Revisiting Sequence-to-Sequence Video Object Segmentation with Multi-Task Loss and Skip-Memory

Fatemeh Azimi*,[1,2], Benjamin Bischke*,[1,2], Sebastian Palacio[1,2], Federico Raue[2], Jörn Hees[2], Andreas Dengel[1,2]
[1]DFKI GmbH
[2]TU Kaiserslautern
Kaiserslautern, Germany
firstname.lastname@dfki.de
*Equal Contribution

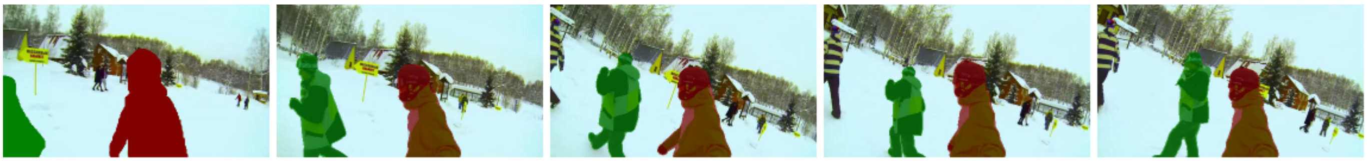

Figure 1. In one-shot video object segmentation, the binary masks of the first appearance of the target objects are provided. The task is to segment these objects throughout the whole sequence. Here the first image on the left shows the initial frame and masks. The other segmentation masks are the outputs of our method for the next frames in the sequence. Every two frames are shown here for more coverage of the sequence.

***Abstract*—Video Object Segmentation (VOS) is an active research area of the visual domain. One of its fundamental subtasks is semi-supervised / one-shot learning: given only the segmentation mask for the first frame, the task is to provide pixel-accurate masks for the object over the rest of the sequence. Despite much progress in the last years, we noticed that many of the existing approaches lose objects in longer sequences, especially when the object is small or briefly occluded. In this work, we build upon a sequence-to-sequence approach that employs an encoder-decoder architecture together with a memory module for exploiting the sequential data. We further improve this approach by proposing a model that manipulates multi-scale spatio-temporal information using memory-equipped skip connections. Furthermore, we incorporate an auxiliary task based on distance classification which greatly enhances the quality of edges in segmentation masks. We compare our approach to the state of the art and show considerable improvement in the contour accuracy metric and the overall segmentation accuracy. Our source code is available via https://github.com/fatemehazimi990/RS2S.**

I. Introduction

One-shot Video Object Segmentation (VOS) is the task of segmenting an object of interest throughout a video sequence with many applications in areas such as autonomous systems and robotics. In this task, the first mask of the object appearance is provided and the model is supposed to segment that specific object during the rest of the sequence. VOS is a fundamental task in Computer Vision dealing with various challenges such as handling occlusion, tracking objects with different sizes and speed, and drastic motion either from the camera or the object [35]. Within the last few years, video object segmentation has received a lot of attention from the community [4], [20], [29], [31]. Although VOS has a long history [5], [8], [18], only recently it has resurfaced due to the release of large-scale and specialized datasets [21], [22].

To solve VOS, a wide variety of approaches have been proposed in the literature ranging from training with static images without using temporal information [4] to using optical flow for utilizing the motion information and achieving better temporal consistency [29]. However, the methods relying solely on static images lack temporal consistency and using optical flow is computationally expensive and imposes additional constraints.

With the release of YoutubeVOS [33], the largest video object segmentation dataset to date, the authors demonstrated that having enough labeled data makes it possible to train a sequence-to-sequence (S2S) model for video object segmentation. In S2S, an encoder-decoder architecture is used similar to [1]. Furthermore, a recurrent neural network (RNN) is employed after the encoder (referred to as bottleneck) to track the object of interest in a temporally coherent manner.

In this work, we build on top of the S2S architecture due to its simplicity and elegant design that reaches impressive results compared to the state of the art while remaining efficient [33]. In the YoutubeVOS dataset, there are sequences with up to five objects with various sizes to be segmented. By having a close look at the S2S behavior, we noticed that it often loses track of smaller objects. The problem in the failure cases is that early in the sequence, the network prediction of the segmentation masks has a lower confidence (the output of the *sigmoid* layer is around 0.5). This uncertainty increases and propagates rapidly to the next frames resulting in the model losing the object as shown in Figure 2. Therefore, the segmentation score of that object would be zero for the rest of the sequence which has a strong negative impact on the overall performance. We argue that this is partly due to lack of information in the bottleneck, especially for small objects.

To improve the capacity of the model for segmenting smaller objects, we propose utilizing spatio-temporal information at multiple scales. To this end, we propose using additional skip connections incorporating a memory module (henceforth referred to as skip-memory). Our intuition is based on the role of ConvLSTM in the architecture that is remembering the area of interest in the image. Using skip-memory allows the model to track the target object at multiple scales. This way, even if the object is lost at the bottleneck, there is still a chance to track it by using information from lower scales.

Our next contribution is the introduction of an auxiliary task for improving the performance of video object segmentation. The effectiveness of multi-task learning has been shown in different scenarios [25], but it has received less attention in video object segmentation. We borrow ideas from Bischke et al. [2] for satellite image segmentation and adapt it for the task at hand. The auxiliary task defined here is distance classification. For this purpose, a border class regarding the distance to the edge of the target object is assigned to each pixel in the ground-truth mask, providing fine-grained information about each pixel location relative to the object border. Learning the pixel distance with respect to the border, provides useful information for foreground-background segmentation (pixel inside or outside the object).

We adapt the decoder network with an extra branch for the additional distance classification mask and use its output as an additional training signal for predicting the segmentation mask. The distance classification objective provides more specific information about the precise location of each pixel, resulting in significant improvement of the F score (measuring the quality of the segmentation boundaries). The overall architecture is shown in Figure 3. In the rest of the paper we refer to our method as S2S++.

II. Related Work

One-shot video object segmentation can be seen as pixel-wise tracking of target objects throughout a video sequence, where the first segmentation mask is provided as shown in Figure 1. This field has a long history in the literature [3], however, with the rise of deep learning, classical methods based on energy minimization, using super-voxels and graph-based methods [6], [10], [19], [27] were replaced with deep learning based approaches. In the following we provide a brief overview of the state of the art approaches in this domain.

Having the first segmentation mask at hand, two training mechanisms exist in the literature: *offline training* which is the standard training process and *online training* which is performed at the test time. In online training, heavy augmentation is applied to the first frame of the sequence in order to generate more data and the network is further trained on the specific sequence [4], [17], [20], [31]. Using online training as an additional training step leads to better results, however, it makes the inference phase quite slow and computationally more demanding.

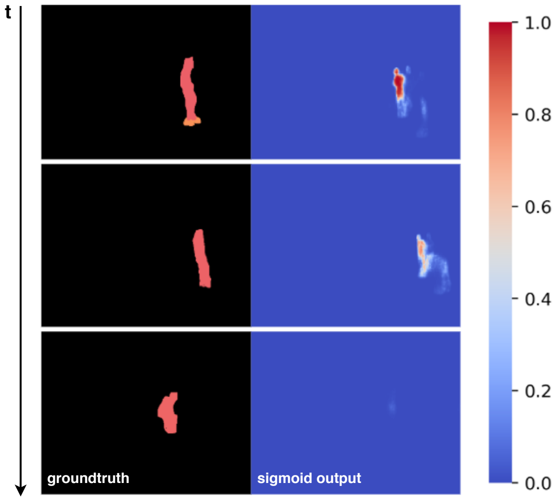

Figure 2. In this image, the ground-truth mask is shown on the left and the output of the decoder of the S2S architecture, on the right. The output of the *sigmoid* function (last layer in the decoder) acts like a probability distribution over the binary classification, measuring the model confidence. The output of around 0.5 (white color coding) implies a low confidence in the prediction while values close to 0 or 1 (blue and red colors) show confident outputs w.r.t. to background and foreground classes). Our observation is that the model is not often confident when predicting masks for small objects. This uncertainty propagates to the next predictions causing the model to lose the target object within a few time steps. We argue that part of this issue is because the RNN located in the bottleneck of the encoder-decoder architecture, does not receive enough information from the small objects.

Regarding offline training, various approaches have been suggested in the literature. Some approaches are based on using static images and extending image segmentation for video [4], [20]. In [4], authors use a VGG architecture [28] pre-trained on ImageNet [14] and adapt it for video object segmentation. Further offline and online training accompanied by contour snapping, allow the model to keep the object of interest and discard the rest of the image (classified as background). [20] treats the task as guided instance segmentation. In this case, the previous predicted mask (first mask in the beginning followed by using predicted masks at next time steps) is used as an additional input, serving as the guidance signal. Moreover, the authors experiment with different types of signals such as bounding boxes and optical flow demonstrating that even a weak guidance signal such as bounding box can be effective. In OSMN [34], the authors propose using two visual and spatial modulator networks to adapt the base network for segmenting only the object of interest. The main problem with these methods is that they do not utilize sequential data and therefore suffer from a lack of temporal consistency.

Another approach taken in the literature is using object proposals based on RCNN-based techniques [9]. In [16] the task is divided into two steps: First, generating the object proposals and second, selecting and fusing the promising mask

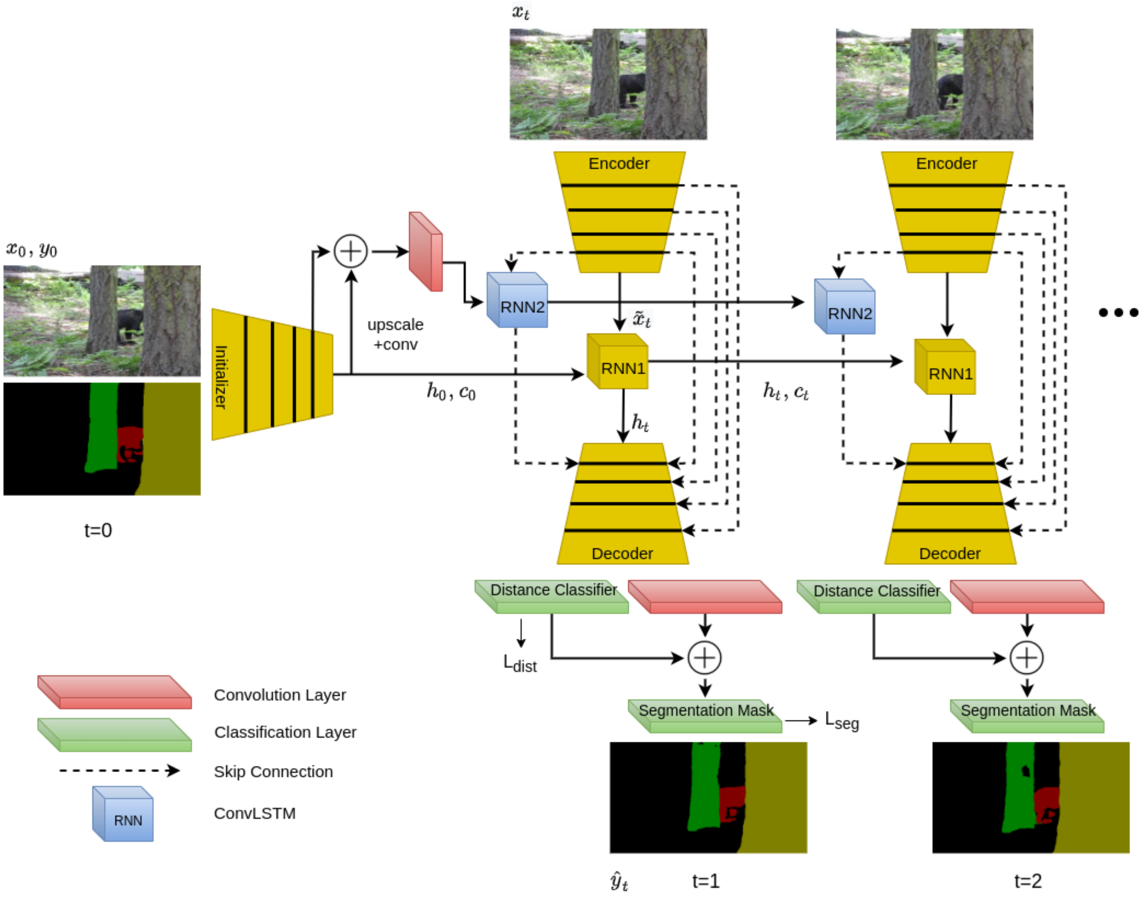

Figure 3. The overall architecture of our approach. We utilize information at different scales of the video by using skip-memory (RNN2). Experiments with more than one skip-memory connection are possible (only one is shown here for simplicity). We use an additional distance-based loss to improve the contour quality of the segmentation masks. For this purpose, a distance class is assigned to each pixel in the mask, based on its distance to the object boundary. We use a $softmax$ at the distance classification branch and a $sigmoid$ at the segmentation branch to compute the L_{dist} and L_{seg}, respectively. Yellow blocks show the architecture of the original S2S model (Equations (1) to (4)) and all other blocks depict our extension to this model.

proposals trying to enforce the temporal consistency by utilizing optical flow. This method achieve a good performance, with the downside of being quite complex and slow.

Before the introduction of a comprehensive dataset for VOS, it was customary to pre-train the model parameters on image segmentation datasets and then fine-tune on video data [31]. Khoreva et al. suggest an advanced data augmentation method for video segmentation including non-rigid motion, to address the lack of labeled data in this domain [11]. However, with the release of YoutubeVOS dataset [33], the authors show that it is possible to train an end-to-end, sequence-to-sequence model for video object segmentation when having enough labeled data. They deploy a ConvLSTM module [32] to process the sequential data and to maintain temporal consistency.

In [29], the authors propose a two-stream architecture composed of an appearance network and a motion network. The result of these two branches are merged and fed to a ConvGRU module before the final segmentation. [30] extends the spatial recurrence proposed for image instance segmentation [26] with temporal recurrence, designing an architecture for zero-shot video object segmentation (without using the first ground-truth mask).

In this paper, we focus on using ConvLSTM for processing sequential information at multiple scales, following the ideas in [33]. In the next sections, we elaborate on our method and proceed with the implementation details followed by experiments as well as an ablation study on the impact of different components and the choice of hyper-parameters.

III. Method

In this section we describe our method including the modifications to the S2S architecture and the use of our multi-task loss for video object segmentation. The S2S model is illustrated with yellow blocks in Figure 3 where the segmentation mask is computed as (adapted from [33]):

$$h_0, c_0 = Initializer(x_0, y_0) \tag{1}$$

$$\tilde{x}_t = Encoder(x_t) \tag{2}$$

$$h_t, c_t = RNN1(\tilde{x}_t, h_{t-1}, c_{t-1}) \tag{3}$$

$$\hat{y}_t = Decoder(h_t) \tag{4}$$

with x referring to the RGB image and y to the binary mask.

A. Integrating Skip-Memory Connections

We aim to better understand the role of the memory module used in the center of the encoder-decoder architecture in S2S method. To this end, we replaced the ConvLSTM with simply feeding the previous mask as guidance for predicting the next mask, similar to [20]. Doing so, we observed a drastic performance drop of about ten percent in the overall segmentation accuracy. This suggests that only the guidance signal from the previous segmentation mask is not enough and that features from the previous time step should be aligned to the current time step. As a result, we hypothesise that the role of ConvLSTM in the architecture is twofold: First, to remember the object of interest through the recurrent connections and the hidden state, and to mask out the rest of the scene and second, to align the features from the previous step to the current step, having a role similar to optical flow.

As mentioned earlier, the S2S model incorporates a memory module at the bottleneck of the encoder-decoder network. By inspecting the output of this approach, we noticed that the predicted masks for small objects are often worse than the other objects (see Figure 5 and Figure 6 for visual examples). The issue is that the target object often gets lost early in the sequence as shown in Figure 2. We reason that this is partially due to the lack of information for smaller objects in the bottleneck. For image segmentation, this issue is addressed via introducing skip connections between the encoder and the decoder [1], [24]. This way the information about small objects and fine details are directly passed to the decoder. Using skip connections is very effective in image segmentation; however, when working with video, if the information in the bottleneck (input to the memory) is lost, the memory concludes that there is no object of interest in the scene anymore (since the memory provides information about the target object and its location). As a result, the information in the simple skip connections will not be very helpful in this failure mode.

As a solution, we propose a system that keeps track of features at different scales of the spatio-temporal data by using a ConvLSTM in the skip connection as shown in Figure 3. We note that some technical considerations should be taken into account when employing ConvLSTM at higher image resolutions. As we move to higher resolutions (lower scales) in the video, the motion is larger and also the receptive field of the memory is smaller. As stated in [23], capturing the displacement is limited to the kernel size in kernel-based methods such as using ConvLSTMs. Therefore, adding ConvLSTMs at lower scales in the decoder, without paying attention to this aspect might have negative impact on the segmentation accuracy. Moreover, during our experiments we observed that it is important to keep the simple skip connections (without ConvLSTM) intact in order to preserve the uninterrupted flow of the gradients. Therefore, we add the ConvLSTM in an additional skip connection (RNN2 in Figure 3) and merge the information from different branches using weighted averaging with learnable weights. Hence, it is possible for the network to access information from different branches in an optimal way.

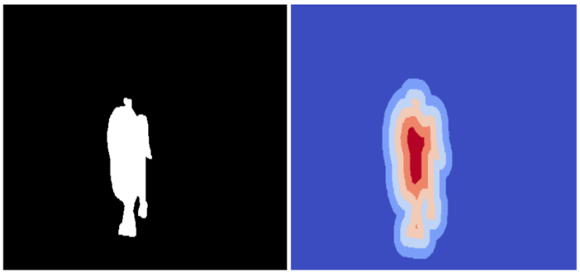

Figure 4. In this figure, we show a binary mask (left) together with a heat-map depicting the distance classes (right) as explained in Equation 6. The number of distance classes is determined by two hyper-parameters for the *number of border pixels* around the edges and the *bin size* for each class. The visualization shows that unlike previous works, our representations captures distance classes inside (reddish colors) as well as outside of the objects (blueish colors).

For the training objective of the segmentation branch in Figure 3, we use the sum of the balanced binary-cross-entropy loss [4] over the sequence of length T, defined as:

$$\begin{aligned} L_{seg}(W) = \sum_{t=1}^{T}(-\beta \sum_{j\in Y_+} logP(y_j = 1|X; \mathbf{W}) \\ - (1-\beta) \sum_{j\in Y_-} logP(y_j = 0|X; \mathbf{W})) \end{aligned} \tag{5}$$

where X is the input, $\mathbf{W}$ is the learned weights, Y_+ and Y_- are foreground and background labeled pixels, $\beta = |Y_-|/|Y|$, and Y is the total number of pixels.

B. Border Distance Mask and Multi-Task Objective

As the second extension, we build upon previous work of Bischke et al. [2] and train the network parameters in addition to the object segmentation mask with an image representation based on a distance transformation (see Figure 4 for an example). This image representation was successfully used in a multi-task learning setup to explicitly bias the model to focus more on those pixels which are close to the object boundary and more error prone for misclassification, compared to the ones further away from the edge of the object.

In order to derive this representation, we first apply the distance transform to the object segmentation mask. We truncate the distance at a given threshold to only incorporate the nearest pixels to the border. Let Q denote the set of pixels on the object boundary and C the set of pixels belonging to the object mask. For every pixel p we compute the truncated distance $D(p)$ as:

$$\begin{aligned} D(p) = \delta_p \;\inf\{ \min_{\forall q \in Q} d(p,q), R \;\}, \\ \text{where } \delta_p = \begin{cases} +1 & \text{if} \quad p \in C \\ -1 & \text{if} \quad p \notin C \end{cases} \end{aligned} \tag{6}$$

where $d(p, q)$ is the Euclidean distance between pixels p and q and R is the maximal radius (truncation threshold). The

Table I
COMPARISON OF OUR BEST-OBTAINED RESULTS WITH THE STATE OF THE ART APPROACHES IN VIDEO OBJECT SEGMENTATION USING YOUTUBEVOS DATASET. THE VALUES ARE REPORTED IN PERCENTAGES AND DIVIDED INTO COLUMNS FOR EACH SCORE AS IN [33]. THE TABLE IS DIVIDED TO TWO PARTS FOR METHODS WITH AND WITHOUT ONLINE TRAINING. WE CAN SEE THAT OUR APPROACH (EVEN WITHOUT ONLINE TRAINING) ACHIEVES THE BEST OVERALL SCORE.

Method	Online training	J_{seen}	J_{unseen}	F_{seen}	F_{unseen}	overall
OSVOS [4]	yes	59.8	**54.2**	60.5	**60.7**	**58.08**
MaskTrack [20]	yes	59.9	45.0	59.5	47.9	53.08
OnAVOS [31]	yes	**60.1**	46.6	**62.7**	51.4	55.20
OSMN [34]	No	60.0	40.6	60.1	44.0	51.18
RVOS [30]	No	63.6	45.5	67.2	51.0	56.83
S2S [33]	No	66.7	48.2	65.5	50.3	57.68
S2S++(ours)	No	**68.68**	**48.89**	**72.03**	**54.42**	**61.00**

pixel distances are additionally weighted by the *sign function* δ_p to represent whether pixels lie inside or outside objects. The continuous distance values are then uniformly quantized with a bin-size s into $\lfloor R/s \rfloor$ bins. Considering both inside and outside border pixels, this yields to $2 * R/s$ binned distance classes as well as two classes for pixel distances that exceeds the threshold R. We one-hot encode every pixel p of this image representation into k classification maps $D_K(p)$ corresponding each of the border distance.

We optimize the parameters of the network with a multi-task objective by combining the loss for the segmentation mask L_{seg} and the loss for the border distance mask L_{dist} as a weighted sum as follows. Since we consider a multi-class classification problem for the distance prediction task we use the cross-entropy loss. L_{dist} is defined as the cross entropy loss between the derived distance output representation $D_K(p)$ and the network output:

$$L_{total} = \lambda \ L_{seg} + (1 - \lambda) \ L_{dist} \tag{7}$$

The loss of the object segmentation task is the balanced binary-cross-entropy loss as defined in Equation 5. The network can be trained end-to-end.

IV. IMPLEMENTATION DETAILS

In this section we describe implementation details of our approach.

A. Initializer and Encoder Networks

The backbone of the initializer and the encoder networks in Figure 3 is a VGG16 [28] pre-trained on ImageNet [14]. The last layer of VGG is removed and the fully-connected layers are adapted to a convolution layer to form a fully convolutional architecture as suggested in [15]. The number of input channels for the initializer network is changed to 4, as it receives the RGB and the binary mask of the object as the input. The initializer network has two additional 1×1 convolution layers with 512 channels to generate the initial hidden and cell states of the ConvLSTM at the bottleneck (RNN1 in Figure 3). For initializing the ConvLSTMs at higher scales, up-sampling followed by convolution layers are utilized, with the same fashion as the decoder. Additional convolution layers are initialized with Xavier initialization [7].

B. RNNs

The ConvLSTM $1, 2$ (shown as RNN1 and RNN2 in Figure 3) both have a kernel size of 3×3 with 512 channels. The ConvLSTM at the next level has a kernel size of 5×5 with 256 channels. Here, we chose a bigger kernel size to account for capturing larger displacements at lower scales in the image pyramid.

C. Decoder

The decoder consists of five up-sampling layers with bilinear interpolation, each followed by a convolution layer with kernel size of 5×5 and Xavier initialization [7]. the number of channels for the convolution layers are $512, 256, 128, 64, 64$ respectively. The features from the skip connections and the skip-memory are merged using a 1×1 convolution layer. To adapt the decoder for the multi-task loss, an additional convolution layer is used to map 64 channels to the number of distance classes. This layer is followed by a $softmax$ to generate the distance class probabilities. The distance scores are merged into the segmentation branch where a $sigmoid$ layer is used to generate the binary segmentation masks.

We use the Adam optimizer [12] with an initial learning rate of 10^{-5}. In our experiments we set the value of λ in Equation 7 to 0.8. We choose this value by means of casual hyper-parameter tuning. When the training loss is stabilized, we decrease the learning rate by a factor of 0.99 every 4 epochs. Due to GPU memory limitations, we train our model with batch size 4 and a sequence length of 5 to 12 frames.

D. Border Output Representation

The number of border pixels and the bin size per class are hyper-parameters which determine the resulting number of distance classes. In our internal experiments (see Section V-B), we noticed better results can be achieved if the number of distance classes is increased. In the following experiments, we set the *border_pixels*=20, the *bin_size*=1. Thereby we obtain for each object a segmentation mask with 42 distance classes (the number of output classes is $2 \times \frac{border\_pixels}{bin\_size} + 2$). Having the edge as the center, we have $\frac{border\_pixels}{bin\_size}$ classes at each of the inner and outer borders plus two additional classes for pixels which do not lie within the borders (inside and outside of the object) as shown in Figure 4.

Table II

Ablation study on the impact of skip-memory and multi-task loss. We can notice that multi-task loss and skip-memory individually improve the results, but lead to the best results when combined.

Method	J_{seen}	J_{unseen}	F_{seen}	F_{unseen}	overall
base model	65.36	43.55	67.90	47.50	56.08
base model + multi-task loss	67.65	44.62	70.81	49.84	58.23
base model + one skip-memory	66.89	46.82	69.22	50.08	58.25
base model + one skip-memory + multi-task loss	67.18	47.04	70.24	52.30	59.19
base model + two skip-memory + multi-task loss	**68.68**	**48.89**	**72.03**	**54.42**	**61.00**

Table III

Results for different hyper-parameters for the multi-task loss on our best model. We can see that a higher number of distance classes slightly improves the metrics.

border size	bin size	number of classes	J_{seen}	J_{unseen}	F_{seen}	F_{unseen}	overall
20	10	6	68.37	47.68	71.54	52.38	59.99
20	1	42	**68.68**	**48.89**	**72.03**	**54.42**	**61.00**
10	1	22	68.40	47.91	71.61	52.83	60.19

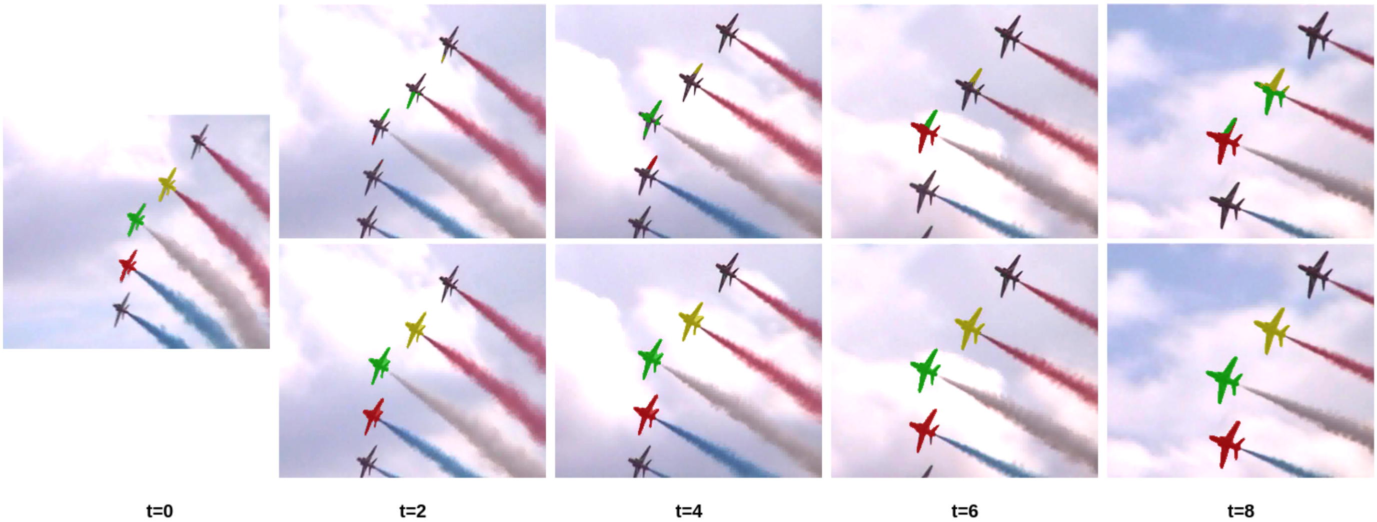

Figure 5. Qualitative comparison between the results obtained from S2S approach (first row) and the results from our method (second row). The first mask ($t = 0$) is provided at test time and the target objects are segmented independently throughout the whole sequence. Every second frame is shown here and the brightness of the images is adjusted for better visibility. As it can be seen, our approach successfully tracks the target airplanes throughout the sequence while the S2S method loses and mixes the object masks early in the sequence.

E. Data Pre- and Post-Processing

In line with the previous work in multiple object video segmentation, we follow a training pipeline, in which every object is tracked independently and at the end the binary masks from different objects are merged into a single mask. For pixels with overlapping predictions, the label from the object with the highest probability is taken into account. For data loading during the training phase, each batch consists of a single object from a different sequence. We noticed that processing multiple objects of the same sequence degrades the performance. The images and the masks are resized to 256×448 as suggested in [33]. For data augmentation we use random horizontal flipping and affine transformations. For the results provided in Section V, we have not used any refinement setp (e.g. CRF [13]) or inference-time augmentation. Moreover, we note that pre-training on image segmentation datasets can greatly improve the results due to the variety of present object categories in these datasets. However, in this work we have solely relied on pre-trained weights from ImageNet [14].

V. Experiments and Results

In this section a comparison with the state-of-the-art methods is provided in Table I. Additionally, we perform an ablation study in Table II to examine the impact of skip-memory and multi-task loss in our approach.

We evaluate our method on YoutubeVOS dataset [33] which is currently the largest dataset for video object segmentation. We use the standard evaluation metrics [21], reporting *Region Similarity* and *Contour Accuracy* (J&F). J corresponds to the average intersection over union between the predicted segmentation masks and the ground-truth, and F is defined as $F = 2\frac{precision*recall}{precision+recall}$, regarding the boundary pixels after applying sufficient dilation to the object edges. For an overall comparability, we use the *overall* metric of the dataset [33] that refers to the average of J&F scores.

A. Comparison to state-of-the-art approaches

In Table I, we provide a comparison to state of art methods with and without online training. As mentioned in Section II, online training is the process of further training at test time through applying a lot of data augmentation on the first mask to generate more data. This phase greatly improves the performance, at the expense of slowing down the inference phase. As it can be seen in Table I, the scores are measured for two categories of seen and unseen objects. This is a difference between other datasets and YoutubeVOS [33] which consists of new object categories in the validation set. Specifically, the validation set in YoutubeVOS dataset includes 474 videos with 65 seen and 26 unseen categories. The score for unseen categories serves as a measure of generalization of different models. As expected, the unseen object categories achieve a higher score when using online training (since the object is already seen by the network during the online training). However, despite not using online training (and therefore also having lower computational demands during test time), S2S and S2S++ achieve higher overall performance. It is worth mentioning, that both F_{seen} and F_{unseen} scores improve by more than 4 percentage points in our approach.

Figure 5 illustrates a qualitative comparison between our results and the ones from the S2S method. We provide additional examples in Figure 6.

B. Ablation Study

Since the S2S method is the base of our work and the source code is not available, we provide a comparison between our implementation of S2S and S2S++ in Table II, when adding each component. As it can be seen from the results, the best performance is achieved when using two skip-memory modules and multi-task loss. We then take this model and experiment with different hyper-parameters for multi-task loss, as shown in Table III. The results show that a higher number of border classes that is closer to regression yields to a higher overall score.

It is worth mentioning that the distance loss (L_{dist}) has less impact for small objects, especially if the diameter of the object is below the border size (in this case no extra distance classes will be added). Hence, we suspect the improvement in segmenting small objects (shown in Figure 5 and Figure 6) is mainly due to the use of skip-memory connections.

VI. Conclusion

In this work we observed that the S2S method often fails when segmenting small objects. We build on top of this approach and propose using skip-memory connections for utilizing multi-scale spatio-temporal information of the video data. Moreover, we incorporate a distance-based multi-task loss to improve the predicted object masks for video object segmentation. In our experiments, we demonstrate that this approach outperforms state of the art methods on the *YoutubeVOS* dataset [33]. Our extensions to the S2S model require minimal changes to the architecture and greatly improves the contour accuracy score (F) and the overall metric.

One of the limitations of the current model is a performance drop for longer sequences, especially in the presence of multiple objects in the scene. In future, we would like to study this aspect and investigate the effectiveness of using attention as a potential remedy. Furthermore, we would like to study the multi task loss in more. One interesting direction is to learn separate task weights for the segmentation and distance prediction task as in [2] rather than using fixed task weights as in our work. In this context, we would also like to examine the usage of a regression task rather than classification task for predicting the distance to the object border.

Acknowledgments

This work was supported by the TU Kaiserslautern CS PhD scholarship program, the BMBF project DeFuseNN (Grant 01IW17002) and the NVIDIA AI Lab (NVAIL) program. Further, we thank all members of the Deep Learning Competence Center at the DFKI for their feedback and support.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[2] Benjamin Bischke, Patrick Helber, Joachim Folz, Damian Borth, and Andreas Dengel. Multi-task learning for segmentation of building footprints with deep neural networks. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1480–1484. IEEE, 2019.

[3] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *European conference on computer vision*, pages 282–295. Springer, 2010.

[4] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017.

[5] Jason Chang, Donglai Wei, and John W Fisher. A video representation using temporal superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2051–2058, 2013.

[6] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *BMVC*, page 8, 2014.

[7] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[8] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. Efficient hierarchical graph-based video segmentation. In *2010 ieee computer society conference on computer vision and pattern recognition*, pages 2141–2148. IEEE, 2010.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[10] Suyog Dutt Jain and Kristen Grauman. Supervoxel-consistent foreground propagation in video. In *European conference on computer vision*, pages 656–671. Springer, 2014.

[11] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for video object segmentation. *International Journal of Computer Vision*, 127(9):1175–1197, 2019.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

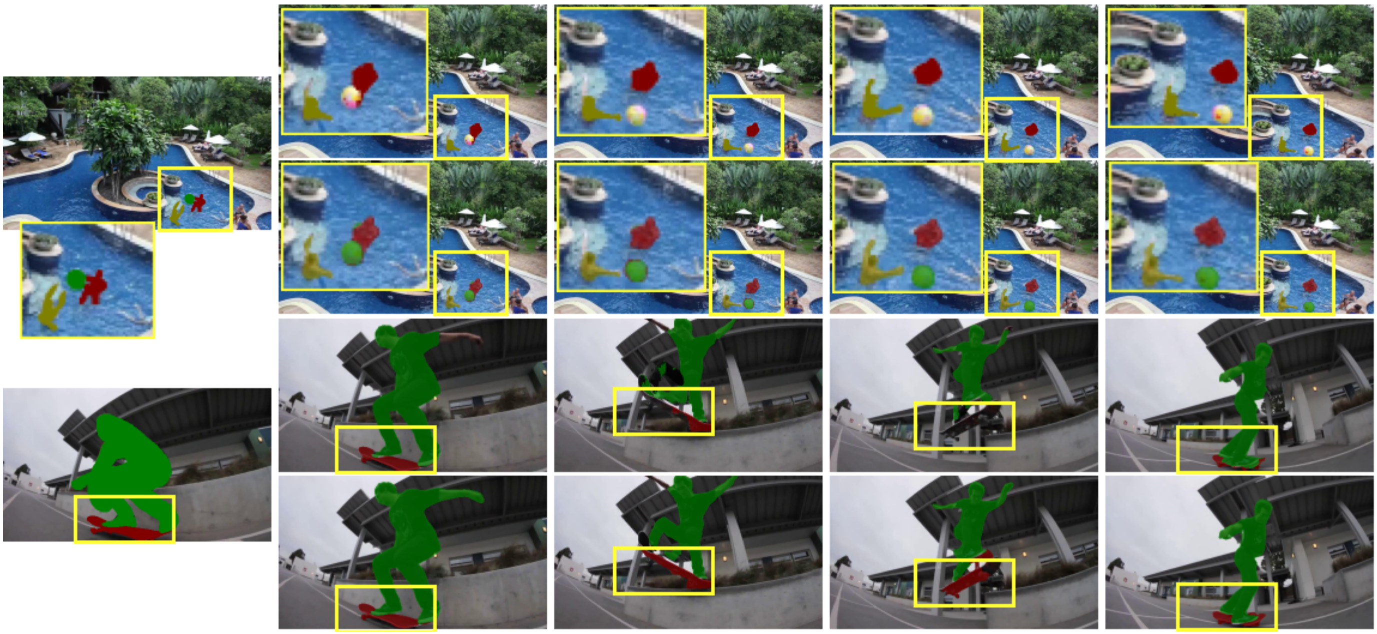

Figure 6. Additional examples for qualitative comparison between the S2S method in the first row, and ours in the second row. In the top part, the yellow frames were zoomed in for better visibility. We can observe a better capacity for tracking small objects.

[15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[16] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *Asian Conference on Computer Vision*, pages 565–580. Springer, 2018.

[17] Kevis-Kokitsi Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.

[18] Nicolas Märki, Federico Perazzi, Oliver Wang, and Alexander Sorkine-Hornung. Bilateral space video segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 743–751, 2016.

[19] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1777–1784, 2013.

[20] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2663–2672, 2017.

[21] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016.

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

[23] Fitsum A Reda, Guilin Liu, Kevin J Shih, Robert Kirby, Jon Barker, David Tarjan, Andrew Tao, and Bryan Catanzaro. Sdc-net: Video prediction using spatially-displaced convolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 718–733, 2018.

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[25] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[26] Amaia Salvador, Miriam Bellver, Victor Campos, Manel Baradad, Ferran Marques, Jordi Torres, and Xavier Giro-i Nieto. Recurrent neural networks for semantic instance segmentation. *arXiv preprint arXiv:1712.00617*, 2017.

[27] Naveen Shankar Nagaraja, Frank R Schmidt, and Thomas Brox. Video segmentation with just a few strokes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3235–3243, 2015.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[29] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4481–4490, 2017.

[30] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5277–5286, 2019.

[31] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. *arXiv preprint arXiv:1706.09364*, 2017.

[32] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[33] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

[34] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6499–6507, 2018.

[35] Rui Yao, Guosheng Lin, Shixiong Xia, Jiaqi Zhao, and Yong Zhou. Video object segmentation and tracking: A survey. *arXiv preprint arXiv:1904.09172*, 2019.