STESA-GRU: Spatio-Temporal Self-Attended Gated Recurrent Units for Video Object Segmentation

Diego Valderrama
Universidad de Los Andes
df.valderrama@uniandes.edu.co

Mateo Rueda
Universidad de Los Andes
ms.rueda10@uniandes.edu.co

Abstract

Semi-supervised video object segmentation is a challenging task specially when multiple instances exist during the video. In this work, we propose a Spatio-Temporal Self-Attented Gated Recurrent Units (STESA-GRU) into an end-to-end off-line framework. In particular, we use 2D Convolutional GRU to find out the objects within a frame (spatial information) and to maintain the coherence of the predicted masks along the time (temporal information). Furthermore, the network is self-attented with a Non Local Block module to improve the segmentation quality of the objects along the sequences. Our proposed STESA-GRU achieved 0.429, 0.457 and .0443 in terms of region similarity ($\mathcal{J}$), contour accuracy ($\mathcal{F}$) and global score reaching comparable results to techniques not using online learning in DAVIS-2017.

1. Introduction

Since the rise of the deep learning, the neural networks have driven the computer vision leading the state-of-the-art in many fields of this branch of research including object detection and object segmentation on both static images and video clips problems. The development on computer vision have increased considerable due to the growing ability of the neural networks to get better results [13, 22] going from region predictions to pixel predictions which requires high pixel-wise level annotations as well as a delicate process in network designs.

Video Object Segmentation (VOS) is a task aim on separating objects from a video sequence background (Figure 1) to get a segmentation mask for each object along all frames [17]. Although video object tracking (VOT) is a task aim on generating objects bounding boxes, VOS is the combination of VOT and multiple object segmentation [28] which has several beneficial applications such as video editing, autonomous vehicles, scene understanding, etc.

Recently, the state-of-the-art methods are based on Convolutional Neural Networks (CNNs) and have been proposed to address the branches of this field: unsupervised VOS, semi-supervised VOS and interactive VOS. Semi-supervised VOS is a task which the algorithm takes as input the first frame ground-truth objects mask and has to predict the mask of the objects in the rest of the frames [18]. This task has been faced by two different techniques [28]. On one hand, there are some motion-based approaches [4, 9, 11, 12] which capture the temporal coherence of the motion of the object. On the other hand, some methods are detection-based [3, 14, 15] which do not take into account the temporal information.

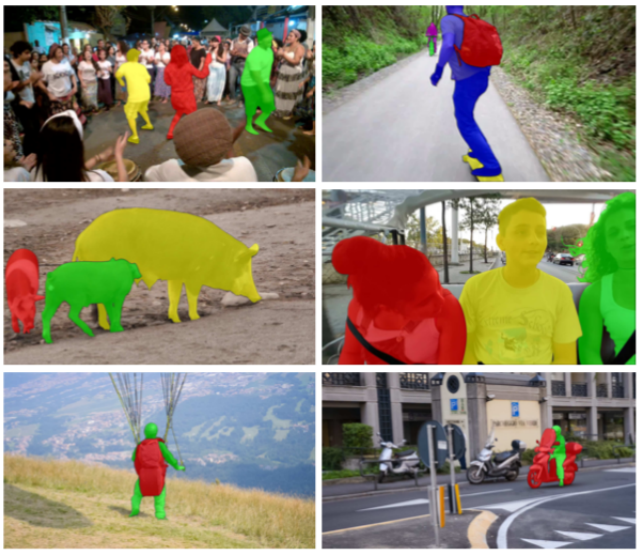

Figure 1. Example annotations of DAVIS 2017 dataset [18] for semi-supervised object segmentation.

In this work, to tackle this challenging problem we propose a Spatio-Temporal Self-Attented Gated Recurrent Units (STESA-GRU) for semi-supervised video object segmentation (Figure 2). Our model is based on [20], a recurrent model that predicts a mask for each object individually and then combine them to get the whole prediction for unsupervised and semi-supervised video object segmentation task. Thanks to the spatio-temporal recurrence and the RNN

memory, the model can predict masks for each object taking into account the differences between frames of the same object and the total number of instances per frame without the need of post-processing making our method end-to-end trainable.

Our model neglects the long-term memory to use a short-term memory due to the sequences on DAVIS dataset [18] are not so long. Additionally, we add a self-attention module to take advantage of the spatial in the intermediate layers. To this purpose, we force the model to learn using a new factor in our cost function. Experimental results in Davis 2017 [18] benchmark, show that our method reach comparable results to techniques not using online learning

2. Related work

The breakthrough in static image segmentation through the use of convolutional neural networks (CNNs) has been a starting point for video object segmentation in semi-supervised approximations. Among the techniques that use this type of networks, two large groups can be distinguished: detection-based and motion based.

2.1. Detection based methods

These methods do not use temporary information and learn a model that performs pixel-level detection and segmentation of the object of interest in each of the frames. They are based on fine-tuning a deep convolutional network using the first frame as an annotation.

One-Shot Video Object Segmentation (OSVOS) [3] was the first work under this technique and it is based on a fully-convolutional neural network (FCN) architecture. This technique can transfer generic semantic information (learned on ImageNet) to the task of foreground segmentation to learn the appearance of a single annotated object of the test sequence. Although all frames are processed independently, the results are temporally coherent and stable. However, at test time OSVOS just uses the fine-tuned network and is not able to adapt to large changes in object appearance. To overcome this limitation, Online Adaptive Video Object Segmentation (OnAVOS) [23] updates the network online using training examples selected based on the confidence of the network and the spatial configuration. Additionally, a pretraining step based on objectness is added, which is learned on PASCAL.

These methods have great results in Video Object Segmentation (VOS). However, the problem is that these methods are online and require fine-tuning which is time-consuming and computationally expensive, hence the algorithms are far from real time. Additionally, detection-based approaches do not model any temporary information which is proved to have advantage in recent methods [24, 25].

2.2. Motion-based methods

Motion-based methods are those that use the temporary consistency of object moving and formulate the problem as a mask propagation starting from a frame annotated toward the subsequent. In motion-based methods two large groups can be distinguished: optical flow based and RNN-based.

The first class of these methods are those that use optical flow. This technique represents how and where each pixel in the image will move in the next frame in order to establish a temporary consistency [4, 9, 10, 12, 26]. Cheng *et al*. proposed Segflow [4] which use two branches for video object segmentation, a segmentation branch based on a binary-type fully-convolutional network (Resnet-101 [7]) to segment (foreground and background) and another branch of optical flow using the network FlowNet [6]. PremVOS [9] instead of predicting masks directly from video pixels as done in [4, 10, 26] detect regions of interest using an object detection network, and then predict accurate masks only on the cropped and resized bounding boxes. PremVOS, also generates proposals independently for each frame and joins them using optical flow-based proposal warping, ReID embeddings and objectness scores, as well as taking into account the presence of other objects in the multi-object VOS scenario.

In optical flow methods the pixel displacement is usually small, since optical flow estimation is operated on two adjacent frames. However, in VOS, the pixel displacements could be very large because the target object could be located at any position in the target frame. Consequently, the optical flow needs to be operated densely to cover the whole feature map in VOS. This task is computationally costly and time-consuming to train.

On the other hand, there are also some approaches that use Recurrent Neural Networks (RNN). The objective is to perform temporal consistency by the propagation mask between frames and be able to learn long-term dependency from sequential data. MaskRNN [8] build a RNN approach which fuses on each frame the output of a binary segmentation net and a localization net with optical flow. This method can take advantage of long-term temporal structures of the video data as well as rejecting outliers. Another example is the method Recurrent network for multiple object Video Object Segmentation (RVOS [20]) that is fully end-to-end trainable. This model incorporates recurrence on two different domains: (i) the spatial, which allows to discover the different object instances within a frame, and (ii) the temporal, which allows to keep the coherence of the segmented objects along time. RVOS was adapted for one-shot video object segmentation by using the masks obtained in previous time steps as inputs to be processed by the recurrent module. Moreover, this model achieves faster inference runtimes than previous methods, reaching 44ms/frame on a P100 GPU [20].

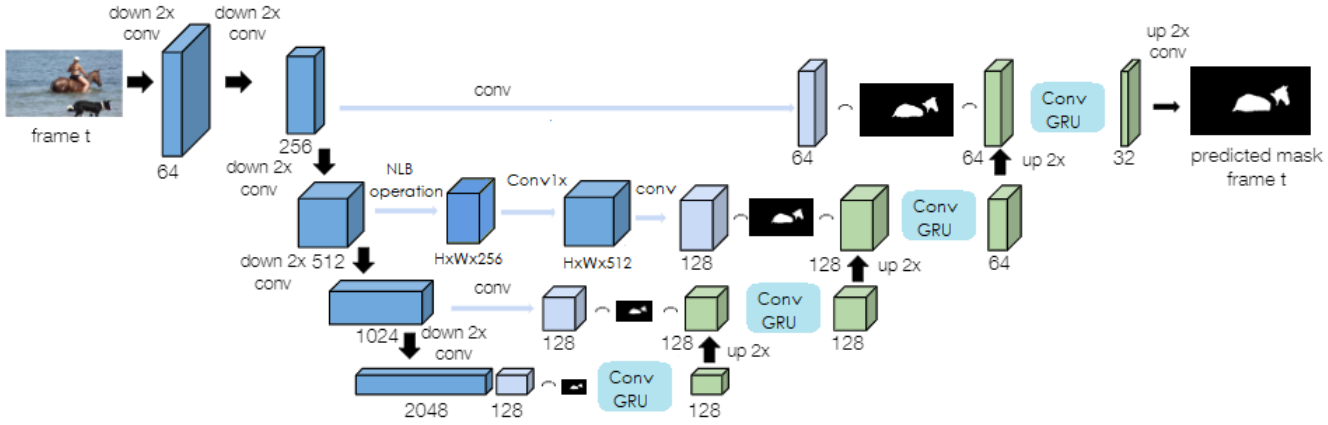

Figure 2. Our proposed architecture. The network receives as input an image and encode some features using ResNet101 as backbone [7]. During the decoding process, a NLB module after res3 is applied just before the skips connections and the spatio-temporal recurrence works. The figure illustrates a single forward of the decoder.

Our model is based on RVOS approach because it is a light and fast model in contrast to detection based models. Additionally, temporal and spatial recurrence is modeled along the entire video which provides important clues to infer the segmentation mask in the following frames. However, instead of modeling temporal recurrence using LSTM we incorporate we modeled GRU (Gated Recurrence Unit) which is computationally more efficient (as they contain less tunable parameters) [5]. Additionally, we add NLB (Non-local block) which is a recent technique to model spatio-temporal recurrence and can be incorporated in any part of the network without changing the features dimension and also keeping the model light and fast [25].

3. Baseline

RVOS [20] one-shot object segmentation was implemented as a baseline. The encoder proposed by the authors consists of a ResNet-101 model pre-trained on ImageNet. This architecture does instance segmentation by predicting a sequence of masks. In the encoder, the input x_t of the encoder is an RGB image, which is the frame t in the video sequence. The output corresponds to f_t = {$f_{t,1}$, $f_{t,2}$, ..., $f_{t,k}$} which are a set of features at different resolutions.

The decoder is designed as a hierarchical recurrent network architecture of ConvLSTMs which can leverage the different resolutions of the features f_t ={$f_{t,1}$, $f_{t,2}$, ..., $f_{t,k}$} extracted at a level k on the encoder for a frame t in the video sequence. The output of the decoder is a set of objects St_i which is a set of object predictions {$S_{t,1}$, , ..., $S_{t,i}$, ..., $S_{t,N}$}, where $S_{t,i}$ is the segmentation of object i at frame t. The recurrence in temporal domain has a special feature which is that the mask predicted for the same object at different frames has the same index in the spatial recurrence. Hence, the number of object segmentation predictions in the decoder is constant along all sequence so if one object disappears in a sequence, the segmentation mask will be empty at the following frames. The cost function is calculated using the Hungarian algorithm with a soft intersection over union (IoU) score.

Additionally, RVOS uses in the decoder a spatiotemporal recurrence so the output $h_{t,i,k}$ of the $k-th$ ConvLSTM layer for the object i at frame t depends on many aspects: 1) the features f_t obtained from the encoder in the frame t. 2) the preceding ConvLSTM layer. 3) The hidden state representation from the previous object $i1$ at the same frame t, i.e. $h_{t,i1,k}$, which corresponds to the spatial hidden state. 4) the hidden state representation from the same object i at the previous frame $t1$, i.e. $h_{t1,i,k}$, which will be referred to as the temporal hidden state. 5) The object segmentation prediction mask $S_{t1,i}$ of the object i at the previous frame t–1. In summary, the output $h_{t,i,k}$ is obtained using:

$$h_{input} = [B2(h_{t1,i,k})|f'_{t,k}|S_{t1,i}] \quad (1)$$

$$h_{state} = [h_{t,i1,k}|h_{t1,i,k}] \quad (2)$$

$$h_{t,i,k} = ConvLSTM_k(h_{input}, h_{state}) \quad (3)$$

where B2 is the bilinear upsampling operator by a factor of 2 and $f'_{t,k}$ is the result of projecting $f_{t,k}$ to a lower dimensionality. The equation 3 is applied repeatedly as the number of blocks in the encoder.

4. STESA-GRU

Our model is based on RVOS [20] approach with some modifications in order to bost the performance of the original method. GRU recurrence is used instead of LSTM units

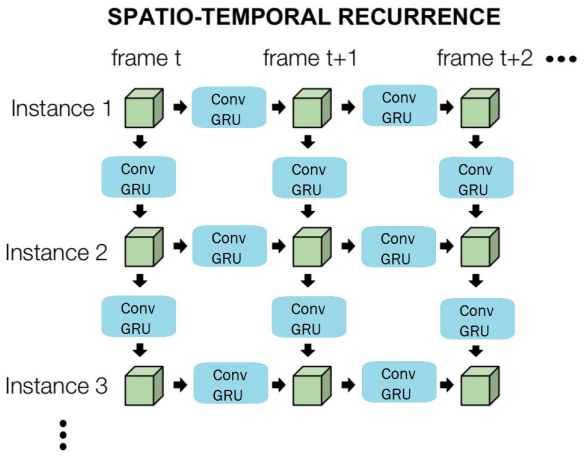

Figure 3. Proposed convolutional spatio-temporal GRU.

and we also add a self-attention module. Figure 2 shows the architecture of our implementation

4.1. Conv Spatio-Temporal GRU

As well as the Conv Spatio-Temporal LSTM proposed in [19] and the Spatio-Temporal GRU used by Altaf *et al*. [1], our Conv Spatio-Temporal GRU (Fig 3) takes into account two hidden states (spatial hS and temporal hT) to perform the recurrence. We define the update gate (u_g) and the reset gate (r_g) using a convolutional layer based on the hidden size and a sigmoid function. The current memory $\tilde{h_t}$ is define as hyperbolic tangent of the sum of the reset hidden r_h which is the matrix product between the reset gate and the sum of the previous hidden states (hS_{t-1} hT_{t-1}), and the current c which is also convolutional layer. Our implementation can be written as follows:

$$r_g = \sigma\Big(W_{xr}x_t + W_{hr}h_{t-1} + b_r\Big) \tag{4}$$

$$u_g = \sigma\Big(W_{xu}x_t + W_{hu}h_{t-1} + b_u\Big) \tag{5}$$

$$c = \sigma\Big(W_{xc}x_t + b_c\Big) \tag{6}$$

$$r_h = r_g * \Big(hS_{t-1} + hT_{t-1}\Big) \tag{7}$$

$$\tilde{h_t} = tanh\Big(c + r_h\Big) \tag{8}$$

$$h = (u_g * hS_{t-1}) + (1 - u_g) * \tilde{h_t} \tag{9}$$

Where * means matrix product and h the new hidden state

4.2. Self-Attention

The self attention part of our model consists of a two dimensional Non Local Block (NLB) module [25] with a Gaussian noise. To take advantage of the spatial information and following the suggestion of the authors, we add the NLB after $res3$ just before the skip connections. Moreover, to help the model to learn, we modify the cost function. While RVOS [20] uses the mIoU score between the ground-truth and the final prediction ($mIoU_P$), STESA-GRU add a new factor which is the mIoU score between the prediction get after the NLB module and the ground-truth at the corresponding scale ($mIoU_{NLB}$) to force the model to learn a way to take advantage of the NLB weights. Our loss function can be written as follows:

$$Loss = mIoU_{NLB} + mIoU_P \tag{10}$$

5. Experiments

5.1. Benchmarks

To train and evaluate our model, we use the two largest video object segmentation datasets using the respectively splits for each step.The metrics to evaluate the performance of our method are those proposed in [17]. The region similarity $\mathcal{J}$ is calculated as the average IoU between the proposed masks and the ground-truth masks. F-measure score $\mathcal{F}$ is calculated as the average boundary similarity measure between the bounders of the ground-truth and proposed masks. The general accuracy is the average between $\mathcal{J}$ and $\mathcal{F}$.

5.1.1 YouTube-VOS

YouTube Video Object Segmentation benchmark [27] is the largest scale dataset which consist of 3471 videos for training and 474 videos for validation with more than 190k annotations. The dataset includes videos of animals, vehicles, accessories, common objects and humans in various activities with one or more instances per video. We use the Youtube-VOS train set (seen categories) to pretrain our model due to it is the more challenging dataset on this task.

5.1.2 DAVIS - 2017

Densely-Annotated Video Segmentation [18] provides a dataset with 150 high definition videos with all frames annotated with pixel-wise object masks (Fig 1 shows some examples). We use DAVIS 2017 data [18] to fine-tune and evaluate our method. This dataset provides 60 videos in the train set, 30 videos in the val set and 30 videos in test-dev whose predictions are submitted into Codalab server.

Table 1 shows the results on DAVIS 2017 test-dev set compare the state-of-the-art methods with different of our configurations: STESA-GRU (Ours), STESA-GRU without the auxiliary loss (Ours w/o AL), RVOS model without

Model	OL	$\mathcal{J}-\mathcal{F}$ mean	$\mathcal{J}$	$\mathcal{F}$
PremVOS [13]	Yes	71,6	67,5	75,7
MRF [2]	Yes	67,5	64,5	70,5
OnAVOS [22]	Yes	56,5	53,4	59,6
FeelVOS [21]	No	57,8	55,2	60,5
RGMP [16]	No	52,9	51,4	54,4
RVOS [20]	No	50,3	47,9	52,6
Ours	No	42.6	40.6	44.6
Ours w/o AL	No	44.3	42.9	45.7
$\text{RVOS}_{w/o}$	No	33,6	32,1	35,0
$\text{Ours}_{w/o}$	No	36,1	33,9	38,4

Table 1. Comparison against state-of-the-art results for semi-supervised video object segmentation in DAVIS-2017 test-dev. OL refers to online learning. $RVOS_{w/o}$ and $Ours_{w/o}$ are the models without the pre-training on YouTube-VOS dataset. Ours w/o AL refers to our model pretrained without the auxiliary loss

Model	FT	$\mathcal{J}-\mathcal{F}$ mean	$\mathcal{J}$	$\mathcal{F}$
Ours	No	41.5	40.1	42.9
Ours w/o AL	No	42.9	40.8	44.9
Ours	Yes	42.6	40.6	44.6
Ours w/o AL	Yes	44.3	42.9	45.7

Table 2. Comparison between model which are just pretrained and those that are fine-tuned in DAVIS dataset.All the results are on DAVIS-2017 test-dev set. FT refers to fine-tuning on DAVIS dataset.Ours w/o AL refers to our model without the auxiliary loss

pre-training ($RVOS_{w/o}$) and STESA-GRU without pre-training ($Ours_{w/o}$). Moreover, the results presented on Table 2 illustrates that fine-tuning step was necessary to improve the performance in some points.

5.2. Qualitative Results

Figure 8 illustrates some qualitative results. As can be seen there are some videos in which the model can predict good masks for all objects in the video. However, there are some failures cases in which the model cannot tackle oclusions and fast movements.

Figure 5 shows the pre-training loss in Youtube-VOS considering the model without auxiliary loss. The loss is decreasing as epochs increase. A plateau is never reached thus pre-training with more epochs may lead to better results during fine-tuning in DAVIS2017. Moreover, the results of the model with and without auxiliary loss cannot be directly compared since the pre-training phase was performed with different number of epochs (Figure 6, 7). Pre-training needs to be performed until reaching a loss plateau phase to obtain the best results in DAVIS2017 dataset.

Figure 6 shows that training loss at DAVIS2017 in the model without auxiliary loss is decreasing as epochs increase in number. The train loss is greater than the validation loss so we can conclude that there is no over-fitting in our model. However, it should be noted that the validation loss does not decrease as we expected, probably exploring different learning rates the loss could better its performance in the validation set. Likewise, the weights were initialized under the pre-trained model in the Youtube-VOS database, so it is quite likely that pre-training with more epochs on Youtube-VOS will improve the loss decreasing behavior in both train and validation sets. Figure 7 shows that training with more than 30 epochs leads to over-fitting in the model. Additionally, the model without auxiliary loss started with a lesser loss than the model with it since the Youtube-VOS pre-training was performed with a higher number of epochs(Figure 6, 7).

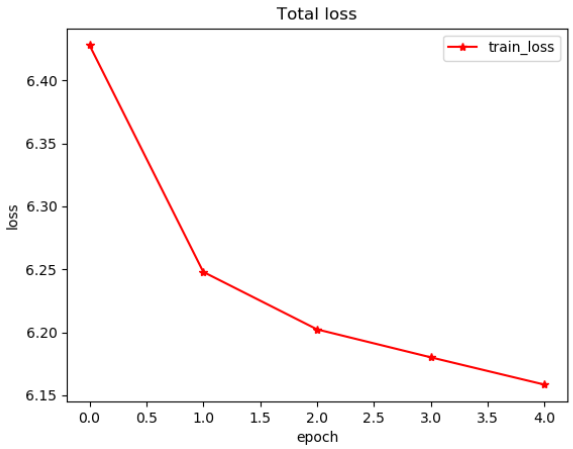

Figure 4. Loss curve during pre-training in Youtube-VOS with the auxiliary loss

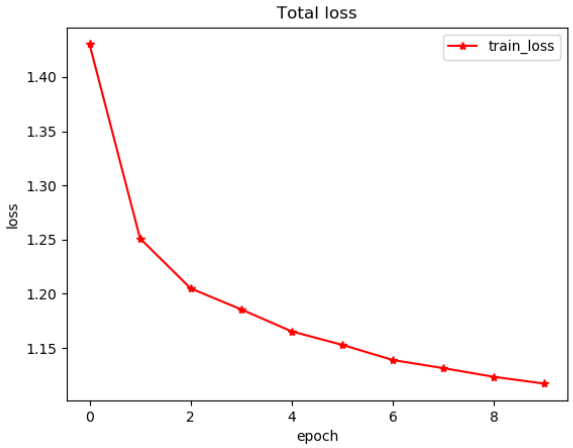

Figure 5. Loss curve during pre-training in Youtube-VOS without the auxiliary loss

6. Ablation study

The models used in the ablation study were not pre-trained on YouTube-VOS and was tested only in DAVIS-

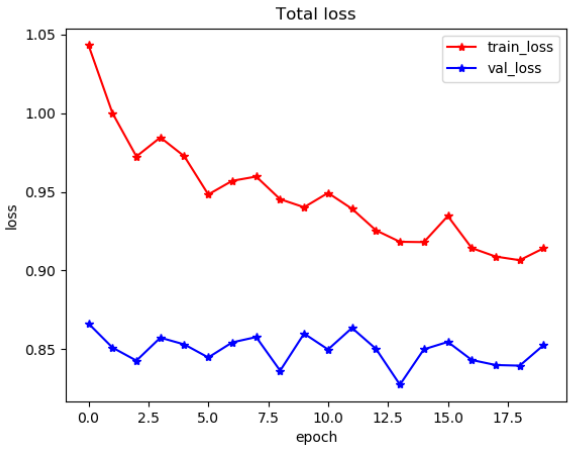

Figure 6. Loss curve of the STESA-GRU without the auxiliary loss during fine-tuning in DAVIS-2017 dataset

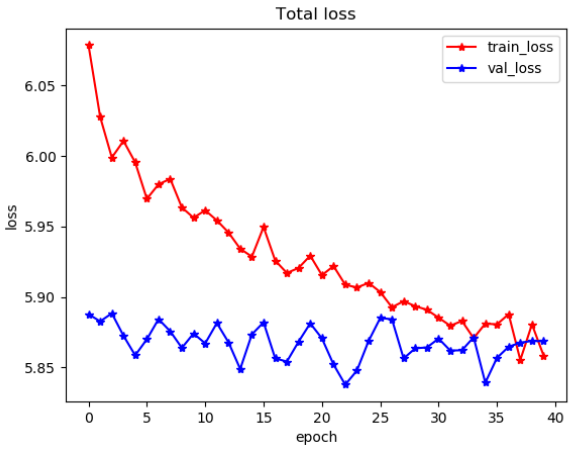

Figure 7. Loss curve of the STESA-GRU with the auxiliary loss during fine-tuning in DAVIS-2017 dataset

2017 test-dev set.

6.1. Number of reset gates

Since the model has two hidden states (spatial and temporal), some experiments were carried out to analyze the difference of using just one single reset gate for both hidden states or two reset gates for each one of the states. As can be seen (Table 3), our model outperforms the baseline (without pre-training) by 2 points which is expected due to DAVIS sequences are not so long. Contrary to what was expected, one reset gate that considers both hidden states adding them performs better in this task.

6.2. Self-Attention

We experiment with different positions of the NLB in the encoder. Our best results were obtained setting the NLB in the third convolutional block in the encoder (Table 4) using ConvGRU. The addition of this block only increase the memory usage in approximately 1GB. This NLB implementation surprisingly decreased the performance of the LSTM compared with the baseline results and were worse compared with GRU implementation.

Model	r_g	$\mathcal{J}-\mathcal{F}$ mean	$\mathcal{J}$	$\mathcal{F}$
Ours w/o NLB	1	**35,6**	**34,2**	**37,0**
Ours w/o NLB	2	31,7	30,1	33,3

Table 3. Ablation study about number of reset gates (r_g) on GRU implementation in DAVIS 2017 dataset without pre-training in YouTube-VOS.

Recurrence	NLB	$\mathcal{J}-\mathcal{F}$ mean	$\mathcal{J}$	$\mathcal{F}$
GRU	Res3	**35,4**	**33,5**	**37,4**
GRU	Res4	34,6	32,4	36,8
GRU	Res5	30,4	28,9	32,0
LSTM	Res3	31,7	30,0	33,4
LSTM	Res4	31,2	29,8	32,5
LSTM	Res5	32,1	30,6	33,7

Table 4. Ablation study about the effect of using the NLB module after different parts of the decoder architecture with different kind of recurrence in DAVIS 2017 dataset without pre-training in YouTube-VOS

Model	NLB	$\mathcal{J}-\mathcal{F}$ mean	$\mathcal{J}$	$\mathcal{F}$
STESA-GRU	Res3	**36,1**	**33,9**	**38,4**
STESA-GRU$_2$	Res4	33,7	31,9	35,6

Table 5. Ablation study about the effect of using the NLB module with auxiliary loss after different positions in DAVIS 2017 dataset without pre-training in YouTube-VOS

Due to the best performance obtained using NLB was lower than the performance without using only this module, we add an auxiliary loss which measures the IoU score between the predicted mask for all instances at level k and the ground-truth with the same size of the predicted mask. More specifically, the level that we chose is the same where the NLB module is used in the encoder. Then, this auxiliary loss was sum IoU score between the ground-truth and the final prediction (equals as in RVOS implementation) in order to force the model to learn some weights related with the NLB module. Table 5 shows the results of this experiment and as was expected, adding this loss improve the performance of the method.

Training details The original RGB frames and annotations have been resized to 256x448 in order to have a fair comparison with RVOS [20] in terms of image resolution. During training, each mini-batch is composed with 3 or 4 clips of 5 consecutive frames for pretrained models and without the pretrained step, respectively. The initial learning rate was set to 10^{-6} with an Adam optimizer both in

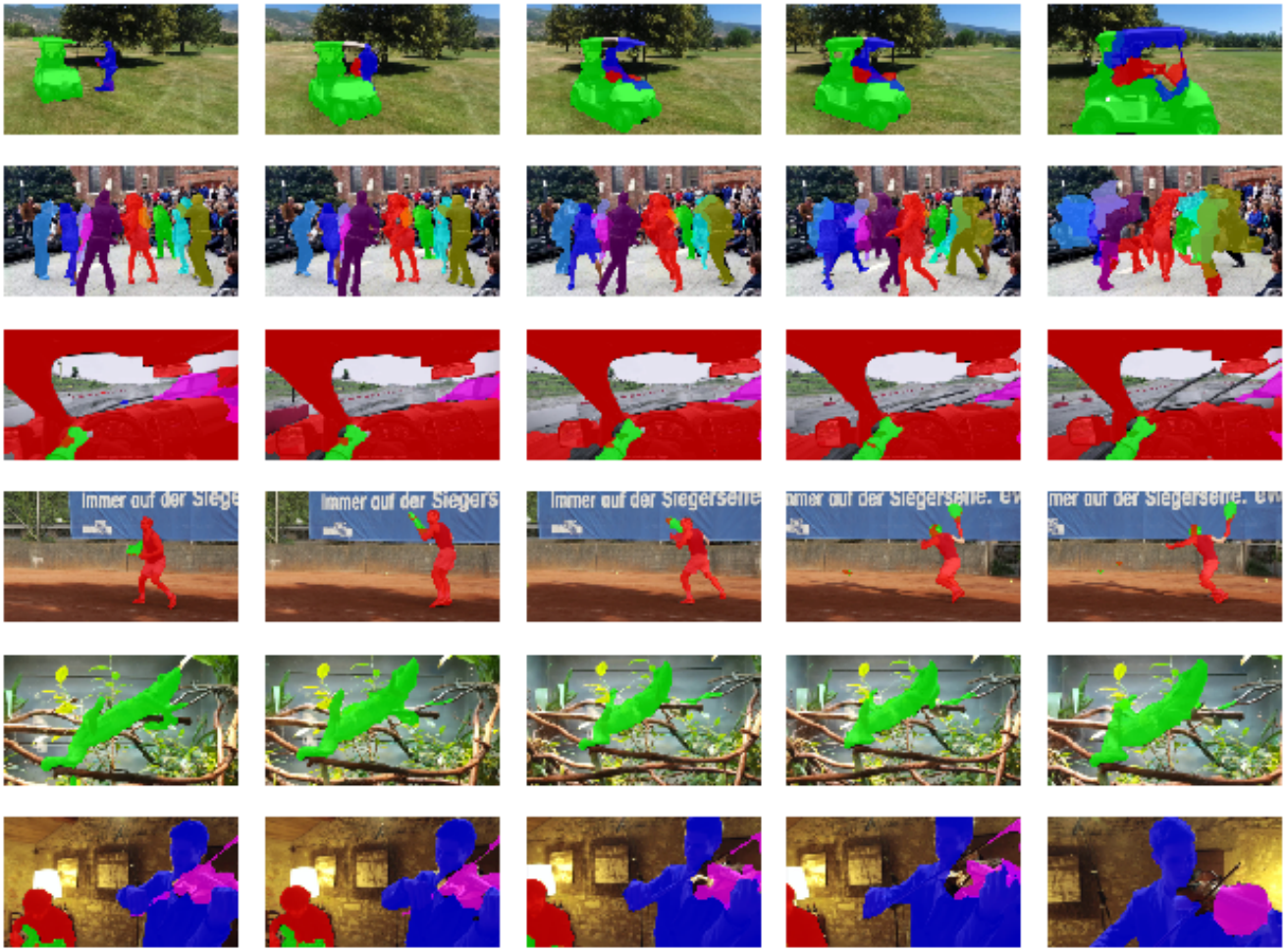

Figure 8. Qualitative results in DAVIS-2017. Frames are sampled at important moments (*e.g* before and after occlusions)

the pretraining and training phase. The models were pretrained in Youtube-VOS with 10 epochs for STESA-GRU w/o AL model and 5 epochs for STESA-GRU model. The pretraining with 10 epochs lasted aproximately 48 hours. We later train these models for 20 epochs in DAVIS2017 taking about 10 hours. The models with NLB but without auxiliary loss were trained during 40 epochs (19 hours). All models were run in a single GPU Tesla K40C with 12GB RAM.

7. Conclusions

In this work, we have presented a fully end-to-end trainable spatio-temporal recurrent network with a self-attention module. Our method reach comparable results to methods which not use online learning. We believe that augmenting the number of epochs during pre-training in YouTube-VOS will improve the performance of our model. As future work, we will prove to train the model using different values of learning rate and we will also test our method in validation set of YouTube-VOS dataset.

References

[1] B. Altaf, L. Yu, and X. Zhang. Spatio-temporal attention based recurrent neural network for next location prediction. *2018 IEEE International Conference on Big Data (Big Data)*, pages 937–942, 2018.

[2] L. Bao, B. Wu, and W. Liu. CNN in MRF: video object segmentation via inference in A cnn-based higher-order spatio-temporal MRF. *CoRR*, abs/1803.09453, 2018.

[3] S. Caelles, K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. V. Gool. One-shot video object segmentation. *CoRR*, abs/1611.05198, 2016.

[4] J. Cheng, Y. Tsai, S. Wang, and M. Yang. Segflow: Joint learning for video object segmentation and optical flow. *CoRR*, abs/1709.06750, 2017.

[5] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[6] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. *CoRR*, abs/1504.06852, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[8] Y.-T. Hu, J.-B. Huang, and A. Schwing. Maskrnn: Instance level video object segmentation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 325–334. Curran Associates, Inc., 2017.

[9] B. L. J. Luiten, P. Voigtlaender. Premvos: Proposal-generation, refinement and merging for the davis challenge on video object segmentation 2018. *The 2018 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*, 2018.

[10] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele. Lucid data dreaming for video object segmentation. *International Journal of Computer Vision*, 127(9):1175–1197, Sep 2019.

[11] A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. *CoRR*, abs/1612.02646, 2016.

[12] X. Li and C. C. Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. *CoRR*, abs/1803.04242, 2018.

[13] J. Luiten, P. Voigtlaender, and B. Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. *CoRR*, abs/1807.09190, 2018.

[14] K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. Video object segmentation without temporal information. *IEEE transactions on pattern analysis and machine intelligence*, 41(6):1515–1530, 2018.

[15] A. Newswanger and C. Xu. One-shot video object segmentation with iterative online fine-tuning. *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*, 2017.

[16] S. W. Oh, J. 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*, pages 7376–7385, June 2018.

[17] 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.

[18] 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.

[19] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *CoRR*, abs/1506.04214, 2015.

[20] C. Ventura, M. Bellver, A. Girbau, A. Salvador, F. Marques, and X. G. i Nieto. Rvos: End-to-end recurrent network for video object segmentation, 2019.

[21] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L. Chen. FEELVOS: fast end-to-end embedding learning for video object segmentation. *CoRR*, abs/1902.09513, 2019.

[22] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. *CoRR*, abs/1706.09364, 2017.

[23] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. *CoRR*, abs/1706.09364, 2017.

[24] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool. Temporal segment networks: Towards good practices for deep action recognition. *CoRR*, abs/1608.00859, 2016.

[25] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.

[26] H. Xiao, J. Feng, G. Lin, Y. Liu, and M. Zhang. Monet: Deep motion exploitation for video object segmentation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1140–1148, June 2018.

[27] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. S. Huang. Youtube-vos: A large-scale video object segmentation benchmark. *CoRR*, abs/1809.03327, 2018.

[28] R. Yao, G. Lin, S. Xia, J. Zhao, and Y. Zhou. Video object segmentation and tracking: A survey. *arXiv preprint arXiv:1904.09172*, 2019.