SHLS: Superfeatures Learned From Still Images For Self-supervised VOS - Supplementary Material

Marcelo Mendonça[1,2]
marceloms@ufba.br
Jefferson Fontinele[1]
jeffersonfs@ufba.br
Luciano Oliveira[1]
lrebouca@ufba.br

[1] Intelligent Vision Research Lab
Federal University of Bahia

[2] Federal Institute of Education, Science and Technology of Bahia - IFBA
Bahia, Brazil

1 Object mask estimation and pseudo-sequence synthesis

To obtain pseudo-masks, our underlying assumption is that image regions with high saliency response also present high objectness. We employ a learning-based saliency detector [5] that is trained in a self-supervised manner. Figure 1 shows saliency maps estimated for images from the MSRA10K [1] dataset. Moving from left to right in the figure, we observe the following issues: (i) certain foreground objects, such as the boy's bike, may go undetected; (ii) in the case of multiple objects, such as the individuals, the saliency map may not distinguish between them; (iii) object sub-parts often appear rounded and fused together, as seen in the petals of the flower; (iv) saliency maps generated from background-only images, such as the window, tend to be diffuse; and (v) detection failures are more likely to occur, particularly when the foreground is not centered in the image, as in the case of the dog.

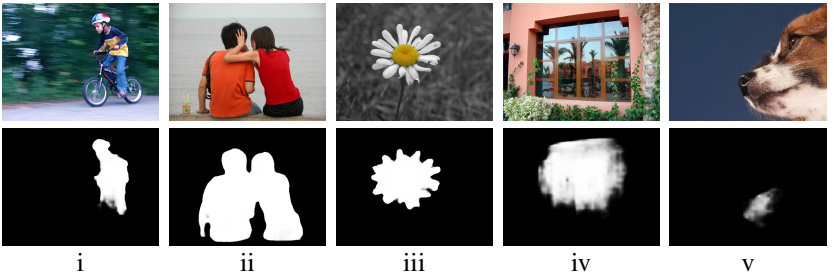


Figure 1: Saliency detection by applying [5] on images from MSRA10K [1]. The examples show issues that eventually occur in the detection process: (i) object not detected (bike); (ii) multiple objects not distinguished from each other (persons); (iii) object sub-parts rounded and glued together (flower petals); (iv) diffuse saliency map generated for background-only image (window); and (v) detection failure due to object not centralized (dog).

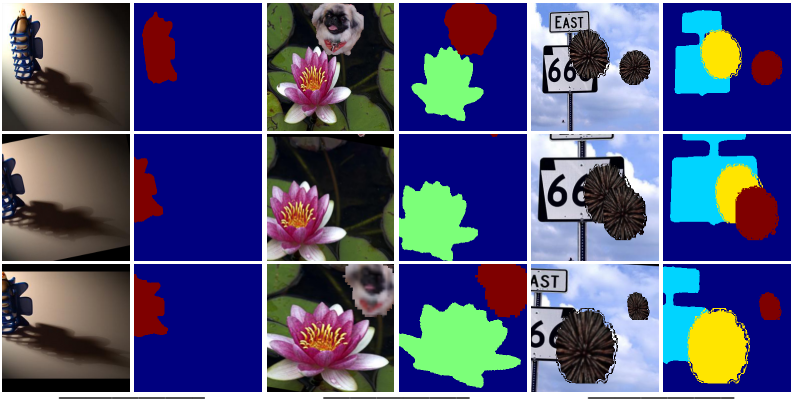


Figure 2: Examples of generated pseudo-sequences. The sequences are comprised of pairs of pseudo-frames and pseudo-masks containing a variable number of objects in different conditions: (a) single-object sequence with partial disappearance; (b) multi-object sequence with total disappearance and reappearance situation; and (c) multi-object sequence with a cloned foreground object.

We treat those issues by discarding images with a saliency response that is too small or completely empty. The other minor issues illustrated in Fig. 1 are less crucial since our approach involves extracting any detected foreground to create a pseudo-frame. In these cases, whether sub-parts of an object are misidentified or multiple objects are combined, they will be considered distinct and complete objects in the resulting composite frame.

Figure 2 displays some frame-mask pair sequences from our pseudo-sequence generation process, showcasing the different data augmentation techniques we have implemented in our algorithm. The sequences are based on the images from the MSRA10K dataset [1], which contains 10,000 images. With our methodology, we can create an unlimited number of pseudo-sequences to train our VOS method in a self-supervised manner.

2 Feature extractor architecture

Our feature extractor is a modified version of ResNet-18 [2]. The specific configuration is outlined in Table 1, which consists of five primary layers. The first layer is a simple convolution, while the remaining layers are residual blocks, each comprising a convolution, batch normalization, ReLU non-linearity, another convolution, and batch normalization. To generate the superfeatures, the outputs of layers 2 and 5 (with 1/2 and 1/4 spatial sizes, respectively) feed our superfeature embedding model.

3 Details on the segmentation refinement module

The proposed segmentation refinement module is designed as a single-object module. This means that, although our VOS method is intended for multi-object segmentation, during the segmentation refinement, the task is divided into a series of single-object segmentations. The goal is to allow for this module to learn a simpler, more specific task, with support from the

Input: $H \times W \times 3$ (RGB image)			
Stage	Type	Output size	
		Spatial	Depth
layer 1	convolutional	$H/2 \times W/2$	64 channels
layer 2	residual block	$H/2 \times W/2$	64 channels
layer 3	residual block	$H/4 \times W/4$	128 channels
layer 4	residual block	$H/4 \times W/4$	256 channels
layer 5	residual block	$H/4 \times W/4$	256 channels

Table 1: Feature extractor configuration. This network is a modified version of the ResNet-18 [7] to enlarge the spatial size of the output feature map. The architecture includes five layers, with the first layer comprising a single convolution. The remaining layers are residual blocks, each one formed by the sequence: convolution, batch normalization, rectified linear units (ReLU), convolution and batch normalization again.

previous modules. As depicted in Figure 3, the components of the module comprise three main stages: ROI selection, feature modulator and feature decoder.

ROI selection: For each object, the ROI obtained from the pre-segmentation mask is used to select the inputs to feed the feature modulator stage. As shown in Figure 3 (top), the ROI selection process includes two possible pathways: in case of inputs with unchanged spatial dimensions, we simply select the ROI and resize it to the target dimensions; in case of down-sampled inputs, we apply the ROI align function [4].

Feature modulator: This stage receives ROI-based feature maps from the feature extractor, specifically the L1 and L4 feature maps, as well as attention maps from the memory clustering. The feature modulator is then fed with features and attention maps from the current frame, along with the last segmented frame in the sequence. This facilitates the network's ability to learn to segment the current frame smoothly based on the previous segmentation. As depicted in Figure 3 (middle), the feature modulator architecture includes a branch with the softmax function, serving as a gate to modulate features from the parallel branch. The resulting feature map, denoted as M4 in the figure, is subsequently passed to the next stage, the feature decoder.

Feature decoder: The feature decoder, as shown in Figure 3 (bottom), utilizes refinement blocks [6] to integrate features from branches at different scales. The first merged branch combines the mask prediction of the previous frame with the concatenated L1 feature maps and mask annotation of the first frame. The second merged branch incorporates the L4 features from the first frame and the modulated features from the modulator stage. Prior to entering the first refinement block, these features undergo convolutions and residual blocks [3] to adjust the channel count. A second refinement block is then employed to establish skip connections by incorporating previously used inputs, including the L1 features of the first frame and the attention maps of the current frame. Finally, the output of this refinement block passes through a final convolution layer, generating a 2-channel object prediction.

Figure 4 compares ground-truth annotations (red) with pre-segmentations (blue) and refined masks (green) generated by our method from some frames of the DAVIS-2017 dataset [7]. As can be seen, the refined segmentations present a more regular object mask, with smoother contours that resemble the shape delineated in the ground-truth more closely.

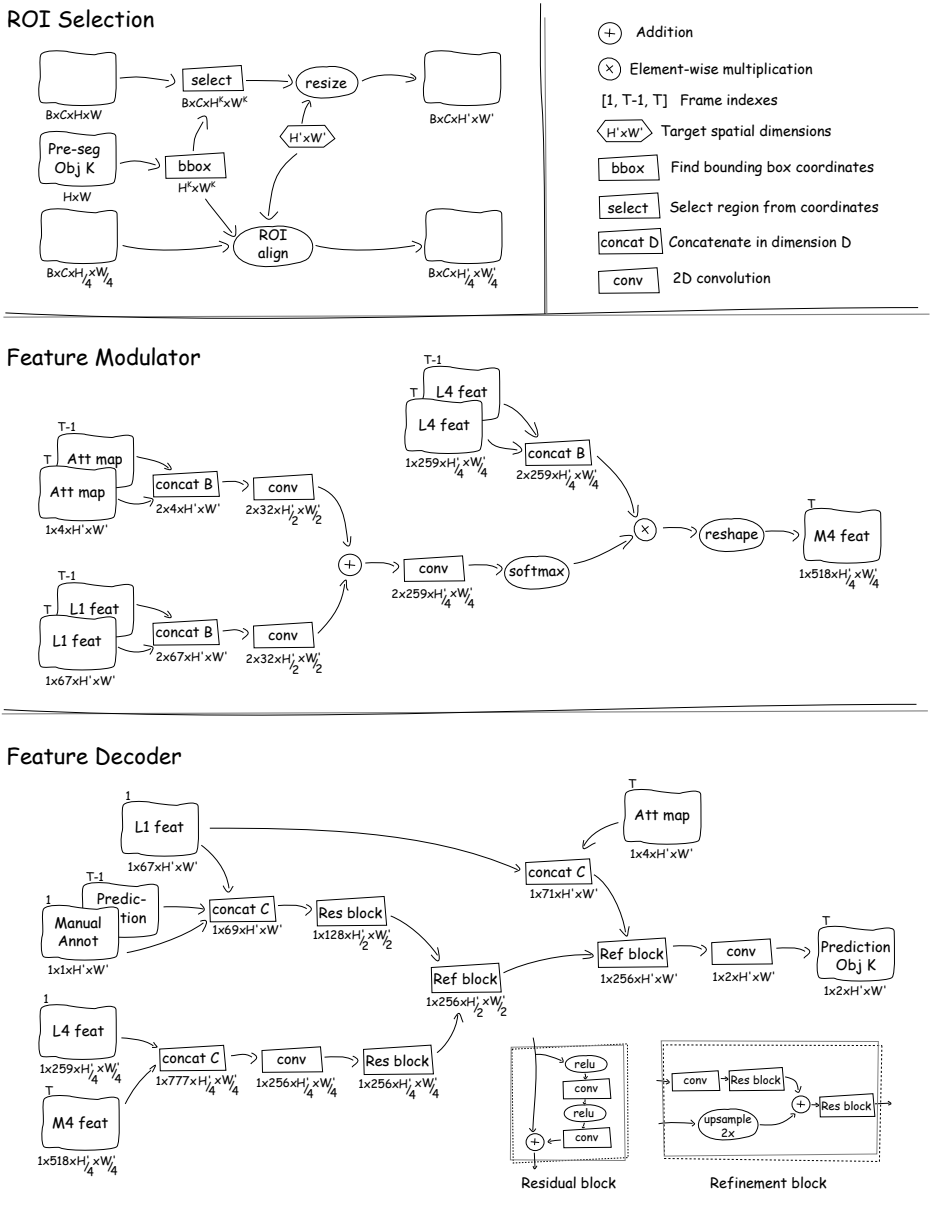
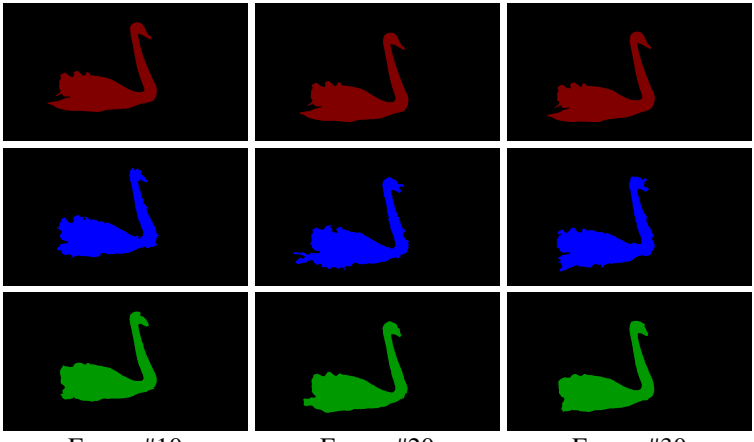


Figure 3: Segmentation refinement module. The ROI selection stage (top) provides two possible pathways: for inputs with unchanged scale, the ROI is selected and the height and width are resized to the target dimensions; for down-sampled inputs, the ROI align function [4] is applied. The feature modulator stage (middle) combines the L1 features and the attention maps in a softmax-based gate mechanism that modulates the L4 features. Finally, the feature decoder stage (bottom) relies on residual and refinement blocks to reduce the channels and increase the spatial dimensions of the features in order to predict the object masks.



Frame #10 Frame #20 Frame #30

Figure 4: Comparison between ground-truth and generated masks. The top row displays the ground-truth masks (red) for the object in the frames number 10, 20 and 30 of a video sequence from DAVIS-2017 [7]. The middle row displays the pre-segmentation (blue) obtained by classifying the superpixels directly. The bottom row displays the mask predictions (green) produced by the segmentation refinement module operating pixel-wisely.

4 Further analysis of the memory clustering

Figure 5 illustrates the behavior of our model over time in terms of *segmentation self-consistency*. We have formulated this metric to measure the variation in segmentation performance throughout the frame sequence. For each video, self-consistency is computed by dividing the segmentation accuracy (in terms of $\mathcal{J}\&\mathcal{F}$) achieved for each predicted mask in the sequence by the accuracy achieved in the first predicted mask. The values shown in Figure 5 represent the average self-consistency of our method in segmenting the videos of the validation split in the DAVIS-2017 dataset [7], considering an interval of 80 frames.

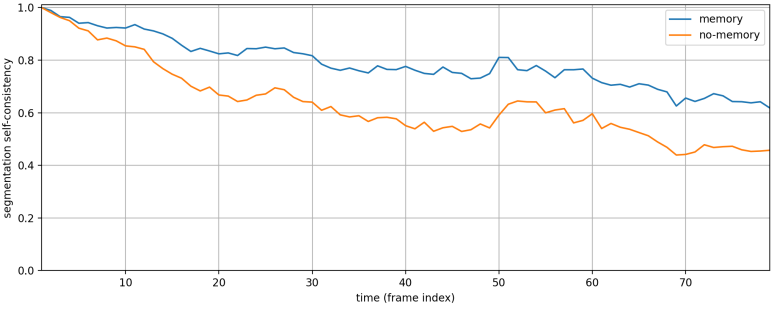


Figure 5: Self-consistency over time. The plots depict the variation of segmentation performance of our VOS method along the frame sequence, comparing two scenarios: with a memory mechanism (blue) and without memory (orange). The graph highlights the benefits of including a memory mechanism in the proposed solution, as it improves segmentation self-consistency over time.

The graph demonstrates a significant degradation in segmentation performance when the memory mechanism is absent, with accuracy dropping to less than 50% of the initial performance around frame #65. Conversely, when the memory clustering is employed, the performance loss is considerably reduced, maintaining above 60% of the initial result throughout the entire interval.

5 Qualitative results

Figures 6 and 7 provide some qualitative results generated by SHLS on videos of the DAVIS-2017 [7]. Each row corresponds to a different video from the validation set, and includes the manually annotated first frame on the left and three segmented frames on the right (at 33%, 66%, and 99% of the video progress time). The videos are arranged in descending order based on the $\mathcal{J}\&\mathcal{F}$ score achieved by our method for each video individually.

In Figure 6, we show examples where SHLS achieved $\mathcal{J}\&\mathcal{F}$ higher than its average performance (68.5). In Figure 7, we show examples where severe segmentation failures occurred.

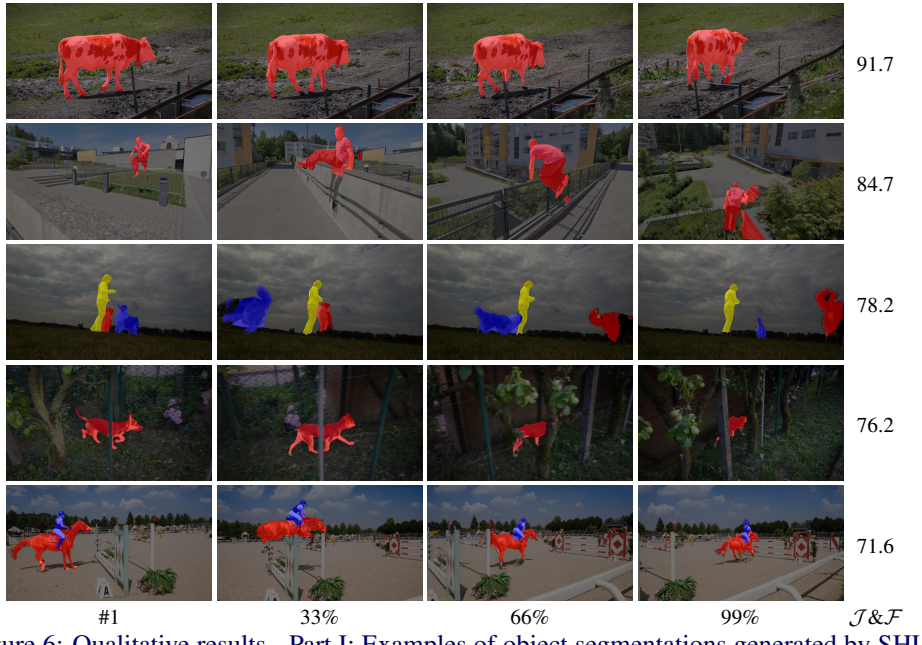


Figure 6: Qualitative results - Part I: Examples of object segmentations generated by SHLS on videos of the DAVIS-2017 validation set [7] with $\mathcal{J}\&\mathcal{F}$ higher than average (68.5). From left to right: the manually annotated first frame followed by three segmented frames at 33%, 66%, and 99% of the video progress time. The rows are arranged in descending order based on the $\mathcal{J}\&\mathcal{F}$ score achieved by SHLS for each video individually.

Figure 7: Qualitative results - Part II: Examples of object segmentations generated by SHLS on videos of the DAVIS-2017 validation set [7] with $\mathcal{J}\&\mathcal{F}$ lower than average (68.5). From left to right: the manually annotated first frame followed by three segmented frames at 33%, 66%, and 99% of the video progress time. The rows are arranged in descending order based on the $\mathcal{J}\&\mathcal{F}$ score achieved by SHLS for each video individually.

References

[1] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE TPAMI*, 37(3):569–582, 2015. doi: 10.1109/TPAMI.2014.2345401.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, 2016. ISBN 978-3-319-46493-0.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[5] Duc Tam Nguyen, Maximilian Dax, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Zhongyu Lou, and Thomas Brox. *DeepUSPS: Deep Robust Unsupervised Saliency Prediction with Self-Supervision*. 2019.

[6] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *2018 IEEE/CVF Con-*

ference on Computer Vision and Pattern Recognition, pages 7376–7385, 2018. doi: 10.1109/CVPR.2018.00770.

[7] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017.