Efficient Regional Memory Network for Video Object Segmentation

Haozhe Xie[1,2] Hongxun Yao[1] Shangchen Zhou[3] Shengping Zhang[1,4] Wenxiu Sun[2,5]
[1] Harbin Institute of Technology [2] SenseTime Research and Tetras.AI
[3] S-Lab, Nanyang Technological University [4] Peng Cheng Laboratory [5] Shanghai AI Laboratory
https://haozhexie.com/project/rmnet

Abstract

Recently, several Space-Time Memory based networks have shown that the object cues (e.g. video frames as well as the segmented object masks) from the past frames are useful for segmenting objects in the current frame. However, these methods exploit the information from the memory by global-to-global matching between the current and past frames, which lead to mismatching to similar objects and high computational complexity. To address these problems, we propose a novel local-to-local matching solution for semi-supervised VOS, namely Regional Memory Network (RMNet). In RMNet, the precise regional memory is constructed by memorizing local regions where the target objects appear in the past frames. For the current query frame, the query regions are tracked and predicted based on the optical flow estimated from the previous frame. The proposed local-to-local matching effectively alleviates the ambiguity of similar objects in both memory and query frames, which allows the information to be passed from the regional memory to the query region efficiently and effectively. Experimental results indicate that the proposed RMNet performs favorably against state-of-the-art methods on the DAVIS and YouTube-VOS datasets.

1. Introduction

Video object segmentation (VOS) is a task of estimating the segmentation masks of class-agnostic objects in a video. Typically, it can be grouped into two categories: unsupervised VOS and semi-supervised VOS. The former does not resort to any manual annotation and interaction, while the latter needs the masks of the target objects in the first frame. In this paper, we focus on the latter. Even the object masks in the first frame are provided, semi-supervised VOS is still challenging due to object deformation, occlusion, appearance variation, and similar objects confusion.

With the recent advances in deep learning, there has been tremendous progress in semi-supervised VOS. Early methods [10, 18, 24] propagate object masks from previous frames using optical flow and then refine the masks with a fully convolutional network. However, mask propagation usually causes error accumulation, especially when target objects are lost due to occlusions and drifting. Recently, matching-based methods [4, 17, 22, 28, 32, 41] have attracted increasing attention as a promising solution to semi-supervised VOS. The basic idea of these methods is to perform global-to-global matching to find the correspondence of target objects between the current and past frames. Among them, the Space-Time Memory (STM) based approaches [17, 22, 28] exploit the past frames saved in the memory to better handle object occlusion and drifting. However, these methods memorize and match features in the regions without target objects, which lead to mismatching to similar objects and high computational complexity. As shown in Figure 1, they are less effective to track and distinguish the target objects with similar appearance.

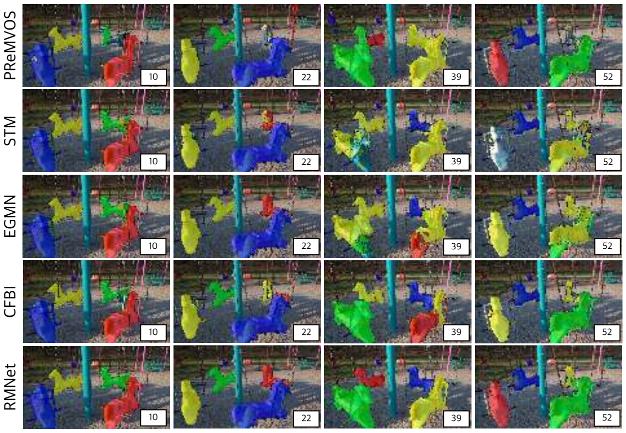

Figure 1. A representative example of video object segmentation on the DAVIS 2017 dataset. Compared to existing methods rely on optical flows (*e.g.* PReMVOS [18]) and global feature matching (*e.g.* STM [22], EGMN [17], and CFBI [41]), the proposed RMNet is more robust in segmenting similar objects.

The mismatching in the global-to-global matching can be divided into two categories, as illustrated in Figure 2. **(i)** The target object in the current frame matches to the wrong object in the past frame (solid red line). **(ii)** The target ob-

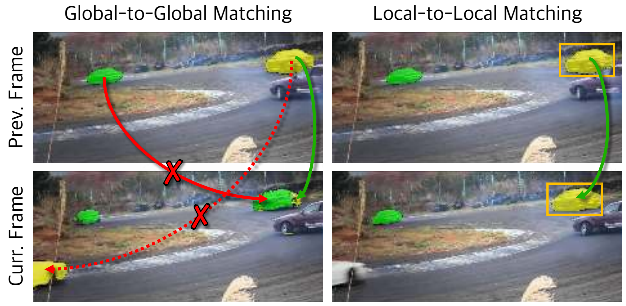

Figure 2. Comparison between global-to-global matching and local-to-local matching. The green and red lines represent correct and incorrect matches, respectively. The proposed local-to-local matching performs feature matching between the local regions containing target objects in past and current frames (highlighted by the yellow bounding box), which alleviates the ambiguity of similar objects.

ject in the past frame matches to the wrong object in the current frame (dotted red line). The two types of mismatching are caused by unnecessary matching between the regions without target objects in the past and current frames, respectively. Actually, the target objects appear only in small regions in each frame. Therefore, it is more reasonable to perform local-to-local matching in the regions containing target objects.

In this paper, we present the Regional Memory Network (RMNet) for semi-supervised VOS. To better leverage the object cues from the past frames, the proposed RMNet only memorizes the features in the regions containing target objects, which effectively alleviates the mismatching in **(i)**. To track and predict the target object regions for the current frame, we estimate the optical flow from two adjacent frames and then warp the previous object masks to the current frame. The warped masks provide rough regions for the current frame, which reduces the mismatching in **(ii)**. Based on the regions in the past and current frames, we present Regional Memory Reader, which performs feature matching between the regions containing target objects. The proposed Regional Memory Reader is time efficient and effectively alleviates the ambiguity of similar objects.

The main contributions are summarized as follows:

- We propose Regional Memory Network (RMNet) for semi-supervised VOS, which memorizes and tracks the regions containing target objects. RMNet effectively alleviates the ambiguity of similar objects.
- We present Regional Memory Reader that performs local-to-local matching between object regions in the past and current frames, which reduces the computational complexity.
- Experimental results on the DAVIS and YouTube-VOS datasets indicate that the proposed RMNet outperforms the state-of-the-art methods with much faster running speed.

2. Related Work

Propagation-based Methods. Early methods treat video object segmentation as a temporal label propagation problem. ObjectFlow [31], SegFlow [6] and DVSNet [39] consider video segmentation and optical flow estimation simultaneously. To adapt to specific instances, MaskTrack [24] fine-tunes the network on the first frame during testing. MaskRNN [10] predicts the instance-level segmentation of multiple objects from the estimated optical flow and bounding boxes of objects. CINN [1] introduces a Markov Random Field to establish spatio-temporal dependencies for pixels. DyeNet [16] and PReMVOS [18] combine temporal propagation and re-identification functionalities into a single framework. Apart from propagating masks with optical flows, object tracking is also widely used in semi-supervised VOS. FAVOS [5] predicts the mask of an object from several tracking boxes of the object parts. LucidTracker [14] synthesizes in-domain data to train a specialized pixel-level video object segmenter. SAT [3] takes advantage of the inter-frame consistency and deals with each target object as a tracklet. Despite promising results, these methods are not robust to occlusion and drifting, which causes error accumulation during the propagation [21].

Matching-based Methods. To handle object occlusion and drifting, recent methods perform feature matching to find objects that are similar to the target objects in the rest of the video. OSVOS [2] transfers the generic semantic information to the task of foreground segmentation by fine-tuning the network on the first frame. RGMP [20] takes the mask of the previous frame as input, which provides a spatial prior for the current frame. PML [4] learns a pixel-wise embedding using a triplet loss and assigns a label to each pixel by nearest neighbor matching in pixel space to the first frame. VideoMatch [11] uses a soft matching layer that maps the pixels of the current frame to the first frame in the learned embedding space. Following PML and VideoMatch, FEELVOS [32] extends the pixel-level matching mechanism by additionally matching between the current frame and the first frame. Based on FEELVOS, CFBI [41] promotes the results by explicitly considering the feature matching of the target foreground object and the corresponding background. STM [22] leverages a memory network to perform pixel-level matching from past frames, which outperforms all previous methods. Based on STM, KMN [28] applies a Gaussian kernel to reduce the mismatched pixels. EGMN [17] employs an episodic memory network to store frames as nodes and capture cross-frame correlations by edges. Other STM-based methods [36, 42] use depth estimation and spatial constraint modules to improve the accuracy of STM. However, matching-based methods perform unnecessary matching in regions without target objects and are less effective to distinguish objects with similar appearance.

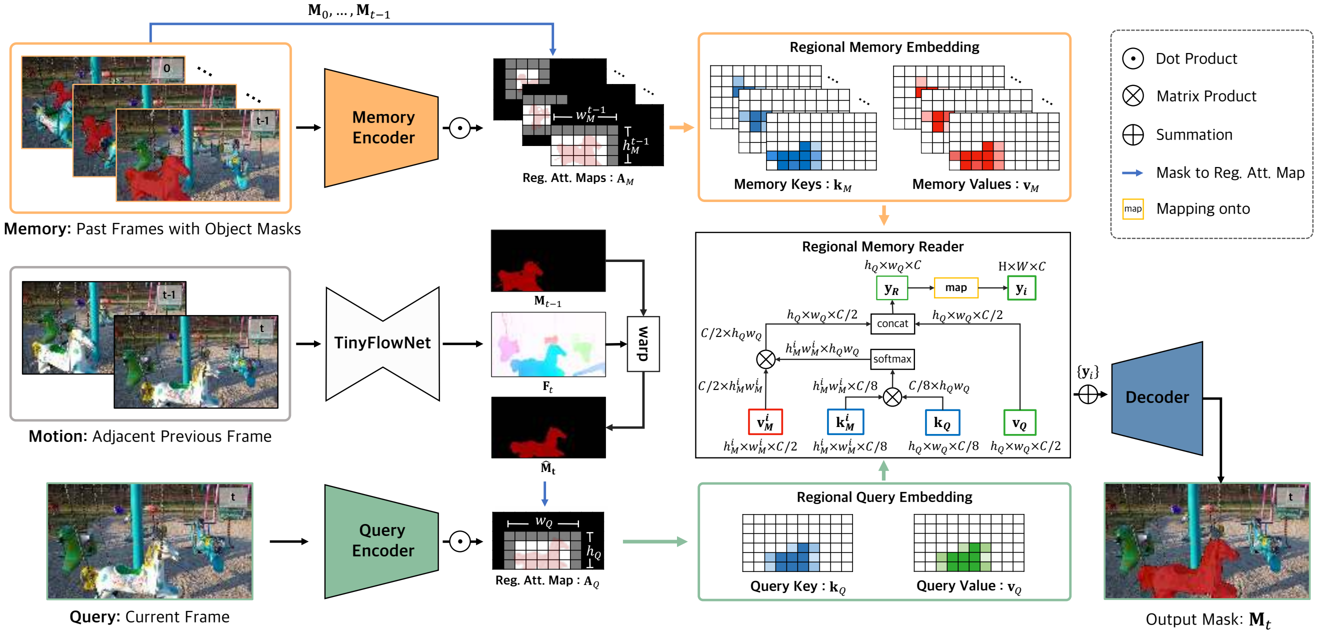
$\mathbf{M}_0, \dots, \mathbf{M}_{t-1}$
0
t-1
Memory: Past Frames with Object Masks
Memory Encoder
w_M^{t-1}
h_M^{t-1}
Reg. Att. Maps : $\mathbf{A}_M$
Regional Memory Embedding
Memory Keys : $\mathbf{k}_M$
Memory Values : $\mathbf{v}_M$
Dot Product
Matrix Product
Summation
Mask to Reg. Att. Map
map
Mapping onto
t-1
t
Motion: Adjacent Previous Frame
TinyFlowNet
$\mathbf{M}_{t-1}$
$\mathbf{F}_t$
warp
$\hat{\mathbf{M}}_t$
Regional Memory Reader
$h_Q \times w_Q \times C$
$\mathrm{H} \times W \times C$
$\mathbf{y}_R$
map
$\mathbf{y}_i$
$h_Q \times w_Q \times C/2$
concat
$h_Q \times w_Q \times C/2$
$C/2 \times h_Q w_Q$
$h_M^i w_M^i \times h_Q w_Q$
softmax
$C/2 \times h_M^i w_M^i$
$h_M^i w_M^i \times C/8$
$C/8 \times h_Q w_Q$
$\mathbf{v}_M^i$
$\mathbf{k}_M^i$
$\mathbf{k}_Q$
$\mathbf{v}_Q$
$h_M^i \times w_M^i \times C/2$
$h_M^i \times w_M^i \times C/8$
$h_Q \times w_Q \times C/8$
$h_Q \times w_Q \times C/2$
$\{\mathbf{y}_i\}$
Decoder
t
Query: Current Frame
Query Encoder
w_Q
h_Q
Reg. Att. Map : $\mathbf{A}_Q$
Regional Query Embedding
Query Key : $\mathbf{k}_Q$
Query Value : $\mathbf{v}_Q$
t
Output Mask: $\mathbf{M}_t$

Figure 3. Overview of RMNet. The proposed network considers the object motion for the current frame and the object cues from the past frames in memory. To alleviate the mismatching to similar objects, the regional memory and query embedding are extracted from the regions containing target objects. Regional Memory Reader efficiently performs local-to-local matching only in these regions. Note that "Reg. Att. Map" denotes "Regional Attention Map".

3. Regional Memory Network

The architecture of the proposed Regional Memory Network (RMNet) is shown in Figure 3. As in STM [22], the current frame is used as the query, and the past frames with the estimated masks are used as the memory. Different from STM that constructs the global memory and query embedding from all regions, RMNet only embeds the regions containing target objects in the memory and query frames. The regional memory and query embedding are generated by the dot product of the regional attention maps and feature embedding extracted from the memory and query encoders, respectively. Both of them consist of a regional **key** and a regional **value**.

In STM, the space-time memory reader is employed for global-to-global matching between all pixels in the memory and query frames. However, Regional Memory Reader in RMNet is proposed for local-to-local matching between the regional memory embedding and query embedding in the regions containing target objects, which alleviates the mismatching to similar objects and also accelerates the computation. Given the output of Regional Memory Reader, the decoder predicts the object masks for the query frame.

3.1. Regional Feature Embedding

3.1.1 Regional Memory Embedding

Recent Space-Time Memory based methods [17, 22, 28] construct the global memory embedding for the past frames by using the features of the whole images. However, the features outside the regions where the target objects appear may lead to the mismatching to the similar objects in the query frame, as shown with the red solid line in Figure 2. To solve this issue, we present the regional memory embedding that only memorizes the features in the regions containing the target objects.

Mask to Regional Attention Map. To generate the regional memory embedding, we apply a regional attention map to the global memory embedding. At the time step i, given the object mask $\mathbf{M}_i \in \mathbb{N}^{H \times W}$ at the feature scale, the regional attention map $\mathbf{A}_i^j \in \mathbb{R}^{H \times W}$ for the j-th object is obtained as

$$\mathbf{A}_i^j(x, y) = \begin{cases} 1, & x_{\min} \leq x \leq x_{\max} \text{ and } y_{\min} \leq y \leq y_{\max} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $(x_{\min}, y_{\min})$ and $(x_{\max}, y_{\max})$ are the top-left and bottom-right coordinates of the bounding box for the target object, which are determined by

$$\begin{aligned} x_{\min} &= \max((\arg\min_x \mathbf{M}_i(x, y) = j) - \phi, 0) \\ x_{\max} &= \min((\arg\max_x \mathbf{M}_i(x, y) = j) + \phi, W) \\ y_{\min} &= \max((\arg\min_y \mathbf{M}_i(x, y) = j) - \phi, 0) \\ y_{\max} &= \min((\arg\max_y \mathbf{M}_i(x, y) = j) + \phi, H) \end{aligned} \tag{2}$$

where ϕ denotes the padding of the bounding box, which

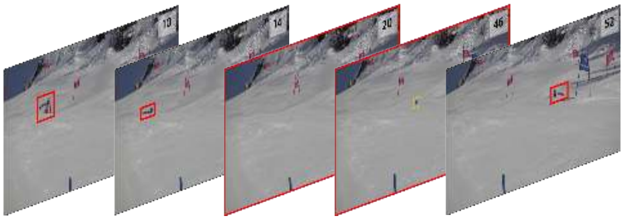

Figure 4. The changes of matching regions for the target object before and after occlusion, which is highlighted by red bounding boxes for each frame.

determines the error tolerance of the estimated masks in the past frames. Specially, we define $\mathbf{A}_i^j = 0$ if the j-th object disappears in $\mathbf{M}_i$.

Regional Memory Key/Value. Given the regional attention map $\mathbf{A}_M^j = [\mathbf{A}_0^j, \ldots, \mathbf{A}_{t-1}^j]$ of j-th object in the memory frames, the **key** $\mathbf{k}_M^j$ and **value** $\mathbf{v}_M^j$ in the regional memory embedding are obtained by the dot product of $\mathbf{A}_M^j$ and the global memory embedding from the memory encoder.

3.1.2 Regional Query Embedding

As illustrated by the red dotted line in Figure 2, multiple similar objects in the query frame are easily mismatched in the global-to-global matching. Similar to the regional memory embedding, we present the regional query embedding that alleviates the mismatching to the similar objects in the query frame.

Object Mask Tracking. To obtain the possible regions of target objects in the current frame, we track and predict a rough mask $\hat{\mathbf{M}}_t^j$. Specifically, we warp the mask $\mathbf{M}_{t-1}^j$ of the previous frame with the optical flow $\mathbf{F}_t$ estimated by the proposed TinyFlowNet.

Mask to Regional Attention Map. As in regional memory embedding (Sec. 3.1.1), the estimated mask $\hat{\mathbf{M}}_t^j$ is used to generate the regional attention map $\mathbf{A}_Q^j$ for the j-th object in the query frame. To better deal with occlusions, we define $\mathbf{A}_Q^j = 1$ if the number of pixels is lower than a threshold η, which triggers the global matching for the target object in the query frame. As illustrated in Figure 4, the matching region is expanded to the whole image when the target object disappears. Then, it shrinks to the region containing the target object when the object appears again. This mechanism benefits the optical flow based tracking, which allows the network to perceive the disappearance of objects and makes it more robust to object occlusion.

Regional Query Key/Value. Similar to regional memory embedding, the **key** $\mathbf{k}_Q^j$ and **value** $\mathbf{v}_Q^j$ in the regional query embedding are obtained by the dot product of $\mathbf{A}_Q^j$ and the global query embedding from the query encoder.

3.2. Regional Memory Reader

In STM [22], the space-time memory reader is proposed to measure the similarities between the pixels in the query frame and memory frames. Given the embedded **key** of the memory $\mathbf{k}_M^j = \{k_M^j(\mathbf{p})\} \in \mathbb{R}^{T \cdot H \cdot W \times C/8}$ and the query $\mathbf{k}_Q^j = \{k_Q^j(\mathbf{q})\} \in \mathbb{R}^{H \cdot W \times C/8}$ of the j-th object, the similarity between $\mathbf{p}$ and $\mathbf{q}$ can be computed as

$$s^j(\mathbf{p}, \mathbf{q}) = \exp\left(k_M^j(\mathbf{p}) k_M^j(\mathbf{q})^{\mathrm{T}}\right) \tag{3}$$

where C and T denote the channels of the embedded key and the number of frames in memory, respectively. Let $\mathbf{p} = [p_t, p_x, p_y]$ and $\mathbf{q} = [q_x, q_y]$ be the grid cell locations in $\mathbf{k}_M^j$ and $\mathbf{k}_Q^j$, respectively. Then, the query at position $\mathbf{q}$ retrieves the corresponding **value** from the memory by

$$v^j(\mathbf{q}) = \sum_{\mathbf{p}} \frac{s(\mathbf{p}, \mathbf{q})}{\sum_{\mathbf{p}} s(\mathbf{p}, \mathbf{q})} v_M^j(\mathbf{p}) \tag{4}$$

where $\mathbf{v}_M^j = \{v_M^j(\mathbf{p})\} \in \mathbb{R}^{T \times H \times W \times C/2}$ is the embedded **value** of the memory. The output of the space-time memory reader at position $\mathbf{q}$ is

$$y^j(\mathbf{q}) = \left[v_Q^j(\mathbf{q}), v^j(\mathbf{q})\right] \tag{5}$$

where $\mathbf{v}_Q^j = \{v_Q^j(\mathbf{q})\} \in \mathbb{R}^{H \times W \times C/2}$ denotes the embedded **value** of the query and $[\cdot]$ represents the concatenation.

Based on the regional feature embedding, we propose Regional Memory Reader, which performs local-to-local matching in the regions containing the target objects, as shown in Figure 3. Compared to the global-to-global memory readers in [17, 21, 28], the proposed Regional Memory Reader alleviates the mismatching to the similar objects in both the memory and query frames.

Let $\mathcal{R}_M^j = \{\mathbf{p}\}$ and $\mathcal{R}_Q^j = \{\mathbf{q}\}$ be the feature matching regions of the j-th object in the memory and query frames, respectively. In the global-to-global memory readers, the similarities are derived by a large matrix product. That is, $\mathcal{R}_M^j$ and $\mathcal{R}_Q^j$ are all locations in the feature embedding of **key** and **value**, respectively. While in the proposed Regional Memory Reader, $\mathcal{R}_M^j$ and $\mathcal{R}_Q^j$ are defined as

$$\begin{aligned} \mathcal{R}_M^j &= \left\{\mathbf{p} | \mathbf{A}_M^j(\mathbf{p}) \neq 0\right\} \\ \mathcal{R}_Q^j &= \left\{\mathbf{q} | \mathbf{A}_Q^j(\mathbf{q}) \neq 0\right\} \end{aligned} \tag{6}$$

Therefore, for locations $\mathbf{p} \notin \mathcal{R}_M^j$ or $\mathbf{q} \notin \mathcal{R}_Q^j$, the similarity between $\mathbf{p}$ and $\mathbf{q}$ is defined as

$$s^j(\mathbf{p}, \mathbf{q}) = 0, \mathbf{p} \notin \mathcal{R}_M^j \text{ or } \mathbf{q} \notin \mathcal{R}_Q^j \tag{7}$$

Let h_Q^j and w_Q^j be the height and width of the region for the j-th object in the query frame, respectively. h_M^j and w_M^j denote the maximum height and width of the regions in the memory frames, respectively. Therefore, the time complexity of the space-time memory reader is $\mathcal{O}(TCH^2W^2)$. Compared to the space-time memory reader, the proposed

Regional Memory Reader is computationally efficient with the time complexity of $\mathcal{O}(TCh_Q^j w_Q^j h_M^j w_M^j)$. As shown in Figure 6, $h_Q^j, h_M^j \ll H$ and $w_Q^j, w_M^j \ll W$. Actually, the space-time memory reader is also a non-local network [35], which usually suffers from high computational complexity [43] due to the global-to-global feature matching. The proposed local-to-local matching enables the time complexity of the memory reader to be significantly reduced.

3.3. Network Architecture

TinyFlowNet. Compared to existing methods for optical flow estimation [8, 27, 29, 37], TinyFlowNet does not use any time-consuming layers such as correlation layer [8], cost volume layer [29, 37], and dilated convolutional layers [27]. To reduce the number of parameters of TinyFlowNet, we use small numbers of input and output channels. Consequently, TinyFlowNet is 1/3 the size of FlowNetS [12]. To further accelerate the computation, the input images are downsampled by 2 before fed into TinyFlowNet.

Encoder. The memory encoder takes an RGB frame along with the object mask as input, in which the object mask is represented as a single channel probability map between 0 and 1. The input to the query encoder is only the RGB frame. Both the memory and query encoders use ResNet50 [9] as the backbone network. To take a 4-channel tensor, the number of the input channels of the first convolutional layer in the memory encoder is changed to 4. The first convolutional layer in the query encoder remains unchanged as in ResNet50. The output **key** and **value** features are embedded by two parallel convolutional layers attached to the convolutional layer that outputs a $1/16$ resolution feature with respect to the input image.

Decoder. The decoder takes the output of Regional Memory Reader and predicts the object mask for the current frame. The decoder consists of a residual block and two stacks of refinement modules [22] that gradually upscale the compressed feature map to the size of the input image.

4. Experiments

4.1. Datasets

DAVIS. DAVIS 2016 [25] is one of the most popular benchmarks for video object segmentation, whose validation set is composed of 20 videos annotated with high-quality masks for individual objects. DAVIS 2017 [26] is a multi-object extension of DAVIS 2016. The training and validation sets contain 60 and 30 videos, respectively.

YouTube-VOS. YouTube-VOS [38] is the latest large-scale dataset for the video object segmentation, which contains 4,453 videos annotated with multiple objects. Specifically, YouTube-VOS contains 3,471 videos from 65 categories for training and 507 videos with additional 26 unseen categories for validation.

Table 1. The quantitative evaluation on the DAVIS 2016 validation set. † indicates using YouTube-VOS for training. The time is measured on an NVIDIA Tesla V100 GPU without I/O time.

Methods	$\mathcal{J}$ Mean	$\mathcal{F}$ Mean	Avg.	Time (s)
OnAVOS [33]	0.861	0.849	0.855	0.823
OSVOS [2]	0.798	0.806	0.802	0.642
MaskRNN [10]	0.807	0.809	0.808	-
RGMP [20]	0.815	0.820	0.818	0.104
FAVOS [5]	0.824	0.795	0.810	0.816
CINN [1]	0.834	0.850	0.842	-
LSE [7]	0.829	0.803	0.816	-
VideoMatch [11]	0.810	0.808	0.819	-
PReMVOS [18]	0.849	0.886	0.868	3.286
A-GAME † [13]	0.822	0.820	0.821	0.258
FEELVOS † [32]	0.817	0.881	0.822	0.286
STM † [22]	0.887	0.899	0.893	0.097
KMN † [28]	**0.895**	**0.915**	**0.905**	-
CFBI † [41]	0.883	0.905	0.894	0.156
RMNet	0.806	0.823	0.815	**0.084**
RMNet †	0.889	0.887	0.888	**0.084**

4.2. Evaluation Metrics

Following the previous works [22, 41], we take the region similarity and contour accuracy as evaluation metrics.

Region Similarity $\mathcal{J}$. We employ the region similarity $\mathcal{J}$ to measure the region-based segmentation similarity, which is defined as the intersection-over-union of the estimated segmentation and the ground truth segmentation. Given an estimated segmentation M and the corresponding ground truth mask G, the region similarity $\mathcal{J}$ is defined as

$$\mathcal{J} = \left|\frac{M \cap G}{M \cup G}\right| \tag{8}$$

Contour Accuracy $\mathcal{F}$. Let $c(M)$ be the set of the closed contours that delimits the spatial extent of the mask M. The contour points of the estimated mask M and ground truth G are denoted as $c(M)$ and $c(G)$, respectively. The precision P_c and recall R_c between $c(M)$ and $c(G)$ can be computed by a bipartite graph matching [19]. Therefore, the contour accuracy $\mathcal{F}$ between $c(M)$ and $c(G)$ is defined as

$$\mathcal{F} = \frac{2P_cR_c}{P_c + R_c} \tag{9}$$

4.3. Implementation Details

We implement the network using PyTorch [23] and CUDA. All models are optimized using the Adam optimizer [15] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Following [22, 28], the network is trained in two phases. First, it is pretrained on the synthetic dataset generated by applying random affine transforms to a static image with different pa-

Table 2. The quantitative evaluation on the DAVIS 2017 validation set. † indicates using YouTube-VOS for training.

Methods	$\mathcal{J}$ Mean	$\mathcal{F}$ Mean	Avg.
OnAVOS [33]	0.645	0.713	0.679
OSMN [40]	0.525	0.571	0.548
OSVOS [2]	0.566	0.618	0.592
RGMP [20]	0.648	0.686	0.632
FAVOS [5]	0.546	0.618	0.582
CINN [1]	0.672	0.742	0.707
VideoMatch [11]	0.565	0.682	0.624
PReMVOS [18]	0.739	0.817	0.778
A-GAME † [13]	0.672	0.727	0.700
FEELVOS † [32]	0.691	0.740	0.716
STM † [22]	0.792	0.843	0.818
KMN † [28]	0.800	0.856	0.828
EGMN † [17]	0.800	0.859	0.829
CFBI † [41]	0.791	0.846	0.819
RMNet	0.728	0.772	0.750
RMNet †	**0.810**	**0.860**	**0.835**

Table 3. The quantitative evaluation on the DAVIS 2017 test-dev set.

Methods	$\mathcal{J}$ Mean	$\mathcal{F}$ Mean	Avg.
OnAVOS [33]	0.534	0.596	0.565
OSMN [40]	0.377	0.449	0.413
RGMP [20]	0.513	0.544	0.529
PReMVOS [18]	0.675	0.757	0.716
FEELVOS [32]	0.552	0.605	0.578
STM [22]	0.680	0.740	0.710
CFBI [41]	0.711	**0.785**	0.748
RMNet	**0.719**	0.781	**0.750**

rameters. Then, it is fine-tuned on DAVIS and YouTube-VOS. The parameters ϕ and η are set to 4 and 10, respectively. For all experiments, the network is trained with a batch size of 4 on two NVIDIA Tesla V100 GPUs. All batch normalization layers are fixed during training and testing. The initial learning rate is set to 10^{-5} and the optimization is set to stop after 200 epochs.

4.4. Video Object Segmentation on DAVIS

Single object. We compare proposed RMNet with state-of-the-art methods on the validation set of the DAVIS 2016 dataset for single-object video segmentation. DAVIS contains only a small number of videos, which leads to overfitting and affects the generalization ability. Following the latest works [5, 22, 28, 41], we also present the results trained with additional data from YouTube-VOS, which are denoted as † in Table 1. The experimental results indicate that the proposed RMNet is comparable to other competitive methods but with a faster inference speed.

Table 4. The quantitative evaluation on the YouTube-VOS validation set (2018 version). The results of other methods are directly copied from [22, 41].

Methods	Seen		Unseen		Avg.
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	
OnAVOS [33]	0.601	0.627	0.466	0.514	0.552
OSMN [40]	0.600	0.601	0.406	0.440	0.512
OSVOS [2]	0.598	0.605	0.542	0.607	0.588
RGMP [20]	0.595	-	0.452	-	0.538
BoLTVOS [34]	0.716	-	0.643	-	0.711
PReMVOS [18]	0.714	0.759	0.565	0.637	0.669
A-GAME [13]	0.678	-	0.608	-	0.661
STM [22]	0.797	0.842	0.728	0.809	0.794
KMN [28]	0.814	0.856	0.753	0.833	0.814
EGMN [17]	0.807	0.851	0.740	0.809	0.802
CFBI [41]	0.811	**0.858**	0.753	**0.834**	0.814
RMNet	**0.821**	0.857	**0.757**	0.824	**0.815**

Multiple objects. To evaluate the performance of multi-object video segmentation, we test the proposed RMNet on the DAVIS 2017 benchmark. We report the performance of the val set of DAVIS 2017 in Table 2, which shows that RMNet outperforms all competitive methods. With additional YouTube-VOS data, RMNet archives a better accuracy and outperforms all state-of-the-art methods. We also evaluate RMNet on the test-dev set of DAVIS 2017, which is much more challenging than the val set. As shown in Table 3, RMNet surpasses the state-of-the-art methods. The qualitative results on DAVIS 2017 are shown in Figure 1.

4.5. Video Object Segmentation on YouTube-VOS

Following the latest works [22, 28], we compare the proposed RMNet to the state-of-the-art methods on the YouTube-VOS validation set (2018 version). As shown in Table 4, RMNet achieves an average score of 0.815, which outperforms other methods. The qualitative results on the YouTube-VOS dataset are shown in Figure 5, which demonstrate that RMNet is more effective in distinguishing similar objects and performs better in segmenting small objects.

4.6. Ablation Study

To demonstrate the effectiveness of each component in the proposed RMNet, we conduct the ablation study on the DAVIS 2017 val set.

Regional Memory Reader. The proposed Regional Memory Reader performs regional feature matching in both the memory and query frames, which reduces the number of mismatched pixels and therefore saves computational time. To evaluate the effectiveness and efficiency of Regional Memory Reader, we replace the regional feature matching with global matching in the query frame and memory

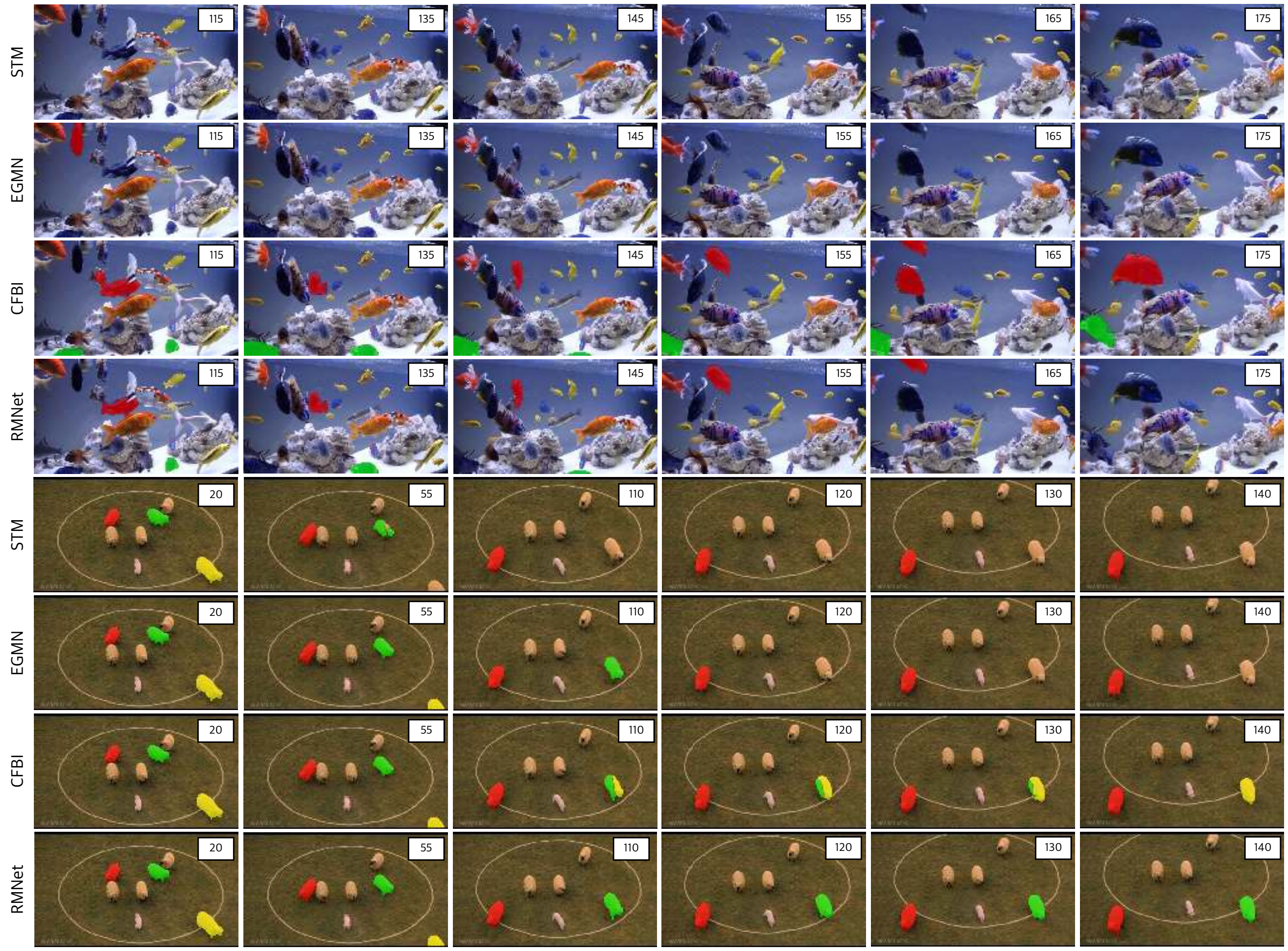

Figure 5. The challenging examples of multi-object video segmentation on the YouTube-VOS validation set (2018 version). All methods are tested at 720p without test-time augmentation. For the first video, both STM [22] and EGMN [17] fail to segment the target objects before the 115-th frame.

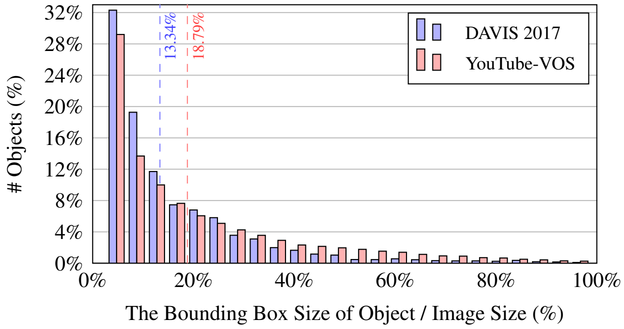

Figure 6. The long tail distribution of the area ratio of the bounding boxes of target objects on the training set of the DAVIS 2017 and YouTube-VOS datasets. The vertical dashed lines denote the mean value of the area ratio of bounding boxes for both datasets.

frame. Compared to the memory readers that adopt global matching for memory or query frames, the proposed Regional Memory Reader achieves better results in terms of both accuracy and efficiency. As shown in Figure 6, the areas of the regions containing target objects are usually smaller than 20% of the whole image. Table 5 shows that the local-to-local matching in Regional Memory Reader is about 5 times faster (around 25 times smaller in FLOPS) than the global-to-global matching. Figure 7 presents the visualization of the similarity scores of the target object in the memory readers, where the target object is highlighted in a red bounding box. Given the estimated mask of the target object in the query frame, we compute the similarity scores in the previous frame, where the label of the target object in the query frame is determined by the pixels with high similarities. Similarly, the pixels with similarity scores in the query frame are assigned with the label of the target object in the previous frame. As shown in Figure 7, the proposed Regional Memory Reader avoids the mismatching in the regions outside the target object in memory and query frames, which obtains better segmentation results.

Query Region Prediction. In RMNet, the regions for feature matching in the query frame are determined by

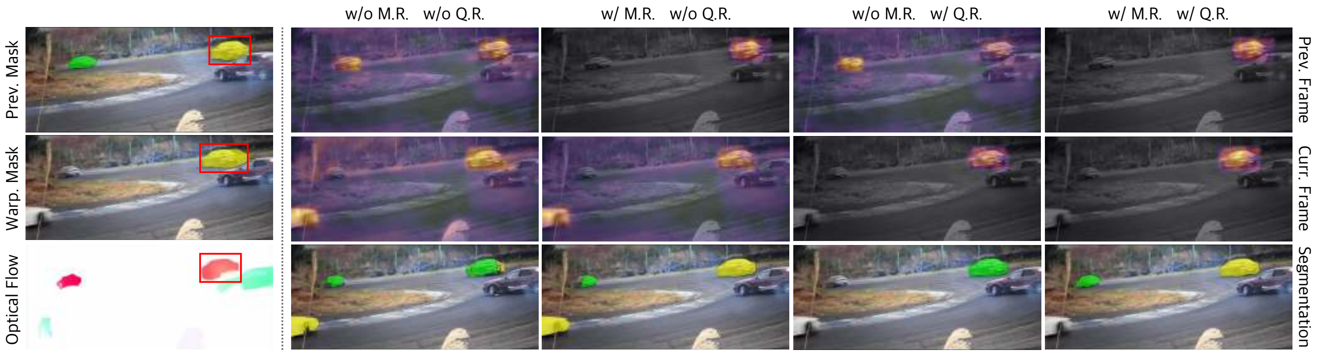

Figure 7. The similarity scores of the target object in the previous and current frames. The target object is highlighted by a red bounding box. "M.R." and "Q.R." denote for "Memory Region" and "Query Region" in the regional memory and query embedding, respectively.

Table 5. The effectiveness of Regional Memory Reader. "M.R." and "Q.R." denote for "Memory Region" and "Query Region", where ✓ and × represent the feature matching is regional or global for the frame, respectively. The time for feature matching is measured on an NVIDIA Tesla V100 GPU without I/O time.

M.R.	Q.R.	$\mathcal{J}$ Mean	$\mathcal{F}$ Mean	Avg.	Time (ms)
×	×	0.792	0.843	0.818	10.68
✓	×	0.798	0.847	0.822	5.50
×	✓	0.803	0.853	0.828	5.50
✓	✓	**0.810**	**0.860**	**0.835**	**2.09**

Table 6. The effectiveness of the "Flow-based Region" compared to "Previous Region" and "Best-match Region" used in FEELVOS [32] and KMN [28], respectively.

Method	$\mathcal{J}$ Mean	$\mathcal{F}$ Mean	Avg.
Previous Region	0.762	0.822	0.792
Best-match Region	0.792	0.845	0.819
Flow-based Region	**0.810**	**0.860**	**0.835**

Table 7. The effectiveness of TinyFlowNet compared to FlowNet2-CSS [12] and RAFT [30] for optical flow estimation. The time for optical flow estimation is measured on an NVIDIA Tesla V100 GPU without I/O time.

Method	$\mathcal{J}$ Mean	$\mathcal{F}$ Mean	Avg.	Time (ms)
FlowNet2-CSS	**0.814**	**0.860**	**0.837**	59.93
RAFT	0.808	0.859	0.834	157.78
TinyFlowNet	0.810	**0.860**	0.835	**10.05**

the previous mask and the estimated optical flow. In FEELVOS [32], the regions for local feature matching are a local neighborhood of the locations where the target objects appear in the previous frame. KMN [28] performs feature matching in the regions determined by a 2D Gaussian kernel whose center is the best-matched pixel with the highest similarity score. To evaluate the effectiveness of our "Flow-based Region", we compare its performance with different regions used for feature matching. In Table 6, "Previous Region" and "Best-match Region" represent the regions determined by the methods used in FEELVOS and KMN, respectively. As shown in Table 6, "Flow-based Region" outperforms "Previous Region" and "Best-match Region" in segmentation. "Previous Region" is based on the assumption that the motion between two frames is usually small, which is not robust to object occlusion and drifting. "Best-match Region" only considers the region determined by the best-matched pixels. However, the best-matched pixels are easily affected by lighting conditions and may be wrong for similar objects.

TinyFlowNet. TinyFlowNet is designed to estimate optical flow between two adjacent frames. To evaluate the effectiveness of the proposed TinyFlowNet, we replace the TinyFlowNet with FlowNet2-CSS [12] and RAFT [30] that are pretrained on the FlyingChairs [8]. As shown in Table 7, the segmentation accuracies are almost the same when TinyFlowNet is replaced with FlowNet2-CSS and RAFT, which indicates that TinyFlowNet meets the need of region prediction in the query frame. Moreover, the proposed RMNet with TinyFlowNet is 6 and 16 times faster than with FlowNet2-CSS and RAFT, respectively.

5. Conclusion

In this paper, we propose Regional Memory Network (RMNet) for semi-supervised VOS. Compared to the STM-based methods, RMNet memorizes and tracks the regions containing target objects, which effectively alleviates the ambiguity of similar objects and also reduces the computational complexity for feature matching. Experimental results on DAVIS and YouTube-VOS indicate that the proposed method outperforms the state-of-the-art methods with much faster running speed.

Acknowledgements This work is conducted in collaboration with SenseTime. This work is in part supported by A*STAR through the Industry Alignment Fund - Industry Collaboration Projects Grant, in part by National Natural Science Foundation of China (Nos. 61772158, 61872112, and U1711265), and in part by National Key Research and Development Program of China (Nos. 2018YFC0806802 and 2018YFC0832105).

References

[1] Linchao Bao, Baoyuan Wu, and Wei Liu. CNN in MRF: video object segmentation via inference in a cnn-based higher-order spatio-temporal MRF. In *CVPR*, 2018.

[2] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017.

[3] Xi Chen, Zuoxin Li, Ye Yuan, Gang Yu, Jianxin Shen, and Donglian Qi. State-aware tracker for real-time video object segmentation. In *CVPR*, 2020.

[4] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, 2018.

[5] Jingchun Cheng, Yi-Hsuan Tsai, Wei-Chih Hung, Shengjin Wang, and Ming-Hsuan Yang. Fast and accurate online video object segmentation via tracking parts. In *CVPR*, 2018.

[6] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. SegFlow: Joint learning for video object segmentation and optical flow. In *ICCV*, 2017.

[7] Hai Ci, Chunyu Wang, and Yizhou Wang. Video object segmentation by learning location-sensitive embeddings. In *ECCV*, 2018.

[8] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *ICCV*, 2015.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR 2016*, 2016.

[10] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G. Schwing. MaskRNN: Instance level video object segmentation. In *NIPS*, 2017.

[11] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G. Schwing. VideoMatch: Matching based video object segmentation. In *ECCV*, 2018.

[12] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017.

[13] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. A generative appearance model for end-to-end video object segmentation. In *CVPR*, 2019.

[14] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for video object segmentation. *IJCV*, 127(9):1175–1197, 2019.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[16] Xiaoxiao Li and Chen Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *ECCV*, 2018.

[17] Xiankai Lu, Wenguan Wang, Martin Danelljan, Tianfei Zhou, Jianbing Shen, and Luc Van Gool. Video object segmentation with episodic graph memory networks. In *ECCV*, 2020.

[18] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. PReMVOS: Proposal-generation, refinement and merging for video object segmentation. In *ACCV*, 2018.

[19] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *TPAMI*, 26(5):530–549, 2004.

[20] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, 2018.

[21] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Fast user-guided video object segmentation by interaction-and-propagation networks. In *CVPR*, 2019.

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

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith and Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

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

[25] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus H. Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

[26] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbelaez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv*, 1704.00675, 2017.

[27] René Schuster, Oliver Wasenmüller, Christian Unger, and Didier Stricker. SDC - stacked dilated convolution: A unified descriptor network for dense matching tasks. In *CVPR*, 2019.

[28] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In *ECCV*, 2020.

[29] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.

[30] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, 2020.

[31] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J. Black. Video segmentation via object flow. In *CVPR*, 2016.

[32] 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 *CVPR*, 2019.

[33] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017.

[34] Paul Voigtlaender, Jonathon Luiten, and Bastian Leibe. Boltvos: Box-level tracking for video object segmentation. *arXiv*, 1904.04552, 2019.

[35] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR 2018*, 2018.

[36] Haozhe Xie, Yunmu Huang, Anni Xu, Jinpeng Lan, and Wenxiu Sun. Depth-aware space-time memory network for video object segmentation. In *CVPR Workshops*, 2020.

[37] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate optical flow via direct cost volume processing. In *CVPR*, 2017.

[38] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian L. Price, Scott Cohen, and Thomas S. Huang. YouTube-VOS: Sequence-to-sequence video object segmentation. In *ECCV*, 2018.

[39] Yu-Syuan Xu, Tsu-Jui Fu, Hsuan-Kung Yang, and Chun-Yi Lee. Dynamic video segmentation network. In *CVPR*, 2018.

[40] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K. Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018.

[41] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, 2020.

[42] Peng Zhang, Li Hu, Bang Zhang, and Pan Pan. Deep residual learning for image recognition. In *CVPR Workshops*, 2020.

[43] Zhen Zhu, Mengdu Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *ICCV*, 2019.