Fast Video Object Segmentation by Reference-Guided Mask Propagation

Seoung Wug Oh[*] Yonsei University
Joon-Young Lee Adobe Research
Kalyan Sunkavalli Adobe Research
Seon Joo Kim Yonsei University

Abstract

We present an efficient method for the semi-supervised video object segmentation. Our method achieves accuracy competitive with state-of-the-art methods while running in a fraction of time compared to others. To this end, we propose a deep Siamese encoder-decoder network that is designed to take advantage of mask propagation and object detection while avoiding the weaknesses of both approaches. Our network, learned through a two-stage training process that exploits both synthetic and real data, works robustly without any online learning or post-processing. We validate our method on four benchmark sets that cover single and multiple object segmentation. On all the benchmark sets, our method shows comparable accuracy while having the order of magnitude faster runtime. We also provide extensive ablation and add-on studies to analyze and evaluate our framework.

1. Introduction

Video object segmentation – separating a foreground object from a video sequence – is one of most important tasks in video analysis and editing, and commercial applications such as Adobe After Effects have dedicated tools for it. However, automatic video object segmentation is far from a solved problem, and post-production video editing often requires significant manual interaction to achieve pleasing results. While recent work has addressed this problem, performance is still limited in terms of either the quality or the speed. In this paper, our goal is to develop an accurate video object segmentation algorithm that is also fast enough to be used in interactive settings.

Video object segmentation methods typically rely on two important cues. *Propagation-based* methods [13, 37, 28, 30] mainly leverage the temporal coherence of object motion and formulate this problem as object mask propagation (*i.e.* pixel-level tracking) starting from a given annotated frame. These methods rely on the spatiotemporal connections between pixels, and thus can adapt to complex deformation and movement of a target object as long as the changes in the appearance and the location are smooth. However, these methods are vulnerable to temporal discontinuities like occlusions and rapid motion, and can suffer from drifting once the propagation becomes unreliable.

Detection-based methods [5, 27, 45] learn the appearance of the target object from a given annotated frame, and perform a pixel-level detection of the target object at each frame. As they rarely depend on temporal consistency, they are robust to occlusion and drifting. However, as their estimation is mostly based on the object appearance in an annotated frame(s), they often fail to adapt to appearance changes and have difficulty separating object instances with similar appearances.

Recent approaches to this problem have utilized deep networks. Most of these approaches heavily rely on online learning, where a pre-trained deep network is fine-tuned on the test video [45, 30, 40, 5, 27, 18, 21]. While online training improves segmentation accuracy by letting the network adapt to the target object appearance, it is computationally expensive, thus limiting its practical use (*e.g.* it requires several minutes of GPU-powered training for each test video).

In this paper, we present a new *hybrid* method for semi-supervised video object segmentation. We construct a Siamese encoder-decoder network that simultaneously

[*]This work is done during an internship at Adobe Research

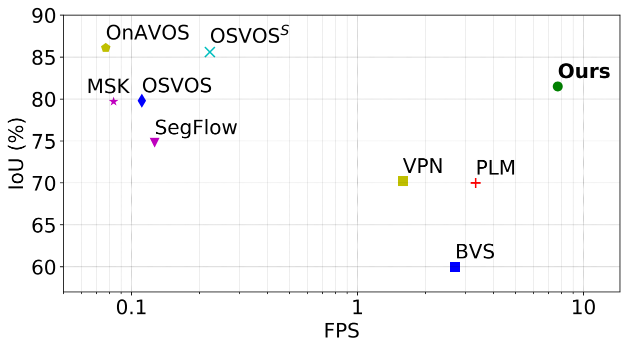


Figure 1: A comparison of the quality and the speed of previous video object segmentation methods (DAVIS-2016 benchmark). We visualize the intersection-over-union (IoU) with respect to the frames-per-second (FPS). Note that the FPS axis is in the log scale.

makes use of both the previous mask to be *propagated* to the current frame and the reference frame which specifies the target object to be *detected* in the current frame. Our network is designed to generate a sharp object mask without time-consuming post-processing. To address the lack of large segmented training video datasets, we use a two-stage scheme that pre-trains the network on synthetically generated image data and then fine-tunes it on video data.

Our network architecture and training scheme have been carefully designed to take advantage of both propagation and detection cues. Consequently, the network works robustly without any online learning or post-processing, leading to tremendous efficiency at test time. Our method not only achieves state-of-the-art performance on public benchmark datasets, but also runs orders of magnitude faster than previous methods that rely on online learning (as shown in Fig. 1). We also provide extensive experimental analysis and evaluation on the influence of each component through the ablation and the add-on studies.

2. Related Work

Unsupervised methods. Unsupervised methods aim to segment a foreground object in a fully automatic way without any user annotation. The main sources of information include visual saliency [42] and difference in motion (*e.g.* optical flow [35] and long-term trajectory [4]). However, the criteria for a foreground object are often ambiguous and the unsupervised segmentation does not fit well with the interactive video editing scenario. We focus on semi-supervised methods in this paper.

Propagation-based methods. Many video segmentation methods start from user annotations (*e.g.* segmentation masks or scribbles at key-frames) that roughly specify the object of interest. To propagate these sparse labels through the entire video sequence, graph representations are often used [13, 37, 28]. A spatiotemporal graph where pixels (or superpixels) are connected with space-time neighbors is built from a video. Energy-based optimization like graph-cut is performed to assign the optimal label for each node.

For professional video editing applications, interactive methods are often preferred over automatic methods [41, 11, 2, 25]. These methods focus on designing an efficient way for users to specify segmentation constraints and to quickly respond to these constraints.

Recent approaches have used deep learning for label propagation in videos. A temporal bilateral network was proposed for spatiotemporal dense filtering in [20]. In [30], a deep network was trained to refine the previous frame mask to create the current frame mask. They trained a network for this task using only static images. They use online fine-tuning using the first frame of the test video to memorize target object appearance, leading to a boost in the performance. Khoreva *et al*. [21] extended [30] by proposing a heavy data augmentation strategy for online learning, to achieve higher accuracy. In [18], Hu *et al*. developed a recurrent neural network framework for multi-instance segmentation. With a recurrent network, they capture temporal coherence effectively and take advantage of long-term temporal structure of a video.

Detection-based methods. Another approach in the semi-supervised setting is to exploit the appearance of the target object in a given reference frame. Methods in this category frame video object segmentation as pixel-level object detection in each frame, processing a video frame-by-frame without considering temporal consistency. In [5], Caelles *et al*. applied one-shot online learning that fine-tunes a deep network on a labeled frame using a pre-trained model and used that fined-tuned network as the detector. Maninis *et al*. [27] extended this idea by incorporating additional information from an auxiliary instance segmentation network [26]. Voigtlaender and Leibe [40] further developed the idea from [5] by employing an online adaptation mechanism originating from the box-level tracking. Yoon *et al*. [45] proposed a Siamese network for the pixel-level matching to detect a target object.

3. Method

Given a reference frame with an object mask, the goal of our method is to automatically segment the target object from the entire video sequence. The key idea of our method is exposing both the reference frame with annotation and the current frame with previous mask estimation to a deep network, so that the network *detects* the target object by matching the appearance at the reference frame and also *tracks* the previous mask by referencing the previous target mask in the current frame.

3.1. Network Structure

Fig. 2 depicts our network structure. We construct the model as a Siamese encoder-decoder structure that can efficiently handle four inputs and produce a sharp mask output. The network consists of two encoders with shared parameters, a global convolution block, and a decoder. The network is designed to be fully convolutional, which can handle arbitrary input size and generate sharp output masks.

Siamese encoder. The encoder takes a pair of RGB images, each with a mask map, as an input. The encoder includes a reference and a target stream, and the filter weights are shared between the streams. Inputs to the reference stream include a reference image which is usually the first frame of the video, and the groundtruth mask. For the target stream, a target (current) image and a guidance mask corresponding to the previous frame are provided. We concatenate the image frame and the mask along the channel axis, then feed it into the encoder. The parameter-shared encoders map the two stream data into the same feature space.

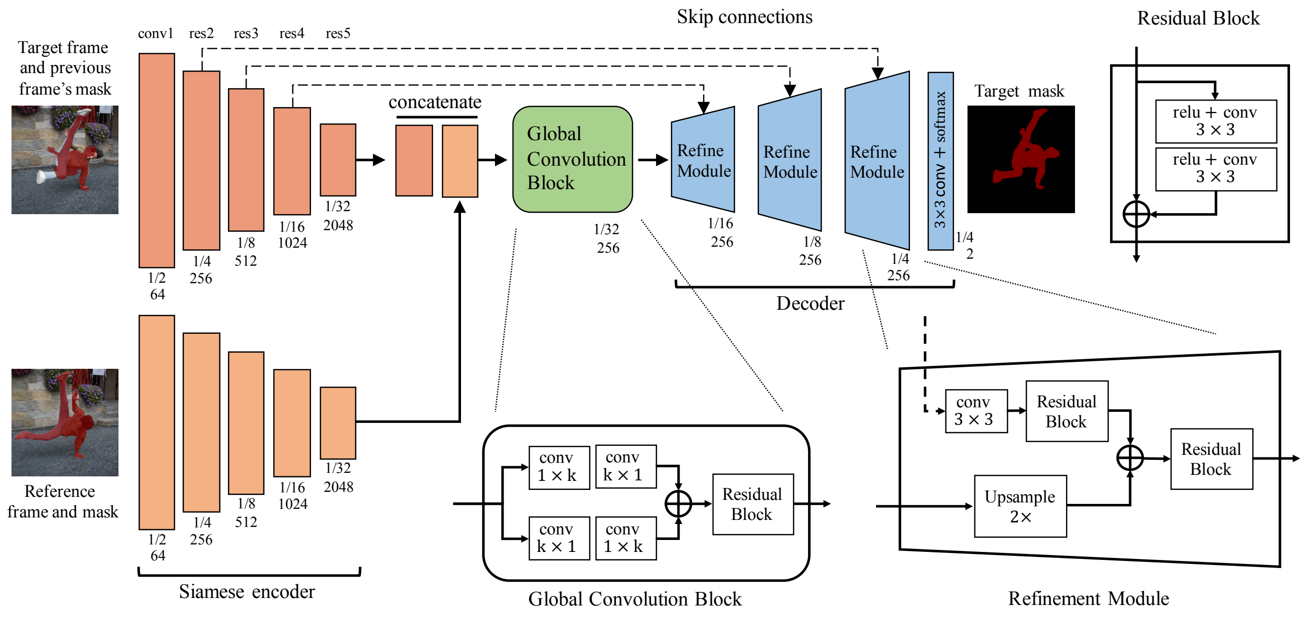


Figure 2: Our network architecture. The network consists of two encoders with shared parameters, a global convolution block, and a decoder. The network is fully convolutional. The relative spatial scales and channel dimensions of feature maps are shown below each block.

Our encoder network is based on ResNet50 [16] and is modified to be able to take a 4-channel tensor by implanting additional single channel filters at the first convolution layer. The network weights are initialized from the ImageNet pre-trained model, except for newly added filters that are initialized randomly.

Global convolution block. The outputs of the two encoder streams are concatenated and fed into a global convolution block. This block is designed to perform global feature matching between the reference and the target streams to localize the target object. To overcome the locality of convolution operations, we adopt global convolution [29] that efficiently enlarges the receptive field by combining $1\times k+k\times 1$ and $k\times 1+1\times k$ convolution layers (k=7 in our implementation). The output of the global convolution block is further processed by one residual block [17]. Note that we remove the batch normalization [19] from the original shape. All convolution layers in this block produce a feature map with 256 channels (*e.g*. the number of filters is 256).

Decoder. The decoder takes the output of the global convolution block and also features in the target encoder stream through skip-connections to produce a mask output. To efficiently merge features in different scales, we employ the refinement module [32] as the building block of our decoder. We make several modifications from the original structure by replacing convolution layers with residual blocks [17], as shown in Fig. 2. Our decoder consists of three refinement modules, a final convolution layer, and a softmax layer to generate the object mask. The size of the mask output is $1/4$ of the input image size. Every convolution layer in the refinement module produces a feature map with 256 channels and the last one produces a two-channel mask map.

3.2. Two-Stage Training

DAVIS-2017 [33, 31] is the largest public benchmark dataset for the video object segmentation, and provides a training set consisting of 60 videos. This is not enough to train our deep network from scratch even though we use pre-trained weights for the encoder. To address this issue, we present a two-stage training scheme. Our network is first trained on simulated samples using static image datasets and then fine-tuned on video segmentation data.

Pre-training on simulated samples. In the first stage, we used image datasets with instance object masks (Pascal VOC [10, 14], ECSSD [34], and MSRA10K [8]) to simulate training samples. For our two-stream encoder, we need both the reference and the target frame data that contain the same object. To automatically generate the training samples, we used the following two strategies.

- Strategy 1: From an image with an object mask, we generate a pair of images by applying two different sets of random transformations (rotation, scaling, color perturbation). We used the Pascal VOC dataset [10, 14] as the source image database.
- Strategy 2: From a pair of a foreground object and a background image, we applied two different sets

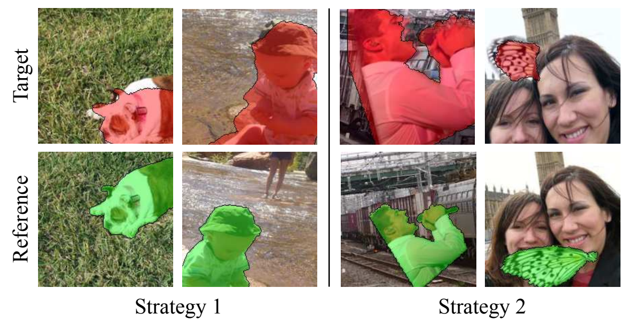


Figure 3: Training samples automatically generated from static images. We have two different strategies for generating training samples as described in Sec. 3.2.

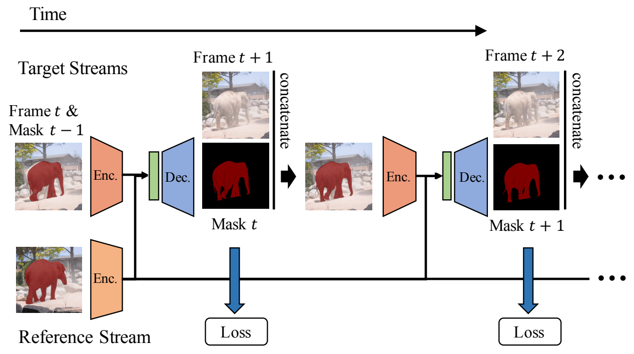


Figure 4: Training with recurrence. We compute training losses at every time step and update our model by the BPTT [43].

of random transformations to the foreground object, then generated a pair of images by blending the transformed foreground images with the background image. We used the saliency detection datasets [34, 8] to segment foreground objects and the Pascal VOC dataset [10, 14] for background images. In addition, we simulated occlusions by using the object mask in the background image (*e.g*. the butterfly in the target image (Fig. 3) is occluded by a person).

For both strategies, we further deformed the mask of the target frame using a random affine transform to simulate the guidance mask from the previous frame similar to [30]. We then randomly crop a training sample that contains at least 50% of the target object from each generated image. Fig. 3 shows some examples generated by the two strategies.

Strategy 1 simulates the environment changes (camera angle, zoom, illumination) of a static scene. Strategy 2 simulates more complex changes and also covers a larger variety of object classes as the saliency detection datasets have more diverse class of objects than the Pascal VOC dataset. The images from Strategy 2 sometimes look unnatural and have blending artifacts, while the images from Strategy 1 are natural without the artifacts. We empirically found that both strategies are helpful, thus we generate training samples using both strategies with an equal probability. We analyze the effect of this pre-training stage in Sec. 4.2.

Fine-tuning on video data. After pre-training on the simulated samples, we fine-tune the network with video segmentation data. By training on real video sequences, our network learns to adapt for long-term appearance changes (between the reference and the target frames) and short-term motions (between the target frame and the previous frame's mask). We trained our network on the DAVIS-2017 training dataset [33, 31] that consists of 60 short HD videos (4029 frames in total) with pixel-level instance label maps. To prepare training samples from a video, we take reference and target frames at random time indices. We just select one instance if there are multiple instances in the video.

The naive fine-tuning explained above may not be compatible with a real test scenario as it does not reflect the error accumulation over time. To resolve this problem, we fine-tune our model with its own estimation that often comes with mistakes. Specifically, we recurrently connect our model through time similar to [18] and feed the softmax (not binarized) output of the previous frame as the guidance mask of the current frame in order to preserve the uncertainty of the estimation. This enables us to use back-propagation-through-time (BPTT) for training the recurrently-connected network [43]. For this training, we use N successive target frames from a random time index of a video. Our recurrently-connected-through-time network is depicted in Fig. 4. We will discuss the effect of fine-tuning on video data with and without applying the recurrence in Sec. 4.2.

3.3. Inference

We assume the groundtruth mask of the first frame is given following the common semi-supervised setting of video object segmentation. We set the first frame as the reference and estimate masks of the remaining frames sequentially. Note that we pass the output probability map of the previous frame as the guidance mask for the target frame without binarization. When testing a video sequence, we compute the feature of the reference (first frame) stream encoder only once and this makes our inference more efficient as shown in Fig. 4. To capture objects at different sizes, we process frames in three different scale inputs (*e.g.* 0.5, 0.75, and 1) and average the results.

Multiple Objects. In the case of multiple objects, we still use the same model but handle the scenario at the inference time. One *naive* way is to run each object independently and assign the label with the largest output probability. Another approach is the *winner-take-all* approach

	OL	PP	OF	$\mathcal{J}$ Mean	$\mathcal{F}$ Mean	Time
PLM [45]	✓	✓		70.0	62.0	**0.3***s*
SegFlow [7]	✓			74.8	74.5	7.9*s*
MSK [30]	✓	✓	✓	79.7	75.4	12*s*
LCT [21]	✓	✓	✓	80.5	77.6	-
MaskRNN [18]	✓		✓	80.7	80.9	-
OSVOS [5]	✓	✓		79.8	80.6	9*s*
OSVOS^S [27]	✓	✓		85.6	**86.4**	4.5*s*
OnAVOS [40]	✓	✓		**86.1**	84.9	13*s*
BVS [28]				60.0	58.8	0.37*s*
OFL [38]				68.0	63.4	120*s*
VPN [20]				70.2	65.5	0.63*s*
SegFlow† [7]				67.4	66.7	-
MaskRNN† [18]			✓	56.3	-	-
OnAVOS† [40]				72.7	-	-
Ours				**81.5**	**82.0**	**0.13***s*

Table 1: Quantitative evaluation on the DAVIS-2016 validation set. We highlight common features of each method: online learning (OL), post-processing (PP), and optical flow input (OF). We group methods according to whether online learning is used or not. Time shows the approximated runtime (seconds per frame). †indicates a variant of each method without online learning and post-processing.

that exploits the disjoint constraint of instances, *i.e.* each pixel cannot belong to multiple instances, by setting non-maximum instance probabilities to zeros at each estimation. The *winner-take-all* approach improved the accuracy of benchmarks compared to the *naive* approach, but is still far from the optimal as it discards beneficial information.

To this end, we propose *softmax aggregation* that combines multiple instance probabilities softly while constraining them to be positive and sum to 1:

$$p_{i,m} = \sigma\big(\text{logit}(\hat{p}_{i,m})\big) = \frac{\hat{p}_{i,m}/(1-\hat{p}_{i,m})}{\sum_{j=0}^{M} \hat{p}_{i,j}/(1-\hat{p}_{i,j})}, \quad (1)$$

where σ and logit represent the softmax and logit functions respectively, $\hat{p}_{i,m}$ is the network output probability of the instance m at the pixel location i, m=0 indicates the background, and M is the number of instances. To compute the probability of the background, we compute the network output of the merged foreground then subtract it from 1. We aggregate the network outputs of instances using Eq. (1) at each time step and pass it to the next frame.

3.4. Implementation Details

We used 256×256 and 256×512 sized patches for the pre-training and the fine-tuning, respectively. In the fine-tuning, we set the number of recurrences as 5 and randomly skipped frames to simulate fast motion. We also augmented all the training samples using a random affine transform. We use Adam [22] optimizer for all of our experiments with a fixed learning rate 1e-5. The pre-training takes about 3 days and the fine-tuning takes about 2 days using a single NVIDIA GeForce 1080 Ti GPU.

	$\mathcal{J}$ Mean	$\mathcal{F}$ Mean
OFL [38]	43.2	-
OSVOS [5]	52.1	-
MaskRNN [18]	60.5	-
MaskRNN† [18]	45.5	-
OnAVOS [40]	61.0	66.1
OnAVOS+ [39]	64.5	**71.1**
Ours	**64.8**	68.6

Table 2: The quantitative evaluation of multi-object video object segmentation on DAVIS-2017 validation set. †: a variant without online learning. +: the result of the challenge entry obtained from an ensemble model. The results for OFL [38] and OSVOS [5] are directly copied from [18].

4. Experiments

We evaluate our method on standard benchmark datasets [31, 33, 24, 11] and compare our performance with state-of-the-art methods. Then, we perform comprehensive ablation and add-on studies to validate the effect of each component of our method.

4.1. Main results

We used DAVIS [31, 33], SegTrack v2 [24], and JumpCut [11] datasets for evaluation: 1) DAVIS-2016 validation set for single object segmentation, 2) DAVIS-2017 validation set and SegTrack v2 for multi-object segmentation, 3) JumpCut for the video cutout scenario. For the DAVIS datasets, we measured the region similarity $\mathcal{J}$ and the contour accuracy using the provided benchmark code [31]. For SegTrack v2 [24] and JumpCut [11], since videos has various resolutions, we rescaled video frames to have 480 pixels on the shorter edge before processing and we measured performance according to the evaluation protocols suggested by the original papers. Result videos are included in the supplementary material. The code and the model are available online.

DAVIS-2016. We compare our method with state-of-the-art methods in Table 1. In the table, we highlight common features of each method. Most of previous methods rely on online learning that fine-tunes a network on the first frame of each test video. Post-processing is often employed to refine the output (dense CRF [23] in [30, 21, 40] and boundary snapping in [5, 27]). Some methods are also aided by additional optical flow information [30, 21].

Among the methods without online learning, our method significantly outperforms all other methods. Compared to methods with online learning, our technique achieves

	BVS	OFL	MSK	OSVOS	MaskRNN	LCT	Ours
IoU	58.4	67.5	70.3	65.4	72.1	77.6	71.1

Table 3: Quantitative results on Segtrack v2 [24]. We report IoU according to [24].

Error	RB [2]	DA [46]	SS [1]	JC [11]	PLM [45]	Ours
$d = 8$	20.0	14.8	15.7	7.21	-	**4.89**
$d = 16$	28.7	23.7	18.9	9.82	9.55	**6.91**
$d = 32$	39.9	34.9	27.0	16.2	-	**10.3**

Table 4: Performance on JumpCut [11] (lower is better). According to the standard evaluation protocol of this dataset, we sample multiple key-frames $(0, 16, \ldots, 96)$ from a video and propagate each by the transfer distance d (frames). The errors are measured at the end of each propagation as the ratio between wrongly classified area and the actual object area.

comparable accuracy without further online fine-tuning and post-processing.

In the table, the runtimes of OSVOS [5] and OFL [38] are from [40] and [30], respectively. Otherwise, we put the runtimes reported in the original papers. Even considering the differences in implementations and running environments, our method shows incomparable efficiency against previous methods thanks to our efficient inference without online learning and post-processing.

DAVIS-2017. In Table 2, we report the result of multi-object video segmentation on DAVIS-2017. Our method achieves state-of-the-art performance, which is comparable with the challenge entry of [40] that comes with bells and whistles including an ensemble model [39].

Generalization on SegTrack v2. We evaluate our model on SegTrack v2 [24]. We estimate object masks using exactly the same model and parameters as the DAVIS experiment. In Table 3, our method shows competitive performance with the latest methods that use online learning. Note that, as we avoid online learning, our network trained on the DAVIS-2017 training set is completely blind to the domain of SegTrack v2 data. Thus, this experiment demonstrates the generalization performance of our method.

Video cutout on JumpCut. To evaluate our method on the video cutout scenario, we further test our network on the JumpCut [11] dataset. Again, our network is totally blind to JumpCut as we use the model pretrained on DAVIS without any modification. As shown in Table 4, our method significantly outperforms all previous methods.

Fig. 7 shows some qualitative visual results. Our method works well on various types of objects and motions, and is able to handle multiple instances well.

	Our	-Ref	-Prev	-PT	-FT	-Rec
$\mathcal{J}$ Mean	81.5	68.3	73.5	68.6	55.0	74.3
$\mathcal{F}$ Mean	82.0	68.2	74.2	68.9	59.1	74.8
Δ	-	-13.5	-7.9	-13.0	-24.7	-7.2

Table 5: Ablation study on the DAVIS-2016 validation set. We compare models with ablations from our complete model, and report the performance change with respect to our full model in terms of the global mean $(\mathcal{J} + \mathcal{F})/2$).

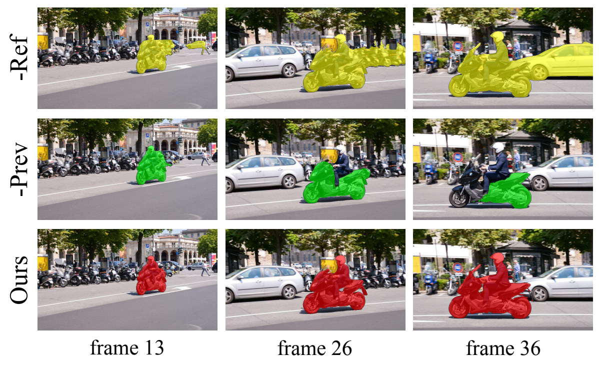


Figure 5: The effect of the network inputs. Results from ablated models (-Prev and -Ref) and our complete model is shown.

4.2. Ablation Study

We run an extensive ablation study to demonstrate the effects of different components of our method. We summarize the results in Table 5.

Network inputs. Our model takes two sets of image and mask, one for the reference frame and the other for target frame. We investigate the importance of each input stream of our model. If we block the access to the reference input stream, the network should propagate the previous mask to the current frame without reference information. To evaluate the setup, we zero out the reference stream input rather than modifying our network structure. We named this model *-Ref*. On the other hand, if we do not feed the previous mask, the network should detect the target object in the reference frame without any temporal prior. To simulate this setup, we zero out to the previous mask input at the target stream. We named this model *-Prev*. Note that, for this experiment, we train two separate models using our two-stage training scheme.

Table 5 shows the experimental result. In both ablation setups, we observe significant performance drops. The model *-Ref* has a similar setup with [30] but without online learning. The low score of this setup shows that simply refining the previous frame mask according to the current frame image is not enough to get good results as it is prone to drifting and cannot handle occlusions. [30] overcome this

	Our	+OL	+CRF	+GRU
$\mathcal{J}$ Mean	81.5	82.4	81.9	79.6
$\mathcal{F}$ Mean	82.0	82.2	79.9	81.0
time	0.13s	+ 1.74s	+ 2.53s	+ 0.01s

Table 6: Add-on study on the DAVIS-2016 validation set. We compare models with additional components added to our model. We also provide additional time per frame with each component.

issue by employing online learning and optical flow.

For the model *-Prev*, while the problem setup is similar to [45], our *-Prev* model performs better than [45] (+3.5 in terms of $\mathcal{J}$ mean). We argue that the improvement comes from our pre-training. Nonetheless, this model still suffers from its structural limitation as it purely depends on the appearance of the target object in the reference frame, and has difficulty handling object appearance changes or multiple instances with a similar appearance. This limitation is resolved in [40] by online adaptation which updates the model at every time step.

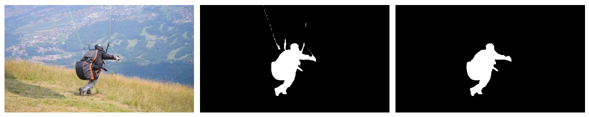
In Fig. 5, we compare the ablation variants with our complete model to demonstrate their limitations. *-Ref* drifts to the background textures and *-Prev* fails to adapt to the appearance changes over time, while our complete model shows a stable result.

Training stages. As described Sec. 3.2, we train our model through two training stages: pre-training on simulated samples and fine-tuning on video data. In Table 5, we empirically verify the effect of each training stage. The model *-PT* skipped the pre-training stage and the model *-FT* skipped the fine-tuning stage. In addition, to highlight the effect of the recurrence while training on video data, the model *-Rec* is trained with both stages but without the recurrence during the fine-tuning. As shown in Table 5, both training stages are important to get accurate results and training with recurrence further boosts our performance to reach the state-of-the-art.

4.3. Add-on Study

While our model by itself is able to achieve the state-of-the-art performance, we investigate how additional components can further boost the performance. Table 6 summarizes the result of this add-on study on the DAVIS-2016 validation set.

Online learning. Similar to other methods with online learning, we fine-tune our model on the reference frame of a test video to adapt the model to the appearance of the target object. To train our model using a single frame, we use Strategy 1 of in Sec. 3.2. *i.e.*, we automatically generate both the reference and the target frame inputs from a single image by applying different random transformations. For this online fine-tuning, we set learning rate as 1e-7 and the number of iteration as 1000 with ADAM optimizer [22].



(a) Scene (b) Before CRF (c) After CRF

Figure 6: The effect of the CRF.

While online fine-tuning improves ($\mathcal{J}$: 81.5 to 82.4) from our model, it is relatively small compared to previous methods [30, 45, 18] that experience more than 10 point gains. This result is natural as the performance of all the state-of-the-art methods is saturated at a similar point on the dataset. We argue that this implies that our model already exploits the appearance information of the target object from the reference stream input. Thus, unlike previous methods [5, 40, 27, 21], we can avoid the considerable computational overhead of online learning (*e.g.* it takes more than 2 minutes per video with a high-end GPU in Table 6).

Refinement with CRF. We apply the dense CRF [23] as a post-processing to refine our outputs. We find the hyper-parameter of the dense CRF using a grid search on the validation set following [6, 21].

The CRF affects the two measures differently; it boosts $\mathcal{J}$ mean (+0.4), but degrades $\mathcal{F}$ mean (-2.1). We observed that the CRF helps to refine mask boundaries to be aligned with objects and increases the overall overlapping area ($\mathcal{J}$) but sometimes smooths out fine details where $\mathcal{F}$ measure is very sensitive as shown in Fig. 6 (*e.g.* parachute strings). We argue that our network structure, especially the refinement module used in the decoder, is able to recover fine details without additional post-processing, unlike previous methods [30, 21, 36] that are based on the DeepLab architecture [6] that produces the output at a coarser scale ($1/8$ compared to $1/4$ of ours).

Visual memory (RNN). Inspired by [36], we augmented our model with visual memory. While our current training scheme uses recurrence (Fig. 4), it can be helpful to have an extra memory module that directly connects internal features at different time steps. To this end, we extend our model to have visual memory by implanting a RNN cell to the output of the global convolution block. In particular, we combine the feature from the previous time step with the current one using a 3×3 convolutional GRU unit [9, 3]. Note that we insert the GRU cell after the pre-training stage (that uses synthetic samples) since RNN training requires sequential data. We randomly initalized the GRU weights and trained it after fixing the weights of other network filters. Following [44], we employ a curriculum learning scheme that increases the number of recursion by 1 every 3000 iterations until it reaches 5. After this GRU training, we fine-tune all the weights together.

As shown in Table 6, we get no improvement with an additional GRU unit. We conjecture that this is due to

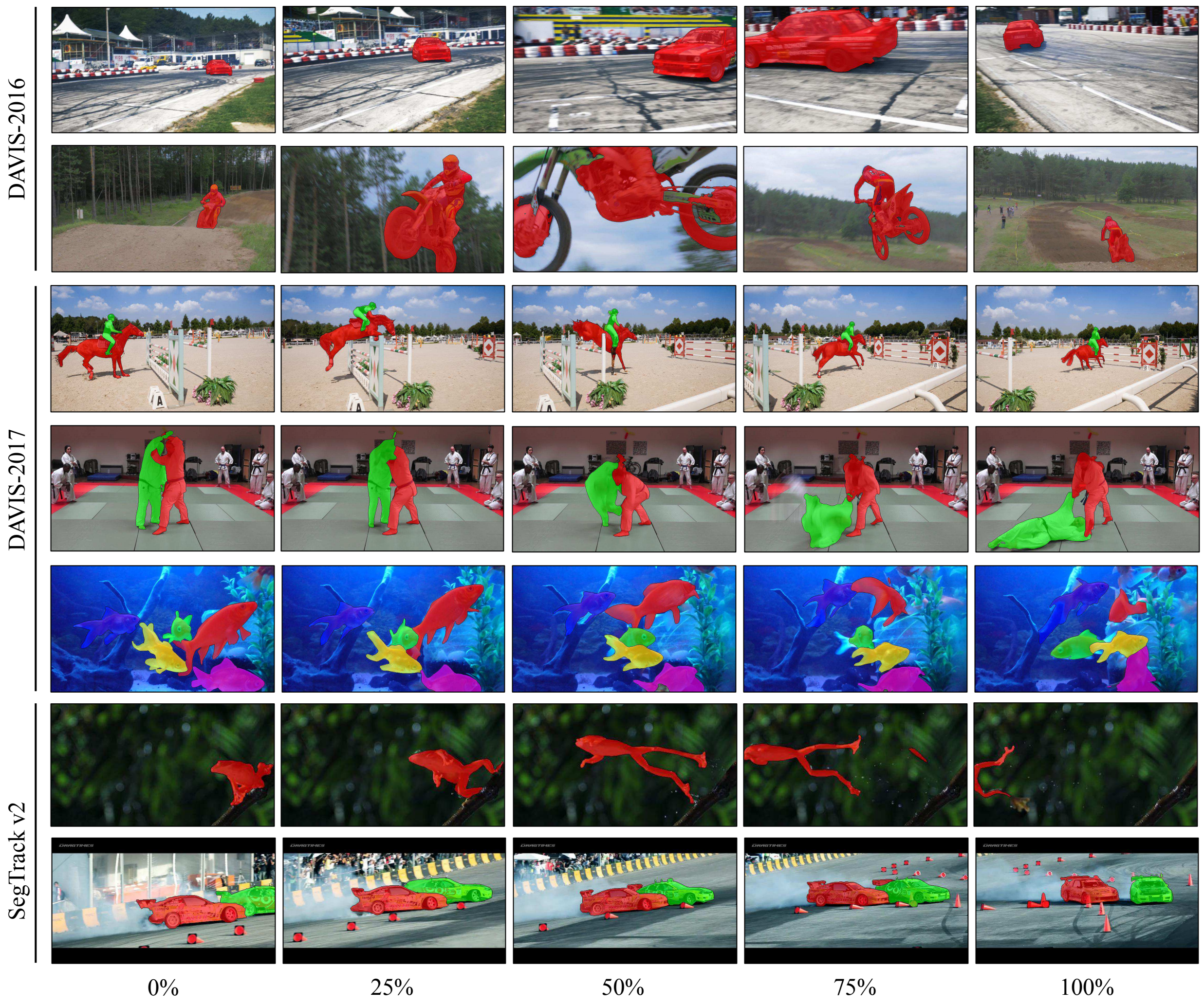


Figure 7: The qualitative results on DAVIS and SegTrack v2. Frames are sampled uniformly.

over-fitting as we observe that the training loss is much lower than our model (mean cross-entropy: 0.012 *vs.* 0.02). In practice, the video training data is very limited (60 sequences in total) to train RNNs.

5. Conclusion

In this paper, we have presented a novel approach for semi-supervised video object segmentation. We demonstrate that our Siamese encoder-decoder network trained using a two-stage scheme reaches the current state-of-the-art performance without online learning and post-processing, making it much faster than comparable methods.

There are several future directions for this problem. First, we could incorporate ROI extraction techniques such as ROI-pooling [12] and ROI-align [15]. In our experiments, we found that the estimation is sensitive to object scale. Currently, we alleviate it with multi-scale inference. The ROI of a target object in the previous frame captures the object scale. Thus, ROI extraction techniques can make the problem easier by normalizing scales. While we failed to get much improvement from RNNs due to over-fitting, we still believe that long-term memory has the potential to handle challenging scenarios (*e.g.* momentary occlusions) if we have more video training data. We are also interested in extending our approach to be interactive. Currently we only use the first frame as the reference frame following the standard benchmark setup. However, our model is flexible and fast enough to allow users to change the reference frame (by choosing one of visually confirmed intermediate results or providing additional frame annotations).

Acknowledgment. This work was supported by the ICT R&D program of MSIT/IITP (2017-0-01772, Development of QA systems for Video Story Understanding to pass the Video Turing Test) and (2014-0-00059, Development of Predictive Visual Intelligence Technology).

References

[1] S. Avinash Ramakanth and R. Venkatesh Babu. Seamseg: Video object segmentation using patch seams. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 376–383, 2014. 6

[2] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: robust video object cutout using localized classifiers. In *ACM Transactions on Graphics (ToG)*, volume 28, page 70. ACM, 2009. 2, 6

[3] N. Ballas, L. Yao, C. Pal, and A. Courville. Delving deeper into convolutional networks for learning video representations. *arXiv preprint arXiv:1511.06432*, 2015. 7

[4] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *European Conference on Computer Vision*, pages 282–295. Springer, 2010. 2

[5] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017. 1, 2, 5, 6, 7

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016. 7

[7] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 5

[8] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2015. 3, 4

[9] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 7

[10] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 3, 4

[11] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen. Jumpcut: non-successive mask transfer and interpolation for video cutout. *ACM Transactions on Graphics (TOG)*, 34(6):195, 2015. 2, 5, 6

[12] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 8

[13] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2141–2148. IEEE, 2010. 1, 2

[14] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 991–998. IEEE, 2011. 3, 4

[15] K. He, G. Gkioxari, P. Dollr, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 8

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition*, pages 770–778, 2016. 3

[17] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016. 3

[18] Y.-T. Hu, J.-B. Huang, and A. Schwing. Maskrnn: Instance level video object segmentation. In *Advances in Neural Information Processing Systems*, 2017. 1, 2, 4, 5, 7

[19] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015. 3

[20] V. Jampani, R. Gadde, and P. V. Gehler. Video propagation networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 5

[21] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele. Lucid data dreaming for object tracking. *arXiv preprint arXiv:1703.09554*, 2017. 1, 2, 5, 7

[22] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 5, 7

[23] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, pages 109–117, 2011. 5, 7

[24] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2192–2199, 2013. 5, 6

[25] W. Li, F. Viola, J. Starck, G. J. Brostow, and N. D. Campbell. Roto++: Accelerating professional rotoscoping using shape manifolds. *ACM Transactions on Graphics (TOG)*, 35(4):62, 2016. 2

[26] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[27] K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. Video object segmentation without temporal information. *arXiv preprint arXiv:1709.06031*, 2017. 1, 2, 5, 7

[28] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 743–751, 2016. 1, 2, 5

[29] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun. Large kernel matters–improve semantic segmentation by global convolutional network. *arXiv preprint arXiv:1703.02719*, 2017. 3

[30] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 2, 4, 5, 6, 7

[31] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3, 4, 5

[32] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *European Conference on Computer Vision*, pages 75–91. Springer, 2016. 3

[33] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 3, 4, 5

[34] J. Shi, Q. Yan, L. Xu, and J. Jia. Hierarchical image saliency detection on extended cssd. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):717–729, 2016. 3, 4

[35] P. Tokmakov, K. Alahari, and C. Schmid. Learning motion patterns in videos. *arXiv preprint arXiv:1612.07217*, 2016. 2

[36] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 7

[37] D. Tsai, M. Flagg, A. Nakazawa, and J. M. Rehg. Motion coherent tracking using multi-label mrf optimization. *International Journal of Computer Vision*, 100(2):190–202, 2012. 1, 2

[38] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3899–3908, 2016. 5, 6

[39] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for the 2017 davis challenge on video object segmentation. *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*, 2017. 5, 6

[40] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *British Machine Vision Conference*, 2017. 1, 2, 5, 6, 7

[41] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen. Interactive video cutout. In *ACM Transactions on Graphics (ToG)*, volume 24, pages 585–594. ACM, 2005. 2

[42] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3395–3402, 2015. 2

[43] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990. 4

[44] J. Yang, S. E. Reed, M.-H. Yang, and H. Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *Advances in Neural Information Processing Systems*, pages 1099–1107, 2015. 7

[45] J. S. Yoon, F. Rameau, J. Kim, S. Lee, S. Shin, and I. S. Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 1, 2, 5, 6, 7

[46] F. Zhong, X. Qin, Q. Peng, and X. Meng. Discontinuity-aware video object cutout. *ACM Transactions on Graphics (TOG)*, 31(6):175, 2012. 6