Tracking-by-Trackers with a Distilled and Reinforced Model

Tracking-by-Trackers with a Distilled and Reinforced Model

Matteo Dunnhofer[0000−0002−1672−667X], Niki Martinel[0000−0002−6962−8643], Christian Micheloni[0000−0003−4503−7483]
{matteo.dunnhofer, niki.martinel, christian.micheloni}@uniud.it

Machine Learning and Perception Lab, University of Udine, Italy

Abstract. Visual object tracking was generally tackled by reasoning independently on fast processing algorithms, accurate online adaptation methods, and fusion of trackers. In this paper, we unify such goals by proposing a novel tracking methodology that takes advantage of other visual trackers, offline and online. A compact student model is trained via the marriage of knowledge distillation and reinforcement learning. The first allows to transfer and compress tracking knowledge of other trackers. The second enables the learning of evaluation measures which are then exploited online. After learning, the student can be ultimately used to build (i) a very fast single-shot tracker, (ii) a tracker with a simple and effective online adaptation mechanism, (iii) a tracker that performs fusion of other trackers. Extensive validation shows that the proposed algorithms compete with real-time state-of-the-art trackers.

1 Introduction

Visual object tracking corresponds to the persistent recognition and localization –by means of bounding boxes– of a target object in consecutive video frames. This problem comes with several different challenges including object occlusion and fast motion, light changes, and motion blur. Additionally, real-time constraints are often posed by the many practical applications, such as video surveillance, behavior understanding, autonomous driving, and robotics.

In the past, the community has proposed solutions emphasizing different aspects of the problem. Processing speed was pursued by algorithms like correlation filters [1,2,3,4,5] or offline methods such as siamese convolutional neural networks (CNNs) [6,7,8,9,10,11,12]. Improved performance was attained by online target adaptation methods [13,14,15,16,17]. Higher tracking accuracy and robustness were achieved by methods built on top of other trackers [18,19,20,21,22]. All these characteristics belong to an optimal tracker but they were studied one independently from the other. The community currently lacks a general framework to tackle them jointly. In this view, a single model should be able to (i) track an object in a fast way, (ii) implement simple and effective online adaptation mechanisms, (iii) apply decision-making strategies to select tracker outputs.

It is a matter of fact that a large number of tracking algorithms has been produced so far, with different principles exploited. Preliminary solutions were

based on mean shift algorithms [23], key-point [24] or part-based methods [25,26], or SVM learning [27]. Later, correlation filters gained popularity thanks to their fast processing times [1,2,3,4,5]. Since more recently, CNNs have been exploited to extract efficient image features. This kind of representation has been included in deep regression networks [6,7], online tracking-by-detection methods [13,14], solutions that treat visual tracking as a reinforcement learning (RL) problem [28,29,30,31,32,33], CNN-based discriminative correlation filters [34,15,16,17], and in siamese CNNs [8,9,10,12,35,36]. Other methods tried to take advantage of the output produced by multiple trackers [18,19,20,21,22]. Thus, one can imagine that different trackers incorporate different knowledge, and this may constitute a valuable resource to leverage during tracking.

Lately, the knowledge distillation (KD) framework [37] was introduced in the deep learning panorama as paradigm for, among the many [38,39,40,41], knowledge transferring between models [42] and model compression [43,44,45]. The idea boils down into considering a student model and one or more teacher models to learn from. Teachers explicit their knowledge through demonstrations on a never seen before transfer set. Through specific loss functions, the student is set to learn a task by matching the teachers' output and the ground-truth labels. As visual tracking requires fast and accurate methods, KD can be a valuable tool to transfer the tracking ability of more accurate teacher trackers to more compact and faster student ones. However, the standard setup of KD does not provide methods to exploit teachers online, but just offline. This makes this methodology unsuitable for tracking, which has been shown to benefit from both offline and online methods [16,17,15,28,32]. In contrast to such an issue, RL techniques offer established methodologies to optimize not only policies but also policy evaluation functions [46,47,48,49,50], which are then used to extract decision strategies. Along with this, RL also gives the possibility to maximize arbitrary and non-differentiable performance measures, and thus more tracking oriented objectives can be defined.

For the aforementioned motivations, the contribution of this paper is a novel tracking methodology where a student model exploits off-the-shelf trackers offline and online (tracking-by-trackers). The student is first trained via an effective strategy that combines KD and RL. After that, the model's compressed knowledge can be used interchangeably depending on the application's needs. We will show how to exploit the student in three setups which result in, respectively, (i) a fast tracker (TRAS), (ii) a tracker with a simple online mechanism (TRAST), and (iii) a tracker capable of expert tracker fusion (TRASFUST). Through extensive evaluation procedures, it will be demonstrated that each of the algorithms competes with the respective state-of-the-art class of trackers while performing in real-time.

2 Related Work

Visual Tracking. Here we review the trackers most related to ours. The network architecture implemented by the proposed student model takes inspiration

from GOTURN [6] and RE3 [7]. These regression-based CNNs were shown to capture the target's motion while performing very fast. However, the learning strategy employed optimizes parameters just for coordinate difference. Moreover, great amount of data is needed to make such models achieve good accuracy. In contrast, our KD-RL-based method offers parameter optimization for overlap maximization and extracts previously acquired knowledge from other trackers requiring less labeled data. Online adaptation methods like discriminative model learning [51,13,14] or discriminative correlation filters [15,16,17] have been studied extensively to improve tracking accuracy. These procedures are time-consuming and require particular assumptions and careful design. We propose a simple online update strategy where an off-the-shelf tracker is used to correct the performance of the student model. Our method does not make any assumption on such tracker, and thus it can be freely selected to adapt to application needs. Present fusion models exploit trackers in the form of discriminative trackers [18], CNN feature layers [52], correlation filters [53] or out-of-the-box tracking algorithms [19,20,21,22]. However, such models work just online and do not take advantage of the great amount of offline knowledge that expert trackers can provide. Furthermore, they are not able to track objects without them. Our student model addresses these issues thanks to the decision making strategy learned via KD and RL.

KD and RL. We review the learning strategies most related to ours. KD techniques have been used for transferring knowledge between teacher and student models [54,37], where the supervised learning setting was employed more [42,38,39,40] than the setup that uses RL [55,56]. In the context of computer vision, KD was employed for action recognition [57,58], object detection [43,59], semantic segmentation [60,61], person re-identification [62]. In the visual tracking panorama, KD was explored in [63,64] to compress, CNN representations for correlation filter trackers and siamese network architectures, respectively. However, these works offer methods involving teachers specifically designed as correlation filter and siamese trackers, and so cannot be adapted to generic-approach visual trackers as we propose in this paper. Moreover, to the best of our knowledge, no method mixing KD and RL is currently present in the computer vision literature. Our learning procedure is also related to the strategies that use deep RL to learn tracking policies [28,29,31,32,65]. Our formulation shares some characteristics with such methods in the markov decision process (MDP) definition, but our proposed learning algorithm is different as no present method leverages on teachers to learn the policy.

3 Methodology

The key point of this paper is to learn a simple and fast student model with versatile tracking abilities. KD is used for transferring the tracking knowledge of off-the-shelf trackers to a compressed model. However, as both offline and online strategies are necessary for tracking [16,17,28,32], we propose to augment the

KD framework with an RL optimization objective. RL techniques deliver unified optimization strategies to directly maximize a desired performance measure (in our case the overlap between prediction and ground-truth bounding boxes) and to predict the expectation of such measure. We use the latter as base for an online evaluation and selection strategy. Put in other words, combining KD and RL lets the student model extract a tracking policy from teachers, improve it, and express its quality through an overlap-based objective.

3.1 Preliminaries

Given a transfer set of videos $\mathcal{D} = \{\mathcal{V}_0, \cdots, \mathcal{V}_{|\mathcal{D}|}\}$, we consider the j-th video $\mathcal{V}_j = \left\{F_t \in \mathcal{I}\right\}_{t=0}^{T_j}$ as a sequence of frames F_t, where $\mathcal{I} = \{0, \cdots, 255\}^{w \times h \times 3}$ is the space of RGB images. Let $b_t = [x_t, y_t, w_t, h_t] \in \mathbb{R}^4$ be the t-th bounding box defining the coordinates of the top left corner, and the width and height of the rectangle that contains the target object. At time t, given the current frame F_t, the goal of the tracker is to predict b_t that best fits the target in F_t. We formally consider the student model as $\mathbf{s} : \mathcal{I} \rightarrow \mathbb{R}^4 \times \mathbb{R}$ that is a function which outputs the relative motion between b_{t-1} and b_t, and the performance evaluation v_t, when inputted with frame F_t. Similarly, we define the set of tracking teachers as $\mathbf{T} = \left\{\mathbf{t} : \mathcal{I} \rightarrow \mathbb{R}^4\right\}$ where each $\mathbf{t}$ is a function that, given a frame image, produces a bounding box estimate for that frame.

3.2 Visual Tracking as an MDP

In our setting, $\mathbf{s}$ is treated as an artificial agent which interacts with an MDP defined over a video $\mathcal{V}_j$. The interaction happens through a temporal sequence of states $s_1, s_2, \cdots, s_t \in \mathcal{S}$, actions $a_1, a_2, \cdots, a_t \in \mathcal{A}$ and rewards $r_1, r_2, \cdots, r_t \in [-1, 1]$. In the t-th frame, the student is provided with the state s_t and outputs the continuous action a_t which consists in the relative motion of the target object, i.e. it indicates how its bounding box, which is known in frame $t-1$, should move to enclose the target in the frame t. a_t is rewarded by the measure of its quality r_t. We refer this interaction process as the episode E_j, which dynamics are defined by the MDP $M_j = (\mathcal{S}, \mathcal{A}, r, f)$.

States. Every $s_t \in \mathcal{S}$ is defined as a pair of image patches obtained by cropping F_{t-1} and F_t using b_{t-1}. Specifically, $s_t = \rho(F_{t-1}, F_t, b_{t-1}, c)$, where $\rho(\cdot)$ crops the frames F_{t-1}, F_t within the area of the bounding box $b'_{t-1} = [x'_{t-1}, y'_{t-1}, c \cdot w_{t-1}, c \cdot h_{t-1}]$ that has the same center coordinates of b_{t-1} but which width and height are scaled by c. By selecting $c > 1$, we can control the amount of additional image context information to be provided to the student.

Actions and State Transition. Each $a_t \in \mathcal{A}$ consists in a vector $a_t = [\Delta x_t, \Delta y_t, \Delta w_t, \Delta h_t] \in [-1, 1]^4$ which defines the relative horizontal and vertical translations ($\Delta x_t, \Delta y_t$, respectively) and width and height scale variations

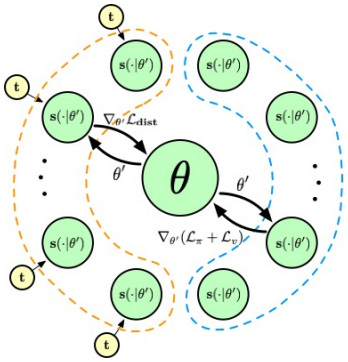

Fig. 1: Scheme of the proposed KD-RL-based learning framework. S students interact independently with M_j. After each E_j done, a copy θ' of the shared weights θ is sent to each one. Every t_{max} steps each student send the computed gradients to apply an update on θ. The distilling students (highlighted by the orange dashed contour) extract knowledge from teachers $\mathbf{t}$ by optimizing $\mathcal{L}_{\text{dist}}$. Autonomous students (circled with the blue dashed contour) learn an autonomous tracking policy by optimizing jointly $\mathcal{L}_\pi$ and $\mathcal{L}_v$.

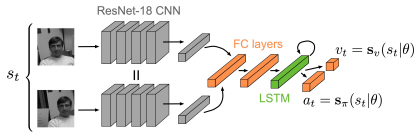

Fig. 2: The student architecture is composed by two branches of convolutional layers (gray boxes) with shared weights followed by, two fully-connected layers (orange boxes), an LSTM layer (in green) and two parallel fully connected layers for the prediction of v_t and a_t respectively.

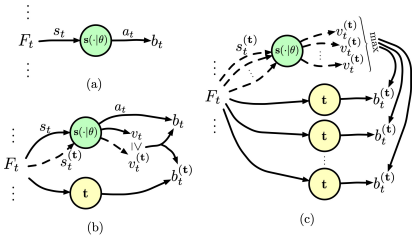

Fig. 3: Visual representation showing how the student and teachers are employed in the proposed trackers at every frame F_t. (a) represents TRAS, (b) TRAST, and (c) TRASFUST.

($\Delta w_t, \Delta h_t$, respectively) that have to be applied to b_{t-1} to predict b_t. The latter step is obtained through $\psi : \mathcal{A} \times \mathbb{R}^4 \rightarrow \mathbb{R}^4$.[1] After performing a_t, the student moves through f from s_t to s_{t+1} which is defined as the pair of cropped images obtained from F_t and F_{t+1} using b_t.

Reward. The reward function expresses the quality of a_t taken at s_t and it is used to feedback the student. Our reward definition is based on the Intersection-over-Union (IoU) metric computed between b_t and the ground-truth bounding box, denoted as g_t, i.e.,

$$\text{IoU}(b_t, g_t) = (b_t \cap g_t)/(b_t \cup g_t) \in [0, 1]. \tag{1}$$

At every interaction step t, the reward is formally defined as

$$r_t = r(b_t, g_t) = \begin{cases} \omega\left(\text{IoU}(b_t, g_t)\right) & \text{if IoU}(b_t, g_t) \geq 0.5 \\ -1 & \text{otherwise} \end{cases} \tag{2}$$

with $\omega(z) = 2(\lfloor z \rfloor_{0.05}) - 1$ that floors to the closest 0.05 digit and shifts the input range from $[0, 1]$ to $[-1, 1]$.

[1] Please refer to Appendix A.1 for the definition of $\psi(\cdot)$.

3.3 Learning Tracking from Teachers

The student $\mathbf{s}$ is first trained in an offline stage. Through KD, knowledge is transferred from $\mathbf{T}$ to $\mathbf{s}$. By means of RL, such knowledge is improved and the ability of evaluating its quality is also acquired. All the gained knowledge will be used for online tracking. We implement $\mathbf{s}$ as a parameterized function $\mathbf{s}(s_t|\theta) : \mathcal{S} \rightarrow \mathcal{A} \times \mathbb{R}$ that given s_t outputs at the same time the action a_t and state-value v_t. In RL terms, $\mathbf{s}$ maintains representations of both the policy $\mathbf{s}_\pi : \mathcal{S} \rightarrow \mathcal{A}$ and the state value $\mathbf{s}_v : \mathcal{S} \rightarrow \mathbb{R}$ functions. The proposed learning framework, which is depicted in Figure 1, provides a single offline end-to-end learning stage. S students are distributed as parallel and independent learning agents. Each one owns a set of learnable weights θ' that are used to generate experience by interacting with M_j. The obtained experience, in the form of $\nabla_{\theta'}$, is used to update asynchronously a shared set of weights θ. After ending E_j, each student updates its θ' by copying the values of the currently available θ. The entire procedure is repeated until convergence. This learning architecture follows the recent trends in RL that make use of distributed algorithms to speed up the training phase [66,50,67]. We devote half of the students, which we refer to as distilling students, in acquiring knowledge from the teachers' tracking policy. The other half, called autonomous students, learn to track by interacting with M_j autonomously.

Distilling Students. Each distilling student interacts with M_j by observing states, performing actions and receiving rewards just as an autonomous student. However, to distill knowledge independently from the teachers' inner structure, we propose the student to learn from the actions of $\mathbf{t} \in \mathbf{T}$, which are executed in parallel. In particular, $\mathbf{t}$ is exploited every t_{max} steps with the following loss function

$$\mathcal{L}_{\text{dist}} = \sum_{i=1}^{t_{max}} |a_t^{(\mathbf{t})} - \mathbf{s}_\pi(s_t|\theta')| \cdot m_i, \tag{3}$$

which is the L1 loss between the actions performed by the student and the actions $a_t^{(\mathbf{t})} = \phi(b_t^{(\mathbf{t})}, b_{t-1})$ that the teacher would take to move the student's bounding box b_{t-1} into the teacher's prediction $b_t^{(\mathbf{t})}$.[2] At every t, $\mathbf{t}$ is selected as

$$\mathbf{t} \in \mathbf{T} \ : \ \text{IoU}(b_t^{(\mathbf{t})}, g_t) = \max_{\mathbf{t} \in \mathbf{T}} \text{IoU}(b_t^{(\mathbf{t})}, g_t) \tag{4}$$

as we would like to learn always from the best teacher. The absolute values are multiplied by $m_i \in \{0, 1\}$. Each of these is computed along the interaction and determines the status in which $\mathbf{s}$ performed worse than $\mathbf{t}$ ($m_i = 1$) or better ($m_i = 0$) in terms of the rewards $r(s_t, a_t)$ and $r(s_t, a_t^{(\mathbf{t})})$. The whole Eq. (3) is similar to what proposed in [43] for KD from bounding-box predictions of object detectors. However, here we provide a temporal formulation of such objective and

[2] Please refer to Appendix A.1 for the definition of $\phi(\cdot)$.

we swap the L2 loss with the L1, which was shown to work better for regression-based trackers [6,7]. By optimizing Eq. (3), the weights θ are changed only if the student's performance is lower than the performance of the teacher. In this way, we make the teacher transferring its knowledge by suggesting actions only in bad performing cases. In the others, we let the student free to follow its current tracking policy since it is superior.

Autonomous Students. The learning process performed by the autonomous students follows the standard RL method for continuous control [68]. Each student interacts with M_j for a maximum of t_{max} steps. At each step t, the students sample actions from a normal distribution $\mathcal{N}(\mu, \sigma)$, where the mean is defined as the student's predicted action, $\mu = \mathbf{s}_\pi(s_t|\theta')$, and the standard deviation is obtained as $\sigma = |\mathbf{s}_\pi(s_t|\theta') - \phi(g_t, b_{t-1})|$ (which is the absolute value of the difference between the student's action and the action that obtains, by shifting b_{t-1}, the ground-truth bounding box g_t). Intuitively, $\mathcal{N}$ shrinks when a_t is close to the ground-truth action $\phi(g_t, b_{t-1})$, reducing the chance of choosing potential wrong actions when approaching the correct one. On the other hand, when a_t is distant from $\phi(g_t, b_{t-1})$, $\mathcal{N}$ spreads letting the student explore more. The students also predict $v_t = \mathbf{s}_v(s_t|\theta)$ which is the cumulative reward that the student expects to receive from s_t to the end of the interaction. Since the proposed reward definition is a direct measure of the IoU occurring between the predicted and the ground-truth bounding boxes, $\mathbf{s}_v(s_t|\theta)$ gives an estimate of the total amount of IoU that $\mathbf{s}$ expects to obtain from state s_t on wards. Thus, this function can be exploited as a future-performance evaluator. After t_{max} steps of interaction, the gradient to update the shared weights θ is built as

$$\nabla_{\theta'}(\mathcal{L}_\pi + \mathcal{L}_v) \tag{5}$$

$$\mathcal{L}_\pi = -\sum_{i=1}^{t_{max}} \log \mathbf{s}_\pi(s_i|\theta')\big(r_i + \gamma \mathbf{s}_v(s_{i+1}|\theta') - \mathbf{s}_v(s_i|\theta')\big) \tag{6}$$

$$\mathcal{L}_v = \sum_{i=1}^{t_{max}} \frac{1}{2}\big(R_i - \mathbf{s}_v(s_i|\theta')\big)^2, R_i = \sum_{k=1}^{i} \gamma^{k-1} r_k \tag{7}$$

where (6) is the policy loss and (7) is the value loss. These definitions follow the standard advantage actor-critic objective [48].

To further facilitate and improve the learning, a curriculum learning strategy [69] is built for each parallel student. During learning, the length of the interaction is increased as $\mathbf{s}$ performs better than $\mathbf{T}$. Details are given in Appendix A.2.

3.4 Student Architecture

The architecture used to maintain the representation of both the policy $\mathbf{s}_\pi$ and the state value $\mathbf{s}_v$ functions, which is pictured in Figure 2, is simple and presents a structure similar to the one proposed in [6,7]. The network gets as input

two image patches that pass through two ResNet-18 based [38] convolutional branches that share weights. The feature maps produced by the branches are first linearized, then concatenated together and finally fed to two consecutive fully connected layers with ReLU activations. After that, features are given to an LSTM [70] layer. Both the fully connected layers and the LSTM are composed of 512 neurons. The output of the LSTM is ultimately fed to two separate fully connected heads, one that outputs the action $a_t = \mathbf{s}_\pi(s_t|\theta)$ and the other that outputs the value of the state $v_t = \mathbf{s}_v(s_t|\theta)$.

3.5 Tracking after Learning

After the learning process, the student $\mathbf{s}(\cdot|\theta)$ is ready to be used for tracking. Here we describe three different ways in which $\mathbf{s}(\cdot|\theta)$ can be exploited:

1. the student's learned policy $\mathbf{s}_\pi(s_t|\theta)$ is used to predict bounding boxes b_t independently from the teachers. We call this setting TRAS (TRAcking Student).
2. the learned policy $\mathbf{s}_\pi(s_t|\theta)$ and value function $\mathbf{s}_v(s_t|\theta)$ are used to, respectively, predict b_t and evaluate $\mathbf{s}$ and $\mathbf{t} \in \mathbf{T}$ tracking behaviors, in order to correct the former's performance. We refer to this setup as TRAST (TRAcking Student and Teacher).
3. the learned state-value function $\mathbf{s}_v(s_t|\theta)$ is used to evaluate the performance of the pool of teachers $\mathbf{T}$ in order to choose the best $b_t^{(\mathbf{t})}$ and perform tracker fusion. We call this setup TRASFUST (TRAcking by Student FUSing Teachers).

In the following, we provide more details about the three settings. For a better understanding, the setups are visualized in Figure 3.

TRAS. In this setting, each tracking sequence $\mathcal{V}_j$, with target object outlined by g_0, is considered as M_j described in section 3.2. States s_t are extracted from frames F_{t-1}, F_t, actions are performed by means of the student's learned policy $a_t = \mathbf{s}_\pi(s_t|\theta)$ and are used to output the bounding boxes $b_t = \psi(a_t, b_{t-1})$. This setup is fast as it requires just a forward pass through the network to obtain a bounding box prediction.

TRAST. In this setup, the student makes use of the learned $\mathbf{s}_\pi(s_t|\theta)$ to predict b_t and $\mathbf{s}_v(s_t|\theta)$ to evaluate its running tracking quality and the one of $\mathbf{t} \in \mathbf{T}$ which is run in parallel. In particular, at each time step t, $v_t = \mathbf{s}_v(s_t|\theta)$ and $v_t^{(\mathbf{t})} = \mathbf{s}_v(s_t^{(\mathbf{t})}|\theta)$ are obtained as performance evaluation for $\mathbf{s}$ and $\mathbf{t}$ respectively. The teacher state is obtained as $s_t^{(\mathbf{t})} = \rho(F_{t-1}, F_t, b_{t-1}^{(\mathbf{t})}, c)$. By comparing the two expected returns, TRAST decides if to output the student's or the teacher's bounding box. More formally, if $v_t \geq v_t^{(\mathbf{t})}$ then $b_t := \psi(a_t, b_{t-1})$ otherwise $b_t := b_t^{(\mathbf{t})}$. This assignment has the side effect of correcting the tracking behaviour of the student as, at the successive time step, the previously known bounding

Table 1: Teacher-based statistics of the transfer set.

Teachers	$\beta = 0.5$			$\beta = 0.6$			$\beta = 0.7$			$\beta = 0.8$			$\beta = 0.9$		
	# traj	AO	$\lvert\mathcal{D}\rvert$	# traj	AO	$\lvert\mathcal{D}\rvert$	# traj	AO	$\lvert\mathcal{D}\rvert$	# traj	AO	$\lvert\mathcal{D}\rvert$	# traj	AO	$\lvert\mathcal{D}\rvert$
$\mathbf{T}_\mathtt{K}$	1884	0.798	9225	1097	0.836	5349	439	0.873	2122	73	0.914	356	0	0.0	0
$\mathbf{T}_\mathtt{M}$	1600	0.767	7859	781	0.808	3831	216	0.851	1052	18	0.898	86	0	0.0	0
$\mathbf{T}_\mathtt{E}$	2754	0.808	13526	1659	0.843	8122	720	0.879	3507	160	0.915	773	1	0.954	4
$\mathbf{T}_\mathtt{S}$	3913	0.829	19259	2646	0.854	12997	1447	0.878	7080	431	0.908	2097	9	0.947	42
$\mathbf{T}_\mathtt{P}$	4519	0.840	22252	3092	0.863	15195	1698	0.887	8307	496	0.915	2414	10	0.948	46

box becomes the previous prediction of the teacher. Thus, the online adaption consists in a very simple procedure that evaluates $\mathbf{t}$'s performance to eventually pass control to it. Notice that, at every t, the execution of $\mathbf{t}$ is independent from $\mathbf{s}$ as the second does not need the first to finish because the evaluations are done based on the predictions given at $t-1$. Hence, the executions of the two can be put in parallel, with the overall speed of TRAST resulting is the lowest between the one of $\mathbf{s}$ and $\mathbf{t}$.

TRASFUST. In this tracking setup, just the student's learned state-value function $\mathbf{s}_v(s_t|\theta)$ is exploited. At each step t, teachers $\mathbf{t} \in \mathbf{T}$ are executed following their standard methodology. States $s_t^{(\mathbf{t})} = \rho(F_{t-1}, F_t, b_{t-1}^{(\mathbf{t})}, c)\ \forall \mathbf{t} \in \mathbf{T}$ are obtained. The performance evaluation of the teachers is obtained through the student as $v_t^{(\mathbf{t})} = \mathbf{s}_v(s_t^{(\mathbf{t})}|\theta)$. The output bounding box is selected as $b_t = b^{(\mathbf{t}')}$ by considering the teacher $\mathbf{t}'$ that achieves the highest expected return, i.e.

$$\mathbf{t}' \in \mathbf{T} \ : \ v_t^{(\mathbf{t}')} = \max_{\mathbf{t} \in \mathbf{T}} v_t^{(\mathbf{t})}. \tag{8}$$

This procedure consists in fusing sequence-wise the predictions of $\mathbf{T}$ and, similarly as for TRAST, the execution of teachers and student can be put in parallel. In such setting, the speed of TRASFUST results in the lowest between the ones of $\mathbf{s}$ and of each $\mathbf{t} \in \mathbf{T}$.

4 Experimental Results

4.1 Experimental Setup

Teachers. The tracking teachers selected for this work are KCF [2], MDNet [13], ECO [15], SiamRPN [9], ATOM [16], and DiMP [17], due to their established popularity in the visual tracking panorama. Moreover, since they tackle visual tracking by different approaches, they can provide knowledge of various quality. In experiments, we considered exploiting single teacher or a pool of teachers. In particular, the following sets of teachers were examined $\mathbf{T}_\mathtt{K} = \{\mathtt{KCF}\}, \mathbf{T}_\mathtt{M} = \{\mathtt{MDNet}\}, \mathbf{T}_\mathtt{E} = \{\mathtt{ECO}\}, \mathbf{T}_\mathtt{S} = \{\mathtt{SiamRPN}\}, \mathbf{T}_\mathtt{A} = \{\mathtt{ATOM}\}, \mathbf{T}_\mathtt{D} = \{\mathtt{DiMP}\}, \mathbf{T}_\mathtt{P} = \{\mathtt{KCF}, \mathtt{MDNet}, \mathtt{ECO}, \mathtt{SiamRPN}\}$.

Transfer Set. The selected transfer set was the training set of GOT-10k dataset [71], due to its large scale. Just $\mathbf{T}_\mathtt{K}, \mathbf{T}_\mathtt{M}, \mathbf{T}_\mathtt{E}, \mathbf{T}_\mathtt{S}$ were used for offline learning, as none of these was trained on this dataset. This is an important point because

unbiased examples of the trackers' behavior should be exploited to train the student. Moreover, predictions that exhibit meaningful knowledge should be retained. Therefore, we filtered out all the videos $\mathcal{V}_j$ which teacher predictions did not satisfy $\text{IoU}(b_t^{(\mathbf{t})}, g_t) > \beta$ for all $t \in \{1, \dots, T_j\}$. We considered $\beta = 0.5$ as minimum threshold for a prediction to be considered positive, and we then varied β among 0.6, 0.7, 0.8, 0.9 for more precise predictions. To produce more training samples, videos, and filtered trajectories were split in five randomly indexed sequences of 32 frames and bounding boxes, similarly as done in [7]. In Table 1 a summary of $\mathcal{D}$ is presented. The number of positive trajectories, the average overlap (A0) [71] on the transfer set, and the total number of sequences $|\mathcal{D}|$ are reported per teacher and per β.

Benchmarks and Performance Measures. We performed performance evaluations on the GOT-10k test set [71], UAV123 [72], LaSOT [73], OTB-100 [74] and VOT2019 [75] datasets. These offer videos of various nature and difficulty, and are all popular benchmarks in the visual tracking community. The evaluation protocol used for GOT-10k is the one-pass evaluation (OPE) [74], along with the metrics: AO, and success rates (SR) with overlap thresholds 0.50 and 0.75. For UAV123, LaSOT, and OTB-100 the OPE method was considered with the area-under-the-curve (AUC) of the success and precision plot, referred to as success score (SS) and precision scores (PS) respectively [74]. Evaluation on VOT2019 is performed in terms of expected average overlap (EAO), accuracy (A), and robustness (R) [76]. Further details about the benchmarks are given in Appendix B.

Implementation Details. The image crops of s_s were resized to $[128 \times 128 \times 3]$ pixels and standardized by the mean and standard deviation calculated on the ImageNet dataset [77]. The ResNet-18 weights were pre-trained for image classification on the same dataset [77]. The image context factor c was set to 1.5. The training videos were processed in chunks of 32 frames. At test time, every 32 frames, the LSTM's hidden state is reset to the one obtained after the first student prediction (i.e. $t = 1$), following [7]. Due to hardware constraints, a maximum of $S = 24$ training students were distributed on 4 NVIDIA TITAN V GPUs of a machine with an Intel Xeon E5-2690 v4 @ 2.60GHz CPU and 320 GB of RAM. The discount factor γ was set to 1. The length of the interaction before an update was defined in $t_{max} = 5$ steps. The Radam optimizer [78] was employed and the learning rate for both distilling and autonomous students was set to 10^{-6}. A weight decay of 10^{-4} was also added to $\mathcal{L}_{\text{dist}}$ as regularization term. To control the magnitude of the gradients and stabilize learning, $(\mathcal{L}_\pi + \mathcal{L}_v)$ was multiplied by 10^{-3}. The student was trained until the validation performance on the GOT-10k validation set stopped improving. Longest trainings took around 10 days. The speed of the parallel setups of TRAST and TRASFUST was computed by considering the speed of the slowest tracker (student or teacher) plus an

Table 2: Performance of the proposed trackers. Results of removing some components of our methodology are also reported. Best values, per contribution, are highlighted in red.

Contribution	GOT-10k			UAV123		LaSOT		OTB-100	
	AO	$SR_{0.50}$	$SR_{0.75}$	SS	PS	SS	PS	SS	PS
TRAS-GT	0.444	0.495	0.286	0.483	0.616	0.331	0.271	0.438	0.581
TRAS-KD-GT	0.448	0.499	0.305	0.491	0.630	0.354	0.298	0.448	0.606
TRAS-KD	0.422	0.481	0.239	0.494	0.634	0.340	0.276	0.457	0.635
TRAS-no-curr	0.474	0.547	0.307	0.501	0.644	0.385	0.323	0.447	0.600
TRAS	0.484	0.556	0.326	0.515	0.655	0.386	0.330	0.481	0.644
TRAST-no-curr	0.530	**0.630**	**0.347**	0.602	0.770	0.484	0.464	0.595	0.794
TRAST	**0.531**	0.626	0.345	0.603	0.773	0.490	0.470	0.604	0.818
TRASFUST-no-curr	0.506	0.599	0.278	0.627	0.819	0.496	0.484	**0.665**	0.879
TRASFUST	0.519	0.616	0.287	**0.628**	**0.823**	**0.510**	**0.505**	0.660	**0.890**

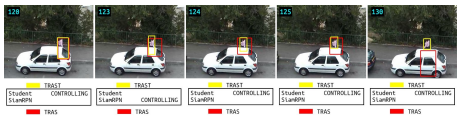

Fig. 4: Visual example of how TRAST relies effectively on the teacher, passing control to $\mathbf{T_S}$ and saving the simple student (TRAS) from the drift.

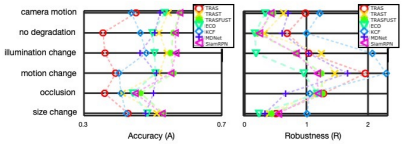

Fig. 5: Analysis of the accuracy (A) and robustness (R) on VOT2019 over different classes of tracking sequences.

overhead. Code was implemented in Python and is available here[3]. Source code publicly available was used to implement the teacher trackers. Default configurations were respected as much as possible. For a fair comparison, we report the results of such implementations, that have slightly different performance than stated in the original papers.

4.2 Results

In the following sections, when not specified, the three tracker setups regard the student trained using $\mathbf{T_P}$ and $\beta = 0.5$, paired with $\mathbf{T_S}$ in TRAST, and managing $\mathbf{T_P}$ in TRASFUST.

General Remarks. In Table 2 the performance of TRAS, TRAST, TRASFUST are reported, while the performances of the teachers are presented in the first six rows of Table 5. TRAS results in a very fast method with good accuracy. Combining KD and RL results in the best performance, outperforming the baselines that use for training just the ground-truth (TRAS-GT), KD and ground-truth (TRAS-KD-GT), and just KD (TRAS-KD). We did not report the performance of $\mathbf{s}$ trained only by RL because convergence was not attained due to the large state and action spaces. Benefiting the teacher during tracking is an effective online procedure. Indeed, TRAST improves TRAS by 24% on average, and a qualitative example of the ability to pass control to the teacher is given in Figure 4. The performance of TRASFUST confirms the student's evaluation ability. This is the most accurate and robust tracker thanks to the effective fusion of

[3] `https://github.com/dontfollowmeimcrazy/vot-kd-rl`

Table 3: Performance of the proposed trackers while considering different teacher setups for training and tracking. Best results per tracker are highlighted in red, second-best in blue.

	Training Teachers	Tracking Teachers	GOT-10k AO	GOT-10k $SR_{0.50}$	GOT-10k $SR_{0.75}$	UAV123 SS	UAV123 PS	LaSOT SS	LaSOT PS	OTB-100 SS	OTB-100 PS	FPS
TRAS	$\mathbf{T}_K$	-	0.371	0.418	0.178	0.464	0.598	0.321	0.241	0.390	0.524	90
	$\mathbf{T}_M$	-	0.414	0.473	0.214	0.462	0.606	0.336	0.262	0.390	0.545	
	$\mathbf{T}_E$	-	0.422	0.484	0.232	0.507	**0.652**	0.357	0.286	0.422	0.567	
	$\mathbf{T}_S$	-	**0.441**	**0.499**	**0.290**	**0.517**	0.646	**0.377**	**0.310**	**0.447**	**0.599**	
	$\mathbf{T}_P$	-	**0.484**	**0.556**	**0.326**	**0.515**	**0.655**	**0.386**	**0.330**	**0.481**	**0.644**	
TRAST	$\mathbf{T}_K$	$\mathbf{T}_K$	0.390	0.440	0.191	0.526	0.682	0.388	0.319	0.495	0.660	90
	$\mathbf{T}_M$	$\mathbf{T}_M$	0.452	0.521	0.223	0.572	0.776	0.433	0.386	0.569	0.793	5
	$\mathbf{T}_E$	$\mathbf{T}_E$	0.491	0.571	0.249	0.580	0.768	0.442	0.397	0.583	0.786	15
	$\mathbf{T}_S$	$\mathbf{T}_S$	0.532	0.632	0.354	0.605	0.779	0.485	0.457	0.601	0.806	40
	$\mathbf{T}_P$	$\mathbf{T}_K$	0.469	0.541	0.297	0.562	0.727	0.422	0.376	0.560	0.760	90
	$\mathbf{T}_P$	$\mathbf{T}_M$	0.494	0.573	0.302	0.604	0.798	0.466	0.431	0.596	0.815	5
	$\mathbf{T}_P$	$\mathbf{T}_E$	0.521	0.607	0.307	0.606	0.795	0.456	0.419	0.608	0.822	15
	$\mathbf{T}_P$	$\mathbf{T}_S$	0.531	0.626	0.345	0.603	0.773	0.490	0.470	0.604	0.818	40
	$\mathbf{T}_P$	$\mathbf{T}_A$	**0.557**	**0.640**	**0.393**	**0.634**	**0.823**	**0.513**	**0.488**	**0.623**	**0.838**	20
	$\mathbf{T}_P$	$\mathbf{T}_D$	**0.604**	**0.708**	**0.469**	**0.647**	**0.837**	**0.545**	**0.524**	**0.643**	**0.865**	25
TRASFUST	$\mathbf{T}_P$	$\mathbf{T}_K \cup \mathbf{T}_M$	0.317	0.319	0.105	0.493	0.720	0.396	0.372	0.666	**0.901**	5
	$\mathbf{T}_P$	$\mathbf{T}_M \cup \mathbf{T}_E$	0.384	0.398	0.131	0.563	0.791	0.422	0.392	**0.701**	**0.931**	5
	$\mathbf{T}_P$	$\mathbf{T}_E \cup \mathbf{T}_S$	**0.526**	**0.624**	**0.305**	**0.634**	0.815	0.507	0.500	0.670	0.877	15
	$\mathbf{T}_P$	$\mathbf{T}_A \cup \mathbf{T}_D$	**0.617**	**0.729**	**0.490**	**0.679**	**0.873**	**0.576**	**0.574**	**0.692**	0.895	20
	$\mathbf{T}_P$	$\mathbf{T}_M \cup \mathbf{T}_E \cup \mathbf{T}_S$	0.517	0.615	0.294	0.633	**0.823**	**0.513**	0.504	0.682	0.897	5
	$\mathbf{T}_P$	$\mathbf{T}_P$	0.519	0.616	0.287	0.628	**0.823**	0.510	**0.505**	0.660	0.890	5

the underlined trackers. Overall, all three trackers show balanced performance across the benchmarks, thus demonstrating good generalization. No use of the curriculum learning strategy (TRAS-no-curr, TRAST-no-curr, TRASFUST-no-curr) slightly decreases the performance of all. In Figure 5 the performance of the trackers is reported for different classes of sequences of VOT2019, while in Figure 7 some qualitative examples are presented. [4] These results demonstrate the effectiveness of our methodology and that the proposed student model respects, respectively, the goals (i), (ii), (iii) introduced in Section 1.

Impact Of Teachers. In Table 3 the performance of the proposed trackers in different student-teacher setups is reported. The general trend of the three trackers reflects the increasing tracking capabilities of the teachers. Indeed, on every considered benchmark, the tracking ability of the student increases as a stronger teacher is employed. For TRAST, this is also proven by Figure 6 (a), where we show that better teachers are exploited more. Moreover, using more than one teacher during training leads to superior tracking policies and to better exploitation of them during tracking. Although in general student models cannot outperform their teachers due to their simple and compressed architecture [79], TRAS and TRAST show such behavior on benchmarks where teachers are weak. Using two teachers during tracking is the best TRASFUST configuration, as in this setup it outperforms the best teacher by more than 2% on all the considered benchmarks. When weaker teachers are added to the pool, the performance tends to decrease, suggesting a behavior similar to the one pointed out in [21]. Part of the error committed by TRAST and TRASFUST on benchmarks like OTB-100 and VOT2019 is explained by Figure 8. In situations of ambiguous ground-truth, such trackers make predictions that are qualitatively better but quantitatively worse. TRAST and TRASFUST show to be unbiased to the training teachers, as their capabilities generalize also to $\mathbf{T}_A$ and $\mathbf{T}_D$ which are not exploited during training. In Table 4 we present the performance of the proposed trackers while

[4] For more, please see `https://youtu.be/uKtQgPk3nCU`

Table 4: Results of the proposed trackers considering $\mathbf{T_P}$'s increasingly better predictions. Best values, per tracker, are highlighted in red, second-best in blue.

	Tracker	GOT-10k AO	GOT-10k $SR_{0.50}$	GOT-10k $SR_{0.75}$	UAV123 SS	UAV123 PS	LaSOT SS	LaSOT PS	OTB-100 SS	OTB-100 PS
$\beta = 0.5$	TRAS	**0.484**	**0.556**	**0.326**	**0.515**	**0.655**	**0.386**	**0.330**	**0.481**	**0.644**
	TRAST	**0.532**	**0.632**	**0.354**	**0.605**	**0.779**	**0.485**	**0.457**	**0.601**	**0.806**
	TRASFUST	**0.519**	**0.616**	0.287	0.628	0.823	0.510	**0.505**	0.660	0.890
$\beta = 0.6$	TRAS	**0.426**	**0.488**	**0.244**	**0.481**	**0.609**	**0.343**	**0.277**	**0.452**	**0.617**
	TRAST	**0.518**	**0.616**	**0.326**	**0.599**	**0.768**	0.475	0.452	**0.608**	**0.809**
	TRASFUST	**0.507**	**0.599**	**0.295**	**0.639**	**0.827**	**0.514**	**0.510**	**0.683**	**0.901**
$\beta = 0.7$	TRAS	0.404	0.449	0.231	0.430	0.552	0.334	0.260	0.390	0.522
	TRAST	0.513	0.603	0.310	0.594	0.766	**0.478**	**0.456**	0.586	0.781
	TRASFUST	**0.507**	**0.599**	**0.289**	**0.638**	**0.827**	**0.513**	**0.505**	**0.675**	**0.894**
$\beta = 0.8$	TRAS	0.326	0.344	0.155	0.387	0.489	0.243	0.170	0.323	0.414
	TRAST	0.505	0.598	0.297	0.592	0.764	0.457	0.426	0.589	0.774
	TRASFUST	0.494	0.575	0.260	0.624	0.815	0.494	0.482	0.672	0.888
$\beta = 0.9$	TRAS	0.140	0.070	0.014	0.064	0.045	0.086	0.019	0.132	0.104
	TRAST	0.471	0.541	0.250	0.547	0.697	0.445	0.409	0.574	0.746
	TRASFUST	0.403	0.425	0.169	0.534	0.743	0.401	0.374	0.626	0.836

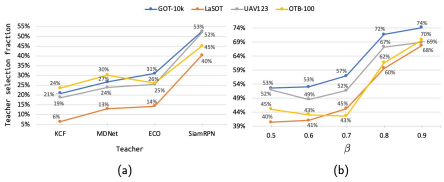

Fig. 6: Per benchmark fractions of predictions attributed to $\mathbf{t}$ in the TRAST setup.

Fig. 7: Qualitative examples of the proposed trackers.

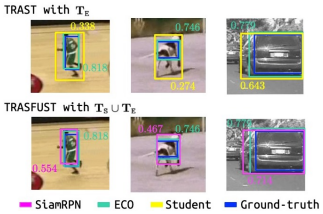

Fig. 8: Behaviour of TRAST and TRASFUST with ambiguous ground-truths. In the presented frames, TRAST selects the bounding box predicted by the student, while TRASFUST to one given by $\mathbf{T_S}$. Those outputs are qualitative better but have much less IoU (quantified by the colored numbers) with respect to g_t. This impacts the overall quantitative performance.

considering different quality of teacher actions. Increasing the quality, thus reducing the number of videos, results in decreasing the performance of all three trackers. The loss is not significant between $\beta = 0.5$ and $\beta = 0.7$, while considering more precise actions, TRAS suffers majorly, suggesting that more data is a key factor for an autonomous tracking policy. Interestingly, TRAST and TRASFUST are able to perform tracking even if the student is trained with limited training samples. The plot (b) in Figure 6 confirms that the student relies effectively to its teacher, as the latter's output is selected more often as $\mathbf{s}$ loses performance.

Running the student takes just 11ms on our machine. TRAS performs at 90 FPS. The speed of TRAST and TRASFUST depends on the chosen teacher and varies between 5 and 40 FPS, as shown in Table 3. In parallel setups, TRAST and TRASFUST run in real-time if the teachers do so.

Table 5: Performance of the proposed trackers (in the last block of rows) in comparison with the the state-of-the-art. First block of rows reports the performance of the selected teachers; second block shows generic-approach tracker performance; third presents trackers that exploit experts or perform fusion. Best results are highlighted in red, second-best in blue.

Tracker	GOT-10k			UAV123		LaSOT		OTB-100		VOT2019			FPS
	AO	$SR_{0.50}$	$SR_{0.75}$	SS	PS	SS	PS	SS	PS	EAO	A	R	
KCF [2]	0.203	0.177	0.065	0.331	0.503	0.178	0.166	0.477	0.693	0.110	0.441	1.279	105
MDNet [13]	0.299	0.303	0.099	0.489	0.718	0.397	0.373	0.673	0.909	0.151	0.507	0.782	5
ECO [15]	0.316	0.309	0.111	0.532	0.726	0.324	0.301	0.668	0.896	0.262	0.505	0.441	15
SiamRPN [9]	0.508	0.604	0.308	0.616	0.785	0.508	0.492	0.649	0.851	0.259	0.554	0.572	43
ATOM [16]	0.556	0.634	0.402	0.643	0.832	0.516	0.506	0.660	0.867	**0.292**	**0.603**	**0.411**	20
DiMP [17]	**0.611**	**0.717**	**0.492**	**0.653**	**0.839**	**0.570**	**0.569**	0.681	0.888	**0.379**	0.594	**0.278**	25
GOTURN [6]	0.347	0.375	0.124	0.389	0.548	0.214	0.175	0.395	0.534	-	-	-	100
RE3 [7]	-	-	-	0.514	0.667	0.325	0.301	0.464	0.582	0.152	0.458	0.940	150
ADNet [28]	-	-	-	-	-	-	-	0.646	0.880	-	-	-	3
ACT [32]	-	-	-	0.415	0.636	-	-	0.625	0.859	-	-	-	30
DRL-IS [31]	-	-	-	-	-	-	-	0.671	0.909	-	-	-	10
SiamRPN++ [10]	-	-	-	0.613	0.807	0.496	-	**0.696**	**0.914**	0.285	**0.599**	0.482	35
GCT [80]	-	-	-	0.508	0.732	-	-	0.648	0.854	-	-	-	50
GradNet [81]	-	-	-	-	-	0.365	0.351	0.639	0.861	-	-	-	80
SiamCAR [82]	0.569	0.670	0.415	0.614	0.760	0.507	0.510	-	-	-	-	-	52
ROAM [83]	0.436	0.466	0.164	-	-	0.368	0.390	0.681	0.908	-	-	-	13
MEEM [18]	0.253	0.235	0.068	0.392	0.627	0.280	0.224	0.566	0.830	-	-	-	10
HMMTxD [22]	-	-	-	-	-	-	-	-	-	0.163	0.499	1.073	-
HDT [52]	-	-	-	-	-	-	-	0.562	0.844	-	-	-	10
Zhu et al. [84]	-	-	-	-	-	-	-	0.587	0.788	-	-	-	36
Li et al. [53]	-	-	-	-	-	-	-	0.621	0.864	-	-	-	6
TRAS	0.484	0.556	0.326	0.515	0.655	0.386	0.330	0.481	0.644	0.131	0.400	1,020	90
TRAST	0.604	0.708	0.469	0.647	0.837	0.545	0.524	0.643	0.865	0.203	0.517	0.693	25
TRASFUST	**0.617**	**0.729**	**0.490**	**0.679**	**0.873**	**0.576**	**0.574**	**0.701**	**0.931**	0.266	0.592	0.597	20

State of the Art Comparison. In Table 5 we report the results of the proposed trackers against the state-of-the-art. In the following comparisons, we consider the results of the best configurations proposed in the above analysis.

TRAS outperforms GOTURN and RE3 which employ a similar DNN architecture but different learning strategies. On GOT-10k and LaSOT it also surpasses the recent GradNet and ROAM, and GCT on UAV123. TRAST outperforms ATOM and SiamCAR on GOT-10k, UAV123, LaSOT, while losing little performance to DiMP. The performance is better than RL-based trackers [28,32,31] on UAV123 and comparable on OTB-100. Finally, TRASFUST outperforms all the trackers on all the benchmarks (where the pool $\mathbf{T_E} \cup \mathbf{T_S}$ was used). Remarkable results are obtained on UAV123 and OTB-100, with SS of 0.679 and 0.701 and PS of 0.873 and 0.931, respectively. Large improvement is achieved over all the methodologies that include expert trackers in their methodology.

5 Conclusions

In this paper, a novel methodology for visual tracking is proposed. KD and RL are joined in a novel framework where off-the-shelf tracking algorithms are employed to compress knowledge into a CNN-based student model. After learning, the student can be exploited in three different tracking setups, TRAS, TRAST and TRASFUST, depending on application needs. An extensive validation shows that the proposed trackers TRAS and TRAST compete with the state-of-the-art, while TRASFUST outperforms recently published methods and fusion approaches. All trackers can run in real-time.

Acknowledgements. This work is supported by the ACHIEVE-ITN project.

References

1. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2010) 2544–2550
2. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence **37** (2015) 583–596
3. Danelljan, M., Hager, G., Khan, F.S., Felsberg, M.: Discriminative Scale Space Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **39** (2017) 1561–1575
4. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.H.: Staple: Complementary learners for real-time tracking. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 2016-Decem. (2016) 1401–1409
5. Lukežič, A., Vojíř, T., Čehovin Zajc, L., Matas, J., Kristan, M.: Discriminative Correlation Filter Tracker with Channel and Spatial Reliability. International Journal of Computer Vision **126** (2018) 671–688
6. Held, D., Thrun, S., Savarese, S.: Learning to Track at 100 FPS with Deep Regression Networks. In: European Conference on Computer Vision. Volume abs/1604.0. (2016)
7. Gordon, D., Farhadi, A., Fox, D.: Re 3 : Real-time recurrent regression networks for visual tracking of generic objects. IEEE Robotics and Automation Letters **3** (2018) 788–795
8. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. European Conference on Computer Vision **9914 LNCS** (2016) 850–865
9. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High Performance Visual Tracking with Siamese Region Proposal Network. In: IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2018) 8971–8980
10. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: SIAMRPN++: Evolution of siamese visual tracking with very deep networks. IEEE Conference on Computer Vision and Pattern Recognition **2019-June** (2019) 4277–4286
11. Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware siamese networks for visual object tracking. In: European Conference on Computer Vision. Volume 11213 LNCS. (2018) 103–119
12. Zhang, Z., Peng, H.: Deeper and Wider Siamese Networks for Real-Time Visual Tracking. IEEE Conference on Computer Vision and Pattern Recognition (2019)
13. Nam, H., Han, B.: Learning Multi-domain Convolutional Neural Networks for Visual Tracking. IEEE Conference on Computer Vision and Pattern Recognition **2016-Decem** (2016) 4293–4302
14. Jung, I., Son, J., Baek, M., Han, B.: Real-Time MDNet. In: European Conference on Computer Vision. (2018)
15. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: ECO: Efficient Convolution Operators for Tracking. In: IEEE Conference on Computer Vision and Pattern Recognition. (2017)
16. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: ATOM: Accurate Tracking by Overlap Maximization. In: IEEE Conference on Computer Vision and Pattern Recognition. (2019)
17. Bhat, G., Danelljan, M., Van Gool, L., Timofte, R.: Learning Discriminative Model Prediction for Tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019)

18. Zhang, J., Ma, S., Sclaroff, S.: MEEM: Robust tracking via multiple experts using entropy minimization. In: European Conference on Computer Vision. Volume 8694 LNCS., Springer Verlag (2014) 188–203
19. Yoon, J.H., Kim, D.Y., Yoon, K.J.: Visual tracking via adaptive tracker selection with multiple features. In: European Conference on Computer Vision. Volume 7575 LNCS. (2012) 28–41
20. Wang, N., Yeung, D.Y.: Ensemble-based tracking: Aggregating crowdsourced structured time series data. In: 31st International Conference on Machine Learning, ICML 2014. Volume 4. (2014) 2807–2817
21. Bailer, C., Pagani, A., Stricker, D.: A superior tracking approach: Building a strong tracker through fusion. In: European Conference on Computer Vision. Volume 8695 LNCS., Springer Verlag (2014) 170–185
22. Vojir, T., Matas, J., Noskova, J.: Online adaptive hidden Markov model for multi-tracker fusion. Computer Vision and Image Understanding **153** (2016) 109–119
23. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. IEEE Conference on Computer Vision and Pattern Recognition **2** (2000) 142–149
24. Maresca, M.E., Petrosino, A.: MATRIOSKA: A multi-level approach to fast tracking by learning. In: International Conference on Image Analysis and Processing. Volume 8157 LNCS. (2013) 419–428
25. Čehovin, L., Kristan, M., Leonardis, A.: Robust visual tracking using an adaptive coupled-layer visual model. IEEE Transactions on Pattern Analysis and Machine Intelligence **35** (2013) 941–953
26. Nam, H., Hong, S., Han, B.: Online graph-based tracking. In: European Conference on Computer Vision. Volume 8693 LNCS., Springer Verlag (2014) 112–126
27. Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.M., Hicks, S.L., Torr, P.H.: Struck: Structured Output Tracking with Kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence **38** (2016) 2096–2109
28. Yun, S., Choi, J., Yoo, Y., Yun, K., Choi, J.Y.: Action-decision networks for visual tracking with deep reinforcement learning. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 2017-Janua., IEEE (2017) 1349–1358
29. Supancic, J., Ramanan, D.: Tracking as Online Decision-Making: Learning a Policy from Streaming Videos with Reinforcement Learning. Proceedings of the IEEE International Conference on Computer Vision **2017-Octob** (2017) 322–331
30. Choi, J., Kwon, J., Lee, K.M.: Real-time visual tracking by deep reinforced decision making. Computer Vision and Image Understanding **171** (2018) 10–19
31. Ren, L., Yuan, X., Lu, J., Yang, M., Zhou, J.: Deep Reinforcement Learning with Iterative Shift for Visual Tracking. In: European Conference on Computer Vision. (2018) 684–700
32. Chen, B., Wang, D., Li, P., Wang, S., Lu, H.: Real-time 'Actor-Critic' Tracking. In: European Conference on Computer Vision. (2018) 318–334
33. Dunnhofer, M., Martinel, N., Foresti, G.L., Micheloni, C.: Visual Tracking by means of Deep Reinforcement Learning and an Expert Demonstrator. In: Proceedings of The IEEE/CVF International Conference on Computer Vision Workshops. (2019)
34. Danelljan, M., Robinson, A., Khan, F.S., Felsberg, M.: Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In: European Conference on Computer Vision. (2016)
35. Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.S.: Fast Online Object Tracking and Segmentation: A Unifying Approach. In: IEEE Conference on Computer Vision and Pattern Recognition. (2019)

36. Dunnhofer, M., Antico, M., Sasazawa, F., Takeda, Y., Camps, S., Martinel, N., Micheloni, C., Carneiro, G., Fontanarosa, D.: Siam-U-Net: encoder-decoder siamese network for knee cartilage tracking in ultrasound images. Medical Image Analysis (2020)
37. Hinton, G., Vinyals, O., Dean, J.: Distilling the Knowledge in a Neural Network. In: Deep Learning Workshop NIPS 2014. (2014)
38. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 2016-Decem. (2016) 770–778
39. Tang, Z., Wang, D., Zhang, Z.: Recurrent neural network training with dark knowledge transfer. In: IEEE International Conference on Acoustics, Speech and Signal Processing. Volume 2016-May. (2016) 5900–5904
40. Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., Li, L.J.: Learning from Noisy Labels with Distillation. In: Proceedings of the IEEE International Conference on Computer Vision. Volume 2017-Octob. (2017) 1928–1936
41. Phuong, M., Lampert, C.H.: Distillation-Based Training for Multi-Exit Architectures. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019)
42. Geras, K.J., Mohamed, A.r., Caruana, R., Urban, G., Wang, S., Aslan, O., Philipose, M., Richardson, M., Sutton, C.: Blending LSTMs into CNNs. (2015)
43. Chen, G., Choi, W., Yu, X., Han, T., Chandraker, M.: Learning efficient object detection models with knowledge distillation. In: Advances in Neural Information Processing Systems. Volume 2017-Decem. (2017) 743–752
44. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. (2017)
45. Polino, A., Pascanu, R., Alistarh, D.: Model compression via distillation and quantization. In: International Conference on Learning Representations, International Conference on Learning Representations, ICLR (2018)
46. Watkins, C.J.C.H., Dayan, P.: Q-learning. Machine Learning **8** (1992) 279–292
47. Konda, V.R., Tsitsiklis, J.N.: Actor-Critic Algorithms. In: Advances in Neural Information Processing Systems. (2000)
48. Sutton, R.S., McAllester, D., Singh, S., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. In: Advances in Neural Information Processing Systems. (2000) 1057–1063
49. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing Atari with Deep Reinforcement Learning. CoRR **abs/1312.5** (2013)
50. Mnih, V., Badia, A.P., Mirza, L., Graves, A., Harley, T., Lillicrap, T.P., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. 33rd International Conference on Machine Learning, ICML 2016 **4** (2016) 2850–2869
51. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **34** (2012) 1409–1422
52. Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., Yang, M.H.: Hedged Deep Tracking. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 2016-Decem. (2016) 4303–4311
53. Li, Z., Wei, W., Zhang, T., Wang, M., Hou, S., Peng, X.: Online Multi-Expert Learning for Visual Tracking. IEEE Transactions on Image Processing **29** (2019) 934–946

54. Bucilă, C., Caruana, R., Niculescu-Mizil, A.: Model compression. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Volume 2006. (2006) 535–541
55. Rusu, A.A., Colmenarejo, S.G., Gülçehre, Ç., Desjardins, G., Kirkpatrick, J., Pascanu, R., Mnih, V., Kavukcuoglu, K., Hadsell, R.: Policy distillation. In: 4th International Conference on Learning Representations, ICLR 2016. (2016)
56. Parisotto, E., Ba, J., Salakhutdinov, R.: Actor-mimic deep multitask and transfer reinforcement learning. In: 4th International Conference on Learning Representations, ICLR 2016, International Conference on Learning Representations, ICLR (2016)
57. Garcia, N.C., Morerio, P., Murino, V.: Modality distillation with multiple stream networks for action recognition. In: European Conference on Computer Vision. Volume 11212 LNCS. (2018) 106–121
58. Wang, X., Hu, J.F., Lai, J., Zhang, J., Zheng, W.S.: Progressive Teacher-student Learning for Early Action Prediction. Computer Vision and Pattern Recognition (CVPR) (2019) 3556–3565
59. Shmelkov, K., Schmid, C., Alahari, K.: Incremental Learning of Object Detectors without Catastrophic Forgetting. In: Proceedings of the IEEE International Conference on Computer Vision. Volume 2017-Octob. (2017) 3420–3429
60. Liu, Y., Chen, K., Liu, C., Qin, Z., Luo, Z., Wang, J.: Structured Knowledge Distillation for Semantic Segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. (2019) 2599–2608
61. He, T., Shen, C., Tian, Z., Gong, D., Sun, C., Yan, Y.: Knowledge Adaptation for Efficient Semantic Segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. (2019) 578–587
62. Wu, A., Zheng, W.S., Guo, X., Lai, J.H.: Distilled Person Re-identification:Towards a More Scalable System. IEEE Conference on Computer Vision and Pattern Recognition (2019) 1187–1196
63. Wang, N., Zhou, W., Song, Y., Ma, C., Li, H.: Real-Time Correlation Tracking Via Joint Model Compression and Transfer. IEEE Transactions on Image Processing **29** (2020) 6123–6135
64. Liu, Y., Dong, X., Lu, X., Khan, F.S., Shen, J., Hoi, S.: Teacher-Students Knowledge Distillation for Siamese Trackers. (2019)
65. Meshgi, K., Mirzaei, M.S., Oba, S.: Long and short memory balancing in visual co-tracking using q-learning. In: 2019 IEEE International Conference on Image Processing (ICIP). (2019) 3970–3974
66. Nair, A., Srinivasan, P., Blackwell, S., Alcicek, C., Fearon, R., De Maria, A., Panneershelvam, V., Suleyman, M., Beattie, C., Petersen, S., Legg, S., Mnih, V., Kavukcuoglu, K., Silver, D.: Massively Parallel Methods for Deep Reinforcement Learning. (2015)
67. Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Yotam, B., Vlad, F., Tim, H., Dunning, I., Legg, S., Kavukcuoglu, K.: IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures. In: 35th International Conference on Machine Learning, ICML 2018. Volume 4. (2018) 2263–2284
68. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. 2nd edn. MIT Press, Cambridge, MA, USA (2018)
69. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: 26th International Conference on Machine Learning, ICML '09, New York, New York, USA, ACM Press (2009) 1–8

70. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation **9** (1997) 1735–1780
71. Huang, L., Zhao, X., Huang, K.: GOT-10k: A Large High-Diversity Benchmark for Generic Object Tracking in the Wild. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019) 1–1
72. Mueller, M., Smith, N., Ghanem, B.: A Benchmark and Simulator for UAV Tracking. In: European Conference on Computer Vision, Springer, Cham (2016) 445–461
73. Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H.: LaSOT: A High-quality Benchmark for Large-scale Single Object Tracking. In: IEEE Conference on Computer Vision and Pattern Recognition. (2019)
74. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society (2013) 2411–2418
75. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Pflugfelder, R., Kämäräinen, J.K., Zajc, L., Drbohlav, O., Lukežič, A., Berg, A., Eldesokey, A., Käpylä, J., Fernández, G., Gonzalez-Garcia, A., Memarmoghadam, A., Lu, A., He, A., Varfolomieiev, A., Chan, A., Shekhar Tripathi, A., Smeulders, A., Suraj Pedasingu, B., Xin Chen, B., Zhang, B., Wu, B., Li, B., He, B., Yan, B., Bai, B., Li, B., Li, B., Hak Kim, B., Ma, C., Fang, C., Qian, C., Chen, C., Li, C., Zhang, C., Tsai, C.Y., Luo, C., Micheloni, C., Zhang, C., Tao, D., Gupta, D., Song, D., Wang, D., Gavves, E., Yi, E., Shahbaz Khan, F., Zhang, F., Wang, F., Zhao, F., De Ath, G., Bhat, G., Chen, G., Wang, G., Li, G., Cevikalp, H., Du, H., Zhao, H., Saribas, H., Min Jung, H., Bai, H., Yu, H., Peng, H., Lu, H., Li, H., Li, J., Li, J., Fu, J., Chen, J., Gao, J., Zhao, J., Tang, J., Li, J., Wu, J., Liu, J., Wang, J., Qi, J., Zhang, J., Tsotsos, J.K., Hyuk Lee, J., van de Weijer, J., Kittler, J., Ha Lee, J., Zhuang, J., Zhang, K., Wang, K., Dai, K., Chen, L., Liu, L., Guo, L., Zhang, L., Wang, L., Wang, L., Zhang, L., Wang, L., Zhou, L., Zheng, L., Rout, L., Van Gool, L., Bertinetto, L., Danelljan, M., Dunnhofer, M., Ni, M., Young Kim, M., Tang, M., Yang, M.H., Paluru, N., Martinel, N., Xu, P., Zhang, P., Zheng, P., Zhang, P., Torr, P.H., Zhang Qiang Wang, Q., Guo, Q., Timofte, R., Krishna Gorthi, R., Everson, R., Han, R., Zhang, R., You, S., Zhao, S.C., Zhao, S., Li, S., Li, S., Ge, S., Bai, S., Guan, S., Xing, T., Xu, T., Yang, T., Zhang, T., Ojí, T., Feng, W., Hu, W., Wang, W., Tang, W., Zeng, W., Liu, W., Chen, X., Qiu, X., Bai, X., Wu, X.J., Yang, X., Chen, X., Li, X., Sun, X., Chen, X., Tian, X., Tang, X., Zhu, X.F., Huang, Y., Chen, Y., Lian, Y., Gu, Y., Liu, Y., Chen, Y., Zhang, Y., Xu, Y., Wang, Y., Li, Y., Zhou, Y., Dong, Y., Xu, Y., Zhang, Y., Li, Y., Wang Zhao Luo, Z., Zhang, Z., Feng, Z.H., He, Z., Song, Z., Chen, Z., Zhang, Z., Wu, Z., Xiong, Z., Huang, Z., Teng, Z., Ni, Z.: The Seventh Visual Object Tracking VOT2019 Challenge Results. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. (2019)
76. Kristan, M., Matas, J., Leonardis, A., Vojir, T., Pflugfelder, R., Fernandez, G., Nebehay, G., Porikli, F., Čehovin, L.: A Novel Performance Evaluation Methodology for Single-Target Trackers. IEEE Transactions on Pattern Analysis and Machine Intelligence **38** (2016) 2137–2155
77. Deng, J., Dong, W., Socher, R., Li, L.J., Kai Li, Li Fei-Fei: ImageNet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2009) 248–255
78. Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., Han, J.: On the Variance of the Adaptive Learning Rate and Beyond. (2019)

79. Cho, J.H., Hariharan, B.: On the efficacy of knowledge distillation. In: Proceedings of the IEEE International Conference on Computer Vision. Volume 2019-Octob., Institute of Electrical and Electronics Engineers Inc. (2019) 4793–4801
80. Gao, J., Zhang, T., Xu, C.: Graph Convolutional Tracking. In: IEEE Conference on Computer Vision and Pattern Recognition. Number 1 (2019) 4649–4659
81. Li, P., Chen, B., Ouyang, W., Wang, D., Yang, X., Lu, H.: GradNet: Gradient-Guided Network for Visual Object Tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019)
82. Guo, D., Wang, J., Cui, Y., Wang, Z., Chen, S.: SiamCAR: Siamese Fully Convolutional Classification and Regression for Visual Tracking. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020)
83. Yang, T., Xu, P., Hu, R., Chai, H., Chan, A.B.: ROAM: Recurrently Optimizing Tracking Model. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020)
84. Zhu, Y., Wen, J., Zhang, L., Wang, Y.: Visual Tracking with Dynamic Model Update and Results Fusion. In: Proceedings - International Conference on Image Processing, IEEE Computer Society (2018) 2685–2689