Introducing a Self-Supervised, Superfeature-Based Network for Video Object Segmentation

Marcelo Mendonça*
Postgraduate Program in Mechatronics
Federal University of Bahia
Prof. at Federal Institute of Bahia (IFBA)
marcelomendonca@ifba.edu.br

Luciano Oliveira
Dept. of Computer Science
Federal University of Bahia
Salvador, Brazil
lrebouca@ufba.br

***Abstract*—This work introduces a novel video object segmentation (VOS) method, called SHLS, which combines superpixels and deep learning features to construct image representations in a highly compressed latent space. The proposed approach is entirely self-supervised and is trained solely on a small dataset of unlabeled still images. The result of embedding convolutional features into the corresponding superpixel areas is ultra-compact vectors named superfeatures. The superfeatures form the basis of a memory mechanism to support the video segmentation. Through it we are able to efficiently store and retrieve past information, enhancing the segmentation of current frames. We evaluated SHLS on the DAVIS dataset, the primary benchmark for VOS, and achieved superior performance in single-object segmentation as well as competitive results in multi-object segmentation, outperforming state-of-the-art self-supervised methods that require much larger video-based datasets. Our code and trained model are publicly available at: github.com/IvisionLab/SHLS.**

I. Introduction

VOS aims to classify pixels along a frame sequence into foreground and background regions. The simplest case is single-object segmentation, where no differentiation among distinct foreground objects is required. The task becomes tougher in the multi-object scenario, where each foreground object must be assigned a different label. The common approach to solving this problem relies on supervision. However, providing pixel-wise annotations for thousands of frames is complex, time-consuming, and costly. Alternatively, self-supervised methods can learn inter-frame correspondences from supervisory signals extracted directly from raw videos, eliminating the need for human annotations. However, most self-supervised methods trades off the benefit of avoiding manual labeling by requiring unprecedented volumes of training data. In extreme cases [1]–[6], the training demands hundreds of hours of videos from enormous datasets, including Kinetics [7], VLOG [8], and TrackingNet [9], or millions of images from ImageNet [10], as in [11].

We introduce here a different approach by pursuing to learn VOS not only from unlabeled images but using as little training data as possible, as highlighted in the comparison in Fig. 1, top part. The proposed method, called *superfeatures in a highly compressed latent space* (SHLS), combines superpixels and deep convolutional features to produce ultra-compact

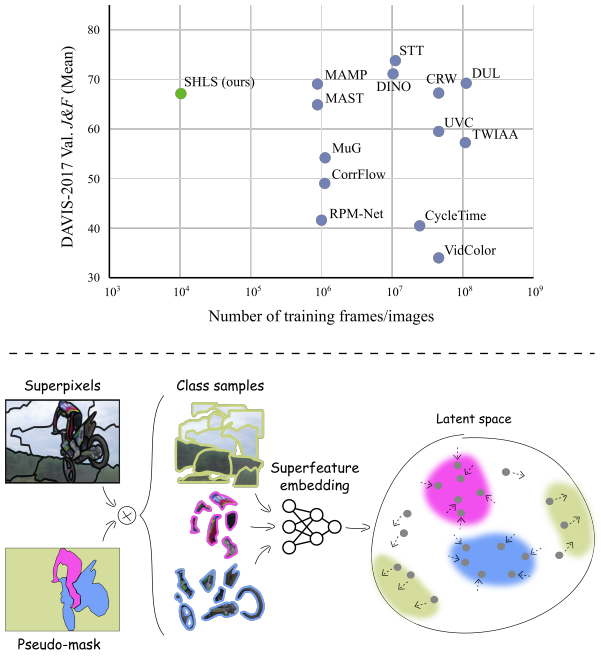

Fig. 1. The high-compressed latent space (bottom plot) generated from the superfeatures and a dataset containing at least 10^2 orders of magnitude less training images than other approaches evaluated over DAVIS-2017 (top plot).

representations in the superpixel domain. These representations, referred to here as *superfeatures*, are generated via a metric learning approach, in which our model learns to join superfeatures that come from parts of the same object (Fig. 1, bottom part). This process gives rise to a feature (latent) space where correlated superfeatures compound clusters. At the inference, such clusters are properly retrieved, identified, and used to classify the superpixels in order to reassemble the objects in the image domain. Relying on superpixels for self-supervised VOS benefits from three main aspects: (i) the lack of annotations to guide self-supervised methods makes these methods more error-prone regarding the object contours, which can be solved by relying on the superpixels contours; (ii) the high data compression provided by the superfeatures enables a memory mechanism that can efficiently retrieve information from virtually all past frames in a video sequence;

*The present work relates to the author's Ph.D. thesis.

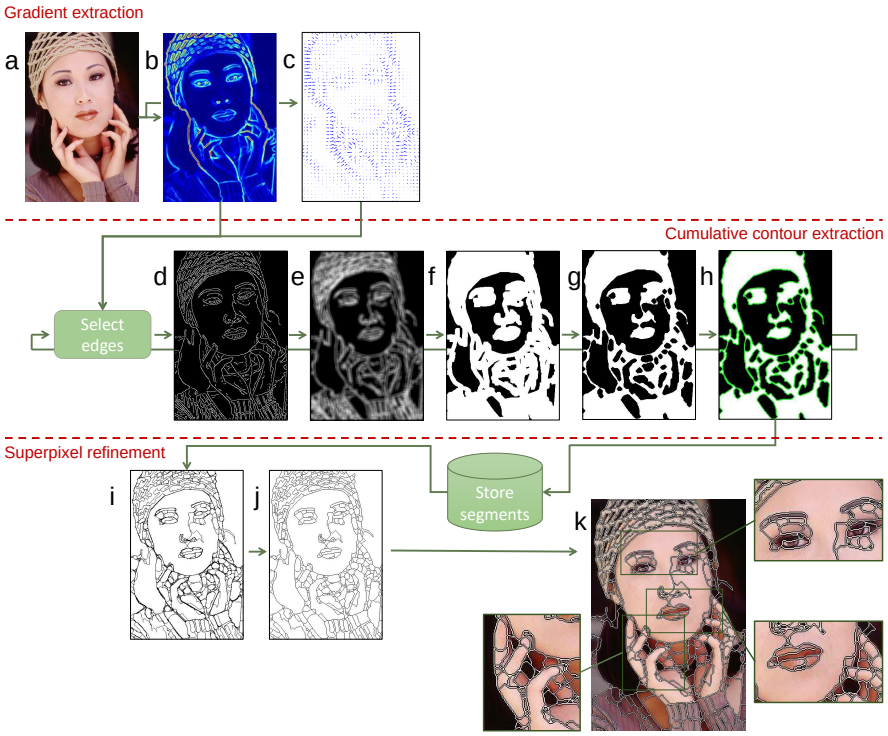

Fig. 2. Top-down view of ISEC. Gradient extraction: An input image (a) is used to compute gradient magnitudes (b) and orientations (c) in the x and y axes. Cumulative contour extraction: For each iteration, an edge set (d) is selected from the gradient; the edges are stretched (e) by edge density filtering, being binarized to form clusters (f); a thinning operation is performed on the clusters to readjust their shapes (g); the borders of the clusters are extracted (h) and stored. Superpixel refinement: The accumulated segments (i) are refined to produce the final result (j). The generated superpixels are showed over the input image (k); some parts are zoomed to highlight segmentation details.

and (iii) objects from either foreground or background can be assigned to specific superfeature clusters, ultimately resulting in more robust representations able to encompass the dynamics of both image regions.

The proposed model is trained using only the RGB images (not the annotated masks) of MSRA10K [12], a relatively small dataset comprised of 10k still images. From these images, we generate synthetic videos, pseudo-masks and superpixels to drive our embedding model toward learning the superfeatures. The result are ultra-compact vectors with dimension $1 \times S$ (in practice, we use $S = 32$), each representing the whole set of pixels contained in the corresponding superpixel area. Since a typical 480p resolution frame can be segmented with less than a thousand superpixels, we end up with $\sim 1k \times 32$ vectors to represent each frame's content – a very manageable volume that permits our memory mechanism to maintain information from every frame in a sequence. This approach makes SHLS able to learn the VOS task from a bunch of static images, showing competitive performance compared to state-of-the-art self-supervised methods trained with much larger video datasets.

A. Contributions

The main contributions of this work are twofold. First, a superpixel method called Iterative Over-segmentation via Edge Clustering (ISEC) [13], which is especially useful for video segmentation. ISEC has the convenient ability to adapt the number of generated superpixels in response to changes in the frame content along the video sequence.

Based on ISEC, we developed SHLS [14]. This one is a new VOS method that comprises several innovative characteristics, including a model based on compressed features using superpixels and metric learning, a memory mechanism based on clustering, and synthetic videos and pseudo-labels generation based on still images.

II. ISEC SUPERPIXELS

ISEC superpixels are obtained from image edges, which are iteratively selected and grouped to form clusters. Clustered pixels represent image locations where the spatial neighborhood is densely filled by edges. Since the edges are strongly related to the objects contours and texture, the clusters resemble the object shapes. Superpixels are ultimately obtained by extracting the borders that separate these clusters from image regions where there are no edges. By adjusting the edge selection procedure so that a group of edges are selected at the beginning of the process, and edges are progressively removed at each new iteration, the input image is over-segmented from the outer contours of the objects to their small internal parts. Figure 2 depicts the top-down view of ISEC.

Given an input image (Fig. 2.a), the first step is to compute the gradient magnitude (Fig. 2.b) and orientation along the

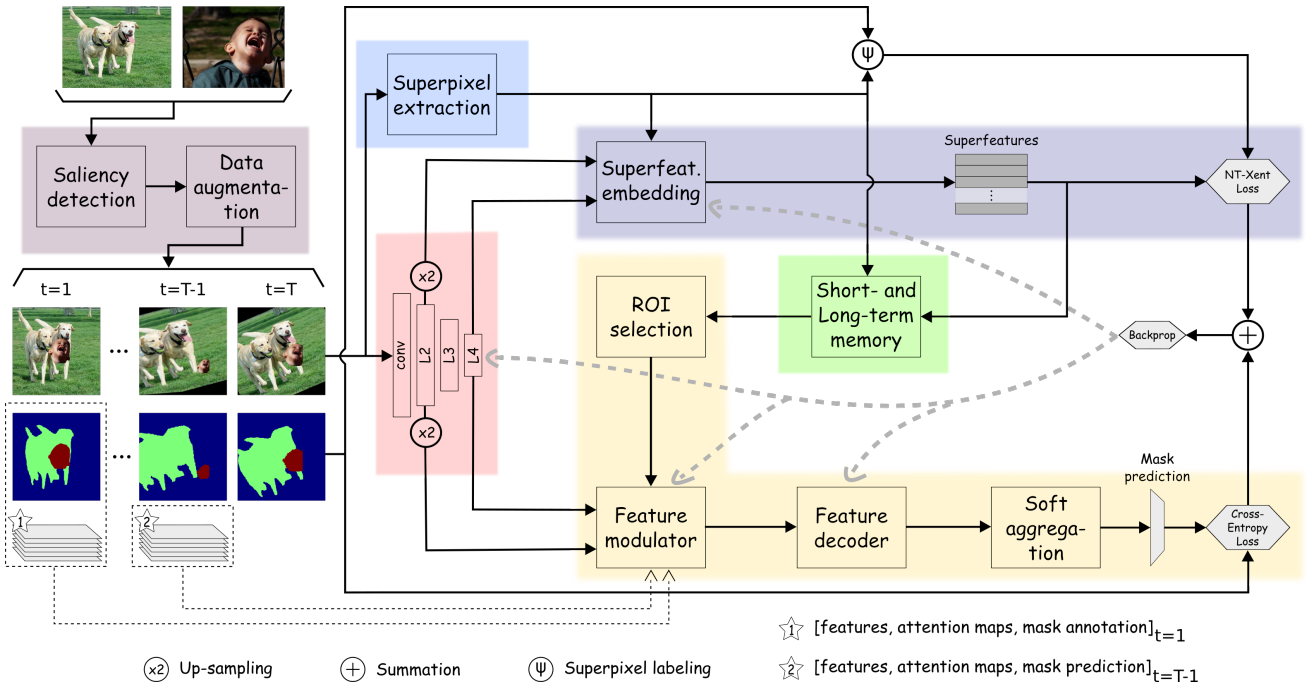

Fig. 3. Overview of SHLS at the training stage. Offline phase: Given some input still images, the pseudo-sequence generation module yields a sequence of frames and masks; following, a superpixel method extracts superpixels from the frames. Online phase: Feature maps in different scales are extracted by a CNN backbone and shared into two main branches. The uppermost branch encompasses the superpixel embedding module, which generates the superfeatures based on a contrastive NT-Xent loss. The lowermost branch accomplishes the segmentation refinement, in which the pixel-wise multi-object prediction is learned through a cross-entropy loss. This prediction is supported by the memory clustering module, which transfers information between branches by means of attention maps. At each iteration, both losses are summed and back-propagated in an end-to-end training process.

x and y axes (Fig. 2.c). Next, a set of edges (Fig. 2.d) is initially selected from the gradient map. We fix discontinuities in the edges by applying a spatial linear filtering to blur the edges over the filtered area (Fig. 2.e). Image areas massively occupied by edges form clusters that contrast with the empty surroundings after binarization (Fig. 2.f). We apply a morphological thinning operation to recover the contour's original position (Fig. 2.g). The superpixels are compound by the outer boundaries of the edge clusters (Fig. 2.h). The accumulated segments (Fig. 2.i) are then refined (Fig. 2.j), yielding the final result showed in Fig. 2.k.

ISEC stands out from most counterparts in that it generates superpixels in an adaptive fashion. While for modern methods the number of generated superpixels usually consists in a fixed hyper-parameter, our formulation leads ISEC to concentrate the superpixels in the image regions containing objects, while avoiding to over-segment empty areas unnecessarily. Such characteristic allows for a proper trade-off between the number of generated segments and the segmentation accuracy, especially in the context of video segmentation, as demonstrated in [13].

III. SHLS Video Object Segmentation

Our SHLS framework is turned to the one-shot VOS modality. During inference, it receives the ground truth mask of the first frame and propagates it to the subsequent frames. To emulate this scenario in the training, an initial offline stage is firstly accomplished, where the necessary training inputs are generated based on a bunch of still images randomly selected from the dataset [12]. Fig. 3 shows an overview of our framework. Offline-generated training inputs consist of a pseudo-sequence containing the frames and object masks and each frame's superpixel segmentation. Once generated, these inputs are processed sequentially at the online stage. The initial step is feature extraction, where convolutional feature maps of different scales are produced and shared into two main branches. The uppermost branch is dedicated to the superfeature generation. In this branch, the superpixel embedding network receives the features and superpixels of the current frame and generates the superfeatures according to a contrastive NT-Xent objective [15].

Along the frames, the generated superfeatures are stored by memory clustering. This module provides short- and long-term memory mechanisms that retrieve past frame's information to support the current frame segmentation. The memory clustering yields a set of object-focused attention maps, which are passed to the segmentation refinement branch (lowermost, in Fig. 3). Segmentation refinement is run at the pixel-level, for each foreground object individually. For this, the object region of interest (ROI) is selected from the attention maps and passed to the feature modulator and feature decoder modules. Both are network-based modules, where the former modulates the features of the current frame. This is accomplished according to the object ROI selected in the attention maps and the

features, attention maps and mask prediction of the previous frames. The modulated features and the ROI-selected attention maps are then passed to the feature decoder module. There, they are fed into the decoder network along with previous information from the first and the last iterations. The feature decoder predicts individual masks for each object in the frame. Ultimately, these masks are joined via soft-aggregation [16] to generate the final multi-object prediction. A cross-entropy function computes the error between this prediction and the corresponding pseudo-mask. At each iteration, the NT-Xent and cross-entropy losses are summed and back-propagated in an end-to-end training process.

A. Synthetic videos and masks for training

We combine saliency detection and data augmentation to create synthetic videos with object masks for self-supervised training. This strategy is completely free of manual annotations and involves three steps: (i) a random image from the dataset is selected as a template; (ii) the selected image and its estimated mask (saliency map) are replicated N times, where N is the sequence length, and each replica is an augmented version of the template; (iii) a random number of other images and corresponding saliency maps are obtained from the dataset, their foreground pixels are extracted based on the saliency, augmented, and randomly pasted into each template instance.

With these augmentation techniques, we can create an unlimited number of pseudo-sequences to train our VOS method in a self-supervised fashion.

B. Convolutional Features

We extract convolutional features by using a ResNet-18 [17] modified to enlarge the spatial size of the output feature map. The first layer of the backbone is just a convolution; the remaining are residual blocks, each one comprised of convolution, batch normalization, ReLU non-linearity, convolution, and batch normalization again. To form the superfeatures, we upsample the output of the layer with 1/2 of the original spatial dimensions to match the input size, and pass these maps (L1) to the superfeature model along with the (L4) feature maps with 1/4 of the input size.

C. Superfeature Model

The superfeature embedding process (Fig. 4) starts by averaging the convolutional features that overlap with each superpixel area. This produces feature vectors that are no longer related to the spatial dimensions of the input image but rather to the number of superpixels in the image, *i.e.*, $N \times C$, where N is the number of superpixels and C is the number of channels of the corresponding maps. Next, each row of the generated vectors are passed through fully connected (FC) layers. There are two FC heads, one for the $N \times C_1$ vector and the other for the $N \times C_4$ vector. Each head outputs a superfeature prototype of size $1 \times S$, which are concatenated and passed through a 1x1 convolution to generate the final superfeature.

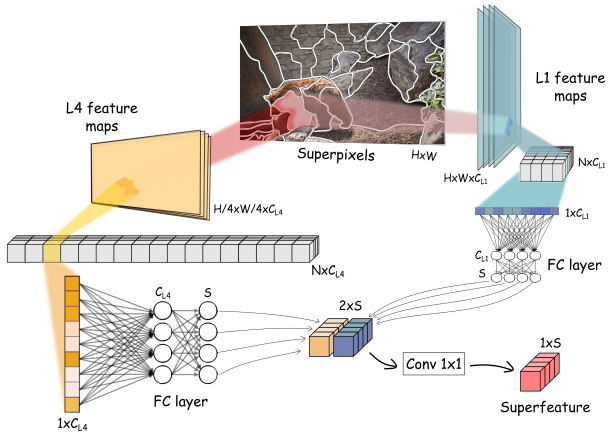

Fig. 4. To generate the superfeature, the features inside a superpixel are averaged, for each channel, yielding $N \times C_{L1}$ and $N \times C_{L4}$ vectors. These vectors are fed into fully connected layers, resulting in a $2 \times S$ vector, which is passed through a 1×1 convolution.

D. Metric Learning

The superfeature model is trained via Metric Learning by using the NT-Xent loss [15], a contrastive and multi-class objective. Once generated a synthetic sequence of frames and masks, as well as the superpixels provided by ISEC, they are passed to the model, which outputs the superfeatures. Meanwhile, the superpixels are combined with the corresponding masks to form the ground-truth labels. Each superfeature-label pair is then confronted by the NT-Xent function, and the resulting loss is summed throughout the sequence.

E. Memory Clustering

Most state-of-the-art VOS methods rely on memory mechanisms to improve segmentation stability [20], [22], [27]–[29]. Usually, this solution implies a trade-off between computational cost and segmentation performance. We overcome such dilemma with a new mechanism that treats memory management as a clustering problem. The idea combines two approaches to provide short- and long-term information through similarity measures among the superfeatures.

a) Short-term memory: it aims to provide a quick-response memory by incorporating information from more immediate changes in objects during short time intervals. This mechanism is based on k-nearest neighbor (k-NN) searches performed on the superfeature latent space. We compute the k-NN distances between each query superfeature and its nearest labeled superfeatures. This mechanism is recurrently updated by incorporating into the search pool those samples for which the class is assigned with high confidence.

b) Long-term memory: it is designed to capture the general tendency that each object presents throughout the entire video sequence. Instead of measuring the similarity between the query and the neighbors, the long-term mechanism performs per-class clustering of the superfeatures. We measure query similarity with respect to the centroids of the clusters at the prediction. Unlike it occurs with the short-term

Method	Year	Training datasets		DAVIS-2016			DAVIS-2017		
		Images	Videos (hrs)	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
VidColor [1]	2018	-	K (833)	38.9	30.8	34.9	34.6	32.7	33.7
CorrFlow [18]	2019	-	O (14.0)	48.9	39.1	44.0	47.7	51.3	49.5
CycleTime [2]	2019	-	V (344)	55.8	51.1	53.5	41.9	39.4	40.7
UVC [5]	2019	-	K (833)	-	-	-	57.7	61.3	59.5
RPM-Net [19]	2020	-	D17+Y (5.75)	-	-	-	41.0	42.2	41.6
MAST [20]	2020	-	Y (5.67)	-	-	-	63.3	67.6	65.5
MUG [21]	2020	-	O (14.0)	63.1	61.8	62.5	52.6	56.1	54.3
CRW [3]	2020	-	K (833)	-	-	-	64.8	70.2	67.6
DUL [4]	2021	-	T (140)	-	-	-	67.1	71.7	69.4
TWIAA [6]	2021	-	V+K (1,177)	-	-	-	58.2	56.7	57.5
STT [11]	2022	I	Y (5.67)	-	-	-	**71.1**	**77.1**	**74.1**
MAMP [22]	2022	-	Y (5.67)	-	-	-	68.3	71.2	69.7
SHLS (ours)	2023	M	-	**76.6**	**70.4**	**73.5**	68.3	68.7	68.5

TABLE I

COMPARISON OF SHLS WITH OTHER SELF-SUPERVISED METHODS USING STANDARD VOS METRICS: REGION JACCARD SIMILARITY ($\mathcal{J}$), BOUNDARY F-MEASURE ($\mathcal{F}$), AND THE MEAN OF BOTH ($\mathcal{J}\&\mathcal{F}$). "-" INDICATES NOT REPORTED RESULTS. THE TESTS WERE PERFORMED ON THE VALIDATION SETS OF DAVIS-2016 [23] AND DAVIS-2017 [24] FOR SINGLE AND MULTI-OBJECT VOS TASKS, RESPECTIVELY. TRAINING DATASETS: I: IMAGENET [10]; M: MSRA10K [12]; D17: DAVIS-2017 [24]; Y: YOUTUBE-VOS [25]; K: KINETICS [7]; O: OXUVA [26]; V: VLOG [8]; T: TRACKINGNET [9].

mechanism, changes in the centroids are gradual as the clusters incorporate new members when they are updated.

The similarity measures from the memory clustering are used to create a set of attention maps (Fig. 5.b). We select a region of interest (ROI) by propagating labels from the attention maps to each pixel inside a superpixel. The label of the *ith* pixel p belonging to the *jth* superpixel P, with $p_i \subset P_j \;\; \forall \; i \in 1..I_j$ and $\; j \in 1..N$, is estimated as

$$f(p_{i,k}) = \mathsf{S}_j^k + \mathsf{L}_j^k - (\mathsf{S}_j^l + \mathsf{L}_j^l) \;\; \forall \;\; k,l \in 1..C \text{ and } k \neq l\,,$$
$$p_i = \underset{k}{argmax}(f(p_{i,k}))\,, \tag{1}$$

where N is the number of superpixels in the frame, C is the number of classes present in the video, S and L are the attention maps from the short- and long-term memories, respectively.

F. Refinement Module

Fig. 5.c shows an example of segmentation produced in the superpixel level (blue) and its refined version (green) in the pixel level. The proposed refinement module is a CNN architecture with two stages: feature modulator and feature decoder.

a) Feature modulator: it receives the convolutional features and attention maps regarding the current frame and the last segmented frame in the sequence. The network learns to segment the current frame as a smooth transformation of the previous one.

b) Feature decoder: it is responsible for bringing the features back to the spatial dimensions of the input frame while reducing their channels towards the final prediction. This module is composed by refinement blocks with the function of merging features from branches at different scales.

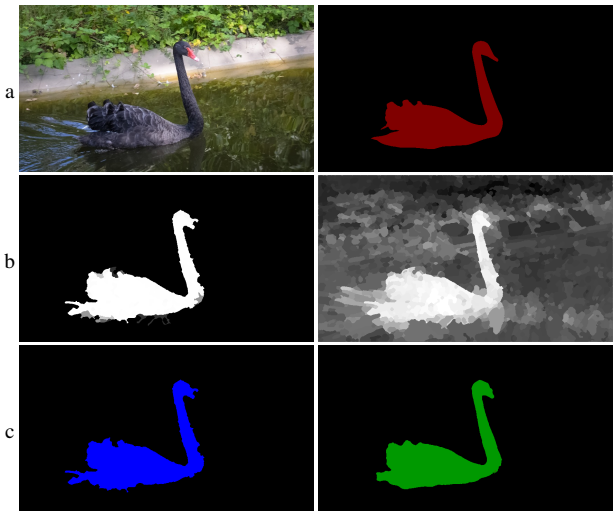

Fig. 5. The effect of the attention maps and segmentation refinement. From left to right: (a) the input frame and the ground-truth mask; (b) the attention maps from the short-term and long-term memory mechanisms; and (c) the superpixel-level segmentation and the pixel-wisely refined segmentation.

The final result is obtained by blending the refined segmentations of each object into a unified multi-object mask via soft-aggregation [16]. During the trainning, we compute a pixel-wise cross-entropy loss to adjust the weights of the refinement module.

IV. EXPERIMENTS AND ANALYSIS

Table I shows the results of the comparison between SHLS and several state-of-the-art self-supervised methods. The comparison is based on the standard VOS metrics, region Jaccard similarity ($\mathcal{J}$), boundary F-measure ($\mathcal{F}$), as well as the mean of both ($\mathcal{J}\&\mathcal{F}$). The experiments were conducted on the validation sets of the DAVIS-2016 [23] and DAVIS-2017 [24]

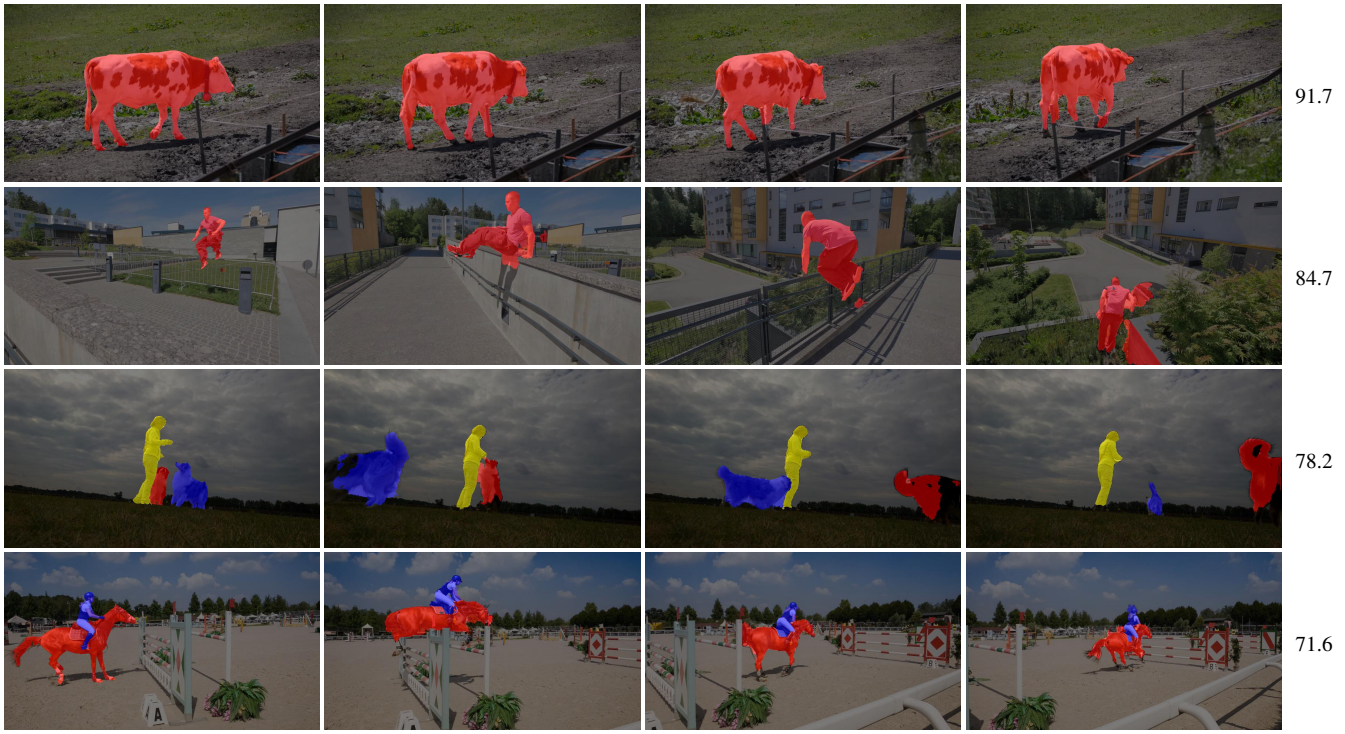

Fig. 6. Examples of object segmentations generated by SHLS on videos of the DAVIS-2017 [24] validation set. From left to right: first frame annotation, followed by generated segmentations at 33%, 66%, and 99% of the video progress time. Last column shows the $\mathcal{J}\&\mathcal{F}$ score achieved for each video.

datasets, regarding the single and multi-object VOS tasks, respectively.

a) Single-object VOS: Among the self-supervised methods that have reported results on the DAVIS-2016, SHLS ranks first in all metrics, outperforming the second-best method, MUG [21], by a large margin.

b) Multi-object VOS: As for self-supervised methods, STT [11] achieved an impressive 74.1% of $\mathcal{J}\&\mathcal{F}$. Following, there are a group of methods with $\mathcal{J}\&\mathcal{F}$ values greater than 65%, which includes MAMP [22], DUL [4], CRW [3], MAST [20], and the proposed SHLS, the only method in this comparison trained exclusively with still images.

The contrast between SHLS and the other methods is further highlighted in the graph presented at the beginning of this paper (Fig. 1), where the overall performance on the DAVIS-2017 was plotted in terms of the number of images and/or frames used for training. The plot makes clear that our method is competitive even being trained with at least 10^2 orders of magnitude less data than top-performance competitors.

We show some qualitative results illustrating the performance of our method in Fig. 6. The top rows bring examples of single-object segmentation and the bottom rows include frames with multiple objects from the DAVIS dataset. In both scenarios SHLS is able to accomplish the segmentation with reasonably correctness.

V. Conclusion

We presented SHLS, a self-supervised VOS method leveraging highly compressed superpixel-based representations called superfeatures. This novel approach organizes superfeatures into per-object clusters using a memory clustering mechanism to retrieve information from past frames. Our fully self-supervised training methodology, utilizing only 10k still images, demonstrates SHLS's efficacy. Experiments on the DAVIS dataset reveal that SHLS significantly outperforms other self-supervised methods in the single-object test and remains competitive in the multi-object test, despite the smaller training data volume. Future work will focus on incorporating automatic foreground detection during inference, extending SHLS to the zero-shot VOS modality.

VI. Achievements

The following achievements were obtained as a result of our Ph.D. thesis:

- Registered Patent: *Device and Method for Intelligent Traffic Light Control* [30]
- Articles:
  - *ISEC: Iterative over-Segmentation via Edge Clustering* [13]
  - *SHLS: Superfeatures Learned from Still Images for Self-Supervised VOS* [14]
  - *Faster α-Expansion via Dynamic Programming and Image Partitioning* [31]
- Award: *Best Ph.D. Thesis at the XVI Brazilian Congress on Computational Intelligence (CBIC 2023)* [32]

REFERENCES

[1] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy, "Tracking emerges by colorizing videos," in *Computer Vision – ECCV 2018: 15th European Conference*, 2018, p. 402–419.

[2] X. Wang, A. Jabri, and A. A. Efros, "Learning correspondence from the cycle-consistency of time," in *CVPR*, 2019.

[3] A. Jabri, A. Owens, and A. A. Efros, "Space-time correspondence as a contrastive random walk," *Advances in Neural Information Processing Systems*, 2020.

[4] N. Araslanov, S. Schaub-Meyer, and S. Roth, "Dense unsupervised learning for video segmentation," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 25 308–25 319.

[5] X. Li, S. Liu, S. De Mello, X. Wang, J. Kautz, and M.-H. Yang, "Joint-task self-supervised learning for temporal correspondence," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[6] W. Zhu, J. Meng, and L. Xu, "Self-supervised video object segmentation using integration-augmented attention," *Neurocomput.*, vol. 455, no. C, p. 325–339, 2021.

[7] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4724–4733.

[8] D. F. Fouhey, W. Kuo, A. A. Efros, and J. Malik, "From lifestyle vlogs to everyday interactions," in *CVPR*, 2018.

[9] M. Muller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, "Trackingnet: A large-scale dataset and benchmark for object tracking in the wild," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[11] R. Li and D. Liu, "Spatial-then-temporal self-supervised learning for video correspondence," 2022.

[12] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE TPAMI*, vol. 37, no. 3, pp. 569–582, 2015.

[13] M. Mendonça and L. Oliveira, "Isec: Iterative over-segmentation via edge clustering," *Image and Vision Computing*, vol. 80, pp. 45–57, 2018.

[14] M. Mendonça, J. Fontinele, and L. Oliveira, "Shls: Superfeatures learned from still images for self-supervised vos," in *34th British Machine Vision Conference BMVC, Aberdeen, UK*, 2023.

[15] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML'20, 2020.

[16] S. W. Oh, J.-Y. Lee, K. Sunkavalli, and S. J. Kim, "Fast video object segmentation by reference-guided mask propagation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7376–7385.

[17] 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)*, 2016, pp. 770–778.

[18] Z. Lai and W. Xie, "Self-supervised learning for video correspondence flow," in *BMVC*, 2019.

[19] Y. Kim, S. Choi, H. Lee, T. Kim, and C. Kim, "Rpm-net: Robust pixel-level matching networks for self-supervised video object segmentation," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 2046–2054.

[20] Z. Lai, E. Lu, and W. Xie, "MAST: A memory-augmented self-supervised tracker," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[21] X. Lu, W. Wang, J. Shen, Y. Tai, D. J. Crandall, and S. H. Hoi, "Learning video object segmentation from unlabeled videos," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8957–8967.

[22] B. Miao, M. Bennamoun, Y. Gao, and A. Mian, "Self-supervised video object segmentation by motion-aware mask propagation," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*, 2022, pp. 1–6.

[23] 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.

[24] 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.

[25] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang, "Youtube-vos: Sequence-to-sequence video object segmentation," in *Computer Vision – ECCV 2018: 15th European Conference*, 2018, p. 603–619.

[26] J. Valmadre, L. Bertinetto, J. F. Henriques, R. Tao, A. Vedaldi, A. W. Smeulders, P. H. Torr, and E. Gavves, "Long-term tracking in the wild: a benchmark," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[27] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[28] H. Seong, S. W. Oh, J.-Y. Lee, S. Lee, S. Lee, and E. Kim, "Hierarchical memory matching network for video object segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 12 889–12 898.

[29] X. Xu, J. Wang, X. Li, and Y. Lu, "Reliable propagation-correction modulation for video object segmentation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, pp. 2946–2954, 2022.

[30] L. Oliveira and M. Mendonça, "Device and method for intelligent traffic light control," Patent BR102 015 010 366-2, 2023.

[31] J. Fontinele, M. Mendonça, M. Ruiz, J. Papa, and L. Oliveira, "Faster -expansion via dynamic programming and image partitioning," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–8.

[32] J. Veloso, "Teses da ufba em geografia e mecatrônica conquistam prêmios em eventos específicos das áreas," *Edgard Digital*, available at: https://www.edgardigital.ufba.br/?p=27221(Accessed: June 21th, 2024).