VITAL: VIsual Tracking via Adversarial Learning

Yibing Song¹, Chao Ma²*, Xiaohe Wu³, Lijun Gong⁴, Linchao Bao¹, Wangmeng Zuo³, Chunhua Shen², Rynson W.H. Lau⁵, and Ming-Hsuan Yang⁶

¹Tencent AI Lab  ²The University of Adelaide  ³Harbin Institute of Technology  ⁴Tencent
⁵City University of Hong Kong  ⁶University of California, Merced

https://ybsong00.github.io/cvpr18_tracking/index

Abstract

The tracking-by-detection framework consists of two stages, i.e., drawing samples around the target object in the first stage and classifying each sample as the target object or as background in the second stage. The performance of existing trackers using deep classification networks is limited by two aspects. First, the positive samples in each frame are highly spatially overlapped, and they fail to capture rich appearance variations. Second, there exists extreme class imbalance between positive and negative samples. This paper presents the VITAL algorithm to address these two problems via adversarial learning. To augment positive samples, we use a generative network to randomly generate masks, which are applied to adaptively dropout input features to capture a variety of appearance changes. With the use of adversarial learning, our network identifies the mask that maintains the most robust features of the target objects over a long temporal span. In addition, to handle the issue of class imbalance, we propose a high-order cost sensitive loss to decrease the effect of easy negative samples to facilitate training the classification network. Extensive experiments on benchmark datasets demonstrate that the proposed tracker performs favorably against state-of-the-art approaches.

1. Introduction

There has been an increasing need for tracking target objects in bounding boxes to understand video contents. Current state-of-the-art trackers are typically based on a two-stage tracking-by-detection framework. The first stage draws a sparse set of samples around the target object and the second stage classifies each sample as either the target object or as the background using a deep neural network. Despite the favorable performance on recent tracking benchmarks [59, 60, 31], the performance of the two-stage methods is limited by two aspects. First, the positive samples are spatially overlapped, and they cannot capture a variety of appearance changes over time. Second, the extreme foreground-background class imbalance negatively affects training the classification networks. It is of great importance to investigate how to eliminate these barriers to advance the tracking-by-detection framework in the deep learning era.

Prior trackers have made limited efforts on increasing the diversity of training data in learning deep classifiers. Since classifiers tend to learn a discriminative boundary between positive and negative samples, they emphasize on the
most discriminative ones. However, as the target appearance varies frame-by-frame in the whole video sequence, the most discriminative samples in the current frame may not persist over a long temporal span. Typical examples of appearance changes caused by partial occlusion or out-of-plane rotation easily result in model overfitting, as current training samples may differ much from the previous ones. To alleviate this problem, existing trackers incrementally update the classifier through online sample collections. The noisy updates occur and bring tracker drift problem. Hence, a natural question is how can we augment positive samples in the feature space to capture target appearance variations in the temporal domain.

In this work, we take advantage of the recent progress in adversarial learning to augment training data to facilitate classifier training. For a deep classification network, such as the VGG-M model [47], we add a generative network between the last convolutional layer and the first fully connected layer. The generative network augments positive samples by generating weight masks randomly applied to the features, where each mask represents a specific type of appearance variation. Through adversarial learning, our network can identify the mask that maintains the most robust features of target appearance in the temporal domain. We show that the learned mask tends to decrease the weights of discriminative features, which tends to overfit in a single frame. Meanwhile, these features are hardly robust to appearance changes over the temporal span. In other words, adversarial learning helps our tracker exploit the most robust features over a long temporal span in classifier training, rather than overfitting to discriminative features in a single frame. Moreover, to mitigate the issue of class imbalance, we propose a high-order cost sensitive loss to decrease the effect of easy negative samples. Taking advantages of adversarial learning and high-order cost sensitive loss, our tracking method achieves favorable results against state-of-the-art trackers.

We summarize the main contributions of this work as follows:

- We propose to use a generative adversarial network (GAN) to augment positive samples in the feature space to capture a variety of appearance changes over a temporal span.
- We propose to use higher-order cost sensitive loss to mine hard negative samples to handle class imbalance.
- We extensively validate our method on benchmark datasets with large-scale sequences. We show that our VITAL tracker performs favorably against state-of-the-art trackers.

2. Related Work

Visual tracking has long been an active research topic with extensive surveys [48] over the last decade. In this section, we mainly discuss the representative visual trackers and the related issues on generative adversarial learning and class imbalance.

**Visual Tracking.** Visual tracking has a wide range of applications including action recognition [7], target analysis [52, 51, 49] and augmented reality [8, 44]. State-of-the-art trackers are mainly based on the one-stage regression framework or the two-stage classification framework. As one of the most representative types of the one-stage regression framework, the correlation filter based trackers regress all the circular-shifted version of the input features into soft labels generated by a Gaussian function. By computing the correlation as an element-wise product in the Fourier domain, these trackers have received a lot of attention recently. Starting from the MOSSE tracker [5], many efforts have been made to improve the correlation filter for robust tracking. Extensions include, but are not limited to, kernelized correlation filters [23], scale estimation [10], re-detection [37], spatial regularization [12, 14, 9], ADMM optimization [29], sparse representation [32, 42, 33], CNN feature integrations [36, 43, 64, 28] and end-to-end CNN predictions [57, 55, 50].

In contrast, the two-stage classification framework poses the tracking task as a binary classification problem. The two-stage trackers emphasize on a discriminative boundary between the samples of the target object and background. Numerous learning schemes are proposed including P-N learning [27], multiple instance learning [2], structured SVMs [20, 39], CNN-SVMs [24], domain adaptation [38], and ensemble learning [19]. Unlike the existing two-stage tracking-by-detection trackers, our method, for the first time, takes advantage of the recent progress in generative adversarial learning to augment training samples in the feature space. The augmented samples capture a variety of appearance changes and thus strengthen the robustness of the classifier. In addition, we exploit hard negative samples to handle class imbalance limitation.

**Generative Adversarial Learning.** It is introduced in [17] to generate realistic-looking images from random noise via the CNN. The generative adversarial network (GAN) consists of two subnetworks. One serves as a generator and the other as a discriminator. The generator aims at synthesizing images to fool the discriminator, while the discriminator tries to discriminate between real images and images synthesized by the generator. The generator and the discriminator are trained simultaneously by competing with each other. An advantage of adversarial learning is that the generator is trained to produce similar image statistics to those of the training samples so that the discriminator cannot differentiate. This manner is hardly achieved by existing empirical objective functions with supervised learning. The progress in generative adversarial learning has attracted a
Prediction: Binary classification
Training: Classification loss
Fully Connected Layers
Weight Mask

Second, we need to train the classifier via supervised learning using labeled samples rather than random noise. Unlike existing GAN-based methods, which expect to obtain generator mapping samples from one distribution to another after the training process, we expect to obtain a discriminator which is robust to target object variations. Fig. 2 shows the pipeline of our method, and the details are discussed below.

3.1. Adversarial Learning

In the traditional adversarial learning [17], the generator G takes a noise vector z from a distribution P_{noise}(z) as an input and outputs an image G(z). The discriminator D takes either G(z) or a real image x with a distribution P_{data}(x) as an input and outputs the classification probability. The generator G is learned to maximize the probability of D making a mistake. Using the standard cross entropy loss, the objective loss function for training G and D is defined as:

$$L = \min_{G} \max_{D} \mathbb{E}_{x \sim P_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim P_{noise}(z)}[\log(1 - D(G(z)))], \quad (1)$$

where the G and D networks are trained simultaneously. The training encourages G to fit P_{data}(x) so that D will not be able to discriminate x from G(z). Note that in Eq. 1, there are no ground truth annotations for z and the learning process is unsupervised. After the training process, G is removed and only D is kept for inference.

Although GANs have been investigated in many computer vision tasks, a direct applying of Eq. 1 in the tracking-by-detection framework is not feasible. First, the input data to the framework are usually candidate object proposals rather than random noise. Second, we need to train the classifier via supervised learning using labeled samples rather
than unlabeled ones. Third, we expect to use the classifier (i.e., $D$) for inference rather than $G$. These three factors limit the usage of GANs on visual tracking where both the input and learning strategy differ significantly.

We propose VITAL to narrow the gap between GANs and the tracking-by-detection framework. We add $G$ between feature extraction and the classifier as shown in Fig. 2. $G$ will predict a weight mask which operates on the extracted features. This mask is set randomly at the beginning and gradually identifies the discriminative features through adversarial learning. We define the input feature as $C$, the mask generated by the $G$ network as $G(C)$, the actual mask identifying the discriminative features as $M$. We define the objective function as:

$$L_{VITAL} = \min_G \max_D \mathbb{E}_{(C,M)\sim P(C,M)} [\log D(M \cdot C)] + \mathbb{E}_{C\sim P(C)} [\log(1 - D(G(C) \cdot C))] + \lambda \mathbb{E}_{(C,M)\sim P(C,M)} ||G(C) - M||^2,$$

(2)

where the dot is the dropout operation on the feature $C$. The mask contains only one channel and has the same resolution as $C$. We express the predicted mask as $\hat{M}$ and the value of the element $(i, j)$ as $\hat{M}_{ij}$. Meanwhile, we define the value of the element $(i, j, k)$ on feature $C$ as $C_{ijk}$. The dropout operation is defined as follows:

$$C_{ijk}^\circ = C_{ijk} \hat{M}_{ij},$$

(3)

where $C_{ijk}^\circ$ is the feature $C$ after the dropout operation and passed onto the classifier.

In Eq. 2, we integrate the adversarial learning into the tracking-by-detection framework. We keep the input (i.e., the candidate object proposals) unchanged. When training $D$ (i.e., classifier), we extract features and enrich their representations in the feature space. Instead of empirically proposing data augmentation strategies, we let $G$ to identify the discriminative features, which are crucial for training $D$. Initially, $G$ produces several random masks, which are akin to the random noise in Eq. 1. Each mask represents a specific type of appearance variation, and we expect these masks to cover the whole object variations. Through the adversarial learning process, $G$ will gradually identify the mask that degrades the classifier most. This indicates that the mask has identified the discriminative features. On the other hand, $D$ will gradually be trained without overfitting to the discriminative features from individual frames while relying on more robust features over a long temporal span.

In each iteration of the adversarial learning, we first train $D$ and then $G$. The detailed training procedure is presented in the following:

Training $D$. In one iteration of the training process, we pass the input feature through $G$ and obtain the predicted mask $\hat{M}$. We then conduct the dropout operation on this feature and send the modified feature into $D$. We keep the labels unchanged and train $D$ through supervised learning. Note that during this training process, there are multiple input features. $G$ will predict different masks according to different input features. It enables $D$ to focus on the temporal robust features without discriminative feature interference from single frames.

Training $G$. After training $D$ once, given an input feature, we create multiple output features based on several random masks. This feature diversifying process is performed through the dropout operation illustrated in Eq. 3. These features are passed onto $D$, and we pick up the one with the highest loss. The corresponding mask of the selected feature is said to be effective in decreasing the impact of the discriminative features. We set this mask as $M$ in Eq. 2 and update $G$ accordingly.

Visualization. Adversarial learning enables the classifier to focus on the temporal robust features instead of the discriminative ones in individual frames. Fig. 3 shows an example of how adversarial learning affects the classifier in practice. Fig. 3(a) shows the input frame with the ground truth annotation located at the face region. We use our VITAL tracker to represent the tracking-by-detection framework for illustration. We compute the entropy distribution based on the
predicted probabilities from the classifier. The entropy measures the uncertainty of the prediction and is computed for binary classification as:

$$H = -(p \cdot \log p + (1 - p) \cdot \log(1 - p)),$$  \hspace{1cm} (4)$$

where $p$ is the predicted probability of the target object and $1 - p$ is the background. When $p = 0.5$, the value of the entropy $H$ is highest, which means that the classifier is uncertain to predict the label. When $p = 0$ or $p = 1$, the value of the entropy $H$ is lowest, which means that the classifier is certain about the prediction.

We compute the entropy distribution of Fig. 3(a) using VITAL without adversarial learning as shown in Fig. 3(b) and with adversarial learning as shown in Fig. 3(c). We note that these two distributions are similar despite some tiny variances. However, when the target undergoes partial occlusion and out-of-plane rotation as shown in Fig. 3(d), the entropy of VITAL without adversarial learning increases rapidly as shown in Fig. 3(e), which indicates that the classifier becomes uncertain around the target region. This is because the classifier is trained to focus on the discriminative features of the samples in the previous frames. As the target appearance varies in the following frames, these discriminative features vanish and decrease the classification accuracy. In comparison, the entropy distribution shown in Fig. 3(f) does not vary as significant as that in Fig. 3(e). It is because the classifier trained via diversified samples will not focus on the most discriminative features in individual frames. Instead, it tends to focus on more robust features over a long period of time. In sum, with the adversarial learning, VITAL becomes temporally robust while preserving the classification accuracy on individual frames.

### 3.2. Cost Sensitive Loss

We first revisit the cross entropy (CE) loss for binary classification. Formally, we define $y \in \{0, 1\}$ as the class labels and $p \in [0, 1]$ as the estimated probability for a class with label $y = 1$. Meanwhile, we define the probability for a class with label $y = 0$ as $1 - p$. The CE loss is formulated as:

$$L(p, y) = -(y \cdot \log(p) + (1 - y) \cdot \log(1 - p)),$$  \hspace{1cm} (5)$$

One notable problem of the CE loss is that easy negative samples, i.e., when $p \ll 0.5$ and $y = 0$, produce the loss with non-trivial magnitude. When summed over a large number of easy negative examples, these small loss values overwhelm the valuable rare positive class. In visual tracking, class imbalance lies between the limited positive samples and a substantial amount of negative samples across the whole background. Easy negative samples take over the majority of the CE loss and dominate the gradient.

Existing solutions to class imbalance include hard negative mining [15, 46] and training data reweighing [45]. The simplest method to make a classifier cost sensitive involves a modification of the class importance. For example, when the ratio of positive and negative classes is 1:100, the importance factor of the negative class is set to be 0.01. Note that simply using a fixed factor to balance the importance of positive/negative examples does not identify the easiness or hardness of each example. We align our motivation to the recently proposed focal loss [35] and add a modulating factor to the CE loss in terms of the network output probability $p$. Formally, we build our cost sensitive loss upon the entropy loss as:

$$L(p, y) = -(y \cdot (1-p) \cdot \log(p) + (1 - y) \cdot p \cdot \log(1 - p)).$$  \hspace{1cm} (6)$$

With the cost sensitive loss, we reformulate the objective function in Eq. 2 as:

$$L_{\text{VITAL}} = \min_G \max_D \mathbb{E}_{(C,M) \sim \mathcal{P}_{(C,M)}} [K_1 \cdot \log D(M \cdot C)] + \mathbb{E}_{C \sim P_C} [K_2 \cdot \log (1 - D(G(C) \cdot C))] + \lambda \mathbb{E}_{(C,M) \sim \mathcal{P}_{(C,M)}} \|G(C) - M\|^2,$$  \hspace{1cm} (7)$$

where $K_1 = 1 - D(M \cdot C)$ and $K_2 = D(G(C) \cdot C)$ are modulating factors that balance the training sample loss.

### 4. Tracking via VITAL

We illustrate how we perform VITAL for visual tracking. Note that we only involve $G$ when training the classifier and remove it in the test stage. The details are as follows:

**Model initialization.** We initialize our model through a two-stage training. In the first step we offline pretrain the model using positive and negative samples from the training data, which is from [38]. In the second step we draw the samples from the first frame of the input sequence to finetune our model online. During offline pretraining, we randomly initialize $D$ and perform the training in a few iterations, then we involve $G$ for adversarial learning. See Sec. 3.1 for the details of the adversarial learning process where only positive samples are adopted. We mine the hard negative samples through the cost sensitive loss for training $D$ together with the diversified positive samples.

**Online detection.** The online detection scheme is the same as existing tracking-by-detection approaches as we remove $G$ in this step. Given an input frame, we first generate multiple candidate proposals and extract their CNN features. We feed the CNN features of the candidate proposals into the classifier to get the probability scores.

**Model update.** We incrementally update our tracker frame-by-frame. Around the estimated position, we generate multiple samples and assign them with binary labels according to their intersection-over-union scores with the estimated bounding box. We use these training samples jointly train $G$ and $D$ during online update as illustrated in Sec. 3.1.
5. Experiments

In this section, we introduce the implementation details of VITAL and analyze the effects of adversarial learning and cost sensitive loss. Then we compare our VITAL tracker with state-of-the-art trackers on the benchmark datasets OTB-2013 [59], OTB-2015 [60] and VOT-2016 [30] for performance evaluation.

Experimental Setup. Our backbone feature extractor is based on the first three convolutional layers of the VGG-M model [47]. When training $G$, we prepare 9 random masks. The resolution of each mask is the same as that of the input features. We split this mask into 9 parts equally. We assign each part with label 1 in turn and the remaining parts with label 0. These masks are different from each other and cover all the parts in total. When training $D$, we apply 9 masks to the input features independently to generate 9 diversified versions of each input feature. We then feed these diversified features into $D$ and select the one with the highest loss. The corresponding mask is denoted by $M$ as illustrated in Eq. 2 to train $D$. During the adversarial learning, we iteratively apply the SGD solver to both $G$ and $D$. We use 100 iterations to initialize both networks. The learning rate for training $G$ and $D$ are $10^{-3}$ and $10^{-4}$, respectively. We update both networks every 10 frames using 10 iterations. Our VITAL tracker runs on a PC with an i7 3.6GHz CPU and a Tesla K40c GPU with the MatConvNet toolbox [56] and the average speed is 1.5 FPS.

Evaluation Metrics. We follow the standard evaluation approaches. In the OTB-2013 and OTB-2015 datasets we use the one-pass evaluation (OPE) with precision and success plots metrics. The precision metric measures the frame locations rate within a certain threshold distance from ground truth locations. The threshold distance is set as 20 pixels. The success plot metric is set to measure the overlap ratio between the predicted bounding boxes and the ground truth. In the VOT-2016 dataset [30], we measure the performance in terms of Expected Average Overlap (EAO), Accuracy Ranks (Ar) and Robustness Ranks (Rr).

Ablation Studies. In VITAL, we train the classifier using the diversified positive samples with a cost sensitive loss. To validate the effectiveness of each component, we first implement a baseline algorithm by not enabling the adversarial training and using the standard cross entropy loss. We implement three alternative approaches based on the baseline algorithm. First, we train the classifier by generating random masks. Second, we train the classifier using adversarial learning (i.e., GAN). Third, we train the classifier using adversarial learning with the cost sensitive loss. Fig. 4 shows the results on the OTB-2013 dataset. We observe that using random masks deteriorates the classifier and results in inferior performance. It is because the spatial discriminative and temporal robust features are blocked randomly, which degrades the classifier to focus on either. In contrast, the mask predicted by adversarial learning effectively exploits the most robust features by blocking partial discriminative features in individual frames. The cost sensitive loss further improves the performance. However, the improvement of the cost sensitive loss is not as salient as that of the adversarial learning.

OTB-2013 Dataset. We compare VITAL with 29 trackers from the OTB-2013 benchmark [59] and other 28 state-of-the-art trackers including DSST [10], KCF [22], TGPR [16], MEEM [63], RPT [34], LCT [37], MUSTer [25], HCFT [36], FCNT [57], SRDCF [12], CNN-SVM [24], DeepSRDCF [11], DAT [41], Staple [3], SRDCFdecon [13], CCOT [14], GOTURN [21], SINT [54], SiamFC [4], HDT [43], SCT [6], MDNet [38], DLS-SVM [39], ADNet [61], ECO [9], MCPF [64], CFNet [55] and
Figure 6: Overlap success plots over eight tracking challenges of illumination variation, deformation, in-plane rotation, out-of-plane rotation, background clutter, occlusion, scale variation and low resolution.

CREST [50]. We evaluate all the trackers on 50 video sequences using the one-pass evaluation with distance precision and overlap success metrics.

Figure 5 shows the results from all compared trackers. For presentation clarity, we only show the top 10 trackers. The numbers listed in the legend indicate the AUC overlap success and 20 pixel distance precision score values. Overall, our VITAL tracker performs favorably against state-of-art trackers in both distance precision and overlap success. Figure 6 compares the performance under eight video attributes using one-pass evaluation. Our VITAL tracker handles large appearance variations well caused by deformation, in-plane and out-of-plane rotations. Compared to the representative tracking-by-detection tracker MDNet, we attribute our performance improvement by the diversified positive samples for training robust classifiers. The mask generated via adversarial learning captures a variety of object attributes. It masks out the discriminative features in individual frames while maintaining the most robust features over a long temporal span. The advantage of exploiting the temporally robust features is clearly proved when dealing with occlusion. Through focusing on the persistently robust features, our VITAL tracker performs better than MDNet in a large margin.

Meanwhile, our cost sensitive loss effectively decreases the loss from easy negative samples and forces the classifier to focus on hard ones. This facilitates discriminative classifiers to separate the target object from background. Our VITAL achieves leading performance in the presence of illumination variation and background clutter. However, for the low resolution sequences, our tracker does not perform as well as MDNet. This is because the target size of these sequences is small and the resolution of the weight masks predicted by adversarial learning is far low.

Table 1: Comparison with the state-of-the-art trackers on the VOT 2016 dataset. The results are presented in terms of expected average overlap (EAO), accuracy rank (Ar) and robustness rank (Rr).

<table>
<thead>
<tr>
<th></th>
<th>ECO</th>
<th>CCOT</th>
<th>Staple</th>
<th>MDNet</th>
<th>VITAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAO</td>
<td>0.374</td>
<td>0.331</td>
<td>0.295</td>
<td>0.257</td>
<td>0.323</td>
</tr>
<tr>
<td>Ar</td>
<td>1.55</td>
<td>1.63</td>
<td>1.65</td>
<td>1.63</td>
<td>1.63</td>
</tr>
<tr>
<td>Rr</td>
<td>1.57</td>
<td>1.70</td>
<td>2.67</td>
<td>2.4</td>
<td>2.17</td>
</tr>
</tbody>
</table>

For the scale variance sequence, the fixed size of weight mask cannot precisely mask out the discriminative features as the object size increases. Our future work will consider adaptively changing the size of the weight mask.

OTB-2015 Dataset. We compare our VITAL tracker on the OTB-2015 benchmark [60] with the state-of-the-art trackers. Figure 7 shows that our VITAL tracker overall performs well. The ECO tracker achieves the best result in overlap success, while our VITAL ranks first in distance precision. Since the OTB-2015 dataset contains more videos with large scale changes and low resolution, our VITAL tracker does not perform as well as ECO in overlap success.

VOT-2016 Dataset. We compare our VITAL tracker with state-of-the-art trackers on the VOT-2016 benchmark, including Staple [3], MDNet [38], CCOT [14] and ECO [9]. VOT-2016 report [30] shows that the strict state-of-the-art bound is 0.251 under EAO metric. Trackers whose EAO value exceeds this bound is defined as state-of-the-art. Table 1 shows that ECO performs best under the EAO metric. The performance of VITAL is comparable to that of CCOT and better than Staple and MDNet. According to the definition of the VOT report, all these trackers are state-of-the-art.
Qualitative Evaluation. Fig. 8 qualitatively compare the results of the top performing trackers: CNN-SVM [24], CCOT [14], MDNet [38], ECO [9] and VITAL on 12 challenging sequences. In a majority of these sequences, CNN-SVM fails to locate the target objects or estimates scale incorrectly because of the limited performance of the SVM classifier. MDNet improves CNN-SVM through an end-to-end CNN network formulation. It performs well on deformation (Trans), low resolution (Skiing) and fast motion (Diving). However, the classifier of MDNet is trained to focus on the discriminative features from individual frames, which may lead to overfitting in the presence of noisy update. It does not perform well in handling out-of-plane rotation (Ironman) and occlusion (Human4). The correlation filter based trackers (i.e., CCOT and ECO) extract CNN features and learn correlation filters independently. They do not take full advantage of the end-to-end deep architecture. In contrast, our VITAL tracker emphasizes on the most temporally robust features. The adversarial learning scheme makes the classifier aware a variety of appearance changes. The cost sensitive loss mines hard negative samples to further facilitate classifier learning. Our tracker VITAL performs favorably against state-of-the-art trackers.

6. Conclusion

In this paper we integrate adversarial learning into the tracking-by-detection framework to reduce overfitting on single frames. We adaptively dropout the discriminative features in single frame which draws the classifier attention. It enables the classifier to focus on the temporal robust features which are originally diminished during the training process. The adaptive dropout is achieved via adversarial learning to predict discriminative features according to different inputs. It enriches the target appearances in the feature space and augment the positive samples. Meanwhile, we use the cost sensitive loss to reduce the effect from easy negative samples. Extensive experiments on benchmarks demonstrate that our VITAL tracker performs favorably against state-of-the-art trackers.

Acknowledgements. Xiaohe Wu and Wangmeng Zuo are supported by the National Natural Scientific Foundation of China (NSFC) under Grant No. 61671182. Ming-Hsuan Yang is supported by NSF CAREER (No. 1149783).
References


