A memory reader with spatial-constrain kernel for Video Object Segmentation

Changhao Qiao[1] Haochen Wang[4] Cheng Chen[3] Yan Gao[2] Qimeng Wang[3] Xu Tang[5]

[1]Harbin Institute of Technology, Shenzhen [2]Institute of Computing Technology, Chinese Academy
[3]Huazhong University of Science and Technology [4]Beihang University [5]ShanghaiTech University

19s153188@stu.hit.edu.cn, haochenwang@buaa.edu.cn,
{eic_chencheng, qimengwang}@hust.edu.cn, yangao0119@gmail.com, tangxu@shanghaitech.edu.cn

Abstract

Semi-supervised video object segmentation is a challenging task in computer vision. It aims to segment some particular instances in the video given the ground truth objects' masks of the first frame. Space-time memory network(STM) has a great influence on VOS. It makes full use of the features of past frames through a memory network, which significantly improves the accuracy. However, some problems remain to be resolved. Its non-local matching mechanism makes it easy to match multiple wrong objects that have similar appearances. So we propose the SKN network, which can improve the feature matching accuracy by using a spatial-constrain kernelized memory reader. Spatial-constrain prior is generated according to the last mask estimations. This prior can improve the features' spatial continuity and ensure that the most similar point of features only appears in the right object's region. Then a gaussian kernel is used to suppress the response of features that are far away from the most similar point. Besides, adversarial training strategy is used to improve the robustness of SKN. Finally, the proposed method achieves the J&F mean score of 83.6% on the third YouTube-VOS competition.

1. Introduction

Video object segmentation (VOS) is one of the most challenging tasks in computer vision. It has been widely used in many fields, including autonomous driving, video editing, video composition, etc. In the semi-supervised VOS task, the ground truth mask of the first frame is provided. The algorithms are desired to predict all the masks of the subsequent frames in a video. This task is challenging because the appearance, position, and size of the objects can vary greatly in the video. Besides, objects in the video may have similar appearances, which may make the algorithms find the wrong object.

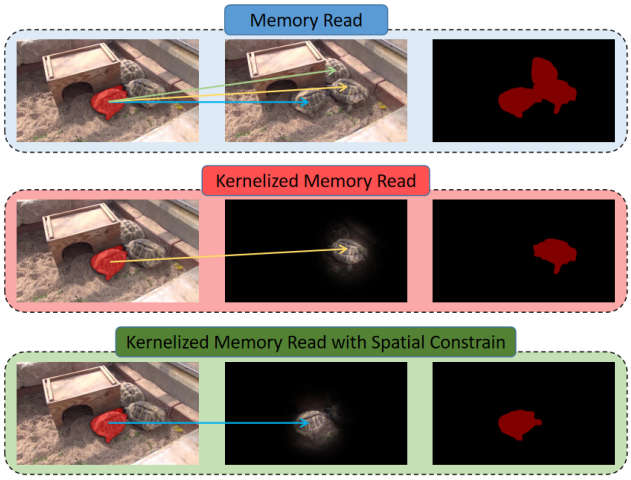

Figure 1. Illustration of the SKN. The right object is the left tortoise in the image. For STM, the memory reader module will find multiple tortoises that have similar appearances. So multiple tortoises will be detected finally. For the kernelized memory reader module of KMN, it will find the most similar point in the feature map, then use a gaussian kernel to suppress the response of feature map that is far away from the point. However, the appearances of tortoises are so similar that the most similar feature point appears in the region of other tortoises. For our spatial-constrain kernelized memory reader, the most similar point will be found in the right tortoise's region according to the previous mask.

Space-time memory networks(STM)[1] have achieved great success in VOS. It uses a memory network to store the features of past frames. Then space-time attention mechanism is used to match pixel-level features of memory with those of the query frame. This approach takes full advantage of the prior information of previous masks, so it has a good ability to handle occlusion and fast motion. However, many problems still remain to be resolved. Although the non-local matching mechanism of STM can make good use of the features of the previous frames, it often occurs that

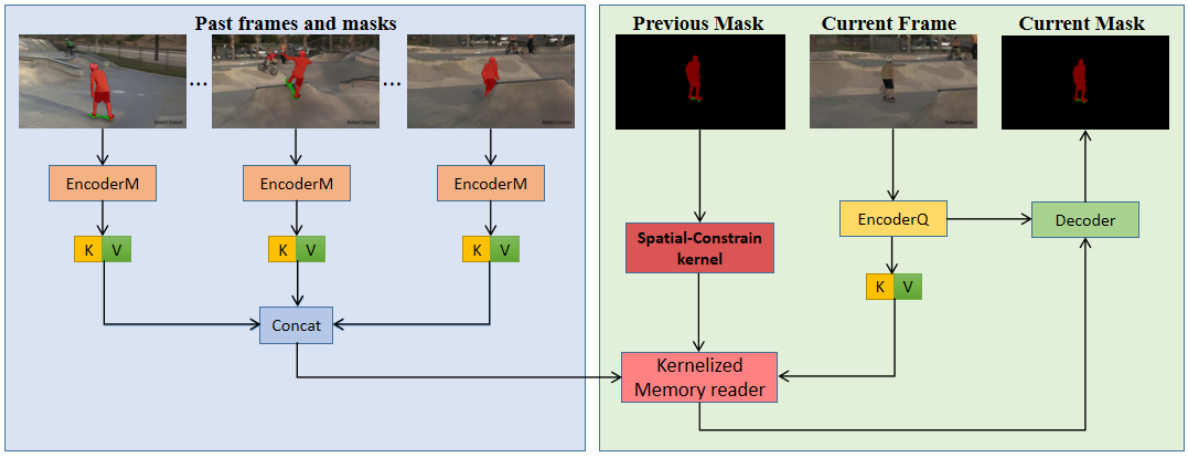

Figure 2. Overall architecture of spatial-constrain kernelized memory network (SKN). The overall structure is similar to the STM network. We added a spatial-constrain kernel generated by the previous mask. Then a kernelized memory reader is used to suppress the response of regions far from this point in the feature map.

the right object matches multiple wrong objects, especially when they have similar appearances.

KMN[2] network proposed a kernelized memory reader. This kind of memory reader uses a gaussian kernel to suppress responses of similar objects' features and prevent the algorithm from finding multiple similar targets.

Nevertheless, KMN also faces the problem of spatial discontinuity. When different objects have similar appearances, the most similar point may appear in the wrong objects' regions, resulting in matching errors. To solve this problem, we propose the spatial-constrain kernelized memory reader. On the basis of KMN, we add the spatial-constrain prior. Specifically, we use the previous mask estimation as the prior of spatial position. Mask's center point is taken to generate a spatial gaussian kernel, which can be used to suppress the features' response which is far away from the previous objects' region. In this way, the feature map keeps the spatial continuity. So the most similar point in feature map will be always found in the right object's region to prevent the model from detecting wrong objects.

Generally, training videos are so short that VOS algorithms are easily disturbed by errors in the inference process. In order to improve the robustness of the network, we introduce an adversarial training strategy. With this strategy, SKN has less confidence in previously estimated results and avoids the accumulation of errors.

Finally, the SKN network achieves the J&F mean score of 77.8% for the Davis2017 test-dev dataset, 83.7% for the Youtube-VOS val dataset and 83.6% on the third YouTube-VOS competition.

2. Approch

2.1. Architecture

The memory reader of STM uses the feature matching mechanism of query-to-memory. Since there isn't any spatial constraint, it is easy to get many similar features from memory frames. KMN adds the memory-to-query matching mechanism to ensure that only the most similar objects are selected. However, since spatial continuity of features is not taken into account, the most similar object is likely to be wrong when objects have similar appearances. SKN further adds the mask-to-memory mechanism, introducing spatial continuity of the most similar objects. The overall structure of SKN is shown in Figure 2.

2.2. Memory reader with spatial-constrained kernel

For STM, previous images and corresponding masks are encoded to get the features: $K^M \in \mathsf{R}^{T\ H\ W\ C/8}$ and $V^M \in \mathsf{R}^{T\ H\ W\ C/2}$. The former stores addressing information and generates the matching scores of memory frames. The latter stores detailed information for current mask estimation. H,W,C are height, width and channel of the feature map respectively, while T is the number of frames in the memory network. After that, the current image is encoded to generate two features: $K^Q \in \mathsf{R}^{H\ W\ C/8}, V^Q \in \mathsf{R}^{H\ W\ C/2}$. The correlation maps can be generated as $c \in \mathsf{R}^{THW\ HW}$ from K^M, K^Q

$$c = K^M \left(K^Q\right)^T \tag{1}$$

Then, The weight matrix of V^Q can be obtained after a softmax layer:

$$W_{i,j} = \frac{\exp\left(c_{i,j}\right)}{\sum_i \exp\left(c_{i,j}\right)} \tag{2}$$

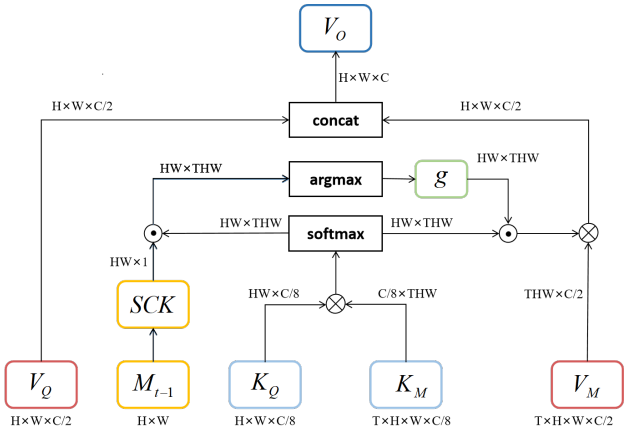

Figure 3. The memory reader of the SKN

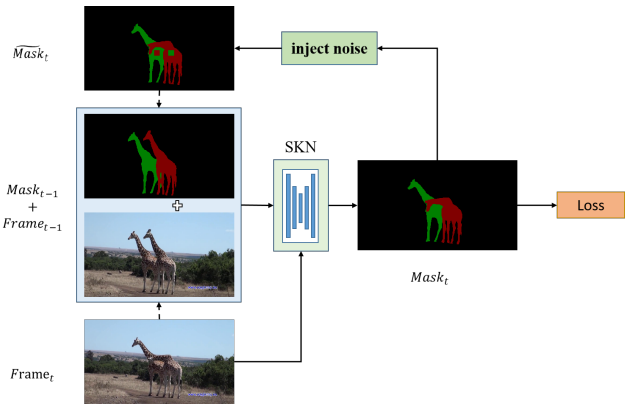

Figure 4. Adversarial training process

where i,j are the location of a pixel in the feature map. The final output $F \in \mathsf{R}^{H\ \ W\ \ C}$ of memory reader is then obtained:

$$F = \left[V^Q, W^T V^M\right] \tag{3}$$

$[\cdot,\cdot]$ denotes the concatenation.

In SKN network, spatial-constrain kernel $S_{t\ \ 1}$ was firstly obtained according to the last binary mask estimation $M_{t\ \ 1} \in \mathsf{R}^{H\ \ W}$. To be specific, we first take the set of non-zero points' positions of $M_{t\ \ 1}$ as: $P_{t\ \ 1} \in \mathsf{R}^{N\ \ 2}$. N is the number of non-zero points in $M_{t\ \ 1}$. Take the position of the center point as:

$$r = (r_x, r_y) = \text{Median}\left(P_{t\ \ 1}\right) \in \mathsf{R}^{1\ \ 2} \tag{4}$$

Then take the standard deviation of the distances between all the points of $P_{t\ \ 1}$ and r:

$$\sigma_s = \sqrt{\frac{\sum_{p2N_{t-1}} \left(p_x - r_x\right)^2 + \left(p_y - r_y\right)^2}{N}} \tag{5}$$

A spatial-constrain gaussian kernel $SCK \in \mathsf{R}^{H\ \ W}$ can be obtained according to r, σ_s :

$$SCK_{i,j} = \exp\left(-\frac{\left(i - r_x\right)^2 + \left(j - r_y\right)^2}{2\sigma_s^2}\right) \tag{6}$$

After that, SCK is used to filter W to enhance the spatial continuity of features

$$W_1 = W \odot SCK \tag{7}$$

$\odot$ means the element-wise multiplication. Then, the maximum point of W_1 is taken as the most similar point:

$$m = \arg\max\left(W_1\right), \tag{8}$$

A gaussian kernel $g \in \mathsf{R}^{H\ \ W}$ is generated with this point as the center.

$$g_{i,j} = \exp\left(-\frac{\left(i - m_x\right)^2 + \left(j - m_y\right)^2}{2\sigma_g^2}\right) \tag{9}$$

σ_g means the standard deviation of g. This kernel will be used to suppress the region's response which is far from the most similar point:

$$W_2 = W \odot g \tag{10}$$

Finally, the output of memory reader will be obtained:

$$F = \left[V^Q, W_2^T V^M\right] \tag{11}$$

F will be used as the input of the decoder to get the current mask estimation. The whole structure of spatial-constrain kernelized memory reader is shown in Fig4.

It can be seen that the memory reader of SKN firstly generates a spatial-constrain gaussian kernel based on the previous mask. It is used as spatial prior to enhance the spatial continuity of features. Then use the maximum point of correlatraion map as the most similar point. Because of the spatial continuity of features, the most similar point is always located in the right object's region, even though many objects have similar appearance. Then another gaussian kernel g is generated using the most similar point as its center. This gaussian kernel will suppress the region's response which is far away from the most similar point.

3. Adversarial training

As the previous mask estimation is strongly correlated with the current mask, models often have excessive confidence in it, which may lead to the accumulation and amplification of errors. Besides, because of the limitation of GPU's memory, researchers typically use some very short videos

Method	Davis17 test-dev	Youtube val
Baseline(Resnest101)	72.7	80.2
+ASPP	73.4(+0.7)	81.1(+0.9)
+Adversarial training	75.8(+2.4)	82.0(+0.9)
+Spatial Kernel	76.9(+1.1)	82.7(+0.7)
+Flip and Multi-scale	77.8(+0.9)	83.7(+1.0)

Table 1. Ablation study on Davis17 test-dev and Youtube-VOS val dataset

to train their models, which may prevent models from identifying errors in time. To solve this problem, we propose the adversarial training strategy. Different from the general training strategy, we apply various random noises to the previous mask estimation to simulate the errors in the inference process. Specifically, for the videos containing multiple objects, we randomly select a square region of two objects and swap them. For the videos containing a single object, the foreground and background of two random rectangular regions are swapped. Besides, the mask estimations are randomly dilated or eroded to simulate the errors of edges in the inference process.

Applying these noises in the training stage has two advantages: First, It reduces the dependence of the model on the previous prediction results and forces the query encoder to learn more robust features from the query frame; Second, It can adapt the model to various errors that may occur in the inference process, which can prevent the accumulation of errors and improve the robustness of the model. This training strategy is shown in Fig4.

4. Implementation details

To improve the accuracy, we used Resnest101[3] as the backbone network of the encoder. In addition, the ASPP[4] module is added to improve the receptive field of the model like [5]. The ablation study in Youtube-VOS val dataset is shown in Table1.

4.1. Training

In order to train the model fully, our training strategy can be divided into three stages. In the first stage, coco[6] and other static instance segmentation datasets are used for pretraining. Every image and its corresponding mask are randomly flipped and rotated to generate three images, which can be seen as a short video. In the second stage, we use the BL30K dataset for training. This dataset is a super large synthetic dataset proposed by MiVOS[7]. In the third stage, the joint dataset of YouTubeVos[8], Davis16[9] and Davis17[10] is used for the final training. We resize the shortest edge of the image to 480, and crop a 380×380 patch from the image as the input. Adversarial training strategy is only used in the third stage.

4.2. Inference

To improve the final accuracy, we apply the flip and multi-scale testing. Finally, our SKN network achieves the J&F mean score of 77.8% for the Davis2017 test-dev dataset, 83.7% for the Youtube-VOS val dataset and 83.6% on the third YouTube-VOS competition.

References

[1] 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 (ICCV)*, pages 9226–9235, October 2019.

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

[3] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R. Manmatha, Mu Li, and Alexander Smola. Resnest: Split-attention networks, 2020.

[4] 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, 2018.

[5] Zhang B et al Zhang P, Hu L. Spatial consistent memory network for semi-supervised video object segmentation. In *CVPR Workshops*, June 2020.

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[7] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion, 2021.

[8] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 585–601, September 2018.

[9] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[10] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation, 2018.