The 2017 DAVIS Challenge on Video Object Segmentation - CVPR 2017 Workshops

One-Shot Video Object Segmentation with Iterative Online Fine-Tuning

Amos Newswanger
University of Rochester
Rochester, NY 14627
anewswan@u.rochester.edu

Chenliang Xu
University of Rochester
Rochester, NY 14627
chenliang.xu@rochester.edu

Abstract

Semi-supervised or one-shot video object segmentation has attracted much attention in the video analysis community recently. The OSVOS model [1] achieves state of the art results on the DAVIS 2016 dataset by first fine-turning a CNN model on the first pre-segmented frame of a video, and then independently segmenting the rest of the frames in that video. However, the model lacks the ability to learn new information about the object as the object evolves throughout the video displaying features that were not present in the first frame. To address this issue, we propose an iterative online training method whereby the model is fine-tuned on the first frame, segments several consecutive frames independently, and then gets updated on its own output segmentation. This process is repeated until all frames of a video are segmented. To segment multiple similar objects in a video, we use an object tracker to filter the output of the individually trained CNN object models before being used for iterative fine-turning. This reduces the possibility for error propagation, and helps the model increase its discriminative power as it is being iteratively fine-tuned. Our method shows improvement over the standard OSVOS model on both DAVIS 2016 and 2017 datasets.

1. Introduction

In recent years, Convolutional Neural Networks (CNNs) have achieved state of the art results in many computer vision tasks, such as image classification [8] and object detection [2]. Video object segmentation, or the separation of an object from its background in a video sequence, is a related task that has also come to be dominated by deep learning methods [3, 9, 1, 4]. Among them, the One-Shot Video Object Segmentation (OSVOS) model, a fully convolutional network introduced in [1], achieves state of the art performance in the DAVIS 2016 competition [5].

The OSVOS model is based on the VGG [7] network, which is pre-trained on a generic task of image classification on the ImageNet. The network performs convolution on intermediate values taken from the VGG network to produce a segmentation mask. This network is further trained offline on the training videos of DAVIS 2016 dataset to learn a general concept of foreground objects, and hence, it is called the parent network. To perform video object segmentation on a given testing video, the parent network is first fine-tuned on the pre-segmented ground-truth frame to learn the appearance features of the object in question, and then is used to independently segment the rest of the frames in the video. Although this approach has many desirable qualities, it lacks the ability to learn new information about the object as it evolves throughout the video. This reduces its performance on sequences where the initial frame lacks information about the object that becomes important later on in the sequence. For instance, Table 1 shows the scooter-black sequence, in which the first frame contains relatively little information about the object. As the object gets closer to the camera, the network fails to segment it properly. However, segmentations leading up to the failure are correct, and contain information about the object that could be used to correct the failure.

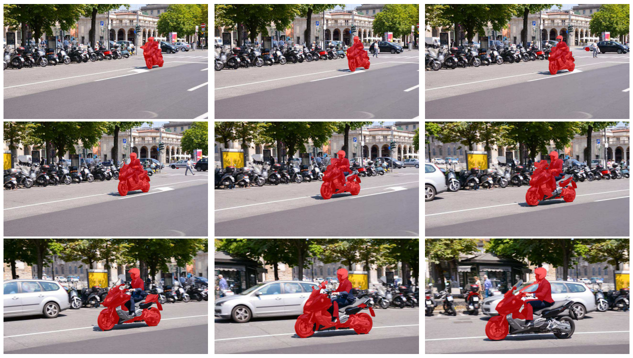

Figure 1. The first frame contains relatively little information about the object, and as a result, the OSVOS model fails to segment it correctly towards the end. However, earlier correct segmentations contain useful information about the object that can be used to further train the model

We present a method for iterative online fine-tuning of the OSVOS network. As shown in Fig. 2, we first fine-

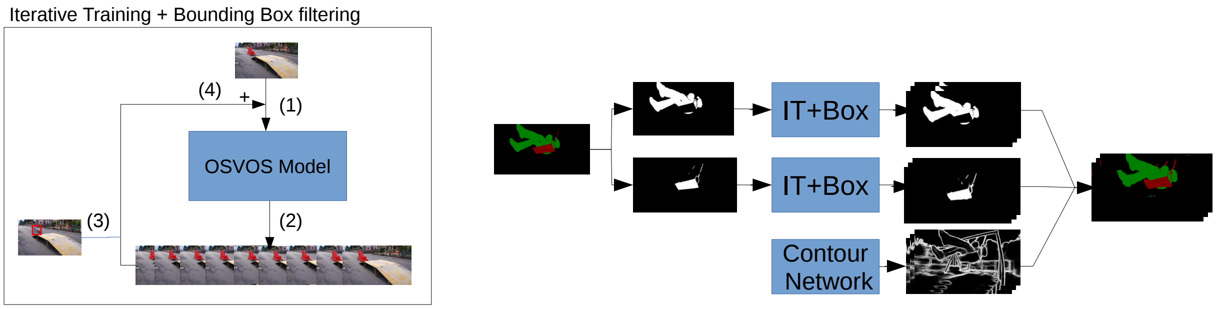

Figure 2. Overview of our method. On the left: (1) The OSVOS model is first trained on the ground truth segmentation of the first frame. (2) This model is used to segment some number of frames. (3) These segmentations are filtered using a bounding box tracker. (4) The filtered segmentations are added to the training set, and the model is further fine-tuned. The diagram on the left shows how we independently manage each object in in a multi-object mask, and then combine the results and snap the boundaries to a contour.

tune the OSVOS parent network on the first frame of the video. We then use this model to independently segment some number of frames. These frames are then used to further fine-tune the network. This process is repeated until all frames of the video are segmented. We evaluate our method on both the DAVIS 2016 and 2017 datasets. As shown in Fig. 2, we deal with the multiple object masks in the DAVIS 2017 dataset by first separating them into separated binary masks and running our method independently on each one, and finally combining the results by taking the maximum output of all the models and snapping the boundaries to a contour.

We evaluate our method with three metrics: Intersection-over-Union (IoU or J), contour accuracy (F) and temporal stability (T). We compare the performance of our method on each sequence in the DAVIS 2016 validation set to the performance of the standard OSVOS model. Furthermore, we perform additional evaluations on DAVIS 2017 videos where a single video contains multiple objects. Our method shows improvement over the standard OSVOS model on both DAVIS 2016 and 2017 datasets.

2. Method

Our method is straightforward. We use the parent network provided by Caelles, *et al.* [1], which is trained for 50,000 iterations on the DAVIS 2016 dataset (augmented by mirroring and zooming) with Stochastic Gradient Descent and momentum of 0.9. For the online training, we fine-tune the model for 300 iterations on the first frame of the video to train the model to recognize the specified object in question. We then use this model to independently segment the next 10 frames in the sequence. These 10 frames are then added to the training set, and the model is fine-tuned for 100 iterations on them. This process is repeated every 10 frames until all the frames in the sequence are segmented. To refine the segmentation, we snap the boundaries to contours generated by the same contour network used by Caelles, *et al.* Our method based on iterative fine-tuning adapts the network to the object as it evolves throughout the sequence. However, it also presents the possibility to propagate errors made early on in the segmentation process. To mitigate this problem, we experiment with several ways of filtering the output of the network before being used for fine-tuning in Sec. 3.

For the DAVIS 2017 dataset, we adapt our method to handle multiple objects. We use the same parent network as that for the DAVIS 2016 dataset, but train it for additional 10,000 iterations on the DAVIS 2017 TrainVal set, using the merged binary mask as the ground truth, such that the model has a better idea of DAVIS 2017 objects. To deal with the multiple objects in a same video, we first split the multi-object mask into separate binary masks for each object. We then run our method on each mask independently and get a probability map for each object, which we merge into a single multi-object mask by taking the maximum output value of each model for each pixel. The mask is then further refined by snapping the boundaries to contours generated by the same contour network used by Caelles, *et al.* We find that because in many cases the DAVIS 2017 dataset has multiple objects with similar appearances, the OSVOS model has a hard time distinguish between them. To mitigate this problem, we use the OpenCV KCF object bounding box tracker to filter the output segmentation before being used to iteratively fine-tune the model. This reduce the possibility of error propagation, and helps the model increase its discriminatory ability as it is being iteratively fine-tuned.

3. Experiments

3.1. DAVIS 2016

Our first set of experiments was performed on the DAVIS 2016 dataset [5], which contains 50 video sequences, each with one object segmented in all the frames at pixel level. Our main metrics are Intersection-over-Union (IoU or J)

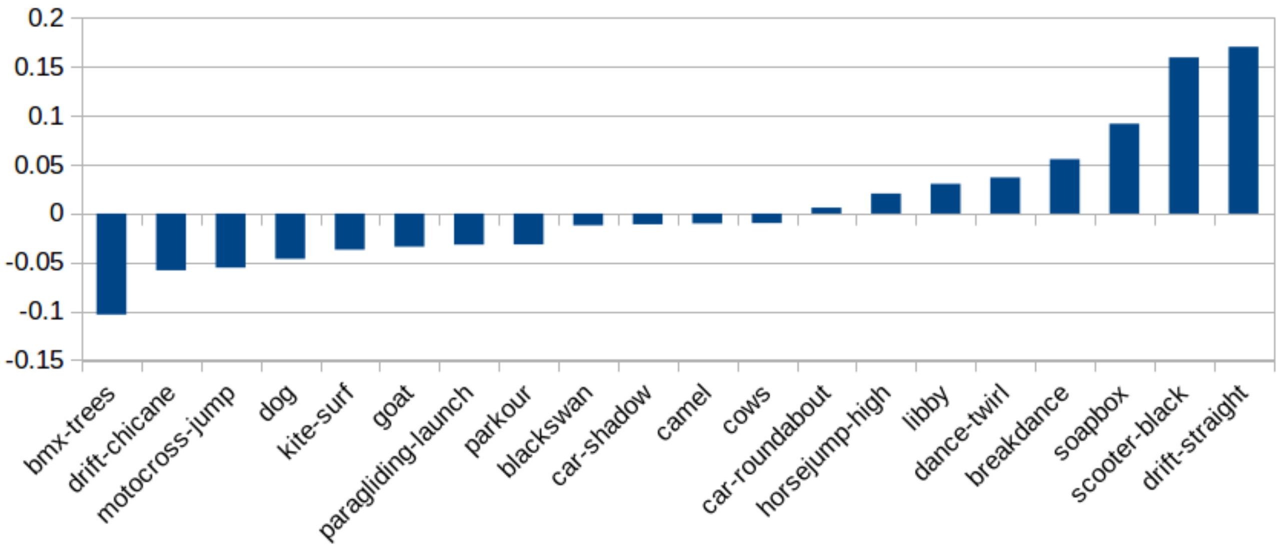

Figure 3. Relative difference in IoU between the normal OSVOS model and our best performing method (IT) on DAVIS 2016.

and Contour Accuracy (F). We mainly compare our results to the state of the art results obtained by the OSVOS model [1] in the DAVIS 2016 competition.

Table 1 shows the overall results on the DAVIS 2016 validation set. Our method performs slightly better in all metrics. Figure 3 shows the relative performance for each sequence, and reveals that most of the gains come from relatively few sequences, while the accuracy on the majority of the sequences is slightly reduced. The most improved sequence (drift-straight, shown in Fig. 4) only displays the front side of the car in the initial frame. As the sequence progresses, the broad side of the car is shown, and then the back side. Similarly, the second and third most improved sequences display objects at an angle in the first frame and display more and more features as the sequence progresses. This demonstrates the methods ability to pick up new features as the model is iteratively trained.

On the other end of the spectrum, the most harmed sequence (bmx-tree, show in Fig. 5) shows the shortcomings of the method. The OSVOS model picks up many false positives in the bmx-trees sequence and the iterative training method propagates these error. This same effect can be seen in the other sequences, though to a lesser extent. To mitigate this issue, we experimented with several ways of filtering the segmentation before being used for iterative training. The simplest solution is to only train the model on the largest blob (shown as IT+LB in Table 1), with the assumption that the largest blob is most likely to be the correct object. For some sequences, this method works well, but fails in many cases because the largest blob may not be the correct object, or the correct segmentation may not be a continuous blob. We also experimented with using the OpenCV KCF bounding box tracker to filter the segmentation by setting everything outside of the box to zero (shown as IT+Box in Table 1). However, this method also fails to improve the results due to the poor performance of the tracker on the DAVIS 2016 validation set.

Table 1. DAVIS 2016 Validation Results (top two results are bold)

	Measure	OSVOS	IT	IT+LB	IT+Box
J	Mean (↑)	**0.798**	**0.804**	0.794	0.777
	Recall (↑)	**0.936**	**0.974**	0.926	0.911
	Decay (↓)	0.149	**0.091**	**0.138**	0.158
F	Mean (↑)	**0.806**	**0.809**	**0.806**	0.799
	Recall (↑)	0.926	**0.934**	**0.937**	0.915
	Decay (↓)	**0.150**	**0.124**	0.152	0.157

3.2. DAVIS 2017

Our second set of experiments was performed on the DAVIS 2017 data set, which contains 150 sequences (90 in the TrainVal set, 30 in the Test Dev set, and 30 in the Test Challenge set) [6]. Each sequence has multiple objects segmented at pixel accuracy for all frames in the sequence. The metrics used to evaluate the results are the same as those used on the DAVIS 2016 dataset.

Table 2 shows our results on the Test Challenge dataset for three different methods. IT stands for iterative training and Box stands for the use of the OpenCV KCF bounding box tracker for filtering the segmentation before being used for iterative training. The first thing we found is that the accuracy on the DAVIS 2017 dataset is much lower than on the DAVIS 2016 dataset. This could be for several reasons. Many of the DAVIS 2017 sequences contain objects that look very similar, which could present a challenge for he OSVOS model, given that it has no information about mo-

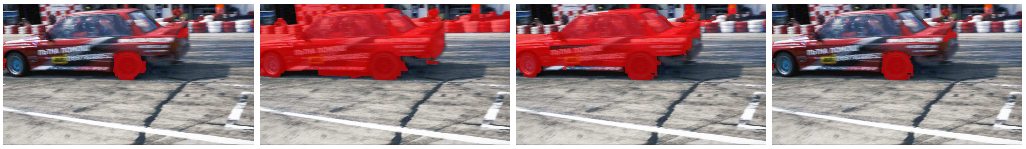

Figure 4. Comparison of different methods drift-straight from the DAVIS 2016 dataset. In order: OSVOS, IT, IT+LB, IT+Box.

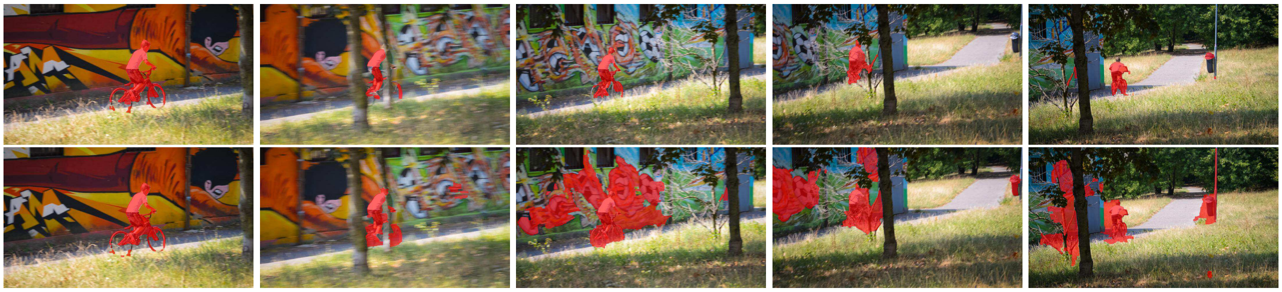

Figure 5. Most harmed sequence (OSVOS on top, IT on bottom) from the DAVIS 2016 dataset.

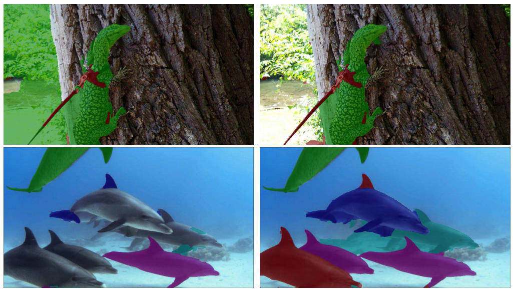

Figure 6. Comparison between OSVOS(left) and IT+Box (Right) on DAVIS 2017 videos.

Table 2. DAVIS 2017 Test Challenge Results (top two results are in bold). The overall metric is the mean of J and F over all object instances

	Measure	OSVOS	IT	IT+LB	IT+Box
	Overall (↑)	0.488	0.471	**0.500**	**0.509**
J	Mean (↑)	0.462	0.448	**0.481**	**0.490**
	Recall (↑)	0.515	0.479	**0.533**	**0.551**
	Decay (↓)	0.253	0.286	**0.222**	**0.213**
F	Mean (↑)	0.514	0.494	**0.519**	**0.528**
	Recall (↑)	**0.582**	0.524	0.576	**0.583**
	Decay (↓)	0.257	0.291	**0.240**	**0.237**

tion or temporal continuity. In addition to this, the 2017 dataset has smaller objects than the 2016 dataset, which present more opportunities for false positives. Because of the increase in false positives, simply applying iterative training causes excessive error propagation and reduces the accuracy compared to the standard OSVOS model. To mitigate this problem, we used the OpenCV KCF bounding box tracker to filter the output segmentation before being used for iterative training. This resulted in an improvement over the standard OSVOS model. Figure 6 shows two examples where iterative training improves the results. Notably, in the varanus-tree sequence, the model learns to not segment the leaves that appears in the background towards the end of the sequence, which demonstrates the added discriminatory power that the bounding box adds.

4. Conclusion

In this paper, we show that iterative training provides a way to learn more information about an objects as it evolves through a sequence, and that our method shows an improvement over the state of the art on the DAVIS 2016 dataset, and over the standard OSVOS model on the 2017 dataset. However, the method is also prone to propagate errors made early on in the process. Future work may involve finding ways to reduce the potential for error propagation and learning an automatic model to decide when to update the object model throughout the video.

References

[1] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):142–158, 2016.

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

[4] A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. Technical report, arXiv:1612.02646, 2016.

[5] 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 *Computer Vision and Pattern Recognition*, 2016.

[6] 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.

[7] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[9] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. Technical report, arXiv:1704.05737, 2017.