Local and Global Dynamic Kernels for Video Object Segmentation

Haochen Wang[1], Yongtuo Liu[1], Yan Gao[2], Xiaolong Jiang[3], Efstratios Gavves[1]
University of Amsterdam[1], Chinese Academy of Sciences[2], Beihang University[3]
{h.wang3, y.liu6}@uva.nl, gaoyan@ict.ac.cn , xljiang@buaa.edu.cn, egavves@uva.nl

Abstract

Video object segmentation remains a challenging task due to the significant variations of objects across video frames. The state-of-the-art methods tackle the issue by pixel-level similarity calculation between frames, which typically focus on low-level feature matching and typically lead to missing-or-over-segmentation because of the lack of instance-level awareness. In this paper, we propose Local and Global Dynamic Kernels to improve the awareness of instance-level features. Specifically, we utilize segmented object regions from previous frames to dynamically generate semantic convolution kernels. Then, those kernels, encoded with the overall object features, are convolved with the current frame to generate consistent object masks. The proposed method achieved fourth place in YouTube-VOS 2022 challenge.

1. Introduction

Video object segmentation aims to segment the foreground objects from the background. It has many potential applications, with videos becoming more and more popular in media content, e.g., autonomous driving, augmented reality, and interactive video editing. In this paper, we focus on video object segmentation in a semi-supervised setting, where segmentation of target objects in the first frame is given, and our goal is to segment the target objects in the subsequent frames with consistent identities. It remains a challenging task as objects change dramatically over time due to their intrinsic characteristics and external occlusions.

Some early methods (e.g., MaskRNN [5], PReMVOS [7]) propose mask refinement in the previous frames with one extra convolution network. However, mask propagation accumulates errors across frames, especially in occluded and missing scenarios. Recently, matching-based methods, such as FEELVOS [10], CFBI [11], STM [8], KMN [9], and MiVOS [2], have received more attention due to their promising results compared to refinement-based methods. They typically conduct matching between pixels in the current frame and the previous ones. For example, STM [8], KMN [9], and MiVOS [2] design a memory mechanism to store previous frames and predicted masks of them to make more use of the matching power. While effective, the dense matching-based methods focus on pixel-level appearance matching in details, which lacks instance-level awareness. In this paper, we propose Local and Global Dynamic Kernels to encode the overall object features from previous frames. Specifically, besides dense pixel-level matching, we further propose to utilize previous frames to dynamically generate convolution kernels. Then the dynamic kernels with instance-level features are convolved with the current frame to predict the segmentation mask. In this way, we can exploit the merits of both pixel-level and instance-level matching. We conducted extensive experiments on the Youtube-VOS datasets. Our proposed method achieved fourth place in YouTube-VOS 2022 challenge.

2. Method

In our pipeline, we sequentially process video frames given the mask annotations in the first frame. As STM [8], we consider the current frame as the *query* frame and the previous frames with mask annotations (for the first frame) or predictions (for the preceding frames) as *memory* frames. The overview of our framework is shown in Figure 1. Our framework consists of three main components: Query and Memory Encoder, Local and Global Dynamic Kernels Generation, and Mask Decoder.

2.1. Query and Memory Encoder

Query Encoder. We utilize convolutional networks as the Query Encoder. The Query Encoder takes the query frame as input to generate the query feature maps through convolution layers. The output feature maps are $\mathbf{f}^Q \in \mathbb{R}^{H^Q \times W^Q \times C^Q}$, where H^Q, W^Q, and C^Q are the height, width, and channel dimensions.

Memory Encoder. The architecture of the memory encoder is the same as the query encoder except for the input. The input of the memory encoder is the previous frame concatenated with the according segmentation mask. After feature extraction of convolutional layers, the memory encoder outputs feature maps $\mathbf{f}^M \in \mathbb{R}^{H^M \times W^M \times C^M}$, where

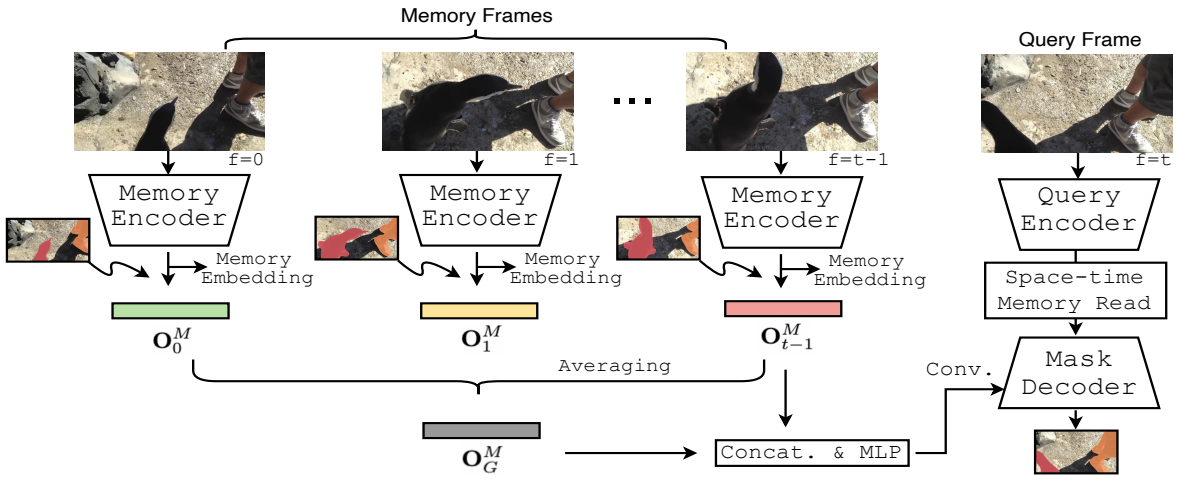

Figure 1. Framework of the proposed method. Object features $\mathbf{O}_0^M$, $\mathbf{O}_1^M$, and $\mathbf{O}_{t-1}^M$ are extracted from memory frames with the guidance of the corresponding segmentation mask. Local and global dynamic kernels are then dynamically generated by concatenating global and local object features (i.e., $\mathbf{O}_G^M$ and $\mathbf{O}_{t-1}^M$) and convolved with the extracted features of the query frame for instance-aware segmentation prediction. *Memory Embedding* and *Space-time Memory Read* are borrowed from STM [8].

H^M, W^M, and C^M are the height, width, and channel dimensions of memory frames. Then the $\mathbf{f}^M$ and $\mathbf{f}^Q$ are fed to the space-time memory read module to propagate the mask information from memory frames to the current frames, as in STM [8].

2.2. Local and Global Dynamic Kernels Generation

After feature extraction of query and memory frames, we propose Local and Global Dynamic Kernels to extract the local and global overall object features and increase the model ability of instance-level awareness. Here, "Local" and "Global" means we utilize different memory frames to consider both short- and long-range dependencies in terms of time.

We first extract the foreground object features of memory frames $\mathbf{O}^M \in \mathbb{R}^{C^M}$ by weighted average given the predicted object mask $\mathbf{m}^M$:

$$\mathbf{O}^M = \frac{\sum_i \mathbf{m}^M \cdot \mathbf{f}^M}{\sum_i \mathbf{m}^M}, \tag{1}$$

where i is the pixel index of the reference frame. Then we obtain the global object features $\mathbf{O}_G^M$ by averaging the $\{\mathbf{O}^M\}_1^{t-1}$ through all the reference frames $1:t-1$.

Finally we utilize three fully connected layers to generate the parameters of convolutional layers in the Mask Decoder, which are further convolved (interacted) with query features to output the segmentation masks. The parameter generation process is formulated as:

$$k_{LG} = \mathcal{F}_L(\mathcal{C}(\mathbf{O}_G^M, \mathbf{O}_{t-1}^M); \theta), \tag{2}$$

where $\mathcal{F}_L(;\theta)$ is the fully connected layers with trainable parameters θ. The $\mathcal{C}$ denote concatenation operation.

2.3. Mask Decoder

We follow the structure in STM [8] to build the the mask decoder. Input with the features from the space-time memory read module, the decoder gradually upscale the input feature map. The refinement module at every stage takes both the output of the previous stage and a feature map from the query encoder at the corresponding scale through skip-connections. The output of the last refinement block is then convoluted with k_{LG} to generate the segmentation mask.

3. Experiments

3.1. Training Detail

We follow the three-stage training strategy in STCN [3]. Specifically, we first train our model on static image datasets (stage 0). During this stage, each static image is expanded into a pseudo video of 3 frames through data augmentation. Then we train our model on BL30k dataset [2] (stage 1). Finally, the pre-trained model is fine-tuned on the combination of DAVIS and YouTube-VOS training sets (stage 2). The sampling rate between DAVIS and Youtube-VOS is set to 1:10. We randomly crop 384×384 patches from images during training. The batch size is set to 8 for stage 0,1 and 4 for stage 2. We use a single Tesla A100 GPU for both training and inference. We use multi-scale and flip as augmentations to obtain stable results during inference. The inference scales are set to 480 and 600. We trained our models with four different backbones: Swin-B [6], Wide-Resnet50 [12], Resnest101 [13] and Seresnet152 [4], and obtain the final results by the ensemble of those models. The ensemble details are shown in Ablation Study 3.3.

Figure 2. Qualitative results of the proposed method on Youtube-VOS 2022.

Team	Overall	$\mathcal{J}_{seen}$	$\mathcal{J}_{unseen}$	$\mathcal{F}_{seen}$	$\mathcal{F}_{unseen}$
Thursday_Group	0.872	0.855	0.817	0.914	0.903
ux	0.867	0.844	0.819	0.903	0.903
zjmagicworld	0.862	0.841	0.816	0.895	0.896
whc	**0.862**	**0.840**	**0.818**	**0.894**	**0.896**
gogo	0.861	0.847	0.808	0.901	0.890
sz	0.857	0.831	0.815	0.886	0.896

Table 1. Ranking results on the YouTube-VOS 2022 test set. "seen" and "unseen" indicate whether the categories of tracking instances appeared in the training set or not. Our results are highlighted in bold.

3.2. Results

As shown in Table 1, our method achieves an overall score of 86.2% on the YouTube-VOS Challenge 2022 test set (Semi-VOS track) and ranks the fourth place. Some qualitative results are shown in Figure 2.

3.3. Ablation Study

We conduct extensive ablation studies on Youtube-VOS 2019 validation set and 2022 test set. As shown in Table 2, the local and global dynamic kernels bring 0.5% and 0.3% overall performance improvements, respectively. By applying both of the dynamic kernels, the overall performance improves from 84.0% to 84.7%. We add the ASPP module [1] before the decoder to obtain features with multiple receptive fields. By adding the dynamic kernels and ASPP module, the proposed method achieves 85.1% without flip and multi-scale testing on Youtube-VOS 2019 Validation set.

We utilize some inference tricks to further boost the performance. Table 3 shows the ablation studies of inference tricks on the Youtube-VOS 2022 test dataset. As shown, by applying multi-scale and flip inference, the performance improves by 1.1%. We also utilize the model ensemble, which consists of four models with different backbones to promote model performance. Specifically, we average the output logits of four different models with backbones: Swin-B [6], Wide-Resnet50 [12], Resnest101 [13] and Seresnet152 [4]. After the ensemble, we achieve 86.2% on Youtube-VOS 2022 test dataset.

Local Kernel	Global Kernel	ASPP	Overall
			0.840
	✓		0.843
✓			0.845
✓	✓		0.847
✓	✓	✓	0.851

Table 2. Module ablation study on Youtube-VOS 2019 Validation set.

Flip & Multi-scale	Ensemble	Overall
		0.830
✓		0.841
✓	✓	0.862

Table 3. Inference tricks on Youtube-VOS 2022 Test set.

3.4. Conclusion

In this paper, we propose Local and Global Dynamic Kernels to improve the awareness of instance-level features in existing video object segmentation methods which focus on low-level feature matching only. Specifically, we utilize segmented object regions from previous frames to dynamically generate semantic convolution kernels, which are then convolved with the current frame to generate consistent object masks. Experiments on YouTube-VOS 2022 show the proposed instance-aware dynamic convolution can

achieve superior performance, especially the fourth place in YouTube-VOS 2022 challenge.

References

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 3

[2] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5559–5568, 2021. 1, 2

[3] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Advances in Neural Information Processing Systems*, 34:11781–11794, 2021. 2

[4] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 2, 3

[5] Yuan-Ting Hu, Jia-Bin Huang, and Alexander Schwing. Maskrnn: Instance level video object segmentation. *Advances in neural information processing systems*, 30, 2017. 1

[6] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2, 3

[7] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *Asian Conference on Computer Vision*, pages 565–580, 2018. 1

[8] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9226–9235, 2019. 1, 2

[9] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In *European Conference on Computer Vision*, pages 629–645, 2020. 1

[10] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9481–9490, 2019. 1

[11] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *European Conference on Computer Vision*, pages 332–348, 2020. 1

[12] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 2, 3

[13] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2736–2746, 2022. 2, 3