RANet: Ranking Attention Network for Fast Video Object Segmentation

Ziqin Wang[1,3], Jun Xu[2,4]*, Li Liu[2], Fan Zhu[2], Ling Shao[2]
[1]The University of Sydney, Sydney, Australia
[2]Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, UAE
[3]Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China
[4]Media Computing Lab, College of Computer Science, Nankai University, Tianjin, China

Project page: https://github.com/Storife/RANet

Abstract

Despite online learning (OL) techniques have boosted the performance of semi-supervised video object segmentation (VOS) methods, the huge time costs of OL greatly restricts their practicality. Matching based and propagation based methods run at a faster speed by avoiding OL techniques. However, they are limited by sub-optimal accuracy, due to mismatching and drifting problems. In this paper, we develop a real-time yet very accurate Ranking Attention Network (RANet) for VOS. Specifically, to integrate the insights of matching based and propagation based methods, we employ an encoder-decoder framework to learn pixel-level similarity and segmentation in an end-to-end manner. To better utilize the similarity maps, we propose a novel ranking attention module, which automatically ranks and selects these maps for fine-grained VOS performance. Experiments on DAVIS$_{16}$ and DAVIS$_{17}$ datasets show that our RANet achieves the best speed-accuracy trade-off, e.g., with 33 milliseconds per frame and $\mathcal{J}\&\mathcal{F}$=85.5% on DAVIS$_{16}$. With OL, our RANet reaches $\mathcal{J}\&\mathcal{F}$=87.1% on DAVIS$_{16}$, exceeding state-of-the-art VOS methods. The code can be found at https://github.com/Storife/RANet.

1. Introduction

Semi-supervised Video Object Segmentation (VOS) [4, 41, 42] aims to segment the object(s) of interests from the background throughout a video, in which only the annotated segmentation mask of the first frame is provided as the template frame at test phase. This challenging task is of great importance for large scale video processing and editing [52–54], and many video analysis applications such as video understanding [15, 46] and object tracking [51].

Early VOS methods [3, 37, 40, 50] mainly resort to online learning (OL) techniques which fine-tune a pre-trained classifier on its first frame. Matching or propagation based methods have also been proposed for VOS. Matching based methods [8, 19] segment pixels according to the pixel-level matching scores between the features of the first frame and of each subsequent frame (Fig. 1 (**a**)), while propagation based methods [9, 10, 38, 40, 54, 59] mainly rely on temporally deforming the annotated mask of the first frame via predictions of the previous frame [40] (Fig. 1 (**b**)).

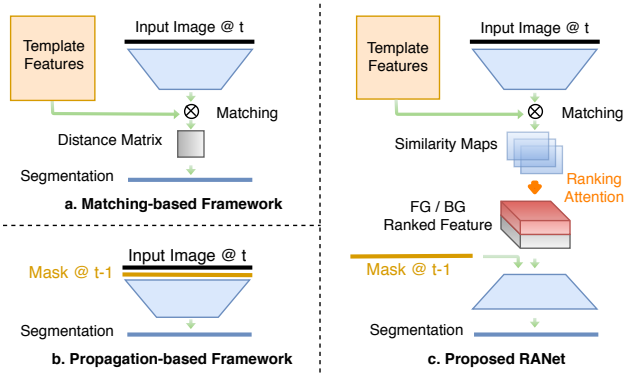


Figure 1: **Comparison of different VOS frameworks**. (**a**) Matching based framework; (**b**) Propagation based framework; and (**c**) Proposed RANet. We propose a novel *Ranking Attention* module to rank and select important features.

The respective benefits and drawbacks of these methods are clear. Specifically, OL based methods [3, 37, 40, 50] achieve accurate VOS at the expense of speed, requiring several seconds to segment each frame [3]. On the contrary, simple matching or propagation based methods [8, 40, 45] are faster, but with sub-optimal VOS accuracy. Matching based methods [8, 19, 38] bear up the mismatching problem, *i.e.*, violating the temporal consistency of the primary object with constantly changing appearance in the video. On the other hand, propagation based methods [9, 10, 38, 40, 47, 59] suffer from the drifting problem due to occlusions or fast motions between two sequential frames. In summary, most existing methods cannot tackle the VOS task with both satisfactory accuracy and speed, which are essential for prac-

*Corresponding author: Jun Xu (nankaimathxujun@gmail.com). This work is done when Ziqin Wang was an intern in IIAI.

tical applications. More efficient methods are still required to reach a better speed-accuracy trade-off for the VOS task.

With the above considerations, in this work, we develop a real-time network for fine-grained VOS performance. The developed network benefits from an encoder-decoder structure, and learns pixel-level matching, mask propagation, and segmentation in an end-to-end manner. Fig. 1 (**c**) shows a glimpse of the proposed network. A Siamese network [2] is employed as the encoder to extract pixel-level matching features, and a pyramid-like decoder is used for simultaneous mask propagation and high-resolution segmentation.

A key problem in our framework is how to connect the pixel-level matching encoder and propagation based decoder in a meaningful manner. The encoder produces dynamic foreground and background similarity maps, which cannot be directly fed into the decoder. To this end, we propose a *Ranking Attention Module* (RAM, see Fig. 1 (**c**)) to reorganize (*i.e.*, rank and select) the similarity maps according to their importance for fine-grained VOS performance. The proposed *Ranking Attention Network* (RANet) can better utilize the pixel-level similarity maps for fine-grained VOS, greatly alleviating the drawbacks of previous matching or propagation based methods. Experiments on DAVIS_{16} and DAVIS_{17} datasets [41, 42] demonstrate that the proposed RANet outperforms previous VOS methods in terms of speed and accuracy, e.g., achieving $\mathcal{J}\&\mathcal{F} = 85.5\%$ at a speed of 30 FPS on DAVIS_{16}.

The contributions of this work are three-fold:

- We integrate the benefits of matching and propagation frameworks in an end-to-end manner and develop a real-time network for the semi-supervised VOS task.
- We propose a novel *Ranking Attention Module* to rank and select conformable feature maps according to their importance for fine-grained VOS performance.
- Experiments on $\text{DAVIS}_{16/17}$ datasets show that the proposed RANet achieves competitive or even better performance than previous VOS methods, at real-time speed. The proposed RANet achieves accurate VOS results even been trained only with static images.

2. Related Works

Online learning based methods. OL based methods [3, 25, 30, 33–35, 37, 40, 50] fine-tune on the first frame of a video to extract the primary object(s), and then segment the video frame-by-frame. OSVOS [3] uses a pre-trained object segmentation network, and fine-tunes it on the first frame of the test video. OnAVOS [50] extends OSVOS with an online adaptation mechanism, and OSVOS-S [37] utilizes semantic information from an instance segmentation network. LucidTracker [25] introduces a data augmentation mechanism for online fine-tuning. DyeNet [30] integrates instance re-identification and temporal propagation, and uses OL to boost the performance. PReMVOS [33–35] integrates techniques from instance segmentation [16], optical flow [12, 20], refinement, and re-identification [57] together with extensive fine-tuning, and achieves satisfactory performance. In summary, OL is very effective for the VOS task. Therefore, subsequent methods [1,30,40] regard OL as a conventional technique to boost VOS performance. However, OL based methods are computationally expensive for practical applications. In this work, we solve the VOS problem with a very fast network that obtains a competitive accuracy at a speed of 30 FPS on DAVIS_{16}, $130 \sim 400$ times faster than previous OL based methods [3, 37, 40, 50].

Propagation or matching based methods. Propagation based methods additionally resort to the previous frame(s) for better VOS performance. Masktrack [40] tackles VOS by combining the image and segmentation mask of the previous frame as the input. This strategy is also used in CINM [1], OSMN [59] and RGMP [38]. RGMP [38] stacks the first, previous and current frames' features during propagation through a Siamese architecture network. In this work, we also utilize the Siamese network, but use a pixel-level matching technique instead of simply stacking, and feed the previous frame's mask into the decoder, instead of the encoder as in RGMP [38]. OSMN [59] introduces a modulator to manipulate the intermediate layers of the segmentation network, by using visual and spatial guidance. Optical flow [12, 20] is also used to guide the propagation process in many methods [10, 23, 40, 47]. However, it fails to distinguish non-rigid objects from motionless sections of the background. All these strategies are effective, but still, suffer from the drifting problem. MaskTrack [40] embraces OL to remember the target object, which eliminates this problem and improves VOS performance. However, since OL is time-consuming, we employ more efficient matching techniques to handle this drifting problem.

Matching based methods [8, 19, 45, 49] are very efficient. They first calculate pixel-level matching between the features of the template frame and the current frame in videos, and then segment each pixel of the current frame directly from the matching results. Pixel-Wise Metric Learning [8] predicts each pixel by nearest neighbor matching in pixel space to the template frame. However, the point-to-point correspondence strategy [43, 45] often results in noisy predictions. To ease this problem, we apply a decoder to utilize the matching results as guidance. Hu *et al*. proposed a soft matching mechanism in VideoMatch [19], which performs soft segmentation upon the averaged similarity score maps of matching features to generate smooth predictions. However, due to the lack of temporal information, they still suffer from the mismatching problem. In this work, we employ both the strategies of point-to-point correspondence matching for pixel-level object location and temporal propagation, to handle the mismatching and drifting problem. FEELVOS [49] employs global and local match-

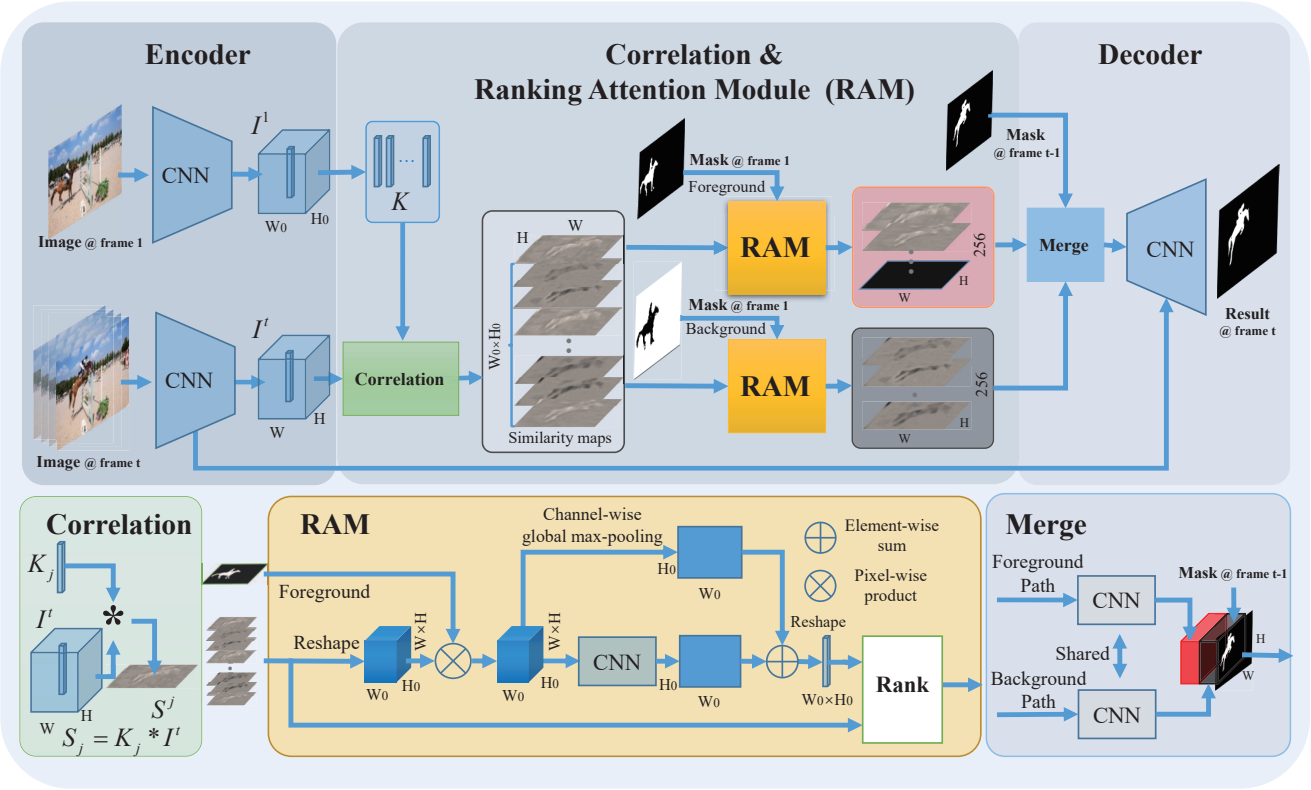


Figure 2: **Illustration of the proposed RANet**. We compute correlation of the features extracted by Siamese networks. The output similarity maps and template mask are fed into the RAM module to rank and select the foreground/background similarity maps. Then these maps and the previous frame's mask and fed into the decoder for final segmentation.

ing for more stable pixel-level matching, but only calculates extreme value maps for final segmentation, losing major information of the similarity maps. Our RAM can better utilize the similarity information. Moreover, for faster speed, we use a light-weight decoder and employ a standard ResNet [17] pre-trained on ImageNet [27] as the backbone, instead of the time-consuming semantic segmentation networks [5–7, 39] used in previous methods [19, 40].

3. Proposed Method

In this section, we first provide an overview of the developed Ranking Attention Network (RANet) in §3.1. In §3.2, we describe the proposed Ranking Attention Module (RAM), and extend it for multi-object VOS in §3.3. Finally, we present the implementation details and training strategies for RANet in §3.4 and §3.5, respectively.

3.1. Network Overview

Our RANet consists of three seamless parts: an encoder for feature extraction, an integration of correlation and RAM, and a decoder for feature merging and final segmentation. An illustration of our RANet is shown in Fig. 2.

Siamese Encoder. To obtain correlation information for accurate VOS, we employ Siamese networks [2] (with shared weights) as the encoder to extract features of the first frame and the current frame. Then we extract pixel-level features from the first frame, reshape it into a conformable shape, as the template features for correlation calculation.

Correlation and RAM for Matching. Correlation is widely used in object tracking. In SiamFC [2], correlation is used to locate the position of the object using similarity maps. In our RANet, to locate each pixel of the object(s) for segmentation, we need pixel-level similarity maps by calculating the correlation between each pixel-level feature of the template and current frames. Note that there is one similarity map for each pixel-level template feature. The detailed formulation of correlation will be described in §3.2. We then utilize the mask of the first frame to select foreground (FG) or background (BG) similarity maps as FG or BG features for segmentation. Since the number of FG or BG pixels varies in different videos, the number of FG or BG similarity maps is dynamic, and hence the decoder has to deal with FG or BG similarity features with dynamic channel sizes. To handle this dynamic channel-size problem, we propose a RAM module to rank and select the most important similarity maps and organize them in conformable shape. This part will also be exhaustively explained in §3.2. The RAM module provides abundant and ordered features for segmentation, and leads to better performance, as will be shown in the ablation study in §4.3. For simplicity, here we only consider the single-object VOS in §3.2. Extension of our RANet for multi-object VOS will be described in §3.3.

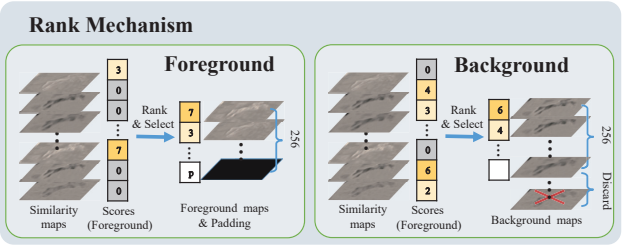


Figure 3: **Mechanism of the proposed Ranking Attention Module**. In FG (or BG) path, only the FG (or BG) similarity maps are selected. The maps are ranked from top to bottom according to ranking scores learned from attention network, and padding or discarding is operated to make the 256 FG (or BG) maps. Finally, these maps are concatenated across the channel as features with the size of $256 \times H \times W$.

Propagation. Here we utilize the simple mask propagation method [40], while other propagation [20, 30] or local-matching [49] methods would potentially improve our RANet. We feed the predicted mask of the previous frame, together with the selected features of FG (or BG) by the proposed RAM, into the subsequent decoder. In this way, our RANet utilizes both matching and propagation techniques.

Light-weight Decoder. This part contains a merge module and a pyramid-like network, which are described in the *Supplementary File*. The merge module refines the two streams of ranked similarity maps, and then concatenates these maps with previous frame's mask. In the merge module, the two streams of the network share the same parameters. A pyramid-like network [31, 44, 56] is employed to obtain the final segmentation, with skip-connections to utilize multi-scale features of different layers.

3.2. Correlation and Ranking Attention Module

Correlation. We utilize correlation to find matching between pixels in the template and current frames. Denote $\boldsymbol{I}^1 \in \mathbb{R}^{C\times H_0\times W_0}$ and $\boldsymbol{I}^t \in \mathbb{R}^{C\times H\times W}$ as the feature of template and current frames, extracted by the Siamese encoder, where C is the number of feature channels, Denote H_0 (W_0) and H (W) as the height (width) of template and current frame feature maps, respectively. We reshape the template features $\boldsymbol{I}^1 \in \mathbb{R}^{C\times H_0\times W_0}$ to the size of $H_0W_0 \times (C \times 1 \times 1)$. Denote the reshaped template feature set as $\mathcal{K} = \{K_j | j = 1, .., H_0 \times W_0\}$, which consist of $H_0 \times W_0$ features with the size of $C \times 1 \times 1$. In our RANet, the correlation is computed between the ℓ_2-normalized features K_j in template frame $\mathcal{K}$ and the current frame $\boldsymbol{I}^t$. After correlation, we have the similarity maps $\boldsymbol{S}_j = \boldsymbol{K}_j * \boldsymbol{I}^t$ whose size is $W \times H$. Denote the tensor $\mathcal{S} \in \mathbb{R}^{H_0W_0\times H\times W}$ as the set of correlation maps. Then we have

$$\mathcal{S} = \{\boldsymbol{S}_j \mid \boldsymbol{S}_j = \boldsymbol{K}_j * \boldsymbol{I}^t\}_{j\in\{1,..,H_0\times W_0\}} \quad (1)$$

In Fig. 4, we present some examples of the similarity maps. Each similarity map is associated with a certain pixel in template frame, whose new position in the current frame is at the maximum (*i.e.*, brightest point) of the similarity map. Additionally, in contrast with SiamFC [2], since we obtain these maps in a weakly-supervised manner, the contours of the bear, which are essentially preserved for segmentation, are maintained. On the right side of Fig. 4, we show some output features of the merging module. The object can be distinguished after the merging networks.

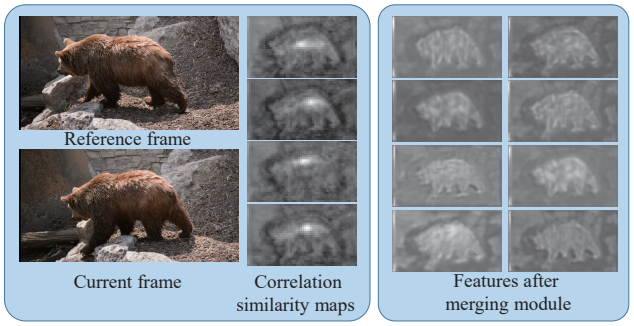
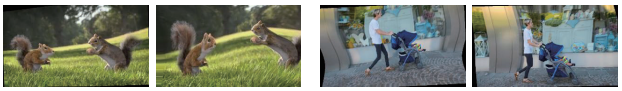


Figure 4: **Visualization of the similarity maps**. *Left*: the template and current frames, and 4 foreground correlation similarity maps. *Right*: the similarity maps after merging.

Ranking Attention Module (RAM). We first utilize the mask of the first frame to filter FG and BG similarity maps. Then we design a FG path and a BG path network to process the similarity features. Since the number of the FG or BG pixels varies in different videos, the number of FG or BG similarity maps changes dynamically. However, regular CNNs require input features with a fixed number of channels. To tackle this issue, we propose a Ranking Attention Module (RAM) to rank and select important features. That is, we learn a scoring scheme for the similarity maps, and then rank and select these maps according to their scores.

As shown in Fig. 2, there are three steps in our RAM. In the first step, we filter FG (or BG) similarity maps using the mask of the first frame. Specifically, we swap the spatial and channel dimensions of similarity maps (reshape $\mathbf{S} \in \mathbb{R}^{H_0W_0\times H\times W}$ into $\hat{\mathbf{S}} \in \mathbb{R}^{HW\times H_0\times W_0}$) and then multiply them with the FG or BG mask (resized to $W_0 \times H_0$), respectively. Thus, we obtain the FG (or BG) features $\hat{\mathbf{S}}^1$ (or $\hat{\mathbf{S}}^0$). In FG component, the features of BG pixels are set as zero, and vice versa. In the second step, for each similarity map S_j, we learn a ranking score r_j which show the importance of each map. Taking the FG tensor $\hat{\mathbf{S}}^1$ for instance, to calculate the ranking scores of similarity maps in $\hat{\mathbf{S}}^1$, we use a two-layer network f_n strengthened by summing with the channel-wise global max-pooling f_{max} of the tensor $\hat{\mathbf{S}}^1$ in an element-wise manner. Larger score indicates greater importance of the corresponding similarity map in $\hat{\mathbf{S}}^1$. The channel-wise maximum of each similarity map represents the possibility of corresponding pixel in template frame to find a matching pixel in current frame. We define the final FG ranking score metric $\mathbf{R}^1 \in \mathbb{R}^{W_0\times H_0}$ as

$$\mathbf{R}^1 = f_n(\hat{\mathbf{S}}^1) + f_{max}(\hat{\mathbf{S}}^1). \quad (2)$$



a. Static image training samples b. Video training samples

Figure 5: **Illustrations of the training samples**.

Then we reshape $\mathbf{R}^1$ into a vector $\mathbf{r}^1 \in \mathbb{R}^{H_0 W_0}$. Similarly, we can obtain the BG ranking score vector $\mathbf{r}^0$.

Finally, we rank the similarity maps in $\mathbf{S}^1$ according to the corresponding scores in $\mathbf{r}^1$ from largest to smallest:

$$\overline{\mathbf{S}}^1 = \text{Rank}(\mathbf{S}^1|\mathbf{r}^1). \tag{3}$$

If the number of the FG similarity maps $\overline{\mathbf{S}}^1$ is less than the target channel size (set as 256), we pad the ranked feature with zero maps; and if the number is larger than the target channel size, the redundant features are discarded, such that the channel size can be fixed. The BG tensor $\hat{\mathbf{S}}^0$ are similarly processed. An illustration of the proposed ranking mechanism is shown in Fig. 3.

3.3. Extension for Multi-object VOS

A trivial extension of single-object VOS methods to perform multi-object VOS is to deal with the multiple objects in videos one-by-one. But this strategy would be inefficient when there are many objects. To make the proposed RANet efficient for multi-object VOS, we share the features extracted by the encoder and also the similarity maps $\mathbf{S}$ computed by correlation for all the N objects. Then, for each object i ($i = 1, ..., N$), we generate its FG and the corresponding BG masks, and segment the FG (or BG) independently using the light-weight decoder. Finally, we use a softmax function to compute the final results on VOS.

3.4. Implementation Details

Here, we briefly describe the encoder and decoder, and present the detailed network structure in *Supplemental File*.

Encoder. The backbone of the two-stream Siamese encoder [2] is the ResNet-101 network [17], pre-trained on ImageNet [27]. We replace the batch normalization [21] with instance normalization [48]. The features from the last three blocks are extracted as multi-scale features. We reduce the channel sizes of these multi-scale features by four-fold via convolutional layers. The features are also resized into the conformable size. The ℓ_2 channel-wise normalization [18] is added after each convolutional layer for feature pruning and multi-scale merging.

Decoder. The decoder is a three-level pyramid-like network with skip connection. The multi-scale features of current frame extracted by encoder are fed into the decoder. However, using all the features in the decoder would bring huge computational costs. To speed up our RANet, we first reduce the channel sizes of the multi-scale features using convolutional layers, and then feed them into the decoder.

3.5. Network Training

We train our network using the Adam [26] with a initial learning rate of 10^{-5}, to optimize an binary cross-entropy loss. During training and test, the input image is resized into 480×864. We use random Thin Plate Splines (TPS) transformations, rotations ($-30°$∼$30°$), scaling (0.75∼1.25), and random cropping for data augmentation, just as [40]. The random TPS transformations are performed by setting 16 control points and randomly shifting the points within a 15% margin of the image size.

Pre-train on static images. Following [40], we pre-train the proposed RANet using static images. To train our RANet for single-object VOS, we use the images from the MSRA10K [11], ECSSD [58], and HKU-IS [29] datasets in the saliency community [14, 15, 32, 55, 60, 61]. To train RANet for multi-object VOS, we add the SOC [13] and ILSO [28] datasets containing multi-object images. Fig. 5 (a) shows a pair of generated static images. As will be shown in §4.2 and §4.3, the proposed RANet achieves competitive results when been trained only with static images.

Video fine-tuning. Though our RANet can achieve satisfactory results when been trained only with static images, we further exploit its performance by performing video fine-tuning on benchmark datasets. To fine-tune our RANet for specific single-object VOS task, we then fine-tune the network on the training set of the DAVIS_{16} dataset [41]. During training, we randomly select two frames with data transformations from one video as the template and current frames, and randomly select the mask of a frame near the current frame (we set the maximum interval as 5). We fine-tune our RANet for specific multi-object VOS task on the training set of the DAVIS_{17} dataset [42]. Fig. 5 (b) shows an example of paired video training images.

4. Experiments

In this section, we first describe our experimental protocol (§4.1), and then compare the proposed ranking attention network (RANet) with the state-of-the-art VOS methods (§4.2). We next perform a comprehensive ablation study to gain deeper insights into the proposed RANet, especially on the effectiveness of the ranking attention module (§4.3). Finally, we present the visual results to show the robustness of RANet against challenging scenarios (§4.4). More results are provided in the *Supplementary File*.

4.1. Experimental Protocol

Training datasets. We evaluate the proposed RANet on the DAVIS_{16} [41] and DAVIS_{17} [42] datasets. The DAVIS_{16} dataset [41] contains 50 videos (480p), annotated with pixel-level object masks (one per sequence) densely on the total 3455 frames, and it is divided into a training set (30 videos), and a validation set (20 videos). The DAVIS_{17}

Method	OL	Time	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}$ Mean↑	$\mathcal{J}$ Recall↑	$\mathcal{J}$ Decay↓	$\mathcal{F}$ Mean↑	$\mathcal{F}$ Recall↑	$\mathcal{F}$ Decay↓
OSVOS [3]	✓	5000	80.2	79.8	93.6	14.9	80.6	92.6	15.0
MaskTrack [40]	✓	12000	77.6	79.7	93.1	8.9	75.4	87.1	9.0
CINM [1]	✓	70000	84.2	83.4	94.9	12.3	85.0	92.1	14.7
OSVOS-S [37]	✓	4500	86.6	85.6	96.8	5.5	87.5	95.9	8.2
OnAVOS [50]	✓	13000	85.5	86.1	96.1	**5.2**	84.9	89.7	**5.8**
PReMVOS [35]	✓	38000	86.8	84.9	96.1	8.8	**88.6**	94.7	9.8
RANet+	✓	4000	**87.1**	**86.6**	**97.0**	7.4	87.6	**96.1**	8.2
PLM [45]	✗	500	66.4	70.2	86.3	11.2	62.5	73.2	14.7
VPN [22]	✗	630	67.9	70.2	82.3	12.4	65.5	69.0	14.4
SiamMask [51]	✗	**28**	70.0	71.7	86.8	**3.0**	67.8	79.8	**2.1**
CTN [23]	✗	30000	71.4	73.5	87.4	15.6	69.3	79.6	12.9
OSMN [59]	✗	130	73.5	74.0	87.6	9.0	72.9	84.0	10.6
SFL [10]	✗	7900	76.1	76.1	90.6	12.1	76.0	85.5	10.4
PML [8]	✗	280	77.4	75.5	89.6	8.5	79.3	93.4	7.8
VideoMatch [19]	✗	320	-	81.0	-	-	-	-	-
FAVOS [9]	✗	1800	81.0	82.4	96.5	4.5	79.5	89.4	5.5
FEELVOS [49]	✗	510	81.7	81.1	90.5	13.7	82.2	86.6	14.1
RGMP [38]	✗	130	81.8	81.5	91.7	10.9	82.0	90.8	10.1
RANet	✗	**33**	**85.5**	**85.5**	**97.2**	6.2	**85.4**	**94.9**	5.1

Table 1: **Comparison on objective metrics and running time (in milliseconds) by different methods on the DAVIS$_{16}$-val dataset**. The best results of online learning (OL) based methods and offline methods are both highlighted in bold.

Method	$\mathcal{J}$ Mean↑	$\mathcal{J}$ Recall↑	$\mathcal{J}$ Decay↓
BVS [36]	66.5	76.4	26.0
OFL [47]	71.1	80.0	22.7
VPN [22]	75.0	90.1	9.3
CTN [23]	75.5	89.0	14.4
MaskTrack [40]	80.3	93.5	8.9
RANet	83.2	94.2	9.3
RANet+OL	**86.2**	**96.2**	**7.6**

Table 2: **Comparison of different methods without video fine-tuning on DAVIS$_{16}$-trainval dataset**. "RANet+OL" denotes the proposed RANet boosted by OL techniques.

dataset [42], that contains videos with multiple objects, is an extension of DAVIS$_{16}$, and it contains a training set with 60 videos, a validation set with 30 videos, and a test-dev set with 30 videos. In all datasets, there is no overlap among the training, validation, and test sets.

Testing phase. Similar to SiamFC [2], we crop the first frame and extract the features as the template features ($\mathcal{K}$ in §3.2), then compute the similarity maps between the features of template frame and of the test frames one-by-one, and finally segment the current test frame. The video data used are in different goals: 1) to evaluate our RANet for single-object VOS, we test it on the validation set (20 videos) of [41]; 2) to judge the effectiveness of our RANet trained only on static images, we evaluate it on the 50 videos of the whole DAVIS$_{16}$ dataset; 3) to assess our RANet for multi-object VOS, we evaluate it on the validation and test sets of [42], which respectively contain 30 videos. To compare with OL based methods, we follow [3, 40], fine-tuning on the first frame with data augmentation for each video. We use the same training strategy as pre-training on static images, but the learning rate is 10^{-6}.

Evaluation metrics. We use seven standard metrics suggested by [41]: three region similarity metrics $\mathcal{J}$ Mean, $\mathcal{J}$ Recall, $\mathcal{J}$ Decay; three boundary accuracy metrics $\mathcal{F}$ Mean, $\mathcal{F}$ Recall, $\mathcal{F}$ Decay; and $\mathcal{J}\&\mathcal{F}$ Mean, which is the average of $\mathcal{J}$ Mean and $\mathcal{F}$ Mean.

4.2. Comparison to the state of the art

Comparison Methods. For single object VOS, we compare our RANet with 6 state-of-the-art OL based and 11 offline methods [1, 3, 8–10, 19, 22, 23, 35, 37, 38, 40, 45, 49–51, 59] in Table 1, including OSVOS-S [37], PReMVOS [35], RGMP [38], FEELVOS [49], *etc*. To evaluate our RANet trained with static images, we compare it with some methods [22, 23, 36, 40, 47] without using DAVIS training set. For multi-object VOS, we compare with some state-of-the-art offline methods [3, 9, 19, 50, 59], and also list results of some OL based methods [1, 3, 19, 37, 50] for reference.

Results on DAVIS$_{16}$-val. As shown in Table 1, without online learning (OL) technique, our RANet still achieves a $\mathcal{J}\&\mathcal{F}$ Mean of 85.5% at a speed of 33 milliseconds (30FPS). For RANet, its metric results are higher than all the methods without OL techniques, while its speed is higher than all the compared methods, except SiamMask [51]. But please note that SiamMask performs badly on objective metrics, e.g., 70.0% at $\mathcal{J}\&\mathcal{F}$, 15.5 points lower than our RANet. Even when compared with the state-of-the-art OL based methods such as OSVOS-S [37]

Method	OL	DAVIS$_{17}$-val $\mathcal{J}\&\mathcal{F}\uparrow$	DAVIS$_{17}$-val $\mathcal{J}$ Mean$\uparrow$	DAVIS$_{17}$-testdev $\mathcal{J}\&\mathcal{F}\uparrow$	DAVIS$_{17}$-testdev $\mathcal{J}$ Mean$\uparrow$
CINM [1]	✓	**70.6**	**67.2**	**67.5**	**64.5**
OSVOS-S [37]	✓	68.0	64.7	57.5	52.9
OnAVOS [50]	✓	65.4	61.6	52.8	49.9
OSVOS [3]	✓	60.3	56.6	50.9	47.0
VideoMatch [19]	✓	61.4	-	-	-
OSVOS [3]	✗	36.6	-	-	-
OnAVOS [50]	✗	39.5	-	-	-
FAVOS [9]	✗	58.2	54.6	43.6	42.9
OSMN [37]	✗	54.8	52.5	41.3	37.7
VideoMatch [19]	✗	56.5	-	-	-
RANet	✗	**65.7**	**63.2**	**55.3**	**53.4**

Table 3: **Comparison of different methods on DAVIS$_{17}$-val and DAVIS$_{17}$-testdev datasets**. The methods are divided into two groups according to whether online learning (OL) technique is employed or not.

and OnAVOS [50], our offline RANet achieves comparative results. The RANet can be improved by OL techniques. The OL boosted RANet, denoted as RANet+, achieves a $\mathcal{J}\&\mathcal{F}$ Mean of 87.1%, outperforming all OL based VOS methods.

Results on DAVIS$_{16}$-trainval. We also evaluate the performance of our RANet trained only with static images (i.e., without video fine-tuning). MaskTrack [40] has the most similar setting as our RANet in this case, since it also uses only static images to train its networks. Contrast to MaskTrack, our RANet does not rely on OL techniques, speeding up for nearly a hundred times faster. In Table 2, we list the results of different methods that do not require fine-tuning/training on video data. Again, our RANet outperforms all the other methods by a clear margin.

DAVIS$_{17}$ dataset: The DAVIS$_{17}$ dataset is challenging due to multi-object scenarios. To evaluate our RANet on DAVIS$_{17}$-val and DAVIS$_{17}$-test sets, we use the RANet trained on multi-instance static images and the DAVIS$_{17}$-train dataset, as described in §3.5. In Table 3, we show the comparison of our RANet with state-of-the-art VOS methods. It can be seen that on the DAVIS$_{17}$-val dataset, our RANet achieves higher metric results than the *w/o* OL methods. Furthermore, on the more challenging DAVIS$_{17}$-testdev dataset, our RANet even outperforms the OL based method OnAVOS in terms of $\mathcal{J}$ Mean.

Speed. Here, we evaluate the speed-accuracy performance of different methods on DAVIS$_{16}$-val set. Our RANet runs on a TITAN Xp GPU. In Table 1, we list the average time of different methods processing a frame of 480p resolution. Note that the proposed RANet spends 33 milliseconds on each frame, much faster than most of the previous methods. As shown in Fig. 6. The recently proposed method SiamMask [51] is a little faster than our RANet but at expenses of much lower results on $\mathcal{J}\&\mathcal{F}$ Mean than ours.

4.3. Validation of the Proposed RANet

We now conduct a more detailed examination of our proposed RANet on the VOS task. We assess 1) the contribution of the proposed ranking attention module (RAM)

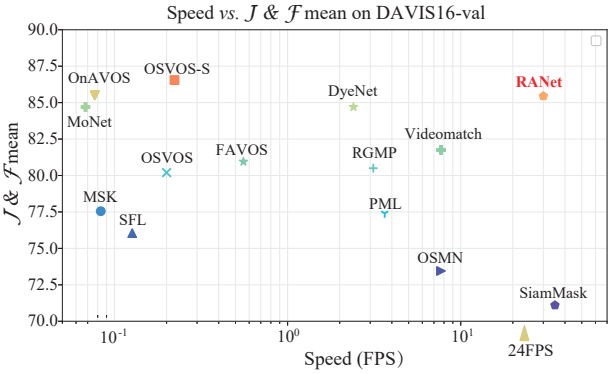


Figure 6: **Comparison of $\mathcal{J}\&\mathcal{F}$ Mean and Speed** (in FPS) by different methods on DAVIS$_{16}$-val dataset.

Variant	*w/ RAM*	*w/o Ranking*	*Maximun*
$\mathcal{J}$ Mean	85.5	81.9	81.1

Table 4: **Comparison of $\mathcal{J}$ Mean by different variants of RANet on DAVIS$_{16}$-val dataset**.

to RANet; 2) the importance of correlation layer (CL) to RANet; 3) the influence of propagating previous frame's mask (PM) on RANet; 4) the effect of static image pre-train (IP) and video fine-tuning (VF) on RANet; and 5) the impact of online learning (OL) technique to RANet.

1. Does the proposed ranking attention module contribute to RANet? To evaluate the contribution of the proposed RAM module to RANet on VOS task. We compare the original RANet, we call it *w/ RAM*, with two baselines. For the first one, *w/o Ranking*, we maintain all the similarity maps in $\mathcal{S}$, and obtain FG (or BG) similarity maps $\mathbf{S}^1$ (or $\mathbf{S}^0$)$\in\mathbb{R}^{H_0W_0\times H\times W}$ by setting corresponding BG (or FG) as zeros according to the template mask. For the second one, *Maximum*, instead of using RAM to obtain abundant embedding maps, we employ channel-wise maximum operation, which is also used in [49], on the similarity maps $\mathbf{S}^1$ and $\mathbf{S}^0$, respectively, to get one FG and one BG map $\mathbf{S}^1_M, \mathbf{S}^0_M \in \mathbb{R}^{H\times W}$. Then we feed them into the decoder.

The comparison of RANet *w/ RAM*, *w/o Ranking*, and *Maximum* is listed in Table 4. It can be seen that, the RANet *w/ RAM* achieves 3.6% and 4.4% higher than the baselines *w/o Ranking* and *Maximum*, respectively. The RANet *w/o Ranking* organizes the similarity maps based on the spacial information of the template frame, while the RANet with *Maximum* losses most useful information in similarity maps by only extracting the maximum values.

2. How important is the correlation and RAM to our RANet? To evaluate the importance of correlation layer in our RANet, we remove the correlation layer, and simply concatenate the features extracted by the encoder, as RGMP [38] does. The following RAM module is also meaningless and is removed. Thus we have a new variant of RANet: *-CL*. However, as shown in Table 5, the performance of this variant is very bad (67.5% on $\mathcal{J}$ Mean). Thus, the correlation layer is important to our RANet, and serves

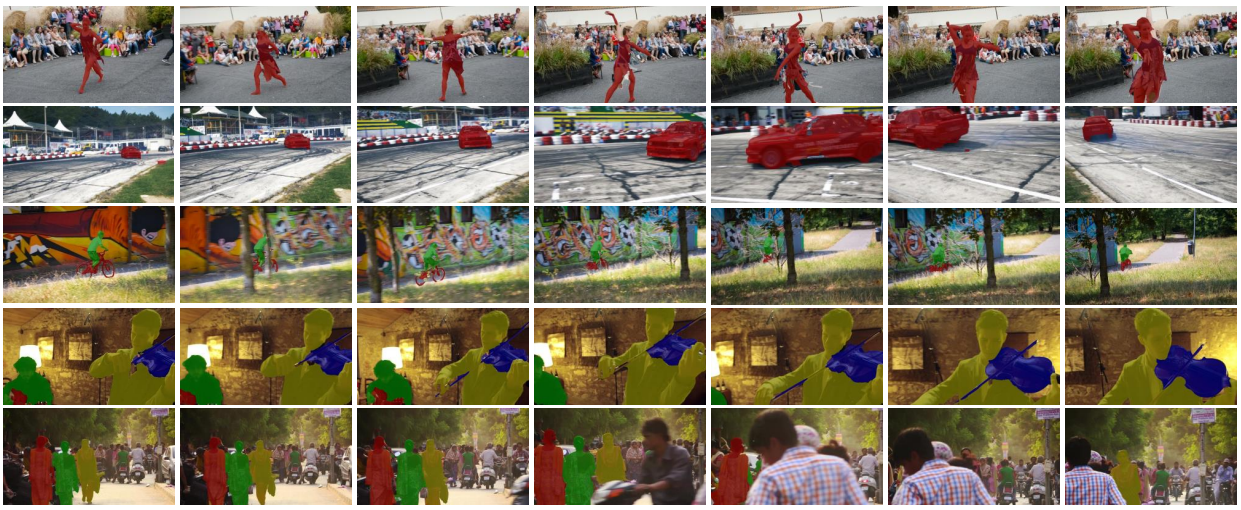


Figure 7: **Qualitative results of the proposed RANet on challenging VOS scenarios**. The test frames are from videos in the DAVIS_{16} set (1-st and 2-nd rows), the DAVIS_{17}-val set (3-rd row), and the DAVIS_{17}-testdev set (4-th and 5-th rows).

Method	origin	*-CL*	*-PM*	*-IP*	*-VF*
RGMP [38]	81.5	-	73.5	68.6	55.0
RANet	85.5	67.5	81.4	73.2	79.9

Table 5: **Ablation study of RANet on $\mathcal{J}$ Mean**. CL, PM, IP, and VF mean Correlation Layer, Previous frame's Mask, static Image Pre-train, and Video Fine-tuning, respectively.

Metric	offline	+online learning			
$\mathcal{J}\&\mathcal{F}$ Mean	85.5	86.2	86.8	86.9	87.1
Time	0.033	0.30	1.00	1.50	4.00

Table 6: **Influence of online learning to RANet** with different iterations on $\mathcal{J}\&\mathcal{F}$ Mean and runtime (in seconds).

as the base for the proposed RAM module.

3. How does the previous frame's mask (PM) influence our RANet? We study how PM influence our RANet. To this end, we set all the pixels of the PM as zero, and re-train our RANet. Thus we have a baseline of *-PM*. Results in Table 5 shows that, the variant *-PM* of RANet will drop $\mathcal{J}$ Mean by 4.1 points. This indicates that the temporal information propagated by PM is very useful for our RANet.

4. What are the effects of pre-training on static images and video fine-tuning in our RANet? To answer this question, we study how each training strategy affect the performance of RANet. We first train RANet only on video data and have a baseline: *-IP*. We then train RANet only on static images and have the second baseline: *-VF*. The results of $\mathcal{J}$ Mean by the variants *-IP* and *-VF* on DAVIS_{16}-val dataset are listed in Table 5. As can be seen, both baselines drop significantly on $\mathcal{J}$ Mean when compared to the original RANet. Specifically, static image pre-train (IP) improves the $\mathcal{J}$ Mean from 73.2% to 85.5%, while video fine-tuning (VF) improves the $\mathcal{J}$ Mean by 5.6 points. The performance drops (from 85.5% to 73.2%) of removing IP is mainly due to the over-fitting of RANet on the DAVIS_{16}-training set, which only contains 30 single-object videos.

5. The trade-off between performance and speed using online learning. In Table 6, we also show the performance and run-time of RANet with or without OL technique. One can see that, as the number of iterations increases in OL, the results of our RANet on $\mathcal{J}\&\mathcal{F}$ Mean are continuously improved with different extents, while at a cost of speed.

4.4. Qualitative Results

In Fig. 7, we show some qualitative visual results of the proposed RANet on the DAVIS_{16} and DAVIS_{17} datasets. It can be seen that, the RANet is very robust against many challenging scenarios, such as appearance changes (1-st row), fast motion (2-nd row), occlusions (3-th row), and multi-objects (4-rd and 5-th rows), *etc*.

5. Conclusion

In this work, we proposed a real-time and accurate VOS network, which runs at 30 FPS on a single Titan Xp GPU. The proposed ranking attention network (RANet) end-to-end learned the pixel-level feature matching and mask propagation for VOS. A ranking attention module was proposed to better utilize the similarity features for fine-grained VOS performance. The network treated the point-to-point matching feature as a guidance instead of the final results, to avoid noisy predictions. Experiments on $\text{DAVIS}_{16/17}$ datasets demonstrate that our RANet achieves state-of-the-art performance on both segmentation accuracy and speed.

This work can be further extended. First, the proposed ranking attention module can be applied to other applications such as object tracking [51] and stereo vision [24]. Second, better propagation [12, 20] or local matching [49] techniques can be employed for better VOS performance.

Acknowledgements. We thank Dr. Song Bai on the initial discussion of this project.

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, 6, 7

[2] Luca Bertinetto, Jack Valmadre, Joao Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV Workshops*, pages 850–865, 2016. 2, 3, 4, 5, 6

[3] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixe, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, July 2017. 1, 2, 6, 7

[4] Sergi Caelles, Alberto Montes, Kevis-Kokitsi Maninis, Yuhua Chen, Luc Van Gool, Federico Perazzi, and Jordi Pont-Tuset. The 2018 davis challenge on video object segmentation. *arXiv:1803.00557*, 2018. 1

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 3

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 3

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 3

[8] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, 2018. 1, 2, 6

[9] 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. 1, 6, 7

[10] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *ICCV*, Oct 2017. 1, 2, 6

[11] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE TPAMI*, 37(3):569–582, 2015. 5

[12] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, pages 2758–2766, 2015. 2, 8

[13] Deng-Ping Fan, Ming-Ming Cheng, Jiang-Jiang Liu, Shang-Hua Gao, Qibin Hou, and Ali Borji. Salient objects in clutter: Bringing salient object detection to the foreground. In *ECCV*. Springer, 2018. 5

[14] Deng-Ping Fan, Zheng Lin, Jia-Xing Zhao, Yun Liu, Zhao Zhang, Qibin Hou, Menglong Zhu, and Ming-Ming Cheng. Rethinking rgb-d salient object detection: Models, datasets, and large-scale benchmarks. *arXiv:1907.06781*, 2019. 5

[15] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *CVPR*, 2019. 1, 5

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3, 5

[18] Elad Hoffer, Ron Banner, Itay Golan, and Daniel Soudry. Norm matters: efficient and accurate normalization schemes in deep networks. In *NIPS*, pages 2164–2174, 2018. 5

[19] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G. Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, pages 56–73. Springer, 2018. 1, 2, 3, 6, 7

[20] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2462–2470, 2017. 2, 4, 8

[21] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 5

[22] Varun Jampani, Raghudeep Gadde, and Peter V. Gehler. Video propagation networks. In *CVPR*, July 2017. 6

[23] Won-Dong Jang and Chang-Su Kim. Online video object segmentation via convolutional trident network. In *CVPR*, July 2017. 2, 6

[24] Sameh Khamis, Sean Ryan Fanello, Christoph Rhemann, Julien Valentin, and Shahram Izadi. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *Europen Conference on Computer Vision (ECCV)*, 2018. 8

[25] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for object tracking. In *The DAVIS Challenge on Video Object Segmentation*, 2017. 2

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 5

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105. 2012. 3, 5

[28] Guanbin Li, Yuan Xie, Liang Lin, and Yizhou Yu. Instance-level salient object segmentation. In *CVPR*, pages 247–256. IEEE, 2017. 5

[29] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. *CVPR*, 2015. 5

[30] Xiaoxiao Li and Chen Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *ECCV*, pages 90–105, 2018. 2, 4

[31] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, pages 5168–5177. IEEE, 2017. 4

[32] 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 *CVPR*, June 2019. 5

[33] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for the davis challenge on video object segmentation 2018. *The 2018 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*, 2018. 2

[34] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for the youtube-vos challenge on video object segmentation 2018. *The 1st Large-scale Video Object Segmentation Challenge - ECCV 2018 Workshops*, 2018. 2

[35] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *ACCV*, 2018. 2, 6

[36] Nicolas Maerki, Federico Perazzi, Oliver Wang, and Alexander Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, June 2016. 6

[37] Kevis-Kokitsi Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taix, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *TPAMI*, 2018. 1, 2, 6, 7

[38] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, 2018. 1, 2, 6, 7, 8

[39] Yanwei Pang, Yazhao Li, Jianbing Shen, and Ling Shao. Towards bridging semantic gap to improve semantic segmentation. In *ICCV*, 2019. 3

[40] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, July 2017. 1, 2, 3, 4, 5, 6, 7

[41] 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 *CVPR*, pages 724–732, 2016. 1, 2, 5, 6

[42] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 1, 2, 5, 6

[43] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Deepmatching: Hierarchical deformable dense matching. *International Journal of Computer Vision*, 120(3):1–24, 2016. 2

[44] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 4

[45] Jae Shin Yoon, Francois Rameau, Junsik Kim, Seokju Lee, Seunghak Shin, and In So Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *ICCV*, Oct 2017. 1, 2, 6

[46] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *ECCV*, 2018. 1

[47] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J. Black. Video segmentation via object flow. In *CVPR*, June 2016. 1, 2, 6

[48] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 5

[49] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, and Liang Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 2019. 2, 4, 6, 7, 8

[50] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017. 1, 2, 6, 7

[51] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*. IEEE, 2019. 1, 6, 7, 8

[52] Wenguan Wang, Xiankai Lu, David Crandall, Jianbing Shen, and Ling Shao. Zero-shot video object segmentation via attentive graph neural networks. In *ICCV*, 2019. 1

[53] Wenguan Wang, Jianbing Shen, and Fatih Porikli. Selective video object cutout. *IEEE Transactions on Image Processing*, 26(12):5645–5655, 2017. 1

[54] Wenguan Wang, Jianbing Shen, Fatih Porikli, and Ruigang Yang. Semi-supervised video object segmentation with super-trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):985–998, 2019. 1

[55] Wenguan Wang, Jianbing Shen, Ruigang Yang, and Fatih Porikli. Saliency-aware video object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(1):20–33, 2018. 5

[56] Ziqin Wang, Peilin Jiang, and Fei Wang. Dense residual pyramid networks for salient object detection. In *ACCV Workshop*, pages 606–621, 2016. 4

[57] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *CVPR*, pages 3415–3424, 2017. 2

[58] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. *CVPR*, 2013. 5

[59] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K. Katsaggelos. Efficient video object segmentation via network modulation. *CVPR*, 2018. 1, 2, 6

[60] Jia-Xing Zhao, Yang Cao, Deng-Ping Fan, Xuan-Yi Li, Le Zhang, and Ming-Ming Cheng. Contrast prior and fluid pyramid integration for rgbd salient object detection. In *CVPR*, 2019. 5

[61] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Jufeng Yang, and Ming-Ming Cheng. Edge-based network for salient object detection. In *ICCV*, 2019. 5