SHLS: Superfeatures Learned from Still Images for Self-supervised VOS

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

Abstract

Self-supervised video object segmentation (VOS) aims at eliminating the need for manual annotations to learn VOS. However, existing methods often require extensive training data consisting of hours of videos. In this paper, we introduce a novel approach that combines superpixels and deep learning features through metric learning, enabling us to learn VOS from a small dataset of unlabeled still images. Our method, called superfeatures in a highly compressed latent space (SHLS), embeds convolutional features into the corresponding superpixel areas, resulting in ultra-compact image representations. This allowed us to construct an efficient memory mechanism to store and retrieve past information throughout a frame sequence to support current frame segmentation. We evaluate our method on the popular DAVIS dataset and achieve competitive results compared to state-of-the-art self-supervised methods, which were trained with much larger video-based datasets. We have made our code and trained model publicly available at: https://github.com/IvisionLab/SHLS.

1 Introduction

Video object segmentation (VOS) is a crucial task with potential applications in various areas, such as video processing [34], visual tracking [29], human pose estimation [39], surveillance [25], to cite a few. Its objective is to classify pixels into foreground and background regions in a sequence of frames. The task is more challenging when it involves multiple objects, requiring each foreground object to be assigned a unique label. The traditional approach to this task is based on human supervision, which is complex, time-consuming, and costly due to the requirement for pixel-wise annotations of numerous frames.

More recently, self-supervised approaches have been proposed as alternatives to allow VOS training based on completely unlabeled data [2, 13, 15, 16, 17, 18, 19, 21, 23, 38, 40, 46]. These methods learn inter-frame correspondences from supervisory signals extracted directly from raw videos, eliminating the need for human supervision. However, many self-supervised methods require extensive volumes of training videos to compensate for the lack of annotated frames. For instance, [2, 13, 19, 38, 40, 46] are trained using massive video

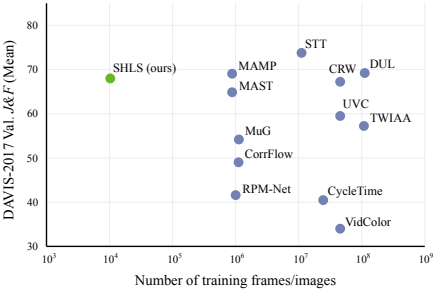

Figure 1: Benchmark on DAVIS-2017 validation set. SHLS is trained with at least 10^2 orders of magnitude fewer images than other self-supervised methods.

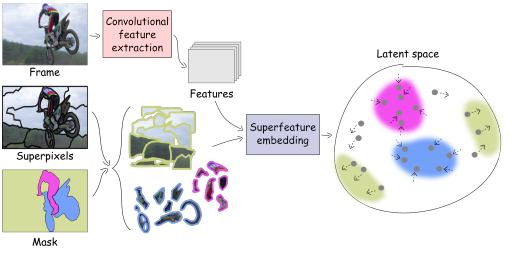

Figure 2: Given an input image and the feature maps from a CNN, the latent space is built by a superfeature embedding model combining superpixels and convolutional features.

datasets, including Kinetics [4], VLOG [11], and TrackingNet [24], each with hundreds of hours of videos. Meanwhile, other methods, like [18], require millions of images from ImageNet [9] for pre-training.

This work presents a novel approach to learning VOS from unlabeled images and using as little training data as possible (Fig. 1). Our model, named superfeatures in a highly compressed latent space (SHLS), introduces the concept of *superfeatures* – ultra-compact representations of superpixels and deep convolutional features learned through metric learning. These superfeatures are generated as embeddings that are positioned close to each other when they come from the same object in the image or distant apart otherwise, as shown in Fig. 2. The resulting clusters allow us to reassemble the pieces of the objects (*i.e.*, their superpixels) by classifying the corresponding superfeatures in the latent space.

Relying on superpixels for self-supervised VOS provides several benefits. First, it reduces the impact of errors in learning object shapes, particularly regarding object contours, which are more prone to occur due to the lack of annotated masks in self-supervised VOS. Second, due to the high data compression of the superfeatures, we can efficiently store them in memory to support video segmentation. In fact, the ability to retrieve past information during frame processing is a crucial feature for many modern VOS methods [17, 23, 28, 32, 42]. Third, in line with previous works that explore background features in VOS [43], SHLS provides image representations in which the background dynamics are also embedded. It is possible since superfeatures originating from background superpixels can be allocated to background-specific clusters.

To learn VOS exclusively from unlabeled data, our approach combines saliency detection with data augmentation to synthesize pseudo-sequences consisting of frames and masks with multiple objects. The proposed training strategy is learned on MSRA10K [7], a relatively small dataset of still images with a maximum resolution of 300×400. The synthesized pseudo-masks are used to train our superfeature embedding model with a multi-class contrastive objective. It results in superfeatures with dimension $1 \times S$ (in practice, we use $S = 32$), each representing the whole bunch of pixels contained in the corresponding superpixel area. Since the superpixel segmentation of a 480p resolution frame typically contains less than a thousand superpixels, we end up with $\sim 1k \times 32$ superfeature vectors representing the entire frame content. Differently from methods based on memory banks where large feature maps are accumulated [17, 23, 28, 32, 42], our superfeature-based memory mechanism does not require any special maintenance protocol to prevent overhead, and efficient similar-

ity search [14] is used for memory access. Our approach allows SHLS to learn VOS from very little data (only the 10k images from the MSRA10K dataset) while achieving competitive performance compared to state-of-the-art self-supervised methods trained with much larger video-based datasets.

2 Related Work

2.1 One-shot VOS

In VOS literature, methods trained with unlabeled data are called *self-supervised*. This term can be confused with the term "semi-supervised," often used to denote the inference phase based on the first frame annotation. To avoid ambiguity, in this paper, we prefer the term "one-shot" instead of "semi-supervised".

Metric learning-based methods learn data representations that are placed close together in the embedding space when they belong to the same class. In [45], metric learning is used for object feature matching based on prior probabilities from past features. In [6], embeddings for individual pixels are learned using a triplet loss. At inference, labels are transferred from annotated pixels in the first frame to query pixels in subsequent frames. These methods rely heavily on manual annotations, and their pixel-wise approach cannot support robust memory clustering. In contrast, our SHLS is fully self-supervised and provides an efficient memory mechanism based on superfeatures to support video segmentation.

Memory-based methods use memory repositories to accumulate spatiotemporal features from past inputs and support current frame segmentation. These mechanisms typically rely on affinity matrices to match the current input and memory entries. To avoid the overhead of computing large affinity matrices, some methods constrain memory size [17, 23] or memory entry routines [28] over frames. More elaborate memory handling schemes have also been proposed [32, 42]. Our proposed memory-clustering mechanism allocates superfeatures from past frames into class-specific clusters, allowing efficient similarity searches [14] on these clusters due to the strong compactness provided by the superfeatures

2.2 Self-supervised VOS

Self-supervised VOS learning uses proxy tasks to explore the intrinsic properties of videos, such as the temporal coherence between frames. Although these proxy tasks differ from the desired objective, they are still effective in driving the VOS learning process.

Frame reconstruction refers to proxy tasks in which a model is trained on incomplete input data to reconstruct the missing information. The original and reconstructed input difference is used as a supervisory signal. For example, in [38], a video re-colorization problem is formulated by converting frames to grayscale before reconstruction. In [16], the lack of color information is alleviated by randomly dropping out one RGB channel. LAB dropout is used in [17], as this color space presents less inter-channel correlation. In [15], the entire frame is reconstructed based on the previous frame. In general, adjacent frames are used to ensure correspondence between pixels at each time step, which can result in frame reconstruction requiring large volumes of video data to capture motion features. In contrast, our proposed method is trained on a relatively small dataset consisting of still images.

Cycle consistency is assessed by tracking frame pixels in a closed cycle, where a video sequence starts and ends at the same frame. The supervisory signal is then computed as the

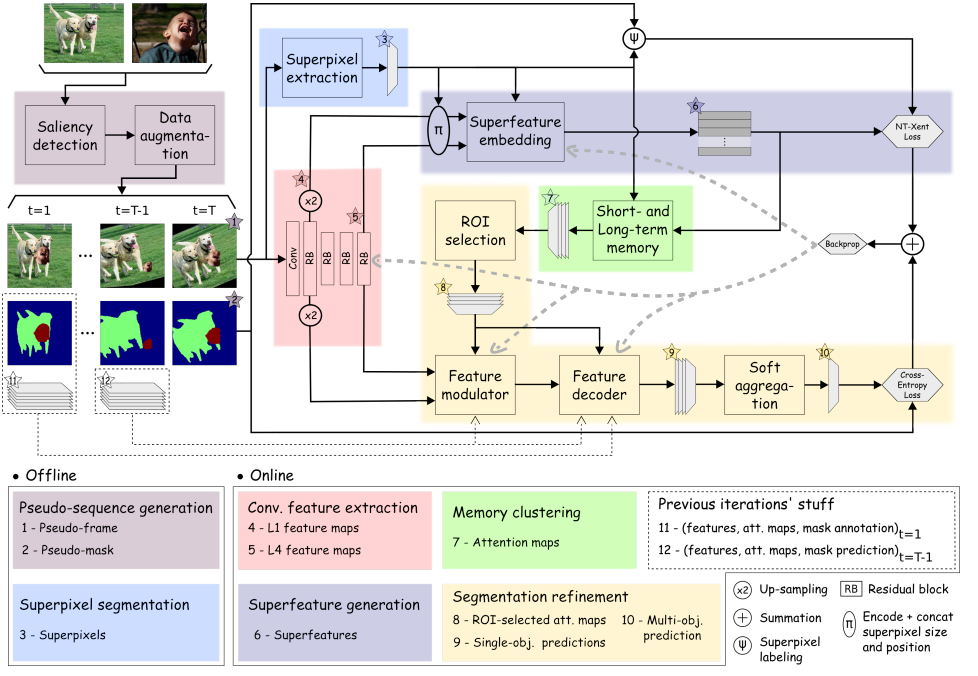

Figure 3: SHLS training involves an offline phase to generate frames and masks from input images and superpixels. The online phase uses a CNN backbone to extract feature maps shared between two branches: one for superfeature generation with a contrastive NT-Xent loss and the other for segmentation refinement with a pixel-wise multi-object prediction and cross-entropy loss. The memory clustering module transfers information between the branches with attention maps. Both losses are back-propagated during end-to-end training.

displacement error in the tracking. In [40], tracking is accomplished via template-matching in a feature space. In [19], a proposed training scheme that jointly considers object- and pixel-level correspondences is proposed. In [13], a graph is formed by frames ordered in *palindromes*, where nodes are frame patches and edges are affinities between frames. The model is encouraged to find paths through the graph to connect patches in the initial and last frame. Cycle consistency assumes that objects change smoothly, which can lead to less robust features in realistic scenarios. In SHLS, we propose a new self-supervised training approach that addresses challenging conditions, *e.g.*, occlusions, abrupt changes, fast-moving objects, etc.

Pseudo-labels are automatically generated data annotations used for self-supervised learning. For instance, [21] applies a saliency detector to estimate foreground masks and guide a learning process based on short- and long-term frame granularity analysis. In [2], pseudo-labels are obtained by generating a transformed view of the original video through data augmentation. In [18], local correlation maps computed from a pyramid feature map are used as pseudo-labels to learn a frame reconstruction task. SHLS is also based on saliency detection, as in [21]. However, instead of generating single masks with only one foreground label, we combine saliency detection with data augmentation to automatically synthesize dynamic pseudo-sequences of varying lengths and containing multiple objects.

3 Superfeatures in a highly compressed latent space

An overview of our SHLS network is shown in Fig. 3. During training, an initial offline stage is first accomplished. The training inputs (pseudo frames, masks, and superpixels) are generated from a bunch of still images randomly selected from the dataset [7]. These inputs are processed sequentially at the online stage, where frame features are extracted and shared into two main branches. The uppermost branch receives the features and superpixels of the current frame and generates the superfeatures according to a contrastive NT-Xent loss [5]. Memory clustering retrieves past generated superfeatures to support current frame segmentation. This module yields a set of attention maps, which are passed to the lowermost branch, segmentation refinement, where the final mask is predicted and the cross-entropy error is computed between this prediction and the corresponding pseudo-mask.

3.1 Pseudo-sequence generation

We synthesize pseudo frames and masks through data augmentation of still images. To make our method self-supervised, we use a saliency detector [26] to estimate foreground masks instead of manual annotations as in [27]. The process involves three steps: (i) selecting a random image from the dataset as a template, (ii) replicating the selected image N times, where N is the sequence length, and augmenting each replica of the template, and (iii) randomly selecting different image-mask pairs from the dataset, extracting their foreground pixels, augmenting them, and randomly pasting them into each template instance.

The generated frames contain diverse dynamics, such as single-object sequences where the foreground and background are taken from the same image to prevent illumination discrepancies, and multi-object sequences that simulate challenging conditions, including partial occlusions, disappear/reappear situations, and cloned objects. All generated samples include different levels of photometric variations, resizing, affine transformations, and other augmentation techniques that are individually and randomly applied to each foreground instance and the background. After generation, the sequence is segmented into superpixels and passed to the next stage. During the online phase, the frames are processed sequentially.

3.2 Combining superpixels and features in superfeatures

Superfeatures are embedding vectors generated from convolutional features within the area covered by corresponding superpixels. Figure 4 illustrates the superfeature embedding process. To extract these features, we use ResNet-18 [12] as a backbone, with minor modifications made to increase the spatial size of the output feature maps, similar to [17]. This results in two feature maps, L1 and L4, corresponding to scale factors of 1 and 1/4, respectively, relative to the input frame. Before feeding the embedding model, we encode the size and position of each superpixel, then concatenate this infor-

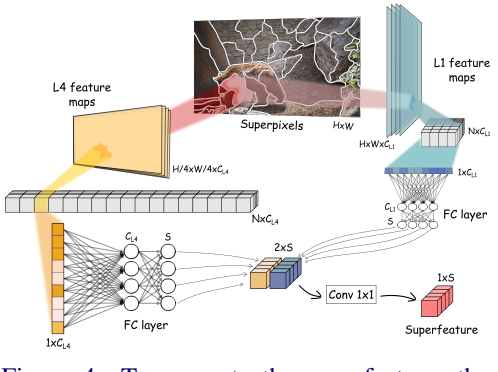

Figure 4: To generate the superfeature, the features inside a superpixel are averaged, for each channel, yielding $N \times C_{L1}$ and $N \times C_{L4}$ vectors. These vectors are fed into fully-connected layers, resulting in a $2 \times S$ vector, which is passed through a 1×1 convolution.

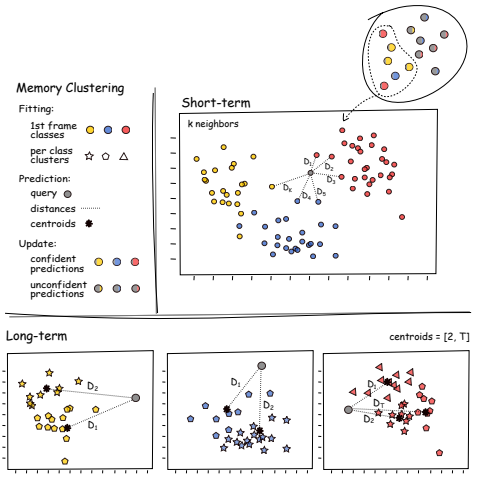

Figure 5: The memory clustering has short-term and long-term mechanisms that measure distances from a superfeature query to k-nearest neighbors of any class and centroids of clusters composed of superfeatures of the same class, respectively.

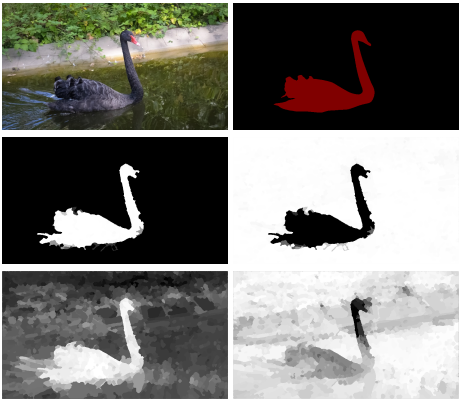

Figure 6: Examples of attention maps: (top row) video frame and ground-truth mask; (middle) positive and contrastive maps from short-term memory mechanism; and (bottom) positive and contrastive maps from long-term memory mechanism.

mation with the L1 and L4 feature maps. To generate the superfeatures, the first step is to calculate the average value of the features that overlap with each superpixel area for each feature map. While reducing the features to their mean value might suggest a substantial loss of information, the homogeneity among pixels within the same superpixel means there is little variability in the corresponding features. This averaging process 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. The next step is to pass each row of the generated vectors 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.

3.3 Memory Clustering

The proposed memory clustering mechanism provides short- and long-term information through a memory structure with three main stages: fitting, prediction, and update. This approach is based on measuring similarity distances among superfeatures in the latent space, as depicted in Figure 5.

The **short-term memory** is designed to respond quickly to immediate changes in the objects during short intervals. It is based on k-NN searches in the superfeature latent space. During the **fitting** stage, the superfeatures of the first frame are labeled based on the provided annotated mask. The labels are assigned to the objects most overlapping with the corresponding superpixels. At **prediction**, we compute the k-NN distances between each query superfeature (*i.e.*, the unclassified superfeatures from the second frame onward) and

its nearest labeled superfeatures. Finally, query superfeatures that are classified with high confidence (*i.e.*, the similarity to the class is above a threshold) are incorporated into the search pool during the **update**.

The **long-term memory** differs from its short-term counterpart in that it is not substantially affected by sudden changes in the video scene. During the **fitting** stage, the superfeatures of the first frame are grouped into class-specific clusters using k-means clustering. At **prediction**, distances from the query superfeatures to the cluster centroids are computed. Because the centroids change gradually as the clusters incorporate new members during the **update**, the long-term mechanism captures the general appearance presented by the objects for longer intervals. Moreover, each object can be associated with a variable number of clusters to reduce intra-cluster variance by avoiding grouping together too many distinct sub-parts of an object. The number of clusters assigned to each object is determined by the number of superpixels that belong to the object in the first frame.

The similarity measures produced by the memory clustering mechanisms are used to generate a set of **attention maps**. For each object, two pairs of positive-contrastive maps are generated. In each pair, the positive maps contain the similarity measure between the query superfeatures and the most similar references (k-neighbors or centroids) belonging to the same class. The contrastive maps, on the other hand, contain the similarity measure between the query superfeatures and the most similar references belonging to a different class. Figure 6 illustrates the attention maps obtained from a video of the DAVIS-17 [31] dataset. The top row shows the video frame and ground-truth mask; the middle and bottom rows show the positive-contrastive pairs from the short-term and long-term memory mechanisms, respectively. As can be observed, the attention maps generated by short-term memory are well-defined and reflect the estimated state of the object at that moment. Conversely, the attention maps generated by long-term memory are more diffuse and represent the prevailing state of the object over a longer period.

3.4 Segmentation Refinement

After the previous superpixel-based stages, SHLS performs a segmentation refinement at the pixel level, which enables it to recover from inaccurate superpixel segmentations or superfeature misclassifications. This module comprises several stages, where predictions related to the current and previous frames (*e.g.*, backbone features, attention maps, object masks) are used to produce a refined segmentation result.

The first stage is **ROI selection**, where a bounding box enclosing the object in a pre-segmentation mask is computed. We obtain this pre-segmentation mask by propagating labels from the attention maps to each pixel inside a superpixel. Therefore, 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

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

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.

The next stage, **feature modulator**, is a network module that acts as a gate mechanism, letting pass or filtering out features based on the object priors given by the ROI-selected

Method	Year	Sup.	Training datasets		DAVIS-2016			DAVIS-2017		
			Images	Videos (hrs)	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
OSVOS[3]	2017	✓	I	D16 (0.05)	79.8	80.6	80.2	56.6	63.9	60.3
RGMP[27]	2018	✓	E+M+P	D17 (0.09)	81.5	82.0	81.8	64.8	68.6	66.7
RVOS[36]	2019	✓	I	D17+Y (5.75)	-	-	-	57.5	63.6	60.6
FEELVOS[34]	2019	✓	I+C	D17+Y (5.75)	81.1	82.2	81.7	69.1	74.0	71.5
STM[28]	2019	✓	I+C+E+M+P	D17+Y (5.75)	88.7	90.1	89.4	79.2	84.3	81.7
CFBI[43]	2020	✓	I+C	D17+Y (5.75)	89.6	91.7	90.7	80.5	86.0	83.3
HMMN[32]	2021	✓	I+C+E+M+P	D17+Y (5.75)	89.6	92.0	90.8	81.9	**87.5**	84.7
AOT[42]	2021	✓	I+C+E+M+P	D17+Y (5.75)	**90.1**	92.1	**91.1**	**82.3**	**87.5**	**84.9**
RPCM[41]	2022	✓	-	D17+Y (5.75)	87.1	**94.0**	90.6	81.3	86.0	83.7
EMVOS[8]	2022	✓	I+C	D17+Y (5.75)	87.9	88.9	88.4	76.9	81.2	79.0
VidColor[38]	2018	×	-	K (833)	38.9	30.8	34.9	34.6	32.7	33.7
CorrFlow[16]	2019	×	-	O (14.0)	48.9	39.1	44.0	47.7	51.3	49.5
CycleTime[40]	2019	×	-	V (344)	55.8	51.1	53.5	41.9	39.4	40.7
UVC[19]	2019	×	-	K (833)	-	-	-	57.7	61.3	59.5
RPM-Net[35]	2020	×	-	D17+Y (5.75)	-	-	-	41.0	42.2	41.6
MAST[17]	2020	×	-	Y (5.67)	-	-	-	63.3	67.6	65.5
MUG[20]	2020	×	-	O (14.0)	63.1	61.8	62.5	52.6	56.1	54.3
CRW[13]	2020	×	-	K (833)	-	-	-	64.8	70.2	67.6
DUL[9]	2021	×	-	T (140)	-	-	-	67.1	71.7	69.4
TWIAA[26]	2021	×	-	V+K (1,177)	-	-	-	58.2	56.7	57.5
STT[18]	2022	×	I	Y (5.67)	-	-	-	**71.1**	**77.1**	**74.1**
MAMP[23]	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 1: Results on the DAVIS single-object (2016) and multi-object (2017) validation sets. Training datasets: I: ImageNet [3]; D16: DAVIS-2016 [30]; E: ECSSD [33]; M: MSRA10K [7]; P: Pascal-VOC [10]; D17: DAVIS-2017 [31]; Y: YouTube-VOS [41]; C: COCO [20]; K: Kinetics [4]; O: OxUvA [35]; V: VLOG [11]; T: TrackingNet [24]. The results shown were obtained from the original papers, with "-" indicating cases where the papers did not provide results for the experiment.

attention maps. The modulated features are then passed to the **feature decoder**, which brings the features back to the spatial dimensions of the input frame while reducing their channels toward the final prediction. Refinement blocks [27] are used to merge features with different scales. Like [43], we always include data from the first frame in the feature decoder, as the first mask provided in one-shot VOS is the most reliable information. The output of the feature decoder is the mask prediction for each object individually. Finally, after yielding all the predictions related to a frame, we apply a **soft-aggregation** operation [27] to combine the individual object predictions into a unified multi-object mask.

4 Experimental evaluation

4.1 Comparative analysis

The comparison of SHLS with various supervised and self-supervised methods was conducted using standard VOS metrics, including region Jaccard similarity ($\mathcal{J}$) and boundary F-measure ($\mathcal{F}$), as well as the mean of both ($\mathcal{J}\&\mathcal{F}$). The tests were performed on the validation sets of DAVIS-2016 [30] and DAVIS-2017 [31] for the single and multi-object VOS tasks, respectively. The comparison results are presented in Table 1.

Single-object VOS: The best performances in this test (DAVIS-2016) were achieved by supervised methods based on memory mechanisms, including STM [28], CFBI [43], HMMN

Superpixel Method	Superpixel # mean	$\mathcal{J}\&\mathcal{F}$
ISEC	1K	66.5
SLIC	1K	65.2
SLIC	2k	66.7
SLIC	3k	67.7
SLIC	4K	68.5
SLIC	5K	68.0

Table 2: Impact of two different superpixel approaches on SHLS: ISEC [22] vs. SLIC [1].

Memory size (# superfeat.)	$\mathcal{J}\&\mathcal{F}$
0	44.7
1k	47.9
2k	53.9
4k	58.0
8k	68.5
max. size	68.5

Table 3: Ablation on increasing the memory limit of SHLS.

Pre-seg. (Eq. 1)	Feat. modulator + Feat. decoder	ROI selection	$\mathcal{J}\&\mathcal{F}$
✓			62.5
✓	✓		65.3
✓	✓	✓	68.5

Table 4: Ablation on the components of the segmentation refinement module.

[32], RPCM [42], and EMVOS [8], as well as the top-1 AOT [44]. Among the self-supervised methods that have reported results on DAVIS-2016, SHLS ranks first in all metrics, outperforming the second-best method, MUG [21], by a large margin.

Multi-object VOS: In this test (DAVIS-2017), once again, the best performances were achieved by supervised methods, with AOT reaching top-1 in this test as well. As for self-supervised methods, STT [18] achieved an impressive 74.1% of $\mathcal{J}\&\mathcal{F}$, which surpasses several supervised methods. Following STT, a group of methods achieved $\mathcal{J}\&\mathcal{F}$ values greater than 65%, which includes MAMP [23], DUL [2], CRW [13], MAST [17], and the proposed SHLS method, the only one trained exclusively with still images. Notably, our method is competitive even when trained with at least 10^2 orders of magnitude fewer data than top-performing approaches.

4.2 Ablation study

Superpixel segmentation: In this experiment, we evaluated the performance of SHLS with distinct superpixel approaches, ISEC [22] and SLIC [1]. The former adapts the number of superpixels to the image content, while the latter fixes this number as a hyperparameter. ISEC automatically generated a mean of 1k superpixels in this test, while for SLIC, we progressively increased the amount from 1k to 5k. The results are presented in Table 2. For an equal number of superpixels, ISEC is superior. However, the best performance was achieved with SLIC and 4k superpixels (no improvements were observed above this number).

Memory size: SHLS uses a memory mechanism based on highly compressed superfeatures to store information. To evaluate the benefits of this mechanism for video segmentation, we conducted experiments where we varied the memory size from zero (without memory) and gradually increased it to the maximum size (corresponding to all generated superfeatures for a video). The results in Table 3 demonstrate that increased memory size is crucial for achieving good performance.

Segmentation refinement: We conducted an experiment to evaluate the effect of the segmentation refinement module on the performance of SHLS. We tested our method in three different configurations: (i) without refinement, *i.e.*, the final result is obtained directly from the pre-segmentation mask given by Eq. 1; (ii) without ROI selection, *i.e.*, the feature modulator and feature decoder are applied to the entire spatial area of the feature maps; and (iii) the segmentation refinement module is fully integrated, *i.e.*, the feature modulator and feature decoder are focused on the ROI indicated by the attention maps. The results presented in Table 4 demonstrate the importance of each component of our refinement module.

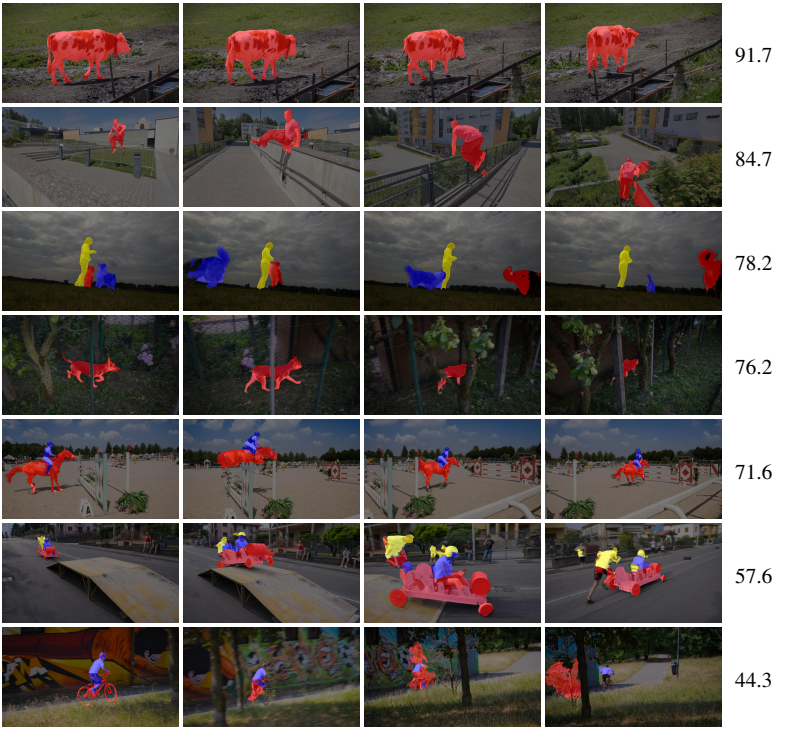

Figure 7: Examples of object segmentations generated by SHLS on videos of the DAVIS-2017 [31] validation set. From left to right: first frame annotation, followed by generated segmentations at 33%, 66%, and 99% of the video progress time. The rows are arranged in descending order based on the $\mathcal{J}\&\mathcal{F}$ score achieved by SHLS for each video individually.

4.3 Qualitative results

In Figure 7, we provide some qualitative results generated by SHLS on videos of the DAVIS-2017 [31] validation set. The examples are arranged in descending order based on the $\mathcal{J}\&\mathcal{F}$ score achieved by our method for each video individually. The last two rows show examples where severe segmentation failures occurred.

5 Conclusion

We introduced SHLS, a self-supervised VOS method that uses highly compressed superpixel-based representations called superfeatures. This innovative approach can retrieve information from past frames using a memory clustering mechanism that organizes the superfeatures into per-object clusters. Our fully self-supervised training methodology enables training with only 10k still images. Our experiments on the DAVIS dataset demonstrate that SHLS outperforms self-supervised methods by a large margin on the single-object DAVIS test and remains competitive on the multi-object test, despite being trained with significantly fewer data than competitors. In future work, we plan to further apply automatic foreground detection during inference, extending SHLS to the zero-shot VOS modality.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(11):2274–2282, 2012. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.120.

[2] Nikita Araslanov, Simone Schaub-Meyer, and Stefan Roth. Dense unsupervised learning for video segmentation. In *Advances in Neural Information Processing Systems*, volume 34, pages 25308–25319, 2021.

[3] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] 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)*, pages 4724–4733, 2017. doi: 10.1109/CVPR.2017.502.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20, 2020.

[6] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] 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.

[8] Suhwan Cho, Woo Jin Kim, MyeongAh Cho, Seunghoon Lee, Minhyeok Lee, Chaewon Park, and Sangyoun Lee. Pixel-level equalized matching for video object segmentation, 2022.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[10] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. 88(2):303–338, 2010. ISSN 0920-5691. doi: 10.1007/s11263-009-0275-4.

[11] David F. Fouhey, Weicheng Kuo, Alexei A. Efros, and Jitendra Malik. From lifestyle vlogs to everyday interactions. In *CVPR*, 2018.

[12] 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.

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

[14] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[15] 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)*, pages 2046–2054, 2020. doi: 10.1109/WACV45572.2020.9093294.

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

[17] Zihang Lai, Erika Lu, and Weidi Xie. MAST: A memory-augmented self-supervised tracker. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[18] Rui Li and Dong Liu. Spatial-then-temporal self-supervised learning for video correspondence, 2022.

[19] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, 2014. ISBN 978-3-319-10602-1.

[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)*, pages 8957–8967, 2020. doi: 10.1109/CVPR42600.2020.00898.

[22] Marcelo Mendonça and Luciano Oliveira. Isec: Iterative over-segmentation via edge clustering. *Image and Vision Computing*, 80:45–57, 2018. ISSN 0262-8856. doi: https://doi.org/10.1016/j.imavis.2018.09.015.

[23] Bo Miao, Mohammed Bennamoun, Yongsheng Gao, and Ajmal Mian. Self-supervised video object segmentation by motion-aware mask propagation. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2022. doi: 10.1109/ICME52920.2022.9859966.

[24] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard 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.

[25] Sundaram Muthu, Ruwan Tennakoon, Tharindu Rathnayake, Reza Hoseinnezhad, David Suter, and Alireza Bab-Hadiashar. Motion segmentation of rgb-d sequences: Combining semantic and motion information using statistical inference. *IEEE Transactions on Image Processing*, 29:5557–5570, 2020. doi: 10.1109/TIP.2020.2984893.

[26] 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.

[27] 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 Conference on Computer Vision and Pattern Recognition*, pages 7376–7385, 2018. doi: 10.1109/CVPR.2018.00770.

[28] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[29] Matthieu Paul, Martin Danelljan, Christoph Mayer, and Luc Van Gool. Robust visual tracking by segmentation. In *Computer Vision – ECCV 2022*, pages 571–588, 2022. ISBN 978-3-031-20047-2.

[30] 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.

[31] 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.

[32] Hongje Seong, Seoung Wug Oh, Joon-Young Lee, Seongwon Lee, Suhyeon Lee, and Euntai Kim. Hierarchical memory matching network for video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12889–12898, 2021.

[33] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):717–729, 2016. doi: 10.1109/TPAMI.2015.2465960.

[34] Jayesh Vaidya, Arulkumar Subramaniam, and Anurag Mittal. Co-segmentation aided two-stream architecture for video captioning. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2442–2452, 2022. doi: 10.1109/WACV51458.2022.00250.

[35] Jack Valmadre, Luca Bertinetto, Joao F. Henriques, Ran Tao, Andrea Vedaldi, Arnold W.M. Smeulders, Philip H.S. Torr, and Efstratios Gavves. Long-term tracking in the wild: a benchmark. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[36] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[37] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L. Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9473–9482, 2019. doi: 10.1109/CVPR.2019.00971.

[38] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *Computer Vision – ECCV 2018: 15th European Conference*, page 402–419, 2018. ISBN 978-3-030-01260-1. doi: 10.1007/978-3-030-01261-8_24.

[39] Urs Waldmann, Jannik Bamberger, Ole Johannsen, Oliver Deussen, and Bastian Goldlücke. Improving unsupervised label propagation for pose tracking and video object segmentation. In *Pattern Recognition*, pages 230–245, 2022. ISBN 978-3-031-16788-1.

[40] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019.

[41] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *Computer Vision – ECCV 2018: 15th European Conference*, page 603–619, 2018. ISBN 978-3-030-01227-4. doi: 10.1007/978-3-030-01228-1_36.

[42] Xiaohao Xu, Jinglu Wang, Xiao Li, and Yan Lu. Reliable propagation-correction modulation for video object segmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(3):2946–2954, 2022. doi: 10.1609/aaai.v36i3.20200.

[43] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *Computer Vision – ECCV 2020: 16th European Conference*, page 332–348, 2020. ISBN 978-3-030-58557-0. doi: 10.1007/978-3-030-58558-7_20.

[44] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[45] Hangshi Zhong, Zhentao Tan, Bin Liu, Weihai Li, and Nenghai Yu. Ppml: Metric learning with prior probability for video object segmentation. In *2019 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4, 2019. doi: 10.1109/VCIP47243.2019.8965961.

[46] Wenjun Zhu, Jun Meng, and Li Xu. Self-supervised video object segmentation using integration-augmented attention. *Neurocomput.*, 455(C):325–339, 2021. ISSN 0925-2312. doi: 10.1016/j.neucom.2021.04.090.