LONG AND SHORT MEMORY BALANCING IN VISUAL CO-TRACKING USING Q-LEARNING

Kourosh Meshgi[⋆†] *Maryam Sadat Mirzaei*[⋆†] *Shigeyuki Oba*[⋆]

[⋆] Graduate School of Informatics, Kyoto University, Japan
[†] RIKEN Center for Advanced Intelligence Project (AIP), Japan

ABSTRACT

Employing one or more additional classifiers to break the self-learning loop in tracing-by-detection has gained considerable attention. Most of such trackers merely utilize the redundancy to address the accumulating label error in the tracking loop, and suffer from high computational complexity as well as tracking challenges that may interrupt all classifiers (e.g. temporal occlusions). We propose the active co-tracking framework, in which the main classifier of the tracker labels samples of the video sequence, and only consults auxiliary classifier when it is uncertain. Based on the source of the uncertainty and the differences of two classifiers (e.g. accuracy, speed, update frequency, etc.), different policies should be taken to exchange information between two classifiers. Here, we introduce a reinforcement learning approach to find the appropriate policy by considering the state of tracker in a specific sequence. The proposed method yields promising results in comparison to the best tracking-by-detection approaches.

Index Terms— visual co-tracking, active learning, Q-learning, long-short memory

1. INTRODUCTION

Tracking-by-detection methods are built around the idea that a single classifier separates the target from its background by labeling (or filtering) several samples from the input image, labeling them, and extrapolating these samples to estimate the current target location and size. This classifier needs to be updated to cope with recent target transformations as well as other challenging factors such as changes in illumination, camera pose, cluttered background, and occlusions. The update process is mainly done using the labels that the classifier selected for the samples, in a self-supervised learning fashion.

A classifier is not always certain about the output labels. Whether it is inefficient features for certain input images, insufficient model complexity to separate some of the samples, lack of proper training data, missing information in the input data (e.g., due to occlusion), or technically speaking, having an input sample that falls very close to decision boundary of the classifier, hampers the classifier ability to be sure about its label and increase the risk of misclassification. Especially in the case of online learning, novel appearances of the target, background distractors, and non-stationarity of the label distribution[1] promotes the uncertainty of the classifier.

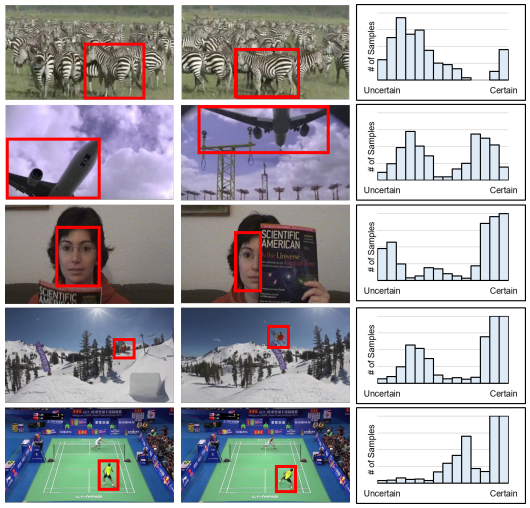

Fig. 1. Consider a classifier of tracking-by-detection that uses color and shape features and is trained on video frames leading to the frame on the left column. When classifying n_s samples from the frame in the middle column, the uncertainty for all samples may have different trends, as plotted in the uncertainty histogram in right panels. The histogram may be skewed toward certainty, uncertainty (e.g. due to feature failures or occlusion), bimodal (where usually background is easy to separate but the foreground is ambiguous), etc. In co-tracking frameworks, various patterns of uncertainty require different policies to enhance tracking performance.

Furthermore, the self-supervised learning loop may lead to model drift due to the accumulation of label errors, and many studies have tried to tackle this problem by using robust loss functions for the classifier [1], merging the sampling and learning [2], and employing unlabeled data [3]. One of

This article is based on results obtained from a project commissioned by the NEDO and was supported by Post-K application development for exploratory challenges from Japan's MEXT.

[1] A sample might be considered as a foreground but later the label become obsolete or become a part of the background.

the most prominent approaches to tackle this problem is to augment the classifier with one or more classifier to break the self-learning loop [4] and provide a teacher for the main classifier [3]. Such ideas are manifested in the form of co-tracking [4, 5] and ensemble tracking [6, 7]. The bring complementary benefits to the tracker, extra classifier(s) should differ from the main one in training data [6], learning model, update mechanism [8], update frequency or memory span [5]. Controlling classifier memory is one of the approaches to promote co-tracking and increase accuracy [2, 5, 8–11].

Here, we take advantage of the information about classifier's uncertainty state in a scenario (Figure 1), to control the information exchange between the main classifier of the tracker, and a more accurate yet slow auxiliary classifier in the co-tracking framework. To cope with rapid target changes and handling challenges such as temporal target deformations and occlusion, we set different memory span and update frequency for classifiers. Naturally, the main classifier is selected to an agile, plastic, easily-updatable and frequently-updating model whereas the auxiliary tracker is more sophisticated (accurate yet slow), more stable (memorizing all labels in the tracker's history), and less-frequently updated. The main classifier only queries a label from the auxiliary one, when it is uncertain about a sample's label in line with the uncertainty sampling [12]. We proposed a Q-learning approach to govern the information exchange between two classifiers w.r.t. the uncertainty state of the first classifier. This scheme automatically balances the stability-plasticity trade-off in tracking [3] and long-short memory trade-off while increasing the speed of the tracker (by avoiding unnecessary queries from the slow classifier) and enhances the generalization ability of the first classifier (by advantaging from the benefits of active learning). The proposed tracker performs better than many of the state-of-the-art in tracking-by-detection.

2. PROPOSED METHOD

In this section, we formalize a tracking-by-detection pipeline, expand it with the notion of traditional and active co-tracking, and elaborate the proposed Q-learning approach that is intended to balance the use of short and long-term memories.

2.1. Active Co-Tracking

A tracking-by-detection pipeline consists of a sampler, that selects n_s samples $\mathbf{x}_t = \{x_t^j\}$ from the given frame I_t, give it to the classifier modeled by θ_t to obtain the labels $\mathbf{l}_t = \{\ell_t^j\}$ of the samples. A label is the result of thresholding the scoring function $h : x \to [0, 1]$ for sample x_t^j,

$$\ell_t^j = \text{sign}\Big(h(x_t^j|\theta_t) - \tau\Big) \tag{1}$$

and threshold τ is typically set to ½. Finally, the model is updated using the obtained labels $\theta_{t+1} = u(\theta_t, \mathbf{x}_t, \mathbf{l}_t)$.

In an attempt to break the self-learning loop, co-tracking [4] uses two parallel classifiers θ_t^1 and θ_t^2 with potentially complementary characteristics, and label a sample based on their weighted vote. Both classifiers are updated each frame, and their voting weights α_t^i are re-adjusted based on their label consistency on the co-labeled samples.

$$\ell_t^j = \begin{cases} \text{sign}\Big(h(x_t^j|\theta_t^1) - \tau\Big) & , h(x_t^j|\theta_t^2) < \tau \\ \text{sign}\Big(h(x_t^j|\theta_t^2) - \tau\Big) & , h(x_t^j|\theta_t^1) < \tau \\ \text{sign}\Big(\alpha_t^1 h(x_t^j|\theta_t^1) + \alpha_t^2 h(x_t^j|\theta_t^2) - \tau\Big) & , \text{otherwise} \end{cases} \tag{2}$$

Co-tracking is built on the premise that when one classifier has difficulty labeling a sample, the other one assists. Co-tracking increases the tracking accuracy (by decreasing label noise) and addresses model drift. On the other hand, *(i)* using two classifiers doubles the computational complexity, and *(ii)* in the challenging cases such as temporal full occlusions and background clutter, both of classifiers may encounter difficulty in labeling and will be updated with noisy labels.

To speed up the tracking, Meshgi et al. [13] proposed that a second auxiliary classifier should be consulted, only when needed by the main classifier. This approach enables using more sophisticated models for the auxiliary classifier and decreases computational complexity. To select the samples, that knowing their label maximally help the main classifier to continue accurate tracking, active learning approaches can be used. When a classifier is more uncertain about the label of a sample, knowing its label would increase the learning of the classifier more [12], thus improves its convergence speed and generalization with a limited number of labeled samples. In tracking such uncertainty may come from the ineffectiveness of the features, the sample being close to the decision boundary [14], or missing information (e.g. due to partial occlusion). Therefore, an active co-tracker only queries the labels of the samples from aux classifier when the main classifier is most uncertain about their labels.

$$\ell_t^j = \begin{cases} \text{sign}\Big(h(x_t^j|\theta_t^1) - \tau\Big) & , |h(x_t^j|\theta_t^1) - \tau| > \delta \\ \text{sign}\Big(h(x_t^j|\theta_t^2) - \tau\Big) & , \text{otherwise} \end{cases} \tag{3}$$

Here, $|h(x_t^j|\theta_t^1) - \tau| > \delta$ means that the uncertainty is less than margin δ. After the labeling, the main classifier is trained with the co-labeled data and learns maximally from the auxiliary classifier. To handle tracking challenges, such as temporal occlusions and deformations, the aux classifier is updated every Δ frames. This in-turn further reduces the computational complexity of the tracker and benefits the tracker from a combination of long and short term memories.

$$\theta_{t+1}^{(2)} = \begin{cases} u'(\theta_t^{(2)}, \mathbf{x}_{t-\Delta..t}, \mathbf{l}_{t-\Delta..t}) & , t = k\Delta \\ \theta_t^{(2)} & , t \neq k\Delta \end{cases} \tag{4}$$

The uncertainty margin δ controls the "activeness" of the tracker. Smaller δ typically reduces the samples that aux classifier labels, focus on recent samples by leaning toward short term memory and reduce the complexity of the tracker. How-

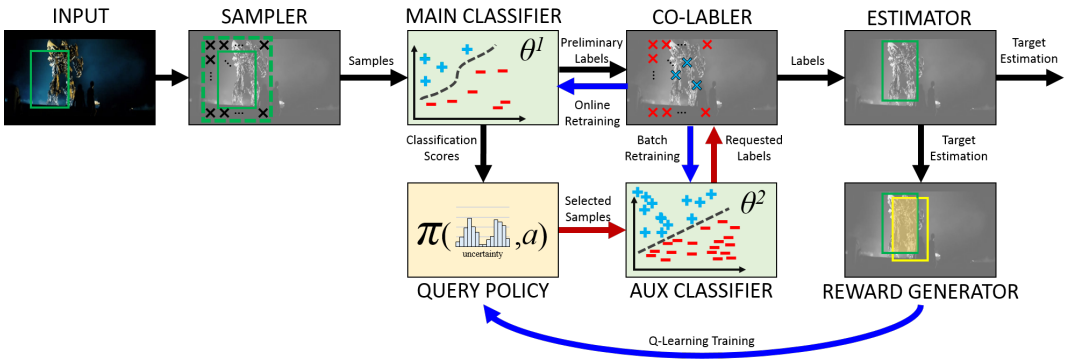

Fig. 2. Schematic of the system. The proposed tracker, collect samples from a pre-defined area around the last known target location. The classification scores are fed to query the policy unit which selects the best value for uncertainty margin hyperparameter. If needed, the aux classifier is queried for the label of the samples. The classifiers are then updated using co-labeled samples and the target state is estimated. In the training phase, the target estimation is compared to the ground truth and their normalized intersection is used as a reward to train the query policy Q-table.

ever, such setting put more burden on the main classifier to label the samples correctly and expose the tracker to the model drift, especially if the target observation is noisy or missing (e.g. in case of partial or temporal occlusions). On the contrary, larger δ increases the number of queries from the aux classifier, exploiting older data at the expense of the speed.

2.2. Uncertainty Margin Adjustment via Q-Learning

Reinforcement learning has been used in visual tracking to learn when to update the tracker [15], learn an early decision policy for different frames [16] and to tune tracking hyperparameters [17]. Here we formulate a Q-learning agent to adjust the uncertainty margin for the active co-tracker.

At each time t, the agent take an action a_t based on the state S_t of the environment, and the environment gives the reward $r(S_t, a_t)$ and updates its state to S_{t+1}. The agent chooses its action w.r.t its policy $\pi(a_t|S_t)$ to maximize the cumulative reward $R_t = \sum_{i=t}^{T} \gamma^{i-t} r(S_i, a_i)$ where $0 < \gamma \leq 1$ is the discount factor to weigh more on earlier rewards. Q-learning proposed to calculate Q-values, the expected maximum scores for each action a_t in state S_t, as

$$Q(S_t, a_t) = r(S_t, a_t) + \gamma Q(S_{t+1}, a_{t+1}) \quad (5)$$

State: The state $S_t \in \mathbf{S}_t$ of the environment is explained using an n_b-bin histogram of uncertainty measurements of main classifier for all samples $\mathbf{x}_t$,

$$S_t = f\Big(\text{hist}\big(1 - 2|h(\mathbf{x}_t|\theta_t^1) - \tau|\big)\Big)$$

To eliminate the effect of the stochastic sampling on the uncertainty histogram, a deterministic sampling approach is used, which obtain n_s equi-distance samples from an area 3x bigger than the target size centered on its last known position. The states are defined by the descriptive features of the histogram $f(.)$, i.e. its mean, variance, and shape.

Action: Actions $a_t \in \mathbf{A}_t$ are n_a equi-distanced values in range [0,½] to be assigned to margin δ.

Reward: During training time, the reward is defined as the intersection-over-union (IOU) of the target estimation and ground truth and if the IOU exceeds 90%, the reward triples. Contrarily, if the IOU drops under 50%, no reward is given to the learner, and if remains under 50% for five consecutive frames, the negative reward of -3 is given to the learner.

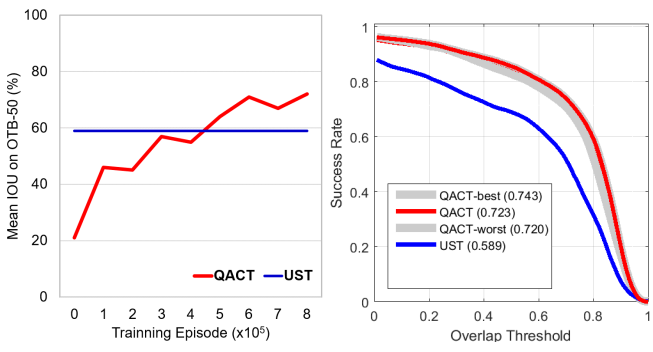

Fig. 3. The proposed Q-learning agent is trained on YouTubeBB datasets 10 independent runs and their performance on OTB-50 is shaded area (mean performance in red

.

2.3. Q-learning for Active Co-Tracking

In the proposed QACT tracker (Figure 2), the main classifier is a KNN classifier, with color names and HOG features. In each frame, t, n_s co-labeled samples are added and samples older than $t - \Delta$ are discarded from KNN. The aux classifier is a part-based SVM classifier [21], that is retrained every Δ frame with all of the co-labeled samples from the beginning of the tracking. In each frame, the target location and scale is defined with a weighted vote of the obtained positive samples, by considering classification score as their weight.

The parameters of the tracker, except those related to the Q-learning module, is defined using cross-validation on the

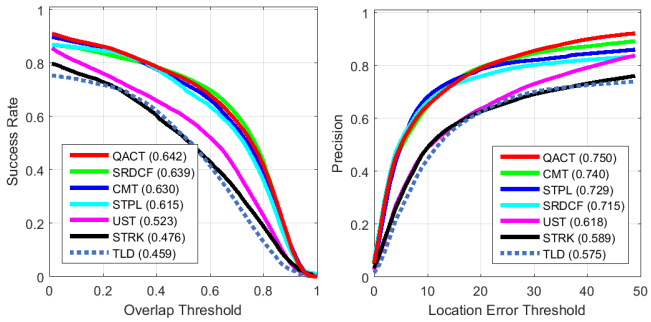

Fig. 4. Quantitative evaluation of trackers using precision plot (left) and success plot (right) for all videos in OTB-100 [24].

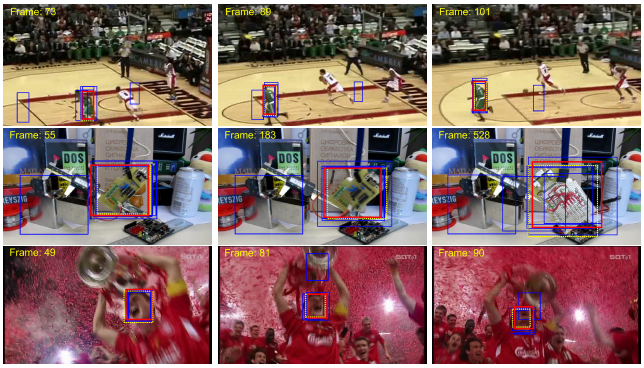

Fig. 5. Exemplary tracking results of proposed tracker (red), ground truth (yellow) and other evaluated trackers (blue) on several challenging video sequences. More results in http://ishiilab.jp/member/meshgi-k/qact.html.

OTB-50 dataset [22]. The Q-learning parameters are then set to fixed values of $n_b = 100$, $n_a = 25$, and $\gamma = 0.99$ and the Q-values are randomly initialized with a small positive white noise. The proposed tracker is trained on annotated frames of YouTubeBB dataset [23] to train the proposed Q-Learning method using the Boltzmann-Gumbel exploration [18] to exploit all the information present in the estimated Q-values with an annealed temperature parameter. For each of 800,000 training episode, we randomly sample a 20-seconds long clip with one annotation for each one second and provide zero rewards for the agent in between annotations. During runtime, the tracker greedily selects the action a_t^* which yields the highest expected reward, $a_t^* = \text{argmax}_{a_t' \in \mathbf{A}_t} Q(S_t, a_t')$.

3. EVALUATION

The proposed tracker is evaluated against its baselines, competitive trackers which leverage memory control for tracking, and finally the state-of-the-art in tracking-by-detection. The experiments are conducted using OTB-50 [22] and OTB-100 [24] benchmarks. Success and precision plots are used to compare the performance of the trackers.

Figure 3 shows that the proposed tracker (QACT) outperforms its baseline, UST (uncertainty sampling tracker) using Q-learning to tune the uncertainty margin parameter. The shaded area indicates 10 independent training runs for the Q-learning method, and the red plot indicates the mean of these runs and will be used hereafter as the QACT result to be compared with other algorithms. Table 1 contains the ablation study as well as a comparison with TLD [9], STRK [2], MEEM [8] and MSTR [10] that handles the memory of the tracker to improve tracking. UST, is the active co-tracker with dual memory explained in section 2.1, with KNN+ [5] as short-term classifier and SVM+ as the long-term one. As the table shows, QACT demonstrates superior performance on many aspects of tracking compared to its baselines and to other competitive methods. It is interesting to note the low-resolution case (LR) that the active co-tracker made from very low-performance trackers (i.e. UST), has a significantly higher performance from its classifiers (KNN+ and SVM+), still by online adjustment of uncertainty margin the performance of QACT has another large improvement. Finally, Figure 4 depicts the promising performance of proposed approach compared to the state-of-the-art in tracking by detection such as CMT [13], STPL [25] and SRDCF [26].

4. CONCLUSION

Active co-tracking combines the results of two classifiers, by using one of them on-demand triggered by the uncertainty of the other classifier. The threshold to trigger is of paramount importance to balance the information exchange between two classifiers, especially if they differ in accuracy, speed, model update frequency, and retraining complexity. We proposed a Q-learning mechanism that monitors the uncertainty state of the first classifiers to control the trigger. We elaborated the active co-tracking framework with dual memory, the design and training of the Q-learning for parameter tuning, and presented the superior performance of this tracker.

Table 1. Quantitative evaluation of trackers under different tracking challenges using AUC(%) of success plot on OTB-50. The first, second and third best results are highlighted. Scenario attributes indicate changes in illumination, scale, in-plane and out-of-plane rotation, deformation, occlusion, out-of-view, clutter, low resolution, fast motion and motion blur.

Attribute	KNN+	SVM+	TLD	STRK	UST	MEEM	MSTR	QACT
IV	24.1	39.9	47.8	53.0	58.5	62.3	72.6	72.6
SV	23.0	42.4	49.1	50.7	58.8	58.3	70.6	72.3
IPR	25.3	44.4	50.4	53.7	61.9	57.7	68.5	73.4
OPR	25.8	43.1	47.8	53.2	59.7	62.1	70.2	70.4
DEF	28.9	41.0	38.2	51.3	55.9	61.9	68.9	66.1
OCC	23.5	39.9	46.1	50.2	58.6	60.8	71.0	71.7
OV	27.7	52.0	53.5	51.5	56.9	68.5	73.3	71.1
LR	13.3	13.6	36.2	33.3	33.1	43.5	50.2	56.0
BC	30.7	40.0	39.4	51.5	48.0	67.0	71.7	71.1
FM	23.0	43.2	44.6	52.0	53.4	64.6	65.0	64.3
MB	22.9	35.0	41.0	46.7	45.2	62.8	65.2	65.6
ALL	27.8	43.5	49.3	54.8	58.9	61.7	71.8	72.3
FPS	76.6	3.8	21.2	11.3	28.3	14.2	8.3	27.1

5. REFERENCES

[1] Hamed Masnadi-Shirazi, Vijay Mahadevan, and Nuno Vasconcelos, "On the design of robust classifiers for computer vision," in *CVPR'10*, 2010.

[2] Sam Hare, Amir Saffari, and Philip HS Torr, "Struck: Structured output tracking with kernels," in *ICCV'11*, 2011.

[3] Helmut Grabner, Christian Leistner, and Horst Bischof, "Semi-supervised on-line boosting for robust tracking," in *ECCV'08*. 2008.

[4] Feng Tang, Shane Brennan, Qi Zhao, and Hai Tao, "Co-tracking using semi-supervised support vector machines," in *ICCV'07*.

[5] Kourosh Meshgi, Mirzaei Maryam Sadat, Shigeyuki Oba, and Shin Ishii, "Efficient asymmetric co-tracking using uncertainty sampling," in *IEEE ICSIPA'17*.

[6] Shai Avidan, "Ensemble tracking," *PAMI*, vol. 29, 2007.

[7] Kourosh Meshgi, Shigeyuki Oba, and Shin Ishii, "Efficient diverse ensemble for discriminative co-tracking," in *CVPR'18*.

[8] Jianming Zhang, Shugao Ma, and Stan Sclaroff, "Meem: Robust tracking via multiple experts using entropy minimization," in *ECCV'14*.

[9] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, "Tracking-learning-detection," *PAMI*, vol. 34, no. 7, pp. 1409–1422, 2012.

[10] Zhibin Hong, Zhe Chen, Chaohui Wang, Xue Mei, Danil Prokhorov, and Dacheng Tao, "Multi-store tracker (muster): a cognitive psychology inspired approach to object tracking," in *CVPR'15*.

[11] Shu Wang, Shaoting Zhang, Wei Liu, and Dimitris N Metaxas, "Visual tracking with reliable memories.," in *IJCAI*, 2016, pp. 3491–3497.

[12] David D Lewis and William A Gale, "A sequential algorithm for training text classifiers," in *ACM SIGIR'94*, 1994, pp. 3–12.

[13] Kourosh Meshgi, Shigeyuki Oba, and Shin Ishii, "Active discriminative tracking using collective memory," in *MVA'17*.

[14] Kourosh Meshgi, Maryam Sadat Mirzaei, and Shigeyuki Oba, "Information-maximizing sampling to promote tracking-by-detection," in *ICIP'18*, 2018.

[15] James Supancic III and Deva Ramanan, "Tracking as online decision-making: Learning a policy from streaming videos with reinforcement learning," in *ICCV'17*.

[16] Da Zhang, Hamid Maei, Xin Wang, and Yuan-Fang Wang, "Deep reinforcement learning for visual object tracking in videos," *arXiv preprint arXiv:1701.08936*, 2017.

[17] Xingping Dong, Jianbing Shen, Wenguan Wang, Yu Liu, Ling Shao, and Fatih Porikli, "Hyperparameter optimization for tracking with continuous deep q-learning," in *CVPR'18*.

[18] Nicolò Cesa-Bianchi, Claudio Gentile, Gábor Lugosi, and Gergely Neu, "Boltzmann exploration done right," in *NIPS'17*, 2017, pp. 6284–6293.

[19] Joost Van De Weijer, Cordelia Schmid, Jakob Verbeek, and Diane Larlus, "Learning color names for real-world applications," *IEEE TIP*, vol. 18, no. 7, pp. 1512–1523, 2009.

[20] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 1, pp. 886–893.

[21] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan, "Object detection with discriminatively trained part-based models," *PAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.

[22] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," in *CVPR'13*. IEEE, 2013, pp. 2411–2418.

[23] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke, "Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video," in *CVPR'17*. IEEE, 2017, pp. 7464–7473.

[24] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Object tracking benchmark," *PAMI*, 2015.

[25] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr, "Staple: Complementary learners for real-time tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1401–1409.

[26] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *ICCV'15*, 2015, pp. 4310–4318.