

YouTube-VOS: Sequence-to-Sequence Video Object Segmentation

Ning Xu¹, Linjie Yang², Yuchen Fan³, Jianchao Yang², Dingcheng Yue³,
Yuchen Liang³, Brian Price¹, Scott Cohen¹, and Thomas Huang³

¹ Adobe Research, USA

{nxu,bprice,scohen}@adobe.com

² Snapchat Research, USA

{linjie.yang,jianchao.yang}@snap.com

³ University of Illinois at Urbana-Champaign, USA

{yuchenf4,dyue2,yliang35,t-huang1}@illinois.edu

Abstract. Learning long-term spatial-temporal features are critical for many video analysis tasks. However, existing video segmentation methods predominantly rely on static image segmentation techniques, and methods capturing temporal dependency for segmentation have to depend on pretrained optical flow models, leading to suboptimal solutions for the problem. End-to-end sequential learning to explore spatial-temporal features for video segmentation is largely limited by the scale of available video segmentation datasets, i.e., even the largest video segmentation dataset only contains 90 short video clips. To solve this problem, we build a new large-scale video object segmentation dataset called YouTube Video Object Segmentation dataset (YouTube-VOS). Our dataset contains 3,252 YouTube video clips and 78 categories including common objects and human activities⁴. This is by far the largest video object segmentation dataset to our knowledge and we have released it at <https://youtube-vos.org>. Based on this dataset, we propose a novel sequence-to-sequence network to fully exploit long-term spatial-temporal information in videos for segmentation. We demonstrate that our method is able to achieve the best results on our YouTube-VOS test set and comparable results on DAVIS 2016 compared to the current state-of-the-art methods. Experiments show that the large scale dataset is indeed a key factor to the success of our model.

Keywords: Video Object Segmentation, Large-scale Dataset, Spatial-Temporal Information.

1 Introduction

Learning effective spatial-temporal features has been demonstrated to be very important for many video analysis tasks. For example, Donahue *et al.* [10] propose long-term recurrent convolution network for activity recognition and video

⁴ This is the statistics when we submit this paper, see updated statistics on our website.

captioning. Srivastava *et al.* [38] propose unsupervised learning of video representation with a LSTM autoencoder. Tran *et al.* [42] develop a 3D convolutional network to extract spatial and temporal information jointly from a video. Other works include learning spatial-temporal information for precipitation prediction [46], physical interaction [14], and autonomous driving [47].

Video segmentation plays an important role in video understanding, which fosters many applications, such as accurate object segmentation and tracking, interactive video editing and augmented reality. Video object segmentation, which targets at segmenting a particular object instance throughout the entire video sequence given only the object mask on the first frame, has attracted much attention from the vision community recently [6,32,50,8,11,22,41,19,44]. However, existing state-of-the-art video object segmentation approaches primarily rely on single image segmentation frameworks [6,32,50,44]. For example, Caelles *et al.* [6] propose to train an object segmentation network on static images and then fine-tune the model on the first frame of a test video over hundreds of iterations, so that it remembers the object appearance. The fine-tuned model is then applied to all following individual frames to segment the object without using any temporal information. Even though simple, such an online learning or one-shot learning scheme achieves top performance on video object segmentation benchmarks [33,21]. Although some recent approaches [11,8,41] have been proposed to leverage temporal consistency, they depend on models pretrained on other tasks such as optical flow [20,35] or motion segmentation [40], to extract temporal information. These pretrained models are learned from separate tasks, and therefore are suboptimal for the video segmentation problem.

Learning long-term spatial-temporal features directly for video object segmentation task is, however, largely limited by the scale of existing video object segmentation datasets. For example, the popular benchmark dataset DAVIS [34] has only 90 short video clips, which is barely sufficient to learn an end-to-end model from scratch like other video analysis tasks. Even if we combine all the videos from available datasets [21,13,26,4,30,15], its scale is still far smaller than other video analysis datasets such as YouTube-8M [1] and ActivityNet [17]. To solve this problem, we present the first large-scale video object segmentation dataset called YouTube-VOS (YouTube Video Object Segmentation dataset) in this work. Our dataset contains 3,252 YouTube video clips featuring 78 categories covering common animals, vehicles, accessories and human activities. Each video clip is about 3~6 seconds long and often contains multiple objects, which are manually segmented by professional annotators. Compared to existing datasets, our dataset contains a lot more videos, object categories, object instances and annotations, and a much longer duration of total annotated videos. Table 1 provides quantitative scale comparisons of our new dataset against existing datasets. We retrain existing algorithms on YouTube-VOS and benchmark their performance on our test set which contains 322 videos. In addition, our test set contains 10 categories unseen in the training set and are used to evaluate the generalization ability of existing approaches.

Table 1: Scale comparison between YouTube-VOS and existing datasets. “Annotations” denotes the total number of object annotations. “Duration” denotes the total duration (in minutes) of the annotated videos.

Scale	JC [13]	ST [26]	YTO [21]	FBMS [30]	DAVIS [33] [34]		YouTube-VOS (Ours)
Videos	22	14	96	59	50	90	3,252
Categories	14	11	10	16	-	-	78
Objects	22	24	96	139	50	205	6,048
Annotations	6,331	1,475	1,692	1,465	3,440	13,543	133,886
Duration	3.52	0.59	9.01	7.70	2.88	5.17	217.21

Based on Youtube-VOS, we propose a new sequence-to-sequence learning algorithm to explore spatial-temporal modeling for video object segmentation. We utilize a convolutional LSTM [46] to learn long-term spatial-temporal information for segmentation. At each time step, the convolutional LSTM accepts last hidden states and an encoded image frame, it then outputs encoded spatial-temporal features which are decoded into a segmentation mask. Our algorithm is different from existing approaches in that it fully exploits the long-term spatial-temporal information in an end-to-end manner and does not depend on existing optical flow or motion segmentation models. We evaluate our algorithm on both YouTube-VOS and DAVIS 2016 and it achieves better or comparable results compared to the current state of the arts.

The rest of our paper is organized as follows. In Section 2 we briefly introduce the related works. In Section 3 and 4 we describe our YouTube-VOS dataset and the proposed algorithm in detail. Experimental results are presented in Section 5. Finally we conclude the paper in Section 6.

2 Related work

In the past decades, several datasets [21,13,26,4,30,15] have been created for video object segmentation. All of them are in small scales which usually contain only dozens of videos. In addition, their video content is relatively simple (*e.g.* no heavy occlusion, camera motion or illumination change) and sometimes the video resolution is low. Recently, a new dataset called DAVIS [33,34] was published and has become the benchmark dataset in this area. Its 2016 version contains 50 videos with a single foreground object per video while the 2017 version has 90 videos with multiple objects per video. In comparison to previous datasets [21,13,26,4,30,15], DAVIS has both higher-quality of video resolutions and annotations. In addition, their video content is more complicated with multi-object interactions, camera motion, and occlusions.

Early methods [21,28,12,31,5] for video object segmentation often solve some spatial-temporal graph structures with hand-crafted energy terms, which are usually associated with features including appearance, boundary, motion and optical flows. Recently, deep-learning based methods were proposed due to its

great success in image segmentation tasks [36,7,49,48]. Most of these methods [6,32,8,11,50,44] build their model based on an image segmentation network and do not involve sequential modeling. Online learning [6] is commonly used to improve their performance. To make the model temporally consistent, the predicted mask of the previous frame is used as a guidance in [32,50,19]. Other methods have been proposed to leverage spatial-temporal information. Jampani *et al.* [22] use spatial-temporal consistency to propagate object masks over time. Tokmakov *et al.* [41] use a two-stream network to model objects' appearance and motion and use a recurrent layer to capture the evolution. However, due to the lack of training videos, they use a pretrained motion segmentation model [40] and optical-flow model [20], which leads to suboptimal results since the model is not trained end-to-end to best capture spatial-temporal features.

3 YouTube-VOS

To create our dataset, we first carefully select a set of object categories including animals (*e.g. ant, eagle, goldfish, person*), vehicles (*e.g. airplane, bicycle, boat, sedan*), accessories (*e.g. eyeglass, hat, bag*), common objects (*e.g. potted plant, knife, sign, umbrella*), and humans in various activities (*e.g. tennis, skateboarding, motorcycling, surfing*). The videos containing human activities have diversified appearance and motion, so instead of treating human videos as one class, we divide different activities into different categories. Most of these videos contain interactions between a person and a corresponding object, such as tennis racket, skateboard, motorcycle, etc. The entire category set includes 78 categories that covers diverse objects and motions, and should be representative for everyday scenarios.

We then collect many high-resolution videos with the selected category labels from the large-scale video classification dataset YouTube-8M [1]. This dataset consists of millions of YouTube videos associated with more than 4,700 visual entities. We utilize its category annotations to retrieve candidate videos that we are interested in. Specifically, up to 100 videos are retrieved for each category in our segmentation category set. There are several advantages to using YouTube videos to create our segmentation dataset. First, YouTube videos have very diverse object appearances and motions. Challenging cases for video object segmentation, such as occlusions, fast object motions and change of appearances, commonly exist in YouTube videos. Second, YouTube videos are taken by both professionals and amateurs and thus different levels of camera motions are shown in the crawled videos. Algorithms trained on such data could potentially handle camera motion better and thus are more practical. Last but not the least, many YouTube videos are taken by today's smart phone devices and there are demanding needs to segment objects in those videos for applications such as video editing and augmented reality.

Since the retrieved videos are usually long (several minutes) and have shot transitions, we use an off-the-shelf video shot detection algorithm ⁵ to automat-

⁵ <http://johmathe.name/shotdetect.html>

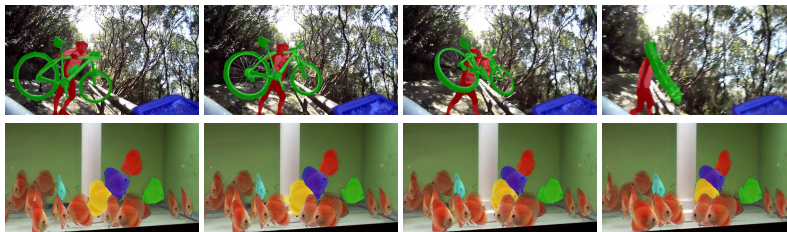


Fig. 1: The ground truth annotations of sample video clips in our dataset. Different objects are highlighted with different colors.

ically partition each video into multiple video clips. We first remove the clips from the first and last 10% of the video, since these clips have a high chance of containing introductory subtitles and credits lists. We then sample up to five clips with appropriate lengths (3~6 seconds) per video and manually verify that these clips contain the correct object categories and are useful for our task (*e.g.* no scene transition, not too dark, shaky, or blurry). After the video clips are collected, we ask human annotators to select up to five objects of proper sizes and categories per video clip and carefully annotate them (by tracing their boundaries instead of rough polygons) every five frames in a 30fps frame rate, which results in a 6fps sampling rate. Given a video and its category, annotators are first required to annotate objects belonging to that category. If the video contains other objects that belong to our 78 categories, we ask the annotators to label them as well, so that each video has multiple objects annotated. In human activity videos, both the human subject and the object he/she interacts with are labeled, *e.g.*, both the person and the skateboard are required to be labeled in a “skateboarding” video. Some annotation examples are shown in Figure 1. Unlike dense per-frame annotation in previous datasets [13,33,34], we believe that the temporal correlation between five consecutive frames is sufficiently strong that annotations can be omitted for intermediate frames to reduce the annotation efforts. Such a skip-frame annotation strategy allows us to scale up the number of videos and objects under the same annotation budget, which are important factors for better performance. We find empirically that our dataset is effective in training different segmentation algorithm.

As a result, our dataset YouTube-VOS consists of 3,252 YouTube video clips and 133,886 object annotations, 33 and 10 times more than the best of the existing video object segmentation datasets, respectively (See Table 1). YouTube-VOS is the largest dataset for video object segmentation to date.

4 Sequence-to-Sequence Video Object Segmentation

Based on our new dataset, we propose a new sequence-to-sequence video object segmentation algorithm. Different from existing approaches, our algorithm learns long-term spatial-temporal features directly from training data in an end-to-end

manner, and the offline trained model is capable of propagating an initial object segmentation mask accurately by memorizing and updating the object characteristics, including appearance, location and scale, and temporal movements, automatically over the entire video sequence.

4.1 Problem formulation

Let us denote a video sequence with T frames as $\{\mathbf{x}_t | t \in [0, T - 1]\}$ where $\mathbf{x}_t \in \mathbb{R}^{H \times W \times 3}$ is the RGB frame at time step t , and denote an initial binary object mask at time step 0 as $\mathbf{y}_0 \in \mathbb{R}^{H \times W}$. The target of video object segmentation is to predict the object mask automatically for the remaining frames from time step 1 to $T - 1$, *i.e.* $\{\hat{\mathbf{y}}_t | t \in [1, T - 1]\}$.

To obtain a predicted mask $\hat{\mathbf{y}}_t$ for \mathbf{x}_t , many existing deep learning methods only leverage information at time step 0 (*e.g.* online learning or one-shot learning [6]) or time step $t - 1$ (*e.g.* optical flow [32]) while the long-term history information is totally dismissed. Their frameworks can be formulated as $\hat{\mathbf{y}}_t = \arg \max_{\bar{\mathbf{y}}_t} \mathbb{P}(\bar{\mathbf{y}}_t | \mathbf{x}_0, \mathbf{y}_0, \mathbf{x}_t)$ or $\hat{\mathbf{y}}_t = \arg \max_{\bar{\mathbf{y}}_t} \mathbb{P}(\bar{\mathbf{y}}_t | \mathbf{x}_0, \mathbf{y}_0, \mathbf{x}_t, \mathbf{x}_{t-1})$. They are effective when the object appearance is similar between time 0 and time t or when the object motion from time $t - 1$ to t can be accurately measured. However, these assumptions will be violated when the object has drastic appearance variation and rapid motion, which is often case in many real-world videos. In such cases, the history information of the object in all previous frames becomes critical and should be leveraged in an effective way. Therefore, we propose to solve a different objective function, *i.e.* $\hat{\mathbf{y}}_t = \arg \max_{\bar{\mathbf{y}}_t} \mathbb{P}(\bar{\mathbf{y}}_t | \mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_t, \mathbf{y}_0)$, which can be transformed into a sequence-to-sequence learning problem.

4.2 Our Algorithm

Recurrent Neural Networks (RNN) has been adopted by many sequence-to-sequence learning problems because it is capable to learn long-term dependency from sequential data. LSTM [18] as a special RNN structure solves vanishing or exploding gradients issue [3]. A convolutional variant of LSTM (convolutional LSTM) [46] is later proposed to preserve the spatial information of the data in the hidden states of the model.

Our algorithm is inspired by the convolutional encoder-decoder LSTM structure [9,39] which has achieved much success in machine translation, where an input sentence in language A is first encoded by a encoder LSTM and its outputs are fed into a decoder LSTM which can generate the desired output sentence in language B. In video object segmentation, it is essential to capture the object characteristics over time. To generate the initial states for our convolutional LSTM (*ConvLSTM*), we use a feed-forward neural network to encode both the first image frame and the segmentation mask. Specifically, we concatenate the initial frame \mathbf{x}_0 and segmentation mask \mathbf{y}_0 and feed it into a trainable network, denoted as *Initializer*, which outputs the initial memory state \mathbf{c}_0 and hidden state \mathbf{h}_0 . These initial states capture object appearance, object location and scale. And they are feed into *ConvLSTM* for sequence learning.

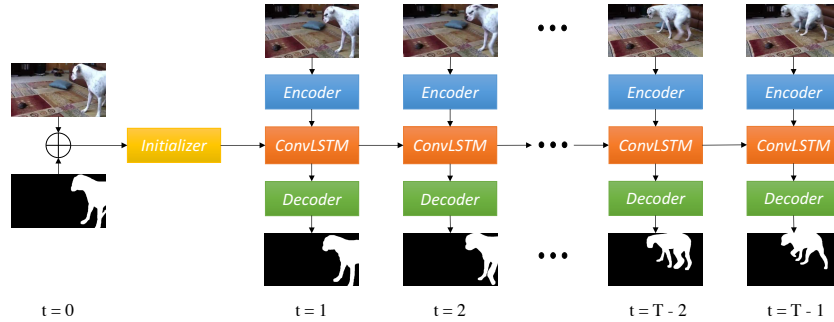


Fig. 2: The framework of our algorithm. The initial information at time 0 is encoded by *Initializer* to initialize *ConvLSTM*. The new frame at each time step is processed by *Encoder* and the segmentation result is decoded by *Decoder*. *ConvLSTM* is automatically updated over the entire video sequence.

At time step t , frame \mathbf{x}_t is first processed by a convolutional encoder, denoted as *Encoder*, to extract feature maps $\tilde{\mathbf{x}}_t$. Then $\tilde{\mathbf{x}}_t$ is sent as the inputs of *ConvLSTM*. The internal states \mathbf{c}_t and \mathbf{h}_t are automatically updated given the new observation $\tilde{\mathbf{x}}_t$, which capture the new characteristics of the object. The output \mathbf{h}_t is passed into a convolutional decoder, denoted as *Decoder*, to get the full-resolution segmentation results $\hat{\mathbf{y}}_t$. Binary cross-entropy loss is computed between $\hat{\mathbf{y}}_t$ and \mathbf{y}_t during training process. The entire model is trained end-to-end using back-propagation to learn parameters for the *Initializer* network, the *Encoder* and *Decoder* networks, and *ConvLSTM* network. Figure 2 illustrates our sequence learning algorithm for video object segmentation. The learning process can be formulated as follows:

$$\mathbf{c}_0, \mathbf{h}_0 = \text{Initializer}(\mathbf{x}_0, \mathbf{y}_0) \quad (1)$$

$$\tilde{\mathbf{x}}_t = \text{Encoder}(\mathbf{x}_t) \quad (2)$$

$$\mathbf{c}_t, \mathbf{h}_t = \text{ConvLSTM}(\tilde{\mathbf{x}}_t, \mathbf{c}_{t-1}, \mathbf{h}_{t-1}), \quad (3)$$

$$\hat{\mathbf{y}}_t = \text{Decoder}(\mathbf{h}_t) \quad (4)$$

$$\mathcal{L} = -(\mathbf{y}_t \log(\hat{\mathbf{y}}_t)) + ((1 - \mathbf{y}_t) \log(1 - \hat{\mathbf{y}}_t)) \quad (5)$$

4.3 Implementation Details

Model structures Both our *Initializer* and *Encoder* use VGG-16 [37] network structures. In particular, all the convolution layers and the first fully connected layer of VGG-16 are used as backbone for the two networks. The fully connected layer is transformed to a 1×1 convolution layer to make our model fully convolutional. On top of it, *Initializer* has two additional convolution layers with ReLU [29] activation to produce \mathbf{c}_0 and \mathbf{h}_0 respectively. Each convolution layer has 512 1×1 filters. The *Encoder* has one additional convolution layer with ReLU activation which has 512 1×1 filters. The VGG-16 layers of the *Initializer*

and *Encoder* are initialized with pre-trained VGG-16 parameters while the other layers are randomly initialized by Xavier [16].

All the convolution operations of the *ConvLSTM* layer use 512 3×3 filters, initialized by Xavier. Sigmoid activation is used for gate outputs and ReLU is used for state outputs (empirically we find ReLU activation produces better results than tanh activation for our model). Following [23], we set the bias of the forget gate to be 1s at initialization.

The *Decoder* has five upsampling layers with 5×5 kernel size and 512, 256, 128, 64 and 64 filters respectively. The last layer of the *Decoder* produces segmentation results, which has one 5×5 filter with sigmoid activation. All the parameters are initialized by Xavier.

Training Our algorithm is trained on the YouTube-VOS training set. At each training iteration, our algorithm first randomly samples an object and T ($5 \sim 11$) frames from a random training video sequence. Then the original RGB frames and annotations are resized to 256×448 for memory and speed concern. At the early stage of training, we only select frames with ground truth annotation as our training samples so that the training loss can be computed and back-propagated at each time step. When the training losses become stable, we added frames without annotations to training data. For those frames without ground truth annotations, loss is set to be 0. Adam [24] is used to train our network and the initial learning rate is set to 10^{-5} , and our model converges in 80 epochs.

Inference Our offline-trained model is able to learn features for general object characteristics effectively. It is able to produce good segmentation results by directly applying it to a new test video with unseen categories. This is in contrast to recent state-of-the-art approaches, which have to fine-tune their models on each new test video over hundreds of iterations. In our experiments, we show that our algorithm without online learning can achieve comparable or better results compared to previous state of the arts with online learning, which implies much faster inference speed for practical applications. Nevertheless, we find that the performance of our model can be further improved with online learning.

Online Learning Given a test video, we generate random pairs of online training examples $\{(\mathbf{x}^0, \mathbf{y}^0), (\mathbf{x}^1, \mathbf{y}^1)\}$ through affine transformations from $(\mathbf{x}_0, \mathbf{y}_0)$. We treat $(\mathbf{x}^0, \mathbf{y}^0)$ as the initial frame and mask and $(\mathbf{x}^1, \mathbf{y}^1)$ as the first frame and ground truth mask. We then fine tune our *Initializer*, *Encoder* and *Decoder* networks on such randomly generated pairs. The parameters of *ConvLSTM* are fixed as it models long-term spatial-temporal dependency that should be independent of object categories.

5 Experiments

In this section, we first evaluate our algorithm and recent state-of-the-art algorithms on our YouTube-VOS dataset. Then we compare our results on the

DAVIS 2016 validation dataset [33], which is an existing benchmark dataset for video object segmentation. Finally, we do an ablation study to explore the effect of data scale and model variants to our method.

5.1 Experiment Settings

We split the YouTube-VOS dataset of 3,252 videos into training (2,796), validation (134) and test (322) sets. To evaluate the generalization ability of existing approaches on unseen categories, the test set is further split into test-seen and test-unseen subsets. We first select 10 categories (*i.e.* *ant*, *bull riding*, *butterfly*, *chameleon*, *flag*, *jellyfish*, *kangaroo*, *penguin*, *slopestyle*, *snail*) as unseen categories during training and treat their videos as test-unseen set. The validation and test-seen subsets are created by sampling two and four videos per category, respectively. The rest of videos are the training set. We use the region similarity \mathcal{J} and the contour accuracy \mathcal{F} as the evaluation metrics as in [33].

5.2 YouTube-VOS

For fair comparison, we re-train previous methods (*i.e.* SegFlow [8], OSMN [50], MaskTrack [32], OSVOS[6] and OnAVOS [44]) on our training set with the same settings as our algorithm. One difference is that other methods leverage post-processing steps to achieve additional gains while our models do not.

The results are presented in Table 2. All the comparison methods use static image segmentation models and four of them (*i.e.* SegFlow, MaskTrack, OSVOS and OnAVOS) require online learning. Our algorithm leverages long-term spatial-temporal characteristics and achieves better performance even without online learning (the second last row in Table 2), which effectively demonstrates the importance of long-term spatial-temporal information for video object segmentation. With online learning, our model is further improved and achieves around 8% absolute improvement over the best previous method OSVOS on \mathcal{J} mean. Our method also outperforms previous methods on contour accuracy and decay rate with a large margin. Surprisingly, OnAVOS which is the best performing method on DAVIS does not achieve good results on our dataset. We believe the drastic appearance changes and complex motion patterns in our dataset makes the online adaptation fail in many cases. Figure 3 visualizes the changes of \mathcal{J} mean over the duration of video sequences. Without online learning, our method is worse than online learning methods such as OSVOS at the first few frames since the object appearance usually has not changed too much from the initial frame and online learning is effective under such scenario. However, our method degrades slower than the other methods and starts to outperform OSVOS at around 25% of the videos, which demonstrates that our method indeed propagates object segmentations more accurately over time than previous methods. With the help of online learning, our method outperforms previous methods in most parts of the video sequences, while maintaining a small decay rate.

Next we compare the generalization ability of existing methods on unseen categories in Table 2. Most methods have better performance on seen categories

Table 2: Comparisons of our approach and other methods on YouTube-VOS test set. The results in each cell show the test results for seen/unseen categories. “OL” denotes online learning. The best results are highlighted in bold.

Method	\mathcal{J} mean \uparrow	\mathcal{J} recall \uparrow	\mathcal{J} decay \downarrow	\mathcal{F} mean \uparrow	\mathcal{F} recall \uparrow	\mathcal{F} decay \downarrow
SegFlow [8]	40.4/38.5	45.4/41.7	7.2/8.4	35.0/32.7	35.3/32.1	6.9/9.1
OSVOS [6]	59.1/58.8	66.2/64.5	17.9/19.5	63.7/63.9	69.0/67.9	20.6/23.0
MaskTrack [32]	56.9/60.7	64.4/69.6	13.4/16.4	59.3/63.7	66.4/73.4	16.8/19.8
OSMN [50]	54.9/52.9	59.7/57.6	10.2/14.6	57.3/55.2	60.8/58.0	10.4/13.8
OnAVOS [44]	55.7/56.8	61.6/61.5	10.3/9.4	61.3/62.3	66.0/67.3	13.1/12.8
Ours (w/o OL)	60.9/60.1	70.3/71.2	7.9/12.9	64.2/62.3	73.0/71.4	9.3/14.5
Ours (with OL)	66.9/66.8	78.7/76.5	10.2/9.5	74.1/72.3	82.8/80.5	12.6/13.4

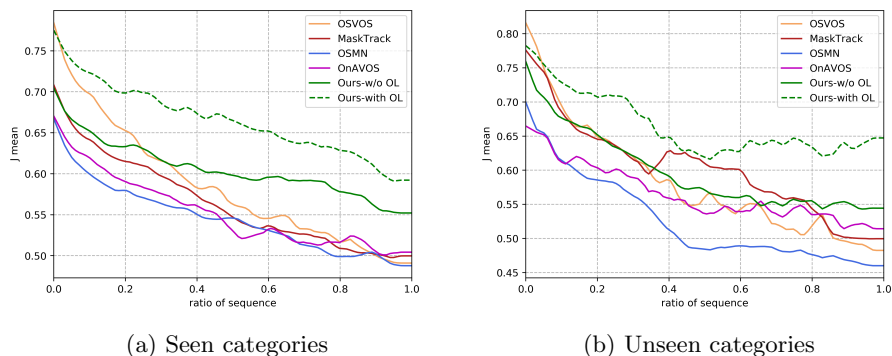


Fig. 3: The changes of \mathcal{J} mean values over the length of video sequences.

than unseen categories, which is expected. But the differences are not obvious, *e.g.* usually within 2% absolute differences on each metric. On one hand, it suggests that existing methods are able to alleviate the mismatch issue between training and test categories by approaches such as online learning. On the other hand, it also demonstrates the diverse training categories in YouTube-VOS helps different methods to generalize to new categories. Experiments on dataset scale in Section 5.4 further suggests the power of data scale on our model. Compared to other single-frame based methods, OSMN has a more obvious degradation on unseen categories since it does not use online learning. Our method without online learning does not have the issue since it leverages spatial-temporal information which is more robust to unseen categories. MaskTrack and OnAVOS have better performance on unseen than seen categories. We believe that they benefit from the guidance of previous segmentation or online adaption, which have advantages to deal with videos with slow motion. There are indeed several objects with slow motion in the unseen categories such as *snail* and *chameleon*.

Some test results produced by our model without online learning are visualized in Figure 4. The first two rows are from seen categories while the last

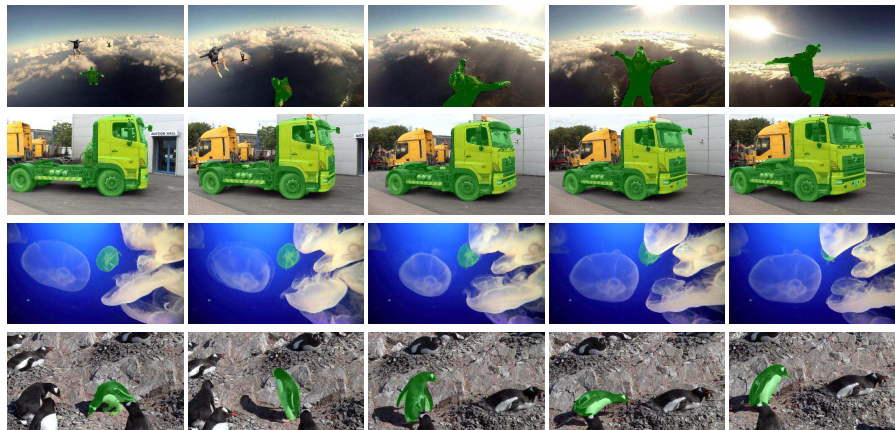


Fig. 4: Some visual results produced by our model without online learning on the YouTube-VOS test set. The first column shows the initial ground truth object segmentation (green color) while the second to the last column are predictions.

two rows are from unseen categories. In addition, each example represents some challenging cases in video object segmentation. For example, the person in the first example has large changes in appearance and illumination. The second and third examples both have multiple similar objects and heavy occlusions. The last example has strong camera motion and the penguin changes its pose frequently. Our model obtains accurate results on all the examples, which demonstrates the effectiveness of spatial-temporal features learned from large-scale training data.

5.3 DAVIS 2016

DAVIS 2016 is a popular prior benchmark dataset for video object segmentation. To evaluate our algorithm, we first fine-tune our pretrained model in 200 epochs on the DAVIS training set which contains 30 videos. The comparison results between our fine-tuned models and previous methods are shown in Table 3.

BVS and OFL are based on hand-crafted features and graphical models, while the rest are all deep learning based methods. Among the methods [32,6,44,50] using image segmentation frameworks, OnAVOS achieves the best performance. However, its online adaption process makes the inference pretty slow (~ 13 s per frame). Our model without online learning (the second last row) achieves comparable results to other online learning methods without post-processing (*e.g.* MaskTrack 69.8% and OSVOS 77.4%), but with a significant speed-up (60 times faster). Previous methods using spatial-temporal information including SegFlow, VPN and ConvGRU get inferior results compared to ours. Among them, ConvGRU is most related to ours since it also incorporates RNN memory cells in its model. However, it is an unsupervised methods to only segment moving foreground, while our method can segment arbitrary objects given the mask super-

Table 3: Comparisons of our approach and previous methods on the DAVIS 2016 dataset. Different components used in each algorithm are marked. “OL” denotes online learning. “PP” denotes post processing by CRF [25] or Boundary Snapping [6]. “OF” denotes optical flows. “RNN” denotes RNN and its variants.

Method	OL	PP	OF	RNN	mean IoU(%)	Speed(s)
BVS [43]	-	✗	✗	-	60.0	0.37
OFL [27]	-	✓	✓	-	68.0	42.2
SegFlow [8]	✓	✓	✓	✗	76.1	7.9
MaskTrack [32]	✓	✓	✗	✗	79.7	12
OSVOS [6]	✓	✓	✗	✗	79.8	10
OnAVOS [44]	✓	✓	✗	✗	85.7	13
OSMN [50]	✗	✗	✗	✗	74.0	0.14
VPN [22]	✗	✗	✗	✗	70.2	0.63
ConvGRU [41]	✗	✓	✓	✓	75.9	20
Ours	✗	✗	✗	✓	76.5	0.16
Ours	✓	✗	✗	✓	79.1	9

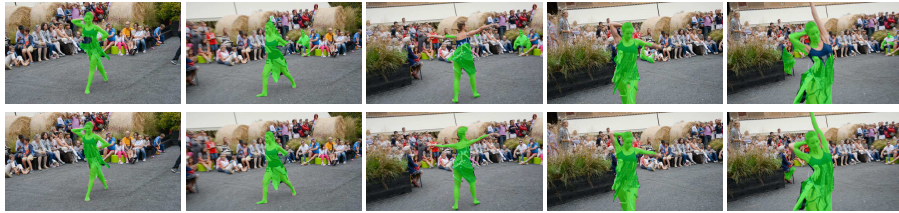


Fig. 5: The comparison results between our model without online learning (upper row) and with online learning (bottom row). Each column shows predictions of the two models at the same frame.

vision. Finally, online learning helps our model segment object boundary more accurately. Figure 5 shows such an example.

To demonstrate the scale limitation of existing datasets, we train our models on three different settings and evaluate on DAVIS 2016.

- Setting 1: We train our model from scratch on the 30 training videos.
- Setting 2: We train our model from scratch on the 30 training videos, plus all the videos from the SegTrackv2, JumpCut and YoutubeObjects datasets, which results in a total of 192 training videos.
- Setting 3: Following the idea of ConvGRU, we use a pretrained object segmentation model DeepLab [7] as our *Encoder* and train the other parts of our model on the 30 training videos.

Our models trained on setting 1 and 2 only get 51.3% and 51.9% mean IoU, which suggests that existing video object segmentation datasets do not have sufficient data to train our models. Therefore our YouTube-VOS dataset is one of the key elements for the success of our algorithm. In addition, there is only little

Table 4: The effect of data scale on our algorithm. We use different portions of training data to train our models and evaluate on the YouTube-VOS test set.

Scale	\mathcal{J} mean \uparrow	\mathcal{J} recall \uparrow	\mathcal{J} decay \downarrow	\mathcal{F} mean \uparrow	\mathcal{F} recall \uparrow	\mathcal{F} decay \downarrow
25%	46.7/40.1	53.5/45.6	8.3/13.6	46.7/40.0	52.2/41.6	8.5/13.2
50%	51.5/50.3	59.2/58.8	10.3/13.1	51.8/50.2	59.5/55.8	11.1/13.3
75%	56.8/56.0	65.7/67.1	7.6/10.0	59.6/56.3	68.8/64.1	8.5/11.1
100%	60.9/60.1	70.3/71.2	7.9/12.9	64.2/62.3	73.0/71.4	9.3/14.5

improvement by adding videos from the SegTrackv2, JumpCut and YoutubeObjects datasets, which suggests that the small scale is not the only problem for previous datasets. For example, videos in the three datasets usually only have one main foreground. SegTrackv2 has low-resolution videos. The annotation of YoutubeObjects videos is not accurate along object boundaries, *etc.* However, our YouTube-VOS dataset is carefully created to avoid all these problems. Setting 3 is a common detour for existing methods to bypass the data-insufficiency issue, *i.e.* using pre-trained models on other large-scale datasets to reduce the parameters to be learned for their models. However, our model using this strategy gets even worse results (45.6%) than training from scratch, which suggests that spatial-temporal features cannot be trivially transferred from representations learned from static images. Thus large scale training data such as our dataset is essential to learn spatial-temporal representation for video object segmentation.

5.4 Ablation study

In this subsection, we perform an ablation study on the YouTube-VOS dataset to evaluate different variants of our algorithm.

Dataset scale. Since the dataset scale is very important to our models, we train several models on different portions of the training set of YouTube-VOS to explore the effect of data. Specifically, we randomly select 25%, 50% and 75% of the training set and retrain our models from scratch. The results are listed in Table 4. It can be seen that using only 25% of the training videos (~ 700 videos) drops the performance almost 30% from the original model. In addition, the performance of the model on unseen categories are much worse than its performance on seen categories, which suggests that the model fails to capture general features for objectness. Since the scale of adding all the videos from all existing datasets is still much less than 700 videos, there is no doubt that existing datasets cannot satisfy the needs of our algorithm. With more and more training videos, our algorithm is improved rapidly, which well demonstrates the importance of large-scale data on our algorithm. We can see the trend of accuracies for 100% data still have not reached a plateau. We are working on collecting more data to explore the impact of data on the algorithm further.

Initializer variants. The *Initializer* in our original model is a VGG-16 network which encodes a RGB frame and an object mask and outputs initial hidden

states of *ConvLSTM*. We would like to explore using the object mask directly as the initial hidden states of *ConvLSTM*. We train an alternative model by removing the *Initializer* and directly using the object mask as the hidden states, *i.e.* the object mask is reshaped to match the size of the hidden states. The \mathcal{J} mean of the adapted model are 45.1% on the seen categories and 38.6% on the unseen categories. This suggests that the object mask alone does not have enough information for localizing the object.

Encoder variants. The *Encoder* in our original model receives a RGB frame as input at each time step. Alternatively, we can use the segmentation mask of the previous step as additional inputs to explicitly provide extra information to the model, similar to MaskTrack [32]. In this way, our *Initializer* and *Encoder* can be replaced with a single VGG-16 network since the inputs at every time step have same dimensions and similar meaning. However, such a framework potentially has the error-drifting issue since segmentation mistakes made at previous steps will be propagated to the current step.

In the early stage of training, the model is unable to predict good segmentation results. Therefore we use the ground truth annotation of the previous step as the input. Such strategy is known as teacher forcing [45] which can make the training faster. After the training losses become stable, we replace the ground truth annotation with the model’s prediction of the previous step so that the model is forced to correct its own mistakes. Such a strategy is known as curriculum learning [2]. Empirically we find that both the two strategies are important to make the model to work well. The \mathcal{J} mean results of the model are 59.4% on the seen categories and 60.7% on the unseen categories, which is similar to our original model.

6 Conclusion

In this work, we introduce the largest video object segmentation dataset (YouTube-VOS) to date. The new dataset, much larger than existing datasets in terms of number of videos and annotations, allows us to design a new deep learning algorithm to explicitly model long-term spatial-temporal dependency from videos for segmentation in an end-to-end learning framework. Thanks to the large scale dataset, our new algorithm achieves better or comparable results compared to existing state-of-the-art approaches. We believe the new dataset will foster research on video-based computer vision in general.

7 Acknowledgement

This research was partially supported by a gift funding from Snap Inc. and UIUC Andrew T. Yang Research and Entrepreneurship Award to Beckman Institute for Advanced Science & Technology, UIUC.

References

1. Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: Youtube-8m: A large-scale video classification benchmark. arXiv preprint arXiv:1609.08675 (2016)
2. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled sampling for sequence prediction with recurrent neural networks. In: *Advances in Neural Information Processing Systems*. pp. 1171–1179 (2015)
3. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* **5**(2), 157–166 (1994)
4. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: *ECCV*. pp. 282–295 (2010)
5. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: *European conference on computer vision*. pp. 282–295. Springer (2010)
6. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: *CVPR* (2017)
7. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. In: *IEEE T-PAMI*. vol. 40, pp. 834–848 (2018)
8. Cheng, J., Tsai, Y.H., Wang, S., Yang, M.H.: Segflow: Joint learning for video object segmentation and optical flow. In: *IEEE International Conference on Computer Vision (ICCV)* (2017)
9. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
10. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: *CVPR* (2015)
11. Dutt Jain, S., Xiong, B., Grauman, K.: Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017)
12. Faktor, A., Irani, M.: Video segmentation by non-local consensus voting. In: *BMVC* (2014)
13. Fan, Q., Zhong, F., Lischinski, D., Cohen-Or, D., Chen, B.: Jumpcut:non-successive mask transfer and interpolation for video cutout. In: *ACM Trans. Graph.*, 34(6) (2015)
14. Finn, C., Goodfellow, I., Levine, S.: Unsupervised learning for physical interaction through video prediction. In: *Advances in neural information processing systems* (2016)
15. Galasso, F., Nagaraja, N.S., Cárdenas, T.J., Brox, T., Schiele, B.: A unified video segmentation benchmark: Annotation, metrics and analysis. In: *ICCV. IEEE* (2013)
16. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. pp. 249–256 (2010)
17. Heilbron, F.C., Escorcia, V., Ghanem, B., Niebles, J.C.: Activitynet: A large-scale video benchmark for human activity understanding. In: *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. pp. 961–970. IEEE (2015)

18. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural computation* **9**(8), 1735–1780 (1997)
19. Hu, Y.T., Huang, J.B., Schwing, A.: Maskrnn: Instance level video object segmentation. In: *NIPS* (2017)
20. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. *CVPR* (2017)
21. Jain, S.D., Grauman, K.: Supervoxel-consistent foreground propagation in video. In: *ECCV* (2014)
22. Jampani, V., Gadde, R., Gehler, P.V.: Video propagation networks. In: *CVPR* (2017)
23. Jozefowicz, R., Zaremba, W., Sutskever, I.: An empirical exploration of recurrent network architectures. In: *International Conference on Machine Learning*. pp. 2342–2350 (2015)
24. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)
25. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. In: *NIPS*. pp. 109–117 (2011)
26. Li, F., Kim, T., Humayun, A., Tsai, D., Rehg, J.M.: Video segmentation by tracking many figure-ground segments. In: *ICCV* (2013)
27. Märki, N., Perazzi, F., Wang, O., Sorkine-Hornung, A.: Bilateral space video segmentation. In: *CVPR* (2016)
28. Nagaraja, N.S., Schmidt, F.R., Brox, T.: Video segmentation with just a few strokes. In: *ICCV*. pp. 3235–3243 (2015)
29. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. pp. 807–814 (2010)
30. Ochs, P., Malik, J., Brox, T.: Segmentation of moving objects by long term video analysis. *IEEE transactions on pattern analysis and machine intelligence* **36**(6), 1187–1200 (2014)
31. Papazoglou, A., Ferrari, V.: Fast object segmentation in unconstrained video. In: *Computer Vision (ICCV), 2013 IEEE International Conference on*. pp. 1777–1784. *IEEE* (2013)
32. Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., A.Sorkine-Hornung: Learning video object segmentation from static images. In: *CVPR* (2017)
33. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: *CVPR* (2016)
34. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. *arXiv:1704.00675* (2017)
35. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: Epicflow: Edge-preserving interpolation of correspondences for optical flow. In: *CVPR* (2015)
36. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence* **39**(4), 640–651 (2017)
37. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014)
38. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using lstms. In: *International conference on machine learning* (2015)

39. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: *Advances in neural information processing systems*. pp. 3104–3112 (2014)
40. Tokmakov, P., Alahari, K., Schmid, C.: Learning motion patterns in videos. In: *CVPR* (2017)
41. Tokmakov, P., Alahari, K., Schmid, C.: Learning video object segmentation with visual memory. In: *ICCV* (2017)
42. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: *ICCV* (2015)
43. Tsai, Y.H., Yang, M.H., Black, M.J.: Video segmentation via object flow. In: *CVPR* (2016)
44. Voigtlaender, P., Leibe, B.: Online adaptation of convolutional neural networks for video object segmentation. *arXiv preprint arXiv:1706.09364* (2017)
45. Williams, R.J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. *Neural computation* **1**(2), 270–280 (1989)
46. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: *Advances in neural information processing systems*. pp. 802–810 (2015)
47. Xu, H., Gao, Y., Yu, F., Darrell, T.: End-to-end learning of driving models from large-scale video datasets. In: *CVPR* (2017)
48. Xu, N., Price, B., Cohen, S., Yang, J., Huang, T.: Deep grabcut for object selection. *arXiv preprint arXiv:1707.00243* (2017)
49. Xu, N., Price, B., Cohen, S., Yang, J., Huang, T.S.: Deep interactive object selection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 373–381 (2016)
50. Yang, L., Xiong, X., Wang, Y., Yang, J., Katsaggelos, A.K.: Efficient video object segmentation via network modulation. In: *CVPR* (2018)