Fast User-Guided Video Object Segmentation by Interaction-and-Propagation Networks

Seoung Wug Oh, Yonsei University
Joon-Young Lee, Adobe Research
Ning Xu, Adobe Research
Seon Joo Kim, Yonsei University

Abstract

We present a deep learning method for the interactive video object segmentation. Our method is built upon two core operations, interaction and propagation, and each operation is conducted by Convolutional Neural Networks. The two networks are connected both internally and externally so that the networks are trained jointly and interact with each other to solve the complex video object segmentation problem. We propose a new multi-round training scheme for the interactive video object segmentation so that the networks can learn how to understand the user's intention and update incorrect estimations during the training. At the testing time, our method produces high-quality results and also runs fast enough to work with users interactively. We evaluated the proposed method quantitatively on the interactive track benchmark at the DAVIS Challenge 2018. We outperformed other competing methods by a significant margin in both the speed and the accuracy. We also demonstrate that our method works well with real user interactions.

1. Introduction

Video object segmentation is a task of separating a foreground object from a video sequence. It is an essential task in video editing with a wide range of applications from the consumer-level video editing to the professional TV and movie post-production. This problem is often solved by either a fully-automatic approach (*i.e*. unsupervised foreground object segmentation [35]) or a semi-supervised approach (*i.e*. ground-truth object masks are given on few frames [5, 28]). However, both solutions have limitations in reflecting a user's intention or refining incorrect estimations.

Interactive video segmentation can potentially resolve this issue by allowing user intervention given in a user-friendly form such as scribbles [37, 31, 2]. However, existing interactive methods require a lot of user interactions to obtain results with acceptable quality for video editing applications. In this paper, we aim to develop an interactive video object segmentation technique that can estimate accurate object masks in a video sequence with minimal user interactions.

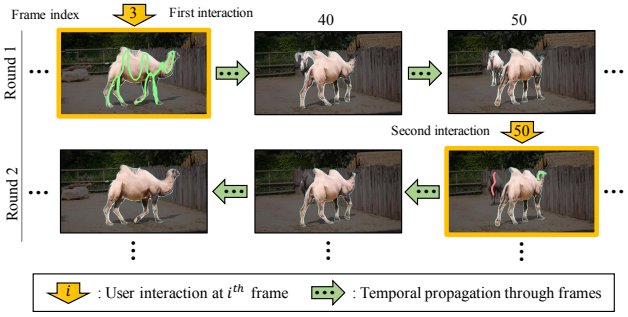

Figure 1: We propose a method that can estimate object masks in a video by interacting with a user. The mask of a target object is generated using user annotations (*e.g*. scribbles at frame 3), and the computed mask is propagated to compute the masks for the entire video. The user can repeatedly provide additional feedback (*e.g*. scribbles on false positive and false negative at frame 50) to refine the segmentation masks. Our method generates high-quality object masks with minimal user interactions and time budget.

Interactive video cutout methods usually follow the procedure of the rotoscoping [4, 20], where a user sequentially processes a video frame-by-frame. In this scenario, the user verifies and updates the object mask with multiple interactions at every frame. This rotoscoping-style interaction requires a lot of effort and is more suitable for professional uses that require high-quality results.

Recently, Caelles *et al*. [6] introduced another workflow for the video object cutout that can minimize the user's effort. In this scenario, which we call as the *round-based interaction*, the user provides annotations on a selected frame and an algorithm computes the segmentation maps for all video frames in a batch process. To refine the results, the process of user annotation and segmentation map computations are repeatedd until the user is satisfied with the results. This round-based interaction is useful for consumer-level applications and rapid prototyping for professional usage,

where the efficiency is the main concern. One can control the quality of the segmentation according to the time budget, as more rounds of interactions will provide more accurate results.

In this paper, we present a deep learning based method for the interactive video object segmentation tailored to the round-based interaction scenario (Fig. 1). While several deep learning approaches for video object segmentation have been proposed [5, 28], they are usually too slow for the interactive scenario as they rely heavily on online learning. Even with a fast video segmentation algorithm [26], designing a deep neural network (DNN) and its training mechanism for the interactive segmentation scenario remains as a challenge.

To solve this challenging problem, we propose the Interaction-and-Propagation Networks and an effective training method. Our framework consists of two deep CNNs, each of which is dedicated to the core operations *interaction* and *propagation* respectively. The interaction network takes the user annotation (*e.g*. scribbles) to segment the foreground object. The propagation network transfers the object mask computed in the source frame to other neighboring frames. These two networks are internally connected using our feature aggregation module and are also externally connected so that each of them takes the other's output as its input.

The two networks are trained jointly to adapt to each other, which reduces unstable behaviors between the two operations. We also propose the concept of multi-round training, which is specifically designed to simulate a real testing scenario of the interactive video segmentation. In this training strategy, a number of user feedback cycles and the response of networks form a single training iteration (see Fig. 3). This new training scheme greatly improves the performance of our model.

Our framework is quantitatively evaluated on the interactive track benchmark at the DAVIS Challenge 2018 [6] and achieves the state-of-the-art performance with a big gap compared to other competing methods [27]. We also demonstrate the usefulness of our method with real interactive cutout use-cases. We will release the source code that contains our trained model and the graphical user interface.

2. Related Work

2.1. Video Object Segmentation

We categorize the video object segmentation into three categories based on different types of user interactions.

Unsupervised Methods. In the unsupervised setting, there is no user interaction. The unsupervised approaches run automatically but they can only segment visually salient objects based on the appearance or the motion. For example, Jain *et al*. [18] combine an appearance model with an optical flow model to segment generic objects in videos. Similarly, Tokmakov *et al*. [35] use a motion estimation network with a recurrent neural network to segment moving foregrounds. The fundamental limitation of the unsupervised methods is that users have no means to select the object of interest.

Semi-supervised Methods. In the semi-supervised setting, the ground-truth mask of an object in the first frame is provided. The goal is to propagate the object mask throughout the entire video sequence. Many recent approaches [5, 36, 24] employ the online learning by fine-tuning deep network models at the testing time in order to remember the appearance of the target object on the given object mask. Then the object segmentation is performed for each frame. Instead of employing the online learning, Jampani *et al*. [19] propagate the object mask by bilateral filtering. Oh *et al*. [26] use Siamese two-stream networks and leverage synthetic training data. Although the semi-supervised methods do not have the limitation of the unsupervised methods, they require a fully annotated object mask in the initial frame, which can be expensive to acquire. Additionally, semi-supervised methods rely on extra information such as fully annotated masks or external tools to further improve the output quality.

Interactive Methods. In the interactive setting, users can provide various types of inputs (*e.g*. bounding box, scribbles, or masks) to select an object of interest in the beginning. Users can also provide more interactions to refine the segmentation results. The goal of this interactive approach is to achieve satisfactory segmentation results with a minimum number of user interactions. Many interactive methods [37, 31, 9, 2, 4, 20] have been proposed. [37, 31, 33] solve spatio-temporal graphs with handcrafted energy terms. Some methods find the corresponding patches between a target frame and a reference frame, then utilize local classifiers [2, 44] or an existing patch-match algorithm [9]. [1, 20] solve the segmentation task by tracking. Recently, [3, 6] proposed deep-learning based methods by modifying semi-supervised methods to the interactive scenario. Benard and Gygli [3] use the deep interactive image segmentation method [39] to select an object given initial strokes or clicks, and use the semi-supervised video object segmentation method [5] to propagate the object mask. Compared to such a simple combination of two separate methods, we carefully design two module networks to interact with each other and train the whole networks jointly using our new multi-round training scheme.

2.2. Interaction with Deep Neural Networks

Recently, several methods have been introduced for integrating user interaction with deep neural networks for various interactive tasks. Xu *et al*. proposed to transform clicks [39] or bounding boxes [38] into Euclidean distance

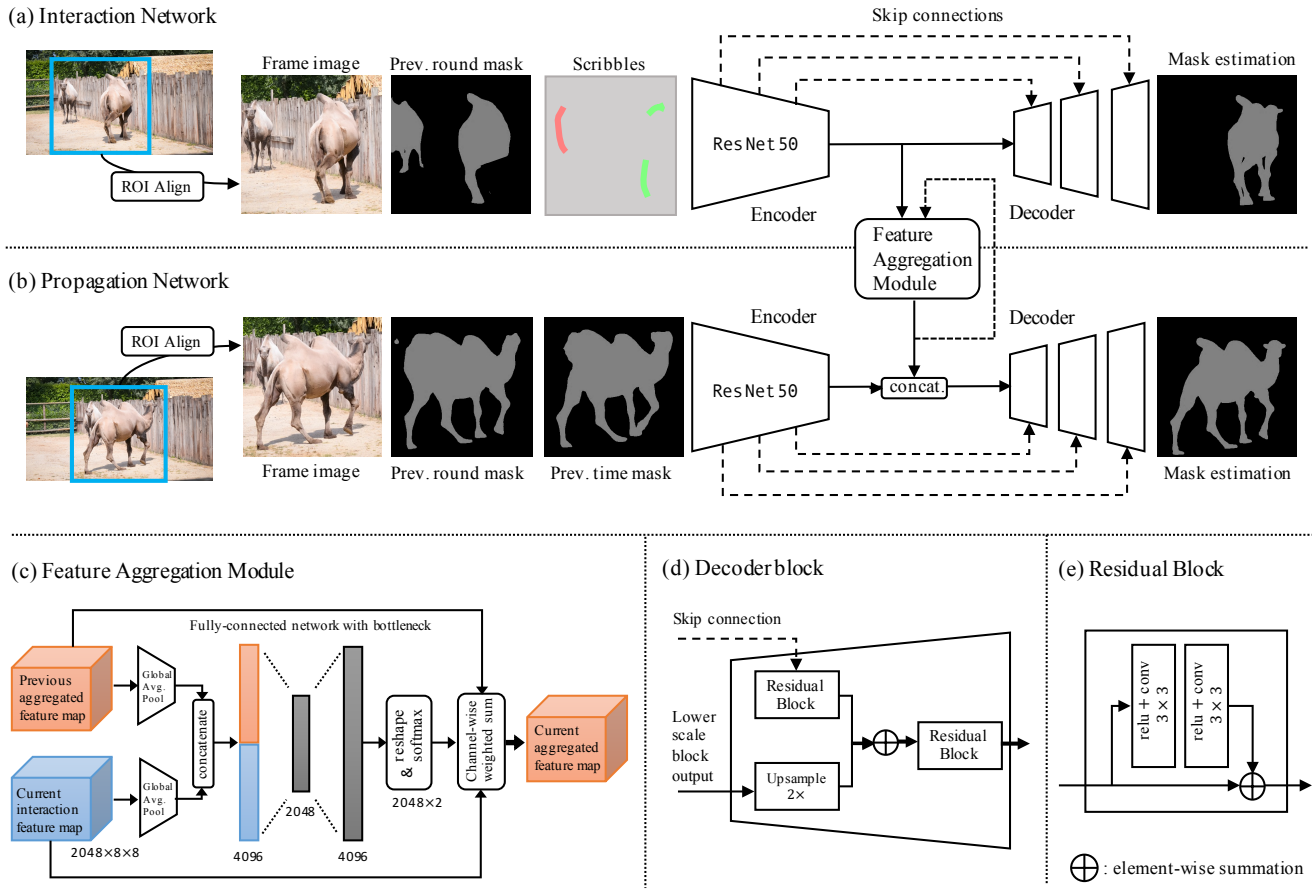

Figure 2: The overall network structure. We have two deep networks dedicated each to (a) interaction and (b) propagation tasks. The two networks are internally connected by (c) our feature aggregation module and also externally connected to take the other's output as their input (a, b). Please see Sec. 3.1 for the details.

maps for the interactive image segmentation. Zhang *et al*. [43] incorporated a user's color selection for the image colorization. Sangkloy *et al*. [32] and Isola *et al*. [16] used sketches to help generate realistic natural images.

Different from the above interactive approaches that only consider an interaction given once onto an image, our model considers multiple user inputs possibly drawn onto different video frames. The sequence of multiple user interactions is aggregated by a specially designed recurrent block called the feature aggregation module. In addition, we use the segmentation results from previous rounds as an additional channel, in order to consider the unique characteristics of the interactive video segmentation.

3. Method

Given user annotations on a video frame (*e.g*. scribbles drawn on the foreground and background pixels of an image), we aim for cutting out the target object in all frames of the given video. From the initial user input, we generate object masks of all frames solely based on the user annotation. If the user provides additional feedback annotations after reviewing the generated masks, our method refines the object masks based on both additional user annotations and the previous mask estimation results.

To this end, we define two basic operations for the task: interaction and propagation. Two deep CNNs dedicated for each operation are proposed as shown in Fig. 2 (a),(b). The interaction network generates the object mask (or refines the previous results) for the annotated frame according to the user inputs. The propagation network generates the object masks (or refines the previous results) by temporally propagating the object mask information both forward and backward starting from the frame with user annotation.

To prevent the error accumulation due to drifts and occlusions during the propagation, the propagation network refers to a reliable visual memory similar to [26, 41, 42]. While [26, 42] employ a Siamese network to access the reference frame directly, we modified the framework to make it more suitable for the interactive video object segmentation.

Specifically, as the most reliable information is contained in the user annotated frames in the interactive scenario, we allow the propagation network to access the features of the interaction network. In addition, we propose a feature aggregation module that accumulates all the previous reference information encoded by the interaction network. This reference-guided propagation is effective, especially for the long-term propagation.

We refer to the series of operations consisting of both the user interactions on one frame and a number of consecutive propagation towards both ends as a *round* (see Fig. 3). Users are able to repeat several rounds of interactions to refine the segmentation results until they are satisfied with the results as shown in Fig. 1. Both networks operate on the results obtained from the previous round. We use the same networks for every round.

3.1. Network Design

We have two networks, interaction and propagation, and both networks are constructed as an encoder-decoder structure that can effectively produce a sharp mask output. We adopt the ROI align before the encoder to make our networks to pay attention to the region of interest (the area around the target object) [13]. We take ResNet50 [14] (without the last global pooling and fully-connected layers) as the encoder network, and also modify it to be able to take additional input channels (*e.g.* scribbles and the previous masks) by implanting additional filters at the first convolution layer [28, 39]. The network weights are initialized from the ImageNet pre-trained model, except for the newly added filters which are initialized randomly.

The decoder takes the output of the encoder and produces an object mask. To reconstruct a sharp mask by fully exploiting the information at different scales, the decoder additionally takes intermediate feature maps inside the encoder through skip connections. We make modifications to the feature pyramid networks [21, 29] by adding residual blocks [15] and use it as the building block of our decoder, as shown in Fig. 2 (d),(e). The decoder estimates the object mask in a quarter scale of an input image. For the multi-object scenario where scribbles for each object are given, we first estimate masks for each object then merge the masks to get the multi-object mask using the soft aggregation proposed in [26].

Interaction Network. The input to the interaction network consists of a frame, the object mask from the previous round (if available), and two binary user annotation maps for the positive and the negative regions respectively. The inputs are concatenated along the channel dimension to form an input tensor $\mathbf{X_i} \in \mathbb{R}^{6\times H\times W}$. The object mask is represented as a probability map filled with values between 0 and 1. If no previous mask is available (*e.g.* at the first round), we feed a neutral mask filled with 0.5 for all pixels. The output of this network is $\hat{\mathbf{Y}}_\mathbf{i} \in \mathbb{R}^{H\times W}$, the probabilities of the target object at every pixel.

Propagation Network. The input to the propagation network consists of a frame, the object mask obtained at the previous frame, and the object mask obtained at the previous round. Similar to the interaction network, the inputs are concatenated along the channel dimension to be a tensor $\mathbf{X_p} \in \mathbb{R}^{5\times H\times W}$. The two object masks are represented with probabilities and the neutral mask is used if the mask is not available. Different from the interaction network, the decoder of this propagation network additionally takes the reference feature map which is computed by our feature aggregation module. The reference feature map and the encoder output of this propagation network are concatenated along the channel dimension and are fed into the decoder.

Feature Aggregation Module. In the interactive video object segmentation, the system often takes multiple user annotations in different frames through multiple rounds. It is important to exploit all previous user inputs for good performance. To achieve this, we propose a feature aggregation module which is specially designed for accumulating information of the target object from all user interactions. We use the encoder output of the interaction network to generate reference feature maps. We update the feature maps recurrently when a new user interaction triggers the interaction network. We design this module to be able to select memorable features by self-attention. As shown in Fig. 2 (c), the module first performs a global average pooling on the spatial dimension of the feature maps to obtain compact feature vectors. The vectors are concatenated and fed into two fully-connected layers with a bottleneck. The outputs of the layers are two channel-wise weight vectors (α and β) after reshaping and a softmax. We place the softmax layer to make sure that $\alpha + \beta = \mathbf{1}$. The two feature maps are channel-wise weighted by α and β, then merged by the summation: $\mathbf{A}_r = \alpha \odot \mathbf{A}_{r-1} + \beta \odot \mathbf{R}_r$. $\mathbf{A}_r$ and $\mathbf{A}_{r-1}$ are the aggregated reference feature map at the round r and $r-1$ respectively, and $\mathbf{R}_r$ is the encoder output of the interaction network at the round r, and $\odot$ is an element-wise multiplication on the channel dimension.

Region of Interest (ROI). While fully convolutional networks for image segmentation [23] can handle image inputs in any resolution, the performance heavily relies on the absolute scale of objects. For example, small objects are easily missed and objects larger than the receptive field need to be estimated by observing only a part of the objects. This issue can be addressed when the network knows where to look. In our case, we can reason about the region of interest (ROI) from the guidance (*e.g.* scribbles and masks).

To take advantage of the guidance, we first compute a tight box that contains all available guiding information (which include user scribbles, the mask from the previous

frame, and the mask from the previous round) and set the ROI to a box that is computed by doubling each side of the tight box. Then, the ROI area for all the inputs is bilinearly warped into a fixed size (*e.g.* 256×256 in our implementation) before we feed them into the encoders [17, 13]. Finally, the prediction made within the ROI is inversely warped and pasted back to the original location. The training losses become scale-invariant as they are computed in the ROI-aligned space, and this enables us to not use the complex balanced loss functions [5]. Note that we set ROI as the whole image at the first round and start to compute ROI using the guidance from the second round.

3.2. Training

Multi-round Training. For the best testing performance, we make our training loop close to the real testing scenario: a user interacts with our model multiple times while providing feedback in the forms of scribbles on multiple frames. We propose a new multi-round training scheme where a single training sample consists of multiple rounds of user interactions. At every round, our model is trained to refine the previous round's results by understanding the user's intention (interaction network) and temporally propagating the object mask (propagation network). Two networks are trained jointly by making an estimation using the previous estimation that can be inferred from the other network. Losses are computed at every intermediate prediction and the back-propagation is performed at every loss computation to update the parameters of the networks. At each round, user inputs are synthesized by simulating user behaviors. Fig. 3 shows an example of a single training iteration in our multi-round training scheme.

User Scribble Synthesis. One challenge in training an interactive model is collecting user input data. For our scenario where a user provides scribbles as feedback, it is not feasible to collect large training data. Instead, we train our model with synthetically generated user interactions. In the first round, positive scribbles are sampled from the foreground region. In the following rounds, scribbles are synthesized within false negative and false positive areas where the areas are computed using the ground-truth mask. We sample positive scribbles from the false negative area and negative scribbles from the false positive area.

We use morphological skeletonization to automatically generate realistic scribbles similar to [6]. Given a candidate area to sample scribbles, we first remove small false estimations isolated from the main body by repeating a binary morphological opening operation. Then, we perform the skeletonization of the mask to get either positive and negative scribbles within the target area. We use a fast implementation of the thinning algorithm [11] for the skeletonization.

A concern can be raised about the gap between the simulated and the real scribbles. We empirically validate that our model trained with simulated user scribbles works well with real user interactions as shown in our demo video.

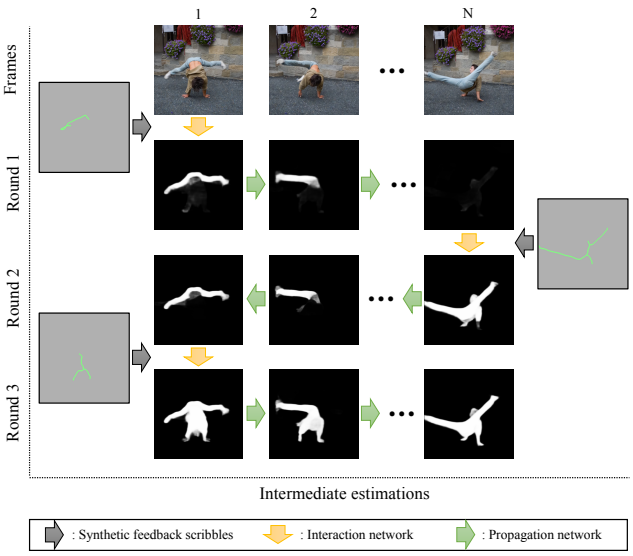

Figure 3: An example of a single training iteration in our multi-round training scheme. The multiple rounds of the network feed-forwarding form a single training iteration so that the networks can experience a real testing scenario and learn how to understand user intention and update incorrect estimations. Training losses are computed at every intermediate estimation.

Pre-training on Images. It is widely known that training deep networks requires a large amount of data. However, video data that comes with object masks are limited due to laborious human annotation process. We bypass the issue by employing two-stage training where our networks are first pre-trained on synthetic image data and then are fine-tuned on real video data. The idea that trains a video segmentation network on image data was proposed in [28], and we follow the data simulation method in [26]. The method produces a set of reference and target frame pairs by applying random affine transforms and object composition. This pre-training is similar to training on videos, but temporal propagation is limited to a single step as there are no consecutive frames.

Implementation Details. For the pre-training, we combine multiple image datasets that come with object masks (salient object detection – [34, 7], semantic segmentation – [8, 12, 22]). After the pre-training, we use the video data from the training subset of DAVIS [30], GyGo [10], and Youtube-VOS [40] to train our networks.

To sample training data, we first resize video frames to be 480-pixels on the shorter edge while keeping the aspect ratio. Then, N consecutive 400×400 sized patches are

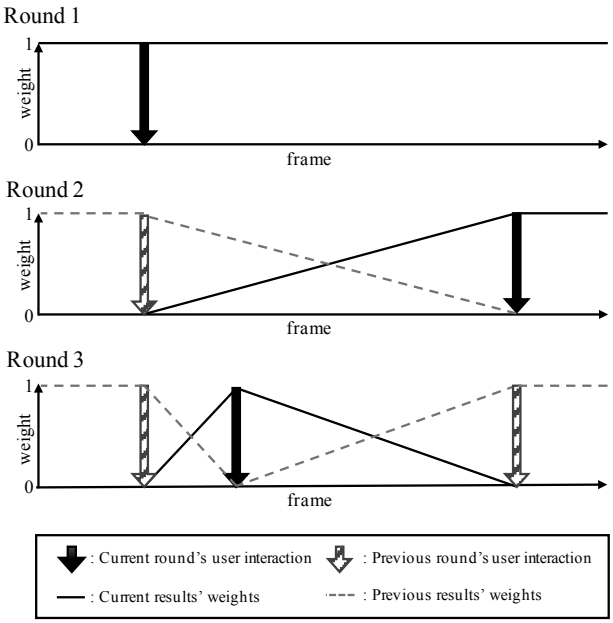

Figure 4: Round-based Testing Scheme. At each round, we update previous object masks with new estimations by the weighted averaging. Solid lines and dashed lines indicate mask updating weights for current estimation and previous estimation, respectively. Weights are inversely proportional to the propagated distance.

sampled from a random location of the video, where N is the length of a training video clip. We randomly skipped frames to simulate fast motion and N is gradually increased from 4 to 8 during training. We also augment all the training samples using random affine transforms. The number of rounds also grows from 1 to 3 during training. The loss is computed by the cross-entropy function and we use Adam optimizer with a fixed learning rate of 1e-5. The training with video data takes about 5 days using a single NVIDIA GeForce 1080 Ti GPU.

3.3. Testing Scheme

One potential issue observed during our testing is that the propagated mask may be worse than the mask from the previous round. This happens especially when the destination is far from the user-selected frame.We conjecture that the long-term propagation may be unstable as our model is trained on short video clips. To address this issue, we modified our testing scheme in two ways; continuous updating and restricted propagation. In continuous updating, we update the previous round's masks with newly estimated masks by the weighted average. The weighting factor is inversely proportional to the propagated distance, and different weighting functions such as a linear and the Gaussian were tested. We empirically found that the different weighting functions end up giving similar performance. We used a simple linear function in our experiments. For the restricted propagation, we propagate the object mask until we reach a frame in which user annotations were given in any previous rounds. The restricted propagation improves not only the accuracy by preventing the drift, but also the runtime speed since it requires a smaller number of propagations. This testing scheme is depicted in Fig. 4.

Method	AUC	J@60
Ours	**0.641**	**0.647**
Najafi *et al.* [25]	0.549	0.395
Lin *et al.*	0.450	0.240
Huang *et al.*	0.328	0.335
Scribble-OSVOS [6]	0.299	0.153
Rakelly *et al.*	0.269	0.273

Table 1: The leaderboard of the interactive track in the DAVIS challenge 2018. The entries are ordered according to the AUC score. Scribble-OSVOS is a baseline method proposed by the challenge organizer [6].

4. Experiments

It is difficult to evaluate interactive video object segmentation methods quantitatively because the user input is directly related to the segmentation results, and vice versa. To tackle this problem with the evaluation, Caelles *et al.* [6] introduced a robot agent service that simulates human interaction according to the intermediate results of an algorithm. We used their method to quantitatively evaluate our method.

4.1. DAVIS Challenge

To fairly compare our method against the state-of-the-art methods, we evaluated our model on the interactive track benchmark in the DAVIS Challenge 2018 [6]. In the challenge, each method can interact with a robot agent up to 8 times and is expected to compute masks within 30 seconds per object for each interaction. The performance of each method is evaluated using two metrics: area under the curve (AUC) and Jaccard at 60 seconds (J@60s). AUC is designed to measure the overall accuracy of the evaluation. J@60 measures the accuracy with a limited time budget (60 seconds). We summarize the evaluation results in Table 1. In both metrics, our method outperforms competing methods by a large margin [27].

4.2. Qualitative Results

Fig. 5 shows examples of our results obtained after 5 interactions with the automatic evaluation robot in the DAVIS Challenge 2018. Our method generates accurate segmentation results for various object types with complex motions even if there are multiple object instances. In the supple-

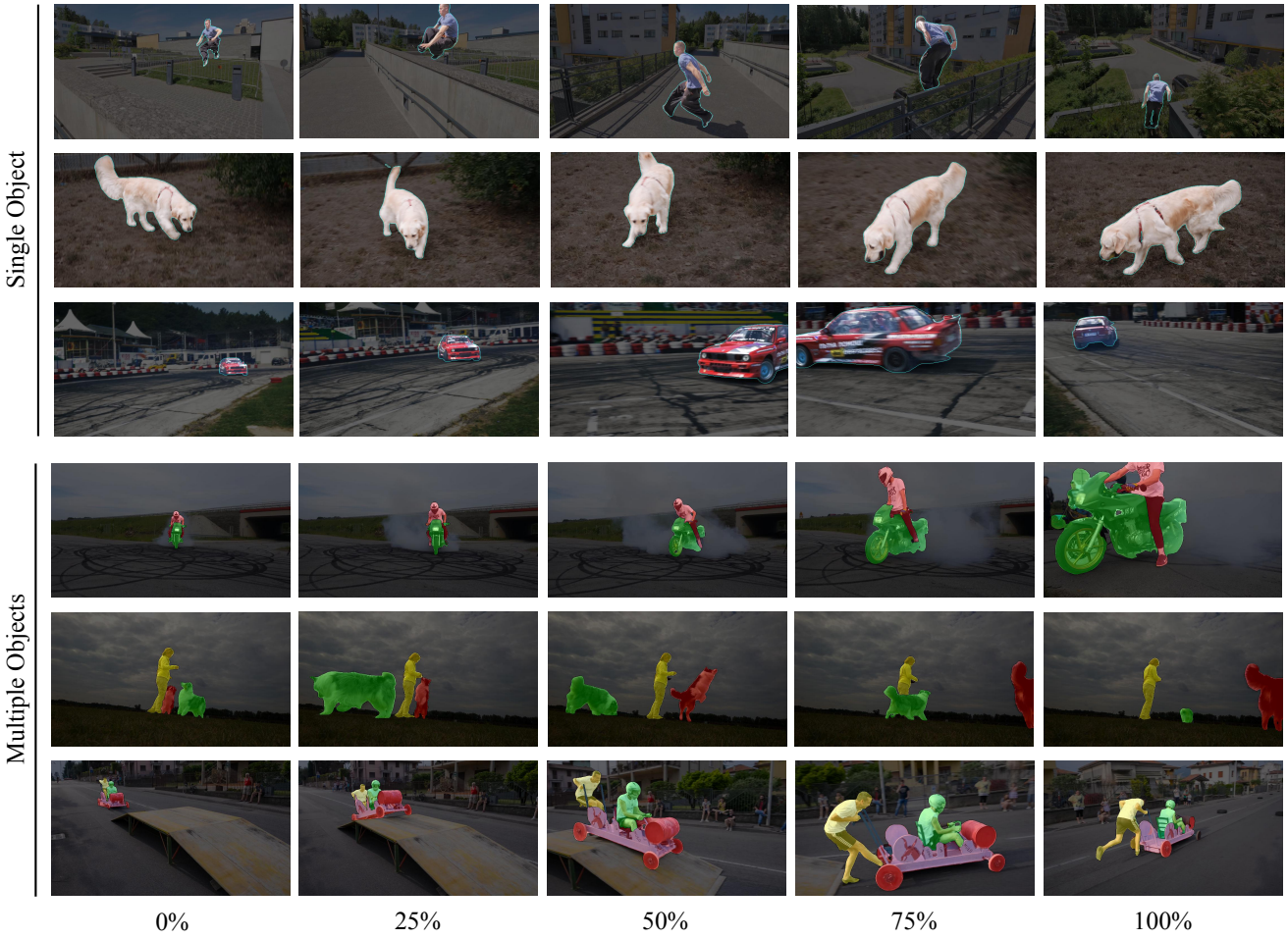

Figure 5: The qualitative results on the DAVIS-2017 validation set. All the user interactions are automatically simulated by the robot agent provided by [6]. The result masks are overlaid to uniformly sampled frames after 5 interactions (rounds).

mentary video, we present the recording of our real-time demo with real user interactions.

4.3. Ablation Studies

We conduct an ablation studies using the DAVIS-2017 validation set to validate the effectiveness of our feature aggregation module and training scheme. Specifically, we compare our complete model with three variant models. *No Reference* is a model without the feature aggregation module. In *No Aggregation* model, the feature aggregation module is replaced with a simple identity connection without feature aggregation. *No Multi-Round* is a model trained with the number of rounds as one (*i.e.* at each training iteration, there is only one interaction from the user).

The Jaccard score of ablation models with growing number of interactions is shown in Fig. 6. As shown in Fig. 6, the proposed multi-round training is crucial for achieving high accuracy and our feature aggregation module further improves the performance by allowing the networks to exploit the reference information from all previous user inputs.

Another ablation study was conducted on the use of the training data. Our complete model is first pre-trained on static image data and then fine-tuned using video data. To validate the effect of the pre-training, we compare variant models that are just trained on the video data without the pre-training. Also, to further inspect the effect of the amount of video training data, we evaluate variants that are fine-tuned with only 60 train videos of DAVIS-2017. Table 2 summarizes the results obtained by our variant models trained using different combinations of training datasets. Without pre-training, our performance drops significantly. The use of additional training video data further raises our performance.

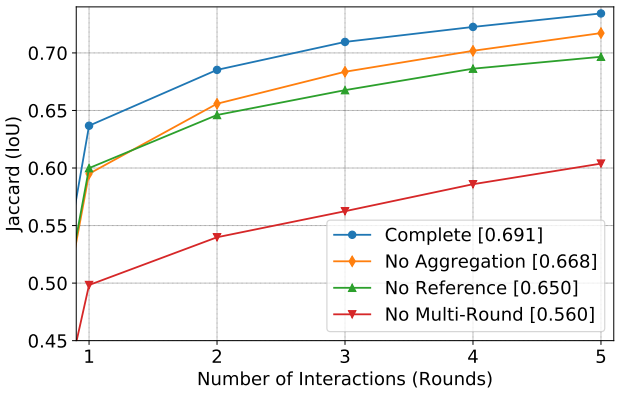

Figure 6: The result of our ablation study on the DAVIS-2017 validation set. We compare models with ablations from our complete model. The AUC of each variant is shown in the squared brackets of the legend.

PT	DV	GG+YV	AUC	J@60s
	✓		0.555	0.589
✓	✓		0.621	0.637
	✓	✓	0.627	0.657
✓	✓	✓	**0.691**	**0.734**

Table 2: We compare our models trained with different combination of training datasets. PT: pre-training on static images [34, 7, 8, 12, 22]. DV, GG and YV: the use of DAVIS [30], GyGo [10], and Youtube-VOS [40] for fine-tuning.

4.4. Failure Cases

While our method demonstrates satisfactory results on both the quantitative and the qualitative evaluations, we found few failure cases as shown in Fig. 7. We observed that rapid and complex object motions may lead our propagation network to drift by the error accumulating as shown in Fig. 7 (top). We believe that a good future direction is to augment the algorithm with a reliable temporal propagation of object masks.

Another limitation we found is that our method may be less stable on very challenging scenes in the current round-based scenario. Our method mostly improves results with additional user interactions, but this is not guaranteed as shown in Fig. 7 (bottom). Since we take only partial annotations from users at each round, the propagated masks from newer round are sometimes less accurate and there is no guarantee that we can always keep better results from different rounds. This is because there is no safety gear in the testing scenario and it can be resolved by asking the user for the confirmation of the mask being good to prevent updating the masks.

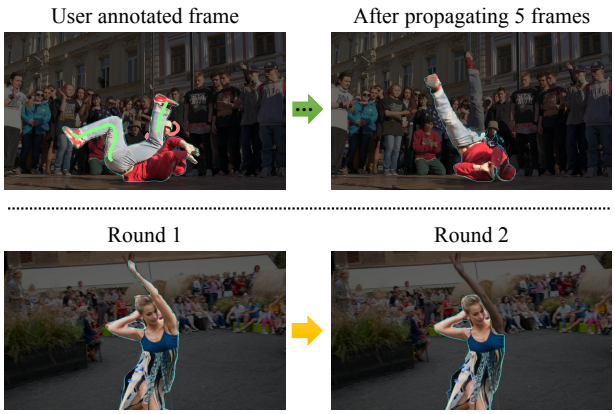

Figure 7: Failure cases. (top) Our propagation network may suffer from error accumulation due to fast and complex object motions. (bottom) We take only partial annotations from users and misunderstanding of the user intention may lead to unstable prediction with additional annotations.

5. Conclusion

While object segmentation in a video is one of the most basic tasks for video editing, it requires a lot of user effort and time with existing tools. To make it more accessible, we have presented a novel technique that generates object segmentation masks in video frames with minimum user inputs. Our method consists of interaction and propagation networks that share information with the feature aggregation module. We proposed the multi-round training scheme designed for interactive tasks and it plays a key role in achieving high accuracy. While our model is trained using synthetic user interactions, our method not only shows the best performance on the quantitative evaluation but also demonstrates good performance with real user interactions.

There are directions to further improve our system. The drifting during propagation is still a major challenge, although we greatly improved the performance with the aggregated reference features and the multi-round training. We believe that a better semantic understanding of the scene will help to resolve this problem by robustly linking the instances with appearance changes across video frames. Another important future work is supporting high-resolution videos. This is one of the common issues in many deep learning-based segmentation algorithms, and we hope that this can be addressed with a better network architecture or by combining our work with additional post-processing modules.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (2018-0-01858).

References

[1] A. Agarwala, A. Hertzmann, D. H. Salesin, and S. M. Seitz. Keyframe-based tracking for rotoscoping and animation. In *ACM Transactions on Graphics (ToG)*, volume 23, pages 584–591. ACM, 2004. 2

[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. 1, 2

[3] A. Benard and M. Gygli. Interactive video object segmentation in the wild. *arXiv preprint arXiv:1801.00269*, 2017. 2

[4] B. Bratt. *Rotoscoping: Techniques and tools for the Aspiring Artist*. Focal Press, 2012. 1, 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. 1, 2, 5

[6] S. Caelles, A. Montes, K.-K. Maninis, Y. Chen, L. Van Gool, F. Perazzi, and J. Pont-Tuset. The 2018 davis challenge on video object segmentation. *arXiv preprint arXiv:1803.00557*, 2018. 1, 2, 5, 6, 7

[7] 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. 5, 8

[8] 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. 5, 8

[9] 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

[10] I. Friedman, I. Chemla, E. Smolyansky, M. Stepanov, I. Afanasyeva, G. Sharir, S. Nadir, and S. Rorlich. Gygo: an e-commerce video object segmentation dataset by visualead. https://github.com/ilchemla/gygo-dataset, 2017. 5, 8

[11] Z. Guo and R. W. Hall. Parallel thinning with two-subiteration algorithms. *Commun. ACM*, 32(3):359–373, Mar. 1989. 5

[12] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *IEEE International Conference on Computer Vision (ICCV)*, pages 991–998. IEEE, 2011. 5, 8

[13] K. He, G. Gkioxari, P. Dollr, and R. Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 4, 5

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 4

[15] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, pages 630–645. Springer, 2016. 4

[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017. 3

[17] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 5

[18] S. D. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[19] V. Jampani, R. Gadde, and P. V. Gehler. Video propagation networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[20] 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. 1, 2

[21] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 4

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 5, 8

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 4

[24] 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. 2

[25] M. Najafi, V. Kulharia, T. Ajanthan, and P. H. S. Torr. Similarity learning for dense label transfer. *The 2018 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*, 2018. 6

[26] S. W. Oh, J.-Y. Lee, K. Sunkavalli, and S. J. Kim. Fast video object segmentation by reference-guided mask propagation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3, 4, 5

[27] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim. Fast user-guided video object segmentation by deep networks. *The 2018 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*, 2018. 2, 6

[28] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 4, 5

[29] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *European Conference on Computer Vision (ECCV)*, pages 75–91. Springer, 2016. 4

[30] 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. 5, 8

[31] B. L. Price, B. S. Morse, and S. Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *IEEE International Conference on Computer Vision (ICCV)*, pages 779–786. IEEE, 2009. 1, 2

[32] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 3

[33] N. Shankar Nagaraja, F. R. Schmidt, and T. Brox. Video segmentation with just a few strokes. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3235–3243, 2015. 2

[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. 5, 8

[35] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2

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

[37] 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. 1, 2

[38] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang. Deep grabcut for object selection. *arXiv preprint arXiv:1707.00243*, 2017. 2

[39] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang. Deep interactive object selection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 373–381, 2016. 2, 4

[40] N. Xu, L. Yang, D. Yue, J. Yang, B. Price, J. Yang, S. Cohen, Y. Fan, Y. Liang, and T. Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *European Conference on Computer Vision (ECCV)*, 2018. 5, 8

[41] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. Efficient video object segmentation via network modulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[42] 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 *IEEE International Conference on Computer Vision (ICCV)*, 2017. 3

[43] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (TOG)*, 36(4):119, 2017. 3

[44] F. Zhong, X. Qin, Q. Peng, and X. Meng. Discontinuity-aware video object cutout. *ACM Transactions on Graphics (TOG)*, 31(6):175, 2012. 2