Video Object Segmentation using Deep Learning

Zack While
Youngstown State University
zackwhile@outlook.com

Chen Chen
University of Central Florida
chenchen870713@gmail.com

Rui Hou
University of Central Florida
houray@gmail.com

Mubarak Shah
University of Central Florida
shah@crcv.ucf.edu

Abstract

In this paper, we present an end-to-end, three-dimensional convolutional neural network architecture that segments the foreground object in video. The three-dimensional encoder-decoder framework takes eight frames as input and returns eight corresponding heatmaps, classifying each pixel as the object or background in each frame. A greater pixel value represents a higher likelihood that it is the foreground object, and these values are thresholded to create the binary mask. This task incurs further challenges in addition to those present in image object segmentation, such as the main object possibly changing appearance or becoming partially obscured from view. We evaluate the model on the challenging DAVIS 2016 dataset, a common benchmark for video segmentation. Our results show promising competitiveness with the current state of the art.

1. Introduction

Identifying the predominant object per-pixel in each frame of a video is a well-researched problem in the computer vision community. In recent years, deep learning has become an important tool in state of the art solutions, training to extract discerning features by using large datasets. Segmentation provides more specified information than a bounding box, differentiating the object per-pixel and taking the shape of the target object. This motivates its use in fields such as robot vision, video surveillance, and content-based video retrieval.

In the course of a video clip, the main object may noticeably change or become partially hidden by part of the background. Also, varying amounts of camera shake could distort how a frame looks, causing phenomena like motion blur. All of these challenges are coupled with those already present in image segmentation, which creates a need for this type of framework to be very robust.

1.1. Related Works

An important distinction between operating on video as opposed to images is the introduction of temporal information. [8] found noticeable improvement in action recognition performance by using a three-dimensional convolutional network with $3 \times 3 \times 3$ convolutional kernels. This method allowed the network to reduce the representation of the input clips in a learned way, both spatially and temporally. Our belief is that this general approach will provide a similar amount of quality when performing video segmentation, combining spatial and temporal information.

[3] approached the problem of video segmentation with a two-stream architecture. One stream focused on learning appearance-base information while the other simultaneously focused on motion-based information, combining the feature maps of each stream at the end to output the mask prediction. One goal of our approach is to avoid splitting the spatial and temporal information, keeping all information together in one end-to-end stream.

[9] utilized a novel approach to maxpooling in their PSPNet architecture, which is designed for scene parsing. After obtaining the feature maps from the final convolutional layer, maxpooling with different scales was utilized to gain information from various representations of the feature map. These various sizes were put through a convolutional layer and upsampled with bilinear interpolation to become a uniform size, lastly concatenating the results to form the final set of feature maps. We consider this approach to maxpooling in the future work section.

2. Methods

2.1. Architecture Description

A visualization of our architecture can be found in **Figure 3**, and detailed information concerning the layers of the



(a) Input Frame **(b)** Output Heatmap

Figure 1: Example Input and Output Frame

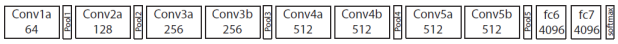


Figure 2: The C3D Pipeline [8]

model can be found in **Table 1**. The overall goal of this pipeline is to temporally and spatially shrink the representation of the eight input frames in a learned way, then rebuilding them in a learned way to create the output heatmaps, and an example for a single frame is shown in **Figure 1**. These output heatmaps are thresholded at a single value for the entire dataset.

The model is implemented in pyCaffe, pretrained with C3D's model on the Sports-1M dataset and fine-tuning on the DAVIS 2016 `train` subset [6]. The encoder portion of our pipeline uses the first eight 3D convolutions from C3D, as everything including and after the fully-connected layers relates to action recognition instead of object segmentation. These layers include 3D convolutions with $3 \times 3 \times 3$ kernels as well as ReLU activations and maxpools with $2 \times 2 \times 2$ kernel dimensions. The first maxpool has a notable $1 \times 2 \times 2$ kernel dimension, as we don't want to start losing the temporal information immediately at the beginning.

In addition to the pretrained layers from C3D, we also introduce the decoder structure, which provides additional 3D convolutions as well as convolution-transpose layers, which increase the size of the data to bring it back to its original dimensions. We also add skip-pooling layers, represented by green arrows in **Figure 3**, where we put the feature maps from the encoder through another 3D convolution and subsequently concatenate them with feature maps of the same size in the decoder. This provides more global information by combining features from two equivalent parts of the pipeline. The model ends with a final convolution followed by a softmax layer, which provides the final foreground/background prediction for each pixel, a likelihood of being the foreground object in the range $[0, 1]$.

2.2. Dataset

2.2.1 DAVIS 2016

The DAVIS 2016 dataset was created by Perazzi et al. [6] to bring forth a challenging video segmentation dataset with modern camera quality and pixel-level annotations for every frame. The dataset is composed of 50 total HD videos, divided into subsets of 30 training and 20 evaluation clips. The videos are all recorded at 24 frames per second, and the creators purposefully chose challenging clips that have attributes such as motion blur, occlusion, and scale variation. Their website provides an updated list of the best current results on the evaluation subset, categorized as semi-supervised (the first ground truth frame is provided during runtime) or unsupervised (no ground truth frames are provided.) There are 480p and full-resolution versions of each clips frames provided. The dataset provides $3,455$ total annotated frames and there are four major classes: *humans, animals, vehicles,* and *objects* [6]. The training subset has $2,079$ total frames.

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	$2 \times 2 \times 2$	$(64 \times 2 \times 30 \times 40)$
conv4d	$3 \times 3 \times 3$	$(64 \times 2 \times 30 \times 40)$
concat4	-	$(128 \times 2 \times 30 \times 40)$
conv3c	$3 \times 3 \times 3$	$(512 \times 2 \times 30 \times 40)$
trans3	$2 \times 2 \times 2$	$(64 \times 4 \times 60 \times 80)$
conv3d	$3 \times 3 \times 3$	$(64 \times 4 \times 60 \times 80)$
concat3	-	$(128 \times 4 \times 60 \times 80)$
conv2c	$3 \times 3 \times 3$	$(512 \times 4 \times 60 \times 80)$
trans2	$2 \times 4 \times 4$	$(16 \times 8 \times 240 \times 320)$
conv2d	$3 \times 3 \times 3$	$(64 \times 8 \times 240 \times 320)$
concat2	-	$(80 \times 8 \times 240 \times 320)$
conv_pred	$3 \times 3 \times 3$	$(2 \times 8 \times 240 \times 320)$
loss	-	$(2 \times 8 \times 240 \times 320)$

2.3. Data Augmentation

In order to increase the size of our training set, we added augmented versions of the original 30 training clips. These included a copy with salt and pepper noise added, a copy that was horizontal mirrored, and a copy that has the frames in the opposite order. These provided further examples to

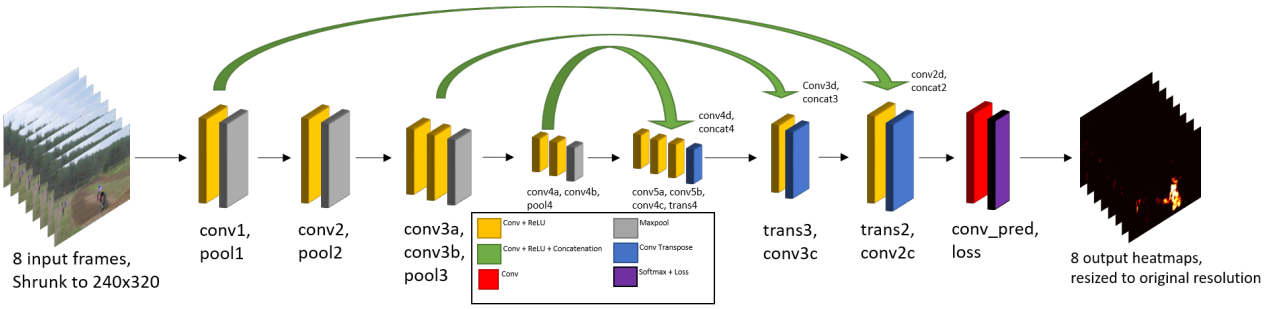


Figure 3: 3D Encoder-Decoder Architecture

train on that were slightly different from the original, and we found a slight improvement in our results with these added. This caused our training set to increase from 30 to 120 total clips.

2.4. Post-Processing

We additionally employed some morphological transformations to improve the quality of our output masks. After thresholding the initial prediction, we first remove all pixel objects that are composed of $3,500$ or less connected pixels. Then, we fill in any remaining white holes that have $15,000$ or more connected pixels. These transformations also provided noticeable improvement in overall segmentation performance; with that said, there are some drawbacks. Firstly, this approach to post-processing is biased toward larger objects, as any clip with a foreground object that is inherently smaller than $3,500$ pixels will be removed. Additionally, our main goal is for this to be a deep learning approach, and we ideally would like to achieve equivalent or better results without requiring post-processing.

2.5. Extended Dataset

A further method of improving the training set is providing additional diverse clips for training; in order to accomplish this, we combined two major datasets: JumpCut [2] and DAVIS 2017 [7]. The JumpCut dataset combines a few smaller datasets, for 22 clips with $6,334$ total frames. Clips include various animals, humans, non-moving objects, and fast-moving objects. DAVIS 2017 requires conversion to binary segmentation, as the challenge for that version provides multiple segmentation classes in the ground truth annotations. We converted all of the classes to a single class, restoring the dataset to binary segmentation. The training subset of DAVIS 2017 includes DAVIS 2016's 30 clips as well as 30 newly-annotated clips of similar length, 60 clips with a total of $4,209$ frames. We trained a separate copy of the same architecture on just this dataset for comparison with the augmented version.

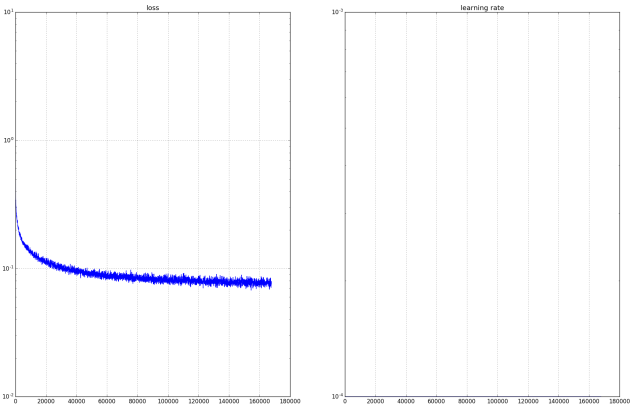


Figure 4: Augmented DAVIS 2016 Training Loss

3. Results

3.1. Training

We fined-tuned the model on the augmented DAVIS 2016 training subset for $186,000$ iterations, stopping it when the loss became noticeably flat for an extended period, pictured in **Figure 4**. The loss rate throughout was 0.0001.

The model was separately fine-tuned on the extended dataset for $147,000$ iterations as of now, and we aim to let it further train and hopefully provide improved results. The loss rate throughout was also 0.0001, and we expect it to continue training effectively for a bit more before flattening out.

3.2. 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 for the 480p DAVIS 2016 evaluation subset. For improved runtime speed, skipping the morphological transformations is recommended, as these can be highly time-consuming compared to the rest of the

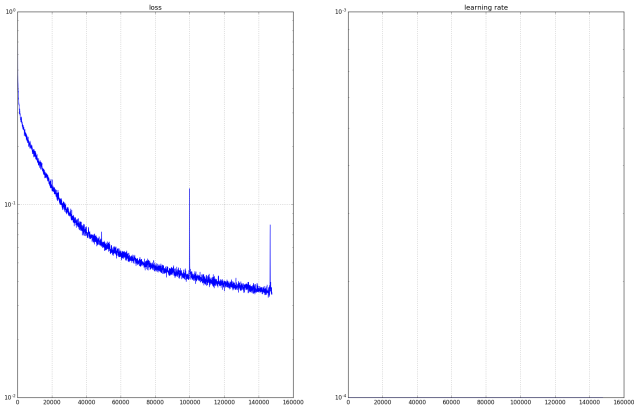


Figure 5: Extended Dataset Training Loss

model.

3.3. Benchmarking Method

In order to quantify the quality of our predicted masks, we calculate the Intersection over Union (IoU), also known as the *Jaccard Index* [6], which for a ground truth mask T and predicted mask P is defined as $\mathcal{J} = \frac{|T \cap P|}{|T \cup P|}$ [6]. The intersection and union are based on comparing pixels between the two masks, as the intersection is the number of foreground pixels the masks share and the union is the number of foreground pixels they have combined. The goal for this benchmark is to get as close to 1.0 (100%) as possible.

3.4. Discussion

We evaluated the model on the 30 DAVIS 2016 evaluation clips, comparing our mean intersection-over-union (mIOU) to some of the other unsupervised methods provided on the DAVIS challenge website [6], shown in **Table 2** and **Table 3**. These results show promise, as we plan on improving the architecture as well as the training set. Compared to [1], our best-performing model has noticeably better performance on clips such as `goat` and `motocross-jump` but noticeably worse performance on clips such as `cows` and `parkour`. Our worst performance occurred on the `bmx-trees`, which is understandable due to its multiple instances of occlusion and noticeable camera shake.

The mIoU increased each time that we added post-processing as well as when the dataset size was increased, which shows the positive impact of both practices. As said before, we aim to achieve similar or better results without the post-processing, which would keep our solution as one purely solved with deep learning as well. This would also eliminate the inherent biases that these post-processing methods can have on extremely small and large object sizes.



Figure 6: Example Results for `horsejump-high`

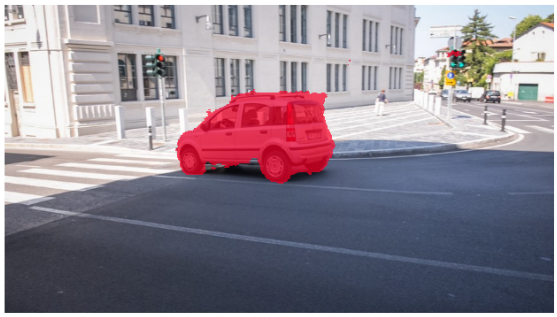


Figure 7: Example Results for `car-shadow`



Figure 8: Example Results for `goat`

4. Future Work

In the upcoming future, we plan to evaluate our model on the YouTube-Objects mask subset [4] as well as Segtrack-v2 [5], providing further comparison for our method to the state of the art. One concern for this evaluation is that we so far have trained on HD video clips, and both datasets have comparatively-lower resolution clips. Segtrack-v2 also requires conversion to binary masks, as some clips have additional foreground objects annotated in a separate copy of the frame. The YouTube-Objects mask subset has some clips

Table 2: Overall Mean Intersection-over-Union [6], where $_{E}$ indicates fine-tuning on the extended dataset of JumpCut and binary-converted DAVIS 2017 training subset and $_{P}$ indicates that post-processing was utilized. Additionally, $_{A}$ indicates training instead on the augmented DAVIS 2016 training subset and, for comparison, $_{T}$ indicates fine-tuning on the original DAVIS 2016 training subset.

Method	NLC	MSG	UCF-C3D$_{E,P}$	UCF-C3D$_{E}$	KEY	UCF-C3D$_{A,P}$	CVOS	TRC	UCF-C3D$_{A}$	UCF-C3D$_{T}$	SAL
mIoU	0.551	0.533	0.526	0.505	0.498	0.486	0.482	0.473	0.459	0.433	0.393

Table 3: Per-Clip Mean Intersection-over-Union [6] compared to NLC [1]

Clip	Our mIoU	NLC mIoU	Diff
blackswan	0.539	0.875	-0.336
bmx-trees	0.149	0.212	-0.063
breakdance	0.395	0.673	-0.278
camel	0.760	0.768	-0.008
car-roundabout	0.717	0.509	0.208
car-shadow	0.787	0.645	0.142
cows	0.281	0.883	-0.602
dance-twirl	0.444	0.347	0.097
dog	0.551	0.809	-0.258
drift-chicane	0.640	0.324	0.316
drift-straight	0.490	0.473	0.017
goat	0.550	0.010	0.540
horsejump-high	0.605	0.834	-0.229
kite-surf	0.593	0.453	0.140
libby	0.367	0.635	-0.268
motocross-jump	0.570	0.251	0.319
paragliding-launch	0.557	0.628	-0.071
parkour	0.502	0.901	-0.399
scooter-black	0.422	0.162	0.260
soapbox	0.594	0.634	-0.040

that are less than eight frames in length, which also limits our ability to fairly evaluate on the entire dataset.

We also aim to test whether the inclusion of multi-scale maxpooling in our pipeline will provide improved results, influenced by the Pyramid Pooling Module in [9]. While they utilized four different scaling factors, we plan to start by using the original maxpool factor and one that shrinks it twice as much. Since [9] used this technique for image segmentation, we will also need to test whether the multi-scale pooling is necessary in all three dimensions or if the temporal space does not require the additional scale of max-pooling. A visualization of this proposed method can be found by comparing **Figure 9** and **Figure 10**. They found greater success by using averagepooling instead of max-pooling, which is another method we can test as well.

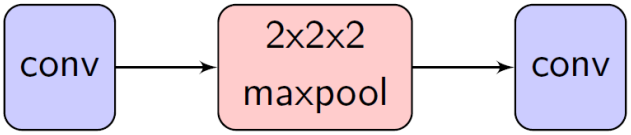


Figure 9: The Current Maxpooling in the Encoder

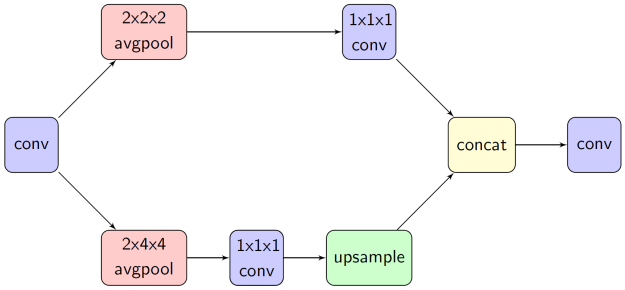


Figure 10: The Proposed Multi-Scale Maxpooling

References

[1] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[2] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen. Jumpcut: Non-successive mask transfer and interpolation for video cutout. *ACM Transactions on Graphics (Proceedings of SIGGRAPH ASIA 2015)*, 34(6), 2015.

[3] S. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. *CVPR*, 2017.

[4] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, 2014.

[5] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013.

[6] 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.

[7] 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.

[8] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[9] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.