Video Object Segmentation using Deep Learning

Zack While[1], Chen Chen[2], Rui Hou[2], Mubarak Shah[2]
Youngstown State University[1], University of Central Florida[2]

Center for Research in Computer Vision
UCF | UNIVERSITY OF CENTRAL FLORIDA

Problem & Motivation

- **Objective:** recognizing the predominant object in a video and distinguishing it per-pixel from the background in each frame.
- **Applications:** video surveillance, machine vision, content-based video retrieval.
- **Challenges:** substantial camera motion and sections where the main object is obstructed or changes appearance.
- **Project goals:**
  1. Training an end-to-end 3D convolutional neural network to perform object segmentation in video.
  2. Comparing results to the current state of the art on the DAVIS 2016 dataset.
  3. Fine-tuning the model to generate results that are competitive with the current state of the art.
- **Dataset:** DAVIS 2016 [1], which includes 50 HD video sequences where each frame is annotated per-pixel.

Architecture Description

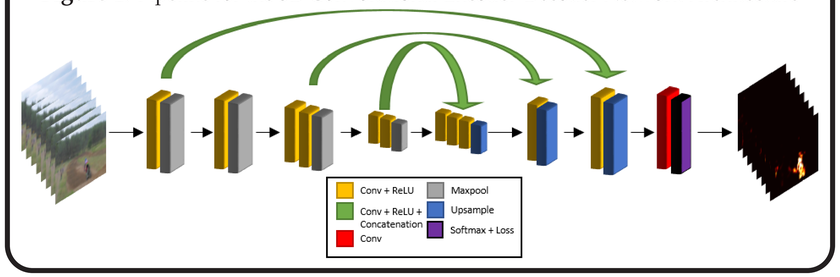
Table 1: Detailed Network Architecture

name	kernel dims $(d \times h \times w)$	dims $(C \times D \times H \times W)$
conv1	$3 \times 3 \times 3$	$(64 \times 8 \times 240 \times 320)$
pool1	$1 \times 2 \times 2$	$(64 \times 8 \times 120 \times 160)$
conv2	$3 \times 3 \times 3$	$(128 \times 8 \times 120 \times 160)$
pool2	$2 \times 2 \times 2$	$(128 \times 4 \times 60 \times 80)$
conv3a	$3 \times 3 \times 3$	$(256 \times 4 \times 60 \times 80)$
conv3b	$3 \times 3 \times 3$	$(256 \times 4 \times 60 \times 80)$
pool3	$2 \times 2 \times 2$	$(256 \times 2 \times 30 \times 40)$
conv4a	$3 \times 3 \times 3$	$(512 \times 2 \times 30 \times 40)$
conv4b	$3 \times 3 \times 3$	$(512 \times 2 \times 30 \times 40)$
pool4	$2 \times 2 \times 2$	$(512 \times 1 \times 15 \times 20)$
conv5a	$3 \times 3 \times 3$	$(512 \times 1 \times 15 \times 20)$
conv5b	$3 \times 3 \times 3$	$(512 \times 1 \times 15 \times 20)$
conv4c	$3 \times 3 \times 3$	$(512 \times 1 \times 15 \times 20)$
trans4	$2 \times 2 \times 2$	$(512 \times 2 \times 30 \times 40)$
conv4d	$3 \times 3 \times 3$	$(64 \times 2 \times 30 \times 40)$
concat4	-	$(576 \times 2 \times 30 \times 40)$
conv3c	$3 \times 3 \times 3$	$(512 \times 2 \times 30 \times 40)$
trans3	$2 \times 2 \times 2$	$(512 \times 4 \times 60 \times 80)$
conv3d	$3 \times 3 \times 3$	$(64 \times 4 \times 60 \times 80)$
concat3	-	$(576 \times 4 \times 60 \times 80)$
conv2c	$3 \times 3 \times 3$	$(512 \times 4 \times 60 \times 80)$
trans2	$2 \times 4 \times 4$	$(512 \times 8 \times 240 \times 320)$
conv2d	$3 \times 3 \times 3$	$(64 \times 8 \times 240 \times 320)$
concat2	-	$(576 \times 8 \times 240 \times 320)$
conv_pred	$3 \times 3 \times 3$	$(2 \times 8 \times 240 \times 320)$
loss	-	$(2 \times 8 \times 240 \times 320)$

- Pretrained on the Sports-1M dataset and fine-tuning on the DAVIS 2016 `train` subset (30 clips).
- **Data Augmentation:** creating additional versions of the 30 training clips that are rotated, flipped, and noised (120 total).
- **Post-Processing:** morphologically removing objects of 5500 connected pixels or less then filling in holes of 20000 connected pixels or less.
- **Input:** 8 consecutive frames, **Output:** 8 corresponding mask heatmaps
- Each pixel's value is the likelihood from 0-1 that it is the foreground object, and the values are thresholded to create the video object mask.

Architecture Visualization

Figure 1: Pipeline for the 3D Convolutional Encoder-Decoder Network Architecture



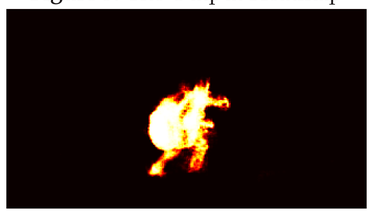
DAVIS 2016 Results

These results are on the DAVIS 2016 `val` subset, 480p resolution. Below is an example of an input frame and corresponding heatmap that the network returns, without any thresholding or post-processing. A brighter color means a larger value, indicating that the pixel is likely the foreground object.

Figure 2: An Input Frame



Figure 3: The Output Heatmap



In order to quantify the quality of our predicted masks, we calculate the Intersection over Union (IoU), also known as the *Jaccard Index* [1], which for a ground truth mask T and predicted mask P is defined as $\mathcal{J} = \frac{|T \cap P|}{|T \cup P|}$ [1]. The intersection and union are based on comparing pixels between the two masks, and the goal is to get as close to 1.0 (100%) as possible.

Table 2: Dataset Mean IoU [1]

Method	KEY	UCF-C3D	CVOS	TRC	SAL
mIoU	0.498	0.485	0.482	0.473	0.393

Figure 4: Training Loss Curve

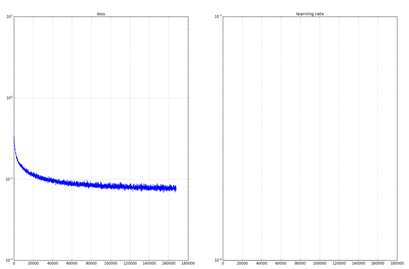


Table 3: Per-Clip Mean IoU

Clip	Our mIoU	KEY mIoU	Diff
blackswan	0.334	0.842	-0.508
bmx-trees	0.191	0.193	-0.002
breakdance	0.298	0.549	-0.251
camel	0.576	0.579	-0.003
car-roundabout	0.500	0.640	-0.140
car-shadow	0.783	0.589	0.194
cows	0.441	0.337	0.104
dance-twirl	0.404	0.380	0.024
dog	0.645	0.692	-0.047
drift-chicane	0.244	0.188	0.056
drift-straight	0.500	0.194	0.306
goat	0.593	0.705	-0.112
horsejump-high	0.590	0.370	0.220
kite-surf	0.571	0.585	-0.014
libby	0.410	0.611	-0.201
motocross-jump	0.599	0.288	0.311
paragliding-launch	0.578	0.559	0.019
parkour	0.519	0.410	0.109
scooter-black	0.379	0.502	-0.123
soapbox	0.549	0.757	-0.208

Visual Examples

Figure 5: Frame from `car-shadow`

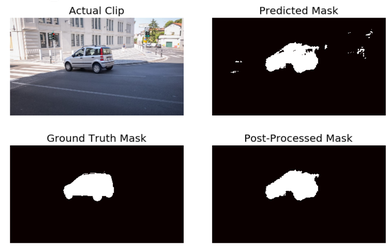


Figure 8: Frame from `goat`

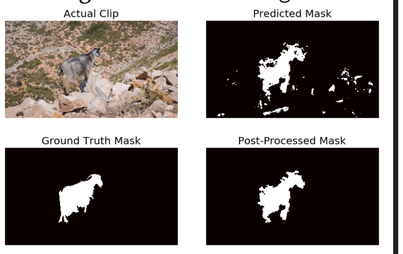


Figure 6: Frame from `dog`

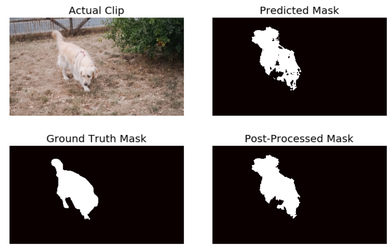


Figure 9: Frame from `motocross-jump`

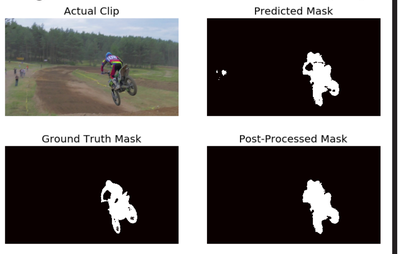


Figure 7: Frame from `camel`

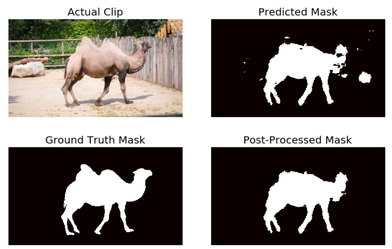
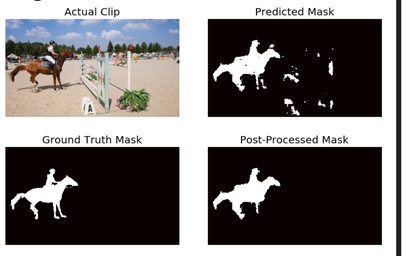


Figure 10: Frame from `horsejump-high`



Runtime

- The computer for evaluation has an `Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz` CPU with 36GB of RAM and an `NVIDIA Titan Xp` GPU.
- On average, it took 0.243 seconds to take an 8-frame clip and output the predicted heatmaps.

Upcoming Work

- Testing the effects of adding a PSPnet module [2] to the model.
- Evaluating on the SegTrack v2 and YouTube-Objects datasets.
- Changing foreground/background weights in the loss function to reflect the ratio of foreground/background pixels per clip.

References

[1] 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.

[2] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *CoRR*, abs/1612.01105, 2016.