Unsupervised Video Object Segmentation via Prototype Memory Network

Minhyeok Lee[1] Suhwan Cho[1] Seunghoon Lee[1] Chaewon Park[1] Sangyoun Lee[1,2,*]
[1] Yonsei University [2] Korea Institute of Science and Technology (KIST)
{hydragon516,chosuhwan,shlee423,chaewon28,syleee}@yonsei.ac.kr

Abstract

Unsupervised video object segmentation aims to segment a target object in the video without a ground truth mask in the initial frame. This challenging task requires extracting features for the most salient common objects within a video sequence. This difficulty can be solved by using motion information such as optical flow, but using only the information between adjacent frames results in poor connectivity between distant frames and poor performance. To solve this problem, we propose a novel prototype memory network architecture. The proposed model effectively extracts the RGB and motion information by extracting superpixel-based component prototypes from the input RGB images and optical flow maps. In addition, the model scores the usefulness of the component prototypes in each frame based on a self-learning algorithm and adaptively stores the most useful prototypes in memory and discards obsolete prototypes. We use the prototypes in the memory bank to predict the next query frame's mask, which enhances the association between distant frames to help with accurate mask prediction. Our method is evaluated on three datasets, achieving state-of-the-art performance. We prove the effectiveness of the proposed model with various ablation studies.

1. Introduction

Video object segmentation (VOS) aims to delineate pixel-level salient object masks in each frame. VOS is used as preprocessing for video captioning [45], interactive segmentation [39], and optical flow estimation [6]. It is also widely applied in robotics and autonomous vehicles [1, 24, 29].

VOS tasks are divided into semi-supervised [7, 31, 52] and unsupervised VOS [13, 43, 63, 15] depending on whether the ground truth mask of the first frame of the video sequence is provided. To be more specific, semi-supervised VOS aims to track and segment a specified object in the initial frame of a video. However, in an unsupervised VOS task, the model must track and segment the most salient objects without being given a specified mask in the first frame.

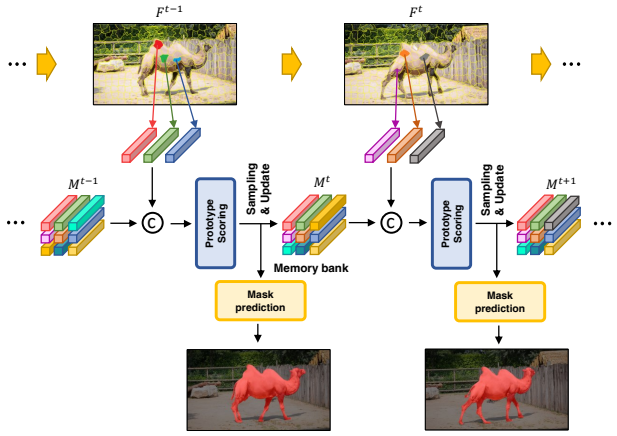

Figure 1. The overall flow of the proposed method. We split the image into superpixels and create prototypes covering each superpixel area. Prototypes are evaluated for their usefulness along with the previous prototypes in the memory bank, and the most useful prototypes are stored in the memory bank for mask prediction in the next sequence.

Therefore, an unsupervised VOS task is very challenging because it is important to search for common objects in the input video sequence and effectively extract their features.

To extract common, consistent features, traditional handcrafted methods [33, 11, 62, 49, 32] applied temporal trajectory, saliency prior, and object proposal techniques. However, these methods perform poorly in complex morphological variations of the target object over time and in extreme lighting conditions. To solve this problem, deep-learning-based unsupervised VOS models [13, 43, 63, 15, 19, 5] have recently been in the spotlight. In particular, many models [13, 43, 63, 15] extract additional motion information from the optical flow and use it as a guideline for common objects. However, because these models generate optical flow maps between two adjacent frames, they ignore the feature associations between long-distance frames, resulting in poor performance. It is also difficult to fuse the two features effectively because of the large domain gap between the RGB image and the optical flow. To solve these

problems, Schmidtt *et al*. [37] applied 3D convolution to make the model learn long-distance frame dependencies, but this method cannot perform real-time prediction.

We propose a novel prototype memory network (PMN) to address the aforementioned difficulties of unsupervised VOS. Figure 1 shows the overall flow of the proposed method. Many studies of segmentation tasks [44, 3, 54] have shown that preprocessing using superpixels can provide useful features to models and improve performance by clustering image pixels. Therefore, we first divide the RGB images and optical flow maps into superpixels using a simple linear iterative clustering (SLIC) algorithm [2] to effectively extract various detail and texture information from the RGB images and motion information from the optical flow maps. We then create component prototypes from the superpixel mask, focusing on prototype learning, which is widely used in few-shot segmentation tasks [9, 46, 26]. We also propose a prototype scoring module (PSM) and memory bank to enhance common feature associations between distant frames. The PSM scores the usefulness of the generated prototypes and samples only the most highly scored prototype features. The prototypes selected by the PSM are stored in a memory bank, and these prototypes are combined with the prototypes generated from the next frame image. The PSM updates the memory bank at every frame by giving the prototypes new scores. Therefore, the memory banks can store useful features for target objects from past frames so the model can use them for prediction in future frames. The proposed PSM is trained with self-learning techniques because the usefulness scores for the prototypes are not labeled manually.

We tested our method on three popular datasets: DAVIS16 [34], FBMS [32], and YouTube-Objects [35]. These datasets contain various challenging scenarios, and the proposed model achieves state-of-the-art performance on all three datasets. In addition, we performed various ablation studies to prove the effectiveness of the proposed model and show that robust VOS is possible in challenging video sequences.

Our main contributions can be summarized as follows:

- We propose a novel PMN to extract detailed information from the RGB image and motion information from the optical flow and strengthen the connectivity between distant frames. Inspired by the prototype learning used in few-shot segmentation tasks, the proposed model generates component prototypes based on the superpixel algorithm.

- We propose a PSM to score the usefulness of the prototypes generated and update the most highly scored prototypes in memory. The PSM is self-supervised because prototype utility scores cannot be labeled.

- The proposed network achieves state-of-the-art performance on the DAVIS-16 [34], FBMS [32], and YouTube-Objects [35] datasets. Additionally, we demonstrate the effectiveness of the proposed method through various ablation studies.

2. Related Work

Unsupervised VOS. Unsupervised VOS aims to segment eye-catching objects in a video sequence without human intervention. This is an extension of previous single-image salient object detection tasks and is more challenging because it detects common salient objects within a video sequence. Traditional methods [33, 11, 62, 49, 32] use motion boundaries, long-term point trajectories, and objectness to segment common objects. However, these methods often fail due to the occlusion of the target object, illumination extremes, and complex foreground and background structures. To perform robust VOS in such challenging situations, deep-learning methods have been in the spotlight recently. In particular, methods [13, 43, 63, 15, 16] that focus on the motion information of an object perform well in unsupervised VOS tasks. For example, Fragkiadaki *et al.* [13] ranked segment proposals by fusing optical flow and static boundaries. Tokmakov *et al.* [43] used only optical flow to capture motion cues, but it is difficult to segment static objects with their method due to insufficient detail information. In addition, MATNet [63] uses motion information to enhance the spatiotemporal object representation. However, this static and motion information fusion method performs poorly with complex moving backgrounds, and there is a problem dependent on the accuracy of the optical flow map.

Prototype Learning. Prototype learning is a method of learning a metric space in which features can be distinguished by calculating the distance to the prototype representation of each feature. In particular, prototype learning-based deep-learning models [9, 46, 26, 53, 57, 20] perform well in few-shot segmentation tasks. For example, Wang *et al.* [46] proposed a model that creates query masks with prototypes generated from support features and then creates support masks with prototypes generated from query features. Yang *et al.* [53] proposed a prototype mixture model, effectively fusing foreground and background prototypes. Li *et al.* [20] extracted robust features through clustering-based adaptive prototype learning and improved the few-shot segmentation performance.

We propose a memory prototype sampling architecture inspired by prototype learning to extract features of primary objects effectively from video sequences. However, unlike previous few-shot segmentation methods, the proposed method generates various prototypes representing objects by using the SLIC [2] algorithm. It also improves the unsupervised VOS performance by sampling prototypes of common objects and storing them in a memory bank.

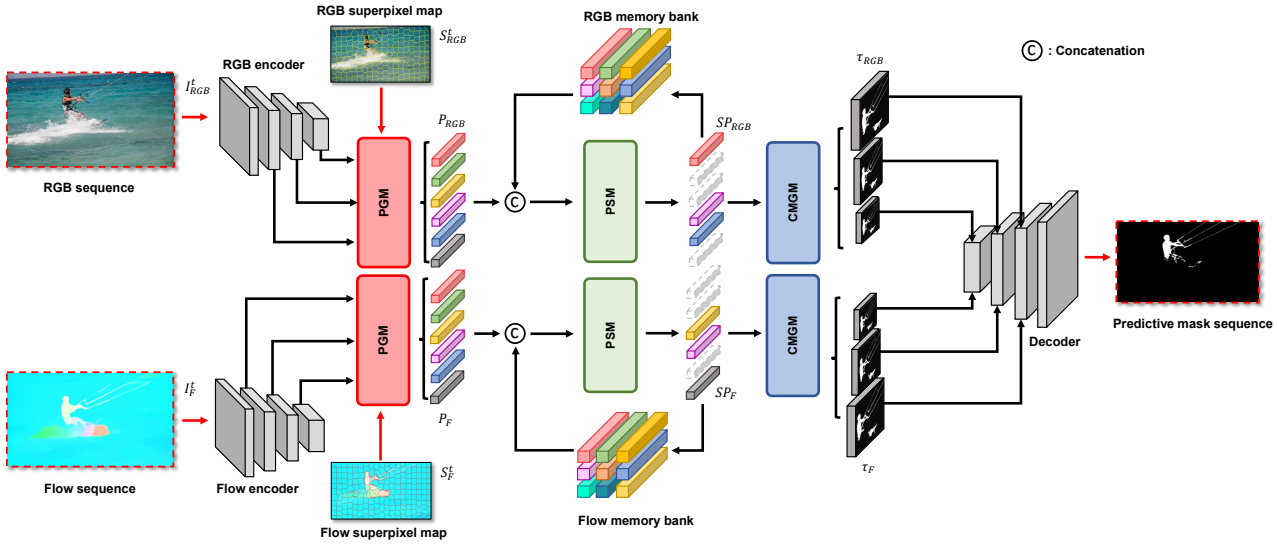

Figure 2. The overall architecture of the proposed prototype memory network (PMN). The prototype generating module (PGM) generates prototypes from images and flow maps. The prototype sampling module (PSM) samples the most useful prototypes by scoring the usefulness of the extracted prototypes. The memory bank stores the sampled prototypes to help predict the mask for the next frame. Finally, the correlation map generating module (CMGM) generates correlation maps from the sampled prototypes.

3. Proposed Approach

3.1. Overall Architecture

Figure. 2 shows the overall architecture of the proposed PMN. As inputs, the PMN uses an RGB image $\mathbf{I^t_{RGB}} \in \mathbb{R}^{3\times H\times W}$ at time t of the video sequence and an optical flow map $\mathbf{I^t_F} \in \mathbb{R}^{3\times H\times W}$ generated from $\mathbf{I^t_{RGB}}$ and $\mathbf{I^{t+1}_{RGB}}$ and their superpixel maps $\mathbf{S^t_{RGB}}$, $\mathbf{S^t_F}$. The proposed model consists of three primary parts: the PGM, PSM, and CMGM. We also create an RGB memory bank and a flow memory bank to store useful prototypes generated at time t and use them for mask prediction at time $t+1$. The PMN also has two encoders for RGB images and flow maps and one decoder.

3.2. Prototype Generating Module

The PGM generates prototypes from the multiscale encoder features $\mathbf{E_1}$, $\mathbf{E_2}$, and $\mathbf{E_3}$. As shown in Figure. 3, the PGM first integrates $\mathbf{E_1}$, $\mathbf{E_2}$, and $\mathbf{E_3}$ using 1×1 convolution and upsampling to generate $\mathbf{E} \in \mathbb{R}^{C\times \frac{H}{8}\times \frac{W}{8}}$. This architecture, based on a feature pyramid network (FPN) [23], effectively integrates multiscale features from the encoder. Furthermore, the PGM generates superpixel masks $\mathbf{SM^t_1}$, $\mathbf{SM^t_2}$, ... $\mathbf{SM^t_N} \in \{0,1\}^{1\times \frac{H}{8}\times \frac{W}{8}}$ from the superpixel map $\mathbf{S^t}$ at time t, where each channel is a binary mask for each superpixel and N is the number of superpixels. To generate prototypes from $\mathbf{E}$, we perform masked average pooling (MAP) using $\mathbf{SM^t}$ as masks, where $\mathbf{SM^t}$ is resized to the same size as $\mathbf{E}$. Consequently, the PGM generates N prototypes $\mathbf{P^E_1}$, $\mathbf{P^E_2}$, ..., $\mathbf{P^E_N} \in \mathbb{R}^{1\times C}$ from $\mathbf{E}$. The PGM process can be summarized as follows:

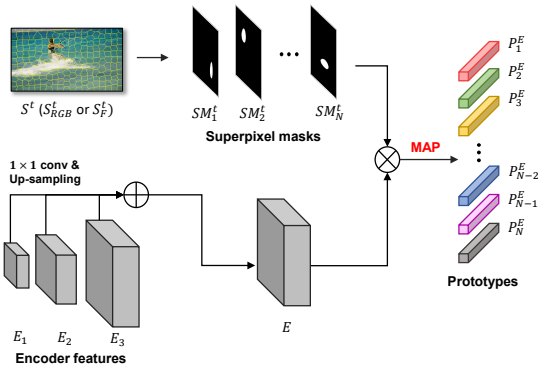

Figure 3. Architecture visualization of the PGM, which creates prototypes that represent the overall features of the subregions that make up the superpixel.

$$\mathbf{P^E_x} = \frac{\frac{H}{8}\times\frac{W}{8}}{\sum \mathbf{SM^t_x}} \times GAP\left(\mathbf{SM^t_x} \circ \mathbf{E}\right), \tag{1}$$

where $GAP\,(.)$ is global average pooling, $\sum(.)$ is the sum of all pixel values, and $\circ$ is element-wise multiplication. Furthermore, $x = 1, 2, ..., N$.

Prototype learning in a typical few-shot segmentation task [9, 46, 26, 53, 57, 20] extracts one representative prototype for each object. However, unlike the previous method,

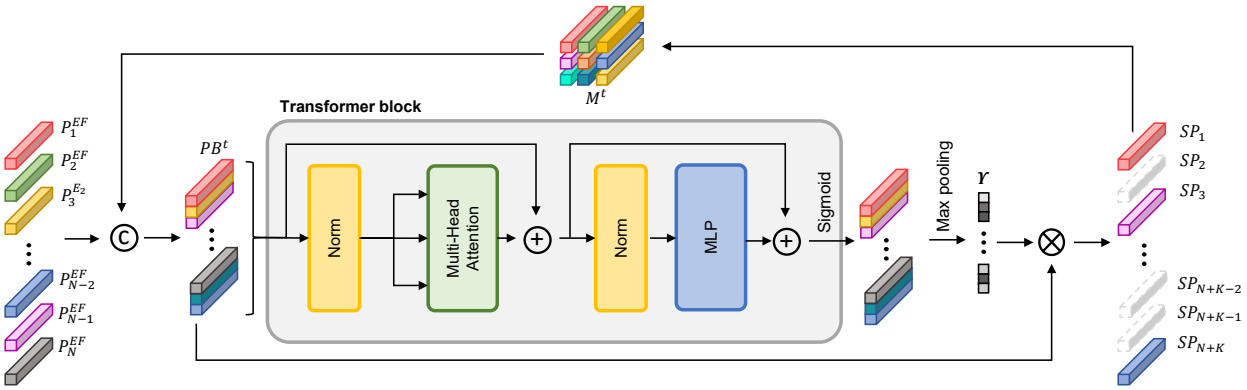

Figure 4. Structure of the proposed PSM and memory bank. The PSM scores the usefulness of the newly extracted prototypes and stores the highest-scoring prototypes in the memory bank.

the proposed method extracts various component prototypes based on superpixels so that it can retain various features for RGB images and optical flows.

3.3. Prototype Scoring Module and the Memory Bank

Prototypes extracted from the PGM have features that are useful for creating object masks, but they also have features that are not. The PSM samples the most useful prototypes and stores them in a memory bank. In other words, the memory bank stores useful prototypes from the previous 1 to t time frames, and when a more useful prototype occurs at frame $t+1$, the memory bank is updated. However, because we cannot define the usefulness of the prototypes using the ground truth, the proposed PSM is trained using a self-supervised mechanism. Therefore, the PSM focuses on correlations between prototypes that contain consistent characteristics for salient objects. Figure 4 shows the structure of the proposed PSM and memory bank.

The first step of the PSM generates a prototype block $\mathbf{PB^t}$ by concatenating the prototypes $\mathbf{P_1^E}$, $\mathbf{P_2^E}$, ..., $\mathbf{P_N^E}$ extracted from the encoder and the prototypes $\mathbf{P_1^{M^t}}$, $\mathbf{P_2^{M^t}}$, ..., $\mathbf{P_K^{M^t}}$ in memory bank M^t at time t, where K is the number of prototypes in M^t. Thus, the size of $\mathbf{PB^t}$ is $(N+K)\times C$. The next step is a transformer block, inspired by various vision transformer studies [10, 61, 51], to enhance the correlation between prototypes. Unlike general vision transformers, the patch-embedding process is omitted because the input is a prototype block rather than a 2D image. In addition, because the proposed PSM samples useful prototypes from pre-extracted object feature vectors, it is composed of a single transformer layer rather than multiple layers. The transformer block, shown in Figure 4, consists of a series connection of two layer normalization steps, a multi-head attention layer, a multi-layer perceptron layer, and a sigmoid layer. Therefore, the size of the prototype block with enhanced correlation between prototypes passed through the transformer block is $(N+K)\times C$, which is equal to the input prototype block size. Finally, the sampling vector $\boldsymbol{\Upsilon} \in [0,1]^{(N+K)}$ to select a useful prototype is calculated by max pooling. Finally, the input prototype block PB is multiplied by this sampling vector to produce a sampled prototype block $\mathbf{SPB^t} \in \mathbb{R}^{(N+K)\times C}$. We also update M^t to M^{t+1} by replacing the prototypes of M^t, selecting the top K samples with large $\boldsymbol{\Upsilon}$ values from among the sampled prototypes $\mathbf{SP_1}$, $\mathbf{SP_2}$, ..., $\mathbf{SP_{N+K}}$. Therefore, the memory bank update process is $\left\{\mathbf{P_n^{M^{t+1}}}\right\} \leftarrow \{\mathbf{SP_n}\}$, where $n = 1, 2, ..., K$.

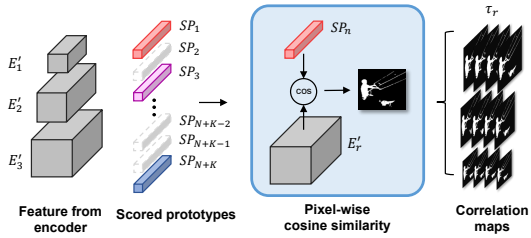

Figure 5. The overall flow of the proposed model. Our model generates and samples component prototypes from the superpixel maps. It also compares the reliability of correlation maps created from component prototypes to generate the predicted mask.

3.4. Correlation Map Generating Module

The CMGM generates correlation maps from the sampled prototypes $\mathbf{SP_1}$, $\mathbf{SP_2}$, ..., $\mathbf{SP_K}$ and encoder features passed through a 1x1 convolutional layer, $\mathbf{E_1^{'}}$, $\mathbf{E_2^{'}}$, and $\mathbf{E_3^{'}}$. Therefore, we compute the pixel-wise cosine similarity between the encoder feature and the sampled prototype, as shown in Figure 5. The correlation map $\tau_\mathbf{r}$ generated from $\mathbf{SP_n}$ and $\mathbf{E_r^{'}}$ is expressed as follows:

$$\tau_{\mathbf{r}}(x,y) = \text{concat}_n\left(\frac{\mathbf{E}'_{\mathbf{r}}(x,y)\cdot\mathbf{SP_n}}{\|\mathbf{E}'_{\mathbf{r}}(x,y)\|\,\|\mathbf{SP_n}\|}\right), \quad (2)$$

where (x, y) are the pixel coordinates and $r = 1, 2, 3$ and $\text{concat}_n(.)$ is the channel concatenation operator, where $n = 1, 2, ..., K$. This process allows the model to generate adaptive correlation maps from critical prototypes.

3.5. Loss Function

We optimize the model with object function L, where L is the intersection over union (IOU) loss [22] between the predicted saliency map I_{pred} and the ground truth mask I_{gt}, expressed as:

$$L = 1 - \frac{\sum \min\left(I_{pred}(x,y), I_{gt}(x,y)\right)}{\sum \max\left(I_{pred}(x,y), I_{gt}(x,y)\right)}, \quad (3)$$

where $\min(.,.)$ and $\max(.,.)$ represent the functions that take two maps as inputs and output the element-wise minimum and maximum, respectively. (x, y) are the pixel coordinates.

4. Experiments

4.1. Datasets and Evaluation Metrics

We perform experiments on the following three popular unsupervised VOS benchmarks to validate the effectiveness of our proposed method: DAVIS-16 [34], FBMS [32], and Youtube-Objects [35]. Note that most of the existing unsupervised VOS works use these datasets as the test sets [61, 68, 32]. Thus, we follow the same practice to ensure a fair comparison.

DAVIS-16. DAVIS-16 [34] is the most popular unsupervised VOS dataset, consisting of 30 training and 20 validation annotated video sequences. We leverage three evaluation metrics: region similarity $\mathcal{J}$, boundary accuracy $\mathcal{F}$, and overall $\mathcal{J}\&\mathcal{F}$ score, which is the average of the $\mathcal{J}$ and $\mathcal{F}$ scores. $\mathcal{J}$ and $\mathcal{F}$ are defined as follows:

$$\mathcal{J} = \frac{\sum \mathbf{S_{pred}} \cap \mathbf{S_{gt}}}{\sum \mathbf{S_{pred}} \cup \mathbf{S_{gt}}}, \quad (4)$$

$$\mathcal{F} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (5)$$

where $\text{Precision} = \sum \mathbf{S_{pred}} \cap \mathbf{S_{gt}} / \sum \mathbf{S_{pred}}$ and $\text{Recall} = \sum \mathbf{S_{pred}} \cap \mathbf{S_{gt}} / \sum \mathbf{S_{gt}}$.

FBMS. FBMS [32] includes 59 video sequences, with 29 are used as the training set and 30 for testing. Following [56, 28, 63], we used the region similarity $\mathcal{J}$ to evaluate our method on the test set without training.

YouTube-Objects. YouTube-Objects [35] contains 126 video sequences of 10 object categories. The ground-truth in YouTube-Objects is sparsely labeled in one of every 10 frames. Following [56, 28, 63], we use the region similarity $\mathcal{J}$ to evaluate our method on test set without training.

4.2. Model Training

We train the model in three steps following the training method of the previous works [15, 36, 25, 28]. First, we use a well-known saliency dataset DUTS [47] to pretrain the model to avoid over-fitting. The DUTS [47] dataset consists of a single RGB image and mask. Therefore, only the RGB encoders, PGM, PSM, CMGM, and decoders of the RGB stream in Figure 2 are pretrained. Second, because the proposed model has perfect symmetry between the RGB stream and the optical flow stream, we apply the pretrained parameters of the RGB stream equally to the optical flow stream. Finally, the entire model is fine-tuned with the DAVIS-16 [34] training set (30 sequences). The optical flow map used for pre-training and fine-tuning is generated using RAFT [42], a pretrained optical flow estimation model. In addition, 30 memory banks are created, the same number as the sequence of DAVIS-16 [34], and the prototypes created in each sequence are stored. To prevent the PSM from overfitting by certain prototypes, all memory blocks are reset every epoch.

4.3. Model Testing

We follow the standard benchmarks [12, 34] to test our model on the validation set of DAVIS-16 [34], the test set of FBMS [32], and the test set of Youtube-Objects [35]. Similar to the training phase, we generate an optical flow map using RAFT [42], a pretrained optical flow prediction model on three test sets. Furthermore, we initialize the memory banks in each test phase and create empty memory banks for each number of sequences in the test dataset.

4.4. Implementation Details

We set the number of superpixels N to 100 and the number of prototypes K in the memory bank to 50. The backbone encoder network is VGG16 [38], with ImageNet [8] pre-trained. All images are uniformly resized to 352×352 pixels for training and inferring. For network training and fine-tuning, we used the Adam optimizer [17] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The learning rate decayed from 10^{-4} to 10^{-5} with the cosine annealing scheduler [27]. The total number of epochs was set to 200 with batch size 12. The experiments were conducted on a single NVIDIA RTX 3090 GPU. We implement the proposed method using the open deep-learning framework PyTorch.

4.5. Results

In Tables 1, 2, and 3 and Figure 6, we compare the performance of the proposed model with previous state-of-the-art methods. Some studies generate a prediction mask and

Table 1. Performance comparison with other state-of-the-art methods on the DAVIS-16 [34] dataset. Higher scores are better. The best and second best are highlighted in red and blue, respectively.

Method	Year	Backbone	CRF	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}$-Mean $\uparrow$	$\mathcal{F}$-Mean $\uparrow$
AGS [50]	CVPR 2019	ResNet101 [14]	✓	78.6	79.7	77.4
COSNet [28]	CVPR 2019	DeepLabv3 [4]	✓	80.0	80.5	79.4
AD-Net [56]	ICCV 2019	ResNet101 [14]		81.1	81.7	80.5
AGNN [48]	ICCV 2019	DeepLabv3 [4]	✓	79.9	80.7	79.1
MATNet [63]	AAAI 2020	ResNet101 [14]		81.6	82.4	80.7
WCS-Net [59]	ECCV 2020	EfficientNetv2 [41]	✓	81.5	82.2	80.7
DFNet [60]	ECCV 2020	DeepLabv3 [4]	✓	82.6	83.4	81.8
3DC-Seg [30]	BMVC 2020	ResNet152 [14]		84.5	84.3	84.7
F2Net [25]	AAAI 2021	DeepLabv3 [4]		83.8	83.1	84.4
RTNet [36]	CVPR 2021	ResNet101 [14]	✓	85.2	85.6	84.7
FSNet [15]	ICCV 2021	ResNet50 [14]	✓	83.3	83.4	83.1
TransportNet [58]	ICCV 2021	ResNet101 [14]		84.8	84.5	85.0
AMC-Net [55]	ICCV 2021	ResNet101 [14]	✓	84.6	84.5	84.6
CFAM [5]	WACV 2022	ResNet101 [14]		82.8	83.5	82.0
IMP [19]	AAAI 2022	ResNet50 [14]		85.6	84.5	86.7
Ours		VGG16 [38]		85.9	85.4	86.4
Ours		VGG16 [38]	✓	85.9	85.6	86.2

Table 2. Performance comparison with other state-of-the-art methods on the FBMS [32] dataset. Higher scores are better. The best and second best are highlighted in red and blue, respectively.

Method	Year	Backbone	CRF	$\mathcal{J}$-Mean $\uparrow$
SFL [6]	CVPR 2017	ResNet101 [14]	✓	56.0
IET [21]	CVPR 2018	DeepLabv2 [4]	✓	71.9
PDB [40]	ECCV 2018	ResNet50 [14]	✓	74.0
COSNet [28]	CVPR 2019	DeepLabv3 [4]	✓	75.6
F2Net [25]	AAAI 2021	DeepLabv3 [4]		77.5
AMC-Net [55]	ICCV 2021	ResNet101 [14]	✓	76.5
IMP [19]	AAAI 2022	ResNet50 [14]		77.5
Ours		VGG16 [38]		77.7
Ours		VGG16 [38]	✓	77.8

Table 3. Performance comparison with other state-of-the-art methods on the Youtube-Objects [35] dataset. Higher scores are better. The best and second best are highlighted in red and blue, respectively.

Method	Year	Backbone	CRF	J -Mean
AGS [50]	CVPR 2019	ResNet101 [14]	✓	69.7
COSNet [28]	CVPR 2019	DeepLabv3 [4]	✓	70.5
AGNN [48]	ICCV 2019	DeepLabv3 [4]	✓	70.8
MATNet [63]	AAAI 2020	ResNet101 [14]		69.0
WCS-Net [59]	ECCV 2020	EfficientNetv2 [41]	✓	70.9
RTNet [36]	CVPR 2021	ResNet101 [14]	✓	71.0
AMC-Net [55]	ICCV 2021	ResNet101 [14]	✓	71.1
Ours		VGG16 [38]		71.8
Ours		VGG16 [38]	✓	71.8

then apply conditional random fields (CRF) [18] to it for post-processing. Therefore, Tables 1, 2, and 3 also present the results of our model applying CRF.

Quantitative Results. Tables 1, 2, and 3 show the quantitative results of the proposed SPSN. The proposed model achieves state-of-the-art performance on all three challenging datasets, even without postprocessing using CRF. We demonstrate the effectiveness of the proposed modules through various ablation studies in the next section.

Qualitative Results. Figure 6 shows visualized results of our model for various challenging video sequences. First, in the breakdance sequence, the proposed model shows robustness in complex background situations with many objects similar in appearance to the target object. It can also be seen in the BMX-Trees sequence that accurate mask generation is possible even when the target object is occluded. Finally, the Motocross-Jump sequence shows that the proposed model is capable of consistent feature extraction, even with extreme scale changes of the objects. These results show that the proposed SPSN extracts the common features of the target object from previous frames and stores them in a memory bank, thereby excluding the influence on non-common objects.

4.6. Ablation Analysis

We verified the performance of our model through various ablation studies. Table 4 shows the effects of the proposed modules in various combinations.

PSMN and the Memory Bank. Indices (c), (d), (e), and (f) in Table 4 show the effects of the proposed PSM and memory banks. In addition, we compared the performance of the proposed transformer block with the simple multi layer perceptron (MLP). Table 4 shows that using the transformer block proposed by PSMN is more effective than using a simple MLP. These results show that the self-attention

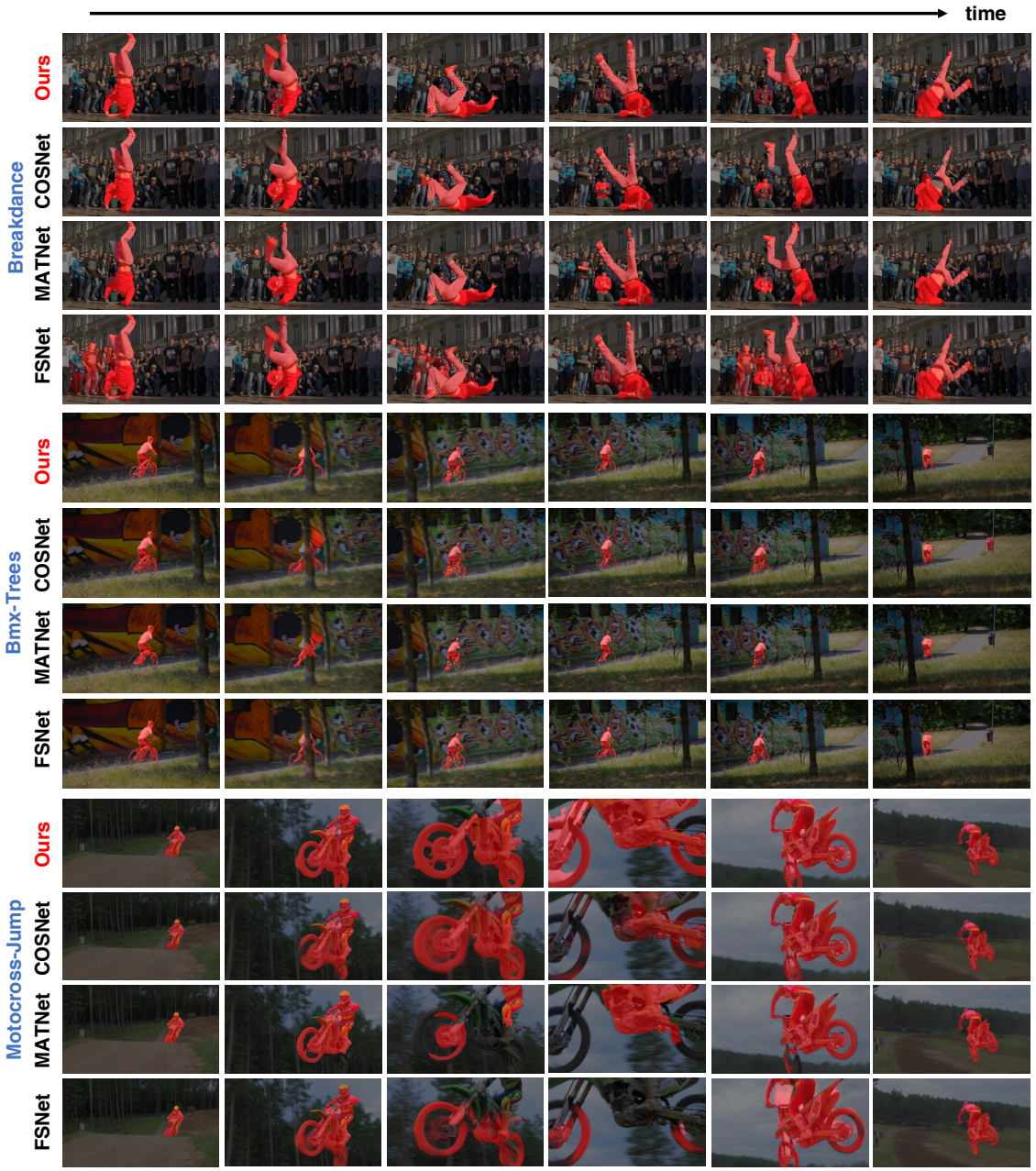

Figure 6. Qualitative comparison of the proposed method with previous state-of-the-art methods, FSNet [15], MATNet [63], and COSNet [28]. Our method demonstrates robust mask prediction in various challenging videos.

mechanism of the transformer block in PSMN can enhance the correlation between prototypes and extract useful features effectively. Furthermore, it shows a significant performance improvement when using the proposed memory bank. This is because the PSM and memory banks can recursively sample and store useful prototypes for accurate mask prediction.

Effects of the Superpixel Algorithm. Table 5 shows the performance of different prototype extraction methods on the DAVIS-16 [34] dataset. The random sampling method (a) considers random pixels in an image as superpoints and generates a prototype from that coordinate. The grid method (b) generates prototypes using evenly divided square masks from an image. We set the number of pro-

Table 4. Performance with different combinations of our contributions on the DAVIS-16 [34] dataset. Higher scores are better. RE and FE represent the encoders for the RGB images and flow maps.

Index	Method						$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}$-Mean $\uparrow$	$\mathcal{F}$-Mean $\uparrow$
	Encoder		PSMN		Memory Bank	CMGM			
	RE&PGM	FE&PGM	Transformer	MLP					
(a)	✓						82.5	82.3	83.7
(b)	✓	✓					83.3	83.0	83.6
(c)	✓	✓		✓		✓	84.1	84.0	83.9
(d)	✓	✓		✓	✓	✓	85.0	84.9	85.1
(e)	✓	✓	✓			✓	84.5	84.4	84.6
(f)	✓	✓	✓		✓	✓	**85.9**	**85.4**	**86.4**

Table 5. Statistical comparisons of the prototype sampling methods on the DAVIS-16 [34] dataset. Higher scores are better.

Index	Method	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}$-Mean $\uparrow$	$\mathcal{F}$-Mean $\uparrow$
(a)	Random	85.0	85.0	85.1
(b)	Grid	85.2	85.1	85.3
(c)	Superpixel	**85.9**	**85.4**	**86.4**

totypes to the same value, $N = 100$. As shown in Table 5, our proposed superpixel-based component sampling method outperformed the other methods, demonstrating its strong ability to capture the common features of the video sequence.

Size of the memory bank Figure 7 compares the $\mathcal{J}\&\mathcal{F}\uparrow$ scores on DAVIS-16 [34] according to the maximum number of prototypes K stored in the memory bank. The results show the model performed best at $K = 50$, with little change in performance when K was greater than 50. This shows that if the capacity of the memory bank is above a certain level, it is possible to hold a sufficient number of prototypes for target object extraction. Furthermore, we visualize the location of newly added prototypes in Figure 8. As shown in Figure 8, newly added prototypes are generally located on the salient object, showing that the proposed model can sample useful prototypes.

5. Conclusion

We propose a novel PMN architecture for unsupervised VOS tasks. The PMN extracts appearance features from RGB images and motion features from a flow map by generating component prototypes from the inputs. Furthermore, the model scores the usefulness of the prototypes in each frame and adaptively stores the most conducive prototypes in memory and removes obsolete prototypes. Prototypes stored in the memory bank enhance the mask prediction performance by emphasizing the connectivity between distant frames. Our model was evaluated on three popular datasets, achieving state-of-the-art performance.

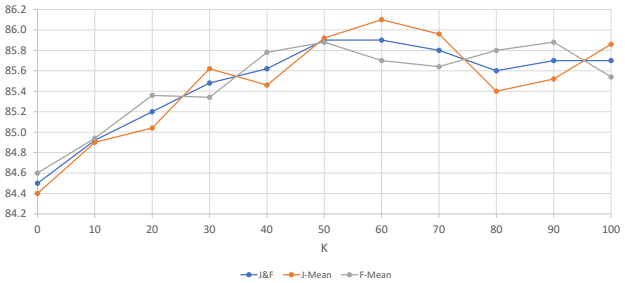

Figure 7. Comparison of performance characteristics with respect to K on the DAVIS-16 [34] dataset. Setting $K = 0$ is the same as not using the memory bank.

Figure 8. Visualization of two prototypes with the highest score added to the memory bank during testing in Car-Roundabout class. Superpixel areas marked in red indicate new additions to the memory bank in each sequence.

Acknowledgement. This research was supported by R&D program for Advanced Integrated-intelligence for Identification (AIID) through the National Research Foundation of KOREA(NRF) funded by Ministry of Science and ICT (NRF-2018M3E3A1057289), the KIST Institutional Program(Project No.2E31051-21-203), and the Yonsei University Research Fund of 2021 (2021-22-0001).

References

[1] Alexey Abramov, Karl Pauwels, Jeremie Papon, Florentin Wörgötter, and Babette Dellen. Depth-supported real-time video segmentation with the kinect. In *2012 IEEE workshop on the applications of computer vision (WACV)*, pages 457–464. IEEE, 2012.

[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien

Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[3] Alberto Bailoni, Constantin Pape, Nathan Hütsch, Steffen Wolf, Thorsten Beier, Anna Kreshuk, and Fred A Hamprecht. Gasp, a generalized framework for agglomerative clustering of signed graphs and its application to instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11645–11655, 2022.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[5] Yi-Wen Chen, Xiaojie Jin, Xiaohui Shen, and Ming-Hsuan Yang. Video salient object detection via contrastive features and attention modules. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1320–1329, 2022.

[6] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Proceedings of the IEEE international conference on computer vision*, pages 686–695, 2017.

[7] Suhwan Cho, Heansung Lee, Minjung Kim, Sungjun Jang, and Sangyoun Lee. Pixel-level bijective matching for video object segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 129–138, 2022.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[9] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, 2018.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[11] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *BMVC*, volume 2, page 8, 2014.

[12] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8554–8564, 2019.

[13] Katerina Fragkiadaki, Pablo Arbelaez, Panna Felsen, and Jitendra Malik. Learning to segment moving objects in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4083–4090, 2015.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Ge-Peng Ji, Keren Fu, Zhe Wu, Deng-Ping Fan, Jianbing Shen, and Ling Shao. Full-duplex strategy for video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4922–4933, 2021.

[16] Woo Jin Kim, Sangwon Hwang, Junhyeop Lee, Sungmin Woo, and Sangyoun Lee. Aibm: Accurate and instant background modeling for moving object detection. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[19] Youngjo Lee, Hongje Seong, and Euntai Kim. Iteratively selecting an easy reference frame makes unsupervised video object segmentation easier. *arXiv preprint arXiv:2112.12402*, 2021.

[20] Gen Li, Varun Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim. Adaptive prototype learning and allocation for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8334–8343, 2021.

[21] Siyang Li, Bryan Seybold, Alexey Vorobyov, Alireza Fathi, Qin Huang, and C-C Jay Kuo. Instance embedding transfer to unsupervised video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6526–6535, 2018.

[22] Huaijia Lin, Xiaojuan Qi, and Jiaya Jia. Agss-vos: Attention guided single-shot video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3949–3957, 2019.

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[24] Dongfang Liu, Yiming Cui, Yingjie Chen, Jiyong Zhang, and Bin Fan. Video object detection for autonomous driving: Motion-aid feature calibration. *Neurocomputing*, 409:1–11, 2020.

[25] Daizong Liu, Dongdong Yu, Changhu Wang, and Pan Zhou. F2net: Learning to focus on the foreground for unsupervised video object segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2109–2117, 2021.

[26] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *European Conference on Computer Vision*, pages 142–158. Springer, 2020.

[27] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[28] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese

networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3623–3632, 2019.

[29] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[30] Sabarinath Mahadevan, Ali Athar, Aljoša Ošep, Sebastian Hennen, Laura Leal-Taixé, and Bastian Leibe. Making a case for 3d convolutions for object segmentation in videos. *arXiv preprint arXiv:2008.11516*, 2020.

[31] Yunyao Mao, Ning Wang, Wengang Zhou, and Houqiang Li. Joint inductive and transductive learning for video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9670–9679, 2021.

[32] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1187–1200, 2013.

[33] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE international conference on computer vision*, pages 1777–1784, 2013.

[34] 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*, pages 724–732, 2016.

[35] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *2012 IEEE Conference on computer vision and pattern recognition*, pages 3282–3289. IEEE, 2012.

[36] Sucheng Ren, Wenxi Liu, Yongtuo Liu, Haoxin Chen, Guoqiang Han, and Shengfeng He. Reciprocal transformations for unsupervised video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15455–15464, 2021.

[37] Christian Schmidt, Ali Athar, Sabarinath Mahadevan, and Bastian Leibe. D2conv3d: Dynamic dilated convolutions for object segmentation in videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1200–1209, 2022.

[38] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[39] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. f-brs: Rethinking backpropagating refinement for interactive segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8623–8632, 2020.

[40] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 715–731, 2018.

[41] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, pages 10096–10106. PMLR, 2021.

[42] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020.

[43] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning motion patterns in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3386–3394, 2017.

[44] Wei-Chih Tu, Ming-Yu Liu, Varun Jampani, Deqing Sun, Shao-Yi Chien, Ming-Hsuan Yang, and Jan Kautz. Learning superpixels with segmentation-aware affinity loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 568–576, 2018.

[45] Bairui Wang, Lin Ma, Wei Zhang, and Wei Liu. Reconstruction network for video captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7622–7631, 2018.

[46] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9197–9206, 2019.

[47] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 136–145, 2017.

[48] Wenguan Wang, Xiankai Lu, Jianbing Shen, David J Crandall, and Ling Shao. Zero-shot video object segmentation via attentive graph neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9236–9245, 2019.

[49] Wenguan Wang, Jianbing Shen, and Fatih Porikli. Saliency-aware geodesic video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3395–3402, 2015.

[50] Wenguan Wang, Hongmei Song, Shuyang Zhao, Jianbing Shen, Sanyuan Zhao, Steven CH Hoi, and Haibin Ling. Learning unsupervised video object segmentation through visual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3064–3074, 2019.

[51] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8741–8750, 2021.

[52] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Shengping Zhang, and Wenxiu Sun. Efficient regional memory network for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1286–1295, 2021.

[53] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. In *European Conference on Computer Vision*, pages 763–778. Springer, 2020.

[54] Fengting Yang, Qian Sun, Hailin Jin, and Zihan Zhou. Superpixel segmentation with fully convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13964–13973, 2020.

[55] Shu Yang, Lu Zhang, Jinqing Qi, Huchuan Lu, Shuo Wang, and Xiaoxing Zhang. Learning motion-appearance co-attention for zero-shot video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1564–1573, 2021.

[56] Zhao Yang, Qiang Wang, Luca Bertinetto, Weiming Hu, Song Bai, and Philip HS Torr. Anchor diffusion for unsupervised video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 931–940, 2019.

[57] Qinji Yu, Kang Dang, Nima Tajbakhsh, Demetri Terzopoulos, and Xiaowei Ding. A location-sensitive local prototype network for few-shot medical image segmentation. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 262–266. IEEE, 2021.

[58] Kaihua Zhang, Zicheng Zhao, Dong Liu, Qingshan Liu, and Bo Liu. Deep transport network for unsupervised video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8781–8790, 2021.

[59] Lu Zhang, Jianming Zhang, Zhe Lin, Radomír Měch, Huchuan Lu, and You He. Unsupervised video object segmentation with joint hotspot tracking. In *European Conference on Computer Vision*, pages 490–506. Springer, 2020.

[60] Mingmin Zhen, Shiwei Li, Lei Zhou, Jiaxiang Shang, Haoan Feng, Tian Fang, and Long Quan. Learning discriminative feature with crf for unsupervised video object segmentation. In *European Conference on Computer Vision*, pages 445–462. Springer, 2020.

[61] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021.

[62] Tianfei Zhou, Yao Lu, Huijun Di, and Jian Zhang. Video object segmentation aggregation. In *2016 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2016.

[63] Tianfei Zhou, Shunzhou Wang, Yi Zhou, Yazhou Yao, Jianwu Li, and Ling Shao. Motion-attentive transition for zero-shot video object segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13066–13073, 2020.