Robust Correlation Filter Tracking with Shepherded Instance-Aware Proposals

Yanjie Liang, Qiangqiang Wu, Yi Liu, Yan Yan, Hanzi Wang*
Fujian Key Laboratory of Sensing and Computing for Smart City, School of Information Science and Engineering, Xiamen University, Xiamen, China
yanjieliang@yeah.net,qiangwu@stu.xmu.edu.cn,liuyi_xmu@163.com,{yanyan,Hanzi.Wang}@xmu.edu.cn

ABSTRACT

In recent years, convolutional neural network (CNN) based correlation filter trackers have achieved state-of-the-art results on the benchmark datasets. However, the CNN based correlation filters cannot effectively handle large scale variation and distortion (such as fast motion, background clutter, occlusion, etc.), leading to the sub-optimal performance. In this paper, we propose a novel CNN based correlation filter tracker with shepherded instance-aware proposals, namely DeepCFIAP, which automatically estimates the target scale in each frame and re-detects the target when distortion happens. DeepCFIAP is proposed to take advantage of the merits of both instance-aware proposals and CNN based correlation filters. Compared with the CNN based correlation filter trackers, DeepCFIAP can successfully solve the problems of large scale variation and distortion via the shepherded instance-aware proposals, resulting in more robust tracking performance. Specifically, we develop a novel proposal ranking algorithm based on the similarities between proposals and instances. In contrast to the detection proposal based trackers, DeepCFIAP shepherds the instance-aware proposals towards their optimal positions via the CNN based correlation filters, resulting in more accurate tracking results. Extensive experiments on two challenging benchmark datasets demonstrate that the proposed DeepCFIAP performs favorably against state-of-the-art trackers and it is especially feasible for long-term tracking.

KEYWORDS

Visual Tracking; Correlation Filter; Shepherded Instance-Aware Proposals

ACM Reference Format:
Yanjie Liang, Qiangqiang Wu, Yi Liu, Yan Yan, Hanzi Wang. 2018. Robust Correlation Filter Tracking with Shepherded Instance-Aware Proposals. In *MM '18: 2018 ACM Multimedia Conference, Oct. 22–26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240709

*The corresponding author.

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240709

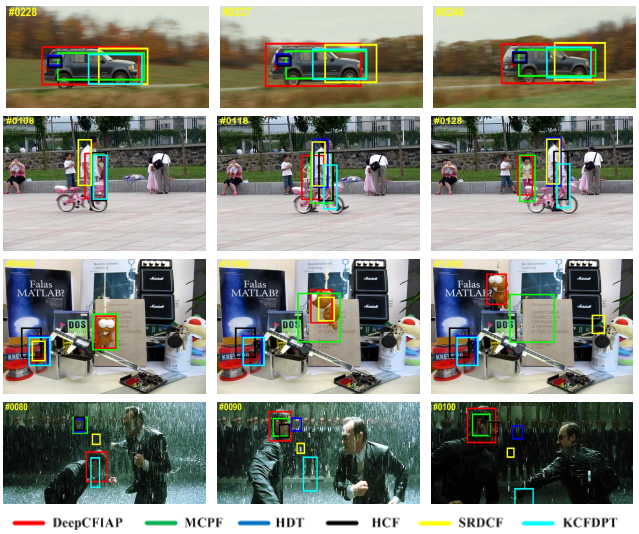

Figure 1: Comparisons of the proposed DeepCFIAP with MCPF, HDT, HCF, SRDCF and KCFDPT in the challenging senarios. Best viewed in color.

1 INTRODUCTION

Visual tracking is a fundamental problem in multimedia and computer vision and it has numerous applications such as video surveillance, human-computer interaction and autonomous driving. In this paper, we address the problem of single object tracking, which aims to track an object in subsequent frames with the target being identified in the first frame. Although much progress has been made in recent years, visual tracking remains a challenging task due to the fluctuant factors, including occlusion, scale variation, background clutter, etc. [17, 21, 34].

Recently, correlation filter based trackers have shown a balanced trade-off between speed and performance [1, 5, 6, 12, 16]. Inspired by the great success of convolutional neural network (CNN) in the visual recognition task, several CNN based correlation filters have been proposed [4, 19, 24]. Experimental results on the benchmark datasets demonstrate that the correlation filter based trackers using deep convolutional features perform favorably against those using hand-crafted features.

Despite achieving promising results, existing CNN based correlation filter trackers still have some drawbacks. These trackers cannot effectively handle large scale variation and distortion (such as fast motion, background clutter, occlusion, etc.). For the scale variation, a DSST tracker [6] is proposed to learn an extra 1D correlation filter for scale estimation. Although it can handle smooth scale variation

to some extent, the tracker will fail when the target undergoes large scale variation. In terms of the distortion, an LCT tracker [20] is developed to leverage an online detector to re-detect the target when it is lost. Despite improving the tracking performance, the tracker is sensitive to the heuristic settings. To address the above issues, we resort to detection proposals [42] to handle these challenges. In a detection proposal based tracker, the proposals can cover the target undergoing large scale variation and distortion. As shown in Figure 1, HCF [19] and HDT [24] cannot effectively adapt to the large scale variation in the *CarScale* sequence. Moreover, both trackers will drift to the occuluder in the *Girl2* sequence. However, DeepCFIAP can accurately track the targets in the challenging sequences.

In detection proposal based trackers, the detection proposals are generated in a region to provide the candidate samples for a classifier [39] or a regressor [16]. The EBT tracker [39] is developed to track the target in the whole image with instance-specific proposals. Although it improves the robustness to various factors, the accuracy is limited. The KCFDPT tracker [16] is proposed to track the target via the detection proposals generated in a small region. Despite improving the tracking performance compared with the baseline tracker [12], it cannot perform reliably in noisy environments. In general, the detection proposal based trackers are more robust when detection proposals are generated in a larger region.

Despite improving the robustness to challenging factors, the detection proposal based trackers have two limitations. On one hand, the computational cost of these trackers is prone to increase with the number of detection proposals. On the other hand, when the detection proposals generated by the object proposal methods cannot cover the target state well, the predicted target state tends to be inaccurate. To address the above issues, we can resort to an effective proposal ranking strategy to reduce the tracking complexity and the CNN based correlation filters to shepherd the detection proposals towards their optimal positions.

In this paper, we propose a novel CNN based correlation filter tracker with instance-aware proposals (DeepCFIAP) for robust visual tracking, which exploits the merits of both instance-aware proposals and CNN based correlation filters. The contributions of this work are given as follows: (1) We propose a DeepCFIAP tracker that can effectively handle large scale variation and distortion via the shepherded instance-aware proposals. Specifically, based on the similarities between proposals and instances, we propose a novel proposal ranking algorithm for instance-aware proposal generation, resulting in more robust tracking performance at a lower computational cost. (2) We shepherd the instance-aware proposals towards their optimal positions via the CNN based correlation filters and chooses the most promising shepherded instance-aware proposal, leading to more accurate tracking results. Extensive experiments on two challenging benchmark datasets demonstrate that the proposed DeepCFIAP performs favorably against state-of-the-art trackers and it is especially feasible for long-term tracking.

2 RELATED WORK

Visual tracking has been studied for several decades. In this section, we review the trackers mostly related to our work: correlation filter (CF) based trackers, CNN based trackers and detection proposal based trackers.

CF Based Trackers. In recent years, CF based trackers [1, 3, 5, 6, 10, 12, 13, 18, 20, 22, 32, 38] have attracted considerable attentions due to their computational efficiency and excellent performance. KCF [12] is the conventional CF tracker, which introduces the circulant structure of shifted samples into the ridge regression. To handle the scale variation, DSST [6] utilizes an extra 1D correlation filter for scale estimation. To alleviate the boundary effect, SRDCF [5] and CSR-DCF [18] introduce a spatial regularization function and a response map respectively to penalize the filter coefficients residing outside the target region. CACF [22] explicitly incorporates the global context in CF learning and deviates a closed-form solution for fast tracking. To strengthen the peak of response map, in [29] and [28], different loss and regularization terms are introduced for CF learning. BACF [10] efficiently models how both the foreground and background vary over time. Staple [1] combines complementary cues (color histogram and HOG features) in CF for robust tracking. LCT [20] and MUSTer [13] are proposed for long-term tracking. In recent tracking community, CF trackers have been integrated with other trackers to complement their merits. The work in [38] fuses the multi-task correlation filter into the particle filter framework to address the scale variation. In [32], CF is combined with SVM to take the merits of strong discriminative ability from SVM and fast running speed from CF. Furthermore, C-COT [7] and ECO [3] extend CFs to continuous convolution operators via an implicit interpolation method, where ECO is the improved version of C-COT. Different from the existing CF based trackers, we resort to instance-aware proposals to handle the large scale variation and distortion.

CNN Based Trackers. Recently, CNNs have successfully employed in visual tracking and achieved state-of-the-art performance. In general, three strategies are usually exploited in the existing CNN based trackers. First, the deep convolutional features extracted from a pre-trained network are transferred for online tracking. HCF [19] and HDT [24] deploy CFs on the features extracted from VGG-Net-19 [26] and obtain the tracking results by combining hierarchical responses and hedging weak trackers, respectively. Second, the tracking problem is formulated as the instance search problem, where the matching function is trained offline via external video data. In [2, 30], Siamese networks are trained offline to address the deep similarity learning problem. CFNet [31] interprets the CFs as differentiable layers in Siamese networks and learns the end-to-end representation via image pairs. In [11], a dynamic Siamese network is trained via video episodes to handle appearance variation and background clutter. In [15], the reinforcement learning is introduced into the Siamese network for adaptive tracking with deep feature cascades. Third, CNNs are fine-tuned during tracking. In MDNet [23], a pre-trained multi-domain CNN integrated with a binary classification layer is fine-tuned online to adapt to the newly tracked target and its appearance variation, achieving state-of-the-art performance. Subsequently, based on the architecture of MDNet, SANet [9] is developed to distinguish the target from its distractors via Recurrent Neural Network (RNN) and ADNet [36] is proposed to adapt to the complex tracking environments via deep reinforcement learning. In [35], the tracking problem is decomposed into a localization task and a classification task and a CNN is trained for each task. In [37], a top-down reasoning model is fine-tuned to cooperate with the appearance based tracker.

Detection Proposal Based Trackers. Recently, several studies have successfully applied the detection proposals in visual tracking. In [14], under the paradigm of tracking-by-detection, the tracking problem is regarded as selecting the detection proposals based on the score and objectness. In [39] and [40], detection proposals generated by EdgeBoxes [42] and Region Proposal Network (RPN) [25] are respectively considered as motion models in the tracking-by-detection framework, resulting in robust tracking performance. An iterative unsupervised method for video detection proposals is proposed in [33] for visual tracking and segmentation, which is competitive with state-of-the-art supervised trackers. In [41], salient proposals are extracted based on the visual saliency map and re-ranked via an effective strategy to estimate the target state. However, it cannot handle the scale variation. The work in [16] integrates the detection proposals into KCF for scale and aspect ratio adaptability but it suffers from the heavy occlusion. Different from these methods, we resort to the CNN based correlation filters to shepherd the instance-aware proposals towards their optimal positions, resulting in more accurate tracking results.

3 PROPOSED ALGORITHM

In the section, we will first present the preliminary blocks (CNN based correlation filters and detection proposal generation) and then illustrate our detection proposal ranking and instance-aware proposal shepherding algorithm. Finally, we show the whole tracking framework with shepherded instance-aware proposals.

3.1 Preliminary Blocks

CNN Based Correlation Filters. The goal of correlation filter formulation is to learn a correlation filter $\mathbf{w}$ from the training samples, which are composed of all cyclic shifts of a base sample. Specifically, the correlation filter learning can be formulated as a ridge regression problem and the objective is to minimize the sum of squared errors from all samples, where the ith error is calculated over the real output of sample $\mathbf{x}_i$ and the desired output y_i :

$$\min_{\mathbf{w}} \sum_{i=1}^{n} (f(\mathbf{x}_i) - y_i)^2 + \lambda ||\mathbf{w}||^2 \tag{1}$$

where λ is a regularization parameter used to avoid overfitting, $\mathbf{x}_i$ is the ith sample corresponding to a cyclic shift of a base sample. y_i is the corresponding label generated by a Gaussian function, where the value equals to 1 for the base sample and gradually decays to 0 for the cyclic shifted samples.

The optimization problem in (1) has a closed-form solution, which can be efficiently solved in the Fourier domain as:

$$\hat{\mathbf{w}} = \frac{\hat{\mathbf{x}}^* \odot \hat{\mathbf{y}}}{\hat{\mathbf{x}} \odot \hat{\mathbf{x}}^* + \lambda} \tag{2}$$

where $\hat{\cdot}$ represents the discrete Fourier transform (DFT), $*$ and $\odot$ denote the complex conjugate and element-wise product, respectively.

Given a test patch $\mathbf{z}$ in a new frame, we first perform DFT and then calculate the response map as:

$$f(\mathbf{z}) = \hat{\mathbf{w}} \odot \hat{\mathbf{z}}^* \tag{3}$$

The estimated target position is the location with the largest response in the response map $f(\mathbf{z})$.

Generally, correlation filter is equipped with the smooth model update scheme when the target undergoes occlusion, deformation, rotation, etc. When the target is detected and the filter is learned in the tth frame, the model is updated with a learning rate η as:

$$\hat{\mathbf{w}}_t = (1-\eta)\hat{\mathbf{w}}_{t-1} + \eta\hat{\mathbf{w}} \tag{4}$$

In the CNN based correlation filters, the outputs of each convolutional layer are used as the multi-channel features to learn an individual correlation filter and generate a corresponding response map. Then these response maps are combined to obtain the final response map, where the location with the largest response corresponds to the estimated target position.

Detection Proposal Generation. We exploit EdgeBoxes [42] to generate the detection proposals for its high recall and fast speed. EdgeBoxes initially utilizes the Structured Edge detector [8] to calculate an edge response for each pixel in the searching region. Then, it traverses the searching region in a sliding window manner and computes the score for each sampled bounding box.

There are several parameters for controlling the sliding window manner. The step size β indicates the intersection over union (IoU) between the neighboring boxes, which controls the sampling density. The minimum box area is constrained by $minArea$. Specifically, in this paper, according to the previous target state, the box aspect ratio ranges from $minAspectRatio$ to $maxAspectRatio$ and the box area ranges from $minBoxArea$ to $maxBoxArea$.

The score for an arbitrary box b is computed as:

$$h_b^{in} = \frac{\sum_{i \in b} w_i m_i - \sum_{p \in b^{in}} m_p}{2(b_w + b_h)^\kappa} \tag{5}$$

where m_i denotes the edge response magnitude of a pixel i and i corresponds to a pixel within b. $w_i \in [0,1]$ indicates how likely the contour (pixel i belongs to) is wholly enclosed by b. b_w and b_h are the width and height of b, respectively. b^{in} denotes the center region of b with the size $[b_w/2, b_h/2]$. κ is used to offset the bias of larger windows having more edges on average.

After scoring all boxes over the position, scale and aspect ratio, these boxes are filtered using the non-maximal suppression (NMS) strategy. γ denotes the IoU threshold when performing NMS, that is, when a box is overlapped with another one and the IoU is higher than the threshold γ, the box with lower score can be removed. NMS will end if the number of passed boxes reaches $maxNumber$.

3.2 Detection Proposal Ranking

After detection proposal generation using EdgeBoxes [42], most proposals are not related to the target. Therefore, a ranking strategy is necessary to convert the detection proposals into the instance-aware proposals. In this paper, we propose a novel proposal ranking strategy based on the appearance similarity and spatial weight.

The similarity measure is calculated between proposals and instances. The instances are collected every T frames and preserved in the set $E = \{e_1, e_2, ..., e_M\}$, where e_i denotes the ith instance and M is the number of instances. The proposals are generated in the current frame and stored in the set $P = \{p_1, p_2, ..., p_N\}$, where p_i represents the ith proposal and N is the number of proposals. We compute two complementary similarity measures: color similarity and shape similarity.

Color Similarity. The color similarity sim_{ij}^{color} between the ith proposal p_i and the jth instance e_j is calculated as the cosine distance between two color histogram vectors of p_i and e_j:

$$sim_{ij}^{color} = \frac{his(p_i)^T \cdot his(e_j)}{||his(p_i)|| \cdot ||his(e_j)||} \tag{6}$$

where $his(\cdot)$ denotes the color histogram vector of the corresponding proposal or instance. The range of color similarity is within $[0, 1]$.

Shape Similarity. The shape similarity sim_{ij}^{shape} between the ith proposal p_i and the jth instance e_j is computed as the cosine distance between two HOG feature vectors of p_i and e_j:

$$sim_{ij}^{shape} = \frac{hog(p_i)^T \cdot hog(e_j)}{||hog(p_i)|| \cdot ||hog(e_j)||} \tag{7}$$

where $hog(\cdot)$ denotes the HOG feature vector of the corresponding proposal or instance. The range of shape similarity is within $[0, 1]$.

Spatial Weight. The spatial weight w_i is defined as the Jaccard similarity coefficient, which is calculated as the IoU between the bounding boxes of the ith proposal p_i and the coarse target state o_c, which is estimated using the CNN based correlation filters:

$$w_i = \frac{|box(p_i) \cap box(o_c)|}{|box(p_i) \cup box(o_c)|} \tag{8}$$

where $box(\cdot)$ represents the bounding box of the corresponding proposal or the coarse target state. The range of spatial weight is within $[0, 1]$.

Since the color and shape similarity are two complementary cues for appearance similarity, we take a combination of the two similarities with θ. The vote v_i for the ith proposal p_i is calculated as follows:

$$v_i = w_i \cdot \max_{e_j \in E}((1-\theta) \cdot sim_{ij}^{color} + \theta \cdot sim_{ij}^{shape}) \tag{9}$$

where $\max(\cdot)$ stands for the maximum of color and shape similarity combination. According to v_i, we rank the detection proposals in the descending order and select the top-ranked ones as the instance-aware proposals.

3.3 Instance-Aware Proposal Shepherding

It is the fact that the instance-aware proposals chosen from the detection proposals usually cannot cover the target state well due to the limitation of EdgeBoxes (the method heavily relies on the edge information of the target rather than the appearance information), leading to the inaccurate tracking results. To address this issue, the instance-aware proposals can be shepherded via the CNN based correlation filters to effectively cover the target state.

As illustrated in Figure 2(a), for an instance-aware proposal i (denoted in red/yellow box), its search region (denoted in red/yellow box with dashed line) is two times the size of this instance-aware proposal and contains the possible translations, which are composed of all circulant shifts of the CNN based correlation filters. Although this instance-aware proposal cannot cover the target state well, a particular circulant shift in the search region can accurately cover the target state. As shown in Figure 2(b), using the CNN based correlation filters, the instance-aware proposal (denoted in red/green/blue box) can be shepherded towards the target state

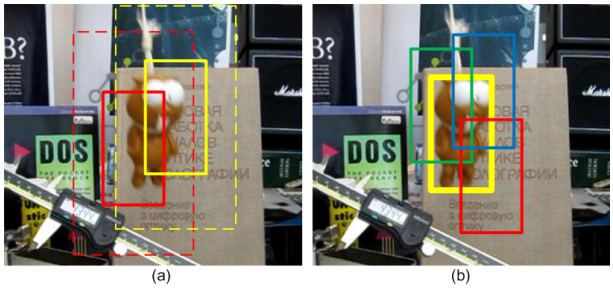

Figure 2: The shepherded instance-aware proposals are able to cover the target state well. (a) The target state can be covered by using search regions (denoted in red/yellow box with dashed line) of instance-aware proposals (denoted in red/yellow box). (b) The instance-aware proposal (denoted in red/green/blue box) can be shepherded towards the target state (denoted in yellow box with bold line).

(denoted in yellow box with bold line). Here, each instance-aware proposal can be considered as a base sample and its circular shifts constitute all dense samples. Therefore, the combination of instance-aware proposals (base samples) in a large search region and their circular shifts (dense samples) in a local search region can increase the possibility to cover the target state, resulting in more robust and accurate tracking performance.

3.4 Tracking with Shepherded Instance-Aware Proposals

After shepherding the instance-aware proposals, how to integrate these proposals into the CNN based correlation filters will be introduced in this section. The pipeline is illustrated in Figure 3. Given the target state (l_1, s_1) in the first frame, where l and s denote the target location and scale respectively, we initialize the model of CNN based correlation filters.

During the process of tracking, we first perform the CNN based correlation filters to estimate the coarse target state. Specifically, when a new frame f_i comes, the CNN based correlation filters are performed on a patch $r_i = (l_{i-1}, c \cdot s_{i-1})$, where the center is the previous target position l_{i-1} and the scale is the previous target scale with padding $c \cdot s_{i-1}$. As the target scale varies during tracking, the patch is resized to the original patch size by bilinear interpolation. Based on the CNN based correlation filters, the coarse target location l_i^c is estimated as the position with the largest response and the coarse target scale s_i^c is retained as the previous target scale s_{i-1}.

To handle the scale variation and distortion well, we develop an effective mode selection strategy based on the quality of response map, which is evaluated by two indicators: the maximum response R_{max} and the Peak-to-Sidelobe Ratio $PSR = (R_{max} - R_\mu)/R_\delta$, where R_μ and R_δ are respectively the mean and standard deviation of response map. R_{max} indicates the likelihood of the target and PSR indicates the discrimination between the target and background, which are two complementary cues to indicate the reliability of the coarse target state. The target re-detection mode is activated if $R_{max} < \tau_{low} \cdot R_{mean}$ or $PSR < \tau_{low} \cdot PSR_{mean}$, where

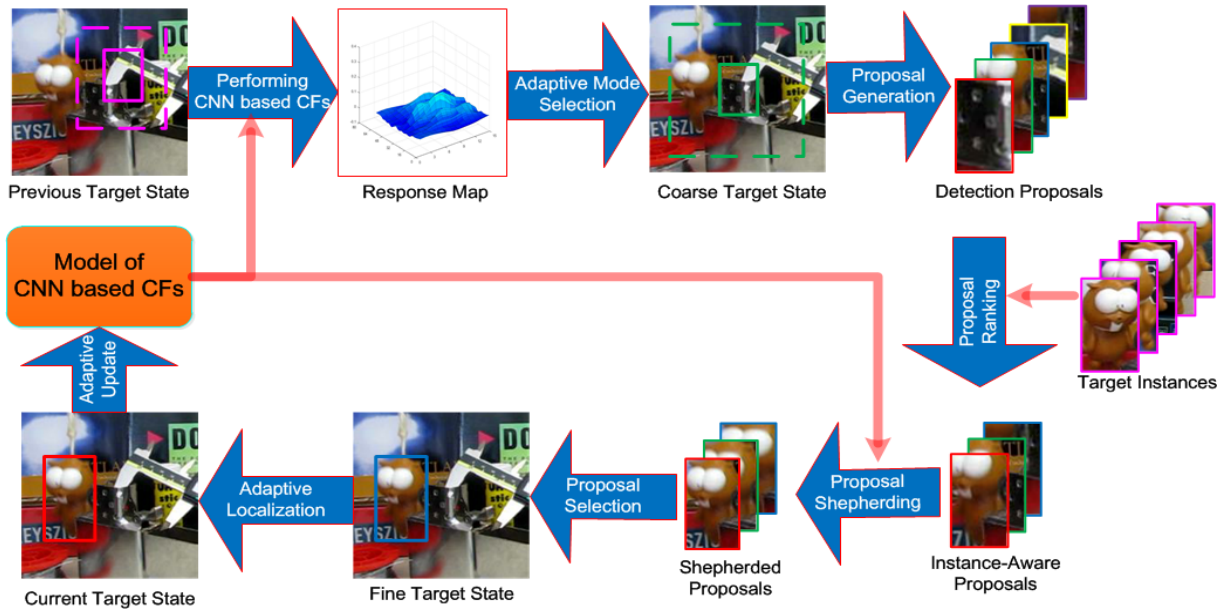

Figure 3: Overview of the proposed DeepCFIAP. The algorithm integrates the shepherded instance-aware proposals into the pipeline of the CNN based CFs. The CNN based CFs focus on the position estimation (coarse target state) while the shepherded instance-aware proposals are responsible for scale estimation/target re-detection (fine target state). Firstly, the CNN based CFs are employed to estimate the coarse target state (the position with the largest response in the response map). Subsequently, in the cropped region (denoted in green box with dashed line), the detection proposals are generated and ranked based on the similarity with target instances to select the instance-aware proposals, which are further shepherded via the CNN based CFs to select the most promising ones as the fine target state. Finally, the current target state is adaptively determined and the model is adaptively updated. The previous/coarse/fine/current target state is denoted in pink/green/blue/red box, respectively.

R_{mean} and PSR_{mean} are the average values of R_{max} and PSR in previous frames; otherwise, the scale estimation mode is activated. Both scale estimation and target re-detection modes share the common tracking framework except for the window factor to control the cropped region at the center of coarse target state. It is evident that the window factor for target re-detection should be larger than that for scale estimation. In practice, the cropped region for target re-detection is gradually enlarged.

Then, in the corresponding cropped region, we generate the detection proposals using EdgeBoxes as described in Section 3.1 and rank these detection proposals by our novel ranking algorithm as described in Section 3.2. The top-ranked detection proposals are chosen as the instance-aware proposals. Subsequently, we shepherd these instance-aware proposals towards their optimal positions (locations with the largest response) via the CNN based correlation filters as described in Section 3.3. The shepherded instance-aware proposal with the largest response is chosen as the most promising proposal to estimate the fine target state (l_i^f, s_i^f). Furthermore, the corresponding maximum response R_{max}^p and Peak-to-Sidelobe Ratio PSR^p are recorded.

After estimating the coarse target state (l_i^c, s_i^c) and fine target state (l_i^f, s_i^f), we adopt the adaptive target localization strategy to estimate the new target state as follows. In the target re-detection mode, the target is successfully re-detected as (l_i^f, s_i^f) when $R_{max}^p \geq \tau_{high} \cdot R_{mean}$ and $PSR^p \geq \tau_{high} \cdot PSR_{mean}$; otherwise, the new target state is estimated as (l_i^c, s_i^c). In the scale estimation mode, the new target state is estimated as the target state with larger maximum response. When the new target state (l_i, s_i) is estimated in frame f_i, we exploit an adaptive model update strategy to update the model of CNN based correlation filters as follows: the model is maintained in the target re-detection mode; the model is updated using the learning rate η when $PSR \geq PSR_{mean}$ and $R_{max} \geq R_{mean}$; otherwise, the model is updated using the adjusted learning rate $c_r \cdot \eta$, where c_r is the relative ratio to reduce the learning rate.

4 EXPERIMENTS

In this section, we first introduce the experimental settings and in depth studies of the proposed DeepCFIAP. Then, we evaluate our DeepCFIAP with state-of-the-art trackers on large-scale benchmark datasets: OTB100 [34] and UAV20L [21].

4.1 Experimental Settings

We use HCF [19] as our baseline tracker and follow the same implementations with the minor revision, where the conv3-4, conv4-4 and conv5-4 convolutional layers of the VGG-Net-19 [26] are used to encode target appearance and the weights of these layers are respectively set to 0.25, 0.50 and 1. We use a kernel width of 0.1 to generate the Gaussian labels and a cosine window to weigh the

feature maps. The learning rate η in (4) is set to 0.01 and the padding is adjusted to 1.56. As KCFDPT [16], we use EdgeBoxes [42] for detection proposal generation. The step size β and NMS threshold γ are respectively set to 0.75 and 0.85. The minimum area of box $minArea$ and the maximum number of boxes $maxNumber$ are respectively set to 200 and 1000. In the scale estimation mode, $maxAspectRatio = 1.3$, $minAspectRatio = 0.7$, $maxBoxArea = 1.3$, $minBoxArea = 0.7$, the window factor is fixed as 1.4. In the target re-detection mode, $maxAspectRatio = 1.5$, $minAspectRatio = 0.5$, $maxBoxArea = 1.5$, $minBoxArea = 0.5$, the window factor is gradually enlarged from 3 to 5 with the step size 0.225. When ranking the detection proposals, we collect the target instances every 50 frames and set the maximum number to 10 in the set. The number of the bins in the color histogram is set to 32. The cell size and the number of orientations in the HOG feature are set to 4 and 9 respectively. The combination weight θ in (9) is set to 0.7. The thresholds for adaptive mode selection, target localization and model update are set as: $\tau_{low} = 0.72$ and $\tau_{high} = 0.9$. The relative ratio c_r is set to 0.7. The previous frame number for calculating R_{mean} and PSR_{mean} is set to 20. With above settings, we implement our tracker in Matlab using MatConvNet toolbox. Our tracker runs around 1 FPS on Intel I7-6700K 4.00GHz CPU and a NVDIA GTX 1080 GPU when the top-ranked 1/2 proportion of detection proposals are selected as the instance-aware proposals.

Our tracker is evaluated on two standard benchmark datasets, OTB100 [34] and UAV20L [21], which contain video sequences with various attributes, such as scale variation, occlusion, background clutter, fast motion, etc. In each frame of the video sequences, the target is annotated by a bounding box, which can be used as the ground truth for quantitative evaluation. In particular, UAV20L [21] is the dataset with 20 long video sequences, where the longest sequence contains 5527 frames and the shortest one contains 1717 frames. We use two metrics for one pass evaluation (OPE): precision and success plots, which are defined as: (1) precision: the percentage of frames of which the center location errors are less than a predefined threshold with the ground truth; (2) success: the percentage of frames of which the overlap ratios are larger than a predefined threshold with the ground truth. We report the distance precision at 20 pixels threshold (DP) in precision plot and the area under curve (AUC) in success plot for one pass evaluation (OPE).

4.2 In Depth Studies

In this section, we make experiments on ablation studies and parameter investigations to evaluate the proposed DeepCFIAP.

Ablation Studies. We analyze the proposed DeepCFIAP on the OTB100 dataset to demonstrate the effectiveness of different components, including the tracker 1) without using the adaptive mode selection strategy for target re-detection (DeepCFIAP-ND), 2) without using the CNN based correlation filters to shepherd the instance-aware proposals (DeepCFIAP-NS).

As shown in Table 1, the proposed tracker integrated all components obtains the highest scores on both the DP and AUC metrics among all the competing trackers. As DeepCFIAP-ND only resorts to the instance-aware proposals for scale estimation, the performance loss can be solely ascribed to the effective mode selection strategy. DeepCFIAP-NS performs much worse than the proposed DeepCFIAP since the instance-aware proposals usually cannot enclose the target well, which demonstrates the necessity of shepherding the instance-aware proposals via the CNN based correlation filters.

Table 1: Different components analysis on the tracking performance on the OTB100 dataset.

Tracker	DeepCFIAP-NS	DeepCFIAP-ND	DeepCFIAP
DP (%)	80.3	85.2	**89.3**
AUC (%)	57.4	63.1	**65.5**

Table 2: Influence of different proportions of detection proposals chosen as the instance-aware proposals on the tracking performance on the OTB100 dataset.

Proportion	1/2	1/3	1/4	1/5	1/6
DP (%)	**89.3**	85.9	86.2	85.7	85.0
AUC (%)	**65.5**	63.7	64.0	63.1	62.9

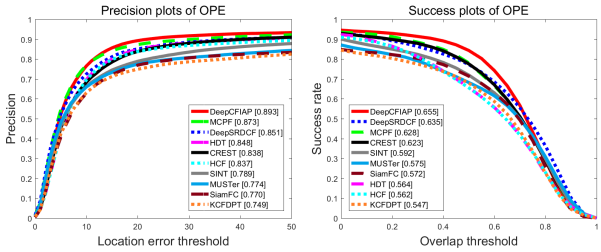

Figure 4: Performance of the proposed DeepCFIAP and nine other state-of-the-art trackers on the OTB100 dataset.

Parameter Investigations. We analyze the influence of the proportion of detection proposals chosen as the instance-aware proposals on the OTB100 dataset. As illustrated in Table 2, we set the proportion as 1/2, 1/3, 1/4, 1/5 and 1/6 respectively. According to the results, when we select the top-ranked 1/2 proportion of detection proposals as the instance-aware proposals, the proposed DeepCFIAP achieves the best tracking performance. The results of 1/3 proportion are slightly inferior to those of 1/4 proportion, which may be caused by the background distrators. The phenomenon demonstrates that the top ranked 1/2 detection proposals can perform robust tracking.

4.3 Evaluation on OTB100

We compare the proposed DeepCFIAP with several state-of-the-art trackers: DeepSRDCF [4], HCF [19], HDT [24], SINT [30], SiamFC [2], CREST [27], MCPF [38], KCFDPT [16] and MUSter [13]. Many of aforementioned trackers are CNN based correlation filters, such as DeepSRDCF [4], HCF [19], HDT [24], CREST [27] and MCPF [38].

Figure 4 reports the evaluation results of the proposed DeepCFIAP and its competitors on the OTB100 dataset. Our proposed

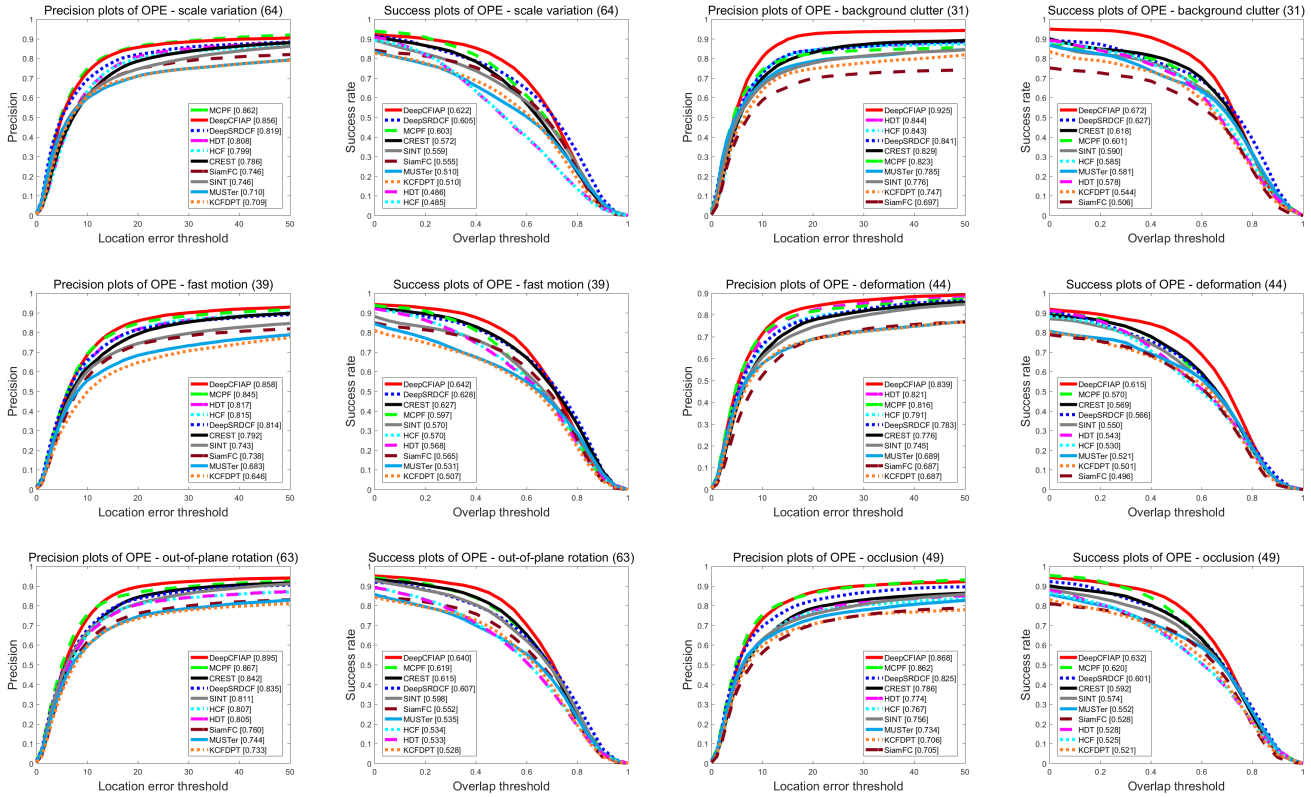

Figure 5: Precision and success plots for six challenging attributes: scale variation, background clutter, fast motion, deformation, out-of-plane rotation and occlusion on the OTB100 dataset.

tracker performs favorably against these compared trackers. Specifically, it achieves 89.3%/65.5% (DP/AUC) in the precision and success plots, outperforming the second best results by 2.3% and 3.1% on the DP and AUC metrics, respectively. In particular, the proposed DeepCFIAP outperforms its baseline tracker HCF by 6.6% and 16.5% on the DP and AUC metrics, respectively, which can be attributed to the shepherded instance-aware proposals incorporated into the proposed DeepCFIAP for target scale estimation and re-detection. Furthermore, the proposed DeepCFIAP outperforms MCPF by 2.3% and 4.3% on the DP and AUC metrics, respectively. Note MCPF is similar to the proposed DeepCFIAP that the sampled particles are shepherded by multi-task correlation filter. In Figure 4, we do not illustrate the results of MDNet [23] and ECO [3], where the former uses external tracking videos for training and the latter uses efficient convolution operators to substitute the correlation filters. MDNet achieves 90.9%/67.8% (DP/AUC) and ECO achieves 91.0%/69.1% (DP/AUC), which are superior to the proposed DeepCFIAP on this dataset. Overall, the evaluation results on the OTB100 dataset demonstrate that our proposed DeepCFIAP performs well against state-of-the-art trackers on the 100 challenging videos.

The 100 sequences on the OTB100 dataset are annotated with 11 attributes, including illumination/scale variation, occlusion, deformation, fast motion, background clutter, etc. In Figure 5, we evaluate the tracking performance under the attributes of scale variation, background clutter, fast motion, deformation, out-of-plane rotation and occlusion. As shown in Figure 5, the proposed DeepCFIAP can effectively handle these challenging situations. Specifically, the proposed DeepCFIAP outperforms the second best results on background clutter on the DP and AUC metrics by a large margin (9.6% and 7.2%). Although MCPF can handle scale variation well via the particle sampling scheme, the proposed DeepCFIAP achieves competitive performance via the shepherded instance-aware proposals in terms of the scale variation attribute. For other attributes, the proposed DeepCFIAP performs much better than the HCT, HDT and CREST trackers on both the DP and AUC metrics, which demonstrates that the proposed DeepCFIAP is more robust to these attributes.

4.4 Evaluation on UAV20L

We then evaluate the proposed DeepCFIAP on the UAV20L dataset and compare the proposed DeepCFIAP with state-of-the-art trackers: ECO [3], MDNet [23], MCPF [38], HCF [19], HDT [24], SiamFC [2], SRDCF [5], KCFDPT [16] and MUSTer [13]. Note that ECO and MDNet achieve state-of-the-art performance on the OTB100 dataset.

Figure 6 shows the evaluation results of the proposed DeepCFIAP and all compared trackers. In general, the proposed DeepCFIAP outperforms all other competitors by a large margin, achieving 67.4%/45.9% (DP/AUC) scores in the precision and success plots.

Table 3: Attribute based comparison with state-of-the-art trackers on the UAV20L dataset. We report DP/AUC scores (%) for these trackers. The attributes are aspect ratio change (ARC), scale variation (SV), illumination variation (IV), viewpoint change (VC), camera motion (CM), fast motion (FM), similar object (SOB), background clutter (BC), full occlusion (FOC), partial occlusion (POC), out-of-view (OV) and low resolution (LR). The best values are highlighted by bold.

Tracker	DeepCFIAP	ECO	MDNet	MCPF	SiamFC	HCF	HDT	SRDCF	KCFDPT	MUSTer
ARC	**61.1/43.6**	49.9/34.8	47.2/34.4	49.9/33.5	51.8/33.5	37.8/27.4	36.1/22.6	38.9/27.0	44.0/30.7	44.3/27.7
SV	**66.7/46.5**	57.6/40.7	54.7/40.4	56.6/36.7	59.2/38.9	46.3/34.0	42.6/25.5	48.1/33.2	49.4/34.2	49.2/31.6
IV	**60.3/46.0**	53.1/39.9	49.6/36.7	54.4/36.0	51.8/38.9	43.1/32.5	43.0/31.3	41.1/29.5	43.9/30.9	37.8/24.2
VC	**61.2/45.5**	51.9/37.9	54.1/42.1	46.3/30.4	54.5/36.4	39.4/30.6	35.8/23.5	41.4/30.3	44.9/33.9	47.3/32.2
CM	**65.7/45.5**	57.6/40.2	54.7/40.3	57.2/36.7	59.2/38.9	46.3/33.1	42.6/26.0	48.2/32.7	49.3/33.9	49.2/30.9
FM	**65.3/41.2**	49.7/27.7	50.6/29.8	47.9/25.7	52.3/25.4	35.9/20.8	35.7/16.9	32.7/19.7	39.4/27.9	42.3/21.2
SOB	**71.8/53.3**	55.1/44.0	58.4/47.9	60.2/43.3	60.3/45.0	44.4/36.8	40.6/26.8	52.2/39.7	47.1/34.8	48.3/34.2
BC	38.3/23.5	**44.0/25.3**	26.6/13.4	38.0/24.1	36.4/23.7	33.9/21.5	33.8/21.0	25.2/15.6	33.3/19.2	42.0/23.0
FOC	**48.1/26.5**	43.4/22.5	36.4/17.8	43.3/23.8	42.7/23.6	37.7/20.6	35.3/17.1	33.1/17.0	38.5/19.0	42.3/20.5
POC	**64.5/44.4**	56.1/38.9	53.7/39.4	57.3/37.2	57.1/37.0	45.2/32.3	40.8/24.5	49.1/32.0	50.3/33.3	49.6/30.8
OV	**62.9/43.9**	55.9/38.5	57.8/43.8	53.0/33.8	61.8/39.2	42.9/32.2	35.2/21.2	49.5/32.9	49.9/33.0	50.2/31.3
LR	**61.7/37.0**	51.8/28.5	52.6/31.5	48.3/27.9	46.7/23.8	41.8/22.6	39.3/17.3	42.9/22.8	45.1/26.1	51.3/27.8
Overall	**67.4/45.9**	59.7/42.0	57.0/41.9	58.8/36.4	61.2/40.3	49.0/35.2	45.5/27.3	50.7/34.3	51.9/35.7	51.7/33.1

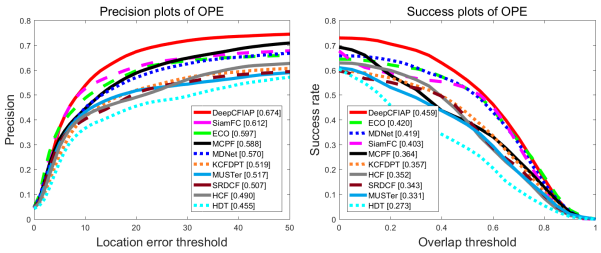

Figure 6: Performance of the proposed DeepCFIAP and nine other state-of-the-art trackers on the UAV20L dataset.

Specifically, despite the inferior performance of the proposed DeepCFIAP compared with ECO and MDNet on the OTB100 dataset, it significantly outperforms them on the DP and AUC metrics (12.9% and 9.3% for ECO, 18.2% and 9.5% for MDNet) on this dataset. Furthermore, the proposed DeepCFIAP surpasses its baseline tracker HCF by 37.6% and 30.4% on the DP and AUC metrics respectively, which can be attributed to the shepherded instance-aware proposals incorporated into the proposed DeepCFIAP for the target scale estimation and re-detection. Surprisingly, SiamFC performs well on this dataset and it achieves better results than the sophisticated trackers ECO and MDNet on the DP metric, which can be ascribed to the large region for instance search. However, it is still inferior to the proposed DeepCFIAP equipped with an adaptive mode selection strategy. Overall, the evaluation results on the UAV20L dataset demonstrate that the proposed DeepCFIAP outperforms state-of-the-art trackers by a large margin on the 20 long challenging videos.

We further evaluate the robustness of the proposed DeepCFIAP on various attributes. The UAV20L benchmark classifies the 20 long videos into 12 challenging scenarios: background clutter, similar object, partial/full occlusion, viewpoint change, scale/illumination variation, aspect ratio change, fast/camera motion, low resolution and out-of-view. Table 3 illustrates the tracking performance of the proposed DeepCFIAP and its counterparts for attribute analysis. We can see that the proposed DeepCFIAP outperforms its competitors on all attributes except for the background clutter, which is slightly inferior to ECO. For the aspect ratio change and scale variation attributes, the proposed DeepCFIAP outperforms the second best results by a large margin, which demonstrates that the proposed DeepCFIAP can handle aspect ratio and scale variation well via the shepherded instance-aware proposals. In terms of the full/partial occlusion attributes, the proposed DeepCFIAP outperforms the second best results by 10.80%/11.52% and 10.17%/11.68% on the DP and AUC metrics respectively, which can be ascribed to the adaptive mode selection strategy for target re-detection to handle the full/partial occlusion.

5 CONCLUSIONS

In this paper, we propose a robust CNN based correlation filter tracking method with shepherded instance-aware proposals. The proposed tracker can effectively handle the scale variation and distortion via the shepherded instance-aware proposals. Specifically, we propose a novel effective proposal ranking algorithm based on the similarities between proposals and instances to select the instance-aware proposals. Furthermore, we shepherd the instance-aware proposals towards their optimal positions via the CNN based correlation filters to accurately cover the target. Extensive experiments on the standard benchmark datasets OTB100 and UAV20L demonstrate that the proposed tracker performs favorably against state-of-the-art trackers. It is worth emphasizing that the proposed tracker is especially sufficient for long-term tracking.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant No. U1605252, 61472334 and 61571379) and the National Key Research and Development Program of China (Grant No. 2017YFB1302400).

REFERENCES

[1] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr. 2016. Staple: Complementary Learners for Real-Time Tracking. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 1401–1409.

[2] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr. 2016. Fully-Convolutional Siamese Networks for Object Tracking. In *Proc. of European Conference on Computer Vision Workshops*. 850–865.

[3] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. 2017. ECO: Efficient Convolution Operators for Tracking. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 6931–6939.

[4] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. 2015. Convolutional Features for Correlation Filter Based Visual Tracking. In *Proc. of IEEE International Conference on Computer Vision Workshops*. 621–629.

[5] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. 2015. Learning Spatially Regularized Correlation Filters for Visual Tracking. In *Proc. of IEEE International Conference on Computer Vision*. 4310–4318.

[6] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. 2017. Discriminative Scale Space Tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 39, 8 (August 2017), 1561–1575. https://doi.org/10.1109/TPAMI.2016.2609928

[7] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg. 2016. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In *Proc. of European Conference on Computer Vision*. 472–488.

[8] P. Dollár and C. L. Zitnick. 2013. Structured Forests for Fast Edge Detection. In *Proc. of IEEE International Conference on Computer Vision*. 1841–1848.

[9] H. Fan and H. Ling. 2017. SANet: Structure-Aware Network for Visual Tracking. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2217–2224.

[10] H. K. Galoogahi, A. Fagg, and S. Lucey. 2017. Learning Background-Aware Correlation Filters for Visual Tracking. In *Proc. of IEEE International Conference on Computer Vision*. 1144–1152.

[11] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang. 2017. Learning Dynamic Siamese Network for Visual Object Tracking. In *Proc. of IEEE International Conference on Computer Vision*. 1781–1789.

[12] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. 2015. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 37, 3 (March 2015), 583–596. https://doi.org/10.1109/TPAMI.2014.2345390

[13] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao. 2015. MUlti-Store Tracker (MUSTer): A Cognitive Psychology Inspired Approach to Object Tracking. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 749–758.

[14] Y. Hua, K. Alahari, and C. Schmid. 2015. Online Object Tracking with Proposal Selection. In *Proc. of IEEE International Conference on Computer Vision*. 3092–3100.

[15] C. Huang, S. Lucey, and D. Ramanan. 2017. Learning Policies for Adaptive Tracking with Deep Feature Cascades. In *Proc. of IEEE International Conference on Computer Vision*. 105–114.

[16] D. Huang, L. Luo, Z. Chen, M. Wen, and C. Zhang. 2017. Applying Detection Proposals to Visual Tracking for Scale and Aspect Ratio Adaptability. *International Journal of Computer Vision* 122, 3 (May 2017), 524–541. https://doi.org/10.1007/s11263-016-0974-6

[17] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Čehovin, G. Fernandez, T. Vojíř, G. Häger, G. Nebehay, R. Pflugfelder, A. Gupta, A. Bibi, A. Lukežič, A. Garcia-Martin, A. Saffari, and A. Petrosino. 2015. The Visual Object Tracking VOT2015 Challenge Results. In *Proc. of IEEE International Conference on Computer Vision Workshops*. 564–586.

[18] A. Lukežič, T. Vojíř, L. Čehovin, J. Matas, and M. Kristan. 2017. Discriminative Correlation Filter with Channel and Spatial Reliability. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 4847–4856.

[19] C. Ma, J. B. Huang, X. Yang, and M. H. Yang. 2015. Hierarchical Convolutional Features for Visual Tracking. In *Proc. of IEEE International Conference on Computer Vision*. 3074–3082.

[20] C. Ma, X. Yang, C. Zhang, and M. H. Yang. 2015. Long-Term Correlation Tracking. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 5388–5396.

[21] M. Mueller, N. Smith, and B. Ghanem. 2016. A Benchmark and Simulator for UAV Tracking. In *Proc. of European Conference on Computer Vision*. 445–461.

[22] M. Mueller, N. Smith, and B. Ghanem. 2017. Context-Aware Correlation Filter Tracking. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 1387–1395.

[23] H. Nam and B. Han. 2016. Learning Multi-domain Convolutional Neural Networks for Visual Tracking. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 4293–4302.

[24] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M. H. Yang. 2016. Hedged Deep Tracking. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 4303–4311.

[25] S. Ren, K. He, R. Girshick, and J. Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Proc. of International Conference on Neural Information Processing Systems*. 91–99.

[26] K. Simonyan and A. Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proc. of International Conference on Learning Representations*.

[27] Y. Song, C. Ma, L. Gong, J. Zhang, R. W. H. Lau, and M. H. Yang. 2017. CREST: Convolutional Residual Learning for Visual Tracking. In *Proc. of IEEE International Conference on Computer Vision*. 2574–2583.

[28] Y. Sui, G. Wang, and L. Zhang. 2018. Correlation Filter Learning Toward Peak Strength for Visual Tracking. *IEEE Trans. on Cybernetics* 48, 4 (April 2018), 1290–1303. https://doi.org/10.1109/TCYB.2017.2690860

[29] Y. Sui, Z. Zhang, G. Wang, Y. Tang, and L. Zhang. 2016. Real-Time Visual Tracking: Promoting the Robustness of Correlation Filter Learning. In *Proc. of European Conference on Computer Vision*. 662–678.

[30] R. Tao, E. Gavves, and A. W. M. Smeulders. 2016. Siamese Instance Search for Tracking. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 1420–1429.

[31] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr. 2017. End-to-End Representation Learning for Correlation Filter Based Tracking. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 5000–5008.

[32] M. Wang, Y. Liu, and Z. Huang. 2017. Large Margin Object Tracking with Circulant Feature Maps. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 4800–4808.

[33] F. Xiao and Y. J. Lee. 2016. Track and Segment: An Iterative Unsupervised Approach for Video Object Proposals. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 933–942.

[34] J. Lim Y. Wu and M. H. Yang. 2015. Object Tracking Benchmark. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 37, 9 (September 2015), 1834–1848. https://doi.org/10.1109/TPAMI.2014.2388226

[35] L. Yang, R. Liu, D. Zhang, and L. Zhang. 2017. Deep Location-Specific Tracking. In *Proc. of ACM on Multimedia Conference*. 1309–1317.

[36] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi. 2017. Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 1349–1358.

[37] M. Zhang, J. Feng, and W. Hu. 2017. Robust Visual Object Tracking with Top-down Reasoning. In *Proc. of ACM on Multimedia Conference*. 226–234.

[38] T. Zhang, C. Xu, and M. H. Yang. 2017. Multi-task Correlation Particle Filter for Robust Object Tracking. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 4819–4827.

[39] G. Zhu, F. Porikli, and H. Li. 2016. Beyond Local Search: Tracking Objects Everywhere with Instance-Specific Proposals. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 943–951.

[40] G. Zhu, F. Porikli, and H. Li. 2016. Robust Visual Tracking with Deep Convolutional Neural Network Based Object Proposals on PETS. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 1265–1272.

[41] G. Zhu, J. Wang, Y. Wu, X. Zhang, and H. Lu. 2016. MC-HOG Correlation Tracking with Saliency Proposal. In *Proc. of AAAI Conference on Artificial Intelligence*. 3690–3696.

[42] L. Zitnick and P. Dollár. 2014. Edge Boxes: Locating Object Proposals from Edges. In *Proc. of European Conference on Computer Vision*. 391–405.