Interactive Video Object Segmentation via Spatio-temporal Context Aggregation and Online Learning

Zihang Lin, Jiafeng Xie, Chuankai Zhou, Jian-Fang Hu and Wei-Shi Zheng
Sun Yat-sen University, China
{linzh59, xiejf6, zhouchk3}@mail2.sysu.edu.cn hujf5@mail.sysu.edu.cn wszheng@ieee.org

Abstract

In this work, we propose a novel network for interactive video object segmentation which mainly consists of a feature extraction module, a spatio-temporal RNN module and an online adaption module. Our approach exploits the spatio-temporal context with a spatio-temporal RNN which is placed on top of a pre-trained CNN network. We also propose an online adaption module which utilizes the user's inputs(scribbles indicating the objects of interest) to tune the network parameters for segmenting specific objects. The online adaption module can be optimized efficiently with closed-form solution so our approach can segment objects very fast. Our method achieved the third place on the interactive track of DAVIS Challenge 2019.

1. Introduction

Video object segmentation, which needs to segment objects in a pixel-level over a sequence of video frames, is important for video editing and it has drawn more and more attention in these years. In this problem, mask or scribble annotations indicating the objects of interest are assumed to be given in some frames.

The main challenge for this problem is to handle the large appearance change caused by motions, occlusion, deformation and interaction with other objects. Recent researches[13, 11] show that exploring useful contextual information is an effective way to confront this challenge. In this paper, we propose a novel spatio-temporal RNN which exploits useful spatio-temporal contexts for video object segmentation. In our spatio-temporal RNN, the local context is explicitly propagated to the neighboring image regions along both spatial and temporal directions. We develop a spatio-temporal RNN layer that injects spatio-temporal context into the learned feature maps, and it can be architecturally integrated with existing convolutional networks to construct an end-to-end network.

In this work, we consider the interactive setting described in [2]. In interactive scenario, a "user" gives scribble annotations to the algorithm to indicate the objects of interest in the first interaction and adds new scribble annotations on wrongly segmented regions to guide the algorithm to refine the segmentation results in the following interactions. It is challenging for the algorithm to understand the user's intension. To address this challenge, we develop an online adaption module to utilize the scribble annotations provided by the "user" to update the parameters in the network for segmenting specific objects. We formulate our online adaption module as an unconstrained quadratic problem, so that it can be optimized efficiently with closed-form solution.

Overall, our main contributions are: (i) a novel spatio-temporal RNN for aggregating spatial and temporal context simultaneously; (ii) an online adaption module for adapting the learned model for segmenting specific objects efficiently; and (iii) based on the spatio-temporal RNN and online adaption, a novel video object segmentation system for segmenting objects at a fast speed.

2. Method

In this work, we treat the video object segmentation problem as a per-pixel classification task like most semantic segmentation models [3, 7], and we present a novel framework which explicitly explores the spatial and temporal contextual information. Our framework mainly consists of three modules: (i) a feature extraction module for extracting basic visual feature from each individual frame; (ii) a spatio-temporal RNN module for aggregating dense local context in both spatial and temporal directions; and (iii) an online adaption module for updating the parameters of the classification layer, referring to a 1×1 convolution layer following the spatio-temporal RNN. The overall architecture of our framework is summarized in Figure 1. In the following, we describe each module in detail.

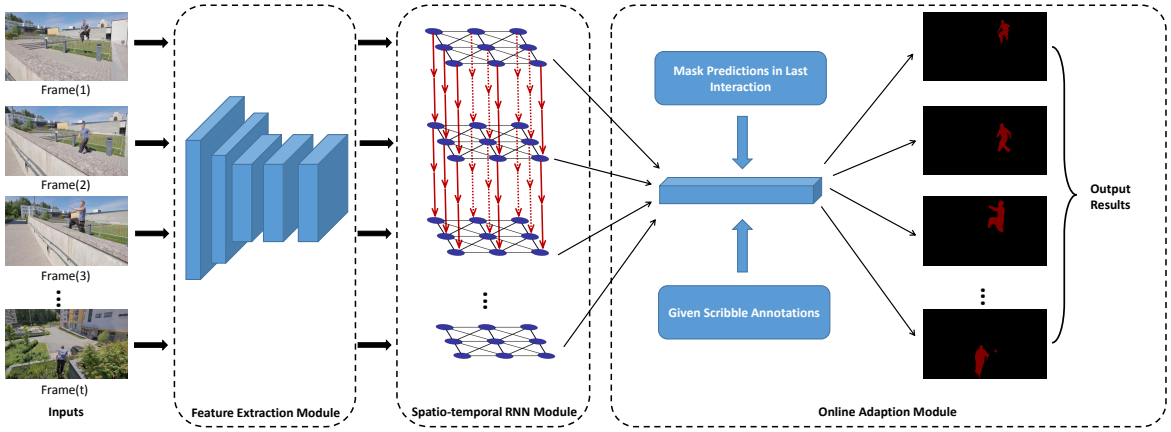

Figure 1. An overview of the proposed framework. Our framework consists of three modules: a feature extraction module with several convolution layers for extracting feature maps from each frame; a spatio-temporal RNN module for embedding spatio-temporal contextual information into the learned feature maps; and finally, an online adaption module to adapt a 1×1 convolution layer online for segmenting specific objects in given video.

2.1. Feature Extraction Module

Our feature extraction module is employed to extract basic visual features from each video frame. Here, we follow PML[5] to use a pre-trained segmentation network[1] without the classification layer as our feature extraction module.

2.2. Spatio-temporal RNN Module

Following the feature extraction module, we develop a spatio-temporal RNN module to exploit useful spatio-temporal contexts in video segmentation. Our spatio-temporal RNN is inspired by the DAG-RNN [10], which models the spatial contextual dependencies among image regions for scene segmentation. It is assumed that the neighboring elements interact with each other, which forms an undirected cyclic graph (UCG). As illustrated in Figure 2, the UCG $\mathcal{G}^{\mathcal{U}}$ is then topologically approximated by 4 directed acyclic graphs (DAGs) as $\mathcal{G}^{\mathcal{U}} = \{\mathcal{G}_{se}, \mathcal{G}_{sw}, \mathcal{G}_{ne}, \mathcal{G}_{nw}\}$. Finally, they capture the spatial dependencies via a vanilla-RNN, which can be mathematically formulated as follows:

$$\mathbf{h}_d^{v_i} = f\left(\mathbf{U}_d \mathbf{x}^{v_i} + \sum_{v_j \in \mathcal{P}_{\mathcal{G}_d}(v_i)} \mathbf{K}_d \mathbf{h}_d^{v_j} + \mathbf{b}_d\right),$$
$$\mathbf{y}^{v_i} = \sum_{d=1}^{|\mathcal{G}^{\mathcal{U}}|} \mathbf{V}_d \mathbf{h}_d^{v_i} + \mathbf{c}, \tag{1}$$

where $\mathbf{x}^{v_i}$ and $\mathbf{y}^{v_i}$ indicate the input and output features for vertex[2] v_i, respectively. $\mathbf{h}_d^{v_i}$ represents the corresponding hidden states. $\mathbf{K}_d$ is the model parameter controlling the information transformation among the vertexes of the graph with varied spatial locations. $\mathbf{U}_d$ and $\mathbf{V}_d$ are parameters to transform the input features and hidden states. $\mathbf{b}_d$ and $\mathbf{c}$ are the bias terms. $\mathcal{P}_{\mathcal{G}_d}(v_i)$ denotes the direct predecessor set of vertex v_i in DAG graph $\mathcal{G}_d$. f is an activation function.

From Equation (1), we can observe that the output feature maps, after a DAG-RNN, encode rich set of neighborhood spatial contexts. However, it does not consider the temporal relationship of the image units, which makes it less effective for segmenting objects in videos. Here, we develop a spatio-temporal RNN by additionally adding connections along temporal direction to each DAG in $\mathcal{G}^{\mathcal{U}} = \{\mathcal{G}_{se}, \mathcal{G}_{sw}, \mathcal{G}_{ne}, \mathcal{G}_{nw}\}$, please refer to Figure 3 for an example. In this way, our spatio-temporal RNN is able to aggregate both the spatial and temporal contexts among image regions. Specifically, we formulated our spatial-temporal RNN as:

$$\mathbf{h}_d^{v_{i,t}} = f\left(\mathbf{U}_d \mathbf{x}^{v_{i,t}} + \mathbf{K}\mathbf{h}_d^{v_{i,t-1}} + \sum_{v_{j,t} \in \mathcal{P}_{\mathcal{G}_d}(v_{i,t})} \mathbf{K}_d \mathbf{h}_d^{v_{j,t}} + \mathbf{b}_d\right),$$
$$\mathbf{y}^{v_{i,t}} = \sum_{d=1}^{|\mathcal{G}^{\mathcal{U}}|} \mathbf{V}_d \mathbf{h}_d^{v_{i,t}} + \mathbf{c}, \tag{2}$$

where $\mathbf{h}_d^{v_{i,t}}$ indicates the hidden state for the i-th vertex in the t-th frame. $\mathbf{K}_d$ and $\mathbf{K}$ are matrices encoding the information propagation along the spatial and temporal directions, respectively. $\mathbf{c}$ and $\mathbf{b}_d$ are the bias terms. $\mathbf{x}^{v_{i,t}}$

[1] In the challenge, we trained two models with different feature extraction backbone and averaged the prediction score. Deeplab-v2[3] with a ResNet-101 backbone and Deeplab-v3+[4] with an Xception-65 backbone are selected as feature extraction backbones and they are both pre-trained on COCO[6].

[2] Each vertex corresponds to a pixel on the feature map.

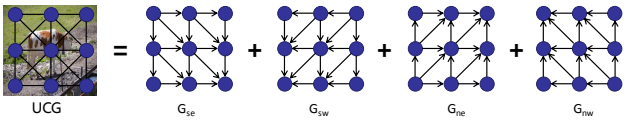

Figure 2. The UCG is decomposed into 4 DAGs with different propagation directions: southeast, southwest, northeast, northwest in [10].

and $\mathbf{y}^{v_{i,t}}$ are the corresponding input and output features for vertex $v_{i,t}$, which form our input and output feature maps.

2.3. Online Adaption Module

Since the spatio-temporal RNN and feature extraction modules can only learn some object-independent spatio-temporal information from the training videos, they can not be directly used to segment specific objects in given videos. Recent researches [1, 8] show that tuning the learned model for each specific video can obtain a better segmentation performance. However, their methods need to update all the parameters of the deep network, which often takes a lot of time. This makes their approaches less applicable in real-world applications. In interactive video object segmentation, it is essential to efficiently utilize the annotated scribbles provided by the user to adapt the model to achieve better segmentation results in a short time.

Here, we tackle the video object segmentation as a pixel-wise classification problem and place a classification layer on top of the spatio-temporal RNN module. The classification layer is defined as a 1×1 convolution layer. During interactions, we only tune the classification layer and fix the parameters of the feature extraction and spatio-temporal RNN modules. We refer this procedure as online adaption module. Our adaption module can be optimized efficiently with closed-form solutions and thus it is suitable in the interactive scenario.

Given annotated scribbles, we tune the parameters in the 1×1 convolution layer (denoted by $\mathbf{W}$) by solving the following optimization problem:

$$\begin{aligned} \mathbf{W}^* &= \arg\min_{\mathbf{W}} [L(\mathbf{W}) + \lambda \|\mathbf{W}\|_F^2], \\ L(\mathbf{W}) &= L_+(\mathbf{W}) + L_-(\mathbf{W}), \end{aligned} \tag{3}$$

where $L_+(\mathbf{W}) = \|\mathbf{W}^T\mathbf{X}^+ - \mathbf{Y}^+\|_F^2$ and $L_-(\mathbf{W}) = \|\mathbf{W}^T\mathbf{X}^- - \mathbf{Y}^-\|_F^2$ are two regression losses. They are employed to regress the selected positive and negative samples. $\mathbf{X}^+$ and $\mathbf{X}^-$ indicate the features extracted from the selected positive (objects of interest) and negative (background) samples, respectively. $\mathbf{Y}^+$ and $\mathbf{Y}^-$ indicate the corresponding label information in one-hot format. $\lambda\|\mathbf{W}\|_F^2$ is a regularization term.

This is an unconstrained convex minimization problem, whose solution is given by: $\mathbf{W}^* = (\mathbf{X}^+\mathbf{X}^{+^T} + \mathbf{X}^-\mathbf{X}^{-^T} + \lambda\mathbf{I})^{-1}(\mathbf{X}^+\mathbf{Y}^{+^T} + \mathbf{X}^-\mathbf{Y}^{-^T})$.

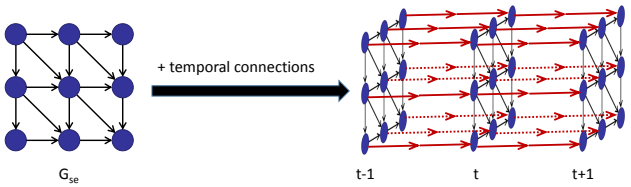

Figure 3. Comparison of the DAG employed in DAG-RNN [10] (left) and our spatio-temporal RNN (right).

In the interactive scenario, some scribbles of the objects to segment are provided in the first interaction in one frame. We select the pixels on the scribbles as positive(foreground) samples. Then we compute the tight bounding boxes of the scribbles and then expand them by 1.2 times. We randomly select some pixels(in the annotated frame) outside all the bounding boxes as negative(background) samples. And in the following interactions, when new scribble annotations(both foreground and background can be annotated in these interactions) are provided, we add the pixels on the newly provided scribbles to the corresponding samples set(positive or negative). And we also randomly select some other pixels to expand the negative set. These selected pixels must satisfy these conditions: (i) in the newly annotated frame; (ii) outside the bounding boxes of the new scribbles; and (iii) predicted as background in the last interaction. We ensure that the amount of the selected foreground pixels and background pixels are equal.

2.4. Model Learning

Here, we describe how to determine the model parameters using the given training set and test videos. In the training stage, our model intends to learn some object-independent spatio-temporal cues that is useful for video object segmentation. While in the test stage, the model needs to learn some object-specific information for segmenting certain objects.

2.4.1 Model Training.

Here, we mainly train the feature extraction module and spatio-temporal RNN from the provided training videos. We first pre-trained the model on Youtube-VOS dataset[12] and then finetuned it on DAVIS2017 dataset[9]. Since training the backbone CNN and spatio-temporal RNN jointly consumes a large amount of computing resources, we have to split the training stage into the following two steps:

- Step1: We discard the temporal connections in the spatio-temporal RNN and train the CNN and spatial RNN parts together.
- Step2: We fix the CNN part and tuning the parameters of spatio-temporal RNN module.

In training stage, we expected that the extracted features corresponding to the same object are closed to each other across different frames. Similar to [5], we randomly select several frames as anchor frames and pool frames. The foreground pixels in anchor frames make up the anchor set $\mathcal{A}$. The foreground pixels and background pixels in pool frames make up the positive pool $\mathcal{P}$ and the negative pool $\mathcal{N}$, respectively. It is expected that the extracted feature of the samples in $\mathcal{A}$ are pushed closer to the samples in $\mathcal{P}$, and stayed away from the samples in $\mathcal{N}$. Let $F(\cdot)$ denotes the feature extracted with the combination of the corresponding CNN and RNN networks described in Step1 and Step2. Then the loss function can be formulated as:

$$\sum_{a \in \mathcal{A}} \frac{\min_{p \in \mathcal{P}} \|F(a) - F(p)\|_2^2}{\min_{n \in \mathcal{N}} \|F(a) - F(n)\|_2^2 + \alpha}, \tag{4}$$

where α is a small constant used to avoid zero denominator.

2.4.2 Model Inference.

Given a test video, we first use the feature extraction module to extract a feature map from each video frame. Then we feed the feature maps into the spatio-temporal RNN module to embed more spatio-temporal contextual information. Finally, we employ the proposed online adaption module to segment the objects indicated by the scribble annotations.

3. Experiments

We evaluated our approach on the interactive track of DAVIS Challenge 2019 using the interactive python package[3] released by the organizers. In the challenge, the methods interact with a server for 8 times to evaluate a sequence and the maximum time for each interaction is 30×n seconds where n is the number of objects. The performance of the methods are measured by two metrics[2]: AUC and $\mathcal{J}\&\mathcal{F}$@60s. AUC is the area under the curve of the plot Time vs $\mathcal{J}\&\mathcal{F}$ and $\mathcal{J}\&\mathcal{F}$@60s is the $\mathcal{J}\&\mathcal{F}$ value at 60s.

Since the features extracted by the first two modules (feature extraction and spatio-temporal RNN module) are shared in all interaction steps, we only compute it in the first interaction to save time. In the last(8th) interaction, we used RGMP[11] to propagate the predicted masks from frames with scribble annotations to the neighboring frames to refine the segmentation results. Our methods took the third place on the challenge with an AUC of 0.621 and a $\mathcal{J}\&\mathcal{F}$@60s of 0.601.

4. Conclusion

In this paper, we present a novel framework for fast video object segmentation. In the framework, a spatio-temporal RNN module is proposed to embed the spatio-temporal contextual information into the feature map computed by a CNN model. Meanwhile, we develop an online adaption module which can efficiently tune the model parameters for segmenting specific objects. Our methods achieved the third place on the interactive track of DAVIS Challenge 2019.

[3] https://interactive.davischallenge.org

References

[1] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *CVPR*, 2017. 3

[2] S. Caelles, A. Montes, K.-K. Maninis, Y. Chen, L. Van Gool, F. Perazzi, and J. Pont-Tuset. The 2018 davis challenge on video object segmentation. *arXiv:1803.00557*, 2018. 1, 4

[3] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 40(4):834–848, 2018. 1, 2

[4] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 2

[5] Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, 2018. 2, 4

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2

[7] Y. Liu, P.-T. Jiang, V. Petrosyan, S.-J. Li, J. Bian, L. Zhang, and M.-M. Cheng. Del: Deep embedding learning for efficient image segmentation. In *IJCAI*, 2018. 1

[8] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 3

[9] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 3

[10] B. Shuai, Z. Zuo, B. Wang, and G. Wang. Scene segmentation with dag-recurrent neural networks. *TPAMI*, 40(6):1480–1493, 2018. 2, 3

[11] S. Wug Oh, J.-Y. Lee, K. Sunkavalli, and S. Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, 2018. 1, 4

[12] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018. 3

[13] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang. Fast visual tracking via dense spatio-temporal context learning. In *ECCV*, 2014. 1