(z_t), \tag{8}$$

where * denotes the convolution operation.

Scale evaluation. To evaluate the scale change of the target, we fix the size of the template and re-scale the feature map of the search region in the current frame to smaller, larger, and fixed ones. During tracking, all these three feature maps are compared with the target template. The scale evaluation is performed by finding the score map containing the highest response.

5. Experimental Results

In this section, we first introduce the implementation details of the proposed tracker. Then, we evaluate the proposed algorithm on five benchmark datasets and compare it with the state-of-the-art methods. In addition, we conduct ablation studies to analyze the effectiveness of each module. More results and videos are available in the supplementary material.

5.1. Implementation Details

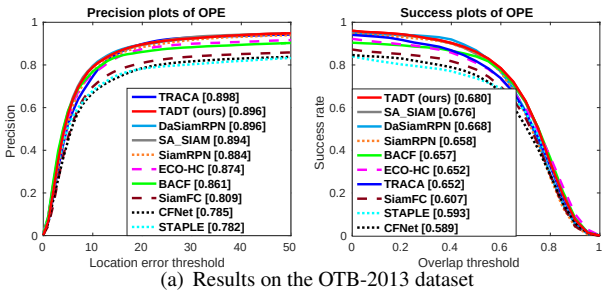
We implement the proposed tracker in Matlab with the MatConvNet toolbox [41] on a PC with 32G memory, an i7 3.6GHz CPU, and a GTX-1080 GPU. The average tracking speed is 33.7 FPS excluding the initialization time. We use the VGG-16 model [34] as the base network. To maintain more fine-grained spatial details, we use the activation outputs of the *Conv4-3* and *Conv4-1* layers as the base deep features. In the initial training, the convergence loss threshold is set to 0.02 and the maximum iteration number is 50. We select the top 250 important filters from the *Conv4-3* layer for learning target-active features and select the top 80 important filters from the *Conv4-1* layers for learning scale-sensitive features. For the Siamese framework, we use the initial target as the template and crop the search region with 3 times of the target size from the current frame. We resize the target template into a proper size if it is too large or small. For the scale evaluation, we generate a proposal pyramid with three scales, *i.e.*, 45/47, 1, and 45/43 times of the previous target size. We set the corresponding changing penalties to the pyramid to 0.990, 1, and 1.005.

Table 1. **Experimental results on the OTB datasets.** The AUC scores on the OTB-2013 and OTB-2015 datasets are presented. The notation * denotes the running speed is reported by the authors as the source code is not available. From top to bottom, the trackers are broadly categorized into three classes: correlation filters based trackers, non-real-time deep trackers, and real-time deep trackers.

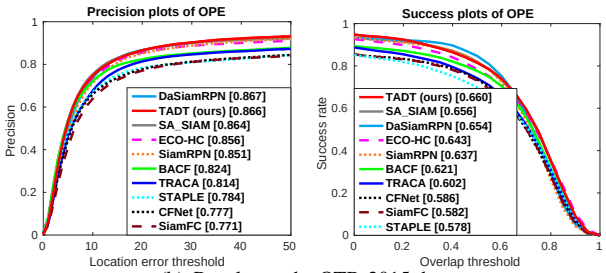
Tracker	OTB-2013	OTB-2015	Real-time	FPS
BACF [17]	0.657	0.621	Y	30
MCPF [48]	0.677	0.628	N	1.8
MCCT-H [43]	0.664	0.642	N	10
CCOT [10]	0.672	0.671	N	0.2
STRCF [22]	0.683	0.683	N	3.1
ECO [7]	0.702	0.694	N	3.1
DRT [38]	0.720	0.699	N	1.0*
DSiamM [11]	0.656	0.605	N	18
ACT [4]	0.657	0.625	N	15
CREST [35]	0.673	0.623	N	2.4
FlowT [52]	0.689	0.655	N	12*
DSLT [25]	0.683	0.660	N	2.5
DAT [31]	0.704	0.668	N	0.79
LSART [37]	0.677	0.672	N	1.0*
MDNet [29]	0.708	0.678	N	1.1
VITAL [36]	0.710	0.682	N	1.2
SiamRPN [21]	0.658	0.637	Y	71*
RASNet [44]	0.670	0.642	Y	83*
SA-Siam [13]	0.676	0.656	Y	50*
CFNet [40]	0.611	0.586	Y	41
SiamFC [2]	0.607	0.582	Y	49
TRACA [5]	0.652	0.602	Y	65
DaSiamRPN [51]	0.668	0.654	Y	97
Ours	**0.680**	**0.660**	**Y**	**33.7**

5.2. Overall Performance

We evaluate the proposed algorithm on five benchmark datasets, including OTB-2013, OTB-2015, VOT-2015, VOT-2016, and Temple color-128. The proposed algorithm is compared with the state-of-the-art trackers, including the correlation filters based trackers, such as SRDCF [9], Staple [1], MCPF [48], CCOT [10], ECO [7], BACF [17], DRT [38], STRCF [22], and MCCT-H [43]; the non-real-time deep trackers such as MDNet [29], CREST [35], L-



(a) Results on the OTB-2013 dataset



(b) Results on the OTB-2015 dataset

Figure 5. **Success and precision plots on the OTB-2013 and OTB-2015 datasets.**

SART [37], FlowT [52], DSLT [25], MetaSDNet [30], VITAL [36], and DAT [31]; and the real-time deep trackers such as ACT [4], TRACA [5], SiamFC [2], CFNet [40], D-SiamM [11], RASNet [44], SA-Siam [13], SiamRPN [21], and DaSiamRPN [51]. In the following, we will present the results and analyses on each dataset.

OTB dataset. The OTB-2013 dataset with 50 sequences and the extended OTB-2015 dataset with additional 50 sequences are two widely used tracking benchmarks. The sequences in the OTB datasets are with a wide variety of tracking challenging, such as illumination variation, scale variation, deformation, occlusion, fast motion, rotation, and background clutters. The OTB benchmark adopts Center Location Error (CLE) and Overlap Ratio (OR) as the base metrics [45]. Based on CLE and OR, the precision and success plots are used to evaluate the overall tracking performance. The precision plot measures the percentage of frames whose CLE is within a given threshold, which is usually set to 20 pixels. The success plot computes the percentage of the successful frames whose OR is larger than a given threshold. The area under the curve (AUC) of the success plot is mainly used to rank tracking algorithms.

Table 1 shows the AUC score and the running speed of the three categories of trackers on the OTB-2013 and OTB-2015 datasets. In the group of real-time trackers, the proposed algorithm achieves the best performance on both the OTB-2013 dataset (AUC score: 0.680) and the OTB-2015 dataset (AUC score: 0.660). Compared with the state-of-the-art Siamese trackers with offline training, the proposed algorithm achieves the best performance on the OTB-2015 dataset. This is because the proposed target-aware deep features best exploits the objectness and semantic information of the targets and are robust to their appearance variations as well as scale changes. The correlation filters based trackers (DRT and ECO) achieve top performance among all the compared trackers due to the benefits from the multi-feature fusion and online updating schemes. Non-real-time deep trackers all achieve good AUC scores. However, they suffer from time-consuming online training and model overfitting. Equipped with the concise Siamese framework and a small set of deep features, the proposed algorithm achieves a real-time tracking speed (33.7 FPS). This demonstrates the effectiveness of the proposed target-aware features, as the performance of the Siamese tracking framework solely hinges on the discriminative power of features. Figure 5 shows the favorable performance of the proposed tracker against the state-of-the-art real-time trackers. For concise representation, we only show the real-time trackers ($\geq$25 FPS) in this figure, and the complete results of other trackers can be found in Table 1.

Table 2. **Experimental results on the VOT-2015 dataset.** The notation (*) indicates the number is reported by the authors.

Tracker	EAO ↑	Accuracy ↑	Failure↓	FPS
SiamFC [2]	0.2915	0.54	1.42	49
Staple [1]	0.30	0.57	1.39	50
SA-Siam [13]	0.31	0.59	1.26	50*
EBT [50]	0.313	0.45	1.02	4.4*
DeepSRDCF [8]	0.3181	0.56	1.0	1*
FlowT [52]	0.3405	0.57	0.95	12*
Ours	**0.318**	**0.59**	**1.07**	**33.7**

VOT dataset. We validate the proposed tracker on the VOT-2015 dataset. The dataset contains 60 short sequences with various challenges. The VOT benchmark evaluates a tracker from two aspects: robustness and accuracy, which are different from the OTB benchmark. The robustness of a tracker is measured by the failure times. A failure is detected when the overlap ratio between the prediction and the ground truth becomes zero. After 5 frames of the failure, the tracker is re-initialized to track the targets. The accuracy of a tracker is measured by the average overlap ratio between the predicted results and the ground truths. Based on these two metrics, Expected Average Overlap (EAO) is used for overall performance ranking.

Table 2 shows the experimental results on the VOT-2015 dataset. The proposed tracker performs favorably against the state-of-the-art trackers on this dataset. We achieves the second-best EAO score (0.318) with the best accuracy (0.59) and a favorable robustness score (1.07) close to the best one (0.95). FlowTrack equipped with optical flow achieves the best EAO score (0.3405). However, it runs

at a slow speed (12 FPS) when compared to the proposed tracker (33.7 FPS). Overall, the proposed tracker performs well in terms of accuracy, robustness, and running speed. It is worth noting that the favorable performance is achieved without an online update or offline training. This demonstrates the effectiveness of the proposed deep features with target-active and scale-sensitive information, which helps to distinguish between the target objects and the background.

Temple color-128 dataset. We report the results on the Temple color-128 dataset, which includes 128 color sequences and uses the AUC score as the evaluation metric. Table 3 shows that the proposed algorithm achieves the best performance among the real-time trackers with an AUC score of 0.562. The proposed tracker is not specially designed for these color sequences and does not exploit additional online adaption schemes, while it achieves a favorable performance and runs at real-time. This shows the generalization ability of the proposed algorithm.

Table 3. **Experimental results on the Temple color-128 dataset.** The notation (*) indicates the number is reported by the authors.

Method	overlap-AUC	Real-time	FPS
MCPF [48]	0.545	N	1*
STRCF [22]	0.553	N	6
C-COT [10]	0.567	N	1*
MDNet [29]	0.590	N	1
ECO [7]	0.600	N	3
STRCF-deep [22]	0.601	N	3
STAPLE [1]	0.498	Y	50
BACF [17]	0.52	Y	35*
ECO-HC [7]	0.552	Y	30
Ours	**0.562**	**Y**	**33.7**

5.3. Ablation Studies

In this section, we analyze the proposed method on the OTB datasets, including the OTB-2013 and OTB-2015 datasets, to study the contributions of different losses and different layer features. Table 4 presents the overlap ratio in terms of AUC scores of each variation. The features from the output of the *Conv4-3* and *Conv4-1* layers are denoted as Conv4-3 and Conv4-1, respectively. We compare the results of different feature layers based on regression loss, ranking loss, and random selection (randomly selecting the same number of filters), which are denoted as Regress, Rank, and Rand, respectively. Compared with the random selection model, the regression loss scheme obtains significant gains in AUC scores for both the *Conv4-1* (+4.3% and +4.4%) and *Conv4-3* (+4.9% and +3.4%) on the OTB-2013 and OTB-2015 datasets. We attribute these gains to the benefits from the regression loss, which helps to select the most effective convolution filters to generate target-aware discriminative features. By exploiting the objectness and semantic information pertaining to the target, the generated features are effective in distinguishing the target from the background and are robust to target variations. The combination of regression-loss guided features from the *Conv4-1* and *Conv4-3* layers slightly improves the performance (+0.7% and +0.7%) on these two datasets. This shows that although from different layers, these filters guided with the same loss do not provide much complementary information. When combining different CNN layer guided by different losses, the improvement becomes larger (+1.8% and +1.6%). The improvement benefits from the scale-sensitive information of the ranking-loss based features, which puts more emphasis on spatial details. The comparison on the last two rows in Table 4 demonstrates the effectiveness of the ranking loss.

Table 4. **Ablation studies on the OTB dataset.**

Conv4-1	Conv4-3	OTB-2013	OTB-2015
Rand	–	0.602	0.597
–	Rand	0.618	0.610
Regress	–	0.645	0.646
–	Regress	0.662	0.644
Regress	Regress	0.669	0.651
Regress+Rank	Regress	0.680	0.660

6. Conclusions

In this paper, we propose to learn target-aware features to narrow the gap between pre-trained classification deep models and tracking targets of arbitrary forms. Our key insight lies in that gradients induced by different losses indicate the importance of the corresponding filters in recognizing target objects. Therefore, we propose to learn target-aware deep features with a regression loss and a ranking loss by selecting the most effective filters from pre-trained CNN layers. We integrate the target-aware feature model with a Siamese tracking framework and demonstrate its effectiveness and efficiency for visual tracking. In summary, we provide a novel way to handle the problems when using pre-trained high-dimensional deep features to represent tracking targets. Extensive experimental results on five public datasets demonstrate that the proposed algorithm performs favorably against the state-of-the-art trackers.

Acknowledgments

This work is supported in part by the NSFC (No. 61672183), the NSF of Guangdong Province (No. 2015A030313544), the Shenzhen Research Council (No. JCYJ20170413104556946, JCYJ20170815113552036, JCYJ20160226201453085), the Shenzhen Medical Biometrics Perception and Analysis Engineering Laboratory, the NSF CAREER Grant No.1149783, and gifts from

Adobe, Verisk, and NEC. Li X. is supported by a scholarship from China Scholarship Council.

References

[1] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr. Staple: Complementary learners for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 6, 7, 8

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision Workshops*, 2016. 2, 3, 6, 7

[3] Goutam Bhat, Joakim Johnander, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Unveiling the power of deep tracking. In *European Conference on Computer Vision*, 2018. 3

[4] Boyu Chen, Dong Wang, Peixia Li, Shuang Wang, and Huchuan Lu. Real-time actor-criticTracking. In *European Conference on Computer Vision*, 2018. 6, 7

[5] Jongwon Choi, Hyung Jin Chang, Tobias Fischer, Sangdoo Yun, Kyuewang Lee, Jiyeoup Jeong, Yiannis Demiris, and Jin Young Choi. Context-aware deep feature compression for high-speed visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 6, 7

[6] Jongwon Choi, H Jin Chang, Sangdoo Yun, Tobias Fischer, Yiannis Demiris, J Young Choi, et al. Attentional correlation filter network for adaptive visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[7] Martin Danelljan, Goutam Bhat, F Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3, 6, 8

[8] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Convolutional features for correlation filter based visual tracking. In *IEEE International Conference on Computer Vision Workshops*, 2015. 7

[9] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 6

[10] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *European Conference on Computer Vision*, 2016. 3, 6, 8

[11] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning dynamic siamese network for visual object tracking. In *IEEE International Conference on Computer Vision*, 2017. 3, 6, 7

[12] Bohyung Han, Jack Sim, and Hartwig Adam. Branchout: Regularization for online ensemble tracking with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[13] Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng. A twofold siamese network for real-time object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3, 6, 7

[14] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *European Conference on Computer Vision*, 2016. 1, 3

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

[16] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. Online tracking by learning discriminative saliency map with convolutional neural network. In *International Conference on Machine Learning*, 2015. 1, 3

[17] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3, 6, 8

[18] Matej Kristan, Aleš Leonardis, Jiři Matas, Michael Felsberg, Roman Pflugfelder, Luka Čehovin, Tomáš Vojír?, Gustav Häger, Alan Lukežič, Gustavo Fernández, et al. The visual object tracking vot2016 challenge results. In *European Conference on Computer Vision Workshops*, 2016. 2

[19] Matej Kristan, Jiri Matas, Ale Leonardis, Michael Felsberg, Luka Cehovin, Gustavo Fernandez, Toma Vojir, Gustav Hager, Georg Nebehay, Roman Pflugfelder, et al. The visual object tracking vot2015 challenge results. In *IEEE International Conference on Computer Vision Workshops*, 2015. 2

[20] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomas Vojir, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2137–2155, 2016. 2

[21] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3, 6, 7

[22] Feng Li, Cheng Tian, Wangmeng Zuo, Lei Zhang, and Ming-Hsuan Yang. Learning spatial-temporal regularized correlation filters for visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 6, 8

[23] Yuncheng Li, Yale Song, and Jiebo Luo. Improving pairwise ranking for multi-label image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 5

[24] Pengpeng Liang, Erik Blasch, and Haibin Ling. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Transactions on Cybernetics*, 24(12):5630–5644, 2015. 2

[25] Xiankai Lu, Chao Ma, Bingbing Ni, Xiaokang Yang, Ian Reid, and Ming-Hsuan Yang. Deep regression tracking with shrinkage loss. In *European Conference on Computer Vision*, 2018. 6, 7

[26] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *IEEE International Conference on Computer Vision*, 2015. 3

[27] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 2, 3

[28] Matthias Mueller, Neil Smith, and Bernard Ghanem. Context-aware correlation filter tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1396–1404, 2017. 3

[29] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 2, 3, 6, 8

[30] Eunbyung Park and Alexander C Berg. Meta-tracker: Fast and robust online adaptation for visual object trackers. In *European Conference on Computer Vision*, 2018. 7

[31] Shi Pu, Yibing Song, Chao Ma, Honggang Zhang, and Ming-Hsuan Yang. Deep attentive tracking via reciprocative learning. In *Annual Conference on Neural Information Processing Systems*, 2018. 6, 7

[32] Yuankai Qi, Shengping Zhang, Lei Qin, Hongxun Yao, Qingming Huang, Jongwoo Lim, and Ming-Hsuan Yang. Hedged deep tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 3

[33] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 3, 4

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 6

[35] Yibing Song, Chao Ma, Lijun Gong, Jiawei Zhang, Rynson WH Lau, and Ming-Hsuan Yang. Crest: Convolutional residual learning for visual tracking. In *IEEE International Conference on Computer Vision*, pages 2574–2583, 2017. 1, 2, 3, 6

[36] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson WH Lau, and Ming-Hsuan Yang. Vital: Visual tracking via adversarial learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 3, 6, 7

[37] Chong Sun, Huchuan Lu, and Ming-Hsuan Yang. Learning spatial-aware regressions for visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 6, 7

[38] Chong Sun, Dong Wang, Huchuan Lu, and Ming-Hsuan Yang. Correlation tracking via joint discrimination and reliability learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 6

[39] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders. Siamese instance search for tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3

[40] Jack Valmadre, Luca Bertinetto, João Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 6, 7

[41] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. In *Proceeding of the ACM International Conference on Multimedia*, 2015. 6

[42] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu. Visual tracking with fully convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 3

[43] Ning Wang, Wengang Zhou, Qi Tian, Richang Hong, Meng Wang, and Houqiang Li. Multi-cue correlation filters for robust visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 6

[44] Qiang Wang, Zhu Teng, Junliang Xing, Jin Gao, Weiming Hu, and Stephen Maybank. Learning attentions: residual attentional siamese network for high performance online visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3, 6, 7

[45] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 2, 7

[46] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015. 2

[47] Sangdoo Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, and Jin Young Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[48] Tianzhu Zhang, Changsheng Xu, and Ming-Hsuan Yang. Multi-task correlation particle filter for robust object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 6, 8

[49] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3, 4

[50] Gao Zhu, Fatih Porikli, and Hongdong Li. Beyond local search: Tracking objects everywhere with instance-specific proposals. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 7

[51] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *European Conference on Computer Vision*, 2018. 6, 7

[52] Zheng Zhu, Wei Wu, Wei Zou, and Junjie Yan. End-to-end flow correlation tracking with spatial-temporal attention. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3, 6, 7