Hybrid-S2S: Video Object Segmentation with Recurrent Networks and Correspondence Matching

Fatemeh Azimi[1,2], Stanislav Frolov[1,2], Federico Raue[2], Jörn Hees[2] and Andreas Dengel[1,2]
[1] *TU Kaiserslautern, Germany*
[2] *DFKI GmbH, Germany*

Keywords: Video Object Segmentation, Recurrent Neural Networks, Correspondence Matching.

Abstract: One-shot Video Object Segmentation (VOS) is the task of pixel-wise tracking an object of interest within a video sequence, where the segmentation mask of the first frame is given at inference time. In recent years, Recurrent Neural Networks (RNNs) have been widely used for VOS tasks, but they often suffer from limitations such as drift and error propagation. In this work, we study an RNN-based architecture and address some of these issues by proposing a hybrid sequence-to-sequence architecture named HS2S, utilizing a dual mask propagation strategy that allows incorporating the information obtained from correspondence matching. Our experiments show that augmenting the RNN with correspondence matching is a highly effective solution to reduce the drift problem. The additional information helps the model to predict more accurate masks and makes it robust against error propagation. We evaluate our HS2S model on the DAVIS2017 dataset as well as Youtube-VOS. On the latter, we achieve an improvement of 11.2pp in the overall segmentation accuracy over RNN-based state-of-the-art methods in VOS. We analyze our model's behavior in challenging cases such as occlusion and long sequences and show that our hybrid architecture significantly enhances the segmentation quality in these difficult scenarios.

1 INTRODUCTION

One-shot Video Object Segmentation (VOS) aims to segment an object of interest in a video sequence, where the object mask in the first frame is provided. The objective of this task is to track a target object in a pixel-wise manner. It has various applications such as robotics, autonomous driving, and video editing to name a few. VOS is a challenging task, and generating quality segmentation masks requires addressing inevitable real-world difficulties such as unconstrained camera motion, occlusion, fast motion, and motion blur as well as handling objects with different sizes.

VOS has been extensively studied in the Computer Vision community with several works based on classical techniques such as energy minimization and utilizing superpixels (Chang et al., 2013; Märki et al., 2016; Grundmann et al., 2010). However, learning-based methods (Perazzi et al., 2017; Maninis et al., 2018) have proved to be more successful by significantly surpassing the traditional approaches.

Amongst the wide variety of the suggested learning-based methods, some works approach the problem by processing the frames independently and learning an object model (Perazzi et al., 2017; Maninis et al., 2018), while others utilize temporal information (Xu et al., 2018; Ventura et al., 2019). Tokmakov *et al.* (Tokmakov et al., 2017) propose utilizing optical flow to propagate the object mask throughout the sequence and make use of the motion cues as well as the spatial information. However, flow-based models need an additional component for flow estimation (Ilg et al., 2017), which is usually trained separately, and the performance of the whole system is dependent on the accuracy of this module. With the same motivation of using temporal data, (Xu et al., 2018; Ventura et al., 2019; Azimi et al., 2020) utilize Recurrent Neural Networks (RNNs) to track the target object in a temporally consistent way. These models are trained end-to-end and rely on learning the spatio-temporal features to track the object and to propagate the object mask across time. A disadvantage of this category is the performance drop in longer sequences caused by drift and error propagation in the RNN.

In this work, we study S2S (Xu et al., 2018), a common RNN-based model for VOS due to the effectiveness of RNNs in utilizing the spatio-temporal

Azimi, F., Frolov, S., Raue, F., Hees, J. and Dengel, A.
Hybrid-S2S: Video Object Segmentation with Recurrent Networks and Correspondence Matching.
DOI: 10.5220/0010339401820192
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 4: VISAPP*, pages 182-192
ISBN: 978-989-758-488-6

features and providing a motion model of the target object, resulting in good segmentation accuracy. Inspired by (Faktor and Irani, 2014; Wug Oh et al., 2018; Yang et al., 2019), we propose a dual propagation strategy by augmenting the spatio-temporal features obtained from the RNN with correspondence matching to reduce the impact of drift. Utilizing the features obtained from similarity matching provides a robust measurement for segmentation, improves the segmentation quality, and reduces the error propagation. This aspect is especially beneficial for the model in long sequences where the RNN performance declines. Additionally, we integrate the first frame features into the model throughout the whole sequence as a reliable source of information (Ebert et al., 2017; Wug Oh et al., 2018; Oh et al., 2019; Yang et al., 2019). By employing these reference features, the model can better handle challenging scenarios such as occlusion (Ebert et al., 2017), since, by definition, the object is present in the first frame. Figure 1 shows an illustration of how correspondence matching together with utilizing the first frame can be helpful in better handling the occlusion. We hypothesize that the RNN also plays a complementary role in correspondence matching. Imagine a scenario where multiple instances of similar objects are present in the scene; in this case, the spatio-temporal model learned by the RNN can act as a location prior and aid the model to distinguish between the target object and the other similar instances.

We evaluate our hybrid sequence-to-sequence (HS2S) method on the Youtube-VOS (Xu et al., 2018) and DAVIS2017 (Pont-Tuset et al., 2017) datasets and demonstrate that our model significantly improves the independent RNN-based models' segmentation quality (Xu et al., 2018; Ventura et al., 2019).

2 RELATED WORK

A large body of research in Computer Vision literature has studied VOS during the last decade. The classical methods for solving VOS were mainly based on energy minimization (Brox and Malik, 2010; Faktor and Irani, 2014; Papazoglou and Ferrari, 2013; Shankar Nagaraja et al., 2015). Brox *et al.* (Brox and Malik, 2010) propose a model based on motion clustering and segment the moving object via the analysis of the point trajectories throughout the video. They also use motion cues to distinguish foreground from background. Faktor *et al.* (Faktor and Irani, 2014) present a method based on consensus voting. They extract the superpixels in each frame, and by computing the similarity of the superpixel descriptors, then use the nearest neighbor method to cluster the most similar superpixels together in a segmentation mask. (Jain and Grauman, 2014) addresses the problem of fast motion and appearance change in the video by extending the idea of using superpixels to using supervoxels (adding the time dimension) and taking into account the long-range temporal connection during the object movement.

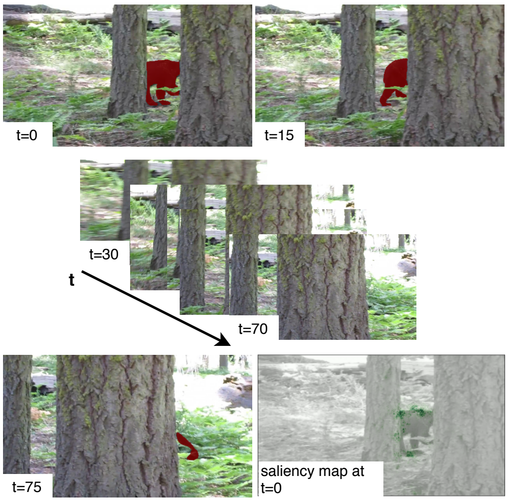

Figure 1: This figure indicates how utilizing the first frame as the reference can help the model recover from occlusion. Here, the object of interest is a bear overlaid with the red mask, which is absent from the middle row frames (from $t = 30$ to $t = 70$). We observe that the model can detect the animal after it appears again, and by looking at the saliency map of the first frame, we note that the model has correctly captured the correspondence between the bear in the first frame and the frame right after the occlusion.

Since the advent of Deep Learning (Krizhevsky et al., 2012), the Computer Vision community has witnessed a significant progress in the accuracy of VOS methods (Maninis et al., 2018; Perazzi et al., 2017; Tokmakov et al., 2017). The success of learning based methods can largely be accounted to progress made in learning algorithms (Krizhevsky et al., 2012; He et al., 2016) and the availability of large-scale VOS datasets such as Youtube-VOS (Xu et al., 2018).

In one-shot VOS, there exist two training schemes, namely offline and online training. Offline training is the standard training phase in learning-based techniques. As the segmentation mask of the first object appearance is available at test time, online training refers to further fine-tuning the model on this mask with extensive data augmentation. This additional step considerably improves the segmentation quality at the expense of slower inference.

Considering offline training, one can divide the proposed solutions into multiple categories. Some methods focus on learning the object masks using only the frame-wise data (Perazzi et al., 2017; Maninis et al., 2018). In (Maninis et al., 2018), authors extended a VGG-based architecture designed for retinal image understanding (Maninis et al., 2016) for VOS. They start with the pre-trained weights on ImageNet (Deng et al., 2009), and then further train the *parent network* on a specialized VOS dataset (Perazzi et al., 2016a). This model relies on online training and boundary snapping for achieving good performance. (Voigtlaender and Leibe, 2017) further improves this method by employing online adaption to handle drastic changes in the object's appearance. Perazzi *et al.* (Perazzi et al., 2017) provide a solution based on guided instance segmentation. They utilize a DeepLab architecture (Chen et al., 2017) and modify the network to accept the previous segmentation mask as an additional input. Therefore, a rough guidance signal is provided to the model to mark the approximate location where the object of interest lies. Yang *et al.* (Yang et al., 2018) take a meta-learning approach and train an additional modulator network that adjusts the middle layers of a generic segmentation network to capture the appearance of the target object.

In (Wug Oh et al., 2018) a Siamese architecture is used to segment the object based on its similarity to the mask template in the first frame. Similarly, (Yang et al., 2019) proposes a zero-shot VOS model, where the object mask at every time step is detected based on the similarity of the current frame to the anchor frames (first frame and the frame at $t-1$). Following this idea, (Johnander et al., 2019) suggests a generative approach for segmenting the target object, introducing an appearance module to learn the probabilistic model of the background and the foreground object. In (Zhang et al., 2020), the authors develop a model that propagates the segmentation mask based on an affinity in the embedding space. They propose to model the local dependencies via using motion and spatial priors and the global dependencies based on the visual appearance learned by a convolutional network. Although these methods obtain good performance on the standard benchmarks (Perazzi et al., 2016a; Pont-Tuset et al., 2017), they do not utilize temporal information and motion cues.

Another line of work relies on region proposal techniques such as (He et al., 2017). For example, (Luiten et al., 2018) takes a multi-step approach, in which they first generate the region proposals and then refine and merge promising regions to produce the final mask. Furthermore, they use optical flow to maintain the temporal consistency. In (Li et al., 2017), an additional re-identification method based on template-matching is used. This way, the model can recapture objects lost at some point in the sequence. These methods are quite complex in architecture design and relatively slow at inference time.

A different group of methods focus on utilizing a memory module to process motion and compute spatio-temporal features. In order to obtain temporally consistent segmentation masks, (Xu et al., 2018; Azimi et al., 2020; Tokmakov et al., 2017; Ventura et al., 2019) employ a ConvLSTM (Xingjian et al., 2015) (or ConvGRU) memory module while (Oh et al., 2019) resorts to using an external memory to process the space-time information.

In this work, we build on top of the S2S (Xu et al., 2018) architecture, which is an RNN-based method, on account of exploiting the spatio-temporal features, good performance, and the simple architecture. We study some of this model's shortcomings stemming from the finite memory and error propagation in RNNs. To address these limitations, we propose a hybrid design that combines the spatio-temporal features from the RNN with similarity matching information. Unlike (Oh et al., 2019), our model does not require any form of external memory. This is advantageous since using external memory results in additional constraints in the inference phase (e.g. memory overflow for long video sequences).

3 METHOD

In this section, we explain our hybrid architecture for VOS. We build on top of the S2S model (Xu et al., 2018), which is an RNN-based architecture and employ a dual mask propagation strategy that utilizes the spatio-temporal features from the RNN as well as correspondence matching to propagate the mask from time step $t-1$ to t. Moreover, we integrate the features from the first frame as a reference throughout the sequence.

The S2S model is composed of an encoder-decoder architecture with a memory module in the bottleneck to memorize the target object and obtain temporal consistency in the predicted segmentation masks. The overall design of this method is illustrated in Figure 2. In this model, the object masks are computed as in (Xu et al., 2018):

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

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

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

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

where x and y refer to the RGB input image and the binary mask of the target object in the first frame.

One of the main limitations of RNN-based models, such as S2S, is the fixed-sized memory, which can be insufficient to capture the whole sequence and long-term dependencies (Bahdanau et al., 2014). Therefore, as the sequence length grows, access to information from earlier time steps decreases. This issue, together with the vanishing gradient problem, adversely impacts the segmentation quality in longer sequences. This problem is especially critical in sequences with occlusion, where the object of interest can be absent for an extended period.

Another obstacle with this category of approaches is drift and error propagation. Due to the recurrent connection, the model output is fed back to the network; as a result, the prediction error propagates to the future, and erroneous model predictions affect the performance for future time steps. This issue is another contributing factor to the performance drop in later frames.

Hybrid Mask Propagation. Based on the challenges in the RNN-based models, we propose a hybrid architecture, combining the RNN output with information derived from correspondence matching. In our model, the segmentation mask is predicted using the location prior obtained from the RNN, as well as similarity matching between the video frames at $t-1$ and t. Our intuition is that the merits of using the spatio-temporal model from RNN-based models and the matching-based methods are complementary. In situations where multiple similar objects are present in the scene, the matching-based approaches struggle to distinguish between the different instances. Hence, the location prior provided by the spatio-temporal features from the RNN can resolve this ambiguity. Moreover, the information obtained from similarity matching provides a reliable measurement for propagating the segmentation mask to the next time step (as investigated in (Yang et al., 2019) for zero-shot VOS). Using this additional data helps the model reduce the prediction error, improving the drift problem, and obtaining better segmentation quality for longer sequences.

To encode the frame at $t-1$, we redefine the initializer network's task in S2S to a reference encoder (as shown in Figure 2), initializing the hidden states of the RNN module with *zeros*. In our experiments, we observed that the initializer network does not play a crucial role, and it is possible to replace it with zero-initialization with little change in the performance.

To perform the similarity matching between the RNN hidden state (h_t) and the reference encoder's output features, one can use different techniques such as using the cosine distance between the feature vectors. Here, we follow the design in (Wug Oh et al., 2018) and use a Global Convolution (Peng et al., 2017) to accomplish the task (merge layer in Figure 2). Global Convolution (GC) approximates a large kernel convolution layer efficiently with less number of parameters. The large kernel size is essential to model both the local connections (as required for localization) and the dense global connections required for accurate classification (foreground, background). This way, the model directly accesses the features from time steps 0 and $t-1$. We note that this operation can also be interpreted as self-attention; as, the features at the current time step, which share a higher similarity to the object features from the reference frames, get a higher weight via the convolution operation in the merge layer.

As shown in Figure 2, we do not use weight sharing between the Reference Encoder and the Encoder, as we observed a considerable performance drop in doing so. We believe the underlying reason is that the functions approximated by these two modules are different; the inputs to the Reference Encoder are aligned in time while the inputs to the Encoder are not. We highlight that compared to S2S (Xu et al., 2018), the only added element is the light-weight Merge Layer (Figure 2). The rest of the components remain unchanged, by modifying the task of the Initializer Network to Reference Encoder.

Attention to the First Frame. As suggested in (Ebert et al., 2017) for the Video Prediction task, the first frame of the sequence is of significant importance as it contains the reference information which can be utilized for recovering from occlusion. We note that by definition, the target object is present in the first frame. By computing the correspondences between the object appearance after occlusion and in the first frame, the model is able to re-detect the target. Additionally, (Yang et al., 2019; Wug Oh et al., 2018) demonstrate the effectiveness of using the first frame as an anchor or reference frame. In (Wug Oh et al., 2018), the authors propose a Siamese architecture that learns to segment the object of interest by finding the feature correspondences between the target object in the first frame and the current frame. Although this model's performance suffers in scenarios with drastic appearance change, it reveals the importance of rigorously using the data in the first frame. We use the same reference encoder and merge layer for integrating the first frame features. We hypothesize that this modification can be considered as an attention mechanism (Bahdanau et al., 2014), where the attention span is limited to the first frame. Using attention is a standard solution to address this finite memory in the

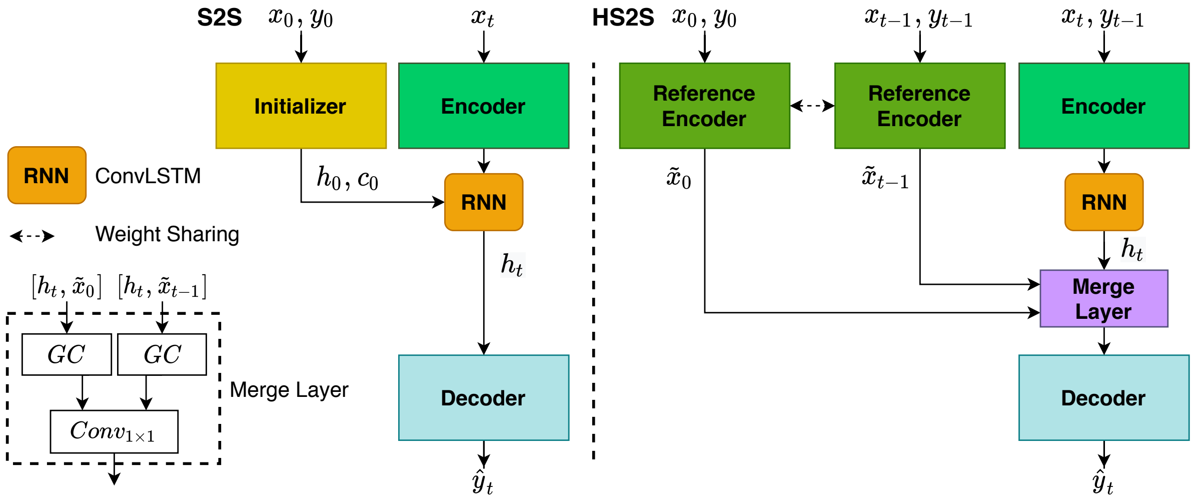

Figure 2: In this figure, we depict the overall architecture of S2S (Xu et al., 2018) (Equations (1) to (4)) and our HS2S method (equations (5) to (10)). In HS2S, we initialize the RNN hidden states (h_0 and c_0) with *zeros*, instead of using the initializer network. We keep track of the target object by feeding the previous segmentation mask (y_{t-1}) to the encoder as an additional input channel, similar to (Perazzi et al., 2017). Furthermore, we use a separate reference encoder to process the input to the matching branch. We highlight that the functions approximated by these two encoders differ, as the inputs to the *Reference Encoder* are aligned in time, but this is not the case for the *Encoder* network. Finally, the hidden state of the RNN (h_t) is combined with the encoded features from the matching branch via a merge layer and passed to the decoder to predict the segmentation mask. The skip connections between the encoder and the decoder networks are not shown for simplicity.

RNNs, by providing additional context to the memory module. The context vector is usually generated from a weighted combination of the embeddings from all the time steps. However, in high-dimensional data such as video, it would be computationally demanding to store the features and compute all the frames' attention weights.

The resulting architecture is shown in Figure 2 and can be formulated as:

$$h_0, c_0 = \mathbf{0} \tag{5}$$

$$\tilde{x}_0 = \text{Reference\_Encoder}(x_0, y_0) \tag{6}$$

$$\tilde{x}_{t-1} = \text{Reference\_Encoder}(x_{t-1}, y_{t-1}) \tag{7}$$

$$\tilde{x}_t = \text{Encoder}(x_t, y_{t-1}) \tag{8}$$

$$h_t, c_t = \text{RNN}(\tilde{x}_t, h_{t-1}, c_{t-1}) \tag{9}$$

$$\hat{y} = \text{Decoder}(\tilde{x}_0, \tilde{x}_{t-1}, h) \tag{10}$$

where x and y are the RGB image and the binary segmentation mask, and $\mathbf{0} \in R^d$ with d as the feature dimension. Here the merge layer is considered as part of the decoder.

Training Objective. For the loss function, we utilize a linear combination of the balanced Binary Cross-Entropy (BCE) loss and an auxiliary loss (Azimi et al., 2020):

$$L_{\text{total}} = \lambda\, L_{\text{seg}} + (1-\lambda)\, L_{\text{aux}} \tag{11}$$

The auxiliary task employed here is border classification. For this task, a border class is assigned to each pixel based on its location with respect to the object boundary, where the boundary target classes are assigned based on a distance transform (Hayder et al., 2017). This term provides fine-grained location information for each pixel resulting in improved boundary detection F-score. For more details, please refer to (Azimi et al., 2020).

The balanced BCE loss is computed as in (Caelles et al., 2017) :

$$\begin{aligned} L_{\text{seg}}(\mathbf{W}) = \sum_{t=1}^{T}(-\beta \sum_{j\in Y_+} \log P(y_j = 1|X;\mathbf{W}) \\ -(1-\beta) \sum_{j\in Y_-} \log P(y_j = 0|X;\mathbf{W})) \end{aligned} \tag{12}$$

with X as input, $\mathbf{W}$ as the model parameters, Y_+ and Y_- standing for the foreground and background groundtruth labels, $\beta = |Y_-|/|Y|$, and $|Y| = |Y_-| + |Y_+|$. This loss addresses the data imbalance between the foreground and the background classes by the weighting factors β.

4 IMPLEMENTATION DETAILS

In this section, we explain the implementation details of our hybrid model. The code and the pre-trained models are publicly available [1].

[1] https://github.com/fatemehazimi990/HS2S

4.1 Encoder Networks

In the S2S model, a VGG network (Simonyan and Zisserman, 2014) is used as the backbone for the initializer and encoder networks. In this work, we utilize a ResNet50 (He et al., 2016) architecture, pretrained on ImageNet (Deng et al., 2009). We remove the last average pooling and the fully connected layers, which are specific for image classification. Furthermore, we add an extra 1 × 1 convolution layer to reduce the number of output channels from 2048 to 1024. The number of input channels is altered to 4, as we feed the RGB image and the binary segmentation mask to the encoder. We utilize skip connections (Ronneberger et al., 2015) between the encoder and the decoder at every spatial resolution of the feature map (5 levels in total) to capture the fine details lost in the pooling operations and reducing the spatial size of the feature map. Moreover, we use an additional RNN module in the first skip connection, as suggested in (Azimi et al., 2020). The impact of changing the backbone network in the S2S model from VGG to ResNet on the segmentation accuracy is studied in Table 5.

4.2 RNN and Merge Layer

For the RNN component, we use a ConvLSTM layer (Xingjian et al., 2015), with a kernel-size of 3×3 and 1024 filters. As suggested in (Xu et al., 2018), *Sigmoid* and *ReLU* activations are used for the gate and state outputs, respectively.

The merging layer's role is to perform correspondence matching between the RNN hidden state (the spatio-temporal features) and the outputs from the reference encoder. There are different ways that can be used for this layer based on similarity matching and cosine distance. Similar to (Wug Oh et al., 2018), we utilize Global Convolution (GC) layers (Peng et al., 2017) for this function. Two GC layers with an effective kernel size of 7×7 are employed to combine the RNN hidden state with the reference features and the features from the previous time step ($\tilde{x}_0$ and $\tilde{x}_{t-1}$ as in Equations (6) and (7)). The output of these two layers are then merged using a 1×1 convolution and then fed into the decoder network.

4.3 Decoder

The decoder network consists of five up-sampling layers followed by 5×5 convolution layers with 512, 256, 128, 64, and 64 number of filters, respectively. In the last layer, a $Conv_{1 \times 1}$ maps the 64 channels to 1 and a *Sigmoid* activation is used to generate the binary segmentation scores (for the foreground and background classes). The features from the skip connections are merged into the decoder using a 1×1 convolution layers. *ReLU* activation is used after each convolution layer, except for the last layer, where we use *Sigmoid* activation to generate the segmentation output.

4.4 Training Details

For data augmentation, we apply random horizontal flipping as well as affine transformations. The λ in Equation 11 is set to 0.8. We use Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 10^{-4}. We gradually lower the learning rate in the final phase of training when the loss is stable. During the training, we use video snippets with 5 to 10 frames and a batch size of 16.

Additionally, we apply a curriculum learning method as suggested for sequence prediction tasks (Bengio et al., 2015). To this end, we use the ground-truth for the segmentation mask input in the earlier stages of training where the model output is not yet satisfactory. This phase is known as *teacher forcing*. Next, with a pre-defined probabilistic scheme (Bengio et al., 2015), we randomly choose between using the ground-truth or the model-generated segmentation mask, on a per-frame basis. This process helps to close the gap between the training and inference data distributions (during the inference, only the model-generated masks are used).

5 EXPERIMENTS

This section provides the experimental results for our hybrid model and a comparison with other state-of-the-art methods. Additionally, we analyze our hybrid model's behavior on the two challenging scenarios occlusion and long sequences.

5.1 Evaluation on Youtube-VOS and DAVIS2017

We evaluate our model on the Youtube-VOS (Xu et al., 2018) dataset (the largest for Video Object Segmentation), as well as the DAVIS2017 dataset (Pont-Tuset et al., 2017).

We report the standard metrics of the task, namely *Region Similarity* and *Boundary Accuracy* (*F&J*) (Perazzi et al., 2016b). The *F* score measures the quality of the estimated segmentation boundaries and the Jaccard index *J* measures the intersection over union area between the model output and the ground-truth segmentation mask.

Table 1 shows a comparison of our model with other state-of-the-art methods. The upper and lower sections include the methods with and without online training. During the online training, the model is further fine-tuned on the first frame (where the segmentation mask is available) at test time; Although this stage significantly improves the segmentation accuracy, it results in slow inference which is not practical for real-time applications. Despite this, we see that our model without online training still outperforms the S2S model with online training. The performance improvement compared to RGMP (Wug Oh et al., 2018) (matching-based) and S2S (Xu et al., 2018) (RNN-based) models strongly indicates that both propagation and matching information are required for better segmentation quality. Moreover, our method achieves similar performance to STM (Oh et al., 2019) when training on the same amount of data (not using synthetic data generated from image segmentation datasets) without relying on an external memory module; Therefore, our model is less memory-constrained during the inference stage compared to methods using external memory that are prone to memory overflow for longer sequences (in (Oh et al., 2019), authors save every 5th frame to the memory to avoid the GPU memory overflow during the test phase).

Figure 3 illustrates some visual examples from our model. As we see, our model can properly track the target object in the presence of similar object instances as well as occlusion. More visual samples are provided in the supplementary material.

To assess the generalization of our model after training on Youtube-VOS, we freeze the weights and evaluate the model on DAVIS2017 dataset (Pont-Tuset et al., 2017). The results can be seen in Table 2. We observe that our hybrid model outperforms the independent RNN-based and matching-based methods, even without fine-tuning on this dataset.

Table 1: A comparison with the state-of-the-art methods on the Youtube-VOS dataset (Xu et al., 2018). The upper part of the table shows models with online training, the lower part without. All scores are in percent. RVOS, S2S, and S2S++ are the RNN-based architectures. As shown in this table, our hybrid model outperforms the S2S(no-OL) baseline model with an average improvement of 11.2 pp. STM- refers to results in (Oh et al., 2019), with the same amount of training data for a fair comparison. We can see that our method can achieve similar results to STM-, without requiring an external memory module.

Method	*J*	*F*	*F&J*
OSVOS (Maninis et al., 2018)	57.0	60.6	58.8
MaskTrack (Perazzi et al., 2017)	52.5	53.7	50.6
S2S(OL) (Xu et al., 2018)	**63.25**	**65.6**	**64.4**
OSMN (Yang et al., 2018)	50.3	52.1	51.2
RGMP (Wug Oh et al., 2018)	52.4	55.3	53.8
RVOS (Ventura et al., 2019)	54.6	59.1	56.8
A-GAME (Johnander et al., 2019)	64.3	67.9	66.1
S2S(no-OL) (Xu et al., 2018)	57.5	57.9	57.7
S2S++ (Azimi et al., 2020)	58.8	63.2	61.0
STM- (Oh et al., 2019)	-	-	68.2
TVOS (Zhang et al., 2020)	65.4	70.5	67.2
HS2S (ours)	**66.1**	**71.7**	**68.9**

Table 2: A comparison between the independent RNN-based (RVOS) and matching-based (RGMP) models and our hybrid method on the DAVIS2017 dataset (Pont-Tuset et al., 2017) (test-val). HS2S- shows the results of our model trained on Youtube-VOS without fine-tuning on DAVIS2017. The results of the S2S model on DAVIS2017 were not available.

Method	*J*	*F*	*F&J*
S2S (Xu et al., 2018)	-	-	-
RVOS (Ventura et al., 2019)	52.7	58.1	55.4
RGMP (Wug Oh et al., 2018)	58.1	61.5 8	59.8
HS2S- (ours)	**58.9**	**63.4**	**61.1**

5.2 Analysis of Sequence Length and Occlusion

To quantitatively assess our model's effectiveness, we evaluate it in challenging scenarios such as occlusion and longer sequences. As the validation set of the Youtube-VOS dataset is not released, we use the 80:20-splits of the training set from (Ventura et al., 2019) for training and evaluation. For the S2S model results, we further used our re-implementation as the code for their work is not publicly available. Furthermore, we use the ResNet50 architecture as backbone for both models for a fair comparison (to our disadvantage, as it improves the overall evaluation score of 57.3% for S2S (as reported in (Xu et al., 2018)) to 60% for our re-implementation S2S*).

Figure 4 shows the sequence length distribution of the Youtube-VOS training set (one sequence per object in each video). As can be seen, the length varies between 1 to about 35 frames in a very non-uniform fashion. To study the impact of the video length on the segmentation scores, we pick the sequences longer than 20 frames and measure the scores for frames with $t < 10$ (considered as early frames) and frames with $t > 20$ (considered as late frames), separately. As presented in Table 3, we observe that the hybrid model improves the late frame accuracy significantly and reduces the performance gap between the early and late frames. This observation confirms the effectiveness of the hybrid path for utilizing the information from spatio-temporal features as well as the correspondence matching.

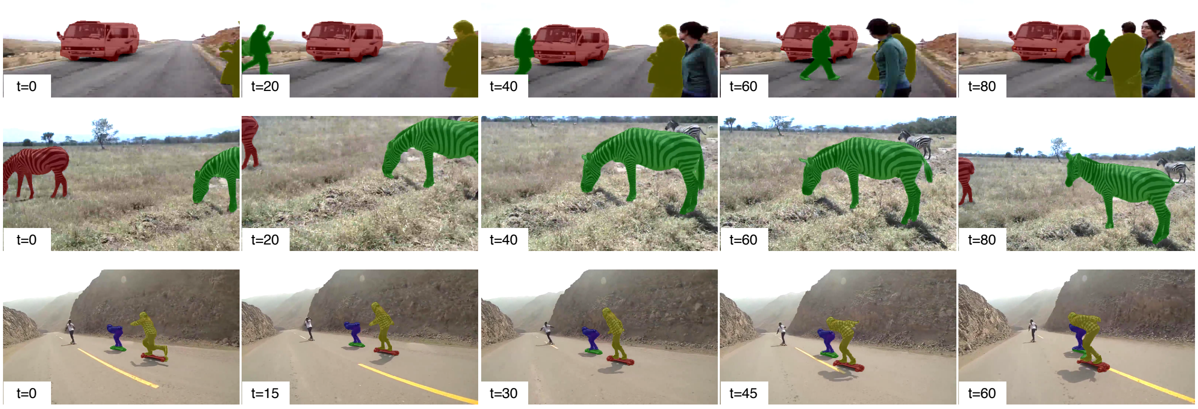

Figure 3: Visual samples of our model on Youtube-VOS validation set. As can be observed, our method can successfully segment sequences with similar object instances, even in the presence of occlusion.

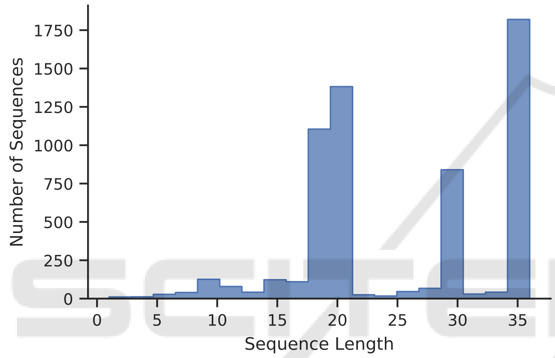

Figure 4: Distribution of the sequence length (per object) in the Youtube-VOS dataset. In Youtube-VOS, the video frame rate is reduced to 30 fps, and the annotations are provided every fifth frame (6 fps). Therefore, a sequence with 36 labeled frames spans 180 time steps in the original frame rate.

Table 3: A study on the impact of sequence length on the segmentation accuracy. For this experiment, we picked the video sequences with more than 20 frames. Then we compute the F and J scores for frames earlier ($t < 10$) and later ($t > 20$) in the sequence. As the results show, there is a performance drop as the time step increases. However, our hybrid model's performance drops a lot less than the baseline's.

Method	$F_{l<10}$	$J_{l<10}$	$F_{l>20}$	$J_{l>20}$
S2S*	74.4	73.7	54.5	54.6
HS2S (ours)	**77.1**	**76.3**	**65.5**	**64.2**

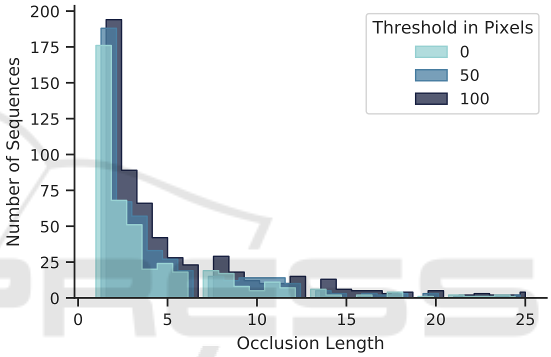

Figure 5: The number of occluded sequences (per object) in Youtube-VOS train set, for different occlusion lengths and with three occlusion thresholds (shifted by 1/3 for better visibility).

Table 4: A study on the impact of occlusion on the segmentation quality. The scores presented in this table are the average of F and J scores in percentages, when considering different thresholds (in pixels) for occlusion. The *avg* score refers to the average result for all the sequences in the 20-split. For the other columns, we only considered the frames after ending the first occlusion period (when the target object re-appears in the scene).

Method	*avg*	*th : 0*	*th : 50*	*th : 100*
S2S*	63.3	33.6	30.8	33.1
HS2S (ours)	**69.0**	**40.2**	**39.3**	**47.7**

The histogram in Figure 5 shows the number of sequences with occlusion in Youtube-VOS training set. Each bin in the histogram shows the occlusion duration, and the y axis indicates the number of sequences that belong to each bin. As can be seen from this plot, the occlusion duration varies between 1 to 25 frames. To study our model's effectiveness in handling occlusion, we report the scores for frames appearing *after* a first occlusion in Table 4. An occlusion is considered a scenario where the object leaves the scene entirely and re-appears again. As the areas below 100 pixels are almost not visible (and could be considered as labeling noise), we also consider occlusions at three different thresholds of 0, 50, and 100 pixels. As we can see in the table, occlusion is a challenging scenario with significantly lower scores than the average sequence scores. However, our proposed ap-

Table 5: An ablation study on the impact of different components in our model. S2S* is our re-implementation of the S2S method with the same backbone as our model, for a fair comparison (this version achieves a better segmentation accuracy). $S2S_0$ refers to our model without the hybrid propagation, only using the first frame as reference. $S2S_{t-1}$ is our model with hybrid propagation and without utilizing the first frame. In $HS2S_{\text{sim}}$, we implemented the merge layer (Figure 2) using cosine similarity instead of Global Convolution.

Method	*J*	*F*	*F&J*
S2S (Xu et al., 2018)	57.5	57.9	57.7
S2S*	59.1	63.7	61.4
$HS2S_0$	64.0	68.95	66.5
$HS2S_{t-1}$	63.6	68.7	66.2
HS2S	**66.1**	**71.7**	**68.9**
$HS2S_{sim}$	64.35	69.35	66.9

proach again succeeds in defending its considerable improvement over the S2S baseline.

6 ABLATION STUDY

In this section, we present an ablation study on the impact of different components of our model. In addition, we provide the results for a variant of our model where we use cosine similarity (Wang et al., 2018) for the merge layer instead of global convolution (referred to as $HS2S_{sim}$).

Table 5 presents the segmentation scores when different components in our model are added one at a time. The results for *S2S** are obtained from our re-implementation of the S2S model with ResNet50 backbone. As it can be seen from the results, utilizing the first frame as the reference ($HS2S_0$) and using the hybrid match-propagate strategy ($HS2S_{t-1}$) both improve the segmentation quality. Moreover, the enhancements add up when they are integrated into a single model (HS2S).

7 CONCLUSION

In this work, we presented a hybrid architecture for the task of one-shot Video Object Segmentation. To this end, we combined the merits of RNN-based approaches and models based on correspondence matching. We showed that the advantages of these two categories are complementary, and can assist each other in challenging scenarios. Our experiments demonstrate that both mechanisms are required for obtaining better segmentation quality.

Furthermore, we provided an analysis of two challenging scenarios: occlusion and longer sequences. We observed that our hybrid model achieves a significant improvement in robustness compared to the baselines that rely on RNNs (Xu et al., 2018) and reference guidance (Wug Oh et al., 2018). However, occlusion remains an open challenge for future investigation, as the performance in this scenario is considerably lower than the average. Moreover, we believe that integrating global information and modeling the interactions between the objects in the scene is a promising direction for future work.

ACKNOWLEDGMENTS

This work was supported by the TU Kaiserslautern CS PhD scholarship program, the BMBF project ExplAINN (01IS19074), 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

Azimi, F., Bischke, B., Palacio, S., Raue, F., Hees, J., and Dengel, A. (2020). Revisiting sequence-to-sequence video object segmentation with multi-task loss and skip-memory. *arXiv preprint arXiv:2004.12170*.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. (2015). Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179.

Brox, T. and Malik, J. (2010). Object segmentation by long term analysis of point trajectories. In *European conference on computer vision*, pages 282–295. Springer.

Caelles, S., Maninis, K.-K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., and Van Gool, L. (2017). One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230.

Chang, J., Wei, D., and Fisher, J. W. (2013). A video representation using temporal superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2051–2058.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on com-*

puter vision and pattern recognition, pages 248–255. Ieee.

Ebert, F., Finn, C., Lee, A. X., and Levine, S. (2017). Self-supervised visual planning with temporal skip connections. *arXiv preprint arXiv:1710.05268*.

Faktor, A. and Irani, M. (2014). Video segmentation by non-local consensus voting. In *BMVC*, page 8.

Grundmann, M., Kwatra, V., Han, M., and Essa, I. (2010). Efficient hierarchical graph-based video segmentation. In *2010 ieee computer society conference on computer vision and pattern recognition*, pages 2141–2148. IEEE.

Hayder, Z., He, X., and Salzmann, M. (2017). Boundary-aware instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5696–5704.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. (2017). Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470.

Jain, S. D. and Grauman, K. (2014). Supervoxel-consistent foreground propagation in video. In *European conference on computer vision*, pages 656–671. Springer.

Johnander, J., Danelljan, M., Brissman, E., Khan, F. S., and Felsberg, M. (2019). A generative appearance model for end-to-end video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8953–8962.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Li, X., Qi, Y., Wang, Z., Chen, K., Liu, Z., Shi, J., Luo, P., Tang, X., and Loy, C. C. (2017). Video object segmentation with re-identification. *arXiv preprint arXiv:1708.00197*.

Luiten, J., Voigtlaender, P., and Leibe, B. (2018). Premvos: Proposal-generation, refinement and merging for video object segmentation. In *Asian Conference on Computer Vision*, pages 565–580. Springer.

Maninis, K.-K., Caelles, S., Chen, Y., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., and Van Gool, L. (2018). Video object segmentation without temporal information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Maninis, K.-K., Pont-Tuset, J., Arbeláez, P., and Van Gool, L. (2016). Deep retinal image understanding. In *International conference on medical image computing and computer-assisted intervention*, pages 140–148. Springer.

Märki, N., Perazzi, F., Wang, O., and Sorkine-Hornung, A. (2016). Bilateral space video segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 743–751.

Oh, S. W., Lee, J.-Y., Xu, N., and Kim, S. J. (2019). Video object segmentation using space-time memory networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9226–9235.

Papazoglou, A. and Ferrari, V. (2013). Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1777–1784.

Peng, C., Zhang, X., Yu, G., Luo, G., and Sun, J. (2017). Large kernel matters–improve semantic segmentation by global convolutional network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361.

Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., and Sorkine-Hornung, A. (2017). Learning video object segmentation from static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2663–2672.

Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., and Sorkine-Hornung, A. (2016a). A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*.

Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., and Sorkine-Hornung, A. (2016b). 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.

Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., and Van Gool, L. (2017). The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Shankar Nagaraja, N., Schmidt, F. R., and Brox, T. (2015). Video segmentation with just a few strokes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3235–3243.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Tokmakov, P., Alahari, K., and Schmid, C. (2017). Learning video object segmentation with visual memory. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4481–4490.

Ventura, C., Bellver, M., Girbau, A., Salvador, A., Marques, F., and Giro-i Nieto, X. (2019). 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.

Voigtlaender, P. and Leibe, B. (2017). Online adaptation of convolutional neural networks for video object segmentation. *arXiv preprint arXiv:1706.09364*.

Wang, X., Girshick, R., Gupta, A., and He, K. (2018). Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803.

Wug Oh, S., Lee, J.-Y., Sunkavalli, K., and Joo Kim, S. (2018). Fast video object segmentation by reference-guided mask propagation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7376–7385.

Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810.

Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., and Huang, T. (2018). Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*.

Yang, L., Wang, Y., Xiong, X., Yang, J., and Katsaggelos, A. K. (2018). Efficient video object segmentation via network modulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6499–6507.

Yang, Z., Wang, Q., Bertinetto, L., Hu, W., Bai, S., and Torr, P. H. (2019). Anchor diffusion for unsupervised video object segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 931–940.

Zhang, Y., Wu, Z., Peng, H., and Lin, S. (2020). A transductive approach for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6949–6958.