Fast Template Matching and Update for Video Object Tracking and Segmentation

Mingjie Sun[1,2], Jimin Xiao[1,*], Eng Gee Lim[1], Bingfeng Zhang[1,2], Yao Zhao[3]
[1]XJTLU, [2]University of Liverpool, [3]Beijing Jiaotong University
mingjie.sun@liverpool.ac.uk, {jimin.xiao, enggee.lim, bingfeng.zhang}@xjtlu.edu.cn, yzhao@bjtu.edu.cn

Abstract

In this paper, the main task we aim to tackle is the multi-instance semi-supervised video object segmentation across a sequence of frames where only the first-frame box-level ground-truth is provided. Detection-based algorithms are widely adopted to handle this task, and the challenges lie in the selection of the matching method to predict the result as well as to decide whether to update the target template using the newly predicted result. The existing methods, however, make these selections in a rough and inflexible way, compromising their performance. To overcome this limitation, we propose a novel approach which utilizes reinforcement learning to make these two decisions at the same time. Specifically, the reinforcement learning agent learns to decide whether to update the target template according to the quality of the predicted result. The choice of the matching method will be determined at the same time, based on the action history of the reinforcement learning agent. Experiments show that our method is almost 10 times faster than the previous state-of-the-art method with even higher accuracy (region similarity of 69.1% on DAVIS 2017 dataset).

1. Introduction

Multi-instance semi-supervised video object segmentation (VOS) is an important computer vision task, serving as the basis of many other related tasks including scene understand, video surveillance and video editing. The task of VOS is to produce instance segmentation masks for each frame in a video sequence where the first-frame ground-truth is provided in advance. It turns out to be a challenging task especially in the situations of deformation, motion blur, illumination change, background clutter, and so on.

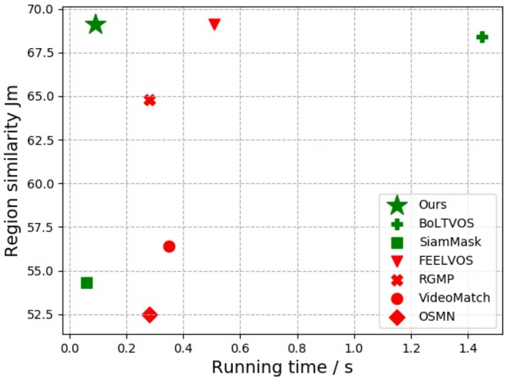

Figure 1. Speed-accuracy trade-off for various multi-instance semi-supervised VOS methods on DAVIS 2017 dataset. Methods in green only rely on the first-frame box-level ground-truth while methods in red rely on the first-frame pixel-level ground-truth.

There are two common ways to provide the first-frame ground-truth, including masks and bounding boxes. Providing the first-frame masks is a conventional way, which has been widely adopted nowadays [32, 27]. Although these methods have already achieved good performance with the pixel-level accurate target object information, it turns to a hard task to utilize these methods to solve practical VOS problems, especially when numerous video sequences need to be processed in a short time, because annotating pixel-level ground-truth masks for each video sequence is time-consuming. To overcome this problem, inspired by the rapid progress in the task of video object tracking (VOT) at bounding box level, some works attempt to rely on the first-frame bounding boxes to provide target object information instead of using the first-frame masks, which dramatically accelerates the annotation process and increases scalability.

This kind of acceleration is, however, built on the sacrifice of ground-truth "accuracy". The reason is that some background area will be incorporated into the bounding box as well, which greatly increases the difficulty of the VOS

[*] corresponding author

[1] The work was supported by National Natural Science Foundation of China under 61972323,61532005 and U1936212, and Key Program Special Fund in XJTLU under KSF-T-02, KSF-P-02.

https://github.com/insomnia94/FTMU

task. In this way, in order to adapt to the characteristics of box-level ground-truth, most existing methods relying on the first-frame bounding boxes adopt the detection-based algorithms. Generally, these detection-based methods include three steps. The first step is to conduct object detection on the whole frame to generate the proposals of all possible objects using the region proposal network [23]. The second step is to do the matching process between the target object and all candidate proposals to find the "correct" proposal. The third step is to do the salience segmentation on the "correct" proposal to generate the final segmentation result. However, existing methods relying on the first-frame bounding boxes [31, 29] are less performing than the methods relying on first-frame masks both in terms of running speed and accuracy.

First, in terms of the running speed, we observe that most existing detection-based algorithms spend too much time on the matching process (e.g., 1.425s for matching process and 0.025s for segmentation process in [29]), using several time-consuming networks to evaluate the appearance similarity, like the re-identification network [25] and the siamese style network [29]. We observe that, for most video sequences, fast matching according to the intersection over union (IOU) between candidate proposals of the current frame and the obtained previous frame's bounding box or segmentation mask can also lead to acceptable performance, because the target object normally moves or changes slowly in two successive frames. However, simple IOU-based matching methods sometimes lose the target, especially when the target disappears out of the sight, and then reappears at a different location. Therefore, neither simple appearance-based matching nor simple IOU-based matching is the best solution for this task.

Second, the major constriction of VOS accuracy is its rough way to update the target template. The target template, containing the target's latest information including appearance, location and so on, plays an essential role in the matching process. Among all candidate proposals of the current frame, the proposal with the highest similarity with the target template will be selected. In this way, whether the correct proposal can be selected is determined by the quality of the target template as well as its update mechanism. Existing methods, however, simply replace the target template with the newly predicted result after one frame is finished, regardless of the correctness of the obtained result. Therefore, error will be gradually introduced into the target template, causing a great accuracy decline.

To achieve a better balance between accuracy and speed, a "smart switch" is required to make two significant decisions, including adopting which matching method, IOU-based matching or appearance-based matching, and whether to update the target template or not. To tackle this problem, we formalize it as a conditional decision-making process where only one simple reinforcement learning (RL) agent is employed to make decisions in a flexible way. As can be observed from Figure 1, provided with the optimal matching method and updating mechanism, our algorithm can be dramatically accelerated without losing targets even in some difficult frames, leading to a higher accuracy against previous state-of-the-art methods. Specifically, the running speed of our method is approximately 10 times faster than the previous state-of-the-art method.

To sum up, most video object tracking and segmentation algorithms consist of three steps. The first step is to conduct the instance segmentation on the current frame to generate a pool of candidate proposals. The second step is to conduct the matching process to find the correct one as the final result among all candidate proposals according to the target template information. The third step is to entirely replace the target template using the prediction of the current frame. In this paper, as we find the first step does not greatly affect the final result, our novelty lies in the improvement of the second and third steps:

- To improve the second step, our method provides a simple way to trade off between running speed and accuracy by selecting the matching method (IOU-based matching or appearance-based matching). The choice of the matching method is determined by the action history of the RL agent, which dramatically reduces the running time of our method.

- To improve the third step, as we observe the importance of the target template update mechanism to avoid drift, we argue that some predicted results with terrible quality should be discarded, and the target template should be kept unchanged in this situation. Specifically, we adopt a RL agent to make the decision on whether to update the target template or not according to the quality of the predicted result, which effectively prevents the drift problem and boots the accuracy of our method.

- The proposed approach has been validated on both VOS and VOT datasets, including DAVIS 2017, DAVIS 2016, SegTrack V2, Youtube-Object and VOT 2018 long-term datasets. Our method is approximately 10 times faster than the previous state-of-the-art method and achieve a higher mean region similarity at the same time. The new state-of-the-art mean region similarity is obtained on several datasets including DAVIS 2017 (69.1%), SegTrack V2(79.2%) and Youtube-Object (79.3%).

2. Related Work

2.1. Video Object Segmentation

VOS can be classified into three different categories including unsupervised VOS [26], interactive VOS [3, 30] and semi-supervised VOS [27, 3].

Unsupervised VOS is the task where no first-frame annotation is available at all. In [24], concatenated pyramid dilated convolution feature map is utilized to improve the final accuracy. Interactive VOS allows user annotation. In [3], an embedding model is trained to tell if two pixels belong to the same object, which proves to be efficient for this task.

Currently, semi-supervised VOS, where the first-frame ground-truth is provided, is still the main battlefield of the VOS tasks. The most common approach is to use the first-frame ground-truth to fine-tune the general segmentation network [2]. To adjust to the object appearance variation, in [28], the segmentation network will be updated during the test time. To overcome the speed shortcoming of the online updating, in [34], a meta learning model is utilized to speed up the process of online updating virtually without a reduction on the accuracy. To overcome the lack of the training data, it is proposed to utilize static images to generate more additional training samples in [32]. When the first-frame ground-truth is provided in the form of bounding box, in [31], original Siamese trackers is modified to generate the segmentation of the target object. In [29], original R-CNN network is modified to a conditional R-CNN network, and a temporal consistency re-scoring algorithm is utilized to find candidate proposals of the target object, followed by a salience segmentation network to find the final result.

2.2. Deep Reinforcement Learning

Currently, RL has been applied to many computer vision applications. In the task of VOT, [33] adopts RL to learn a similarity function for data association. In [4], RL is applied to choose the appropriate template from a template pool. In terms of VOS, Han et al. splits the VOS task into two subtasks including finding the optimal object box, and finding the context box [8]. This work is desired by the fact that the obtained segmentation masks vary under different object boxes and context boxes for an identical frame. In this way, RL is naturally suitable to select the optimal object box and context box for each frame.

3. Our Approach

3.1. Overview

Box-level semi-supervised VOS only provides the first-frame box-level ground-truth, instead of the first-frame pixel-level ground-truth. The main objective of our work is to utilize RL to boost the performance of box-level semi-supervised VOS in terms of both accuracy and running speed, by improving its matching mechanism.To do this, a RL agent is utilized to make two significant decisions simultaneously, including adopting which matching method, IOU-based matching or appearance-based matching, and whether to update the target template or not.

Specifically, as can be observed from Figure 2, the processing of the current frame f_t is split into three steps. The first step is to adopt the IOU-based matching to generate a temporary preliminary result. Specifically, the search region b_s of the targets is determined first (see Figure 3), which will be fed into a general instance segmentation network (e.g. YOLACT [5], Mask R-CNN [9]) to generate numerous candidate predictions. Then, IOU-based matching will be adopted to find the preliminary result among all candidate predictions.

The second step is to determine the update mechanism for the target object information (target template) according to the correctness and quality of the preliminary result, which is judged by the RL agent. If it is good, the target template will be entirely replaced by the preliminary result. Otherwise, the preliminary result will be discarded and the target template will keep unchanged. Ultimately, the final result is generated according to the target template.

The third step is to determine whether the appearance-based re-detection is essential for f_t. If the target is lost, in other words, the preliminary result keeps terrible for N successive frames, the target needs to be re-detected again using the appearance-based matching. Otherwise, re-detection for f_t is not needed, and the next frame f_{t+1} will be processed. In terms of the appearance-based re-detection, the whole frame, rather than b_s, will be fed into a general instance segmentation network. Then, a new result will be selected by the appearance-based matching method among all candidate predictions. The second step is conducted again to generate a new final result. Note that the third step is conducted no more than once for each frame.

3.2. Agent Action

A RL agent is used to address two complicated challenges during the matching process, which have been ignored by all existing detection-based VOS methods.

In our approach, *target template* is used to represent the target information, which is an important concept. As shown in Figure 4, the target template consists of the target's bounding box T_{box}, segmentation mask T_{mask}, cropped image $T_{box'}$ inside T_{box}, cropped image $T_{mask'}$ inside T_{mask} and the whole frame T_{frame}. Correspondingly, the *predicted result* incorporates the bounding box P_{box}, segmentation mask P_{mask}, cropped image $P_{box'}$ inside P_{box}, cropped image $P_{mask'}$ inside P_{mask} and the current frame P_{frame}.

The first challenge is to decide whether to update the tar-

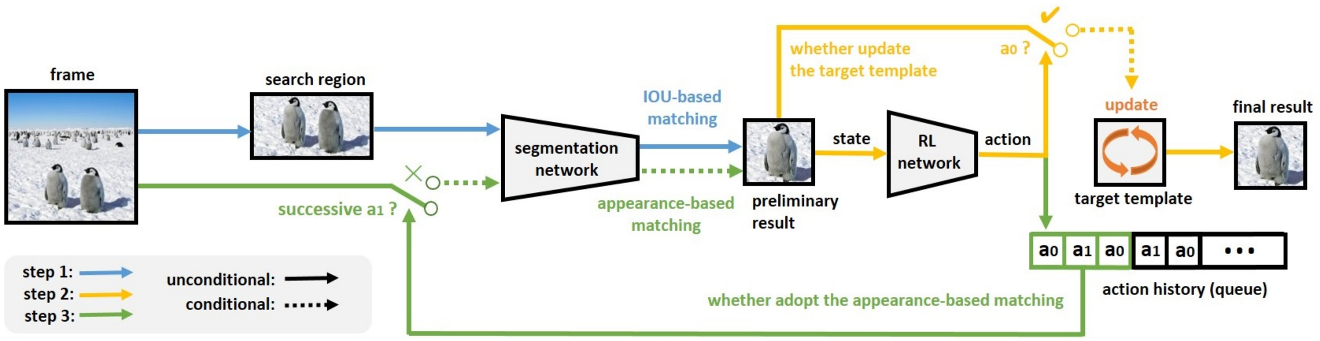

Figure 2. Architecture of our method where unconditional paths (full line) indicate they will be conducted in any situation, while conditional paths (dashed line) indicates they will be conducted only in some particular situations.

get template using the predicted result. Traditionally, the target template is updated in a rough way, without taking the correctness or quality of the predicted result into consideration. Therefore, when the segmentation network predicts a terrible result for current frame f_t, which may even refer to another object rather than the target, the target template will still be replaced by the incorrect result. This mortal error will make the tracker drift to the wrong target, causing a substantial accuracy drop. Note that this error cannot be avoided by adopting a better matching method because the target was decided before the matching process.

In this way, a "smart switch", which is able to decide whether to update the target template according to the quality of the predicted result, may be the best solution for this challenge. Rather than making the decision heuristically, we adopt a RL agent to make such a decision. The action set A for the RL agent contains 2 candidate actions $a_i \in A$, including a_0 to replace the target template using the predicted result of f_t, and a_1 to ignore the predicted result of f_t and keep the target template unchanged.

The second challenge is to decide whether to adopt the fast IOU-based matching method or the accurate appearance-based matching method. In our approach, IOU-based matching views the candidate prediction with the highest IOU score as the correct one, written as:

$$S_{IOU} = \alpha IOU(T_{box}, P_{box}) + \beta IOU(T_{mask}, P_{mask}), \tag{1}$$

$$\alpha + \beta = 1, \tag{2}$$

where α and β refer to the weight of these two IOUs.

Appearance-based matching views the candidate prediction with the highest appearance similarity as the correct one:

$$S_a = Similarity(T_{box'}, P_{box'}), \tag{3}$$

where $Similarity()$ is a Siamese style network, whose inputs are image patches within $T_{box'}$ and $P_{box'}$. Then, their individual embedding vectors are generated. A small L2 distance between these two vectors indicates two patches are similar, and vice versa.

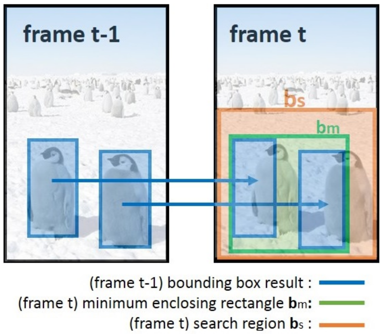
frame t-1
frame t
bs
bm
(frame t-1) bounding box result :
(frame t) minimum enclosing rectangle bm:
(frame t) search region bs :

Figure 3. Illustration of the minimum enclosing rectangle b_m and search region b_s. b_s is generated by b_m by expansion.

Having two matching mechanisms is inspired by the observation that the fast IOU-based matching performs well for most normal frames, while the appearance-based matching is only essential for a few difficult situations, especially when the target disappears and reappears again. Therefore, the selection of the matching method is pretty significant to trade off between running speed and accuracy.

Instead of adding one more RL agent, we decide to choose the matching method according to the action history. In fact, the action history intrinsically indicates the prediction quality. If the RL agent predicts a_1 for N successive frames, it is very likely that the target has been lost, and the appearance-based matching is essential to be adopted to detect the target on the whole frame.

3.3. State and Reward

The state s_t is the input of the RL agent for frame f_t, including the information assisting the RL agent to predict an optimal action a_t.

In our approach, s_t consists of two parts to provide sufficient information to the RL agent. The first part S_T is the modified image of T_{frame} where $T_{box'}$ remains unchanged while the area outside T_{box} is blackened, written as:

$$S_T = T_{box'} \cup \Phi(\{i | i \in T_{frame}, i \notin T_{box'}\}), \tag{4}$$

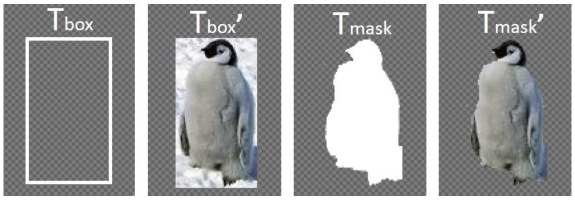

Figure 4. Illustration of elements in the target template (T_{box}, T_{mask}, $T_{box'}$, $T_{mask'}$). Elements in the predicted result (P_{box}, P_{mask}, $P_{box'}$, $P_{mask'}$) are in the same form.

where function Φ is to set all pixels black. It provides both the location and appearance information of the target in T_{frame}.

The second part S_P is the modified image of P_{frame} where $P_{mask'}$ remains unchanged while the area outside P_{mask} is blackened:

$$S_P = P_{mask'} \cup \Phi(\{i | i \in P_{frame}, i \notin P_{mask'}\}). \tag{5}$$

It provides both the location and appearance information of the predicted objects, as well as its segmentation information.

The ultimate s_t is the concatenation of the feature maps of S_T and S_P:

$$s_t = feature(S_T) + feature(S_P). \tag{6}$$

In details, we adopt Resnet-50 [10], pre-trained on the ImageNet classification dataset [6], to extract the feature map of S_T and S_P. We use the first 5 blocks of Resnet-50 [10] which results in a feature map with the size of $\mathbb{R}^{1\times1\times2048}$ for both S_T and S_P, and s_t is with size $\mathbb{R}^{1\times1\times4096}$. Finally, s_t will be fed into the RL agent to predict the action for frame f_t.

The reward function, which reflects the accuracy of the final segmentation result for the video sequence, is defined as $r_t = g(s_t, a_t)$:

$$g(s_t, a) = \begin{cases} 100J_t^3 + 10 & J_t > 0.1 \\ -10 & J_t \leq 0.1 \end{cases}, \tag{7}$$

where J_t refers to the IOU between P_{mask} and the ground-truth mask. Using the cube of J_t expands the difference between the good action's reward and the bad action's reward, which helps to speed up the training of the RL agent.

3.4. Search Region Size

The size of the search region b_s greatly affects the quality of the segmentation result. As can be observed from Figure 3, b_s is generated using b_m (minimal box covering all target objects in the previous frame) by expansion. The expansion ratio of b_m varies according to the video's characteristic. In detail, there are three expansion ratio from b_m to b_s, including a big one, a small one and an intermediate one. First, each target's displacement distance between two adjacent frames is calculated. If anyone is larger than a threshold, the big expansion ratio is selected. Otherwise, if two target objects are close to each other (IoU between their bounding boxes is not zero), the small expansion ratio is chosen. If not, the intermediate ratio is selected.

3.5. Actor-Critic Training

In our approach, the RL agent is trained under the "actor-critic" framework [12], which is a prevalent RL framework consists of two sub-networks including an "actor" sub-network to generate the action and a "critic" sub-network to check the quality of this action. Once the RL agent is fully trained, only the "actor" sub-network is used during the inference time.

In our "actor-critic" framework, given a current frame f_t, the first step is to feed the state s_t into the "actor" network and generate an action a_t to decide whether to update the target template using the predicted result. The corresponding reward r_t will also be obtained after conducting this action. r_t is calculated by the region similarity J_t according to (7).

Our "critic" network will be trained in the value-based way. Specifically, the parameters are updated as follows

$$w = w' + l_c \delta_t \nabla_{w'} V_{w'}(s_t), \tag{8}$$

where

$$\delta_t = r_t + \gamma V_{w'}(s_{t+1}) - V_{w'}(s_t). \tag{9}$$

In (8) and (9), w and w' indicate the weight of the "critic" model after and before update. l_c is the learning rate of the "critic" model. δ_t is the TD error which indicates the difference of the actual score and the predicted score. $V_{w'}(s_t)$ refers to the accumulated reward of state s_t which is predicted by the "critic" model before update. γ refers to the discount factor.

The "actor" network will be updated after the "critic" network in a policy-based way, as follows

$$\theta = \theta' + l_a \nabla(log\pi_{\theta'}(s_t, a_t))A(s_t, a_t), \tag{10}$$

where

$$A(s_t, s_t) = Q(s_t, a_t) - V(s_t) = \delta_t. \tag{11}$$

In (10) and (11), θ and θ' indicate the weight of the "actor" model after and before update. l_a is the learning rate of the "actor" model. $\pi(s, a)$ is the policy function which indicates the probability of selecting action a in state s. $V(s_t)$ is the score of the state s_t. $Q(s_t, a_t)$ is the score of the state s_t if the action a_t is executed. $A(s, a)$ refers to the advantage function.

In this way, our "actor-critic" framework avoids the disadvantages of both value-based and policy-based methods during the training process. In other words, our RL agent is allowed to be trained and updated at each frame, rather than waiting until the end of the episode, which dramatically speeds up the training process yet maintains training stability.

4. Implementation Details

4.1. Segmentation Network Training

In our approach, the training of the instance segmentation network follows the strategy of YOLACT [5]. The first step is to pre-train a ResNet-101 network [10] using the ImageNet classification dataset [6]. Then, this network with FPN [15] is used as the feature backbone for the segmentation network. Finally, the segmentation network will be trained on the PASCAL VOC dataset [7] with three losses including the classification loss, box regression loss and the mask loss calculated by the pixel-level binary cross-entropy between the predicted masks and the ground-truth.

4.2. RL Agent Training

Our RL agent is trained on the training set of the DAVIS 2017 dataset where all video sequences of the training set are divided into video clips with the fixed number of frames in advance. A video clip, consisting of 10 consecutive frames, is used as an episode for the training of the RL agent. 20 video clips will be randomly selected as a batch. At the beginning, the learning rate l_a for the "actor" model is 1e-4, and the learning rate l_c is 5e-4. l_a and l_c decrease gradually during the training, and they decrease by 1% for each 200 iterations. The discount rate γ for the reward is 0.9. The training of our RL agent takes about 10 days on a NVIDIA GTX 1080 Ti GPU and a 12 Core Intel i7-8700K CPU@3.7GHz.

In terms of other hyper-parameters, when calculating the S_{IOU} in (1), we found $\alpha = 1$, $\beta = 0$ for the first frame, and $\alpha = 0.5$, $\beta = 0.5$ for other frames work well, because the pixel-level ground-truth is not available for our task. For the appearance-based matching, we found it is better to re-detect the target using the appearance-based matching when action a_1 is taken for 3 successive frames.

5. Experiments

5.1. Experiment Setup

We split the experimental evaluation into two sections including the evaluation on the VOS dataset and the evaluation on the VOT dataset.

For the VOS experiments, we evaluate our method on four widely-used datasets including DAVIS 2017 dataset [21], DAVIS 2016 dataset [20], Youtube-Object dataset [22], and Segtrack V2 dataset [14]. DAVIS 2016 dataset consists of 50 high quality videos and 3,455 frames, with 30 videos for training and 20 videos for evaluation. In DAVIS 2016 dataset, only a single target is annotated per video sequence. DAVIS 2017 dataset extends DAVIS 2016 dataset, consisting of 60 video sequences for training and 30 video sequences for evaluation, spanning multiple occurrences of common video object segmentation challenges such as occlusions, motion-blur and appearance changes. In DAVIS 2017 dataset, each video sequence contains 2.03 object on average, and a maximum of 5 objects to be tracked in a single video sequence. In Youtube-object, there are 155 video sequences and a total of 570,000 frames. All these video sequences are divided into 10 sets according to the category. Youtube-Object dataset does not split the training set and the evaluation set, so we set all video sequences as the evaluation set. Note that Youtube-Object is not an instance-level dataset, in other words, for some videos, several individual targets are annotated into one object as the foreground, which does not completely match our task. Therefore, we split the annotations of these video sequences, so that each target owns its individual instance-level annotation. In SegTrack V2 dataset, there are 14 video sequences with more occlusion and appearance changes compared with Youtube-Object dataset. Originally, these datasets only provide the pixel-level ground-truth, which does not match our task. Therefore, we generate the bounding boxes according to the pixel-level ground-truth in advance as the first-frame box-level ground-truth.

We valuate our method following the approach proposed in [21]. The adopted evaluation metrics include region similarity J and contour accuracy F. The region similarity is calculated as $J = \left|\frac{m \cap gt}{m \cup gt}\right|$ by the intersection-over-union between the predicted segmentation m and the ground-truth gt. The contour accuracy is defined as $F = \frac{2P_cR_c}{P_c+R_c}$, which indicates the trade-off between counter-based precision P_c and recall R_c using the F-measure.

For the VOT experiments, we evaluate our VOS method on the LTB35 dataset [13] which is a long-term VOT dataset and was adopted to evaluate the long-term tracking performance in the VOT2018 challenge [13]. LTB35 dataset consists of 35 video sequences with 4,200 frames for each video sequence on average. In addition, the target will disappear and reappear again for 12.4 times, with an average target absence period of 40.6 frames per video. In this way, this dataset is suited to check the algorithm's ability to re-detect the disappeared target. We evaluate our method following the standard metric for LTB35 dataset. The performance is measured by the F score which is calculated as $F = \frac{2P_rR_e}{(P_r+R_e)}$, where P_r indicates the precision and R_e indicates the recall. Algorithms will be ranked by the maximum F score under different confidence thresholds.

5.2. Comparison with State-of-the-arts

For the experimental evaluation on the VOS dataset, we compare our method with other state-of-the-art VOS meth-

Table 1. Quantitative comparison with other methods on the DAVIS 2017 (Da 17), DAVIS 2016 (Da 16), SegTrack V2 (ST) and Youtube-Object (YOs) datasets, measured by the mean region similarity (J), as well as the average score of region similarity and boundary similarity ($J\&F$). **FT** indicates fine-tuning, **M** indicates using the first-frame masks, t(s) indicates the average running time per frame in seconds. The method with the best score is bold, and the method with the second best score is marked in underline.

Method	FT	M	t(s)	Da 17 - J	Da 17 - $J\&F$	Da 16 - J	Da 16 - $J\&F$	ST - J	YOs - J
Ours	✘	✘	0.09	**69.1**	70.6	77.5	78.9	**79.2**	**79.3**
BoLTVOS[29]	✘	✘	1.45	68.4	**71.9**	**78.1**	**79.6**	-	-
BoLTVOS *	✘	✘	1.45	60.9	64.9	**78.1**	**79.6**	-	-
SiamMask[31]	✘	✘	**0.06**	54.3	55.8	71.7	69.8	-	-
SiamMask *	✘	✘	0.11	59.5	63.3	75.6	75.9	-	-
MSK(box)[19]	✘	✘	12	-	-	73.7	-	62.4	69.3
STM[18]	✘	✔	0.16	69.2	74.1	84.8	88.1	-	-
FEELVOS[27]	✘	✔	0.51	69.1	71.5	81.5	81.8	-	78.9
RGMP[32]	✘	✔	0.28	64.8	66.7	81.1	81.7	71.7	-
VideoMatch[11]	✘	✔	0.35	56.5	62.4	81.0	80.9	79.9	-
OSMN[36]	✘	✔	0.28	52.5	54.8	74.0	73.5	-	69.0
PReMVOS[16]	✔	✔	37.21	73.9	77.8	85.6	86.5	-	-
OSVOS-S[17]	✔	✔	9	64.7	68.0	84.9	86.8	-	83.2
OnAVOS[28]	✔	✔	26	61.0	63.6	85.7	85.0	66.7	77.4
CINM[1]	✔	✔	>120	64.5	67.5	83.4	84.2	77.1	78.4

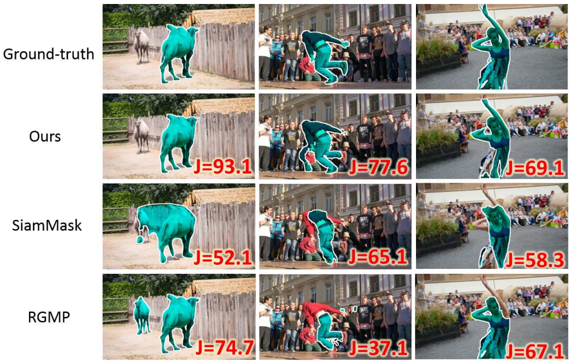

Figure 5. Visualization results of different methods.

ods, which are classified into three groups. Methods in the first group only use the first-frame box-level ground-truth, including BoLTVOS [29], SiamMask [31] and MSK [19]. Methods in the second group adopt the first-frame pixel-level ground-truth but do not fine-tune on it, including STM [18], FEELVOS [27], RGMP [32], VideoMatch [11] and OSMN [36]. Methods in the third group fine-tune on the first-frame pixel-level ground-truth, including PReMVOS [16], OSVOS-S [17], OnAVOS [28], and CINM [1]. All quantitative results of the comparison are summarized in Figure 1 and Table 1. In Table 1, the method with the highest score is highlighted and the method with the second best score is marked with underline.

As can be observed from Table 1, for the evaluation of the DAVIS 2017 dataset, compared with other methods which only rely on the first-frame box-level ground-truth (BoLTVOS* removes the re-scoring network and SiamMask* adopts the Box2Seg network of BoLTVOS), our method is virtually 15 times faster the previous state-of-the-art method BoLTVOS [29], and our accuracy (mean region similarity J_m) is even higher than BoLTVOS [29] at the same time. For another competitive method SiamMask [31], which runs virtually as fast as our method, our accuracy is much higher than it by around 15%. In addition, in the second group, the proposed method approximately achieves the same J_m as the state-of-the-art method STM[18], trained without the additional Youtube-VOS dataset [35], as it was not used in our method. When compared with the methods in the third group, our method also outperforms most of these methods. Only PReMVOS [16] achieves a higher accuracy than our method, but our method runs 370 times faster than it. For the evaluation of DAVIS 2016 dataset, as it is not an instance-level dataset, our method ranks the second among methods in the first group. For the evaluation for SegTrack V2 dataset and Youtube-Object dataset, our methods also achieves a competitive result even compared with methods using the first-frame pixel-level ground-truth. Some visualization results are shown in Figure 5.

For the experimental evaluation of VOT, the comparison with other state-of-the-art methods [13] is conducted on the VOT 2018 long-term dataset. As can be observed from Figure 6, our method achieves an F score of 0.622, which is quite competitive compared with other VOT methods. Overall our method achieves very good speed accu-

Table 2. Ablation studies on the DAVIS 2017 dataset, measured by the mean region similarity (J_m).

Method	J_m	t(s)
no update	27.3	0.03
simple update (IOU)	63.1	0.03
simple update (appearance)	67.2	1.10
RL update w/o re-detection	68.1	0.06
Ours	**69.1**	0.09
Ours (supervised)	60.2	0.09
BoLTVOS	68.4	1.45
SiamMask	54.3	0.06

racy trade-off for the VOT task. This result also shows our method is able to handle both VOS and VOT tasks, even in the situations where the target disappears and reappears again at a different place.

5.3. Ablation studies

Contribution of each component: We conduct ablation studies on DAVIS 2017 [21], where parts of our methods are disabled to investigate the contribution of each component.

First, we totally remove the target template update mechanism, obtaining the method **no update**, which cause numerous drift, and leads to a terrible accuracy (27.3%). Then, we evaluate the simple update mechanisms, where the target template will always be updated, and only IOU-based matching or only appearance-based matching is adopted to select the predicted result, obtaining method **simple update (IOU)** and method **simple update (appearance)** respectively. As can be observed from Table 2, only adopting IOU-based matching already achieves an acceptable accuracy (63.1%). Although the accuracy of method **simple update (appearance)** is 4.1% greater than method **simple update (IOU)**, the sacrifice on speed is unacceptable (from 0.03s to 1.1s), which proves the inefficiency of the simple target template update mechanism, where only appearance-based matching is adopted. In addition, the accuracy of method **simple update (appearance)** is still lower than that of our overall method, which demonstrate that the target template update issue cannot be totally solved simply by adopting an accurate matching method. Finally, we evaluate our method without the usage of the appearance-based matching for re-detection, obtaining method **RL update w/o re-detection**, whose accuracy is 1.0% lower than that of our overall method. This gap is not big, because the situation where the target disappears and reappears from another place is pretty rare in the DAVIS 2017 dataset.

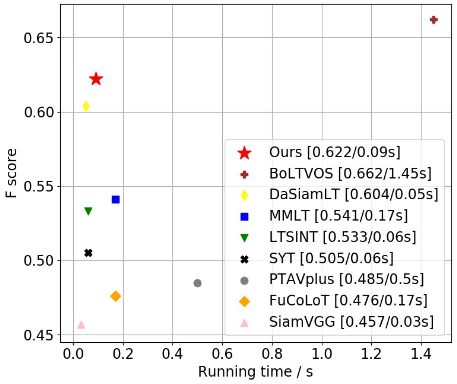

Figure 6. Speed-accuracy trade-off on VOT2018 LT dataset.

RL or supervised learning? Apart from training the network with RL, we also attempt to train the network in the supervised way, i.e. evaluating the fixed label for each frame in advance before training, but finally, as can be observed from Table 2, we find the model trained under the reinforcement learning way performs much better than the model trained under the supervised way, which achieve the accuracy of 69.1% (RL) and 60.2% (supervised). We believe the major reason is that, a RL model considers not only the current profit but also the potential profit in the future, due to the adopted accumulative future reward for training. In other words, the model trained in a supervised way tends to be myopic, while the model trained in the RL way pays more attention to the global and overall performance, making it more suited to video-related tasks.

Choice of instance segmentation: Apart from adopting the one-stage segmentation network, like YOLACT [5], we also attempt to adopt the two-stage segmentation network, like Mask R-CNN [9]. Then, the running speed of our method drops slightly from around 90ms to around 150ms, and it is still around 10 times faster than the previous state-of-the-art method, BoLTVOS [29]. As Mask R-CNN achieves higher accuracy than YOLACT [5], our final accuracy is even slightly higher than the proposed one. Note that the choice of the instance segmentation method does not greatly affect the final result, both for running speed and accuracy.

6. Conclusion

In this paper, an RL-based template matching and updating mechanism is proposed to handle box-level semi-supervised VOS. A single RL agent is applied to make these decisions jointly, which is trained using an actor-critic RL framework. Evaluation on common datasets for both VOS and VOT demonstrates the great performance of our method. In the future, we plan to design more matching mechanisms and template target update mechanisms to further improve the performance of our method.

References

[1] Linchao Bao, Baoyuan Wu, and Wei Liu. CNN in MRF: Video object segmentation via inference in a cnn-based higher-order spatio-temporal MRF. In *CVPR*, 2018. 7

[2] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017. 3

[3] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, 2018. 3

[4] Janghoon Choi, Junseok Kwon, and Kyoung Mu Lee. Real-time visual tracking by deep reinforced decision making. *CVIU*, 171:10–19, 2018. 3

[5] Fanyi Xiao Yong Jae Lee Daniel Bolya, Chong Zhou. YOLACT: Real-time instance segmentation. In *CVPR*, 2019. 3, 6, 8

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*. 5, 6

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge. In *IJCV*, 2010. 6

[8] Junwei Han, Le Yang, Dingwen Zhang, Xiaojun Chang, and Xiaodan Liang. Reinforcement cutting-agent learning for video object segmentation. In *CVPR*, 2018. 3

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *CVPR*, 2017. 3, 8

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 6

[11] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, 2018. 7

[12] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *NIPS*, 2000. 5

[13] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Cehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, et al. The sixth visual object tracking vot2018 challenge results. In *ECCV*, 2018. 6, 7

[14] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013. 6

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 6

[16] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. *arXiv:1807.09190*, 2018. 7

[17] Kevis-Kokitsi Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *arXiv:1709.06031*, 2017. 7

[18] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019. 7

[19] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 7

[20] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 6

[21] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 6, 8

[22] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 6

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2

[24] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *ECCV*, 2018. 3

[25] Peng Sun, Peiwen Lin, Guangliang Cheng, Jianping Shi, Jiawan Zhang, and Xi Li. OVSNet: Towards one-pass real-time video object segmentation. *arXiv:1905.10064*, 2019. 2

[26] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In *ICCV*, 2017. 3

[27] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 2019. 1, 3, 7

[28] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. *arXiv:1706.09364*, 2017. 3, 7

[29] Paul Voigtlaender, Jonathon Luiten, and Bastian Leibe. Boltvos: Box-level tracking for video object segmentation. *arXiv:1904.04552*, 2019. 2, 3, 7, 8

[30] Jue Wang, Pravin Bhat, R Alex Colburn, Maneesh Agrawala, and Michael F Cohen. Interactive video cutout. In *ToG*, 2005. 3

[31] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019. 2, 3, 7

[32] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, 2018. 1, 3, 7

[33] Yu Xiang, Alexandre Alahi, and Silvio Savarese. Learning to track: Online multi-object tracking by decision making. In *ICCV*, 2015. 3

[34] Huaxin Xiao, Bingyi Kang, Yu Liu, Maojun Zhang, and Jiashi Feng. Online meta adaptation for fast video object segmentation. *TPAMI*, 2019. 3

[35] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018. 7

[36] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018. 7