AN ABSTRACT OF THE MASTER'S PROJECT REPORT OF

Rahul Borkar for the degree of Master of Science in Computer Science presented on May 20, 2019.

Title: Video Object Segmentation By Jointly Tracking Foreground and Background

Abstract approved: _____

Sinisa Todorovic

This report presents an efficient method for semi-supervised video object segmentation – the problem of identifying foreground pixels occupied by a target object. The target is specified by the ground-truth mask in the first video frame. While the state of the art achieves a segmentation accuracy greater than 80%, it runs relatively slow at less than 10 frames per second. This limits their application in many domains. In addition, accuracy of existing approaches typically suffers on cases of target occlusion by moving background objects. We address these two shortcomings of prior work by a novel deep architecture aimed at jointly tracking both foreground and background in the video in an efficient manner. Our key hypothesis is that explicitly tracking the dynamic background of the target object helps improve segmentation in cases of target occlusion. We propose using two deep neural networks that work in parallelone for foreground object segmentation, and the other for background segmentation. They use the same architecture. Their output is integrated in another network for fusing the initial foreground and background segmentation into a more accurate target object segmentation. We perform experiments using various configurations of the proposed architecture on the DAVIS 2016 dataset. Our results support the key hypothesis where the joint tracking of the dynamic foreground and background indeed outperforms a baseline that tracks only the target object. On DAVIS 2016, our accuracy is 70.61%, while operating at over 100 frames per second.

Video Object Segmentation By Jointly Tracking Foreground and Background

by

Rahul Borkar

A MASTER'S PROJECT REPORT

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented May 20, 2019
Commencement June 2019

Master of Science master's project report of Rahul Borkar presented on May 20, 2019.

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my master's project report will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my master's project report to any reader upon request.

Rahul Borkar, Author

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

Chapter 1: Introduction

This research report presents an approach to semi-supervised video object segmentation. Our goal is to segment an object of interest from the background in a video, given the ground-truth segmentation for the first frame. Figure 1.1 shows the target input/output for our system. We focus on settings where object classes are not known, and cannot be learned in advance. Our system is not designed for a set of predefined classes; it is designed to segment all types of objects. We also assume that target objects are prominently featured and occupy a large area in input frames.

Several challenges make highly accurate segmentation a difficult task that make object segmentation an actively researched topic. One challenge is that many objects do not move in a smooth manner, making it difficult to accurately capture drastic changes in object appearance. Second, objects may be subject to some level of shape deformation, making object appearances more complex to capture with a model. Next, video capture is not always reliable; viewpoint changes and other camera related issues make it necessary for networks to be more versatile. Finally, dynamic backgrounds, can create a variety of issues, with one of the most challenging being occlusion. The combination of these challenges make creating a state-of-the-art object segmentation network a formidable task. Figure 1.2 exemplifies some of these issues: despite being less than a 4 second video, motion blur and heavy occlusion cause the segmentation system to fail.

Current methods suffer from large processing times. In this work, we focus on efficiency. Therefore, our design choices are limited to relatively simple, feed-forward

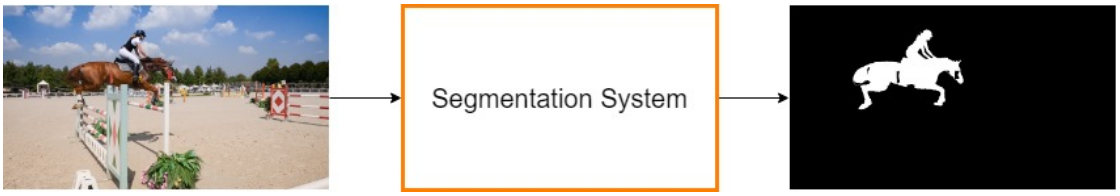


Figure 1.1: Example of the desired input and output for our system: Our system is fed an input frame from a video, and produces a segmentation for that frame

networks. Using more complex architectures e.g., RNNs, and temporal modeling of objects dynamics, would be too expensive. Therefore, we just use appearance modeling for efficiency.

Deep learning methods have led to substantial improvements in solving many types of problems in various types of fields. Particular to the field of computer vision, Convolutional Neural Networks (CNNs) have allowed for researchers to create models that are able to perform many tasks. CNNs have performed very well in object segmentation, despite the numerous challenges associated with the task. Because of this, we use CNNs to implement our approach.

Our key idea is to use a complimentary background segmentation network to improve foreground separation. Explicit background segmentation is an unexplored in tracking. Typically, prior work considers space-time context around the target, but only for the purposes of foreground segmentation. By explicitly segmenting background, we effectively track foreground, which can refine our segmentation result. Our goal is to determine the viability of background segmentation, rather than outperforming SOTA methods. To ensure our results can clearly show the advantages and disadvantages of background segmentation, we forego any complex post-processing, which accounts for why our results do not match SOTA performance. We hypothesize that our approach that segments the background surrounding an object in addition to foreground segmentation would lead to more robust foreground/background separation without the need for additional training data.

1.1 Our Background Segmentation

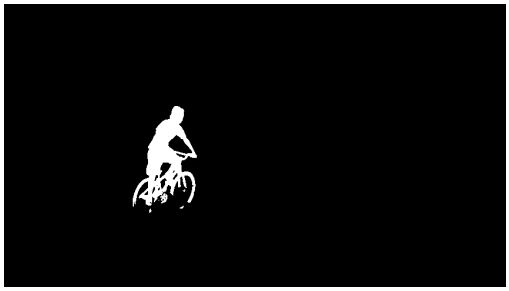
Current deep networks for object segmentation require large training datasets. Typically in training, we do not have access to large numbers of annotated frames. Therefore, SOTA performs various types of data augmentation to allow networks to sufficiently learn object deformations, including: rotation, scaling, and mirroring. We follow existing work, and augment our training set using the same type of image transformations. Given this augmented set of frames, we create complimentary background frames for training a background segmentation network.

(a) bmx-trees frame 45

(b) bmx-trees frame 52

(c) bmx-trees frame 45 segmentation result

(d) bmx-trees frame 52 segmentation result



(e) bmx-trees frame 45 ground-truth

(f) bmx-trees frame 52 segmentation result

Figure 1.2: Segmentation fails (d) only 5 frames after the first frame (c) due to motion blur and occlusion on the bmx-trees sequence from DAVIS 2016 [10]

1.2 Related Work

SOTA methods in video object segmentation focus on object appearance to achieve highly accurate tracking. Most work uses a mask refinement method to create motion modeling. For example, [1] learns to refine detected masks frame by frame by using the mask prediction for the previous frame. Figure 1.2 shows an example of [1] segmenting a frame using its previous prediction. [6] expands on this method by generating several mask proposals for a given frame based on the previous frame, and merging these proposals. Mask propagation style methods also use optical flow to assist in motion modeling [5, 6, 12]. However, optical flow is computationally expensive, and significantly hinders speed.

Despite segmenting objects well, mask propagation methods that use optical flow sacrifice run time for accuracy. Other SOTA methods like One-Shot Video Object Segmentation (OSVOS)[3] do not make use of motion modeling. This is typically done in order to reduce error propagation and run time. Despite not using mask propagation, OSVOS and other similar methods still achieve competitive results. Figure 1.3 shows the system components and dataflow for OSVOS. [8] adds to OSVOS by combining an initial segmentation with semantic instance information and a conditional classifier. [12] expands on OSVOS by updating its network online using training examples selected based on the confidence of the network.

There has been a wide variety of exploration in object tracking methods, particularly with adding post processing to extract further information from images. However, little work has been done in determining whether using background segmentation can improve foreground object segmentation in any situations.

Several approaches for object segmentation are able to achieve accurate results on popular datasets [1, 12, 6, 8, 2]. It is not suitable to use these methods for our purposes because they require large amounts of memory and time [9]. Instead, we focus on efficiently segmenting objects, while maintaining a comparable level of accuracy.

1.3 Motivation

We would like to segment an object in a new video, where the only piece of information available is the foreground/background segmentation in one frame. Previous work has

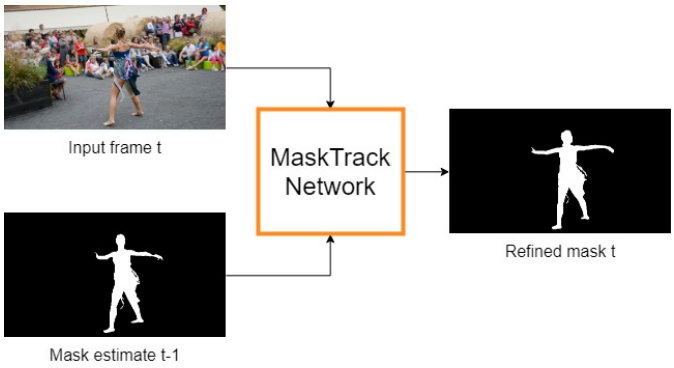


Figure 1.3: Masktrack [1] dataflow diagram. The network takes an input frame t, and a previous frame prediction t-1, and refines the mask for t-1 to produce a prediction for frame t.

shown that appearance modeling is the most critical element in successfully segmenting an object. Given a Fully Convolutional Neural Network (FCN)[3] trained for this task, we would like to determine whether segmenting its surrounding background and combining it with the foreground segmentation can create a more accurate model.

1.4 Approach

To create a segmentation system that is complimented by background segmentation, we begin with a network that is able to segment foreground objects. Because we focus on appearance based modeling, we use the publicly available OSVOS architecture. This network is a VGG16 based architecture, consisting of convolution plus Rectified Linear Unit (ReLU) layers grouped into five stages, separated by max pooling layers. OSVOS makes two specific changes to this VGG architecture. First, the fully connected layers are removed. Second, skip paths from the end of each stage before pooling are made with appropriate upscaling. Feature maps from these paths are then concatenated to create an output that contains varying levels of detail, which are then combined using a final convolution layer to create a prediction. Architecturally, the foreground and background segmentation networks are identical.

To create a new background segmentation network, we first construct complimentary background data by inverting the segmentation mask in a box surrounding the object of

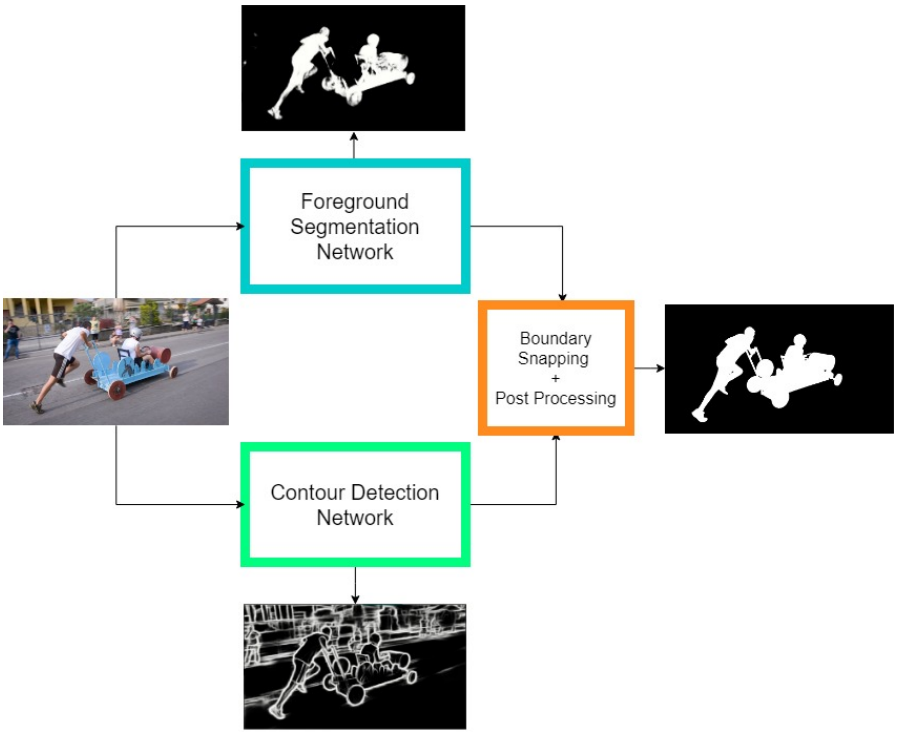


Figure 1.4: Dataflow diagram for One Shot Video Object Segmentation (OSVOS) [3]. Our approach replaces the contour detection branch with our background segmentation branch, and replaces the post processing steps with a shallow integration network.

interest in the original ground-truth data, shown in Figure 1.5. We then train a second network using the same training procedure, but with our new background annotations.

Because we need to combine the outputs of these two networks and the OSVOS architecture concatenates deep features in a final integration layer, we do not use the output of these output layers. Instead we then create a separate network consisting of a single convolution layer that takes the features of the foreground and background networks, and the original image to create our final segmentation mask. We keep the outputs of the foreground and background branches unmodified to allow the integration network to learn how to combine the predictions into a sharper mask. An overview of our system dataflow is shown in Figure 1.4.

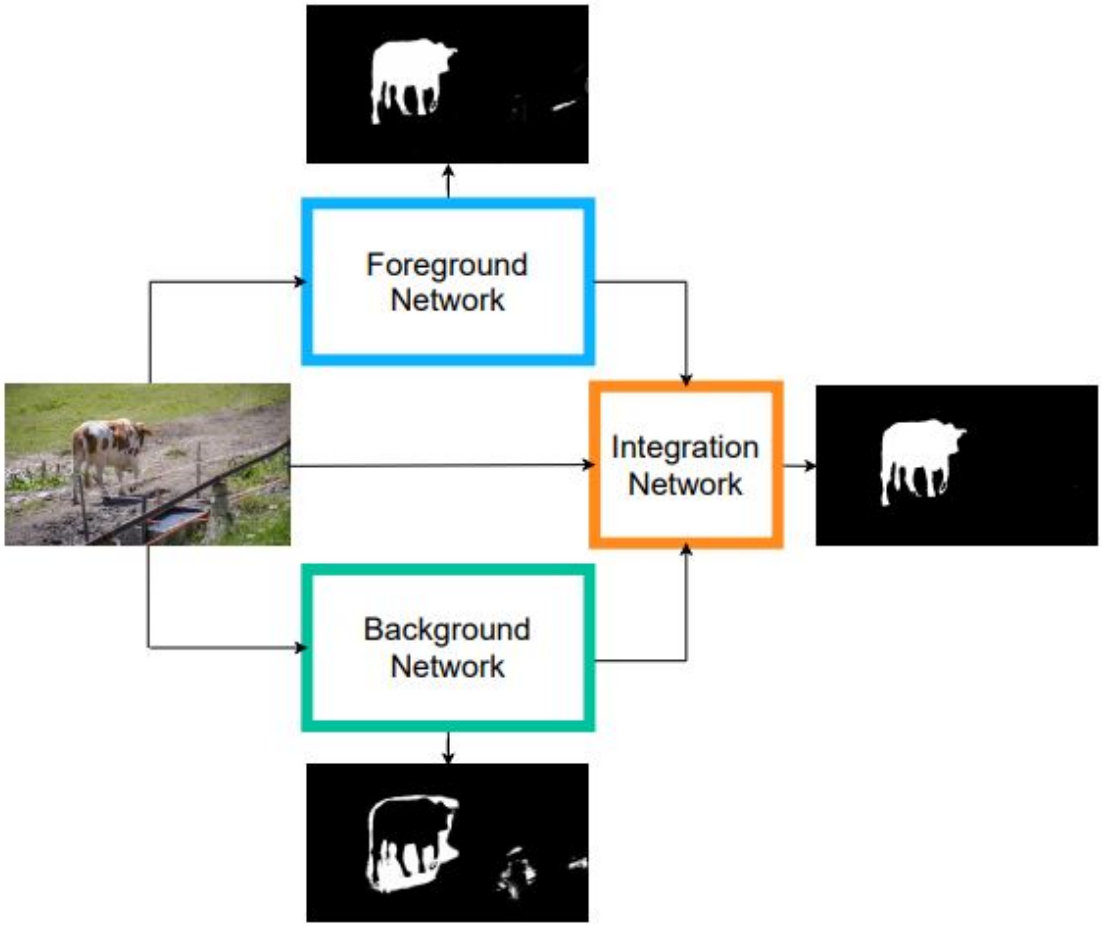


Figure 1.5: Our Segmentation system dataflow diagram: frames are passed to our foreground and background networks, and these outputs are then passed to the integration network along with the original image to make a final mask

1.4.1 Implementation Details

To train our foreground network, we follow the procedure for training OSVOS [3], starting with weights pretrained on ImageNet [4]. Using these weights, we train the network on DAVIS so it can learn what objects are, and how to segment them. For this offline training, we use stochastic gradient descent (SGD), with the learning rate set to 10^{-8}, a weight decay of 2×10^{-4}, and a momentum of 0.9, for 240 epochs. We then perform online training at testing time on the first frame of the sequence to adapt the network

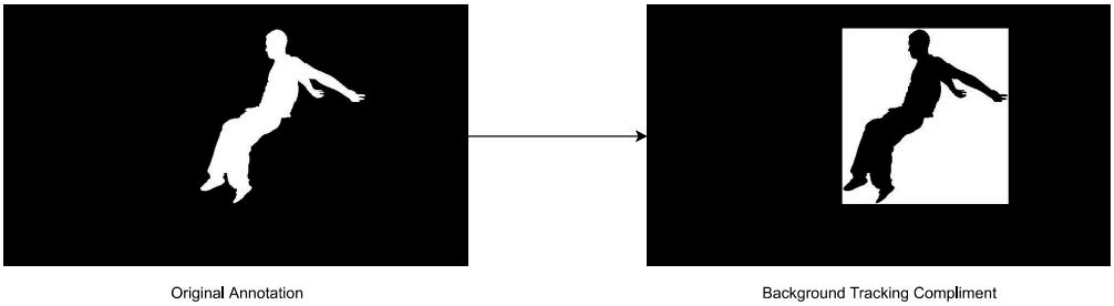


Figure 1.6: Example of how we construct a background ground truth mask from a given foreground ground truth mask

to a specific object using the same settings, but for 10,000 epochs.

To train our background network, we first construct complimentary background data by inverting the segmentation mask in a box surrounding the object of interest in the original ground-truth data, shown in Figure 1.5. We then train a second network using the same training procedure as the foreground network, but with our new background annotations.

After performing online training for both these networks on a specific sequence, we then finally perform the same online training using the adapted foreground and background weights. While this means that online training time is doubled our results show that this trade-off leads to a substantial increase in segmentation quality. This quality increase primarily manifests in the elimination of artifacts, and sharper edges in the final segmentation mask.

1.5 Evaluation

We run our model on DAVIS 2016 [10] and compare our results and compare them to SOTA methods. We use DAVIS 2016 because it is a popular single object segmentation dataset. We do not use DAVIS 2017 or 2018 because they are for multiple object tracking. We also compare our method to the performance of the base network without background segmentation, as well as various training methods for our background segmentation network. We train the network using various background sizes, creating 0 pixel, 2 pixel, and 4 pixel borders around the objects in each sequence, and compare

how they perform.

In our evaluation, we measure mean intersect over union (mIoU), IoU over time, as well as processing time time per frame.

Our model slightly larger than twice the size of the base network; the background branch doubles the size, and the integration network adds one additional layer. We justify this complexity increase because being able to run the foreground and background network in parallel leads to a relatively small increase in time per frame from 4.7 ms/frame to 8.9 ms/frame, and an increase of accuracy from an IoU of 60.70 to 70.60.

Chapter 2: Results

2.1 Datasets

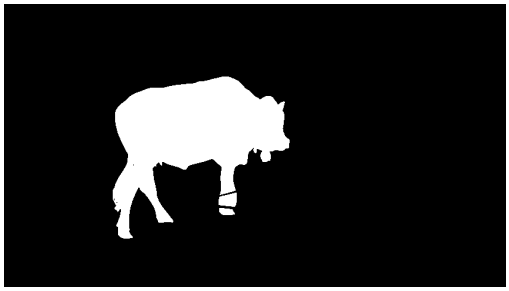
We run various versions of our model over the DAVIS 2016 to get a broad understanding of how the background segmentation network performs for different types of objects and video conditions. We do not use DAVIS 2017 or 2018 because they are designed for multiple object tracking. DAVIS 2016 contains a total of 50 sequences, 3455 annotated frames, all captured at 24 frames per second at a full HD 1080p resolution [10]. However, like many other SOTA segmentation systems, we use the 480p versions of the images.

2.2 Metrics

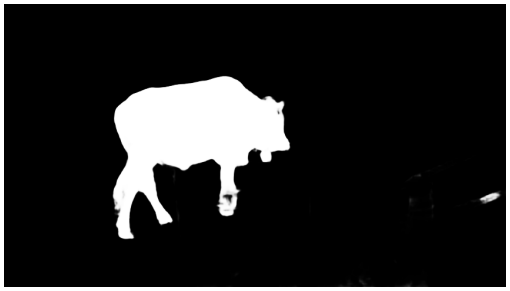
To measure the performance of our network we use mean intersect over union (mIoU), which measures the ratio of the overlap between pixels in the ground-truth and the prediction. We perform semi-supervised training by only performing online training on the first frame of sequences. We also plot the mIoU as a function of time to determine how our model performs for appearances that vary from the first frame. We also measure the percentage of frames over various IoU levels to measure the consistency of our segmentation.

2.3 Quantitative Results

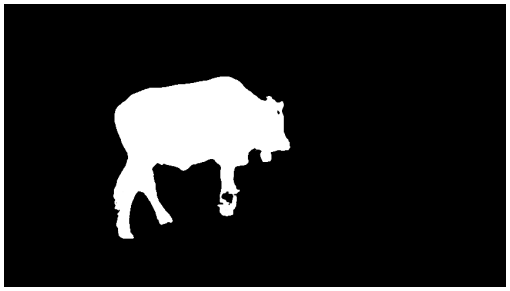
Our best results were achieved by training our background network on foreground ground-truth data offline, and performing online training on background ground-truth data. Similar to the OSVOS training approach, this model learns what foreground objects look like. However, in online training it treats a provided background as an object. An example of how well this model performs is shown in Figure 2.1. Table 2.1 shows our results on DAVIS 2016 across various versions of our model, including the base network which does not use any background segmentation. Each column of Table 2.1 shows: the base network without using background segmentation (BN), foreground offline training



(a) ground-truth segmentation



(b) Base Network segmentation



(c) FGO-2px segmentation

Figure 2.1: Example frame of our FGO model outperforming the base network (0.76 IoU vs. 0.96 IoU) on the cows sequence from DAVIS [10]: Our model eliminates artifacts at the edge of the frame and has a much sharper result than the base network.

for the background network with 2 pixel borders (FGO-2px), and background trained normally with 0 pixel (0px), 2 pixel (2px), and 4 pixel (4px) borders.

Although the our network does not compete with the top results in terms of accuracy, our best model achieves an mIoU of 70.61%, with a remarkably fast speed of 8.9 ms/frame or 112 frames per second. This speed is a result of not having any complex pre-processing or post-processing, common to many SOTA methods. Our method is able to achieve a frame rate more than 10 times higher than top SOTA models, while sacrificing under 11% IoU when compared to [9]. Additionally, with further fine tuning of training parameters, our IoU could improve further.

Comparing the performance of our models, training the background network on foreground data offline and treating the background like an object in online training is the most effective way to use background features. However, based on our 0px, 2px, and 4px models, it seems the size of the background segmentation area does not have a major

Sequence	BN	FGO-2px	0px	2px	4px
Blackswan	76.6	93.88	93.58	92.32	93.18
Bmx-Trees	16.98	29.83	24.01	26.09	22.67
Breakdance	69.3	68.66	68.64	68.89	69.19
Camel	74.2	85.87	85.98	86.11	86.29
Car-Roundabout	77.44	86.2	84.79	82.44	84.99
Car-Shadow	80.94	88.7	88.34	88.45	88.36
Cows	80.29	94.73	93.94	94.12	94.83
Dance-Twirl	67.07	66.00	71.22	71.65	73.55
Dog	61.48	82.17	82.77	82.34	82.95
Drift-Chicane	53.26	64.5	64.38	64.66	67.09
Drift-Straight	64.7	63.62	64.45	63.08	65.38
Goat	71.73	80.79	77.28	77.38	81.72
Horsejump-High	58.87	74.3	72.57	74.33	60.51
Kite-Surf	22.02	36.46	31.21	31.58	29.38
Libby	52.76	68.25	74.47	75.3	74.53
Motocross-Jump	56.58	59.65	57.32	57.86	58.77
Paragliding-Launch	33.33	46.96	42.76	43.2	38.61
Parkour	55.22	75.41	71.29	71.12	72.26
Scooter-Black	65.76	65.07	60.93	59.41	62.93
Soapbox	75.56	81.08	77.21	77.69	77.93
Total	60.70	70.61	69.36	69.40	69.26

Table 2.1: Our IoU performance for various network configurations on DAVIS 2016 [10].

influence in foreground segmentation results.

Table 2.2 and Figure 2.2 demonstrate the consistency of our FGO-2px model across the dataset. The model is able to keep an IoU over 70% with nearly two-thirds of the frames. Several sequences have poor results due to large amounts of shape deformations and occlusion. However, Section 2.4.1 discusses adding additional annotated frames to improve accuracy in these cases.

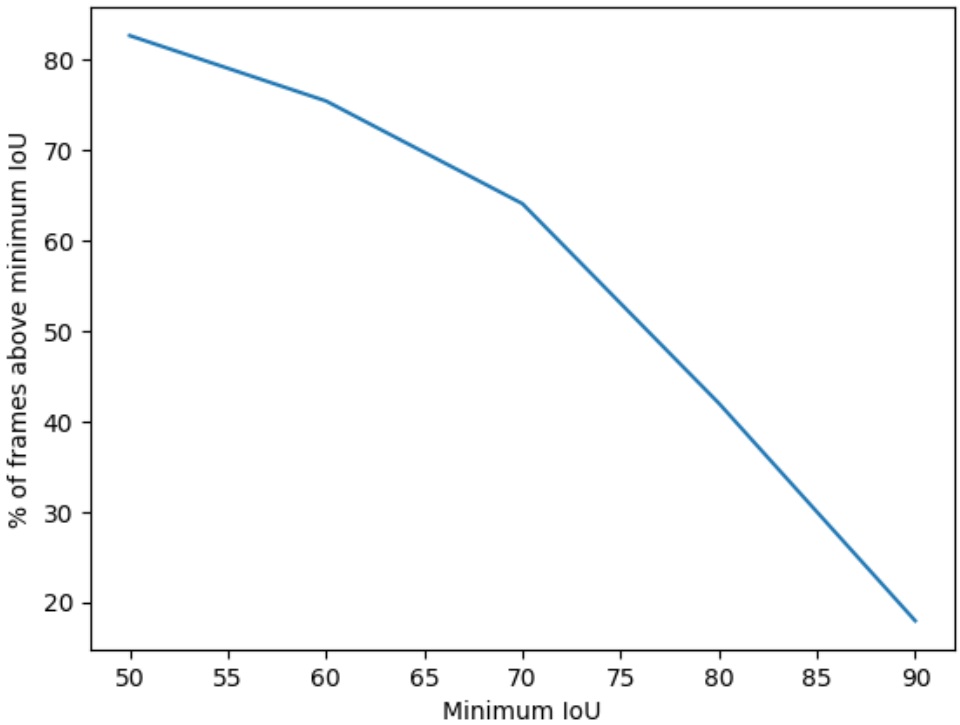


Figure 2.2: Percentage of frames (y axis) above various IoU levels (x axis) across DAVIS [10] using our FGO-2px model: We are able to achieve over 60% IoU in over 75% of the frames from the entire dataset.

Sequence	>50%	>60%	>70%	>80%	>90%
Blackswan	100	100	100	100	100
Bmx-Trees	17.72	16.46	0	0	0
Breakdance	87.95	81.93	53.01	18.07	2.41
Camel	100	100	100	74.16	34.83
Car-Roundabout	100	100	94.59	86.49	25.68
Car-Shadow	100	100	87.18	76.92	64.10
Cows	100	100	100	100	100
Dance-Twirl	95.51	64.04	34.83	9	0
Dog	100	98.31	84.75	66.10	16.95
Drift-Chicane	78.43	70.59	37.25	0	0
Drift-Straight	67.34	63.27	57.14	28.57	6.12
Goat	100	100	100	65.17	0
Horsejump-High	100	97.96	69.39	34.69	0
Kite-Surf	2.04	0	0	0	0
Libby	89.58	75	54.17	4.17	0
Motocross-Jump	74.36	48.72	25.64	15.38	0
Paragliding-Launch	34.18	0	0	0	0
Parkour	98.98	92.92	77.78	32.32	1.01
Scooter-Black	69.05	57.14	47.62	28.57	2.38
Soapbox	100	100	97.96	56.12	0
Total	82.67	75.44	64.09	42.04	17.99

Table 2.2: Percentage of frames above various IoU numbers on DAVIS 2016 [10] using our FGO-2px model

Finally we compare our model to SOTA methods by examining their accuracy relative to their processing time. Table 2.3 shows this comparison. Although our system does not achieve the same degree of accuracy as top performing models, we make up this difference by being able to process frames magnitudes faster. RGMP is the closest in speed to our model, but still runs at under 10 frames-per-second (FPS), while ours runs at over 100 FPS.

Method	mIoU	Seconds per frame
OSVOS-S [8]	85.6	4.5
OSVOS [3]	79.8	9
OnAVOS [12]	86.1	13
RGMP [9]	81.5	0.13
PLM [11]	70.0	0.3
BVS [7]	60.0	0.37
Ours	70.61	0.0089

Table 2.3: Quantitative evaluation of our method against SOTA on DAVIS-2016. Our model makes up for its accuracy by processing frames significantly faster than other methods.

2.4 Qualitative Results

To analyze how background segmentation influences foreground segmentation, we see cases where our model performs significantly better, as well as worse than the base network. Additionally, we examine cases where the network performs poorly in general.

The addition of the background network significantly improves results in the majority of cases. However, we see that in the breakdance, dance-twirl, drift-straight, and scooter-black sequences from DAVIS, the base network performs just as well or better than our models. Unlike many of the sequences where background tracking significantly improves our results, these sequences have many frames where the object appearance vastly differs from the initial frame that our model is trained on. In these cases, background segmentation has an adverse effect on our accuracy. Figures 2.3 and 2.4 demonstrates this; our model outperforms the base network until frame 17 on drift-straight, and does slightly worse than the base network for the rest of the sequence.

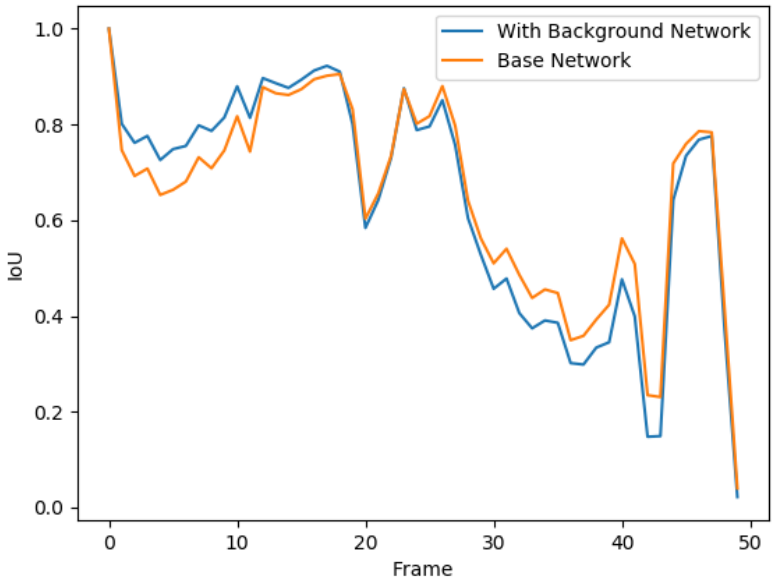
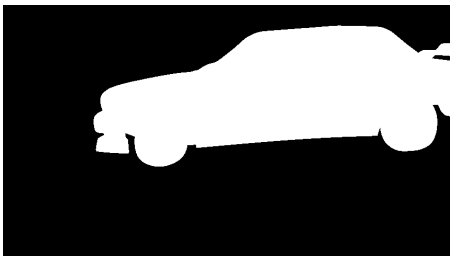


Figure 2.3: Example of Base Network and FGO-2px network performance on drift-straight sequence from DAVIS [10]: Our network outperforms the base network until frame 17, and performs worse on every frame after.

(a) ground-truth of frame 17

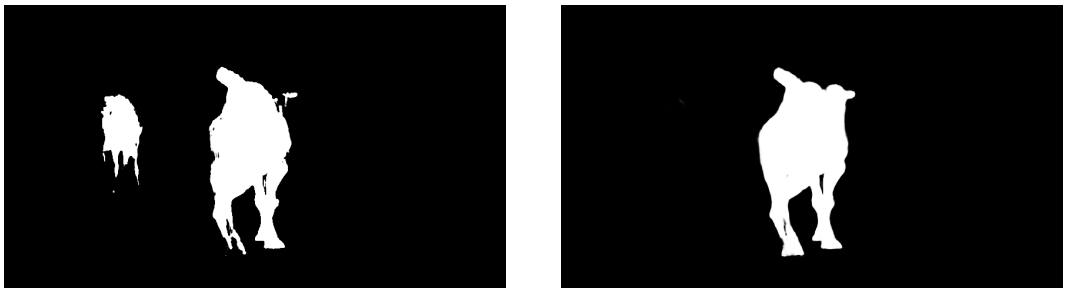


(b) ground-truth of frame 32

(c) Prediction of frame 17

(d) Prediction of frame 32

Figure 2.4: ground-truth (a) and (b) and predictions by FGO-2px model (c) and (d) for close to initial frame and far in drift-straight sequence from DAVIS [10].



(a) Training on frame 0 (b) Training on frames 0 and 87

Figure 2.5: Segmentation of frame 84 of the camel sequence from DAVIS [10]. Adding frame 87 to online training significantly improves segmentation towards the end of the sequence

2.4.1 Upper Bound Performance

Because our network does not require previous frames as input, we can easily incorporate more supervision in the form of additional annotated frames. As an example, we take our results on the camel sequence, where our worst performance is on frame 87, and perform online training using frames 0 and 87. We can continue to add more frames until the accuracy meets a minimum standard. Table 2.3 and Figure 2.5 demonstrate our results adding additional frames to online training. Using 2 annotated frames for the camel sequence provides the best results. However, adding additional frames afterwards significantly reduces the accuracy. This drop in accuracy occurs in the beginning of the video. This is due to the selection of annotated frames: because frames are 0, 87, and 89 are chosen, the network becomes biased towards the poses towards the end of the video.

Annotated frames	1	2	3	4
IoU	85.87	92.69	85.47	86.36

Table 2.4: Accuracy with respect to frames used in online training on camel sequence. Annotated frames are 0, 87, 89, and 29.

From these results, we can conclude that increasing accuracy by adding additional annotations is dependent on the diversity an object's appearance. For example: a basketball is unlikely to benefit from additional annotated frames due to being a rigid uniform object, while a breakdancer has many different poses that a network may need to learn

from. In the case of the sequence we tested on, there were only two poses distinct enough for the network to learn from: the camel facing the right at the beginning, and facing away at the end.

Chapter 3: Conclusion

Through this research, we explored how segmenting backgrounds in videos can help improve foreground object segmentation. Based on a network pre-trained on generic data from ImageNet [4], we propose an approach of segmenting the foreground and background of videos in parallel, and subsequently integrating these features to create a refined segmentation mask. We observe that the introduction of background segmentation consistently improves foreground segmentation, primarily by eliminating artifacts and sharpening the foreground mask. Our method also makes introducing additional training data simple, allowing an operator to easily add annotated frames.

While we achieve a comparable accuracy with SOTA, we are able to gain a significant performance increase over our baseline using background segmentation. Additionally, we offer a lightweight method without any post-processing that can run at over 100 frames per second, over ten times faster than top performing models.

The use of background segmentation in tracking is unexplored, so we selected a model and kept our hyper-parameters fixed to analyze the differences between settings within background segmentation. Further fine tuning of these values could increase our accuracy. We also selected a lightweight network architecture for this project, which may be limiting the contribution background tracking could have. We believe background segmentation could hold promising results in a more refined pipeline. Our method also does not make use of advanced data augmentation methods, such as Lucid Data Dreaming [5], which can diversify online training and reduce the need for additional annotated frames. Finally, our model lacks complex post-processing that could inflate our accuracy further to match leading segmentation methods. Because our model is able to achieve a 10% IoU increase over the baseline, we believe adding pre and post-processing steps could improve our accuracy to reaching more competitive results.

Bibliography

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

[2] Linchao Bao, Baoyuan Wu, and Wei Liu. CNN in MRF: video object segmentation via inference in A cnn-based higher-order spatio-temporal MRF. *CoRR*, abs/1803.09453, 2018.

[3] S. Caelles, K.K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-Shot Video Object Segmentation. *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. *Computer Vision and Pattern Recognition (CVPR)*, 2009.

[5] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele. Lucid data dreaming for object tracking. 2017.

[6] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *Asian Conference on Computer Vision*, 2018.

[7] Nicolas Maerki, Federico Perazzi, Oliver Wang, and Alexander Sorkine-Hornung. Bilateral space video segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] 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. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.

[9] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. 2018.

[10] 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. 2016.

[11] Jae Shin Yoon, Francois Rameau, Junsik Kim, Seokju Lee, Seunghak Shin, and In So Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[12] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. 2017.