VITAL: VIsual Tracking via Adversarial Learning

Yibing Song[1], Chao Ma[2]*, Xiaohe Wu[3], Lijun Gong[4], Linchao Bao[1],
Wangmeng Zuo[3], Chunhua Shen[2], Rynson W.H. Lau[5], and Ming-Hsuan Yang[6]

[1]Tencent AI Lab [2]The University of Adelaide [3]Harbin Institute of Technology [4]Tencent
[5]City University of Hong Kong [6]University 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.

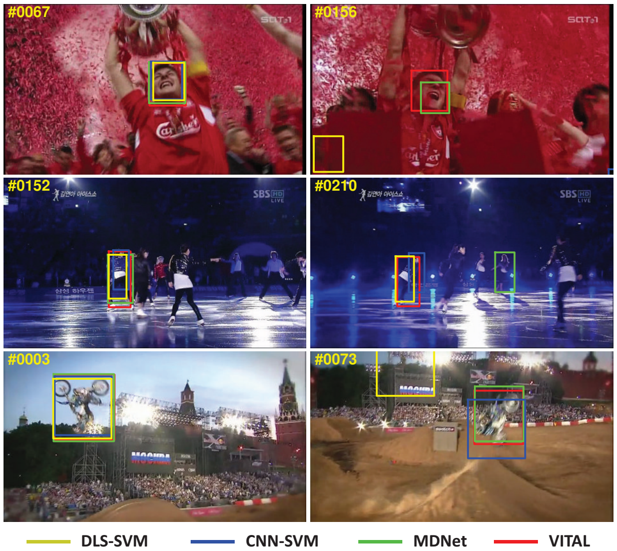


Figure 1: Tracking results with the comparison to state-of-the-art tracking-by-detection trackers including DLS-SVM [39], CNN-SVM [24], and MDNet [38]. Our VITAL tracker learns to diversify positive samples via adversarial learning and to balance training samples via cost sensitive loss. It performs favorably against existing trackers.

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

*C. Ma is the corresponding author.

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 we can 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 discriminates 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

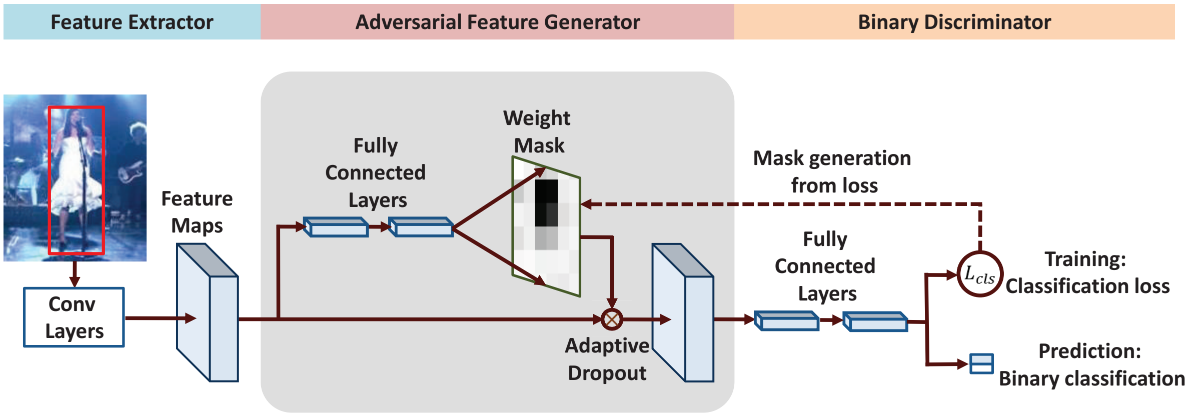


Figure 2: Overview of our network architecture. Our method takes each sampled patch as an input and predicts a possibility score of the patch being the target object. We add one branch of fully connected layers after the last convolutional layer to randomly generate masks, which are applied to the input features to capture a variety of appearance changes. Adversarial learning identifies the mask that maintains the most robust features over a long temporal span while removing the discriminative features from individual frames. It facilitates the classifier training process.

series of works on network training [1, 40, 18] and computer vision applications, such as image generation [62], image stylization [26], object detection [58], and semantic segmentation [53]. Unlike existing GANs that augment data in the image space, we apply adversarial learning to augment training samples in the feature space to capture appearance variations in temporal domain. In sum, our method exploits robust features over the long temporal span, instead of the discriminative features in individual frames.

Class Imbalance. This problem often exists in learning applications, where the amount of training data in one class (usually the positive class) is far less than that of another class (usually the negative class). A large portion of samples from the majority class are easy samples, which dominantly produce a large loss, and make the learning process unaware of the valuable samples from the minority class. Hard negative mining [15, 46] and reweighing training data [45, 35] are useful to alleviate the class imbalance problem to some extent. In visual tracking, class imbalance deteriorates the performance of the classifier, as the number of positive samples are extremely limited but the number of negative samples across the whole background is large. Unlike the aforementioned solutions for the class imbalance problem, we propose cost sensitive loss to decrease the effect from easy negative samples when training the classifier. This not only improves the tracking accuracy, but also accelerates the training convergence.

3. Proposed Algorithm

We build VITAL upon the CNN tracking-by-detection framework, which consists of feature extraction and classification. We interpret the classifier as the discriminator and propose a generator for adversarial learning [17]. 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:

$$\begin{aligned}\mathcal{L} = \min_{G} \max_{D} \ & \mathbb{E}_{x \backsim P_{data}(x)}[\log D(x)] \\ & + \mathbb{E}_{z \backsim P_{noise}(z)}[\log(1 - D(G(z)))],\end{aligned} \tag{1}$$

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:

$$\begin{aligned}\mathcal{L}_{\text{VITAL}} = \min_G \max_D \; & \mathbb{E}_{(C,M)\backsim P_{(C,M)}}[\log D(M \cdot C)] \\ & +\mathbb{E}_{C \backsim P_{(C)}}[\log(1 - D(G(C) \cdot C))] \\ & +\lambda \mathbb{E}_{(C,M)\backsim P_{(C,M)}} ||G(C) - M||^2,\end{aligned} \tag{2}$$

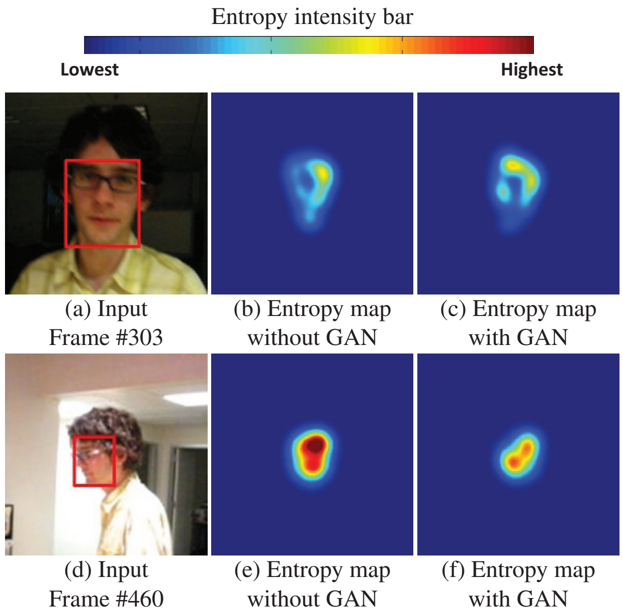
where the dot is the dropout operation on the feature C. The mask contains only one channel and has the same resolution as C. We express the predicted mask as $\hat{M}$ and the value of the element (i, j) as $\hat{M}_{ij}$. Meanwhile, we define the value of the element (i, j, k) on feature C as C_{ijk}.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

$$C^{\text{o}}_{ijk} = C_{ijk}\hat{M}_{ij}, \tag{3}$$

where C^{o}_{ijk} 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 sent 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.



(a) Input Frame #303 (b) Entropy map without GAN (c) Entropy map with GAN

(d) Input Frame #460 (e) Entropy map without GAN (f) Entropy map with GAN

Figure 3: Entropy distribution of two frames on the *David* sequence [59]. We use VITAL with and without GAN integration for comparison. We analyze the entropy based on the predicted probabilities from the classifier. The higher the entropy, the more uncertain the classifier prediction is.

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:

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

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:

$$L(p, y) = -\big(y \cdot \log(p) + (1-y) \cdot \log(1-p)\big). \tag{5}$$

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:

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

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

$$\begin{aligned}\mathcal{L}_{\text{VITAL}} = \min_G \max_D \; & \mathbb{E}_{(C,M)\backsim P_{(C,M)}}[K_1 \cdot \log D(M \cdot C)] \\ & + \mathbb{E}_{C \backsim P_{(C)}}[K_2 \cdot \log(1 - D(G(C) \cdot C))] \\ & + \lambda \mathbb{E}_{(C,M)\backsim P_{(C,M)}} ||G(C) - M||^2,\end{aligned} \tag{7}$$

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.

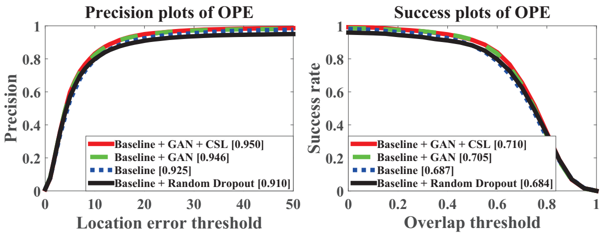


Figure 4: Precision and success plots on the OTB-2013 dataset using the one-pass evaluation. The numbers in the legend indicate the average distance precision scores at 20 pixels and the area-under-the-curve success scores.

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 from 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).

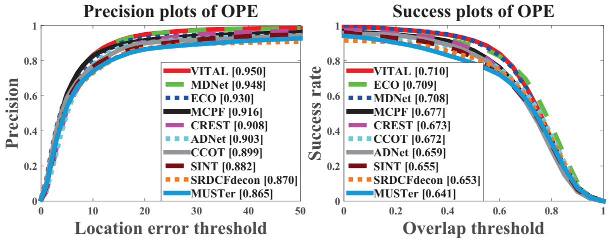


Figure 5: Precision and success plots on the OTB-2013 dataset using one-pass evaluation.

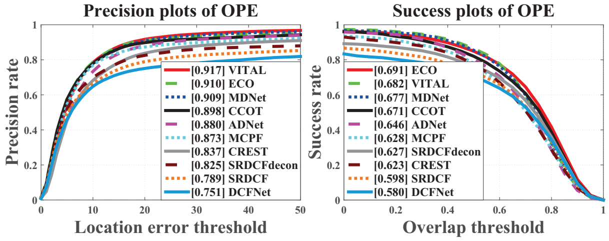


Figure 7: Precision and success plots on the OTB-2015 dataset using one-pass evaluation.

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

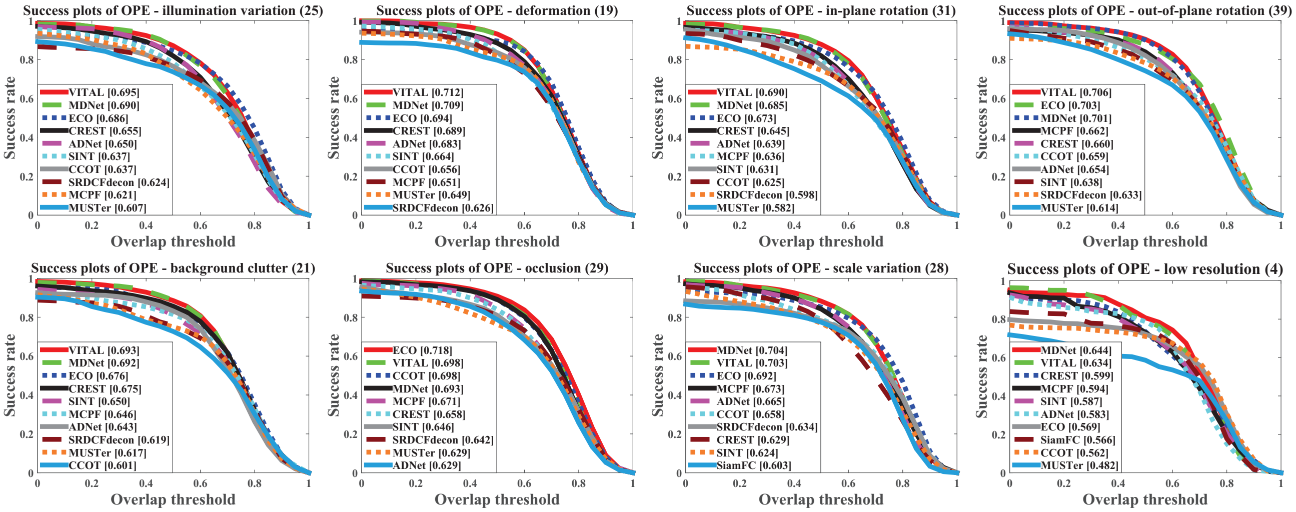


Figure 6: Overlap success plots over eight tracking challenges of illumination variation, deformation, in-plane rotation, out-of-plane rotation, background clutter, occlusion, scale variation and low resolution.

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 scores. 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 variations. It maskouts the discriminative features in individual frames while maintains 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. For the scale variance sequence, the fixed size of weight mask cannot precisely maskout the discriminative features as the object size increases. Our future work will consider adaptively changing the size of the weight mask.

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).

	ECO	CCOT	Staple	MDNet	VITAL
EAO	0.374	0.331	0.295	0.257	0.323
Ar	1.55	1.63	1.65	1.63	1.63
Rr	1.57	1.70	2.67	2.4	2.17

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.

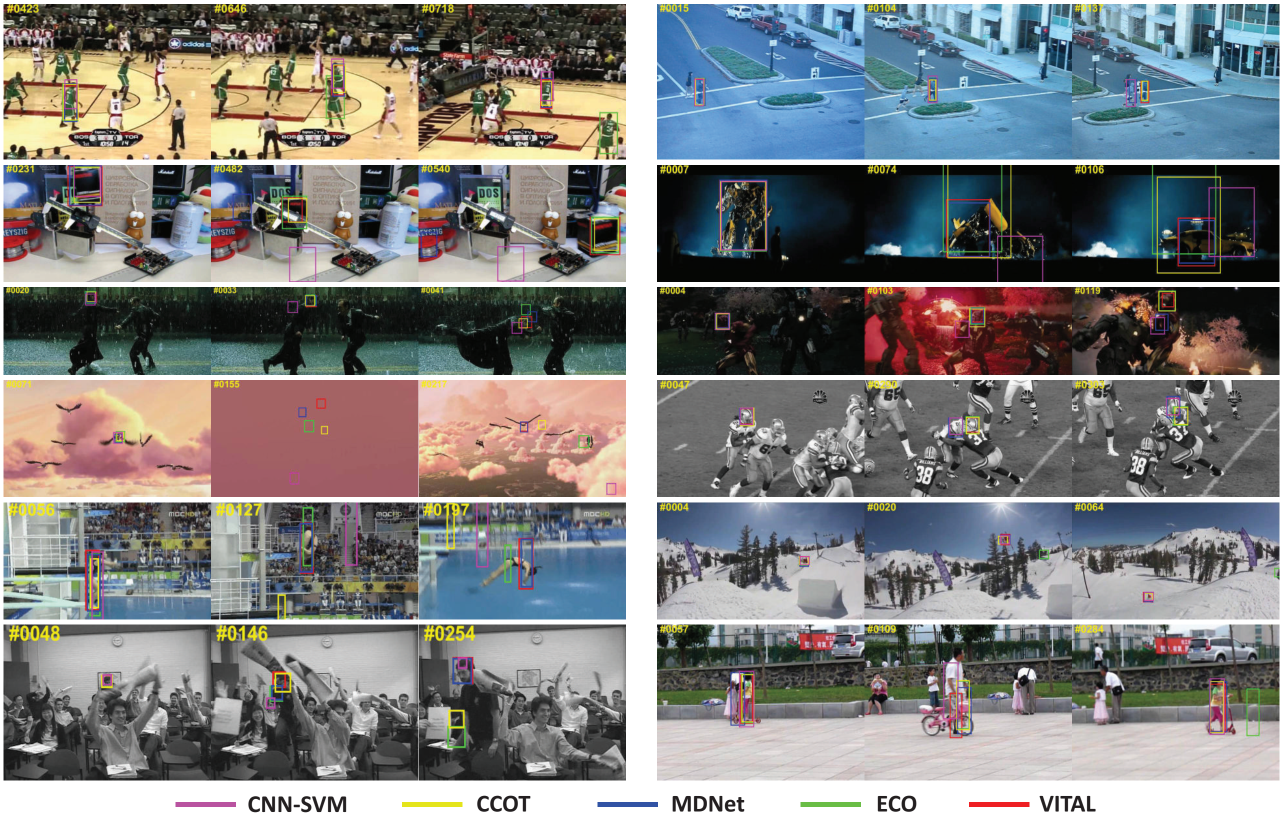


Figure 8: Qualitative evaluation of our VITAL tracker, CNN-SVM [24], CCOT [14], MDNet [38], ECO [9] on 12 challenging sequences (from left to right and top to down: *Basketball*, *Human4*, *Box*, *Trans*, *Matrix*, *Ironman*, *Bird1*, *Football*, *Diving*, *Skiing*, *Freeman4* and *Girl2*, respectively). Our VITAL tracker performs favorably against 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

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, 2017.

[2] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[3] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr. Staple: Complementary learners for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[4] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision Workshop*, 2016.

[5] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[6] J. Choi, H. Jin Chang, J. Jeong, Y. Demiris, and J. Young Choi. Visual tracking using attention-modulated disintegration and integration. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[7] W. Choi and S. Savarese. A unified framework for multi-target tracking and collective activity recognition. In *European Conference on Computer Vision*, 2012.

[8] A. I. Comport, E. Marchand, M. Pressigout, and F. Chaumette. Real-time markerless tracking for augmented reality: the virtual visual servoing framework. *IEEE Transactions on Visualization and Computer Graphics*, 2006.

[9] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. Eco: Efficient convolution operators for tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[10] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference*, 2014.

[11] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Convolutional features for correlation filter based visual tracking. In *IEEE International Conference on Computer Vision Workshops*, 2015.

[12] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *IEEE International Conference on Computer Vision*, 2015.

[13] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[14] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *European Conference on Computer Vision*, 2016.

[15] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[16] J. Gao, H. Ling, W. Hu, and J. Xing. Transfer learning based visual tracking with gaussian processes regression. In *European Conference on Computer Vision*, 2014.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Neural Information Processing Systems*, 2014.

[18] S. Gurumurthy, R. K. Sarvadevabhatla, and V. B. Radhakrishnan. Deligan: Generative adversarial networks for diverse and limited data. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[19] B. Han, J. Sim, and H. Adam. Branchout: Regularization for online ensemble tracking with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[20] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr. Struck: Structured output tracking with kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[21] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *European Conference on Computer Vision*, 2016.

[22] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *European Conferece on Computer Vision*, 2012.

[23] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[24] S. Hong, T. You, S. Kwak, and B. Han. Online tracking by learning discriminative saliency map with convolutional neural network. In *International Conference on Machine Learning*, 2015.

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

[26] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[27] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[28] H. Kiani Galoogahi, A. Fagg, and S. Lucey. Learning background-aware correlation filters for visual tracking. In *IEEE International Conference on Computer Vision*, 2017.

[29] H. Kiani Galoogahi, T. Sim, and S. Lucey. Correlation filters with limited boundaries. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[30] M. Kristan and et al. The visual object tracking vot2016 challenge results. In *European Conference on Computer Vision Workshops*, 2016.

[31] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[32] X. Lan, A. J. Ma, P. C. Yuen, and R. Chellappa. Joint sparse representation and robust feature-level fusion for multi-cue visual tracking. *IEEE Transactions on Image Processing*, 2015.

[33] X. Lan, S. Zhang, P. C. Yuen, and R. Chellappa. Learning common and feature-specific patterns: a novel multiple-sparse-representation-based tracker. *IEEE Transactions on Image Processing*, 2018.

[34] Y. Li, J. Zhu, and S. C. Hoi. Reliable patch trackers: Robust visual tracking by exploiting reliable patches. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[35] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision*, 2017.

[36] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. Hierarchical convolutional features for visual tracking. In *IEEE International Conference on Computer Vision*, 2015.

[37] C. Ma, X. Yang, C. Zhang, and M.-H. Yang. Long-term correlation tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[38] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[39] J. Ning, J. Yang, S. Jiang, L. Zhang, and M.-H. Yang. Object tracking via dual linear structured svm and explicit feature map. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[40] S. Nowozin, B. Cseke, and R. Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Neural Information Processing Systems*, 2016.

[41] H. Possegger, T. Mauthner, and H. Bischof. In defense of color-based model-free tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[42] Y. Qi, L. Qin, J. Zhang, S. Zhang, Q. Huang, and M.-H. Yang. Structure-aware local sparse coding for visual tracking. *IEEE Transactions on Image Processing*, 2018.

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

[44] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang. Single image dehazing via multi-scale convolutional neural networks. In *European conference on computer vision*, 2016.

[45] S. Rota Bulo, G. Neuhold, and P. Kontschieder. Loss max-pooling for semantic image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[46] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[47] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2014.

[48] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[49] Y. Song, L. Bao, S. He, Q. Yang, and M.-H. Yang. Stylizing face images via multiple exemplars. *Computer Vision and Image Understanding*, 2017.

[50] Y. Song, C. Ma, L. Gong, J. Zhang, R. Lau, and M.-H. Yang. Crest: Convolutional residual learning for visual tracking. In *IEEE International Conference on Computer Vision*, 2017.

[51] Y. Song, J. Zhang, L. Bao, and Q. Yang. Fast preprocessing for robust face sketch synthesis. In *International Joint Conference on Artificial Intelligence*, 2017.

[52] Y. Song, J. Zhang, S. He, L. Bao, and Q. Yang. Learning to hallucinate face images via component generation and enhancement. In *International Joint Conference on Artificial Intelligence*, 2017.

[53] N. Souly, C. Spampinato, and M. Shah. Semi supervised semantic segmentation using generative adversarial network. In *IEEE International Conference on Computer Vision*, 2017.

[54] R. Tao, E. Gavves, and A. W. Smeulders. Siamese instance search for tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[55] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. Torr. End-to-end representation learning for correlation filter based tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[56] A. Vedaldi and K. Lenc. Matconvnet: Convolutional neural networks for matlab. In *ACM International Conference on Multimedia*, 2015.

[57] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *IEEE International Conference on Computer Vision*, 2015.

[58] X. Wang, A. Shrivastava, and A. Gupta. A-fast-rcnn: Hard positive generation via adversary for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[59] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[60] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[61] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[62] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE International Conference on Computer Vision*, 2017.

[63] J. Zhang, S. Ma, and S. Sclaroff. Meem: robust tracking via multiple experts using entropy minimization. In *European Conference on Computer Vision*, 2014.

[64] T. Zhang, C. Xu, and M.-H. Yang. Multi-task correlation particle filter for robust object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.