The 2020 DAVIS Challenge on Video Object Segmentation - CVPR 2020 Workshops

A Kernel-based Approach for Video Object Segmentation

Hongje Seong Junhyuk Hyun Euntai Kim*
School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea
{hjseong, jhhyun, etkim}@yonsei.ac.kr

Abstract

Semi-supervised video object segmentation (VOS) is the task of predicting a target object in a video when the ground truth segmentation mask of the target object is given in the first frame. Recently, space-time memory networks (STM) have received significant attention as a promising solution for semi-supervised VOS. In STM, the current frame is considered to be the query frame for which the target is to be predicted, whereas the past (already predicted) frames are used as memory frames. When the standard "vanilla" STM is applied to VOS, however, an important point is overlooked. The solution (i.e., STM) is non-local but the problem (i.e., VOS) is predominantly local. Specifically, STM is non-local since it uses a correlation map to consider all possible correspondences between all parts of the query and memory frames. However, VOS is local since the target object in the query frame usually appears where the target was in the memory frames. Thus, entire images do not need to be searched using STM, but rather the local neighborhood where the target appeared in memory frames should be searched in VOS. To solve the mismatch between STM and VOS, we propose a new network named the kernelized memory network (KMN). Further, before being trained on real videos, our KMN is pre-trained on static images as in previous works. Unlike the previous works, however, we used the Hide-and-Seek strategy in pre-training to obtain the best possible results in handling occlusions and segment boundary extraction. Our approach earned us third place on the semi-supervised track in the 2020 DAVIS challenge.

1. Introduction

Video object segmentation (VOS) is the task of tracking target objects at the pixel level in a video, and is considered to be one of the most challenging problems in computer vision. VOS can be divided into two categories: semi-supervised VOS and unsupervised VOS. In semi-supervised VOS, the ground truth (GT) segmentation mask is given in the first frame and the segmentation mask must be predicted for the subsequent frames. In unsupervised VOS, however, no GT segmentation mask is given, and the task is to find and segment the main object in the video. This paper considers semi-supervised VOS.

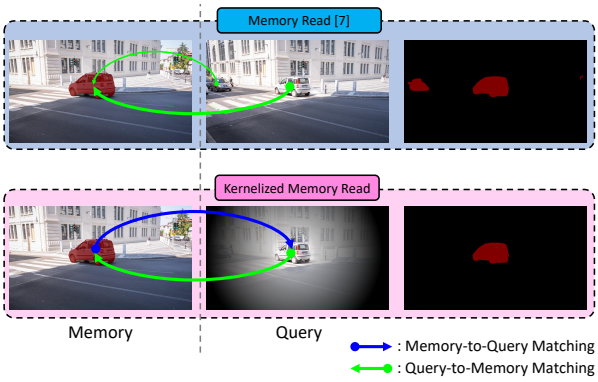

Figure 1. Illustration of KMN. In the vanilla STM, two cars in the query frame are matched with a car in the memory frame due to the non-local matching between the query and memory. The car in the middle is the correct match, while the car on the left is an incorrect match. In KMN, however, non-local matching between the query and memory is controlled by Gaussian kernel and only the car in the middle of the query frame is matched with the car in the memory. The width of the green curve indicates the strength of **key** matching.

Space-time memory networks (STM) [7] have recently received significant attention as a promising solution to semi-supervised VOS. The basic idea behind the application of STM to VOS is to use the intermediate frames between the first frame and current frame. This approach, however, overlooks the important point that the solution (*i.e.*, STM) is non-local, but the problem (*i.e.*, VOS) is predominantly local, as illustrated in Fig. 1. Specifically, STM is based on non-local matching between the query frame and memory frames. On the other hand, in VOS, the target object in the query frame usually appears in the local neighborhood of the target's appearance in the memory frames. To solve the problem arising from the use of the standard "vanilla" STM for VOS, this paper proposes a new memory network

*Corresponding author.

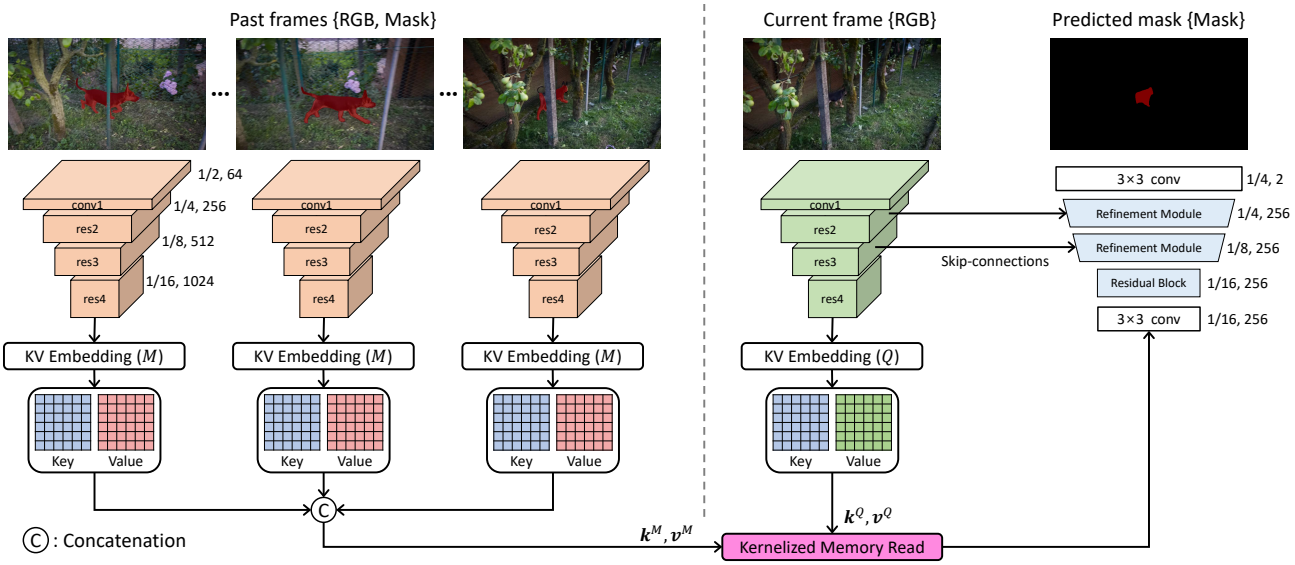

Figure 2. Overall architecture of our kernelized memory network (KMN). We follow the frameworks of [7], and propose a new operation of kernelized memory read. The numbers next to the block indicate the spatial size and channel dimension.

named the kernelized memory network (KMN). In KMN, the Gaussian kernel is employed to reduce the degree of non-localness of the STM and improve the effectiveness of the memory network for VOS.

Further, before being trained on real videos, our KMN is pre-trained on static images as in some previous works. In particular, multiple frames based on a random affine transform were used in [14, 7]. Unlike the training process in the previous works, however, we employ the Hide-and-Seek strategy during pre-training to obtain the best possible results in handling occlusions and segment boundary extraction. Hide-and-Seek [11] was initially developed for weakly supervised object localization, but we utilize it to pre-train KMN. This provides two key benefits. First, when used in the pre-training of KMN, Hide-and-Seek achieves segmentation results that are considerably robust to occlusion. To the best of our knowledge, this is the first time that Hide-and-Seek has been applied to VOS in order to make the predictions robust to occlusion. Second, Hide-and-Seek is used to refine the boundary of the object segment. Since most of the ground truths in segmentation datasets contain unclear and incorrect boundaries, it is fairly challenging to predict accurate boundaries in VOS. The boundaries created by Hide-and-Seek, however, are clear and accurate, and Hide-and-Seek appears to provide instructive supervision for clear and precise cuts for objects, as shown in Fig. 4. Our approach earned us third place on the semi-supervised track in the 2020 DAVIS challenge with a $\mathcal{J}_{\mathcal{M}}\&\mathcal{F}_{\mathcal{M}}$ score of 79.5%.

2. Kernelized Memory Network (KMN)

2.1. Architecture

The overall architecture of KMN is fairly similar to that of STM [7] and is illustrated in Fig. 2. Compared with STM [7], the primary difference in KMN lies in the memory read operation. We recommend that readers refer to [7] for more details about the overall frameworks of **key** and **value** embedding methods.

In the memory read of a vanilla STM [7], only `Query-to-Memory` matching is conducted. In the kernelized memory read of KMN, however, both `Query-to-Memory` matching and `Memory-to-Query` matching are conducted. The detailed explanation about kernelized memory read is provided in the subsequent subsections.

2.2. Kernelized Memory Read

In the memory read operation of STM [7], the non-local correlation map $c(\mathbf{p}, \mathbf{q})$ is generated by using the embedded **key** of the memory $\mathbf{k}^M = \left\{k^M(\mathbf{p})\right\} \in \mathbb{R}^{T\times H\times W\times C/8}$ and query $\mathbf{k}^Q = \left\{k^Q(\mathbf{q})\right\} \in \mathbb{R}^{H\times W\times C/8}$ as follows:

$$c\left(\mathbf{p}, \mathbf{q}\right) = k^M(\mathbf{p})k^Q(\mathbf{q})^\top \tag{1}$$

where H, W, and C are the height, width, and channel size of `res4` [4], respectively. $\mathbf{p} = [p_t, p_y, p_x]$ and $\mathbf{q} = [q_y, q_x]$ indicate the grid cell position of the **key** features. Then the query at position $\mathbf{q}$ retrieves the corresponding **value** from the memory using the correlation map by

$$r\left(\mathbf{q}\right) = \sum_{\mathbf{p}} \frac{\exp\left(c\left(\mathbf{p}, \mathbf{q}\right)\right)}{\sum_{\mathbf{p}} \exp\left(c\left(\mathbf{p}, \mathbf{q}\right)\right)} v^M\left(\mathbf{p}\right) \tag{2}$$

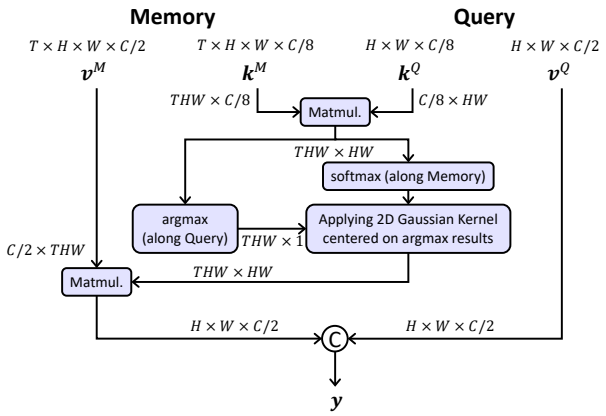

Figure 3. Kernelized memory read operation.

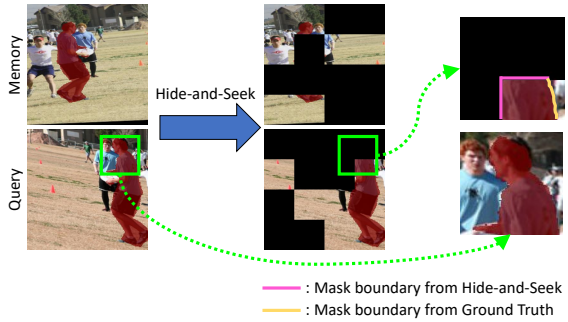

Figure 4. A pair of images generated during pre-training using Hide-and-Seek. The mask indicated in red denotes the ground truth of the target object.

where $\mathbf{v}^M = \{v^M(\mathbf{p})\} \in \mathbb{R}^{T\times H\times W\times C/2}$ is the embedded **value** of the memory. Then the retrieved **value** $r(\mathbf{q})$, which is of size $H \times W \times C/2$, is concatenated with the query **value** $\mathbf{v}^Q \in \mathbb{R}^{H\times W\times C/2}$, and they are fed to the decoder.

However, the vanilla memory read operation has two inherent problems. First, every grid in the query frame searches the memory frames for a target object, but not vice versa. That is, there is only `Query-to-Memory` matching in the STM. Thus, when multiple objects in the query frame look like a target object, all of them can be matched with the same target object in the memory frames. Second, the non-local matching in the STM might be ineffective in VOS since it overlooks the fact that the target object in the query should appear where it was in the memory frames.

To solve these problems, we propose a kernelized memory read operation using 2D Gaussian kernels. First, the non-local correlation map $c(\mathbf{p},\mathbf{q}) = k^M(\mathbf{p})k^Q(\mathbf{q})^\top$ between the query and the memory is computed as in the STM. Second, for each grid $\mathbf{p}$ in the memory frames, the best-matched query position $\widehat{\mathbf{q}}(\mathbf{p}) = [\widehat{q}_y(\mathbf{p}), \widehat{q}_x(\mathbf{p})]$ is searched by

$$\widehat{\mathbf{q}}(\mathbf{p}) = \arg\max_{\mathbf{q}} c(\mathbf{p},\mathbf{q}) \tag{3}$$

and it is the `Memory-to-Query` matching. Third, a 2D Gaussian kernel $\mathbf{g} = \{g(\mathbf{p},\mathbf{q})\} \in \mathbb{R}^{T\times H\times W\times H\times W}$ that is centered on the $\widehat{\mathbf{q}}(\mathbf{p})$ is computed by

$$g(\mathbf{p},\mathbf{q}) = \exp\left(-\frac{(q_y - \widehat{q}_y(\mathbf{p}))^2 + (q_x - \widehat{q}_x(\mathbf{p}))^2}{2\sigma^2}\right) \tag{4}$$

where σ is the standard deviation. Using the Gaussian kernels, **value** in the memory is retrieved in a local manner by

$$r^k(\mathbf{q}) = \sum_{\mathbf{p}} \frac{\exp\left(c(\mathbf{p},\mathbf{q})/\sqrt{d}\right) g(\mathbf{p},\mathbf{q})}{\sum_{\mathbf{p}} \exp\left(c(\mathbf{p},\mathbf{q})/\sqrt{d}\right) g(\mathbf{p},\mathbf{q})} v^M(\mathbf{p}) \tag{5}$$

where d is the channel size of the **key**. This is `Query-to-Memory` matching. Here, $\frac{1}{\sqrt{d}}$ is a scaling factor taken from [12] and it aims to prevent the argument in the softmax from growing large in magnitude, or equivalently preventing the softmax from becoming saturated. The kernelized memory read operation is summarized in Fig. 3.

Note that our kernelized memory read operation is inspired by the kernel soft argmax [5], but its application and objective are completely different. The kernel soft argmax [5] applies a kernel to the memory feature (`Query-to-Memory`) to simply serve as a gradient propagatable argmax function. If the kernel in [5] was directly applied to the example in Fig. 1, the kernel would be applied to the memory feature (the first column) in the figure and VOS would fail as in the STM [7].

3. Pre-training by Hide-and-Seek

As in the previous works [8, 14, 7], our KMN is pre-trained using static image datasets that include foreground object masks [2, 6, 3, 10, 1, 13]. The basic idea of pre-training a VOS network is to generate a video with foreground object masks synthetically from a single static image. Applying random affine transforms to a static image and the corresponding object mask can yield a synthetic video and the video can be used to pre-train a VOS network. The problem with synthetic generation of a video from a static image, however, is that the occlusion of the target object does not occur in a generated video. Thus, the simulated video cannot train the pre-trained KMN to cope with the common occlusion in the VOS. To solve this problem, Hide-and-Seek is used to generate a video with occlusions synthetically. Some patches are randomly hidden or blocked and the occlusions are synthetically generated in training samples. Hide-and-Seek can pre-train KMN to be robust to the common occlusion in VOS. This idea is illustrated in Fig. 4.

Further, it should be noted that most of the segmentation datasets contain inaccurate masks (GTs) near the ob-

ject boundaries. An example is illustrated in Fig. 4; in this figure, the ground truth mask contains incorrect boundaries on the head of the running person. However, Hide-and-Seek creates a clear object boundary as represented by the pink line in Fig. 4.

4. Implementation Details

4.1. Training

For training the KMN, both the DAVIS17 [[9] and Youtube-VOS [15] training sets are used, and our training strategy is identical to that adopted in STM [7], with the difference that we use Hide-and-Seek during pre-training.

During the pre-training, we generate three frames using a single static image by randomly applying rotation, flip, color jittering, and cropping, similar to [14, 7]. We then use the Hide-and-Seek framework, as described in Section 3. We first divide the image into a 24×24 grid, which is the same spatial size as the **key** feature. Each cell in the grid has a uniform probability to be hidden respectively. We gradually increase the probability from 0 to 0.5.

Note that the Gaussian kernel was not applied during training. Since the argmax function, which determines the center point of the Gaussian kernel, is a discrete function, the error of the argmax cannot be propagated backward during training. Thus, if the Gaussian kernel is used during training, it attempts to optimize networks based on the incorrectly selected feature by argmax, and it leads to performance degradation.

4.2. Inference

The inference details in our approach are also almost similar to [7], except for the addition of the Gaussian kernel. We empirically set the hyper-parameter of σ in (4) to 6.

Finally, we achieve $\mathcal{J}_\mathcal{M}\&\mathcal{F}_\mathcal{M}$ score of 78.3% in a 600p resolution video and 79.5% in an ensemble of 320p, 480p, and 600p resolution videos on the DAVIS test-challenge set.

Acknowledgement

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT (NRF-2017M3C4A7069370).

References

[1] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2014. 3

[2] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 3

[3] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, pages 991–998. IEEE, 2011. 3

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[5] Junghyup Lee, Dohyung Kim, Jean Ponce, and Bumsub Ham. Sfnet: Learning object-aware semantic correspondence. In *CVPR*, pages 2278–2287, 2019. 3

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 3

[7] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, October 2019. 1, 2, 3, 4

[8] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, pages 2663–2672, 2017. 3

[9] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 4

[10] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):717–729, 2015. 3

[11] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, pages 3544–3553. IEEE, 2017. 2

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 3

[13] Jingdong Wang, Huaizu Jiang, Zejian Yuan, Ming-Ming Cheng, Xiaowei Hu, and Nanning Zheng. Salient object detection: A discriminative regional feature integration approach. *International Journal of Computer Vision*, 123(2):251–268, 2017. 3

[14] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, pages 7376–7385, 2018. 2, 3, 4

[15] 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 *ECCV*, pages 585–601, 2018. 4