Topic J - STCN Video Segmentation

Monday 28th February, 2022

Timothée GUY
École normale supérieure Paris-Saclay
timotheeguy1@gmail.com

David SOTO
École Normale Supérieure Paris-Saclay
david.soto@ens-paris-saclay.fr

Abstract

In this report, we present an analysis of the Space-Time Correspondence Network *algorithm for segmenting video sequences from first frame annotation. We propose a variation of the inference step, by automatically annotating the first frame with* Mask R-CNN *and* PointRend *image detection methods. We find that on major part of the DAVIS video sequences dataset, we achieve close performances, but on some specific video sequences, we miss the detection of the object of interest.*

1. Introduction

Following the success of the COCO [4] data challenges for detection and segmentation of images, datasets for videos have been developed. Indeed, advanced video processing has become of utmost importance for disruptive technologies from autonomous driving to action and human posture recognition and understanding

The main focus of this project will be the task of *Video Object Segmentation (VOS)*, which aims to segment particular objects of interest with respect to background in video sequences. One algorithm of interest is the *Space Time Correspondence Network (STCN)* [1], released in 2021, which achieves state-of-the-art (SOTA) results on VOS challenge datasets: ranked second place in DAVIS-2017 [7] and Youtube-VOS [9].

Our project contribution will be divided in three studies:

- We reproduce results of STCN on the DAVIS 2017 dataset
- We check the generalization capacity of STCN by testing VOS on the *YouTube-BoundingBoxes Dataset (YT-BB)* [8] and *Something-else Dataset*
- We propose a method to automatically segment the first frame of the sequence by using two existing image segmentation algorithms, Mask R-CNN [2] and PointRend [3]

2. The STCN model

The STCN is derived from the more general class of *Space-Time Memory networks (STM)* [5] which solves the VOS task in the semi-supervised configuration where the first frame segmentation is given as input. This first segmentation is propagated through the video by the network with the two steps:

- Generation of segmentation on the next frame (query) by computing affinity between key features of query frame and key features of memory frame.
- Saving query key and value vectors to memory to ensure propagation of predictions forward in time.

The STCN is a simplified but improved version of the STM. Among changes, the affinity for STCN is computed between images only, the L2 similarity is chosen as the affinity function and fewer frames are stored in memory [1]. This lighter version gives better performance on a wide range of datasets as well as more speed at inference time.

3. The DAVIS 2017 dataset for VOS

Most of the experiments are made on the *Densely-Annotated VIdeo Segmentation (DAVIS)* 2017 VOS dataset, which consist of 150 annotated short videos sequences. A batch of example frames with their ground-truth segmentation is displayed in the appendix (Figures 8, 9, 10, 11, 12, 13).

	train	val
Nb of sequences	60	30
Nb of frames	4219	2023
Mean nb of frames per sequence	70.3	67.4
Nb of objects	138	59
Mean nb of objects per sequence	2.3	1.97

Table 1. **DAVIS train and validation set composition.** [7]

We will take for the evaluation metrics the *region similarity* $\mathcal{J}$ [6], which is the IoU between estimated segmentation and ground-truth mask, and the *contour accuracy* $\mathcal{F}$ [6], which is the F-score of contour matching.

4. Experiment

Part of the project involved doing some experiments in order to check the generalization capacity of STCN.

4.1. Replicating the paper's results

In this section, we replicate a set of results from the STCN paper in order to corroborate some of the paper's results. For this part of the project, we chose to work on the DAVIS dataset and therefore to replicate the results on the DAVIS dataset.

For this part, we used the provided implementation of the STCN paper[1]. For the visualization of the results, we created a code which basically overlays the image segmentation masks obtained with the STCN model and the original images[2]. This is how we obtain qualitative results. Figure 1 displays some examples of qualitative results we obtained on DAVIS dataset.

Moreover, we also obtained quantitative results for the DAVIS dataset. For this part, we used the code from the davis2017-evaluation repository[3] in order to calculate the standard metrics $\mathcal{J}$ and $\mathcal{F}$. Table 2 displays the quantitative results we obtained for the DAVIS dataset.

Figure 1. Examples of qualitative results that we obtained for DAVIS dataset

Method	$\mathcal{J}$ mean	$\mathcal{F}$ mean	$\mathcal{J}\&\mathcal{F}$ mean
Manual	0.820	0.886	0.853

Table 2. Our quantitative results for the DAVIS dataset.

[1]The STCN repository is available at: https://github.com/hkchengrex/STCN

[2]The notebooks used for results visualizations are available in the appendix

[3]The davis2017-evaluation library is available at: https://github.com/davisvideochallenge/davis2017-evaluation

4.2. Application of the STCN model on a new dataset

In this section, we will apply the STCN model on new datasets, that is a datasets that were not used in the STCN paper.

4.2.1 Youtube-BB dataset

Youtube-BB is a dataset containing approximately 380,000 videos segments extracted from 240,000 different publicly visible YouTube videos. Each video of Youtube-BB is more or less 16 seconds long. We extracted some videos from this dataset and applied the STCN model on them. We manually annotated the first frame of each video we evaluated and let the model do the inferences. For this part we used the python code `eval_generic.py` of the STCN paper. Once the inferences were extracted, we applied our code for visualization. Our code overlays the inferences with the original images and this allows us to have qualitative results. Figure 2 displays some examples of qualitative results we obtained for Youtube-BB. The qualitative

Figure 2. Example of qualitative results obtained for Youtube-BB

results obtained show that the STCN model works well on Youtube-BB videos.

4.2.2 Something-else dataset

Our plan for this part of the project was to use the something-something dataset but since the latter's website was down, we decided to work on the something-else dataset[4]. This dataset was created and used for the paper *"Something-Else: Compositional Action Recognition with Spatial-Temporal Interaction Networks"*. The something-else dataset contains more than 180,000 videos, all extracted from the something-something dataset.

For the implementation of the STCN model, we started by manually annotate the first frame of each video we evaluated and then we used the STCN model for the inferences. Once we had the inferences, we used our visualization code in order to obtain qualitative results. Some of our results are displayed in Figure 4. The qualitative results obtained show that the STCN model works well on something-else videos.

[4]The Something-else repository is available at: https://github.com/joaanna/something_else

4.3. Automating the segmentation of the first frame

4.3.1 Principle

The idea of this experiment is to apply a SOTA method to segment the first image, needed by STCN to propagate annotations. However, two challenges are raised by this approach. Firstly, *what to do if the objects of interest are not detected on the video ?* Secondly, the image detection and segmentation algorithm can output many objects. *How to choose among them the real targets ?*

We focus on this project on the study of performance variation between manual segmentation and automatic segmentation. The detection of objects of interest which is needed to move from semi-supervised VOS to fully unsupervised require a whole class of dedicated detection algorithms on the whole video sequence. The workflow will be the following :

- Loop through the DAVIS video sequences dataset and infer first image segmentation with a method.
  - Manually select and assign labels to segmentations that correspond to ground-truth objects of interest.
  - Ground-truth objects that are not detected **are not** manually annotated.
- Apply STCN to video sequence, evaluate average $\mathcal{J}$ and $\mathcal{F}$ and compare with the manual STCN baseline.

4.3.2 Implementation

We choose Mask R-CNN [2] and PointRend [3] to test the segmentation of first image. For the implementation of those models, we adapt code from detectron2 library notebooks[5] to infer on first image of each video sequence and assign labels to each object detected. The notebooks used for model inference but also for results visualizations are available in the appendix.

As can be seen in Figure 6, the image detection and segmentation algorithm ouputs many objects which are not existing in the DAVIS validation set. The definition of an object of interest can be quite ambiguous (in Figure 6, there are many motorbikes detected, but only one is the target). So we select manually the correct objects and assign them labels. In some images, an object can be missed (Figure 11) or detected partially (Figure 13). This shows that the detection phase is sensitive to the classifier associated with the trained image detection model. For some particular classes, it may be necessary to retrain the image classifier algorithm to allow for the detection of specific concepts.

[5]The detectron2 library is available at: https://github.com/facebookresearch/detectron2

4.3.3 Comparison between Mask R-CNN and PointRend

An intuitive idea consists in comparing the overall quality of segmentation on the first image between the three approaches (Manual, Mask R-CNN and PointRend) to extrapolate to the quality of the segmentation on the whole video.

We see in Figure 8, Figure 9, Figure 10 the abilities of Mask R-CNN and PointRend with respect to ground-truth segmentation. We recognize the advantage of PointRend (which is built on Mask R-CNN with a modified architectural head [3]) over Mask R-CNN for sharp edges. Those examples allow us to be confident for the results obtained with PointRend.

4.3.4 Performance on DAVIS validation set

We perform STCN with this time the first frame obtained by the image segmentation methods. Then we use the official DAVIS repository for evaluation of VOS performance[6].

Method	$\mathcal{J}\&\mathcal{F}$ Mean	$\mathcal{J}$ Mean	$\mathcal{F}$ Mean
Manual	0.853	0.820	0.886
Mask R-CNN	0.689	0.667	0.712
PointRend	0.711	0.689	0.733

Table 3. **Global performance of video object segmentation on DAVIS validation dataset**, averaged on all sequences.

We see that we have a drop in performance, more than 10% for the two methods. To understand those results, we must analyse how segmentation is carried out on each sequence, in Table 4. We see that on some sequences, such as *libby*, we are very close to manual results, especially for PointRend (Figure 14). So, the PointRend approach is able to give very good results. The main drawback is shown by the *gold-fish_5* instance, which shows a score of 0. Indeed, PointRend has not detected the fifth fish. But if we set apart those cases, the PointRend-STCN method gives very good results and minor worsening.

5. Conclusion

The automatic segmentation of first frame for semi-supervised VOS is a valuable idea as it reduces the workload of a human made mask. We have shown that with a PointRend approach, we achieve similar results on some sequences as long as the objects present a medium complexity and are detected by PointRend. So, we still need some human interaction and it is not fully unsupervised, but we have gone from manual segmentation of image to selection of objects of interest, which is a big improvement.

[6]The evaluation API is available at : https://github.com/davisvideochallenge/davis2017-evaluation

References

[1] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation, 2021. 1

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018. 1, 3

[3] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering, 2020. 1, 3

[4] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 1

[5] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks, 2019. 1

[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 *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 724–732, 2016. 1

[7] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation, 2018. 1

[8] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video, 2017. 1

[9] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark, 2018. 1

A. Code implementation

For the coding part of the two segmentation algorithms, Mask R-CNN and PointRend (for inference), and for the visualization notebook :

- **Mask R-CNN**[7]
- **PointRend**[8]
- **Visualization**[9]

For the STCN model, we use the github repository from Ho Kei Cheng[10], and for the evaluation part on DAVIS dataset, we use the official repository for DAVIS 2017[11].

[7] https://colab.research.google.com/drive/1RwTU5wZESsvv2Afzfd7aMkNsX6v-5REv?usp=sharing

[8] https://colab.research.google.com/drive/16JuK5eCthjFpRDIatbBYV-sQ3ufU1ccR?usp=sharing

[9] https://colab.research.google.com/drive/1z11Xn5IxKm3pc3j3yEbleTLAO7Alerxa?usp=sharing

[10] https://github.com/hkchengrex/STCN

[11] https://github.com/davisvideochallenge/davis2017-evaluation

B. Qualitative Results on YouTube-BoundingBoxes

Figure 3. More examples of qualitative results obtained for Youtube-BB dataset

C. Qualitative Results on Something-else

Figure 4. Example of qualitative results obtained for the Something-else dataset

D. Performance results per sequence

Sequence	$\mathcal{J}\&\mathcal{F}$ Mean		
	Manual	**Mask R-CNN**	**PointRend**
bike-packing_1	0.833	0.606	0.648
bike-packing_2	0.914	0.875	0.892
blackswan_1	0.980	0.925	0.966
bmx-trees_1	0.742	0.624	0.628
bmx-trees_2	0.856	0.843	0.850
breakdance_1	0.939	0.915	0.914
camel_1	0.983	0.939	0.966
car-roundabout_1	0.987	0.979	0.979
car-shadow_1	0.985	0.981	0.982
cows_1	0.973	0.943	0.955
dance-twirl_1	0.936	0.910	0.914
dog_1	0.972	0.936	0.944
dogs-jump_1	0.907	0.907	0.904
dogs-jump_2	0.935	0.932	0.932
dogs-jump_3	0.967	0.956	0.964
drift-chicane_1	0.905	0.888	0.878
drift-straight_1	0.931	0.929	0.930
goat_1	0.940	0.922	0.931
gold-fish_1	0.876	0.564	0.686
gold-fish_2	0.895	0.749	0.786
gold-fish_3	0.928	0.000	0.832
gold-fish_4	0.948	0.818	0.813
gold-fish_5	0.935	0.000	0.000
horsejump-high_1	0.937	0.893	0.926
horsejump-high_2	0.915	0.886	0.908
india_1	0.924	0.886	0.913
india_2	0.750	0.724	0.724
india_3	0.872	0.809	0.820
judo_1	0.894	0.836	0.862
judo_2	0.852	0.796	0.811
kite-surf_1	0.738	0.000	0.000
kite-surf_2	0.553	0.556	0.543
kite-surf_3	0.894	0.799	0.795
libby_1	0.958	0.922	0.945
loading_1	0.981	0.961	0.976
loading_2	0.676	0.001	0.002
loading_3	0.966	0.840	0.877
mbike-trick_1	0.874	0.844	0.840
mbike-trick_2	0.794	0.841	0.865
motocross-jump_1	0.831	0.626	0.552
motocross-jump_2	0.850	0.799	0.741
parkour_1	0.971	0.965	0.967
pigs_1	0.955	0.950	0.953
pigs_2	0.874	0.852	0.866
pigs_3	0.952	0.919	0.925
scooter-black_1	0.857	0.874	0.862
scooter-black_2	0.896	0.880	0.876

Table 4. **Performance per sequence of video object segmentation on DAVIS validation dataset.** Metrics are averaged on all frames of a given video sequence. *gold-fish_3* correponds to the third instance of the *gold-fish* sequence.

E. STCN Limitation

Figure 5. **From left to right, STCN inference with manual annotation on first frame and ground-truth.** A part of the green dog is moved to the red dog. This problem is intrinsic of STCN, as it does not check for spatio-temporal coherency whithin the image.

F. Segmentation of the first frame

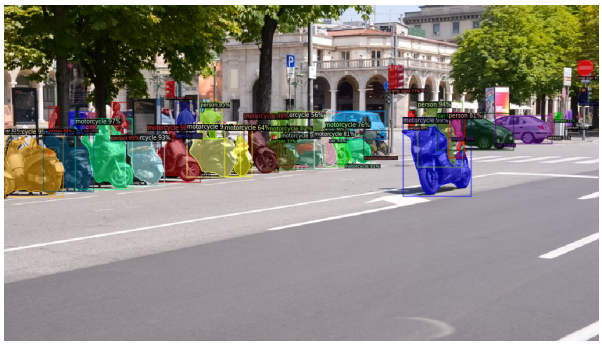

Figure 6. **Mask R-CNN inference on first frame of *scooter-black* video sequence.** The objects of interest of the scene are the dark blue motorbike moving and its biker.

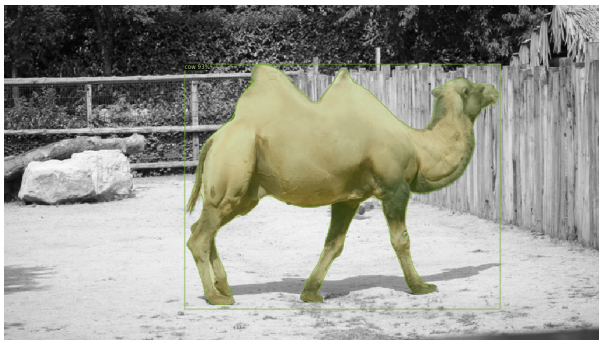

Figure 7. **PointRend inference for *camel* video sequence.** The camel is detected as a cow but the segmentation is still precise.

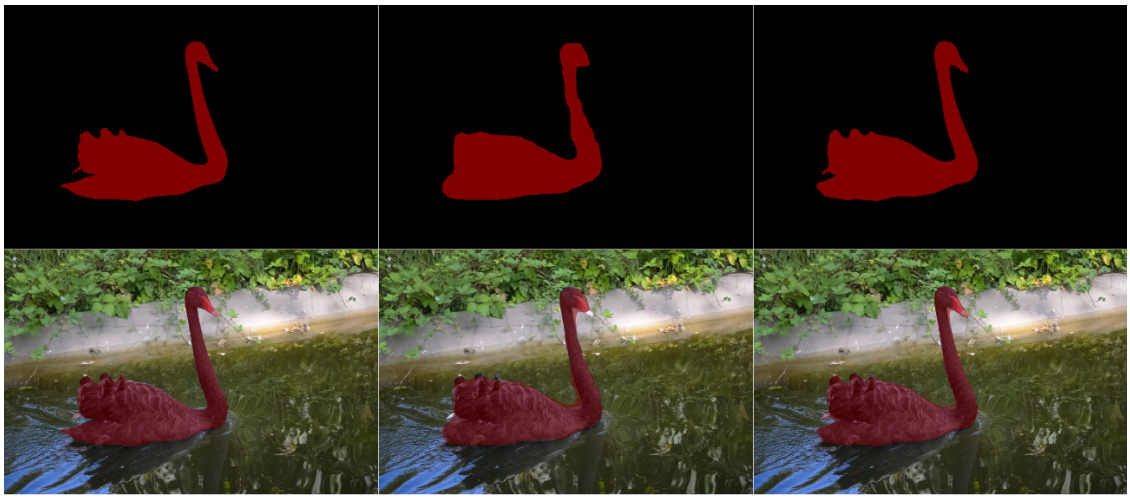

Figure 8. **First frame segmentation. From left to right : Manual, Mask R-CNN, PointRend.** We see that Mask R-CNN lacks capacity to match sharp edges while PointRend is very good on this point.

Figure 9. **Same configuration for another sequence.**

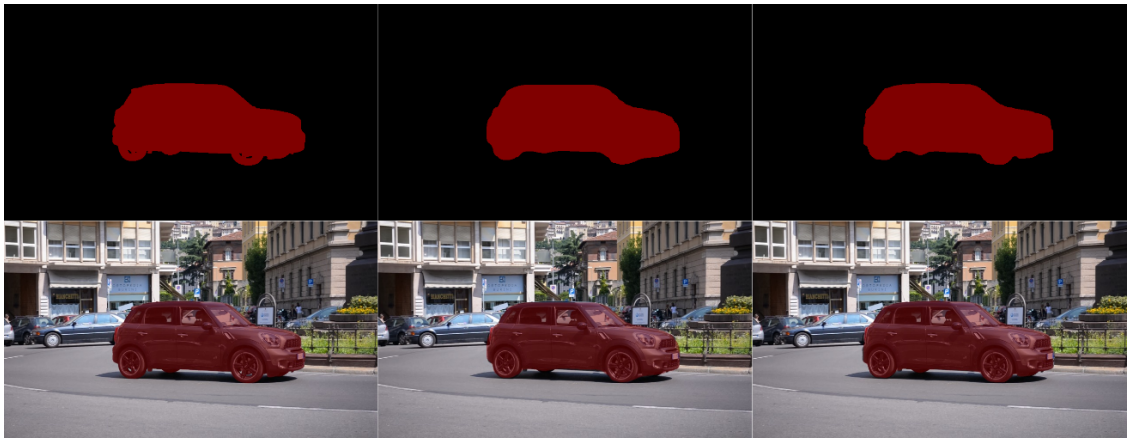

Figure 10. **Same configuration for the *car-roundabout* sequence.** Even the PointRend model is not able to detect the holes in the car rims.

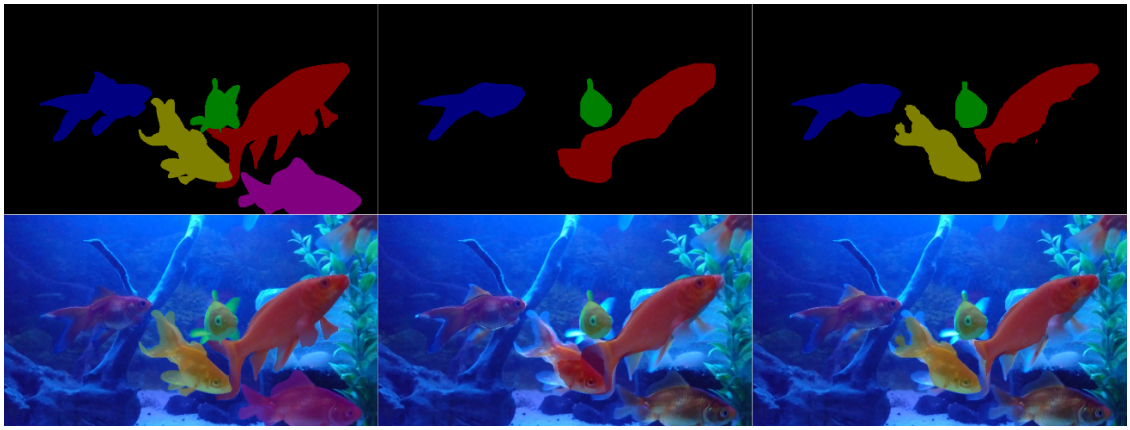

Figure 11. **Same configuration for the *gold-fish* sequence.** Mask R-CNN misses two fishes and mixes one with another, while PointRend misses only one.

Figure 12. **Same configuration for the *kite-surf* sequence.** Both Mask R-CNN and PointRend do not detect harness and cables.

Figure 13. **Same configuration for the *india* sequence.** Bags are detected separatly for Mask R-CNN and PointRend but not for STCN.

Figure 14. **Top sequence : STCN with manual annotation; Bottom sequence : PointRend segmentation of first frame; Sequence : *libby*.** We manage to propagate the dog mask, even with some occultation occurences. We can check in Table 4 that we achieve nearly the baseline score with PointRend method.

Figure 15. **Top sequence : STCN with manual annotation; Bottom sequence : PointRend segmentation of first frame; Sequence : *horsejump-high*.** Here again (Table 4), PointRend shows great success.