Active Collaborative Ensemble Tracking

Kourosh Meshgi, Maryam Sadat Mirzaei, Shigeyuki Oba, Shin Ishii
Graduate School of Informatics, Kyoto University
Yoshida-Honmachi, Sakyo Ward, Kyoto 606–8501, Japan
meshgi-k@sys.i.kyoto-u.ac.jp

Abstract

A discriminative ensemble tracker employs multiple classifiers, each of which casts a vote on all of the obtained samples. The votes are then aggregated in an attempt to localize the target object. Such method relies on collective competence and the diversity of the ensemble to approach the target/non-target classification task from different views. However, by updating all of the ensemble using a shared set of samples and their final labels, such diversity is lost or reduced to the diversity provided by the underlying features or internal classifiers' dynamics. Additionally, the classifiers do not exchange information with each other while striving to serve the collective goal, i.e., better classification. In this study, we propose an active collaborative information exchange scheme for ensemble tracking. This, not only orchestrates different classifiers towards a common goal but also provides an intelligent update mechanism to keep the diversity of classifiers and to mitigate the shortcomings of one with the others. The data exchange is optimized with regard to an ensemble uncertainty utility function, and the ensemble is updated via co-training. The evaluations demonstrate promising results realized by the proposed algorithm for the real-world online tracking.

1. Introduction

Visual tracking is one of the fundamental problems in computer vision, having a broad range of applications from human-computer interfaces, to automatic surveillance, video description/editing/indexing, and autonomous navigation systems. Generative trackers attempt to construct a robust object appearance model, or to learn it on-the-fly using advanced machine learning techniques such as subspace learning [37], hash learning [13], dictionary learning [43], and sparse learning [7]. On the other hand, discriminative models focus on target/background separation using correlation filters [10, 11, 24] or dedicated classifiers [33], which assist them to dominate the visual tracking benchmarks [46]. Tracking-by-detection methods primarily treat tracking as a detection problem to avoid having model object dynamics especially in the case of sudden motion changes, extreme deformations, and occlusions [5, 44].

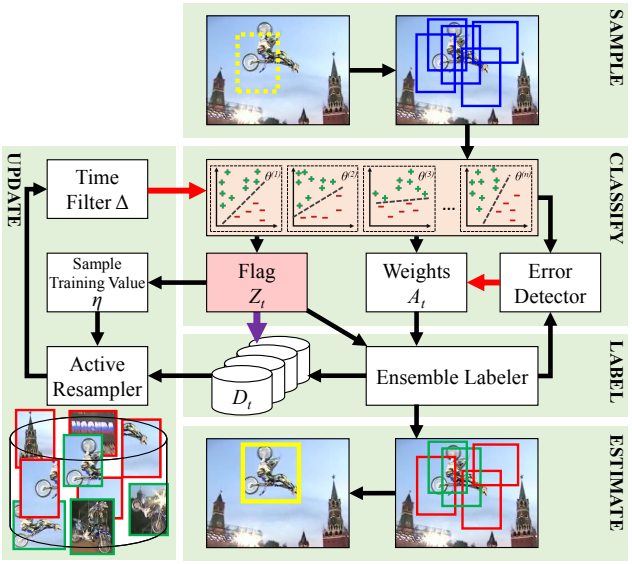

Figure 1. Schematic of the proposed tracker, ACET. Black arrows indicate the flow of the information, red arrows represent update signals, purple arrow represent the co-learning procedure.

However, these trackers are still vulnerable to illumination variation, in-plane and out-of-plane rotations, scale changes, and background clutter. Typical problems of tracking-by-detection schemes are *(i)* label noise, where inaccurate labels confuses the classifier and degrade the classification accuracy [44], *(ii)* self-learning loop, in which the classifier are re-trained by their own output from earlier frames, thus accumulating error over time [5], *(iii)* model drift, that is a side-effect of imperfect model update [29] and mismatch between model update frequency and target evolution rate [17], *(iv)* equal weights for all samples in evaluating the target [20] and training the classifier [26], despite the uneven contextual information in different samples, and *(v)* assuming stationary distribution of target, which does not hold for most of the real-world scenarios with drastic target appearance changes [5].

978-1-5386-2939-0/17/$31.00

Ensemble tracking [3, 4, 6, 15, 16, 27, 34, 38, 39, 48] and co-tracking [44] frameworks provide effective frameworks to tackle one or more of these challenges. In such frameworks, the self-learning loop is broken, and the labeling process is performed by eliciting the belief of a group of classifiers (ensemble) or another classifier that has a stronger belief about the sample's label (collaborator). However, these frameworks typically do not address some of the fundamental demands of tracking-by-detection approaches like a proper model update to avoid model drift or non-stationary of the target sample distribution. Here, the non-stationarity means that the appearance of an object may change so significantly that a negative sample in the current frame looks more similar to a positive example in the previous frames. Besides, ensemble classifiers do not exchange information, and collaborative classifiers entirely trust the other classifier to label the challenging samples for them and are susceptible to label noise.

We propose an effective integration of ensemble tracking and co-tracking, which involves the merits of each while their complementary nature counteracts the demerits of each other. Here, an ensemble of trackers is employed to label a sample. Those classifiers that are uncertain about the label, are excluded from the final decision about the sample's label, and the rest of the classifiers perform a weighted voting for labeling the sample. The contributing classifiers are then retrained with the newly labeled samples, based on the concept of co-training. If the classifiers disagree each other for most of the samples, it is likely that the target is mostly occluded. The use of an ensemble would undermine label noise problem, while co-training breaks the self-learning loop, provides an effective model update, and enforce the diversity into the ensemble. By providing different memory spans for different members of the ensemble, the model update rate of the ensemble is automatically adjusted to the evolution rate of the target, and limited memory horizon resolves the issues with non-stationarity of the observations. By limiting the classifiers' retraining data to only the most informative ones (i.e., to assume different "training values" for samples), the non-stationarity is better addressed, the generalizability of the ensemble is improved, and speed of the tracker is boosted.

We evaluated our proposed framework (active collaborative ensemble tracking or ACET) with other ensemble trackers and also the state-of-the-art in visual tracking on object tracking dataset [46] to demonstrate the effectiveness of this method, and discussed its merits and demerits.

2. Related Work

Ensemble-based Tracking: Using a (linear) combination of several (weak) classifiers with different associated weights has been proposed in a seminal work by Avidan [3]. Align with this study, constructing an ensemble by boosting [16], online boosting [27, 34], multi-class boosting [38] and multi-instance boosting [4, 49] provides better and better performance for ensemble trackers. The boosting may or may not couple with the ensemble changes such as feature adjustment [15] or addition/deletion of the ensemble's members [16, 39]. To date, boosting has been widely used in self-learning based tracking methods despite its low endurance against label noise [40]. An alternative way to tune the weights of an ensemble is via a Bayesian treatment [6]. Aside from using different features, the members of an ensemble may be constructed from randomized subsets of training data [32] or different time snapshots of a classifier evolving by time [48].

Training Value of Samples: Lapedriza et al. [26] discussed that different samples have different training value for a classifier, and using a wisely selected subset of samples for training/retraining the classifier outperforms the training with full dataset, for instance, due to mislabeled or inaccurately demarcated samples. Having a better training set for a tracking-by-detection classifier leads to enhanced generalization and faster convergence to the final performance which is suitable for converging to the piece-wise stationary target distribution (the distribution may change by every drastic change of target's appearance). To address this, researchers came by different approaches to provide good samples for tracking using context [18, 23], saliency maps [25], confidence maps [44], and optical flow [22]. Adaptive weights for the samples based on their appearance similarity to the target [35], occlusion state [30], and spatial distance to previous target location [47] have also been considered, however, selecting an efficient subset for classifier re-training have been ignored, as most of the trackers retrain on all of the data, a randomized subset of it [32], or in special cases re-sample the training data based on their boosting value [28]. A "clean" subset of training samples to re-train the classifier can achieve much higher performance than the full set [36, 51], therefore, a principled ordering and selection of the samples reduces the cost of labeling and accelerate the performance with smaller re-training sample size [45]. Different studies have tried to provide this small clean subset by different approaches: pruning outliers [12] and hard-to-learn samples [1], learning easy-to-classify examples first (as known as the Curriculum learning) [8], treating samples as noisy observations [14], defining a training value for each sample by treating each sample as a separate classifier [26], and robust loss functions for special classifiers (e.g., SVMs). Arguably, the most common setting is active learning, which selects the training samples to be labeled/selected at each step for higher gains in performance. Some approaches focus on learning the hardest examples first (e.g., those closest to the decision boundary), whereas some others gauge the information contained in the sample and select the most informative ones first. For in-

stance, in the case of an ensemble of classifiers, the samples for which the ensemble disagrees the more, contains more information about how to train the ensemble. This concept is known as query-by-committee [41] that tries to provide the best classifier with as few labeled instances as possible.

3. Proposed Method

A tracking-by-detection algorithm usually estimates the target state $\mathbf{p}_t$ in time $t \in \{1, \dots, T\}$ by obtaining several samples $\mathbf{p}_t^j \in \mathcal{P}_t$, scoring them $s_t^j \in \mathcal{S}_t$, labeling them $\ell_t^j \in \mathcal{L}_t$, and aggregating them, $\mathbf{p}_t = \psi(\mathcal{P}_t|\mathcal{S}_t, \mathcal{L}_t)$. To obtain a sample, a distribution or region-of-interest $\mathcal{Y}_t$ is sampled to obtain a transformation $\mathbf{y}_t^j \sim \mathcal{Y}_t$ that defines the state of the sample compared to the previous target location, $\mathbf{p}_{t-1}^j = \mathbf{p}_{t-1} \circ \mathbf{y}_t^j$, and the sample appearance is defined as $\mathbf{x}_t^j \in \mathcal{X}_t$. This sample is then evaluated by the classifier θ_t with scoring function $h : \mathcal{X}_t \to \mathbb{R}$,

$$s_t^j = h(\mathbf{x}_t^j|\theta_t). \tag{1}$$

Based on the score, a label ℓ_t^j is assigned to the sample. For supervised-learning classifiers [2], the label is either positive (target) or negative (background), but semi-supervised classifiers (e.g., [17, 38]) or multi-instance learning (e.g., [4, 50]) allow the samples, which the classifier is uncertain about, to remain unlabeled by the classifier,

$$\ell_t^j = \begin{cases} +1 & , s_t^j > \tau_u \\ -1 & , s_t^j < \tau_l \\ 0 & , \text{otherwise} \end{cases} \tag{2}$$

in which τ_l and τ_u denotes the lower and upper thresholds respectively. The unlabeled data are either discarded, used for later stages of tracking, or labeled by other mechanisms embedded in the tracker [17, 42, 44]. The target state, as mentioned, is estimated using $\psi(\mathcal{P}_t|\mathcal{S}_t, \mathcal{L}_t)$, and the classifier θ_t is updated by the all or a subset of the labeled data denoted by $\xi_t \subseteq \mathcal{P}_{\{1..t\}}$,

$$\theta_{t+1} = u(\theta_t, \mathcal{X}_{\xi_t}, \mathcal{L}_{\xi_t}) \tag{3}$$

where $u(.)$ is the model update function. The subset ξ_t may involve all new data for online trackers ($\xi_t = \mathcal{P}_t$), a subset of the new data ($\xi_t \subset \mathcal{P}_t$) or recent data ($\xi_t \subset \mathcal{P}_{\{t-\Delta,\dots,t\}}$), and keyframe data ($\xi_t = \mathcal{P}_{\{k1,k2,\dots\}}$) [21, 31].

3.1. Ensemble Discriminative Tracking

A popular approach to strengthen the classification in tracking-by-detection frameworks is to construct an ensemble of different (weak) classifiers $\mathcal{C} = \{\theta_t^{(1)}, \dots, \theta_t^{(n)}\}$, and combine their opinion about a sample by voting,

$$s_t^j = \sum_{c=1}^{n} \text{sign}\big(h(\mathbf{x}_t^j|\theta_t^{(c)})\big). \tag{4}$$

Algorithm 1: ACET

input : Ensemble models $\mathcal{C} = \{\theta_t^{(c)}\}$
input : Target position in previous frame $\mathbf{p}_{t-1}$
output: Target position in current frame $\mathbf{p}_t$

for $j \leftarrow 1$ **to** $N_{samples}$ **do**
- *Draw sample* $\mathbf{p}_t^j = \mathbf{p}_{t-1}^j \circ \mathbf{y}_t^j$ *s.t.* $\mathbf{y}_t^j \sim \mathcal{Y}_t$
- *Calculate the classification score of members of* $\mathcal{C}$
- *Indicate the uncertainty flag* $z_t^{(c,j)}$ (eq(8))
- *Calculate ensemble's score* s_t^j *and label* ℓ_t^j (eq(9))
- *Calculate sample's informativeness* η_t^j

for $c \leftarrow 1$ **to** n **do**
- *Obtain the error* $e_t^{(c)}$ *and weight* $\alpha_t^{(c)}$ (eq(12)(13))
- *Obtain the informative sample set* $\mathcal{D}_t^{(c)}$
- *Update the classifier* $\theta_{t+1}^{(c)}$ (eq(11))

if $\frac{1}{n}\sum_{c=1}^{n} e_t^{(c)} \leq \tau_{occ}$ **then** target is not occluded
- *Estimate target state* $\hat{\mathbf{p}}_t$ (eq(10))

In most of the cases, the weak classifiers are linearly combined with different associated weights,

$$s_t^j = \sum_{c=1}^{n} \alpha_t^{(c)} \text{sign}\big(h(\mathbf{x}_t^j|\theta_t^{(c)})\big), \tag{5}$$

where the weights $\alpha_t^{(c)} \in A_t$ are tuned using boosting [3, 16, 32] or Bayesian treatment [5]. A larger weight implies that the corresponding classifier of the ensemble is more discriminative, hence more useful. The labels are calculated from eq(2) with τ_l^c and τ_u^c as the lower and upper thresholds for the ensemble score.

Finally, each classifier's model is updated independently,

$$\theta_{t+1}^{(c)} = u(\theta_t^{(c)}, \mathcal{X}_{\xi_t}, \mathcal{L}_{\xi_t}) \tag{6}$$

indicating that all of the ensemble members are trained with a similar set of samples ξ_t.

3.2. Co-Training

Built on co-training principle [9], collaborative tracking (co-tracking) provides a framework in which the classifiers exchange their information to promote tracking results and break self-learning loop. In this two-classifier framework [44], the challenging samples for one classifier are labeled by the other one, i.e., if a classifier finds a sample difficult to label, it relies on the other classifier to label it for this frame and similar samples in the future.

$$s_t^j = \begin{cases} \alpha_t^{(2)} h(\mathbf{x}_t^j|\theta_t^{(2)}) & , \tau_l < h(\mathbf{x}_t^j|\theta_t^{(1)}) < \tau_u \\ \alpha_t^{(1)} h(\mathbf{x}_t^j|\theta_t^{(1)}) & , \tau_l < h(\mathbf{x}_t^j|\theta_t^{(2)}) < \tau_u \\ \sum_{c=1}^{2} \alpha_t^{(c)} h(\mathbf{x}_t^j|\theta_t^{(c)}) & , \text{otherwise} \end{cases} \tag{7}$$

The collaborative label is obtained by applying eq(2) on this score. The weights of the classifiers are adjusted by comparing the labels of each classifier to the collaborative label. Eventually, the trackers that label a sample are getting updated by it.

3.3. Active Collaborative Ensemble Tracker

The proposed tracker, ACET, is an ensemble tracker in which the co-training rule provides the samples for retraining each classifier, and active learning selects the most informative ones to improve the generalization and efficiency of the model update. Furthermore, by forgetting older samples with different memory horizons, the ensemble is diversified and non-stationary target appearance distributions are better accommodated.

Here, the ensemble $\mathcal{C} = \{\theta_t^{(1)}, \ldots, \theta_t^{(n)}\}$ is constructed of similar classifiers but with different memory spans $\{\Delta^{(1)}, \ldots, \Delta^{(n)}\}$. Sample $\mathbf{x}_t^j$ is obtained from a Gaussian field centered on last target state, $\mathcal{Y}_t = \mathcal{N}(\mathbf{p}_{t-1}, \Sigma_{search})$. This sample is then scored by all members of the ensemble. Those members that are uncertain about labeling the sample are marked by flag $z_t^{(c,j)} \in \mathcal{Z}_t^{(c)}$,

$$z_t^{(c,j)} = \begin{cases} 0 & , \tau_l < h(\mathbf{x}_t^j|\theta_t^{(c)}) < \tau_u \\ 1 & , \text{otherwise} \end{cases} \tag{8}$$

which in turn helps to calculate the score of ensemble,

$$s_t^j = \sum_{c=1}^{n} \alpha_t^{(c)} z_t^{(c,j)} \text{sign}\big(h(\mathbf{x}_t^j|\theta_t^{(c)})\big), \tag{9}$$

and label it using eq(2) with τ_l^c and τ_u^c as thresholds.

Since the number of samples is limited, an approximation of the target location $\hat{\mathbf{p}}_t$ is obtained by calculating the expectation of target, i.e., by taking a weighted average of the target candidates (i.e., positive samples).

$$\hat{\mathbf{p}}_t = \mathbb{E}[\mathbf{p}_t^j] = \sum_{\forall j, \ell_t^j > 0} s_t^j \mathbf{p}_t^j. \tag{10}$$

Following the rule of co-training, only the classifiers that engaged in labeling a sample ($z_t^{(c,j)} = 1$) should be updated with that sample. However, not all the samples are equally useful to train the ensemble. For instance, a sample for which half of the ensemble are uncertain about its label would be better for training compared to a sample for which only one of the classifiers is uncertain. To measure the "informativeness" of a sample, we count the number of the classifiers that elicit a strong belief about its label, $\eta_t^j = \sum_{c=1}^{n} z_t^{(c,j)}$. Then for training of each classifier of the ensemble, based on query-by-committee concept [41], those samples with $z_t^{(c,j)} = 1$ are sorted based on η_t^j and the first m are used for retraining (stored in $\mathcal{D}_t^{(c)}$).

$$\theta_{t+1}^{(c)} = u(\theta_t^{(c)}, \mathcal{D}^{(c)}_{\{t-\Delta^{(c)},\ldots,t\}}, \mathcal{L}'^{(c)}_{\{t-\Delta^{(c)},\ldots,t\}}) \tag{11}$$

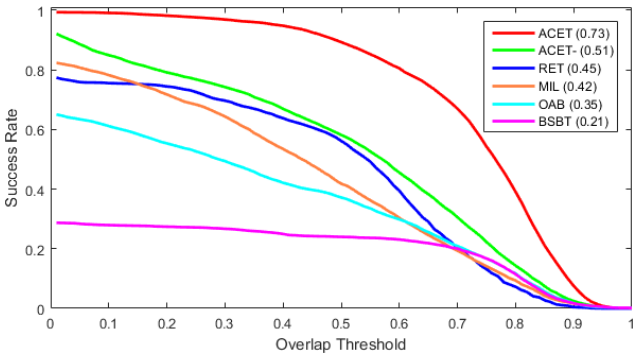

Figure 2. Quantitative evaluation of ensemble trackers using success plot for 13 video sequences.

where $\mathcal{L}'^{(c)}_t$ contains the labels of the samples in $\mathcal{D}_t^{(c)}$. As a result, the diversity of the ensemble is increased by co-training, selective updating, and different memory horizons.

The weights of the classifier is calculated based on its agreement with the whole ensemble. The error of each classifier is determined by

$$e_t^{(c)} = \sum_j \mathbb{1}\Big(\text{sign}\big(h(\mathbf{x}_t^j|\theta_t^{(c)})\big) \neq \ell_t^j\Big) \tag{12}$$

in which $\mathbb{1}(.)$ is the indicator function. Then the weight of each classifier is calculated as,

$$\alpha_t^{(c)} = 1 - \frac{e_t^{(c)} + \epsilon}{\sum_{i=1}^{n} e_t^{(i)} + \epsilon} \tag{13}$$

in which ϵ is a small constant. If the error average of the ensemble is very high, $\frac{1}{n}\sum_{c=1}^{n} e_t^{(c)} > \tau_{occ}$, then the target is likely to be mostly occluded. Algorithm 1 and Figure 1 summarize the proposed tracker.

4. Experiments

The proposed framework is comprised of several parameters: *(i)* Sampling parameters such as number $N_{samples}$ and sampling distribution covariance Σ_{search}, *(ii)* Ensemble parameters such as classifier count n, their memory spans $\Delta_t^{(c)}$ and labeling thresholds $\tau_l, \tau_u, \tau_l^{(c)}, \tau_u^{(c)}$, and *(iii)* Tracking parameters such as occlusion threshold τ_{occ} and retraining subset size m. On a P-IV PC at 3.5 GHz, ACET achieved 37.16 fps with a Mathlab/C++ implementation.

4.1. Evaluation Protocol

To evaluate the tracker, we employ success plot which measures the performance of a tracker which is a combination of its accuracy, reliability, and scale adaptation.

The experiments are conducted to object tracking benchmark videos [46], which become a de-facto standard in comparing the performance of the trackers, and includes several subcategories, exploiting the performance of the

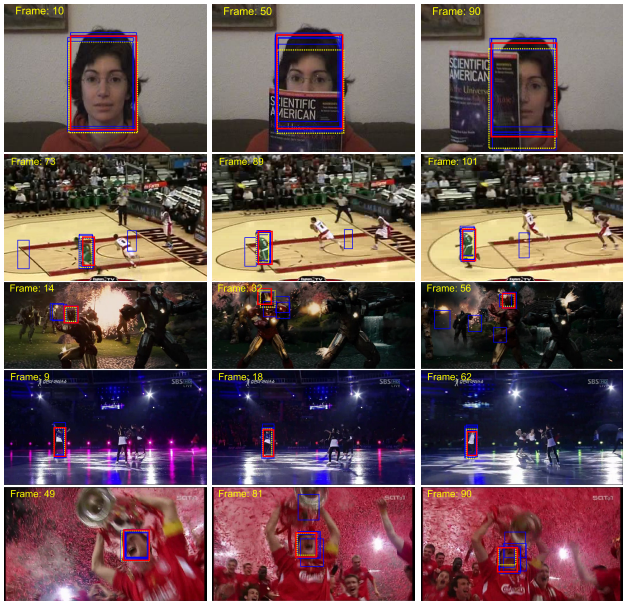

Figure 3. Qualitative results of evaluated algorithms on several challenging videos **(top to down)** *FaceOcc1* with severe occlusions, *Basketball* with deformations, *Ironman* with extreme rotations, *Skating1* with drastic illumination changes, and *Soccer* with background clutter. In these sequences the red box depicts the ACET against other trackers (blue), and the ground truth (yellow). See http://ishiilab.jp/member/meshgi-k/acet/.

trackers against various visual tracking challenges: illumination and scale variations (IV, SV), in- and out-of-plane rotations (IPR, OPR), fast motion and motion blur (FM, MB), deformations and low-resolution (DEF, LR), occlusion and shear problem (OCC, OV), and background clutter (BC).

4.2. Comparison with other Ensemble Trackers

For this experiment, we compare the proposed tracker (ACET) with online boosting tracker (OAB [16]) that utilizes different features to construct weak classifiers as ensemble members, and randomized ensemble tracker (RET [5]) that make different strong classifiers out of a pool of weak classifiers, and construct the ensemble out of those strong classifiers. We also include MIL [4] and BSBT [42] to represent ensemble trackers based on semi-supervised and multi-instance learning. Here, we implement a version of our tracker (ACET-) that use the same feature set to construct different members of the ensemble and the active learning and memory horizon subsampling is disabled. For the sake of compatibility with published RET results, 13 overlapping sequences with OTB-50 have been used.

Figure 2 illustrates that the proposed framework works better than other ensemble methods regardless of the ensemble member construction. Yet, it is evident that using all features along with subsampling schemes for re-training classifier (by active learning and different memory spans) significantly improve the tracking performance.

4.3. Comparison with State-of-the-art

To provide a fair comparison, we compared ACET with state-of-the-art tracking-by-detection algorithms TLD [22], STRK [19], MEEM [48], correlation filter trackers SRDCF [10], CCOT [11] and multi-memory tracker MUSTer [21]. The comparison based on the area under the curve of the success plot is presented in Table 1. It is evident that ACET outperforms the other trackers in most of the categories and in total performance over the 50 videos. Since the tracker utilized two features sensitive to low resolution, (as expected) it is not able to perform well in LR category. The good performance of the tracker in target appearance change categories (IV, DEF, OCC, OV) can be attributed to the robustness of ensemble due to co-learning, while the good results on transformation categories (SV, IPR, OPR) can be attributed to good generalization obtained by active learning sample selection for ensemble retraining. Different memory spans helped the tracker to dominate motion categories (FM, MB), and a robust diverse ensemble obtained by all of these approaches resolved background clutter (BC) effectively. The quality of results is shown in Figure 3.

5. Conclusions

In this study, we proposed a novel framework for ensemble tracking, in which the classifiers co-learns using only the most informative samples to enhance generalization and accelerate convergence to non-stationary distributions of target appearance. Co-learning reduces the label noise, and break the self-learning loops that cause model drift, and together with different memory spans for the ensemble provides a robust model update scheme for ensemble tracking. The proposed tracker, ACET, outperformed other ensemble trackers and state-of-the-art on OTB-50 [46] database.

Acknowledgments

This study is partly supported by the Japan NEDO and the “Post-K application development for exploratory challenges” project of the Japan MEXT.

Table 1. Quantitative evaluation of state-of-the-art under different visual tracking challenges using AUC of success plot.

Attribute	TLD	STRK	MEEM	MUSTer	SRDCF	CCOT	Ours
IV	0.48	0.53	0.62	0.73	0.70	0.75	**0.78**
DEF	0.38	0.51	0.62	**0.69**	0.67	**0.69**	**0.69**
OCC	0.46	0.50	0.61	0.71	0.70	0.76	**0.77**
SV	0.49	0.51	0.58	0.71	0.71	0.76	**0.77**
IPR	0.50	0.54	0.58	0.69	0.70	0.72	**0.77**
OPR	0.48	0.53	0.62	0.70	0.69	0.74	**0.77**
OV	0.54	0.52	0.68	0.73	0.66	0.79	**0.84**
FM	0.45	0.52	0.65	0.65	0.63	0.72	**0.79**
MB	0.41	0.47	0.63	0.65	0.69	0.72	**0.77**
BC	0.39	0.52	0.67	0.72	0.80	0.70	**0.73**
LR	0.36	0.33	0.43	0.50	0.58	**0.70**	0.44
ALL	0.49	0.55	0.62	0.72	0.70	0.75	**0.76**

References

[1] A. Angelova, Y. Abu-Mostafam, and P. Perona. Pruning training sets for learning of object categories. In *CVPR'05*.

[2] S. Avidan. Support vector tracking. *PAMI*, 2004.

[3] S. Avidan. Ensemble tracking. *PAMI*, 29, 2007.

[4] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR'09*, 2009.

[5] Q. Bai, Z. Wu, S. Sclaroff, M. Betke, and C. Monnier. Randomized ensemble tracking. In *ICCV'13*, 2013.

[6] Y. Bai and M. Tang. Robust tracking via weakly supervised ranking svm. In *CVPR'12*, 2012.

[7] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In *CVPR'12*.

[8] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *ICML'09*, 2009.

[9] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT'98*, 1998.

[10] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV'15*, pages 4310–4318, 2015.

[11] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV'16*.

[12] F. De la Torre and M. J. Black. Robust principal component analysis for computer vision. In *ICCV'01*, 2001.

[13] J. Fang, H. Xu, Q. Wang, and T. Wu. Online hash tracking with spatio-temporal saliency auxiliary. *CVIU*, 2017.

[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010.

[15] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *PAMI*, 2011.

[16] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *BMVC'06*, 2006.

[17] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV'08*. 2008.

[18] H. Grabner, J. Matas, L. Van Gool, and P. Cattin. Tracking the invisible: Learning where the object might be. In *CVPR'10*, 2010.

[19] S. Hare, A. Saffari, and P. H. Torr. Struck: Structured output tracking with kernels. In *ICCV'11*, 2011.

[20] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV'12*, pages 702–715. Springer, 2012.

[21] 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 *CVPR'15*.

[22] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *PAMI*, 34(7):1409–1422, 2012.

[23] H. Kiani Galoogahi, A. Fagg, and S. Lucey. Learning background-aware correlation filters for visual tracking. *arXiv*, 2017.

[24] H. Kiani Galoogahi, T. Sim, and S. Lucey. Correlation filters with limited boundaries. In *CVPR'15*, 2015.

[25] J. Kwon, R. Timofte, and L. Van Gool. Leveraging observation uncertainty for robust visual tracking. *CVIU*, 2017.

[26] A. Lapedriza, H. Pirsiavash, Z. Bylinskii, and A. Torralba. Are all training examples equally valuable? *arXiv*, 2013.

[27] C. Leistner, A. Saffari, and H. Bischof. Miforests: Multiple-instance learning with randomized trees. In *ECCV'10*, 2010.

[28] C. Leistner, A. Saffari, P. Roth, and H. Bischof. On robustness of on-line boosting: a competitive study. In *ICCVw'09*.

[29] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *PAMI*, 2004.

[30] K. Meshgi, S.-I. Maeda, S. Oba, and S. Ishii. Data-driven probabilistic occlusion mask to promote visual tracking. In *CRV'16*.

[31] K. Meshgi, S. Oba, and S. Ishii. Active discriminative tracking using collective memory. In *MVA'17*.

[32] K. Meshgi, S. Oba, and S. Ishii. Robust discriminative tracking via query-by-committee. In *AVSS'16*, 2016.

[33] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR'16*.

[34] N. C. Oza. Online bagging and boosting. In *SMC'05*, 2005.

[35] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *ECCV'02*.

[36] N. Razavi, J. Gall, P. Kohli, and L. Van Gool. Latent hough transform for object detection. *ECCV'12*.

[37] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *IJCV*, 2008.

[38] A. Saffari, C. Leistner, M. Godec, and H. Bischof. Robust multi-view boosting with priors. In *ECCV'10*. 2010.

[39] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof. On-line random forests. In *ICCVw'09*.

[40] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. Prost: Parallel robust online simple tracking. In *CVPR'10*.

[41] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *COLT'92*, pages 287–294. ACM, 1992.

[42] S. Stalder, H. Grabner, and L. Van Gool. Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In *ICCVw'09*.

[43] A. Taalimi, H. Qi, and R. Khorsandi. Online multi-modal task-driven dictionary learning and robust joint sparse representation for visual tracking. In *AVSS'15*, 2015.

[44] F. Tang, S. Brennan, Q. Zhao, and H. Tao. Co-tracking using semi-supervised support vector machines. In *ICCV'07*.

[45] S. Vijayanarasimhan and K. Grauman. Cost-sensitive active visual category learning. *IJCV*, 2011.

[46] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR'13*, pages 2411–2418. IEEE, 2013.

[47] Y. Wu, M. Pei, M. Yang, and Y. Jia. Robust discriminative tracking via landmark-based label propagation. *TIP*, 2015.

[48] J. Zhang, S. Ma, and S. Sclaroff. Meem: Robust tracking via multiple experts using entropy minimization. In *ECCV'14*.

[49] K. Zhang and H. Song. Real-time visual tracking via online weighted multiple instance learning. *PR*, 2013.

[50] K. Zhang, L. Zhang, M.-H. Yang, and Q. Hu. Robust object tracking via active feature selection. *CSVT*, 2013.

[51] X. Zhu, C. Vondrick, D. Ramanan, and C. C. Fowlkes. Do we need more training data or better models for object detection?. In *BMVC'12*.