Reinforcement Cutting-Agent Learning for Video Object Segmentation

Junwei Han[1], Le Yang[1], Dingwen Zhang[1,2]*, Xiaojun Chang[3], Xiaodan Liang[3]
[1]Northwestern Polytechincal University, [2]Xidian University, [3]Carnegie Mellon University
junweihan2010@gmail.com, nwpuyangle@gmail.com, zdw2006yyy@mail.nwpu.edu.cn
cxj273@gmail.com, xdliang328@gmail.com

Abstract

Video object segmentation is a fundamental yet challenging task in computer vision community. In this paper, we formulate this problem as a Markov Decision Process, where agents are learned to segment object regions under a deep reinforcement learning framework. Essentially, learning agents for segmentation is nontrivial as segmentation is a nearly continuous decision-making process, where the number of the involved agents (pixels or superpixels) and action steps from the seed (super)pixels to the whole object mask might be incredibly huge. To overcome this difficulty, this paper simplifies the learning of segmentation agents to the learning of a cutting-agent, which only has a limited number of action units and can converge in just a few action steps. The basic assumption is that object segmentation mainly relies on the interaction between object regions and their context. Thus, with an optimal object (box) region and context (box) region, we can obtain the desirable segmentation mask through further inference. Based on this assumption, we establish a novel reinforcement cutting-agent learning framework, where the cutting-agent consists of a cutting-policy network and a cutting-execution network. The former learns policies for deciding optimal object-context box pair, while the latter executes the cutting function based on the inferred object-context box pair. With the collaborative interaction between the two networks, our method can achieve the outperforming VOS performance on two public benchmarks, which demonstrates the rationality of our assumption as well as the effectiveness of the proposed learning framework.

1. Introduction

The video object segmentation (VOS) task [36, 37, 18] focuses on labeling each pixel as foreground or background in the given frame. It is the foundation of many vision tasks, such as scene understanding [8] and video surveillance [7, 4]. There are three kinds of VOS methods in general, including unsupervised VOS methods [29, 43], weakly supervised VOS methods [5, 11, 48], and semi-supervised VOS methods [1, 12, 16], respectively. This paper mainly focuses on the semi-supervised VOS.

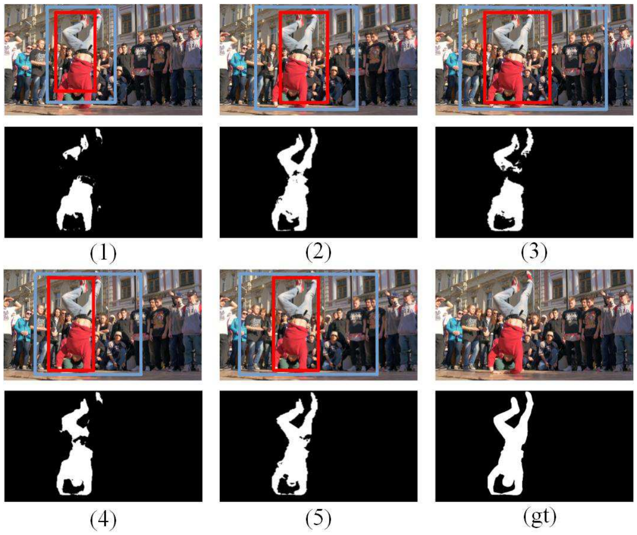

Figure 1. Segmentation masks. For an identical segmentation model, different object boxes (shown in red) and context boxes (shown in blue) can generate different segmentation masks. The *gt* indicates the ground truth segmentation mask for this frame.

One common choice for addressing the semi-supervised VOS problem is based on tracking or matching [1, 16, 43], where the core idea is to find the image (super)pixels that correspond to those in the previous (annotated) mask from the preceding video frame. However, this strategy is hard to obtain satisfactory performance in practice because in the unconstrained real-world scenarios, the blurry motion and heavy occlusion would destroy the matching results. Besides, the cluttered background and textureless object foreground would mislead the matching results. In this paper, we model this problem as a conditional decision-making process rather than the simple matching or tracking process, where one or more agents are employed to decide which (super)pixels are the corresponding foreground object con-

*Corresponding author.

ditional on the object box obtained from the previous frame.

According to [20], biological vision systems are believed to have a sequential process with changing retinal fixations that gradually accumulate evidence of certainty when searching or localizing objects. In this paper, we believe it does the same for segmenting objects. Thus, it is highly desirable, both biologically and computationally, to explore computational models that facilitate object segmentation in such manner. This finding enlightens us to explore a new pipeline of VOS by sequentially exploring for the desired object masks and considering their contextual dependency during the exploration process. To reach a robust decision-making process, instead of making such decisions heuristically, we propose to learn an optimal decision-making policy under the deep reinforcement learning (DRL) framework. Reinforcement learning has obtained successes in areas like robotics [17] and control [23], where real agents and environments are involved naturally. In the recent few years, with the rapid development of the deep learning technique, deep reinforcement learning emerges and shows promising results in many computer vision systems although the interpretation of their strategies that an agent interacts with an environment is not always so intuitive [19]. DRL has been studied for addressing some computer vision problems in object localization [2, 15, 22], tracking [9, 44], and pose estimation [19], while to our best knowledge, we make the first attempt to apply DRL for the video object segmentation problem.

Deep Reinforcement Learning (DRL) is good at making discrete choices about which action to execute, as it has been used in the existing control and computer vision systems. However, in the investigated VOS problem, directly using DRL under the conventional (super)pixel label assignment process to learn segmentation agents would result in a nearly continuous decision-making process. The number of the involved agents (pixels or superpixels) and action steps from the seed (super)pixels to the whole object mask might be incredibly huge, and thus would not obtain good performance. To overcome these difficulties, this paper simplifies the learning of segmentation agents to the learning of a cutting-agent, which only has limited number of action units and can converge in just a few action steps. Our intuition is that an identical segmentation model can generate significantly distinct object masks given different object and context boxes (see Figure 1). The segmentation of the video foreground object can be thus considered as the interaction between object regions and their contexts, where the object regions provide appearance priors for the object of interest, while the context regions provide the optimal contrast priors for discriminating the foreground object from its surrounding background. Based on the above observations, we assume that, with the optimal predicted object foreground (bounding box) region and the (bounding box) context region, the segmentation model is able to obtain perfect segmentation masks of the desired foreground objects. Based on this assumption, we establish a novel reinforcement cutting-agent learning framework, where the cutting-agent consists of a cutting-policy network (CPN) and a cutting-execution network (CEN). The former learns policies for deciding optimal object-context box pair, while the latter executes the cutting function based on the inferred object-context box pair.

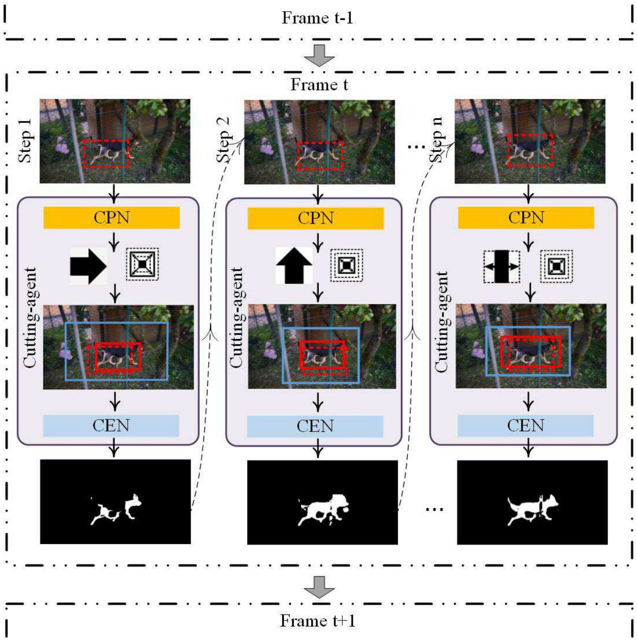

Figure 2. Framework of the proposed reinforcement cutting-agent learning approach for the VOS problem. At each step, CPN observes the current states, determines actions to adjust object box and context box. Then CEN generates the object mask of the refined box. After multi-step box refinements, our method can progressively improve the segmentation performance. The object-context box pairs are shown in red and blue respectively. The dotted line indicates the object box propagated from the segmentation mask of the former step.

Equipped with the cutting-agent, we build a novel VOS framework, as shown in Figure 2. Following the spirit of Perazzi's work[16] and Zhang's work [48], we proceed our approach on a per-frame basis by obtaining the object of interest in each frame guided by the output of the previous frame. The CPN and CEN are pre-trained on auxiliary static images and fine-tuned by using the annotated first frame of each video. The input of CPN is current frame feature, the current state, and the history actions. Then, CPN learns two-fold policies for deciding the optimal action sequence to achieve the appropriate foreground object location as well as its corresponding context for segmentation. CEN takes the predicted object-context box pair as its input and learns the segmentation-aware representations and discriminative cutting functions to separate the desired object from its context. During inference, given the annotated object mask of the first frame in each video sequence, CP-N localizes the foreground box in the second frame, which starts with the box location obtained in the first frame and then gradually reaches the predicted optimal object-context box pair on the second frame until the stop action is executed. The corresponding segmentation mask is obtained by performing CEN based on the predicted optimal object-context box pair. By repeating the aforementioned process, the proposed approach can finely segment the object of interest from each frame of given video sequences.

To sum up, this paper has three main contributions:

- We make the earliest attempt to solve the (semi-supervised) video object segmentation problem as a conditional decision-making process and build the first deep reinforcement learning based video object segmentation framework.

- We reveal the insight of formulating the video object segmentation problem as the inference based on the interaction between optimal object region and their context, resulting in the simple yet effective learning policies for deciding the optimal object-context box pair for video object segmentation.

- We implement a novel DRL-based VOS framework to learn a cutting-agent by collaborating the cutting-policy network and cutting-execution network. Comprehensive experiments have demonstrated the rationality of our assumption as well as the effectiveness of the proposed learning framework.

2. Related work

Video object segmentation: As an extensively studied area, the existing VOS method can be summarized as unsupervised methods [43], the weakly supervised methods [48] and the semi-supervised methods [1, 12, 16]. Essentially, as the weakly supervised or unsupervised methods cannot access pixel-level annotated training exemplars, they typically estimate informative cues, such as boundaries, motion, video saliency and object detection *etc*., then segment video object according to the estimated cues. In [43], Xiao and Lee first generate bounding box proposals for each frame, then regard these boxes as weak supervision to iteratively refine the segmentation masks.

As for semi-supervised VOS task, the learning methods can explore the annotated video frame and learn specific object pattern for the input video sequences. Especially, when applying the deep convnet to segment video object, researchers usually pre-train the convnet on auxiliary annotated data and fine-tune the convnet on the annotated video frames. Recently, Jampani *et al.* [12] propose a Video Propagation Network, which is capable of propagating information across video frames.

Different from the existing works, we formulate the VOS problem as a Markov Decision Process and dispose it from the view of DRL. This is an unknown and worth trying attempt, which may generate an effective VOS method.

Deep reinforcement learning: The reinforcement learning learns an agent to evaluate the impact of certain actions under particular states, and it is effective to optimize the sequential decision problems [40]. In [26], Mnih *et al.* applied a deep neural network as a function approximator to estimate the action-value function for reinforcement learning, resulting in the deep reinforcement learning (DRL) method. Afterwards, a series of approaches have been proposed to assist DRL, such as memory replay [26] and policy gradient [23] *etc*.

Recently, there has occurred some successful attempts to apply the DRL methods in computer vision tasks [3, 44]. In [3], Cao *et al.* apply the DRL method to face hallucination task, and sequentially discover image patches, which should be attached more attention and enhanced. For visual object tracking, Yun *et al.* [44] cast this problem to a decision-making process and apply the DRL method to sequentially move the bounding box, achieving accurate tracking results.

Although the existing works have shown that the DRL method is capable of appropriating global optimization for sequential decision tasks. Directly using it in the investigated VOS problem is still nontrivial. This mainly due to that segmentation is a nearly continuous decision making process, where the number of the involved agents and action steps might be incredibly huge. To this end, this work simplifies the learning of segmentation agents to the learning of a cutting-agent. The cutting-agent only has a limited number of action units and can converge in just a few action steps, making it practical to learn the cutting-agent in the DRL manner. Such simplification strategy makes us become the first to be able to implement VOS under the DRL framework and the proposed CPN-CEN framework is different from any existing DRL frameworks designed for object localization or tracking.

3. Reinforcement cutting-agent learning

We formulate the video object segmentation problem as a Markov Decision Process (MDP) and employ the CEN to sequentially segment the object of interest based on the object-context box pair inferred by the CPN. Essentially, the studied MDP is based on states $s \in \mathcal{S}$, object searching action $a^o \in \mathcal{A}^o$, context embedding action $a^c \in \mathcal{A}^c$, state transition function $s^{'} = T(s, a^o, a^c)$ and the reward function $r(s, s^{'})$.

Given a video sequence $\mathcal{V}$ with the segmentation mask m_1 in the first frame, the proposed method progressively processes each frame. The state s consists of the input frame

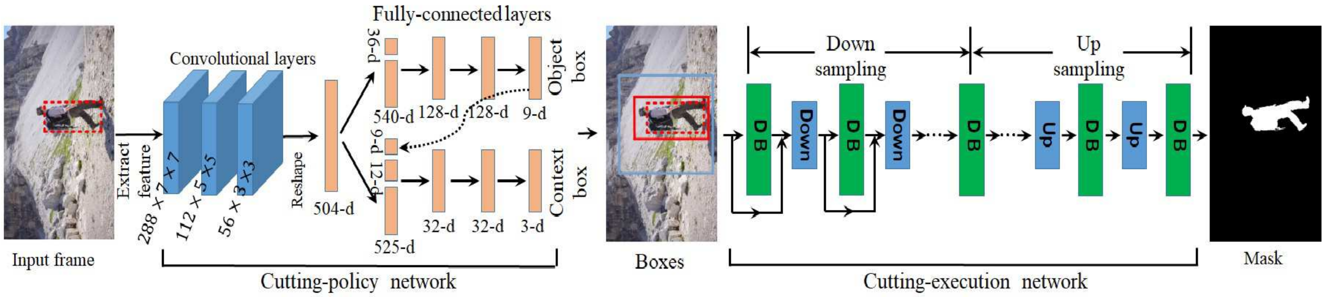

Figure 3. Network architecture of our reinforcement cutting-agent learning method. The cutting-policy network consists of two branches, determining the proper actions for object box and context box respectively. The cutting-execution network takes the architecture of FC-DenseNet [14]. It consists of down sampling path and up sampling path. DB indicates Dense Block (densely connected convolutional layers in a block), Down and Up are used to extract the feature representation and up sample the feature maps, respectively.

information and the action history. Specifically, considering the t^{th} video frame f_t, when the cutting-agent disposes of it for the k^{th} time, the cutting-agent observes the state $s_{t,k}$ and determines the object searching action $a^o_{t,k}$ and context embedding action $a^c_{t,k}$. These actions adjust the object box $b^o_{t,k}$ and context box $b^c_{t,k}$. Then the segmentation mask m_k and the corresponding reward r_k can be obtained. When the stop action is executed or the cutting agent reaches the maximal search steps, our method obtains the object box from the segmentation mask and propagates it to the next frame.

3.1. Agent actions

For an input frame f_t, we design a CPN to learn the expected cutting-agent, which determines action policies of an object searching action $a^o_{t,k}$ and a context embedding action $a^c_{t,k}$ according to the observed state $s_{t,k}$. The architecture of the CPN is shown in the left part of Figure 3. There are three convolutional layers in the front of the CPN, followed by two branches deciding the object searching action $a^o_{t,k}$ and the context embedding action $a^c_{t,k}$ respectively. As shown in Figure 4, the object searching action set $\mathcal{A}^o$ contains 9 kinds of actions, including 4 translation actions $\{$*Right, Down, Left, Up*$\}$, 4 scale change actions $\{$*Horizontal shrink, Vertical shrink, Horizontal zoom, Vertical zoom*$\}$ and 1 *Stop* action. The context embedding action set $\mathcal{A}^c$ consists of 3 actions, selecting the context box $b^c_{t,k}$ with three different magnitudes: $0.2, 0.4, 0.6$. The conventional DRL methods in computer vision task generally adopt a single pathway network architecture [9, 33, 44]. In contrast, the proposed CPN consists of two branches and can simultaneously determines the object searching action $a^o_{t,k}$ and the context embedding action $a^c_{t,k}$.

3.2. State and state transition

The state $s_{t,k}$ contains the current frame information $\mu_{t,k} \in \mathbb{R}^{288\times7\times7}$ and the object searching action history $\nu^o_{t,k}$, the context embedding action history $\nu^c_{t,k}$. The CEN forward propagates frame $f_{t,k}$, and extracts the frame information $\mu_{t,k}$ at the end of down sampling path.

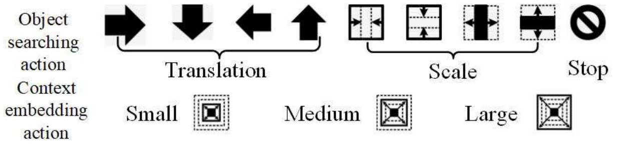

Figure 4. Object searching action set $\mathcal{A}^o$ and context embedding action set $\mathcal{A}^c$.

The action history is a vector, which tracks the past k actions. Each object searching action is represented as a 9-dimensional one-hot vector (or zero vector at the beginning of each frame). The context embedding action is defined similarly. We found $k = 4$ is a good choice, which results in $\nu^o_{t,k} \in \mathbb{R}^{36}$ and $\nu^c_{t,k} \in \mathbb{R}^{12}$. As for the context branch in the CPN (Figure 3), in addition to the feature vector from the convolutional layers, the context embedding action history vector, this branch also considers the object searching action $a^o_{t,k}$ under current state. In this way, the decision made by the context branch can be more rational and stable.

The state transition consists of the action execution function $\Psi^o(a^o_{t,k})$, $\Psi^c(a^c_{t,k})$ and the box channel transform function $\Phi(m_t)$. For object searching action $a^o_{t,k}$, $\Psi^o(a^o_{t,k})$ translates or scales the object box $b^o_{t,k}$ to a certain direction by a factor of 0.2 relative to its current size. As for context embedding action $a^c_{t,k}$, the agent decides four edges for the context box separately. Take the top edge for example, the method first measures the distance $D^u_{t,k}$ between the top edge of the object box $b^o_{t,k}$ and the frame boundary. Then the top edge of the context box is determined by:

$$\Delta^u = \beta D^u_{t,k} \tag{1}$$

where Δ^u is the distance of the top edges between the context box $b^c_{t,k}$ and object box $b^o_{t,k}$. $\beta \in \{0.2, 0.4, 0.6\}$ is the magnitude depending on the context embedding action $a^c_{t,k}$. The other three edges can be determined similarly.

Having obtained the object box $b^o_{t,k}$ and the context box $b^c_{t,k}$ after each inferring step, the CEN generates an object mask $m_{t,k}$, which passes through a box channel transform function $\Phi(m_{t,k})$. Precisely, as $m_{t,k}$ indicates the object box $b^o_{t,k+1}$, we expand the raw image from RGB channels to RGB + object box channels and obtain video frame $f_{t,k+1}$. The box channel indicates the object location. Pixels inside the object box $b^o_{t,k+1}$ are set to 255 and pixels outsides are set to 0.

3.3. Reward

The reward function $r(s_{t,k}, s_{t,k+1})$ reflects the positive / negative variation of the segmentation mask. As the target object varies smoothly among neighboring frames, for each video frame f_t, the initial object box $b^o_{t,1}$ is close to the real desired object box $b^o_{t,e}$. Consequently, in the cutting-agent, CPN only needs to interacts with CEN several times before reaching the optimal state. Considering the cutting-agent translates or scales the object box b^o with small magnitude, a reasonable action sequence would arouse small performance change (measured by the interaction over union, IoU) between neighboring states. To elaborately represent the segmentation mask variation, we define the reward function as following:

$$r(s_{t,k}, s_{t,k+1}) = \begin{cases} +\alpha \cdot 1, & \Delta > +0.1 \\ 10 \cdot \alpha \cdot \Delta, & -0.1 \leq \Delta \leq +0.1 \\ -\alpha \cdot 1, & \Delta < -0.1 \end{cases}$$

$$\text{where,} \qquad (2)$$

$$\Delta = IoU(m_{t,k+1}, y_t) - IoU(m_{t,k}, y_t)$$

$$\alpha = \begin{cases} 1, & a^o_{t,k} \neq stop \\ 3, & a^o_{t,k} = stop \end{cases}$$

where y_t is the ground truth of frame f_t.

From equation (2), we can observe that the reward $r(s_{t,k}, s_{t,k+1})$ would be $+\alpha$ or $-\alpha$ when the segmentation mask remarkably varies (the IoU value increases or decreases more than 0.1). Otherwise, the reward $r(s_{t,k}, s_{t,k+1})$ is linearly correlated with the IoU variation. Essentially, the designed reward function $r(s_{t,k}, s_{t,k+1})$ magnifies the variation for slight IoU change, thus, the cutting-agent can elaborately perceive the impact of each action pair $a^o_{t,k}$ and $a^c_{t,k}$, under current state $s_{t,k}$.

3.4. Deep Q-Learning

Given the current state $s_{t,k}$, the cutting-agent relies on the CPN to determine the object searching action $a^o_{t,k}$ and context embedding action $a^c_{t,k}$. As there is no exemplar actions $a^o_{t,k}$, $a^c_{t,k}$ under specific state $s_{t,k}$, we address the learning problem in the deep Q-learning manner [26]. We utilize CPN to approximate the underlying action-value function $Q^*(s, a^o, a^c)$. Precisely, the optimal action-value function $Q^*(s, a^o, a^c)$ is approximated by updating $Q(s, a^o, a^c)$ according to:

$$Q(s_{t,k}, a^o_{t,k}, a^c_{t,k}) = \begin{cases} r(s_{t,k}, s_{t,k+1}) \\ \quad + \gamma \max\limits_{a^o_{t,k+1}, a^c_{t,k+1}} Q(s_{t,k+1}, a^o_{t,k+1}, a^c_{t,k+1}), & a^o_{t,k} \neq stop \\ r(s_{t,k}, s_{t,k+1}), & a^o_{t,k} = stop \end{cases} \qquad (3)$$

where $r(s_{t,k}, s_{t,k+1})$ is the direct reward, calculated by equation (2). $Q(s_{t,k+1}, a^o_{t,k+1}, a^c_{t,k+1})$ is the reward for state $s_{t,k+1}$, when executing different action pairs $a^o_{t,k+1}$, $a^c_{t,k+1}$. γ is the discount factor, reflecting the connection between current state $s_{t,k}$, action pair $a^o_{t,k}$, $a^c_{t,k}$, and the future reward. As for the conventional DRL methods, the agent may explore thousands of steps before reaching the terminal target. Thus these methods preserve a large discount factor γ (*e.g.* $\gamma = 0.90$) to effectively propagate the terminal reward value to each intermediate state. In contrast, the proposed cutting-agent only needs to adopt the object box b^o_t and context box b^c_t within several steps. Consequently, we adopt a small discount factor $\gamma = 0.2$ in this work, making the action-value function $Q(s, a^o, a^c)$ rely more on the direct reward $r(s_{t,k}, s_{t,k+1})$.

4. DRL-based VOS

4.1. Train CEN

The proposed method follows the strategy of [30, 48] to learn video segmentation model from static images. Specifically, we begin with learning a CEN from static saliency detection datasets. The CEN takes the network architecture of the Fully-Convolutional DenseNet [14] and use FC-DenseNet56. This network can directly learn from the image segmentation data rather than fine-tunes a model pretrained on large scale data. The used saliency detection datasets include MSRA10K [6], PASCAL-S [21], SOD [27] and ECSSD [32]. As illustrated in Section 3.2, we expand the raw video frame from RGB channels to RGB + object box channel, indicating the accurate object location. The images are resized to 224 × 224 to fit the network input. Having trained the CEN with image saliency datasets, we fine-tune the model with the annotated first frame in the video sequence, so as to alleviate the difference between the saliency datasets and the video dataset. Former works [1, 16] generally learn the category-specific model for each video sequence via online fine-tuning. In contrast, we apply an identical model to tackle all sequences in the test dataset.

The training and fine-tuning process are based on data augmentation with random crops and vertical flips. The network parameters are optimized with RMSprop [35], the learning rate is set as $1e-4$ and exponentially decays with the factor 0.995 after each epoch. We monitor the mean

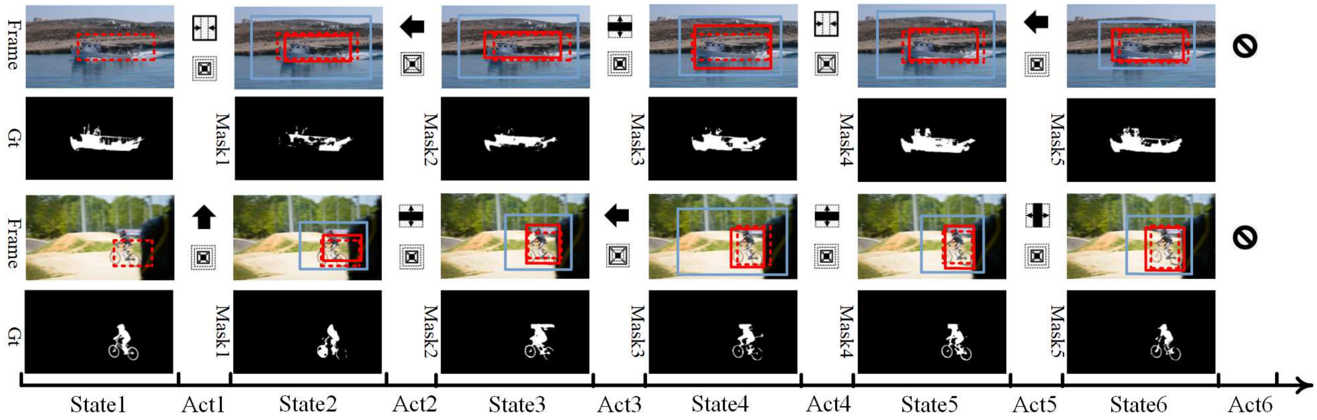

Figure 5. Example video frames and the corresponding states and actions. The displayed frames are from *boat* and *bmx-bumps* in the DAVIS dataset, respectively. The proposed method repeatedly adjusts object and context box, improves the segmentation mask step-by-step. The object box and context box are shown in red and blue respectively. The dotted line indicates the object box propagated from the former frame. Actions take the same meaning with those in Figure 4.

IoU and wait with the patience of 30 (15 for fine-tuning) epochs. After training and fine-tuning, we can obtain the CEN model. Essentially, the *FC-DenseNet56* is based on dense blocks, which consists of densely connected convolutional layers. *FC-DenseNet56* contains the down sampling path and the up sampling path, respectively in charge of extracting the feature representation and up sampling the feature maps. We extract the feature maps from the CEN at the end of the down sampling path, and regard it as the frame information $\mu_{t,k} \in \mathbb{R}^{288\times 7\times 7}$ used in Section 3.2.

4.2. Train CPN

Although we desire to train the CPN in the DRL manner, the only available training data is static saliency detection data rather than the annotated sequential video data. Thus, we add noise to the static saliency detection data so as to simulate the object location variation among neighbor video frames. In precise, before expanding the raw image from RGB channels to RGB + object box channel, we translate the object box b^o to a random direction with a magnitude of $\eta_t \in (0, 0.2)$ relative to it current size. Then we zoom or shrink the object box b^o to random orientation (horizon or vertical) with a magnitude of $\eta_s \in (0, 0.2)$. The transformed box $b^{'o}$ is regarded as the initial object box for current image. Given this noisy data, the CPN learns to properly translate or scale the object box, as well as select an appropriate context box b^c (illustrated in Section 3.1). Then we crop the image according to the context box b^c, resize it to 224 × 224, and feed it to the CEN. Afterwards, the mask produced by the CEN is restored to the original place, generating the segmentation mask and the reward signal (illustrated in Section 3.3). The CPN repeatedly interacts with CEN, updates the network parameter (illustrated in Section 3.4), and finally acquires abundant knowledge for action policies. After training CPN on static saliency detection datasets, we fine-tune it with the annotated first frame in the video sequence, so as to enhance the video specific object knowledge.

The CPN is optimized by the RMSprop [35], where the learning rate is set as $1e-4$ and decays exponentially. We adopt the $\epsilon-$greedy strategy [34] and exponentially anneal ϵ from 0.9 to 0.05. For the purpose of suppressing the correlation among training data, we utilize the experience replay mechanism [24] with the memory volume 5000.

4.3. Test with CPN and CEN

During testing, the CPN does not update network parameters or receives the reward. For the given video sequence with the annotated object mask of the first frame, the CPN adjusts the object box b^o and selects a proper context box b^c. Then our method crops the frame according to b^c and uses the CEN to generate the object mask. Collaborating with CEN, CPN can progressively approach the optimal foreground box and context box, and generates detailed segmentation masks in the end. By repeating this process, the proposed method can segment the object masks for the whole video sequence. Some examples of the testing process are shown in Figure 5.

5. Experiments

5.1. Experimental setup

Datasets. We evaluate the proposed learning framework on two widely used VOS datasets, *i.e*., the DAVIS dataset [30] and the YouTube-Objects dataset [10, 31]. The DAVIS dataset [30] consists of 50 high-quality videos with totally 3455 frames. It contains multiple common challenges for VOS task, *e.g*. motion blur, occlusions and appearance change *etc*. The YouTube-Objects dataset was initially built by Prest *et al*. [31], then Jain *et al*. [10] provided pixel-level

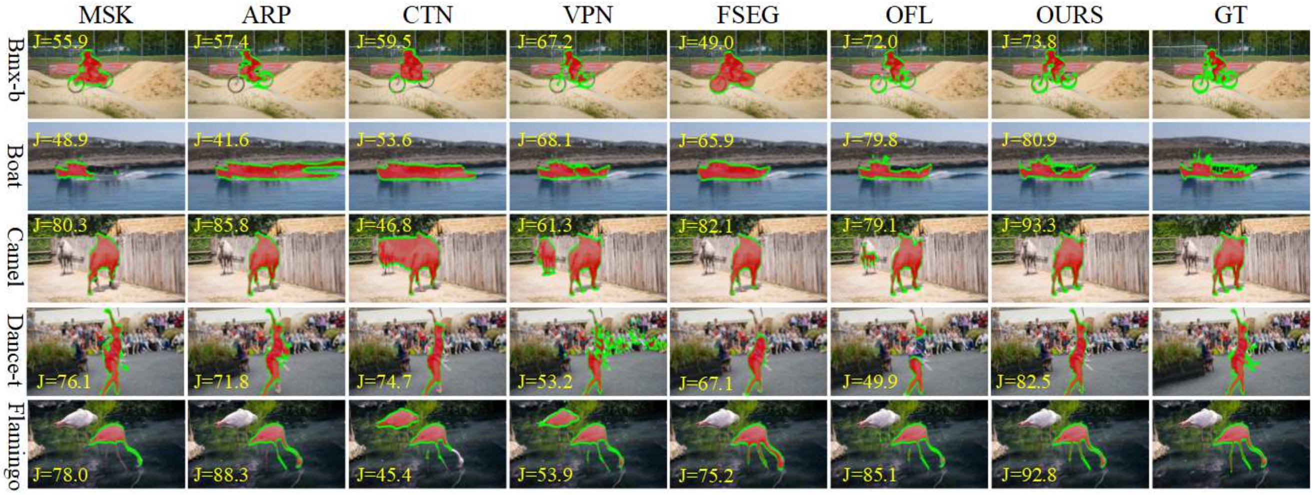

Figure 6. Visualization of segmentation masks for different methods on the DAVIS dataset. To be intuitive, we show the region similarity $\mathcal{J}$ on each frame.

Table 1. Quantitative results of the comparison methods on the DAVIS dataset, measured by the region similarity $\mathcal{J}$, counter accuracy $\mathcal{F}$ and temporal stability $\mathcal{T}$. *m* indicates mean, *r* indicates recall, *d* indicates decay.

	Jm↑	Jr↑	Jd↓	Fm↑	Fr↑	Fd↓	T↓
MSK[16]	80.3	93.5	8.9	75.8	88.2	9.5	18.9
ARP[18]	76.3	89.2	3.6	71.1	82.8	7.3	35.9
CTN[13]	75.5	89.0	14.4	71.4	84.8	14.0	**19.8**
VPN[12]	75.0	90.1	9.3	72.4	84.2	13.6	30.0
FSEG[11]	71.6	87.7	**1.7**	65.8	79.0	4.3	29.5
OFL[38]	71.1	80.0	22.7	67.9	78.0	24.0	22.4
LMP[36]	69.7	82.9	5.6	66.3	78.3	6.7	68.8
BVS[25]	66.5	76.4	26.0	65.6	77.4	23.6	31.7
OURS	**83.9**	**96.9**	5.7	**83.6**	**91.7**	**2.48**	26.8

Table 2. Comparison with OnAVOS on the DAVIS validation set.

	Jm↑	Jr↑	Jd↓	Fm↑	Fr↑	Fd↓	T↓
OnAVOS[39]	**86.1**	96.1	**5.2**	**84.9**	89.7	5.8	**19.0**
OURS	84.1	**97.0**	5.4	84.6	**92.3**	**3.7**	25.7

annotations for 126 video sequences. This dataset consists of 10 categories and more than 20000 frames.

Evaluation. On DAVIS, we follow [30] to simultaneously measure the region similarity $\mathcal{J}$, counter accuracy $\mathcal{F}$ and temporal stability $\mathcal{T}$ to present comprehensive analysis. Specifically, $\mathcal{J}$ is defined as the intersection-over-union. Given a segmentation mask m and the corresponding ground truth y, the region similarity $\mathcal{J}$ is calculated as $\mathcal{J} = \frac{|m \cap y|}{|m \cup y|}$. The counter accuracy $\mathcal{F}$ adopts F-measure to measure the trade-off between counter-based precision P_c and recall R_c. Specifically, it is calculated as $\mathcal{F} = \frac{2P_c R_c}{P_c + R_c}$. The temporal stability $\mathcal{T}$ compensates motion and small deformation. It simultaneously reveals oscillations and inaccuracies of the contours. We calculate it on a subset of DAVIS sequences by following [30]. On the YouTube-Objects dataset, we measure $\mathcal{J}$ in comparison.

Table 3. Quantitative results of the comparison methods on the YouTube-Objects dataset, measured by the region similarity $\mathcal{J}$.

Category	OFL	MSK	JFS	BVS	SCF	OURS
aeroplane	**89.9**	84.5	89.0	86.8	86.3	85.2
bird	84.2	83.7	81.6	80.9	81.0	**86.8**
boat	74.0	77.4	74.2	65.1	68.6	**79.9**
car	**80.9**	64.0	70.9	68.7	69.4	67.2
cat	68.3	69.8	67.7	55.9	58.9	**74.6**
cow	**79.8**	76.7	79.1	69.9	68.6	74.6
dog	76.6	74.5	70.3	68.5	61.8	**82.7**
horse	72.6	64.1	67.8	58.9	54.0	**73.6**
motorbike	48.1	**89.2**	61.5	60.5	60.9	73.7
train	76.3	74.4	78.2	65.2	66.3	**83.0**
Mean	77.6	71.7	74.0	68.0	67.6	**78.1**

5.2. Quantitative and qualitative comparisons

In this section, we compare the proposed method with state-of-the-art VOS methods on two benchmark datasets. On the DAVIS dataset, we compare with MSK [16], ARP [18], CTN [13], VPN [12], FSEG [11], OFL [38], LMP [36], BVS [25]. All of these methods are evaluated on the 50 video sequences on the complete DAVIS dataset. Table 1 summarizes the quantitative results of each method. It is encouraging to observe that out method consistently outperforms the existing state-of-the-art methods under three measures. Compared with the most competitive method MSK [16], our method improves mean $\mathcal{J}$ and mean $\mathcal{F}$ (which are the higher the better) by 2.6% and 7.8% respectively, and decreases the mean $\mathcal{T}$ (which is the lower the better) by 1.1%. This quantitative result demonstrates the effectiveness of the proposed framework. In addition, we also compared the proposed approach with OnAVOS [39] on the DAVIS validation set. The comparison results are shown in Table 2. Figure 6 shows the qualitative segmentation masks for different approaches.

On the YouTube-Objects dataset, we compare with the recent state-of-the-art methods, including OFL [38], MSK[16], JFS[28], BVS[25] SCF[10], and OnAVOS [39]. Among these methods, OnAVOS [39] can achieve 77.4 in terms of the mean region similarity $\mathcal{J}$. While the quantitative comparison results (in terms of the mean region similarity $\mathcal{J}$) of other methods are reported in Table 3. These results demonstrate the effectiveness of the proposed method.

5.3. Ablation studies

To investigate the impact of each component, we conduct the following ablation studies on the DAVIS dataset.

Channel expanding and frame cropping: In our framework, there are two factors that may effect our final results. We first study the frame cropping operation. To evaluate this factor, we implement two baselines. The first one is named as **RGB**, which trains CEN with three channel-input images and tests on each input video frame independently. The second one is named as **RGB+C**. It trains CEN by using the same strategy as the **RGB** baseline. However, in the testing phase, it performs segmentation on the cropped image region that encloses the object box (obtained in the previous frame) with context radio 0.4. In order to evaluate the channel expanding factor, we add an additional channel, i.e., the box channel, into the aforementioned two baselines, which forms the **RGBB** baseline and **RGBB+C** baseline, respectively.

Table 4 shows the performance of the aforementioned baselines, where we use the mean region similarity $\mathcal{J}$ for analysis. As can be observed, **RGBB** exceeds **RGB** by 6.4%, **RGBB+C** exceeds **RGB+C** by 5.8%. This indicates that the object box channel is important in our framework. Besides, the performance gap between **RGB+C** and **RGB** is 3.5%, the performance gap between **RGBB+C** and **RGBB** is 3.1%. This demonstrates the necessity of considering the information from the former frame. Although **RGBB+C** performs good, its performance is 7.1% lower than our complete approach. To our best knowledge, there are two reasons for this performance drop, $1)$the object box propagated from the former frame is inaccurate and misleading, $2)$ the fixed zoom factor between context box b_t^c and object box b_t^o cannot adapt to the various video frames. From these results, we can conclude that the information cues provided by the object boxes and the corresponding context boxes are beneficial for object segmentation.

Object searching action and context embedding action: To study the influence of agent actions, we alternatively remove the object searching action a_t^o and context embedding action a_t^c from our complete approach. Specifically, we first use the cutting-agent to obtain context box b_t^c and fill the entire object channel to 255, obtaining **OURS-O** method. Then, we apply the cutting-agent to adjust the object box b_t^o. We zoom b_t^o with a factor $\alpha = 0.4$ to obtain context box b_t^c. This method is named as **OURS-C**.

Table 4. Ablation studies of the proposed method on DAVIS dataset, measured by the region similarity $\mathcal{J}$, counter accuracy $\mathcal{F}$ and temporal stability $\mathcal{T}$. *m indicates mean, *r indicates recall, *d indicates decay.

	Jm↑	Jr↑	Jd↓	Fm↑	Fr↑	Fd↓	T↓
RGB	67.5	77.4	25.0	66.6	78.4	22.6	32.7
RGB+C	71.0	83.8	11.6	66.9	70.1	12.8	33.1
RGBB	73.9	87.9	14.9	69.7	80.6	12.5	22.4
RGBB+C	76.8	91.4	11.7	76.6	86.6	10.1	20.1
OURS	83.9	96.9	5.7	83.6	91.7	2.5	17.8
OURS-O	78.6	91.7	9.5	74.5	86.0	10.1	20.3
OURS-C	80.7	93.8	8.2	75.8	88.2	9.4	19.2
OURS-F	78.7	95.5	6.8	77.5	**94.9**	12.8	23.8
NetD	**85.4**	**97.3**	**5.2**	**84.5**	92.3	**2.3**	**15.7**

From the results in Table 4, we can observe that compared with our complete approach, using **OURS-O** and **OURS-C** would obtain 5.3% and 3.2% performance drop, respectively. This demonstrates that both precise object (box) region and context (box) region are critical for performing high-quality object segmentation, which also validates our assumption that object segmentation is essentially a kind of interaction between the object region and its context.

Fine-tune and deeper network: To evaluate the effect of the fine-tuning process, we first test the performance of removing the fine-tuning process from the complete method, which obtains the **OURS-F** baseline. In addition, to study the influence of using deeper base network architecture in our framework, we adopt the FC-DenseNet67 [14] as the architecture of CEN, which forms the **NetD** baseline.

As shown in in Table 4, there is a decrease of 5.2% between **OURS** and **OURS-F**, which reflects the impact of fine-tuning on the annotated video frames. In addition, the **NetD** brings 1.5% improvement over **OURS**, revealing that a deeper network architecture for CEN can further improve the segmentation results of our approach.

6. Conclusion

In this paper, we make a pioneer effort to formulate the video object segmentation problem as a Markov Decision Process and propose a novel reinforcement cutting-agent learning framework to tackle this problem. With the successive cooperation of the cutting-policy network and the cutting-execution network, the proposed method can segment out the target object with the interaction between the predicted object box and the context region. Comprehensive experiments on two benchmark datasets demonstrate the effectiveness of the proposed method. In future, we plan to deploy this method into some vision tasks, such as semantic segmentation [41, 42], object localization [47], saliency estimation [46], and 3D shape learning [45], .

Acknowledgement: This work was supported in part by the National Science Foundation of China under Grant 61473231, Grant 61522207, Grant 61373114, and Grant 61661166011.

References

[1] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *CVPR*, 2017.

[2] J. C. Caicedo and S. Lazebnik. Active object localization with deep reinforcement learning. In *ICCV*, 2015.

[3] Q. Cao, L. Lin, Y. Shi, X. Liang, and G. Li. Attention-aware face hallucination via deep reinforcement learning. In *CVPR*, 2017.

[4] D. Cheng, X. Chang, L. Liu, A. G. Hauptmann, Y. Gong, and N. Zheng. Discriminative dictionary learning with ranking metric embedded for person re-identification. In *IJCAI*, 2017.

[5] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In *ICCV*, 2017.

[6] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu. Global contrast based salient region detection. *TPAMI*, 2015.

[7] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, et al. A system for video surveillance and monitoring. *VSAM final report*, 2000.

[8] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[9] C. Huang, S. Lucey, and D. Ramanan. Learning policies for adaptive tracking with deep feature cascades. In *ICCV*, 2017.

[10] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, 2014.

[11] S. D. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. In *CVPR*, 2017.

[12] V. Jampani, R. Gadde, and P. V. Gehler. Video propagation networks. In *CVPR*, 2017.

[13] W.-D. Jang and C.-S. Kim. Online video object segmentation via convolutional trident network. In *CVPR*, 2017.

[14] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *CVPRW*, 2017.

[15] Z. Jie, X. Liang, J. Feng, X. Jin, W. Lu, and S. Yan. Tree-structured reinforcement learning for sequential object localization. In *NIPS*, 2016.

[16] A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017.

[17] J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *IJRR*, 2013.

[18] Y. J. Koh and C.-S. Kim. Primary object segmentation in videos based on region augmentation and reduction. In *CVPR*, 2017.

[19] A. Krull, E. Brachmann, S. Nowozin, F. Michel, J. Shotton, and C. Rother. Poseagent: Budget-constrained 6d object pose estimation via reinforcement learning. In *CVPR*, 2017.

[20] H. Larochelle and G. E. Hinton. Learning to combine foveal glimpses with a third-order boltzmann machine. In *NIPS*, 2010.

[21] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *CVPR*, 2014.

[22] X. Liang, L. Lee, and E. P. Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *CVPR*, 2017.

[23] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. In *ICLR*, 2016.

[24] L.-J. Lin. Reinforcement learning for robots using neural networks. Technical report, Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, 1993.

[25] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, 2016.

[26] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. In *NIPS*, 2013.

[27] V. Movahedi and J. H. Elder. Design and perceptual validation of performance measures for salient object segmentation. In *CVPRW*, 2010.

[28] N. S. Nagaraja, F. R. Schmidt, and T. Brox. Video segmentation with just a few strokes. In *ICCV*, 2016.

[29] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013.

[30] F. Perazzi, J. Ponttuset, B. Mcwilliams, L. Van Gool, M. H. Gross, and A. Sorkinehornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

[31] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012.

[32] J. Shi, Q. Yan, L. Xu, and J. Jia. Hierarchical image saliency detection on extended cssd. *TPAMI*, 2016.

[33] J. S. Supancic III and D. Ramanan. Tracking as online decision-making: Learning a policy from streaming videos with reinforcement learning. In *ICCV*, 2017.

[34] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

[35] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 2012.

[36] P. Tokmakov, K. Alahari, and C. Schmid. Learning motion patterns in videos. In *CVPR*, 2017.

[37] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. 2017.

[38] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *CVPR*, 2016.

[39] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. 2017.

[40] C. J. Watkins and P. Dayan. Q-learning. *Machine learning*.

[41] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *CVPR*, 2017.

[42] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, J. Feng, Y. Zhao, and S. Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *TPAMI*, 39(11):2314–2320, 2017.

[43] F. Xiao and Y. Jae Lee. Track and segment: An iterative unsupervised approach for video object proposals. In *CVPR*, 2016.

[44] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *CVPR*, 2017.

[45] D. Zhang, J. Han, Y. Yang, and D. Huang. Learning category-specific 3d shape models from weakly labeled 2d images. In *CVPR*, 2017.

[46] D. Zhang, J. Han, and Y. Zhang. Supervision by fusion: towards unsupervised learning of deep salient object detector. In *ICCV*, 2017.

[47] D. Zhang, D. Meng, L. Zhao, and J. Han. Bridging saliency detection to weakly supervised object detection based on self-paced curriculum learning. In *IJCAI*, 2016.

[48] D. Zhang, L. Yang, D. Meng, D. Xu, and J. Han. Spftn: A self-paced fine-tuning network for segmenting objects in weakly labelled videos. In *CVPR*, 2017.