Pattern Recognition 106 (2020) 107465

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Adaptive ROI generation for video object segmentation using reinforcement learning

Mingjie Sun[a,b], Jimin Xiao[a,*], Eng Gee Lim[a], Yanchun Xie[a,b], Jiashi Feng[c]

[a] Xi'an Jiaotong-Liverpool University, Suzhou, China
[b] University of Liverpool, Liverpool, UK
[c] National University of Singapore, Singapore

ARTICLE INFO

Article history:
Received 5 October 2019
Revised 19 March 2020
Accepted 17 May 2020
Available online 24 May 2020

Keywords:
Model adaptation
Video object segmentation
Reinforcement learning
Training accelerate

ABSTRACT

The task of the proposed method is semi-supervised video object segmentation where only the ground-truth segmentation of the first frame is provided. The existing approaches rely on selecting the region of interest for model update; however it is rough and inflexible, leading to performance degradation. To overcome this limitation, a novel approach is proposed which utilizes reinforcement learning to select optimal adaptation areas for each frame, based on the historical segmentation information. The RL model learns to take optimal actions to adjust the region of interest inferred from the previous frame for online model updating. To speed up the model adaption, a novel multi-branch tree based exploration method is designed to quickly select the best state action pairs. The proposed method is evaluated on three common video object segmentation datasets including DAVIS 2016, SegTrack V2 and Youtube-Object. The results show that the proposed work improves the state-of-the-art of the mean region similarity to 87.1% on the DAVIS 2016 dataset, and to 79.5% on the Youtube-Object dataset. Meanwhile, competitive performance is obtained on the SegTrack V2 dataset. Code is at https://github.com/insomnia94/ARG.

1. Introduction

Video object segmentation (VOS) is a fundamental problem in the computer vision field with many applications including video editing, video surveillance, and scene understanding. The objective of single target video object segmentation is to label each pixel as foreground or background in a given frame of a video sequence. Labeling these pixels, however, is difficult due to background clutter, illumination change, motion blur, deformation, and so on.

Inspired by the fact that the online adaption has achieved great progress on video object tracking at bounding box level [1,2], some researches start to deploy the online adaption to video object segmentation [3]. However, for online VOS model update, how to identify the region of interest (ROI) for model adaptation is critical. In particular, as can be observed in Fig. 1, if the adaptation area is selected in a rough way, the segmentation model after adaptation performs very poorly when there are multiple similar objects in the frame, especially when the distraction object is very close to the target. As such selecting the optimal adaptation area in a more sophisticated and flexible way is very significant for video object segmentation.

To tackle this problem, VOS is formalized as a conditional decision-making process where two reinforcement learning (RL) agents are employed to infer and adjust the ROI for adaption for both foreground and background in a flexible way. At each frame, the RL agent outputs actions to tune the ROI. Provided with such regions, the VOS model can be updated to be more specific and discriminating for the instance in the current frame.

To select the optimal adaptation area for video object segmentation, a set of features of different adaptation areas of the current frames will be fed into the RL model. Then, the RL model will select the best action, and choose the most suitable adaptation area for the current frame. As a result, the segmentation model can obtain an accurate segmentation result. Though the RL method is promising for region identification, it is notoriously slow to optimize the agent. In this work, to speed up the RL agent training, a multi-branch tree based policy search method is proposed where possible action state pairs are organized in a tree structure.

To sum up, this paper has three main contributions:

- Due to the importance of identifying the accurate ROI for VOS model adaptation, some attempts are made to mine optimal adaptation ROIs for online adaptation in video object segmentation. To improve the mined ROI quality, the VOS segmenta-

* Corresponding author.
E-mail addresses: mingjie.sun@liverpool.ac.uk (M. Sun), jimin.xiao@xjtlu.edu.com (J. Xiao).

https://doi.org/10.1016/j.patcog.2020.107465
0031-3203/

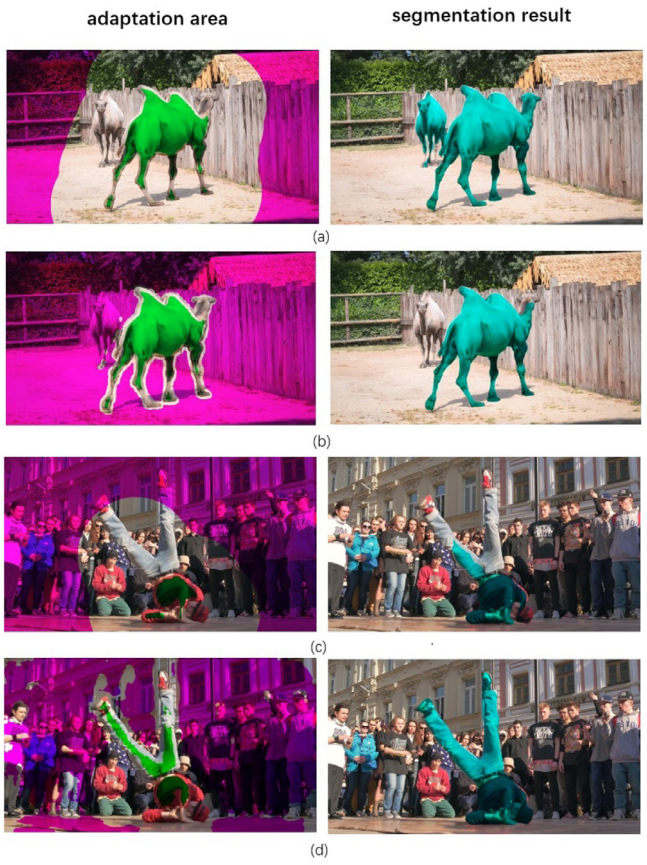

Fig. 1. Different adaptation areas lead to different frame segmentation results. The segmentation models are the same before online adaptation. In the left column, the area in green is the adaptation area for foreground and the area in purple is the adaptation area for background. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

tion accuracy for future frames in a video is used to judge the mined ROI quality for the current frame. Specifically, the actor-critic reinforcement learning framework is deployed to train an agent to generate accurate adaptation ROI areas. To the best of knowledge, this is the first attempt to use RL to mine accurate ROI for VOS model adaptation.

- Both VOS with online adaptation and the RL model training are computationally demanding processes. To speed up the RL training, a novel multi-branch tree based exploration method is designed to quickly select the best state action pairs.
- The proposed approach has been validated on the DAVIS 2016, SegTrack V2 and Youtube-Object datasets. New state-of-the-art results of the mean region similarity (Jm) are obtained for both the DAVIS 2016 dataset (87.1%) and Youtube-Object dataset (79.5%), which are higher than the previous state-of-the-art methods, including [3] for the DAVIS 2016 dataset (85.7%) and [4] for the Youtube-Object dataset (78.4%), by 1.4% and 1.1% respectively. Meanwhile, competitive performance is obtained on the SegTrack V2 dataset.

2. Related work

2.1. Video object segmentation

Recently, with the popularity of deep neural network, various deep learning based video object segmentation models have been proposed. These existing approaches can be classified into three different categories, including unsupervised methods [5], interactive methods [6] and semi-supervised methods [7]. For semi-supervised segmentation, the ground-truth annotation for each pixel of the first frame is provided, and the objective of the segmentation model is to identify pixels of foreground and background for the following frames. Unsupervised methods and interactive methods, however, are more difficult than semi-supervised methods because no pixel-level annotations are available.

For the unsupervised video object segmentation where no target annotation is provided, utilizing motion information and detecting the primary object in a frame are two common ways to address this problem. In [5], Pavel et al. introduce a memory-based method which combines the object appearance information and the motion information together and achieve good performance. Specifically, the two streams of the network are used to extract the target's spatial and temporal information respectively, and the memory module adjusts the appearance feature to the target object change. In [8], Song et al. propose to use concatenated pyramid dilated convolution features and dramatically improve the final accuracy. In detail, this model consists of two parts, including the Pyramid Dilated Convolution (PDC) module to encode the multi-scale spatial information with different receptive fields, and the Deeper Bidirectional ConvLSTM module to extract the sequential spatiotemporal information at different scales.

For the interactive video object segmentation, where the first-frame annotation is in the form of sparse points clicked by the user, Chen et al. [6] adopt a learned embedding space to tackle this problem, where features of the same object are close to each other. In this way, this method supports different kinds of user input, enabling the interactive video object segmentation.

Semi-supervised video object segmentation task, where the pixel-level annotation of the first video frame is available, is also an extensively studied task. One-shot video object segmentation (OSVOS) [9] achieves great success relying on a general image segmentation network. The core idea of this method is to fine-tune an ImageNet [10] pre-trained ConvNet on the video object segmentation dataset, such as DAVIS [11,12], to allow the segmentation model to find general objects in a frame. Then, the first frame of the inference video sequence will be used to fine-tune the segmentation model such that it can rapidly focus on the specified target object instance in the first frame. Despite the promising result obtained using OSVOS, such a method only learns the target object appearance from the first frame of a video sequence. It cannot adapt to the target object appearance variation when deformation occurs or the camera rotates.

In order to adapt to object appearance variation, Voigtlaender et al [3] propose an online adaptive video object segmentation method which enables it to update the segmentation model during inference time. In detail, before finally segmenting the current frame of a video sequence, a part of pixels in the current frame will be used to update the current segmentation model to adapt to the change of the foreground and background. These pixels consist of positive ROI regarded as foreground, and the negative ROI regarded as background, which are chosen according to the temporary (before online adaptation) probability map and two fixed thresholds. However, model drift may interrupt the model online adaptation and lead to performance drop.

In terms of other recent related work, in [13], another method is proposed where proposals will be generated first, and then they will be merged into accurate and temporally consistent pixel-wise object tracks. In [4], this task is viewed as a spatio-temporal Markov random filed problem, and ConvNet is utilized to encode the dependencies among pixels. To overcome the shortage of training data, it is proposed to use static images to generate additional training samples in [14]. In [15], Everingham et al. attempt to utilize the part-based tracking method to generate bounding box and segmentation for each part.

Different from the existing works, the selection of ROI for online segmentation adaptation is formulated as a Markov decision process and utilizes the RL to address this problem.

2.2. Deep reinforcement learning

The RL algorithm learns to achieve a complex objective from past experience. "actor-critic" [16] is a popular RL framework that inherits several previous RL frameworks including Deep Q-learning [17] and policy gradient [18], which are valued-based and policy-based strategies, respectively.

RL has been applied to many areas of computer vision, in particular for visual object tracking at bounding box level. In [19], Yun et al. use RL to choose sequential actions to move the bounding box step by step from the original object location in the previous frame to the correct location. In other words, the tracker is moved by the predicted action from the current state, and then the next action is predicted according to the new position. In [20], Huang et al. propose a novel RL-based model which can choose the most suitable number of deep convolutional layers according to the current frame complexity, which dramatically reduces the running time without losing accuracy. Its core idea is that easy frames are processed with cheap features extracted by fewer convolutional layers, while challenging frames are processed with invariant but expensive deep features extracted by more convolutional layers. In [21], Dong et al. deploy the RL to choose the optimal hyperparameters for correlation filter. In addition, as the traditional continuous Deep Q-Learning algorithms cannot be directly applied, they introduce an efficient heuristic to accelerate the convergence behavior.

In video object segmentation, to the best of our knowledge, there is only one attempt to apply RL to this task. Han et al. establish a novel RL framework which can choose the optimal object box and the context box [22]. This work is motivated by the observation that, for an identical segmentation model, different object boxes and context boxes generate different segmentation masks. Thus, it is natural to utilize RL to choose an optimal object-context box pair to achieve the best segmentation result. Different from Han et al. [22] where RL is utilized to decide the size of the search region which will be fed into the segmentation network to generate the final result during the inference process, the proposed work aims to utilize the RL to choose the optimal ROI to update the segmentation network before the inference process.

3. Our approach

3.1. Overview

In general, the main objective of the proposed work is to utilize RL to improve the performance of video object segmentation with online adaptation. Different from the existing approaches to select the adaptation ROI in a rough way, the proposed work utilizes RL to choose the optimal adaptation ROI for each individual frame to avoid model drift. In other words, different frames will own its particular standard to choose adaptation ROI. The overview diagram of the proposed method is described in Fig. 2.

Existing VOS methods with online adaptation, i.e., OnAVOS [3], adopt a fixed standard to select the adaptation area, where different characteristics of each frame are not considered. To address this problem, flexible thresholds, t_p and t_n, are used to choose the adaptation ROI. Two RL models are built to choose the most suitable t_p and t_n for each frame. The RL framework includes state $s \in S$, threshold selection action $a^p \in A^p$ to determine the value of t_p and threshold selection action $a^n \in A^n$ to determine the value of t_n, state transition function $s' = T(s, a^p, a^n)$ and the reward function $g(s, a^p, a^n)$.

Specifically, given a frame F_t of one sequence, first of all, F_t will be fed into the segmentation network to obtain a temporary probability map M_f. A set of ROIs can be obtained according to the candidate thresholds. In this work, 5 ROIs are used where the probability value is greater than α_{large}, α_{small}, β_{large}, β_{medium} and β_{small}, respectively. Note that β_{micro} is ignored to diminish the complexity of the state. The first two ROIs are used for the RL model to choose t_p while the last three ROIs are used for RL model to choose t_n. In other words, the possible values of t_p can be α_{large} or α_{small}, and the possible values of t_n can be β_{large}, β_{medium}, β_{small} or β_{micro}. After t_p and t_n are determined, pixels with probability values greater than t_p will be viewed as the adaptation ROI for foreground, as follows:

$$S_p = \{i | i \in F_t, M_f(i) > t_p\}, \tag{1}$$

where F_t refers to current frame, S_p indicates the positive ROI consisting of the pixels regarded as foreground, M_f is the temporary (before online adaptation) probability map. Inspired by the great advantages of the recent fusion algorithms [23,24], these pixels with probability value less than t_n, combined with the negative pixels far away from S_p, will be regarded as the adaptation ROI for background, as follows:

$$S_n = \{i | i \in F_t, distance(i) > T_n\} \cup \{i | i \in F_t, M_f(i) < t_n\}, \tag{2}$$

where S_p indicates the negative ROI consisting of the pixels regarded as background, $distance(i)$ is to calculate the distance of pixel i to S_p, and T_n is its threshold. Ultimately, S_p and S_n are used to update the current segmentation model while other pixels are ignored. After the segmentation model has been updated, F_t will be fed into the new segmentation network and the final segmentation result is obtained. The pseudo-code of the algorithm is described in Algorithm 1.

Algorithm 1 RL Based video object segmentation.

Input: Ground-truth of the first frame $gt(1)$
Sequence length L
Distance threshold T_n
Segmentation network *Seg_Net*
Pretrained VGG netwrok *VGG*
RL model to choose positive threshold RL_p
RL model to choose negative threshold RL_n
Output: Segmentaion result of Frame t O_t

```
Fine-tune Seg_Net on F_1
last_mask ← gt(1)
for t = 2 to L do
    Obtain 2 RL states using (3) and (4), respectively.
    Feed the states into the RL models and achieve the optimal threshold t_p and t_n.
    Obtain positive ROI and negative ROI using (1), respectively.
    Update Seg_Net on F_t using S_p and S_n.
    O_t ← forward(Seg_Net, F_t) > 0.5
    last_mask ← O_t
    set h(t) = r(t)
end for
```

3.2. Agent action

The framework of the proposed work consists of two RL models, including one to choose t_p and another to choose t_n, as shown in Fig. 3. The action set A^p used for the first RL model to choose S_p contains 2 candidate thresholds: α_{large} and α_{small}. As the difference of the ROI areas with different t_p is not very big, only two candidate thresholds are set for the action set A^p. The action set A^n used

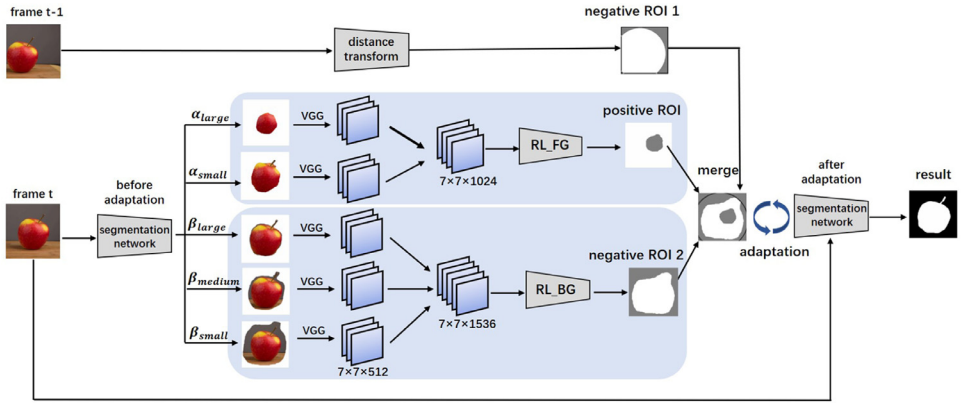

Fig. 2. The network architecture of this proposed work consists of two RL models. One is to choose the adaptation area for foreground, and another is to choose the adaptation area for background. The pre-trained VGG19 model is used to extract the feature of each ROI area of the current frame, and then feed the combined feature into the RL model as the state. Finally, the RL model will choose the best adaptation area and update the segmentation model using the chosen adaptation area.

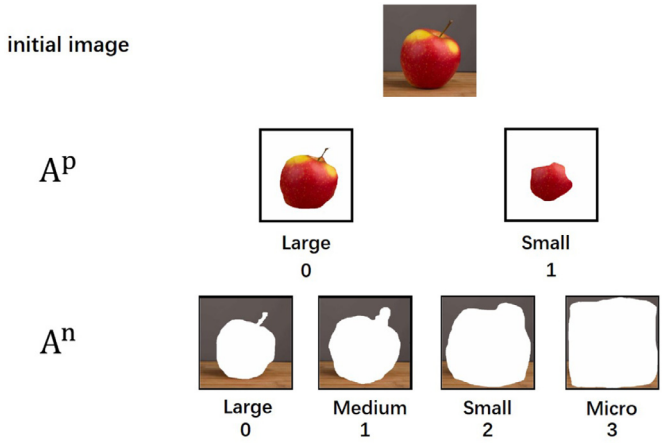

Fig. 3. Foreground adaptation ROI selecting action set A^p and background adaptation ROI selecting action set A^n.

for the second RL model to choose S_n contains 4 candidate thresholds: β_{large}, β_{medium}, β_{small} and β_{micro}. As the size of S_n is much larger than that of S_p, more candidate thresholds for t_n are set for adaptation ROIs for background.

In terms of the design of the candidate values of t_p, in OnAVOS [3], T_p is 0.97 and it is fixed for all frames. This threshold is very safe and conservative because it should work in any situation, especially for some frames with very bad segmentation result. In fact, for some frames with good segmentation result, the value of t_p ought to be much lower, so more correct pixels of the object can be used to update the model and improve the final segmentation result. As the difference of the size of the adaptation ROIs with different t_p is not very huge, only two candidate thresholds are adopted for the action set A^p, which can lower the difficulty of training the RL model.

In terms of the design of the candidate values of t_n, in OnAVOS [3], S_n are chosen according to the distance to the target object rather than the value of the M_f. If a pixel is far away from the target, it will be viewed as S_n. Similarly, the selection of S_n in OnAVOS [3] is very conservative as it works for almost all frames. In this work, the adaptation ROI which are chosen by the distance is kept, and then try to add more areas to include more correct adaptation ROI for background. The additional area is chosen by the value of M_f. Finally, these two parts of areas are combined together as the final area of S_n. As the size of the area for S_n is much larger than the area of S_p, more candidate thresholds for t_n are adopted.

In this way, after training, these two RL models can choose the best action and achieve the optimal thresholds t_p and t_n.

3.3. State and reward

The state s is the input of the RL model. As this method consists of two RL models, two sets of states are adopted for two models. In general, the state s is a feature map combined by the feature maps of different candidate adaptation ROIs where the probability value of all pixels in a certain adaptation ROI is less than a certain threshold.

First, given a certain frame F_t, F_t is fed into the segmentation model before test time adaptation and generate the temporary probability map M_f. Then, 5 ROIs are generated with 5 different candidate thresholds according to the value of M_f. The ROIs where the probability values of all pixels in this area are greater than α_{large} and α_{small}, receptively, are combined as the state of the RL model to choose t_p, as follows,

$$\begin{aligned} state_p = {} & feature(\{i|i \in F_t, M_f(i) > \alpha_{large}\}) \\ & + feature(\{i|i \in F_t, M_f(i) > \alpha_{small}\}). \end{aligned} \tag{3}$$

The ROIs where the probability values of all pixels in this area are greater than β_{large}, β_{medium} and β_{small}, receptively, are combined as the state of the RL model to choose t_n, as follows,

$$\begin{aligned} state_n = {} & feature(\{i|i \in F_t, M_f(i) > \beta_{large}\}) \\ & + feature(\{i|i \in F_t, M_f(i) > \beta_{medium}\}) \\ & + feature(\{i|i \in F_t, M_f(i) > \beta_{small}\}). \end{aligned} \tag{4}$$

Note that β_{micro} is ignored in (4) to diminish the complexity of $state_n$, as $feature(\{i|i \in F_t, M_f(i) > \beta_{small}\})$ already provides sufficient information to help the RL model make the decision. the VGG model [25,26], pre-trained on the ImageNet classification dataset [10], is used to extract the features of these ROIs first. Then, these features are concatenated together as the state of the RL model. The first 5 convolutional blocks of the VGG19 model are used which results in a feature size of $\mathbb{R}^{7\times7\times512}$ for one ROI. For the RL model to choose t_p, the features of two ROIs will be concatenated to generate the final state $s_{t_p} \in \mathbb{R}^{7\times7\times1024}$. For the RL model to choose t_n, the features of three ROIs will be concatenated to generate the final state $s_{t_n} \in \mathbb{R}^{7\times7\times1536}$. Finally, states s_{t_p} and s_{t_n} will be fed into the corresponding RL model and result in the actions to choose the optimal thresholds t_p and t_n.

The reward function is defined as $r_t = g(s_t, a_p, a_n)$ which reflects the performance of the final segmentation result of each

frame in the video sequence:

$$g(s_t, a_t, a_n) = \begin{cases} IOU + 1 & IOU > 0.1 \\ -1 & IOU <= 0.1 \end{cases}, \quad (5)$$

where *IOU* indicates the intersection-over-union (IOU) between the prediction and the ground-truth, which reflect the quality of the predicted segmentation.

3.4. Training in actor-critic framework

In this work, the "actor-critic" framework [16] is adopted for RL training. In general, one "actor-critic" framework consists of two roles including an "actor" role to generate an action and a "critic" role to measure how good this action is. In this work, it is critical to select the optimal adaptation ROIs for both foreground and background separately. Therefore, two "actor-critic" model pairs are adopted, including one "actor-critic" pair for foreground, and another pair for background. Four individual RL models are deployed in total.

In the "actor-critic" framework, given a current frame F_t, the first step is to feed the state into the "actor" network and generate an action a, which is to choose the optimal adaptation ROIs. The corresponding reward r_t will also be obtained after conducting this action. r_t is decided by the IOU of the segmentation result according to (5).

In the training process, after the forward process, the "critic" network will be updated first in the valued-based way, as follows:

$$w = w' + \alpha * \delta_t \nabla_{w'} V_{w'}(s_t), \quad (6)$$

where

$$\delta_t = r_t + \gamma * V_{w'}(s_{t+1}) - V_{w'}(s_t). \quad (7)$$

In (6) and (7), w and w' **indicate the weight of the "critic" model after and before update, respectively.** α 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.

After the "critic" model has been updated, the "actor" model will be updated in a policy-based way, as follows:

$$\theta = \theta' + \beta * \nabla(log\pi_{\theta'}(s_t, a_t)) * A(s_t, a_t), \quad (8)$$

where $A(s, a)$ refers to the advantage function, and $A(s_t, a_t) = \delta_t$ according to (7), θ and θ' indicate the weight of the "actor" model after and before update, respectively. β is the learning rate of the "actor" model. Policy function $\pi(s, a)$ is a network whose input is the state s and a certain action a, and output is the probability of selection action a in state s.

In this way, when training the RL models, the "actor-critic" framework can avoid the shortage of value-based and policy-based methods. Instead of waiting until the end of the episode, the RL models can be updated at each step, which dramatically reduces the training time but maintains the RL training stability.

4. Implementation details

4.1. Train the segmentation network

The proposed method trains the segmentation network follows the strategy of [9] and [3]. The first step is to train a ImageNet pre-trained network on a pixel-level annotated dataset such as PASCAL VOC [27]. In the second step, the DAVIS video dataset [11] is used to train the network so that the network is able to adapt to this dataset. In addition, the network is fine-tuned on the first frame of DAVIS test videos whose ground-truth annotation is provided. In this way, the segmentation network is well trained and can achieve a good result before test time adaptation.

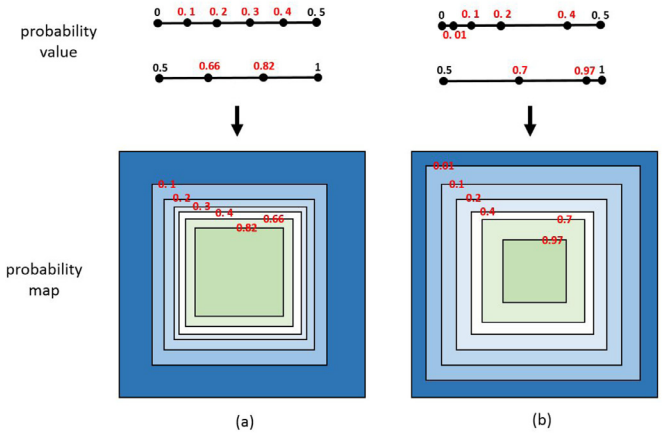

Fig. 4. Value-based and area-based interval for different adjacent candidate thresholds. Blue areas are adaptation ROIs for background while green areas are for foreground. Best Viewed in Color. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.2. Train the RL model

Before training the RL model, the related data need to be stored in advance to accelerate the training process, which will be described in Section 4.3. In terms of the training of the RL models, specifically, all video sequences in DAVIS 2016 training set are divided into video clips with the fixed number of frames. A video clip includes 10/5 consecutive frames is used as a sample for the RL model to select foreground/background ROI. Using stored clips for training dramatically reduces the training time of the RL model.

For the number of the adaptation ROIs, ultimately, 2 foreground adaptation ROIs (large and small) and 4 background adaptation ROIs (large, medium, small and micro) are adopted. During the parameter adjustment process, to make the adaptation ROI more precise and suitable for each frame, it was planed to adopt 10 adaptation ROIs both for foreground and background ROIs, but too many ROIs lead to a complicated state for the RL model, making the training too difficult. Then, the number of adaptation ROIs was gradually decreased, and it finally worked well when 2 foreground adaptation ROIs and 4 background ROIs were adopted. In addition, if we continued to decrease the number of adaptation ROIs, the accuracy dropped down quickly. In this way, the ROI setting as proposed is chosen ultimately.

For the value of each candidate threshold, it is found that $\alpha_{large} = 0.97$, $\alpha_{small} = 0.7$, $\beta_{large} = 0.4$, $\beta_{medium} = 0.2$, $\beta_{small} = 0.1$ and $\beta_{micro} = 0.01$ works well. During the parameter adjustment process, first, it was planned to set the value-based interval between adjacent candidate thresholds equal ($\alpha_{large} = 0.82$, $\alpha_{small} = 0.66$, $\beta_{large} = 0.4$, $\beta_{medium} = 0.3$, $\beta_{small} = 0.2$, $\beta_{micro} = 0.1$), which did not perform well. The reason is that, the area-based interval (in probability map) between adjacent candidate thresholds was not equal in this setting, as shown in (a) of Fig. 4. In this way, we gradually modified the values of each candidate threshold, and ultimately made the area-based interval for each adjacent candidate thresholds approximately equal in the probability map, as shown in (b) of Fig. 4.

For the learning rate, at first, to speed up the training process, a large learning rate (1e-3) was chosen, but it quickly converged towards an incorrect position. Then, the value of learning rate was decreased gradually. Finally, it was found that the learning rate $\alpha = 1e-5$ for "actor" model, and the leaning rate $\beta = 5e-5$ for "critic" model worked well.

In addition, 20 clips are randomly selected as a batch for training the RL models, and it is found that the number of clips has

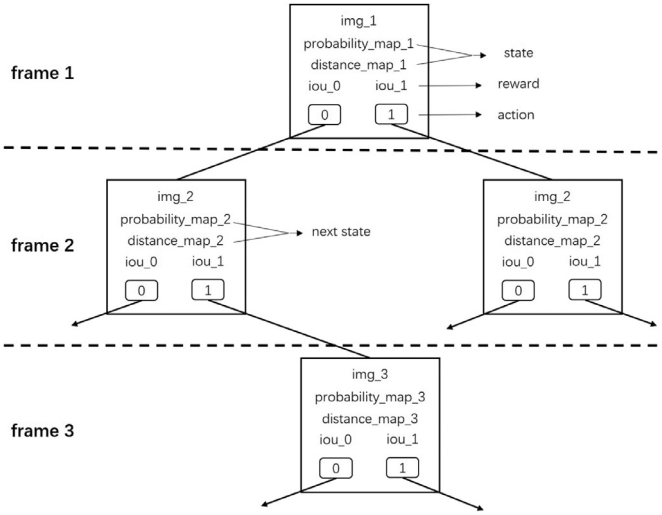

Fig. 5. The data structure used to restore the related data to accelerate the RL model training. For RL model to select the t_n, distance map is required for the training, while the training for the RL mode to selected t_p does not need it.

limited influence on the final result. The training of the RL models takes about 3 days on a NVIDIA GTX 1080 Ti GPU and a 12 Core Intel i7-8700K CPU@3.7GHz.

4.3. Accelerating RL training

Segmentation with online adaptation is slow because the segmentation model should be updated for each frame. Meanwhile, RL training itself is also slow. Training the RL model heavily relies on a large number of attempts for different actions. Normally, to generate a well trained RL model, the model should be trained with more than one million iterations. If the running time for each training process is too long, the total time is unbearable. Thus, it is impossible to train the RL model in a regular way.

To address this issue, inspired by the idea of sacrificing space to improve efficiency, a novel multi-branch tree structure, as shown in Fig. 5, is proposed to store all possible segmentation results using different adaptation ROIs into a repository in advance. Training an "actor-critic" framework needs 4 types of information including the action, the reward, the states before and after the action. For each node in the multi-branch tree, a corresponding directory is generated. The image of the frame, the temporary probability map and the IOU of the segmentation result after executing a certain action are stored as files. All possible actions are stored as a links to next layer of nodes. In this way, the image and the probability map are used to generate the state. The IOU value is used to generate the reward. Finally, this repository will be organized in a multi-branch tree structure, whose stored data will be used to update the RL model during the process of training using the method described in Section 3.4.

5. Experiments

5.1. Experiment setup

The proposed method is evaluated on three widely-used datasets including the DAVIS 2016 dataset [11], Youtube-Object data-set [32] and Segtrack V2 dataset [33]. DAVIS 2016 dataset consists of 50 high quality video sequences and 3455 frames, spanning multiple occurrences of common video object segmentation challenges such as occlusions, motion-blur and appearance changes. 30 video sequences of DAVIS 2016 are used for training, and 20 video sequences are used for testing. In DAVIS 2016, in each video sequence, only a single object instance is annotated. **The DAVIS 2017 dataset** [12] extends the DAVIS 2016 dataset where multiple objects, rather than only one object, are annotated in each frame. As the proposed method targets for single instance segmentation, the experiment is only conducted on DAVIS 2016. In Youtube-Object, there are 155 video sequences and a total of 570,000 frames. These video sequences are divided into 10 classes. Training set and testing set are not separated in Youtube-Object dataset so it is only used for testing. In SegTrack V2 dataset, there are 14 video sequences with more occlusion than appearance changes compared with Youtube-Object dataset.

The proposed method is evaluated following the approach proposed in [11]. 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.

5.2. Comparison with state-of-the-arts

In this section, the proposed work is compared with other state-of-the-art semi-supervised video object segmentation methods, including PReMVOS [13], OnAVOS [3], CINM [4], LucidTracker [28], MSK [29], OSVOS [9], STV [30], ObjectFlow [31]. Note that OSVOS-S [34] is not included in the comparison list as it utilizes additional dataset for training.

Table 1 summarizes the quantitative results of recent methods on the DAVIS 2016 validation set consisting of 20 videos. The top 3 performing methods have been highlighted with different colors. It can be observed that this work has achieved outstanding result under both mean region similarity J_m and the mean contour accuracy F_m. Especially on mean region similarity J_m, the proposed method achieves the best result which outperforms any existing state-of-the-art methods. Compared with the most competitive and related method OnAVOS [3], the proposed method improves the mean region similarity J_m to 87.1%. It should be noted that the gain over [3] is solely due to the fact that better ROIs are obtained for online adaptation using RL. Note that, same to [35], according to the randomness of the segmentation network, the final accuracy may fluctuate around 0.4%. The proposed mean region similarity J_m is the average value from experiments of 10 times.

Fig. 6 shows the qualitative segmentation masks for different methods. As can be observed, this method performs better on videos with significant appearance change for the target object, for instance, the camel and breakdance video sequences. Especially when multiple similar objects are close to each other, e.g., the camel sequence, this method has the ability to distinguish the target object from other similar objects successfully.

On SegTrack V2 and Youtube-Object datasets, as training set and evaluation set are not split, all video sequences are used for evaluation. From Table 1, we can observe that this method also performs well on both datasets. Compared with OnAVOS [3], this method improve the mean region similarity J_m by 10.8% on SegTrack V2 dataset, which demonstrates the effectiveness of the RL models to choose the online adaptation ROIs. In addtion, the proposed approach also improves the mean region similarity J_m by 2.1% on Youtube-Object dataset. This result can show the robustness of this method on different evaluation datasets.

In addition, the run time of the proposed method is compared with other state-of-the-art methods, and the result is also reported in Table 1. In spite of the improvement of the mean region similarity J_m against the baseline method with online adaptation [3] by 10.8% on SegTrack V2 dataset and 1.4% on Davis 2016 dataset, the

Table 1
Quantitative comparison with other methods on the DAVIS 2016, SegTrack V2 and Youtube-Object dataset. For J_m and F_m, the method wiht the best performance is bold, and the method with the second best performance is marked with underline.

Method	DAVIS-16 J_m	DAVIS-16 F_m	SegTrack V2 J_m	Youtube Objs J_m	t(s)
PReMVOS [13]	84.9	**88.6**	-	-	30
OnAVOS [3]	<u>85.7</u>	84.8	66.7	77.4	13
CINM [4]	83.4	85.0	77.1	<u>78.4</u>	120
Lucid [28]	83.7	-	76.8	76.2	40
MSK [29]	79.7	75.4	72.1	75.6	12
OSVOS [9]	79.8	80.6	65.4	78.3	10
STV [30]	73.6	-	**78.1**	-	-
ObjFlow [31]	68.0	-	74.1	77.6	-
OURS	**87.1**	<u>86.1</u>	<u>77.5</u>	**79.5**	14

Fig. 6. Visualization for the segmentation masks of different methods.

run time of this method is only about around 1 s longer than [3], which demonstrates the efficiency of the proposed method.

5.3. Ablation studies

In this section, four ablation studies are conducted on the proposed method using the testing video sequences of DAVIS 2016 dataset and SegTrack V2 dataset.

Contribution of Each Component: the first ablation studies is conducted on the DAVIS 2016 and SegTrack V2 datasets, where parts of this method are disabled to investigate the impact of each component. In this study, the contribution of each individual RL model in this method is explored, where one of these two RL models is disabled during this ablation study respectively, and the results will be compared with the result generated by the method with full RL models. Table 2 shows the result of this ablation study on the DAVIS 2016 dataset. On the DAVIS 2016 dataset, when using both foreground and background adaptation, the proposed method obtains the mean region similarity J_m of 86.5% without CRF [36,37]. J_m of this method with full adaptation is greater than the method without any adaptation by 6.2%, which demonstrates the effectiveness of the online adaptation approach. In addition, compared with the rough approach to choose the adaptation ROI adopted by OnAVOS [3], before CRF, this method is also 1.9% greater than it. Af-

Table 2
Ablation study on the contribution of individual RL model for the DAVIS 2016 dataset, measured by the mean region similarity J_m. WO indicates without.

Method	DAVIS 2016
WO adaptation	80.3 ± 0.4
foreground adaptation	82.1 ± 0.5
background adaptation	85.3 ± 0.5
full adaptation	86.5 ± 0.4
full adaptation + CRF	**87.1** ± 0.4
OnAVOS [3] W/O CRF	84.6 ± 0.6
OnAVOS [3]	85.7 ± 0.6
CINM [4]	84.2

Table 3
Ablation study on the contribution of individual RL model for the SegTrack V2 dataset, measured by the mean region similarity J_m. WO indicates without.

Method	SegTrackV2
WO adaptation	61.4 ± 0.6
foreground adaptation	66.2 ± 0.5
background adaptation	73.2 ± 0.5
full adaptation	76.6 ± 0.5
full adaptation + CRF	**77.5** ± 0.5
OnAVOS [3] W/O CRF	64.9 ± 0.6
OnAVOS [3]	66.7 ± 0.6
CINM [4]	77.1

ter executing CRF, the gain degrades a little to 1.4%, which demonstrates that the flexible way to choose the optimal adaptation ROI for each frame is significant for the final segmentation result. To study the individual influence of each RL model, the RL model is removed to choose the optimal adaptation ROI for background, obtaining the method **foreground adaptation**, and remove the model to choose the adaptation ROI for foreground, obtaining the method **background adaptation**. As can be observed from the Table 2, using **foreground adaptation** method and **background adaptation** method obtains J_m of 82.1% and 85.3% on DAVIS 2016 dataset, respectively, which indicates that both RL models improve the segmentation result while the RL to choose the optimal adaptation ROI for background makes larger contribution to the final segmentation result. This observation also explains the reason why there are more threshold candidates for background than foreground. The same ablation studies is also conducted on the Segtrack V2 dataset and obtain the similar results as can be observed from the Table 3. When using both foreground and background adaptation, the proposed method obtains J_m of 61.4% without CRF. J_m of this method with full adaptation is greater than the method without any adaptation by 15.2%. The comparison between the rough approach to choose the adaptation ROI adopted by OnAVOS [3] and the proposed method is conducted on Segtrack V2 dataset as well, before CRF. The result is that J_m of the proposed method is 11.7% greater than the baseline method, which is a dramatic improvement of the mean region similarity. After the process of CRF, the gain degrades a little to 10.8%, which is still a huge improvement.

To study the individual influence of each RL model on Segtrack V2 and Youtube-Object dataset, the method **foreground adaptation** and the method **background adaptation** are also adopted as on the DAVIS 2016 dataset. As can be observed from the Table 3 and Table 4, this method obtains 66.2% of J_m (**foreground adaptation**) and 73.2% of J_m (**background adaptation**) on Segtrack V2 dataset, and 78.2% of J_m (**foreground adaptation**) and 78.9% of J_m (**background adaptation**) on Youtube-Object dataset, which also indicates that both of the two RL models contribute to the improvement of segmentation result. This experiment also demonstrates the importance of the optimal adaptation ROI mining.

Table 4
Ablation study on the contribution of individual RL model for the Youtube-Object dataset, measured by the mean region similarity J_m. WO indicates without.

Method	Youtube-Object
WO adaptation	78.0 ± 0.4
foreground adaptation	78.2 ± 0.4
background adaptation	78.9 ± 0.4
full adaptation	79.0 ± 0.4
full adaptation + CRF	**79.5** ± 0.4
OnAVOS [3] W/O CRF	77.0 ± 0.6
OnAVOS [3]	77.4 ± 0.6
CINM [4]	78.4

Table 5
Performance comparison between heuristic adaptation ROIs selection and RL-based adaptation ROIs selection, conducted on the DAVIS 2016 dataset, measured by the mean region similarity J_m. t_n indicates the threshold to choose the adaptation ROI for background. t_p indicates the threshold to choose the adaptation ROI for foreground.

t_n	t_p	J_m
0.4	0.97	61.9 ± 0.6
0.4	0.7	61.2 ± 0.6
0.2	0.97	64.2 ± 0.6
0.2	0.7	62.5 ± 0.6
0.1	0.97	73.7 ± 0.6
0.1	0.7	69.6 ± 0.6
0.01	0.97	**83.6** ± 0.6
0.01	0.7	80.1 ± 0.6
OURS		**87.1** ± 0.4

Influence of Different Thresholds: The purpose of the second ablation experiment is to demonstrate that it is important to select different t_n and t_p for each specific frame, rather than adopting a particular fixed set of thresholds. In other words, no matter which set of t_n and t_p are chosen, as long as these thresholds are fixed for all frames, the final segmentation result will be worse than the result generated by adopting the optimal thresholds selected by the RL models for each specific frame. More specifically, the RL model to select the adaptation ROI for background has 4 candidate thresholds including {0.4, 0.2, 0.1, 0.01}. The RL model to select the adaptation ROI for foreground has 2 candidate thresholds including {0.97, 0.7}. In this way, there are totally 8 possible combinations of t_p and t_n, which are listed in Table 5. In this experiment, each combination of t_n and t_p is adopted as the fixed thresholds for the segmentation model adaptation and its corresponding result is evaluated and compared with the result generated by the proposed method. As can be observed from Table 5, among these combinations, the most "conservative" one, i.e. t_p=0.97 and t_n=0.01 performs best. Note that these values are different from the final result of OnAVOS [3] because OnAVOS selects the background adaptation ROI according to the distance, rather than the value of the probability map. The performance decreases dramatically when the value of the fixed threshold gets close to 0.5 because the error of segmentation will propagate quickly when adopting these "greedy" thresholds for all frames. The highest obtained J_m (83.6%) of the method adopting fixed thresholds, however, is still much lower than the result generated the proposed method (87.1%). According to the observation of this experiment, it is obvious that the improvement does result from the RL models that choose the optimal t_n and t_p for each individual frame, rather than the contribution of a certain fixed combination of t_n and t_p.

Quality of Adaptation ROI: The purpose of the third ablation experiment is to demonstrate that the proposed method is able to find a better adaptation ROI, compared with the baseline method. The metric to evaluate the selected adaptation ROI

Table 6
Quality comparison between the selected ROIs of the proposed method and the baseline method, as well as their influences on the final segmentation results, conducted on the DAVIS 2016 dataset, measured by the mean region similarity J_m. In this table, IOU indicates the IOU between the adaptation ROI selected and the ground-truth. J_m indicates the mean region similarity. B refers the background and F refers to the foreground. RL refers to the RL model and Base refers to the baseline method. Δ refers the difference value between the RL model and the baseline method. All values in this table are calculated before CRF.

Video	IOU-B-RL	IOU-B-Base	IOU-F-RL	IOU-F-Base	J_m-RL	J_m-Base	ΔJ_m
blackswan	99.4	44.6	91.0	74.6	95.4	95.4	0
bmx-trees	98.9	75.1	39.9	15.6	55.5	52.5	3.0
breakdance	97.4	55.9	74.9	55.0	80.2	68.4	11.8
camel	98.8	45.0	89.9	70.8	93.8	84.0	9.8
car-roundabout	99.4	41.8	95.2	85.4	97.1	97.1	0
car-shadow	99.7	60.1	92.7	79.2	96.1	96.0	0
cows	99.1	43.1	91.6	76.9	94.6	94.6	0
dance-twirl	98.1	59.3	83.3	63.2	87.3	84.6	2.7
dog	99.3	52.6	92.6	78.9	95.1	95.1	0
drift-chicane	99.6	72.1	83.0	55.6	89.1	87.2	1.8
drift-straight	99.1	64.4	90.1	75.8	92.7	91.3	1.3
goat	99.0	58.4	88.8	73.7	91.2	91.1	0
horsejump-high	98.9	61.8	80.0	52.4	87.3	86.8	0.5
kite-surf	98.3	72.4	57.2	28.0	66.7	66.7	0
libby	99.3	66.3	74.9	45.1	86.1	86.1	0
motocross-jump	93.9	41.2	87.5	70.7	90.4	86.4	4.0
paragliding	97.1	69.1	58.7	44.9	62.5	62.5	0
parkour	99.4	67.3	85.1	61.7	91.8	91.4	0.4
scooter-black	98.6	58.2	86.1	68.1	89.8	89.0	0.8
soapbox	98.3	46.1	86.0	65.4	90.1	86.2	3.8
mean	98.6	57.7	81.4	62.0	86.5	84.6	1.9

is the IOU between the adaptation ROI and the ground-truth. In this way, firstly, the IOU between the adaptation ROI selected by the RL model and ground-truth is calculated. Then, the IOU between the adaptation ROI selected by the baseline method and the ground-truth is calculated. Finally, these two IOUs are compared and their contributions to the final final segmentation result. As can be observed from Table 6, on average, the IOU of the proposed method is 40.9% and 19.4% higher than the baseline method, for background and foreground respectively. To further demonstrate the proposed method's superior performance against the baseline method, a non-parametric test, e.g., Wilcoxon test, is conducted on our method's IOU and the baseline method's IOU, for foreground and background respectively. The null hypothesis is that our method's IOU and the baseline method's IOU are identical populations. Then, to test this hypothesis, the Wilcoxon test is applied to do the comparison, and the p-values turn out to be 6.3e-8 for foreground, and 1.2e-4 for background. As both p-values are less than the 0.05 significance level, the null hypothesis is rejected, which fully demonstrate this method's improvement against the baseline method. This result also coincides with the facts reported in Tables 2 and 3 that background ROI adaptation using RL brings more performance gain than foreground ROI adaptation. Finally, as more correct pixels are adopted to update the segmentation model in the proposed method, the segmentation model is able to properly adjust itself to the change of the target in the video. It is also the reason why the proposed method can achieve improvement on the final segmentation result.

Influence of Different States: The fourth ablation experiment is to study the influence of different states for the RL model. The adopted state for the RL model also greatly affects the convergence difficulty of the training process as well as the final segmentation performance. Three different states are designed to study this factor. The first state is to feed the initial image (3 channels) and the temporary probability map (1 channel) into a train-from-scratch ConvNet without using pre-trained VGG model. This is because the input of the VGG model should be a 3-channel image. The second one is to feed the initial image (3 channels) concatenated with several mask channels to indicate different adaptation area, similarly, without the feature extracting of VGG model. The third one is the proposed one where a set of images with different masks are generated to indicate different adaptation areas. Feature of each image is extracted by the VGG model. Then, the combined feature will be fed into the RL model. When using the first two types of states, after full training, the RL model still cannot choose the optimal adaptation area for each frame, which demonstrates these states are not suitable for this task. The main reason might be that, the pre-trained VGG model is able to extract a better feature which contains more information of the original image. Also, the first two ways to indicate the different adaptation areas are not sufficiently explicit and discriminative.

Selection of Training Sample: The fifth ablation experiment is to study the influence of the selection of training samples. To generate various training samples, two random number lists are created, including list A (3,8,13,15,17,18,19,22,24,26,27,29,32,34,35,37,40,44,47,49) and list B (2,4,12,20,22,23,24,25,26,28,30,31,34,36,39,43,44,46,47,49). Firstly, video sequences in DAVIS 2016, whose video IDs are within list A, are viewed as evaluation set, and the remaining videos are viewed as training set. Trained in this setting, the proposed method obtains a higher J_m than OnAVOS [3] by 2.1% on DAVIS 2016, which is 84.5% and 82.4%, respectively. Then, video sequences in DAVIS 2016, whose video IDs are within list B, are viewed as evaluation set, and the remaining videos are viewed as training set. In this setting, the proposed method still performs better than OnAVOS [3], with the accuracy result of 81.6% and 80.2%, respectively. These results demonstrate the stability of the proposed method against the variation of training samples.

As can be concluded from the ablation studies, especially the third one, the reason why the proposed method outperforms the existing methods adopting online adaptation is that the RL models know how to choose the optimal ROIs using the current segmentation result, both for background and foreground areas, to update the segmentation model, which minimizes the error introduced into the segmentation model. In short, it can be called the error minimizing process for the prediction of the current frame. In this way, apart from the adopted online-learning method, the error minimization process can be naturally applied into other video-related tasks, where the result of the current frame will be

utilized to influence its performance for the remaining frames. For instance, in other VOS methods [13], the prediction result of the current frame will be combined with the next frame's information to compose the model input for the next frame. This method also can be improved using the error minimization process as well. In term of the video object tracking task, the prediction result of the current frame is fed into the memory module in [38] to adapt to the target's appearance changes. However, without the error minimizing process, the error of the prediction of each frame will be accumulated in the memory module, restricting its final performance. Therefore, the error minimizing process in the proposed method is a good assistant for most video-related tasks, and in the future, it is planned to extend this method to a more general model which can be applied to other video-related works directly.

6. Conclusion

In this paper, two RL models are proposed to choose the optimal adaptation ROIs for foreground and background individually. In this way, more correct pixels can be selected to update the segmentation network during the online adaptation process, which effectively prevents the error propagation and accumulation. To demonstrate the effectiveness of the proposed method, it is evaluated on three datasets including DAVIS 2016, SegTrack V2 and Youtube-Object. On DAVIS 2016 dataset and Youtube-Object dataset, the proposed method obtains the new state-of-the-art results, with J_m being 87.1% on DAVIS 2016 dataset, and 79.5% on Youtube-Object dataset. In addition, the tackled problem is a common challenge existing in many video-related tasks, and the core idea of the proposed method can also be adopted in other applications.

In spite of the above achievements, the current method still has several limitations. For example, with the current design, it is not sufficient flexible to adopt it in other video-related tasks. In future, we plan to replace the discrete-value actions with the continue-value actions (e.g. DDPG [39]) to enable the RL model to choose the optimal adaptation ROI in a more general and flexible way, though continue-value actions will make the training process more difficult. Secondly, as the backbone greatly influences the final performance, we plan to try some other backbones, e.g. neural architecture search [40], to further improve the performance.

Acknowledgements

The work was supported by National Natural Science Foundation of China under 61972323 and 61902022, and Key Program Special Fund in XJTLU under KSF-T-02, KSF-P-02.

References

[1] P. Liu, C. Liu, W. Zhao, X. Tang, Multi-level context-adaptive correlation tracking, Pattern Recognit. 87 (2019) 216–225.
[2] D.Y. Kim, B.-N. Vo, B.-T. Vo, M. Jeon, A labeled random finite set online multi-object tracker for video data, Pattern Recognit. 90 (2019) 377–389.
[3] P. Voigtlaender, B. Leibe, Online adaptation of convolutional neural networks for video object segmentation, arXiv:1706.09364 (2017).
[4] L. Bao, B. Wu, W. Liu, CNN in MRF: Video object segmentation via inference in a CNN-based higher-order spatio-temporal MRF, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 5977–5986.
[5] P. Tokmakov, K. Alahari, C. Schmid, Learning video object segmentation with visual memory, in: IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2663–2672.
[6] Y. Chen, J. Pont-Tuset, A. Montes, L. Van Gool, Blazingly fast video object segmentation with pixel-wise metric learning, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 1189–1198.
[7] Y. Wang, J. Liu, Y. Li, J. Fu, M. Xu, H. Lu, Hierarchically supervised deconvolutional network for semantic video segmentation, Pattern Recognit. 64 (2017) 437–445.
[8] H. Song, W. Wang, S. Zhao, J. Shen, K.-M. Lam, Pyramid dilated deeper ConvLSTM for video salient object detection, in: European Conference on Computer Vision (ECCV), 2018, pp. 715–731.
[9] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, L. Van Gool, One-shot video object segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 221–230.
[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248–255.
[11] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, A. Sorkine-Hornung, A benchmark dataset and evaluation methodology for video object segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 724–732.
[12] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, L. Van Gool, The 2017 davis challenge on video object segmentation, arXiv:1704.00675 (2017).
[13] J. Luiten, P. Voigtlaender, B. Leibe, PReMVOS: proposal-generation, refinement and merging for the davis challenge on video object segmentation, in: The 2018 DAVIS Challenge on Video Object Segmentation-CVPR Workshops, vol. 1, 2018, p. 6.
[14] S. Wug Oh, J.-Y. Lee, K. Sunkavalli, S. Joo Kim, Fast video object segmentation by reference-guided mask propagation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7376–7385.
[15] E. Park, A.C. Berg, Meta-tracker: Fast and robust online adaptation for visual object trackers, in: European Conference on Computer Vision (ECCV), 2018, pp. 569–585.
[16] V.R. Konda, J.N. Tsitsiklis, Actor-critic algorithms, in: Conference and Workshop on Neural Information Processing Systems (NIPS), 2000, pp. 1008–1014.
[17] J. Yang, Y. Zhang, R. Feng, T. Zhang, W. Fan, Deep reinforcement hashing with redundancy elimination for effective image retrieval, Pattern Recognit. (2019) 107116.
[18] R.S. Sutton, D.A. McAllester, S.P. Singh, Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, in: Conference and Workshop on Neural Information Processing Systems (NIPS), 2000, pp. 1057–1063.
[19] S. Yun, J. Choi, Y. Yoo, K. Yun, J. Young Choi, Action-decision networks for visual tracking with deep reinforcement learning, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2711–2720.
[20] C. Huang, S. Lucey, D. Ramanan, Learning policies for adaptive tracking with deep feature cascades, in: IEEE International Conference on Computer Vision (ICCV), 2017, pp. 105–114.
[21] X. Dong, J. Shen, W. Wang, Y. Liu, L. Shao, F. Porikli, Hyperparameter optimization for tracking with continuous deep q-learning, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 518–527.
[22] J. Han, L. Yang, D. Zhang, X. Chang, X. Liang, Reinforcement cutting-agent learning for video object segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 9080–9089.
[23] X. Dong, Y. Yan, M. Tan, Y. Yang, I.W. Tsang, Late fusion via subspace search with consistency preservation, IEEE Trans. Image Process. 28 (1) (2018) 518–528.
[24] X. Dong, L. Zheng, F. Ma, Y. Yang, D. Meng, Few-example object detection with model communication, IEEE Trans. Pattern Anal. Mach.Intell. 41 (7) (2018) 1641–1654.
[25] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556 (2014).
[26] C.L. Zhang, J. Wu, Improving CNN linear layers with power mean non-linearity, Pattern Recognit. 89 (2019) 12–21.
[27] M. Everingham, S.M.A. Eslami, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge, Int. J. Comput. Vis. 111 (1) (2015) 98–136.
[28] A. Khoreva, R. Benenson, E. Ilg, T. Brox, B. Schiele, Lucid data dreaming for object tracking, The DAVIS Challenge on Video Object Segmentation, 2017.
[29] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, A. Sorkine-Hornung, Learning video object segmentation from static images, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2663–2672.
[30] W. Wang, J. Shen, J. Xie, F. Porikli, Super-trajectory for video segmentation, in: IEEE International Conference on Computer Vision (ICCV), 2017, pp. 1671–1679.
[31] Y.-H. Tsai, M.-H. Yang, M.J. Black, Video segmentation via object flow, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3899–3908.
[32] A. Prest, C. Leistner, J. Civera, C. Schmid, V. Ferrari, Learning object class detectors from weakly annotated video, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 3282–3289.
[33] F. Li, T. Kim, A. Humayun, D. Tsai, J.M. Rehg, Video segmentation by tracking many figure-ground segments, in: IEEE International Conference on Computer Vision (ICCV), 2013, pp. 2192–2199.
[34] K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, L. Van Gool, Video object segmentation without temporal information, IEEE Trans. Pattern Anal. Mach.Intell. 41 (6) (2018) 1515–1530.
[35] N. Märki, F. Perazzi, O. Wang, A. Sorkine-Hornung, Bilateral space video segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 743–751.
[36] P. Krhenbhl, V. Koltun, Efficient inference in fully connected CRFs with gaussian edge potentials, in: Conference and Workshop on Neural Information Processing Systems (NIPS), 2011, pp. 109–117.
[37] F. Liu, G. Lin, C. Shen, CRF Learning with CNN features for image segmentation, Pattern Recognit. 48 (10) (2015) 2983–2992.
[38] T. Yang, A.B. Chan, Learning dynamic memory networks for object tracking, in: European Conference on Computer Vision (ECCV), 2018, pp. 152–167.

[39] Y. Duan, X. Chen, R. Houthooft, J. Schulman, P. Abbeel, Benchmarking deep reinforcement learning for continuous control, in: International Conference on Machine Learning (ICML), 2016, pp. 1329–1338.
[40] X. Dong, Y. Yang, Searching for a robust neural architecture in four GPU hours, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 1761–1770.

Mingjie Sun received the B.S. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, in 2016, and the M. degree from the Xi'dian University, in 2019. He is now a Ph.D. student in Xi'an Jiaotong Liverpool University. His current research in terest is video object segmentation and visual understanding.

Jimin Xiao received the B.S. and M.E. degrees from the Nanjing University of Posts and Telecommunications in 2004 and 2007, respectively, and the Ph.D. degree from the University of Liverpool in 2013. He is his research interests include image and video proc essing, computer vision, and deep learning.

Eng Gee Lim received the Ph.D. degree from the Universit y of Northumbia, in 2002. He is currently a Professor with the Department of Electrical and Eng ineering, Xian Jiaotong Liverpool University, Suzhou, China. His research interests include antennas, RF and radio propagation for wireless communications and systems.

Yanchun Xie received the B.Eng. from Suzhou University of Science And Technology in 2015. He is now a Ph.D. student in the Department of the Electr ical and Electr onic Engineering of the Xi'an Jiaotong Liverpool University, Suzhou, PR China. His current research interest is object tracking and visual understanding.

Jiashi Feng received the B.E. degree from the University of Science and Technology, Hefei, China, in 2007, and the Ph.D. degree from the National Univ ersity of Singapore, Singapore, in 2014. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, National University of Singapore.