Video Object Segmentation for CVPR 2018 WAD Video Segmentation Challenge Dataset

Simmi Mourya, Mohamed Suhail
University of Pennsylvania

Introduction: Our aim is to segment all the movable objects belonging in the WAD dataset as the objective of the challenge is to distinguish stationary and movable objects for the driver. (All the images are captured in reference with the driver's perspective). The objects include car, motorcycle, bicycle, pedestrian, truck, bus, and tricycle. We tried a couple of different approaches and finalized a slightly modified version of OSVOS or One-Shot Video Object Segmentation [1].

Dataset:
We predict segmentations of different movable objects appearing in the view of a car camera. This dataset contains a large number of segmented and original driving images. There are multiple labels but we only consider the following labels: car, motorcycle, bicycle, pedestrian, truck, bus, and tricycle. The corresponding groups, such as car group and bicycle group, are annotated when boundaries cannot be distinguished by annotaters. These groups are not evaluated currently.

The sample images and corresponding ground truths from the dataset:

Approach:
This paper tackles the task of semi-supervised video object segmentation (classification of all pixels of a video sequence into background and foreground), i.e., the separation of an object from the background in a video, given the mask of the first frame (one shot). It uses a full CNN architecture to transfer semantic information learned via ImagNet to the task of foreground segmentation. It also uses this information to the task of learning the appearance of a single annotated object of test video sequence.

First part: Adapt the CNN to an object instance given a single annotated image. For this, adapt pretrained image recognition CNN to video object segmentation by training it on a set of video frames with manually segmented objects.

Second part: OSVOS processes each frame of a video independently. Hence we do video object segmentation as per-frame segmentation.

The OSVOS model provides one annotated frame and the model can be finetuned in such a way that it has the option to go with a faster training time or more accurate results. Also, more frames can be added for annotation if the segmentation process is not up to the mark and the model will improve segmentation then.

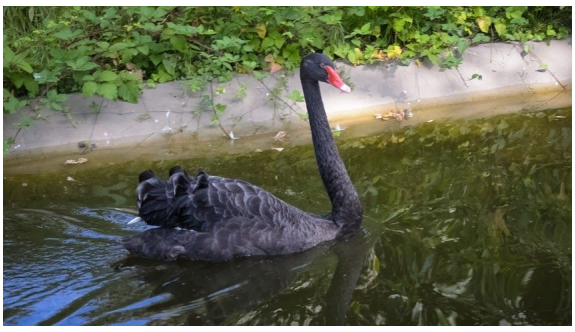
OSVOS Architecture:
The architecture of OSVOS as used in paper: total trainable parameters = 15,267,157.

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 480, 854]	1,792
ReLU-2	[-1, 64, 480, 854]	0
Conv2d-3	[-1, 64, 480, 854]	36,928
ReLU-4	[-1, 64, 480, 854]	0
MaxPool2d-5	[-1, 64, 240, 427]	0
Conv2d-6	[-1, 128, 240, 427]	73,856
ReLU-7	[-1, 128, 240, 427]	0
Conv2d-8	[-1, 128, 240, 427]	147,584
ReLU-9	[-1, 128, 240, 427]	0
Conv2d-10	[-1, 16, 240, 427]	18,448
ConvTranspose2d-11	[-1, 16, 482, 856]	4,096
Conv2d-12	[-1, 1, 240, 427]	17
ConvTranspose2d-13	[-1, 1, 482, 856]	16
MaxPool2d-14	[-1, 128, 120, 214]	0
Conv2d-15	[-1, 256, 120, 214]	295,168
ReLU-16	[-1, 256, 120, 214]	0
Conv2d-17	[-1, 256, 120, 214]	590,080
ReLU-18	[-1, 256, 120, 214]	0
Conv2d-19	[-1, 256, 120, 214]	590,080
ReLU-20	[-1, 256, 120, 214]	0
Conv2d-21	[-1, 16, 120, 214]	36,880
ConvTranspose2d-22	[-1, 16, 484, 860]	16,384
Conv2d-23	[-1, 1, 120, 214]	17
ConvTranspose2d-24	[-1, 1, 484, 860]	64
MaxPool2d-25	[-1, 256, 60, 107]	0
Conv2d-26	[-1, 512, 60, 107]	1,180,160
ReLU-27	[-1, 512, 60, 107]	0
Conv2d-28	[-1, 512, 60, 107]	2,359,808
ReLU-29	[-1, 512, 60, 107]	0
Conv2d-30	[-1, 512, 60, 107]	2,359,808

```
            ReLU-31         [-1, 512, 60, 107]               0
          Conv2d-32          [-1, 16, 60, 107]          73,744
  ConvTranspose2d-33         [-1, 16, 488, 864]          65,536
          Conv2d-34           [-1, 1, 60, 107]              17
  ConvTranspose2d-35          [-1, 1, 488, 864]             256
       MaxPool2d-36          [-1, 512, 30, 54]               0
          Conv2d-37          [-1, 512, 30, 54]       2,359,808
            ReLU-38          [-1, 512, 30, 54]               0
          Conv2d-39          [-1, 512, 30, 54]       2,359,808
            ReLU-40          [-1, 512, 30, 54]               0
          Conv2d-41          [-1, 512, 30, 54]       2,359,808
            ReLU-42          [-1, 512, 30, 54]               0
          Conv2d-43          [-1, 16, 30, 54]           73,744
  ConvTranspose2d-44         [-1, 16, 496, 880]         262,144
          Conv2d-45           [-1, 1, 30, 54]              17
  ConvTranspose2d-46          [-1, 1, 496, 880]           1,024
          Conv2d-47          [-1, 1, 480, 854]              65
================================================================
Total params: 15,267,157
Trainable params: 15,267,157
Non-trainable params: 0
----------------------------------------------------------------
```

Training:

1. Due to lack of GPU resources, we were only able to use 2 full video sequences from the WAD dataset for training: 170908_Camera_5 and 170908_Camera_6 and tested on a held out subset of video sequence 170908_Camera_5.
2. We used the weights of the CNN model pre-trained on Image-Net.
3. Then, to see how well the model was able to perform, we took a small subfolder of the DAVIS 2016 dataset (Black Swan) which consisted of the binary masks for the training images to make the model learn an idea to segment objects from the background.



Results on Black Swan class trained for 100 epochs

4. ADAM optimizer with weight decay of 1e-5, learning rate of 1e-5 and Cross-entropy loss as our loss function were used as our hyperparameters.
5. Using the same hyperparameters, we tried the OSVOS model on the WAD dataset for 1600 epochs. All the training images and the corresponding annotations were resized to a resolution of 846*678 pixels to adapt with the Google Colab RAM & GPU capacity.

Results:

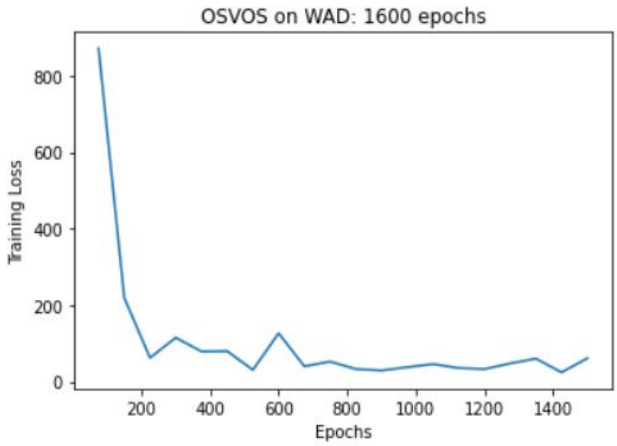
1.We can see from the below sample images that only moving objects (cars) are segmented while stationary elements (trees,poles) aren't recognized thus distinguishing only the moving objects for the driver.




Results on WAD dataset trained for 1600 epochs







References:

[1] Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D. and Van Gool, L., 2017. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 221-230).

Future Work: We have requested and obtained the Mapillary Vistas dataset from the Mapillary website and we are planning to work on it this Summer. We have also signed in to obtain AWS Educate credits to benefit from more computational power and reduce training time.