CapsuleVOS: Semi-Supervised Video Object Segmentation Using Capsule Routing

Kevin Duarte
kevin_duarte@knights.ucf.edu

Yogesh S Rawat
yogesh@crcv.ucf.edu

Mubarak Shah
shah@crcv.ucf.edu

Center for Research in Computer Vision
University of Central Florida
Orlando, FL 32816

Abstract

In this work we propose a capsule-based approach for semi-supervised video object segmentation. Current video object segmentation methods are frame-based and often require optical flow to capture temporal consistency across frames which can be difficult to compute. To this end, we propose a video based capsule network, CapsuleVOS, which can segment several frames at once conditioned on a reference frame and segmentation mask. This conditioning is performed through a novel routing algorithm for attention-based efficient capsule selection. We address two challenging issues in video object segmentation: 1) segmentation of small objects and 2) occlusion of objects across time. The issue of segmenting small objects is addressed with a zooming module which allows the network to process small spatial regions of the video. Apart from this, the framework utilizes a novel memory module based on recurrent networks which helps in tracking objects when they move out of frame or are occluded. The network is trained end-to-end and we demonstrate its effectiveness on two benchmark video object segmentation datasets; it outperforms current offline approaches on the Youtube-VOS dataset while having a run-time that is almost twice as fast as competing methods. The code is publicly available at https://github.com/KevinDuarte/CapsuleVOS.

1. Introduction

Semi-supervised video object segmentation aims to segment objects in a video, given their segmentation masks for the first frame. This is a challenging problem because of issues like occlusion, changes in object appearance over time, motion blur, fast motions, and scale variations of different objects. Deep learning approaches have achieved impressive results and the recent release of the Youtube-VOS dataset [37] has allowed for the training and evaluation of new methods on a wider variety of videos and objects.

The majority of current approaches can be divided into two categories. The first are detection-based methods [2, 4, 14] that learn representations of the object segmented in the first frame and attempt to perform the pixel-wise detection of this object in future frames; the second is propagation-based methods [7, 12, 28, 33, 36] that formulate the task as a tracking problem and attempt to propagate the mask to fit the object over time. The first set of methods tends to segment single frames independently and rarely employ temporal information, while the later set segments single frames sequentially and makes use of temporal information, usually in the form of optical flow or RNNs. There has been some work on hybrid methods, that attempt to unify both approaches [32, 19, 38].

We propose a hybrid method that makes use of a *video capsule network* to segment a video conditioned on the segmented object in the first frame. A capsule is a group of neurons that represents an object, or part of an object. Layers in capsule networks undergo a routing-by-agreement algorithm that finds similarities between these capsules, and allow for the modeling of part-to-whole relationships. Capsule networks have performed well in image classification [26, 11], and have shown outstanding results in various segmentation tasks [18, 8]. In this paper, we leverage the segmentation ability of capsule networks and the ability of the routing algorithm to find similarity between capsules for the task of semi-supervised video object segmentation.

Our video capsule network, CapsuleVOS, contains two branches: a video branch and a frame branch. The video branch processes several frames at once and produces a set of video capsules. This allows the network learn temporal/motion information without the reliance of optical flow. The frame branch processes the first frame and object segmentation and generates a set of frame capsules, which model the object of interest. The frame branch makes use of a recurrent memory module that allows the network to

overcome issues like occlusion or objects exiting the scene.

Both sets of capsules are then passed through our novel *attention-routing procedure* which allows the frame capsules to condition the video capsules. Through this routing algorithm, our network learns where the object of interest is within the video clip, allowing the network to *segment multiple frames simultaneously*.

Moreover, our method makes use of a parametrized zooming module which allows the network to focus on regions of the frame which are relevant to the object of interest. This module allows for the segmentation of smaller objects, which can easily be lost when resizing frames to lower spatial dimensions.

We make the following contributions in this work,

- We present a novel capsule network for the task of video object segmentation that achieves state-of-the-art results on the largest video segmentation dataset.
- We propose a novel attention based EM routing algorithm to condition capsules based on an input segmentation.
- The proposed network contains integrated zooming module and memory module, which we show through experimental results to be effective for segmenting small and occluded objects in the video.

2. Related Work

Semi-supervised video object segmentation: Earlier works in video object segmentation used hand-crafted features based on appearance, boundary and optical flow [1, 9, 15, 27, 23]. The availability of large-scale video object segmentation datasets [25, 37] enabled us to explore deep learning methods for this problem. Most of the early works are mainly motivated by the image segmentation methods [3, 35, 20]. These works [2, 6, 16, 24, 38] lack the integration of sequential modelling which is important from video perspective. In some of these works, the temporal consistency is achieved by taking a guidance from the predicted mask of the previous frame [13, 24, 38]. The majority of recent works also utilize online learning [2] in which the segmentation networks are fine-tune on the first frame of each test video - this greatly improves segmentation results at the expense of inference speed.

Several recent works have utilized recurrent units to learn the evolution of objects over time. The authors in [28] use a ConvGRU to combine the outputs of pretrained appearance and a motion networks and generate a final segmentation. Similarly, the authors in [36] propose a ConvLSTM sequence-to-sequence model that learns to generate segmentations from sequences of frames. Ventura *et al*. [31] also use a ConvLSTM for recurrence in both the temporal domain (between frames) and the spatial domain (between object instances within each frame). Our use of a recurrent memory unit differs from these methods in that we do not generate segmentations directly from the features generated by the ConvLSTM, but rather condition a segmentation network based on these features.

Segmentation of small objects is challenging and zooming in on regions of the frame has been explored to overcome this problem. The authors in [7] demonstrated the effectiveness of processing only a tight region around the foreground object. Although this allows for improved segmentations, it assumes the object moves smoothly within the video - in cases of large motions, this may fail. Our approach can handle this issue, since our network learns the extent to which it must zoom in on the object of interest, allowing the network to learn these cases where large motions occur. The work in [5] performs segmentation by tracking parts - their network zooms in on and processes each part of the object separately. This requires multiple passes through their segmentation model, instead of having a single segmentation of the whole object.

Capsule networks: The idea of capsules was first introduced in [10], and they were popularized in [26], where dynamic routing for capsules was proposed. This was further extended in [11], where a more effective EM routing algorithm was introduced. Recently, capsule networks have shown state-of-the-art results for human action localization in video [8], object segmentation in medical images [18], and text classification [39]. In this work, we propose a capsule based network for video object segmentation where we introduce a novel attention based EM routing which can be used as a conditioning mechanism for capsules.

3. Our Approach

We propose an end-to-end trained network that segments an object throughout an entire video clip when given the object's segmentation mask for the first frame. This network contains two modules, depicted in Figures 1 and 2: a frame-conditioned video capsule network, CapsuleVOS, which segments a short video clip (8 frames) based on the object segmentation in the first frame, and a zooming module, which refines the spatial area processed by the capsule network. Section 3.1 explains how we leverage capsules for the task of video object segmentation, with our attention-based routing algorithm. We then describe the CapsuleVOS architecture and the zooming module in sections 3.2 and 3.3 respectively. This is followed by the objective function used to train this network in section 3.4.

3.1. Conditioning with Capsules

Capsules are groups of neurons that represent different entities or objects. In this work, we employ the version of

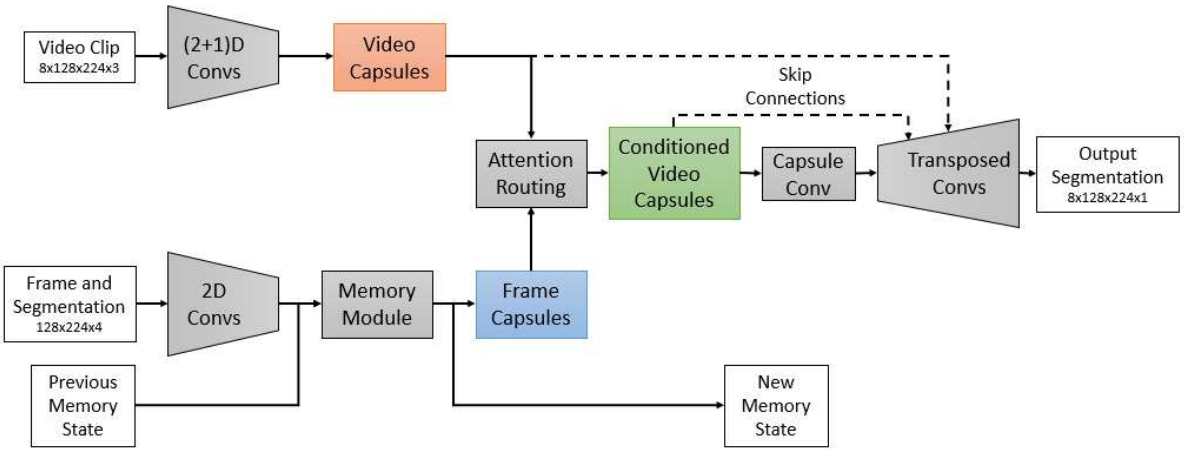

Figure 1. CapsuleVOS Architecture. The network is given the low resolution video clip and the segmented object in the first frame, and generates the foreground segmentations for all frames of the clip. The memory module consists of a ConvLSTM and allows the network to overcome issues like occlusion and objects leaving the frame. The previous and new memory states are the hidden and cell states of the ConvLSTM for time steps t and $t-1$ respectively. The new memory state is passed to the memory module for the following video clip.

capsules described in [11], which have a logistic unit (an activation denoted by a) representing the presence of the entity and a 4×4 pose matrix (denoted by M) which contains the properties of the entity. Capsules in one layer vote for the pose matrices of many capsules in the following layer and an iterative EM routing algorithm finds the agreement between the votes to create the set of capsules in the next layer. For a more comprehensive understanding of capsules, and the intuition behind them, we suggest reading [26, 11].

We view capsule networks' ability to model entities and find agreement between entities as an ideal mechanism to accomplish the semi-supervised video object segmentation task. A given video may contain several objects and the reference segmentation mask specifies the object which must be segmented. If we extract a set of capsules from both the video and the reference frame with a segmentation mask, then the former set (video capsules) models all objects within the video, while the latter set (frame capsules) represents the object of interest. Then, to obtain the object of interest throughout the video, one only needs to filter out all video capsules that are dissimilar to the frame capsules; in other words, an agreement, or similarity, between the video capsules and frame capsules would result in the set of video capsules that represent the object that must be segmented. Although the original EM routing algorithm works well for finding agreement within a set of capsules, it can not explicitly find agreement between two sets of capsules. For this reason, we propose an attention-based routing algorithm which finds the agreement between two sets of capsules.

Here, we use the query, key, value terminology found in [30], as our conditioning algorithm takes inspiration from this attention mechanism. From a video clip we extract a set of the video capsules $M_i^{\mathcal{V}}, a_i^{\mathcal{V}}$, indexed by i; from a reference frame and segmentation mask, we extract a set of frame capsules $M_k^{\mathcal{F}}, a_k^{\mathcal{F}}$, indexed by k. The key-value pairs are votes from the video capsules for the following layers' capsules while the query is the set of votes from the frame capsules. These votes are calculated as follows:

$$\begin{aligned} V_{ij}^k &= M_i^{\mathcal{V}} W_{ij}^k \\ V_{ij}^v &= M_i^{\mathcal{V}} W_{ij}^v \\ V_{kj}^q &= M_k^{\mathcal{F}} W_{kj}^q \end{aligned} \tag{1}$$

where W_{ij}^k, W_{ij}^v, and W_{kj}^q are learned weight matrices. The superscripts k, v, and q correspond to the key, value, and query respectively.

Once these votes are obtained, the EM routing operation is performed for the frame capsule (query) votes. This results in a set of higher-level capsules M_j^q, a_j^q, which represents the object, or parts of the object, in the reference segmentation mask. To find the similarity, or agreement, between the video capsules and the frame capsules, we measure the Euclidean distance between the key votes (V_{ij}^k) and their corresponding higher-level query capsule:

$$D_{ij} = \sum_h \left[\left(M_j^q - V_{ij}^k \right)^2 \right]^h, \tag{2}$$

where h denotes the dimensions of the vote and pose matrices.

This distance is used to compute an assignment coefficient

$$R_{ij}^v = \frac{e^{-D_{ij}}}{\sum_j e^{-D_{ij}}}. \tag{3}$$

The assignment coefficient, R_{ij}^v, determines the amount of information the i^{th} video capsule sends to the j^{th} higher-

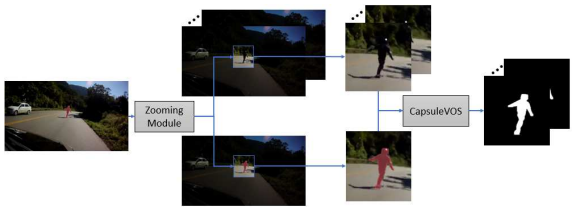

Figure 2. Zooming Module. Given the high-resolution first frame and segmentation mask, the zooming module outputs a bounding box around the object of interest. This bounding box is used to zoom in on the object in the video clip along with the first frame and segmentation mask, which are resized and passed into the CapsuleVOS network.

level capsule. If the distance, D_{ij}, is large, then the i^{th} video capsule does not contain information pertaining the the object represented by the j^{th} higher-level capsule, so its corresponding assignment coefficient is close to 0, and it sends less information to that higher-level capsule; conversely, a small distance leads to a large assignment coefficient, resulting in more information being sent.

We obtain the conditioned set of video capsules by performing the M-step of the EM routing algorithm using the value votes (V^v_{ij}) and the video capsules' assignment coefficients. The result is a set of higher-level video capsules, M^v_j, a^v_j, that receive information from lower-level video capsules which agree with the frame capsules. This procedure of conditioning with capsules is described in Algorithm 1.

Algorithm 1 This routing algorithm returns the activations and pose matrices of the capsules in layer $L+1$ when given the activations and poses of layer L (the video capsules and frame capsules). The indices i and j refer to the capsule types in layer L and $L+1$ respectively. The index h refers to the dimensions of the vote or pose matrices. The EM ROUTING and M-STEP functions referenced are those defined in [11].

procedure ATTROUTING($M^{\mathcal{V}}$, $a^{\mathcal{V}}$, $M^{\mathcal{F}}$, $a^{\mathcal{F}}$)
  $V^v \leftarrow M^{\mathcal{V}} W^v$
  $V^k \leftarrow M^{\mathcal{V}} W^k$
  $V^q \leftarrow M^{\mathcal{F}} W^q$
  $a^q, M^q \leftarrow$ EM ROUTING$(a^{\mathcal{F}}, V^q)$
  $D_{ij} \leftarrow \sum_h \left[\left(M^q_i - V^k_{ij} \right)^2 \right]^h$ ▷ For each i and j
  $R^v_{ij} \leftarrow \frac{e^{-D_{ij}}}{\sum_j e^{-D_{ij}}}$ ▷ For each i
  $a^v_j, M^v_j \leftarrow$ M-STEP$(a^{\mathcal{V}}, R^v, V^v, j)$ ▷ For each j
  return a^v, M^v

3.2. CapsuleVOS Architecture

The CapsuleVOS network segments 8 frames based on the segmentation mask of the first frame. It contains two branches - the video branch and the frame branch - and each creates sets of capsules. The video capsules are conditioned on the frame capsules, to produce a new set of conditioned capsules. These are followed by a convolutional capsule layer and a series of transposed convolutions to generate a segmentation map for all 8 frames.

The video branch passes the 8 RGB frames of size 128×224 through 6 (2+1)D convolutions [29] to obtain feature maps of size $8 \times 32 \times 56 \times 512$. The video capsules are composed of 12 capsule types, which are obtained by passing the feature maps to strided $3 \times 3 \times 3$ convolution operations.

The frame branch concatenates the first frame and the segmentation mask (each of size 128×224) and passes them through 4 2D convolutions. This is followed by the memory module, which consists of a ConvLSTM [34] layer that allows the frame branch to maintain information which might be lost in cases of occlusion or objects leaving the frame. The ConvLSTM produces a set of features of shape $32 \times 56 \times 128$ which are transformed into the frame capsules through a strided 3×3 convolution operation. The frame capsules, which are composed of 8 capsule types, are then tiled 8 times to match the temporal dimension of the video capsules.

Once the video and frame capsules have been formed, we perform capsule conditioning as described in Section 3.1, which results in a set of 16 capsule types. This is followed by a convolutional capsule layer that has 16 capsule types. All routing operations make use of capsule pooling [8] to reduce network's memory consumption.

To obtain a foreground segmentation mask from this capsule representations we flatten the capsules' pose matrices and pass them to a decoder composed of strided transposed convolutions. Skip connections from the video capsules and conditioned capsules are used to maintain spatiotemporal information which is lost from striding. The result of this decoder is 8 frames of binary segmentations corresponding to the object of interest.

3.3. Zooming Module

The zooming module is given the high-resolution first frame and the object of interest segmentation mask, and it outputs the bounding box containing the spatial region which our segmentation network will process. Since our segmentation network processes 8 frames at a time, the predicted bounding box must be large enough to contain the object of interest in all 8 frames, but not too large as to contain extraneous information not necessary for segmentation.

The input for the zooming module is a high-resolution frame (512×896) and the high-resolution binary object segmentation mask. These are passed through a series of strided 2D convolutional layers, a LSTM layer, and a fully-connected layer which outputs two values, $\hat{b}_h$ and $\hat{b}_w$, rep-

resenting the height and the width of the bounding box centered on the object of interest. The LSTM layer allows the network to learn from motion information from previous time steps, resulting in larger bounding boxes for objects with more motion, and tighter bounding boxes for objects with relatively little motion. Once the bounding box is obtained, the network extracts this region from the high-resolution segmentation mask and the next 8 frames of the high-resolution video; these are then resized to 128×224 and passed to CapsuleVOS.

3.4. Objective Function

For each pixel i in the video, we have ground-truth segmentations $y_i \in \{0, 1\}$ and our network predicts $\hat{y}_i \in [0, 1]$. We use both binary cross-entropy

$$L_s = -\frac{1}{N}\sum_{i=1}^{N} y_i \log\left(\hat{y}_i\right) + \left(1 - y_i\right) \log\left(1 - \hat{y}_i\right), \quad (4)$$

and the dice loss [21]

$$L_D = 1 - \frac{\sum_{i=1}^{N} \hat{y}_i y_i + \epsilon}{\sum_{i=1}^{N} \hat{y}_i + y_i + \epsilon} - \frac{\sum_{i=1}^{N} \left(1 - \hat{y}_i\right)\left(1 - y_i\right) + \epsilon}{\sum_{i=1}^{N} 2 - \hat{y}_i - y_i + \epsilon}, \quad (5)$$

to train the network for segmentation. The ϵ term is a small value to ensure stability of the loss. We use this second segmentation loss because video object segmentation methods are evaluated using region similarity, or intersection-over-union (IoU), and the dice loss directly maximizes this metric.

We train the zooming module by computing the L2 loss between the ground-truth bounding box height and width (b_h and b_w) and the predicted height and width ($\hat{b}_h$ and $\hat{b}_w$).

$$L_r = \left(b_h - \hat{b}_h\right)^2 + \left(b_w - \hat{b}_w\right)^2. \quad (6)$$

During training, we define the ground-truth height and width as the bounding box centered at the object in the first frame that contains the object in the following 7 frames (the other frames in the clip to be processed). This ensures that the object of interest will be present in all frames being processed, even if there is a large amount of motion.

In and end-to-end fashion, we train our network with an objective function which is the sum of these three losses:

$$L = L_s + L_D + L_r. \quad (7)$$

4. Experiments

Datasets We evaluate our method on two video object segmentation datasets: Youtube-VOS [37] and DAVIS-2017 [25]. Youtube-VOS contains 4,453 videos - 3,471 for training, 474 for validation, and 508 for testing. The training and validation videos have pixel-level ground truth annotations for every 5th frame (6 fps). The DAVIS-2017 dataset contains a total of 150 videos - 60 for training, 30 for validation, 60 for testing. These testing videos are split into a test-dev and test-challenge set, each with 30 videos; we evaluate our method on the test-dev set. The videos in DAVIS-2017 have annotations for all frames. Both datasets contain a wide variety of objects and both contain videos with multiple object instances.

Training The network is trained using the objective function described in 3.4. Since our segmentation loss requires segmentations for all 8 frames given to the network and the Youtube-VOS training set contains segmentations every 5th frame, we use the method found in [22] to interpolate the segmentation frames that are unavailable. Training is done using the Adam optimizer [17], starting with a learning rate of 0.0001. When training on Youtube-VOS, the method converges in about 400 epochs. For our experiments on DAVIS-2017, we fine-tune the network for an extra 200 epochs on the DAVIS-2017 training videos.

Inference During inference, longer videos are processed one clip (8 frames) at a time; the segmentation generated from one clip is used as the input segmentation for the subsequent clip. We find that having frame overlaps between these clips results in improved segmentations at test time, with only a minor decrease of inference speed. All reported results (both accuracy and speed) use an overlap of 3 frames.

Evaluation Metrics For both datasets, we evaluate the segmentation results using the region similarity $\mathcal{J}$ and the contour accuracy $\mathcal{F}$ as described in [24]. For Youtube-VOS, results are averaged over the "*seen*" categories - those objects found in training videos - and "*unseen*" categories - the objects present in the validation and testing sets but not present in the training set.

4.1. Comparison with State-of-the-art

Since our method does not use online learning, we compare with only offline approaches. The exception to this is OSVOS [2], which is a standard benchmark video object segmentation approach.

Youtube-VOS The performance of our network on Youtube-VOS are shown in Table 1. Overall, our model performs at least 4% better than all offline methods and 3.5% better than OSVOS. OSVOS slightly outperforms us on unseen categories, but our network has a substantial 8% improvement in both of the "seen" metrics. Some qualitative results on Youtube-VOS videos are shown in Figure 3.

Figure 3. Qualitative results showing object segmentations on videos from the Youtube-VOS validation set. The first three rows contain examples in which multiple instances of objects are present within the video; the later two show how our network is able to finely segment larger objects.

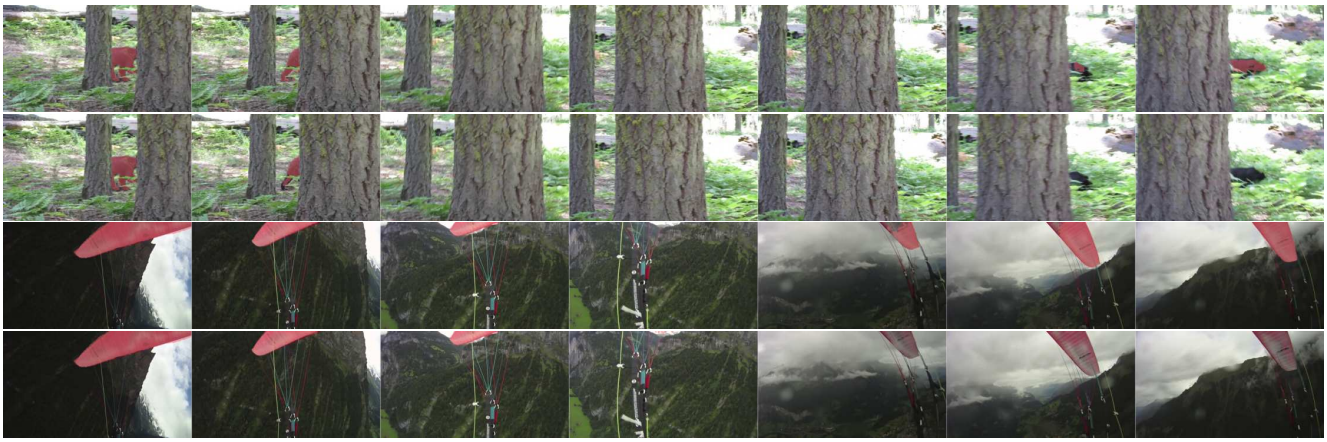

Figure 4. A qualitative comparison between networks with and without the memory module. Rows 1,3: with memory module. Rows 2,4: without memory module. The first example contains a bear which is completely occluded for over 40 frames, but the memory module allows the network to segment the bear when it reappears. The second video shows that the memory module can handle cases where an object leaves and reenters the scene.

Method	OL	$\mathcal{J}$ seen	$\mathcal{J}$ unseen	$\mathcal{F}$ seen	$\mathcal{F}$ unseen	Overall	Speed (frames/s)
OSVOS [2]	✓	59.8	**54.2**	60.5	**60.7**	58.8	0.10
OSMN [38]	✗	60.0	40.6	60.1	44.0	51.2	7.14
S2S (offline) [36]	✗	66.7	48.2	65.5	50.3	57.6	6.25
Our Method	✗	**67.3**	53.7	**68.1**	59.9	**62.3**	**13.5**

Table 1. Our results on the Youtube-VOS validation set. "OL" denotes online learning. We compare with OSVOS [2] and methods which do not perform online learning.

DAVIS-2017 Our performance on the DAVIS-2017 test-dev set are shown in Table 2. We find that our offline network is unable to achieve better results than many contemporary methods because many of the objects found in DAVIS-2017 do not appear in the Youtube-VOS training set. DyeNet [19] is able to outperform our network by a wide margin; we attribute this to the fact that the method is image based, which allows their region-proposal network

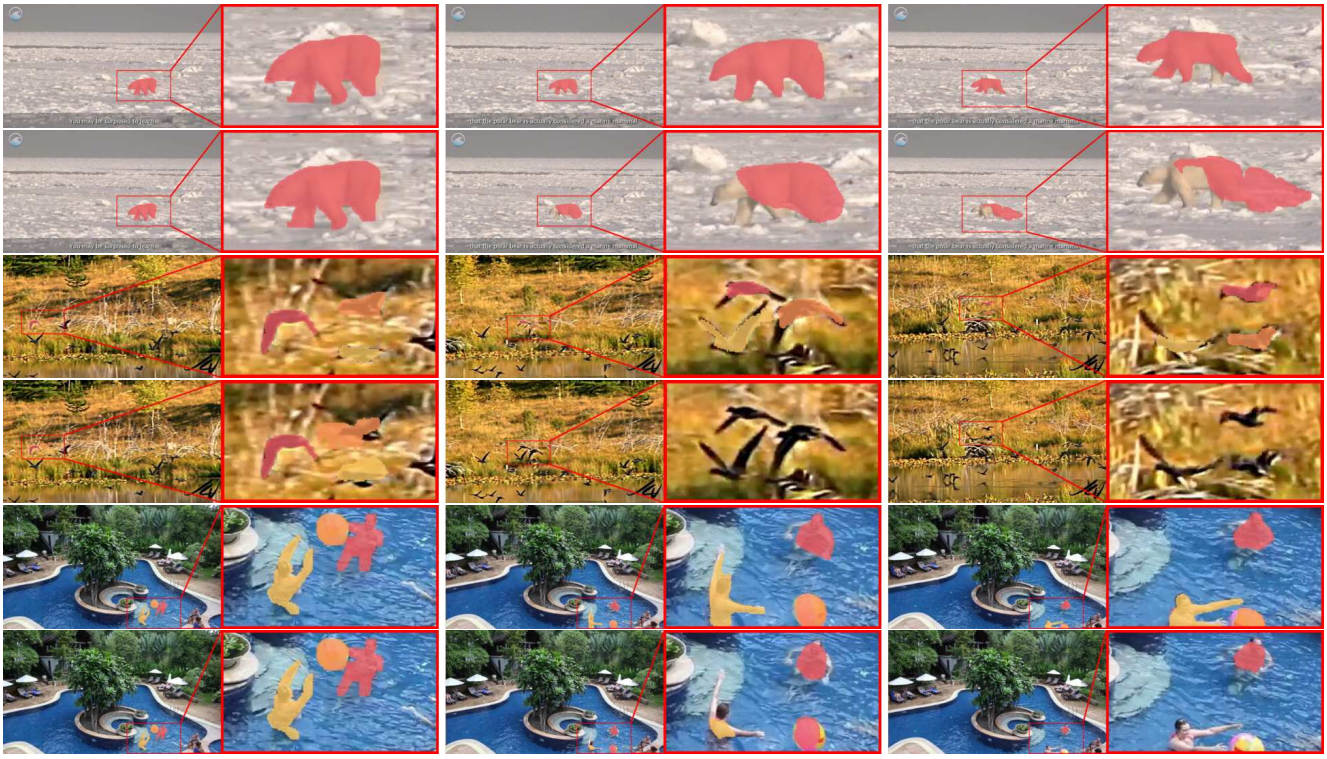

Figure 5. A qualitative comparison between networks with and without the zooming module. Rows 1,3,5: with zooming module. Rows 2,4,6: without zooming module. The first example demonstrates the network's ability to generate fine-grained segmentations on small objects when the zooming module is used. Very small objects that move rapidly, like those in examples 2 and 3, are lost rather quickly when the zooming module is not present.

	OSVOS [2]	DyeNet [19]	Ours
Online Learning	✓	✗	✗
$\mathcal{J}$ Mean ↑	47.2	60.2	47.4
$\mathcal{J}$ Recall ↑	50.8	-	54.1
$\mathcal{F}$ Mean ↑	53.7	64.8	55.2
$\mathcal{F}$ Recall ↑	57.8	-	64.6
Global Mean	50.5	62.5	51.3

Table 2. Our results on the DAVIS-2017 test-dev set. We compare with OSVOS [2] and the offline version of DyeNet [19]

and feature extraction network to be pretrained on larger image datasets.

Speed Analysis Running on a Titan X Pascal GPU, our network segments an average of 13.5 frames per second. We compare our network's inference speed with other approaches in Figure 6. Our network is able to segment frames at a much faster rate than previous methods, because we simultaneously segment 8 frames at once as opposed to one frame at a time.

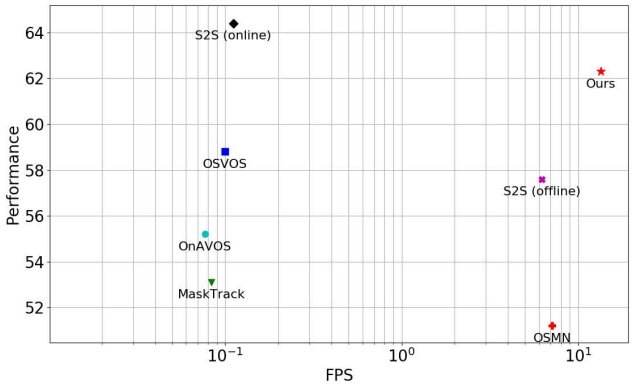

Figure 6. Comparison of quality and speed of previous video object segmentation methods on the Youtube-VOS dataset. We graph the overall performance percentage vs the frames-per-second. The x-axis (fps) is in the log scale.

4.2. Ablation Study

All ablation experiments are performed on the Youtube-VOS dataset. The quantitative results for the ablations are shown in Table 3.

Ablation	$\mathcal{J}$ seen	$\mathcal{J}$ unseen	$\mathcal{F}$ seen	$\mathcal{F}$ unseen	Overall
No Zooming	62.1	45.8	61.3	48.1	54.3
HC Zooming	65.8	51.7	66.5	57.5	60.4
Concat Routing	65.2	51.0	65.6	56.9	59.7
Fully Conv	64.5	51.5	64.8	57.0	59.4
No Memory	64.9	49.6	65.3	53.9	58.4
Full Method	67.3	53.7	68.1	59.9	62.3

Table 3. Our ablation experiment results on the Youtube-VOS validation set. Each row corresponds to a different ablation. The final row contains the results of our method without any changes.

Zooming Module To test the effectiveness of our zooming module, we first evaluate our method without any zooming. In this experiment, we resize all frames to 128×224 and segment them with CapsuleVOS. Without the zooming module, the network's performance decreased by about 8%. The zooming module improves the segmentations in two ways: (1) the network is able to keep track of smaller objects, and (2) the network can generate finer segmentation masks for medium sized objects. Figure 5 shows examples of our method with and without the zooming module; there is a noticeable decrease in segmentation accuracy for smaller objects without the zooming module. We also test if if a simple, hand-crafted zooming method would perform as well as our zooming module. In this experiment, we use a hand-crafted bounding-box around the foreground object in lieu of the zooming module. We find that the hand-crafted bounding-box results in improved segmentations when compared to no zooming, but the zooming module's learned bounding-boxes perform best.

Attention Routing We run two ablations to test the effectiveness of our proposed capsule routing algorithm. The first is performing conventional EM-routing by simply concatenating the video and frame capsules; the second is removing capsules entirely, and having a fully convolutional network with a similar number of parameters. We find that our proposed routing algorithm does improve segmentations when compared to simple capsule concatenation; this is because the proposed routing algorithm conditions the video capsules based on their agreement with the frame capsules, whereas concatenation does not differentiate between frame and video capsules and attempts to find agreement between all capsules. We also find that the network without capsules performs similar to the network with capsule concatenation; this suggests that the standard EM routing algorithm cannot effectively perform the conditioning operation which this tasks requires and that our proposed routing procedure successfully conditions the video capsules based on the frame capsules.

Memory Module In this final ablation, we test the importance of the memory module in the frame network. We find that this ConvLSTM improves results by 4%, because it allows the network to handle issues like occlusion or when the object of interest leaves the frame. Figure 4 contains some qualitative results depicting the two issues that the memory module solves: occlusion and objects leaving the frame. Once the occlusion ends or the object re-enters the frame, the ConvLSTM allows the network to remember the object which it must segment.

5. Conclusion

We have proposed a video capsule network, CapsuleVOS, for semi-supervised video object segmentation. The use of capsules provides an effective modeling of entities present in the video and the attention-based routing helps in the tracking and segmentation of objects. The network contains two additional novel components: a zooming module and a memory module. The zooming module ensures the capture of small objects present in the video and the memory module tracks objects in scenarios when they are occluded or when they move out of the scene. The experimental evaluation demonstrates the effectiveness of our proposed network in video object segmentation and its ability to segment small and occluded objects. Moreover, our ablations show the effectiveness of our proposed routing procedure when compared to the exists EM routing algorithm. The network segments multiple frames at once which allows it to perform segmentation at a much faster rate when compared with existing methods.

5.1. Acknowledgement

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. D17PC00345. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

References

[1] 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. 2

[2] 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. 1, 2, 5, 6, 7

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 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, 2018. 2

[4] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1189–1198, 2018. 1

[5] Jingchun Cheng, Yi-Hsuan Tsai, Wei-Chih Hung, Shengjin Wang, and Ming-Hsuan Yang. Fast and accurate online video object segmentation via tracking parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7415–7424, 2018. 2

[6] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Proceedings of the IEEE international conference on computer vision*, pages 686–695, 2017. 2

[7] Hai Ci, Chunyu Wang, and Yizhou Wang. Video object segmentation by learning location-sensitive embeddings. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 501–516, 2018. 1, 2

[8] Kevin Duarte, Yogesh Rawat, and Mubarak Shah. Videocapsulenet: A simplified network for action detection. In *Advances in Neural Information Processing Systems*, pages 7621–7630, 2018. 1, 2, 4

[9] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *BMVC*, volume 2, page 8, 2014. 2

[10] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer, 2011. 2

[11] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. 2018. 1, 2, 3, 4

[12] Ping Hu, Gang Wang, Xiangfei Kong, Jason Kuen, and Yap-Peng Tan. Motion-guided cascaded refinement network for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1400–1409, 2018. 1

[13] Yuan-Ting Hu, Jia-Bin Huang, and Alexander Schwing. Maskrnn: Instance level video object segmentation. In *Advances in Neural Information Processing Systems*, pages 325–334, 2017. 2

[14] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 54–70, 2018. 1

[15] Suyog Dutt Jain and Kristen Grauman. Supervoxel-consistent foreground propagation in video. In *European Conference on Computer Vision*, pages 656–671. Springer, 2014. 2

[16] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2126. IEEE, 2017. 2

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[18] Rodney LaLonde and Ulas Bagci. Capsules for object segmentation. *arXiv preprint arXiv:1804.04241*, 2018. 1, 2

[19] Xiaoxiao Li and Chen Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 90–105, 2018. 1, 6, 7

[20] 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. 2

[21] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, 2016. 5

[22] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 261–270, 2017. 5

[23] 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. 2

[24] 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. 2, 5

[25] 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. 2, 5

[26] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017. 1, 2, 3

[27] Naveen Shankar Nagaraja, Frank R. Schmidt, and Thomas Brox. Video segmentation with just a few strokes. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2

[28] 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. 1, 2

[29] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 4

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 3

[31] 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. 2

[32] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7376–7385, 2018. 1

[33] Huaxin Xiao, Jiashi Feng, Guosheng Lin, Yu Liu, and Maojun Zhang. Monet: Deep motion exploitation for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1140–1148, 2018. 1

[34] 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. 4

[35] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 373–381, 2016. 2

[36] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 585–601, 2018. 1, 2, 6

[37] 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. 1, 2, 5

[38] 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. 1, 2, 6

[39] Wei Zhao, Jianbo Ye, Min Yang, Zeyang Lei, Suofei Zhang, and Zhou Zhao. Investigating capsule networks with dynamic routing for text classification. *arXiv preprint arXiv:1804.00538*, 2018. 2