Video Object Segmentation
-
Multi/Single object
With/Without first frame annotation

Haller Emanuela
ehaller@bitdefender.com

24 September 2019

Table of contents

Xu et al. [12], Ventura et al. [10]

Table of contents

- **"YouTube-VOS: Sequence-to-Sequence Video Object Segmentation" [12] - ECCV 2018**
- "RVOS: End-to-End Recurrent Network for Video Object Segmentation" [10] - CVPR 2019

Xu et al. [12], Ventura et al. [10]

YouTube-VOS: Sequence-to-Sequence Video Object Segmentation

- Semi-supervised VOS task
- End-to-end sequential learning to explore spatio-temporal features for video object segmentation
- Introduce a large-scale video object segmentation dataset: **YouTube-VOS**
  - more than 3000 videos
  - 3-6 seconds
  - multiple objects manually annotated

YouTube-VOS

Table 1: Scale comparison between YouTube-VOS and existing datasets. "Annotations" denotes the total number of object annotations. "Duration" denotes the total duration (in minutes) of the annotated videos.

Scale	JC [13]	ST [26]	YTO [21]	FBMS [30]	DAVIS [33]	DAVIS [34]	**YouTube-VOS (Ours)**
Videos	22	14	96	59	50	90	**3,252**
Categories	14	11	10	16	-	-	**78**
Objects	22	24	96	139	50	205	**6,048**
Annotations	6,331	1,475	1,692	1,465	3,440	13,543	**133,886**
Duration	3.52	0.59	9.01	7.70	2.88	5.17	**217.21**

- Videos from YouTube-8M [1]
  - Up to 100 videos for each category
  - Automatically split videos in short clips
- Some object categories:
  - animals (e.g. ant, eagle, goldfish, person)
  - vehicles (e.g. airplane, bicylce, boat, sedan)
  - accesories (e.g. eyeglass, hat, bag)
  - common objects (e.g. tennis, skateboarding, motorcycling, umbrella)

Abu-El-Haija et al. [1] - 2016

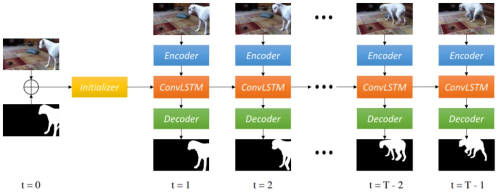
YouTube-VOS

Fig. 1: The ground truth annotations of sample video clips in our dataset. Different objects are highlighted with different colors.

S2S: Problem formulation

- Given:
  - $\{\mathbf{x}_t | t \in [0, T-1]\}$ - video sequence with T frames
  - $\mathbf{x}_t \in \mathbb{R}^{H \times W \times 3}$ - frame t
  - $\mathbf{y}_0 \in \mathbb{R}^{H \times W}$ - first frame binary mask
- Find:
  - $\{\hat{\mathbf{y}}_t | t \in [1, T-1]\}$
  - $\hat{\mathbf{y}}_t = \arg\max_{\forall \overline{\mathbf{y}}_t} \mathbb{P}(\overline{\mathbf{y}}_t | \mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_t, \mathbf{y}_0)$
- Other solutions model:
  - $\hat{\mathbf{y}}_t = \arg\max_{\forall \overline{\mathbf{y}_t}} \mathbb{P}(\overline{\mathbf{y}}_t | \mathbf{x}_0, \mathbf{x}_t, \mathbf{y}_0)$
    or
  - $\hat{\mathbf{y}}_t = \arg\max_{\forall \overline{\mathbf{y}_t}} \mathbb{P}(\overline{\mathbf{y}}_t | \mathbf{x}_0, \mathbf{y}_0, \mathbf{x}_t, \mathbf{x}_{t-1})$

S2S: Architecture (1)



$$\mathbf{c}_0, \mathbf{h}_0 = \text{Initializer}(\mathbf{x}_0, \mathbf{y}_0)$$

$$\widetilde{\mathbf{x}}_t = \text{Encoder}(\mathbf{x}_t)$$

$$\mathbf{c}_t, \mathbf{h}_t = \text{ConvLSTM}(\widetilde{\mathbf{x}}_t, \mathbf{c}_{t-1}, \mathbf{h}_{t-1})$$

$$\hat{\mathbf{y}}_t = \text{Decoder}(\mathbf{h}_t)$$

S2S: Architecture (2)

- Backbone:
  - VGG-16 [9]
- Initializer:
  - Backbone + 2 convolutional layers
- Encoder:
  - Backbone + 1 convolutional layer
- ConvLSTM:
  - 3x3 filters
  - ReLU for state outputs
- Decoder:
  - 5 upsampling layers

- Backbone

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096				1x1 convs	
~~FC-4096~~					
~~FC-1000~~					
soft-max					

Simonyan et al. [9] - ICLR 2015

S2S: Training

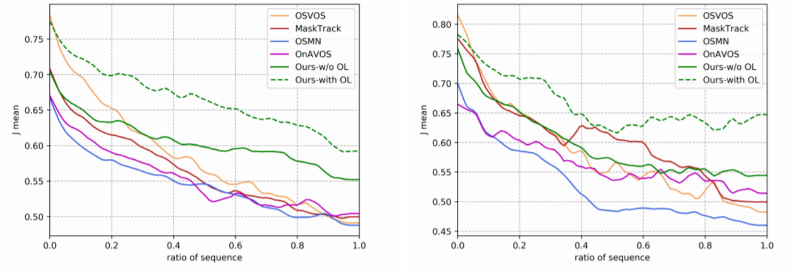
- Offline training
  - Training set of YouTube-VOS
  - For each iteration, randomly select an object and T (5 - 11) frames from a random training video sequence
  - RGB images: 256x448
  - YouTube-VOS doesn't provide annotations for all frames
  - S2S training:
    - Early stages use only annotated frames
    - When the training losses become stable, add frames without annotations
- Online training
  - Update Initializer, Encoder and Decoder
  - From $(\mathbf{x}_0, \mathbf{y}_0)$, through affine transformations, generate possible future frames and corresponding annotations $\Rightarrow$ training pairs $(\mathbf{x}_0, \mathbf{y}_0) - (\mathbf{x}_1, \mathbf{y}_1)$

S2S: Experiments
YouTube-VOS

Table 2: Comparisons of our approach and other methods on YouTube-VOS test set. The results in each cell show the test results for seen/unseen categories. "OL" denotes online learning. The best results are highlighted in bold.

Method	$\mathcal{J}$ mean↑	$\mathcal{J}$ recall↑	$\mathcal{J}$ decay↓	$\mathcal{F}$ mean↑	$\mathcal{F}$ recall↑	$\mathcal{F}$ decay↓
SegFlow [8]	40.4/38.5	45.4/41.7	**7.2/8.4**	35.0/32.7	35.3/32.1	**6.9/9.1**
OSVOS [6]	59.1/58.8	66.2/64.5	17.9/19.5	63.7/63.9	69.0/67.9	20.6/23.0
MaskTrack [32]	56.9/60.7	64.4/69.6	13.4/16.4	59.3/63.7	66.4/73.4	16.8/19.8
OSMN [50]	54.9/52.9	59.7/57.6	10.2/14.6	57.3/55.2	60.8/58.0	10.4/13.8
OnAVOS [44]	55.7/56.8	61.6/61.5	10.3/9.4	61.3/62.3	66.0/67.3	13.1/12.8
Ours (w/o OL)	60.9/60.1	70.3/71.2	7.9/12.9	64.2/62.3	73.0/71.4	9.3/14.5
Ours (with OL)	**66.9/66.8**	**78.7/76.5**	10.2/9.5	**74.1/72.3**	**82.8/80.5**	12.6/13.4

S2S: Experiments
YouTube-VOS



(a) Seen categories

(b) Unseen categories

Fig. 3: The changes of $\mathcal{J}$ mean values over the length of video sequences.

S2S: Experiments
YouTube-VOS

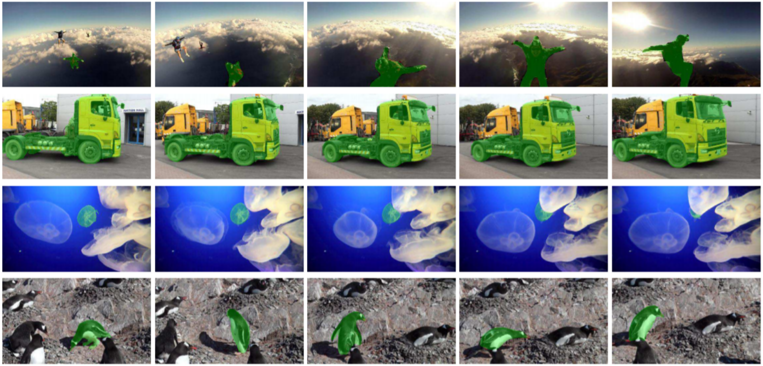


Fig. 4: Some visual results produced by our model without online learning on the YouTube-VOS test set. The first column shows the initial ground truth object segmentation (green color) while the second to the last column are predictions.

S2S: Experiments DAVIS-2016

- Fine-tune the model on DAVIS training set - 30 videos

Table 3: Comparisons of our approach and previous methods on the DAVIS 2016 dataset. Different components used in each algorithm are marked. “OL” denotes online learning. “PP” denotes post processing by CRF [25] or Boundary Snapping [6]. “OF” denotes optical flows. “RNN” denotes RNN and its variants.

Method	OL	PP	OF	RNN	mean IoU(%)	Speed(s)
BVS [43]	-	✗	✗	-	60.0	0.37
OFL [27]	-	✓	✓	-	68.0	42.2
SegFlow [8]	✓	✓	✓	✗	76.1	7.9
MaskTrack [32]	✓	✓	✗	✗	79.7	12
OSVOS [6]	✓	✓	✗	✗	79.8	10
OnAVOS [44]	✓	✓	✗	✗	**85.7**	13
OSMN [50]	✗	✗	✗	✗	74.0	**0.14**
VPN [22]	✗	✗	✗	✗	70.2	0.63
ConvGRU [41]	✗	✓	✓	✓	75.9	20
Ours	✗	✗	✗	✓	76.5	0.16
Ours	✓	✗	✗	✓	79.1	9

S2S: Experiments
DAVIS-2016

Fig. 5: The comparison results between our model without online learning (upper row) and with online learning (bottom row). Each column shows predictions of the two models at the same frame.

S2S: Limitations of other datasets

- Training settings:
  - Setting 1: Train S2S on 30 videos (DAVIS-2016)
  - Setting 2: Train S2S on 192 videos (DAVIS-2016, SegTrackv2, JumpCut and YouTube-Objects)
  - Setting 3: Encoder - pretrained DeepLab [2], other components trained on 30 videos (DAVIS-2016)
- Test on validation set of DAVIS-2016
  - Setting 1: 51.3% JMean
  - Setting 2: 51.9% JMean
  - Setting 3: 45.6% JMean

Chen et al. [2] - TPAMI 2017

S2S: Ablation study
Dataset scale

Table 4: The effect of data scale on our algorithm. We use different portions of training data to train our models and evaluate on the YouTube-VOS test set.

Scale	$\mathcal{J}$ mean↑	$\mathcal{J}$ recall↑	$\mathcal{J}$ decay↓	$\mathcal{F}$ mean↑	$\mathcal{F}$ recall↑	$\mathcal{F}$ decay↓
25%	46.7/40.1	53.5/45.6	8.3/13.6	46.7/40.0	52.2/41.6	8.5/13.2
50%	51.5/50.3	59.2/58.8	10.3/13.1	51.8/50.2	59.5/55.8	11.1/13.3
75%	56.8/56.0	65.7/67.1	7.6/10.0	59.6/56.3	68.8/64.1	8.5/11.1
100%	60.9/60.1	70.3/71.2	7.9/12.9	64.2/62.3	73.0/71.4	9.3/14.5

S2S: Ablation study
Initializer variants

- With Initializer
  - Seen categories: 60.9%
  - Unseen categories: 60.1%
- Without Initializer - initial mask as hidden state
  - Seen categories: 45.1%
  - Unseen categories: 38.6%

S2S: Ablation study
Encoder variants

- Use previous masks as input to the Encoder, in addition to current frame
- Same model for Initializer and Encoder - VGG-16 based architecture

- Original:
  - Seen categories: 60.9%
  - Unseen categories: 60.1%
- Merge Initializer & Encoder
  - Seen categories: 59.4%
  - Unseen categories: 60.7%

Table of contents

- "YouTube-VOS: Sequence-to-Sequence Video Object Segmentation" [12] - ECCV 2018
- **"RVOS: End-to-End Recurrent Network for Video Object Segmentation" [10] - CVPR 2019**

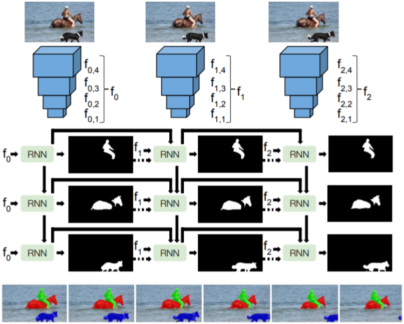
Xu et al. [12], Ventura et al. [10]

RVOS: End-to-End Recurrent Network for Video Object Segmentation

- Recurrent network for multiple object Video Object Segmentation - RVOS
- Zero-shot / Unsupervised VOS task
- Extension for One-Shot / Semi-supervised VOS task
- Datasets:
  - DAVIS-2017
  - YouTube-VOS

RVOS

- Recurrent model - spatial and temporal domains
- Trainable end-to-end and handles multiple objects in a unified manner
- Handles both Zero-shot and One-shot VOS tasks
- Good results



RVOS - related work

- RIS - "Recurrent Instance Segmentation" [6] - ECCV 2016
- RSIS - "Recurrent Neural Networks for Semantic Instance Segmentation" [8] - arXiv 2019
- RVOS - adds recurrence in the temporal domain on top of RSIS

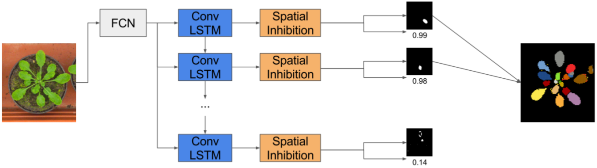
Romera et al. [6], Salvador et al. [8]

RVOS - related work

- **RIS - "Recurrent Instance Segmentation" [6] - ECCV 2016**
- RSIS - "Recurrent Neural Networks for Semantic Instance Segmentation" [8] - arXiv 2019
- RVOS - adds recurrence in the temporal domain on top of RSIS

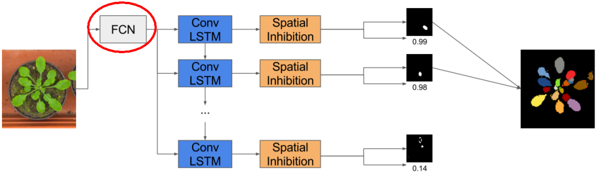
Romera et al. [6], Salvador et al. [8]

RIS



- New instance segmentation paradigm: an end-to-end method that learns how to segment instances sequentially
- Input
  - image $\mathbf{I} \in \mathbb{R}^{h \times w \times c}$
- Output
  - sequence of masks: $\mathbf{Y} = \{\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_n\}$, $\mathbf{Y}_t \in [0,1]^{h \times w}$
  - confidence scores: $s = \{s_1, s_2, ..., s_n\}$, $s_t \in [0,1]$

RIS - Fully Convolutional Network [4]



- $\mathbf{I} \in \mathbb{R}^{h \times w \times c} \Rightarrow \mathbf{B} \in \mathbb{R}^{h' \times w' \times d}$

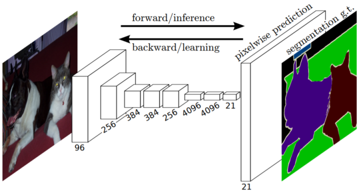
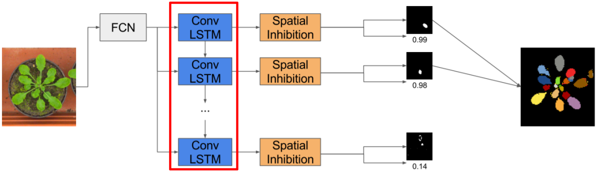


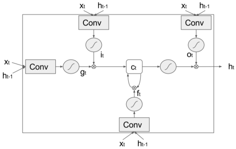
Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

Long et al. [4] CVPR 2015

RIS - ConvLSTM 2 [11]



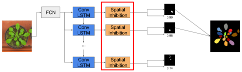
- $\mathbf{i}_t = \sigma(\mathbf{W}_{xi} * \mathbf{B} + \mathbf{W}_{hi} * \mathbf{H}_{t-1} + \mathbf{b}_i)$
- $\mathbf{f}_t = \sigma(\mathbf{W}_{xf} * \mathbf{B} + \mathbf{W}_{hf} * \mathbf{H}_{t-1} + \mathbf{b}_f)$
- $\mathbf{o}_t = \sigma(\mathbf{W}_{xo} * \mathbf{B} + \mathbf{W}_{ho} * \mathbf{H}_{t-1} + \mathbf{b}_o)$
- $\mathbf{g}_t = tanh(\mathbf{W}_{xg} * \mathbf{B} + \mathbf{W}_{hg} * \mathbf{H}_{t-1} + \mathbf{b}_g)$
- $\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t$
- $\mathbf{H}_t = \mathbf{o}_t \odot tanh(\mathbf{C}_t)$

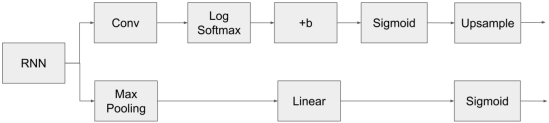


- $\mathbf{H}_t, \mathbf{C}_t \in \mathbb{R}^{h' \times w' \times d}$

Shi et al. [11] NIPS 2015

RIS - Attention by Spatial Inhibition





- $r : \mathbb{R}^{h' \times w' \times d} \to [0,1]^{h \times w}, [0,1]$
- Discriminate one instance
  - conv: d channels $\Rightarrow$ 1 channel
  - log softmax: values in $(-\infty, 0]$ with sum of exponentials 1
    - competing mechanism, inhibiting pixels that do not belong to current instance
  - b: learned threshold
- Compute confidence of predicted candidate

RIS - Loss Function (1)

- Dataset:
  - $\mathbf{I}^{(i)} \in \mathbb{R}^{h \times w \times c}$
  - $\mathbf{Y}^{(i)} = \{\mathbf{Y}_1{}^{(i)}, \mathbf{Y}_2{}^{(i)}, ..., \mathbf{Y}_{n_i}{}^{(i)}\}$, $\mathbf{Y_t}{}^{(i)} \in \{0,1\}^{h \times w}$
- Predictions:
  - $\hat{\mathbf{Y}}^{(i)} = \{\hat{\mathbf{Y}}_1^{(i)}, \hat{\mathbf{Y}}_2^{(i)}, ..., \hat{\mathbf{Y}}_{\hat{n}_i}^{(i)}\}$, $\hat{\mathbf{Y}}_t^{(i)} \in [0,1]^{h \times w}$
  - $s^{(i)} = \{s_1{}^{(i)}, s_2{}^{(i)}, ..., s_{\hat{n}_i}{}^{(i)}\}$
- $s_t{}^{(i)} < 0.5 \Rightarrow$ networks stops producing outputs
- Usually, $\hat{n}_i \neq n_i$; for training: $\hat{n}_i = n_i + 2$ - in order to learn when to stop

RIS - Loss Function (2)

$$\max_{\delta \in \mathcal{S}} f_{\text{Match}}(\hat{\mathbf{Y}}, \mathbf{Y}, \delta),$$

where

$$f_{\text{Match}}(\hat{\mathbf{Y}}, \mathbf{Y}, \delta) = \sum_{\hat{t}=1}^{\tilde{n}} \left(\sum_{t=1}^{n} f_{\text{IoU}} \left(\hat{\mathbf{Y}}_{\hat{\mathbf{t}}}, \mathbf{Y}_{\mathbf{t}} \right) \delta_{\hat{t},t} \right),$$

$$\mathcal{S} = \left\{ \begin{array}{ll} \delta \in \{0,1\}^{\tilde{n} \times n} : & \sum_{\hat{t}=1}^{\tilde{n}} \delta_{\hat{t},t} \leq 1, \ \forall t \in \{1 \dots n\} \\ & \sum_{t=1}^{n} \delta_{\hat{t},t} \leq 1, \ \forall \hat{t} \in \{1 \dots \tilde{n}\} \end{array} \right\},$$

- Optimal δ with Hungarian algorithm
- $\widetilde{n} = min(\hat{n}, n)$
- $f_{IoU}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{\langle \hat{\mathbf{y}}, \mathbf{y} \rangle}{\|\hat{\mathbf{y}}\|_1 + \|\mathbf{y}\|_1 + \langle \hat{\mathbf{y}}, \mathbf{y} \rangle}$ - relaxed version of IoU
- s_t should be 1 as long as $t \leq n$

RIS - Loss Function (3)

$$\ell(\hat{\mathbf{Y}}, \mathbf{s}, \mathbf{Y}) = \min_{\delta \in \mathcal{S}} - \sum_{\hat{t}=1}^{\tilde{n}} \sum_{t=1}^{n} f_{\mathrm{IoU}} \left(\hat{\mathbf{Y}}_{\hat{\mathbf{t}}}, \mathbf{Y}_{\mathbf{t}} \right) \delta_{\hat{t},t} + \lambda \sum_{t=1}^{\hat{n}} f_{\mathrm{BCE}} \left([t \leq n], s_t \right), \qquad (5)$$

where $f_{\mathrm{BCE}}(a, b) = -(a\log(b) + (1-a)\log(1-b))$ is the binary cross entropy, and the Iverson bracket $[\cdot]$ is 1 if the condition within the brackets is true, and 0 otherwise. Finally, λ is a hyperparameter that ponders the importance of the second term with respect to the first one.

RVOS - related work

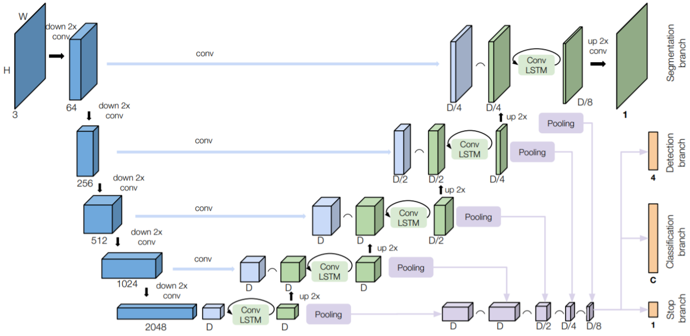
- RIS - "Recurrent Instance Segmentation" [6] - ECCV 2016
- **RSIS - "Recurrent Neural Networks for Semantic Instance Segmentation" [8] - arXiv 2019**
- RVOS - adds recurrence in the temporal domain on top of RSIS

Romera et al. [6], Salvador et al. [8]

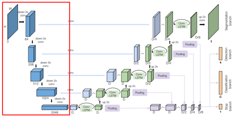
- Semantic instance segmentation
- Input:
  - image $x \in \mathbf{R}^{h \times w \times c}$
- Output:
  - $\hat{y} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_{\hat{n}}\}$
  - $\hat{y}_t = \{\hat{y}_m, \hat{y}_b, \hat{y}_c, \hat{y}_s\}$
    - mask: $\hat{y}_m \in [0, 1]^{h \times w}$
    - bounding box: $\hat{y}_b \in [0, 1]^4$
    - class probabilities: $\hat{y}_c \in [0, 1]^C$
    - objectness score: $\hat{y}_s \in [0, 1]$ - stopping criterion

- Semantic instance segmentation
- Input:
  - image $x \in \mathbf{R}^{h \times w \times c}$
- Output:
  - $\hat{y} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_{\hat{n}}\}$
  - $\hat{y}_t = \{\hat{y}_m, \hat{y}_b, \hat{y}_c, \hat{y}_s\}$
    - mask: $\hat{y}_m \in [0,1]^{h \times w}$
    - **bounding box:** $\hat{y}_b \in [0,1]^4$
    - **class probabilities:** $\hat{y}_c \in [0,1]^C$
    - objectness score: $\hat{y}_s \in [0,1]$ - stopping criterion

RSIS - Encoder-Decoder Architecture



- ResNet-101 [3], pretrained on ImageNet [7]

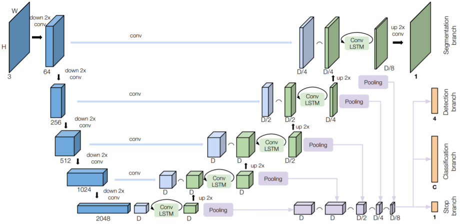


layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	~~average pool, 1000-d fc, softmax~~				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

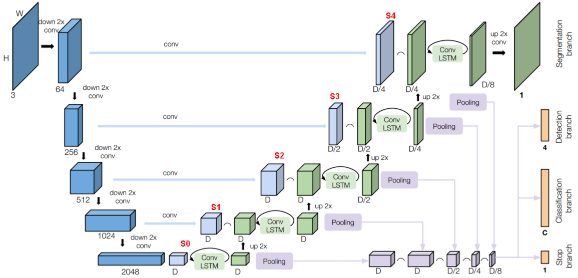
He et al. [3] - CVPR 2016, Russakovsky et al. [7] - IJCV 2015

RSIS - Decoder (1)

- Hierarchical recurrent architecture
- Upsampling network composed of a series of ConvLSTM layers
- Skip connections that bypass the previous recurrent layers
- Reliance on the features changes across different time steps



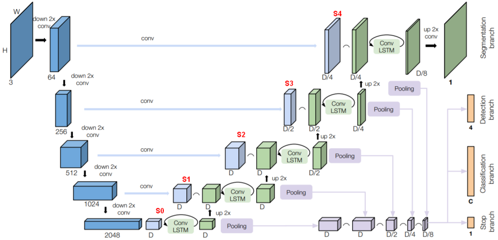
RSIS - Decoder (2)



- $h_{i,t} = \text{ConvLSTM}_i([B_2(h_{i-1,t})|S_i], h_{i,t-1})$
- $h_{0,t} = \text{ConvLSTM}_0(S_0, h_{0,t-1})$

- B_2 - bilinear upsampling operator

RSIS - Decoder (3)



- ConvLSTMs: 3×3 kernels
- Segmentation: 1×1 convolutional layer over $h_{4,t}$
- Bounding box, class and stop prediction: three separate fully connected layers, over $[MP(h_{0,t}), MP(h_{1,t}), MP(h_{2,t}), MP(h_{3,t}), MP(h_{4,t})]$
  - MP - max-pooling operator

RSIS - Loss Function

$$\mathbf{L} = \mathbf{L}_m + \alpha \mathbf{L}_b + \beta \mathbf{L}_c + \gamma \mathbf{L}_s$$

- Segmentation Loss ($\mathbf{L}_m$)
- Classification Loss ($\mathbf{L}_c$)
- Detection Loss ($\mathbf{L}_b$)
- Stop Loss ($\mathbf{L}_s$)

- Loss terms are subsequently added as training progresses
- For large number of objects per image - curriculum learning
  - start by learning to predict two objects and increase the number of objects once the validation loss plateaus

RSIS - Loss Function
Segmentation Loss $\mathbf{L}_m$

- Predicted masks:

  $$\hat{y}_m = (\hat{y}_{m,1}, \hat{y}_{m,2}, ..., \hat{y}_{m,\hat{n}})$$

- Ground truth masks:

  $$y_m = (y_{m,1}, y_{m,2}, ..., y_{m,n})$$

- $\mathbf{L}_m(\hat{y}_m, y_m, \delta) = \sum_{t=1}^{\hat{n}} \sum_{t'=1}^{n} \mathrm{sIoU}(\hat{y}_{m,t}, y_{m,t'}) \delta_{t,t'}$

- $\mathrm{sIoU}(\hat{y}, y) = 1 - \frac{\langle \hat{\mathbf{y}}, \mathbf{y} \rangle}{\|\hat{\mathbf{y}}\|_1 + \|\mathbf{y}\|_1 + \langle \hat{\mathbf{y}}, \mathbf{y} \rangle}$
- δ - matrix of assignments - computed using the Hungarian algorithm and sIoU as cost

RSIS - Loss Function
Classification Loss $\mathbf{L}_c$

- Predicted class probabilities:

  $\hat{y}_c = (\hat{y}_{c,1}, \hat{y}_{c,2}, ..., \hat{y}_{c,\hat{n}})$
- Ground truth class probabilities:
  - $y_c = (y_{c,1}, y_{c,2}, ..., y_{c,n})$

- Categorical cross entropy, considering the pairs determined by δ

RSIS - Loss Function
Detection Loss $\mathbf{L}_b$

- Predicted bounding boxes:
  $$\hat{y}_b = (\hat{y}_{b,1}, \hat{y}_{b,2}, \ldots, \hat{y}_{b,\hat{n}})$$
- Ground truth bounding boxes:
  $$y_b = (y_{b,1}, y_{b,2}, \ldots, y_{b,n})$$

- Mean squared error between the box coordinates of matched pairs determined by δ

RSIS - Loss Function
Stop Loss $\mathbf{L}_s$

- Predicted objectness score:

  $\hat{y}_{s,t}$
- Ground truth:

  $\mathbb{1}_{t \leq n}$

- Binary cross entropy

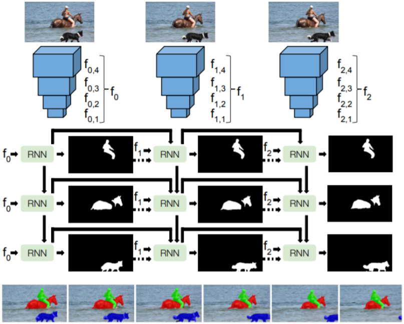
RVOS - related work

- RIS - "Recurrent Instance Segmentation" [6] - ECCV 2016
- RSIS - "Recurrent Neural Networks for Semantic Instance Segmentation" [8] - arXiv 2019
- **RVOS - adds recurrence in the temporal domain on top of RSIS**

Romera et al. [6], Salvador et al. [8]

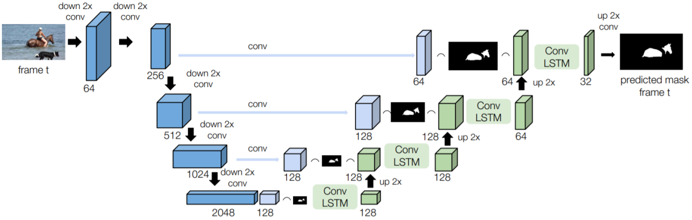
RVOS

- Recurrent model - spatial and temporal domains
- Trainable end-to-end and handles multiple objects in a unified manner
- Handles both Zero-shot and One-shot VOS tasks
- Good results

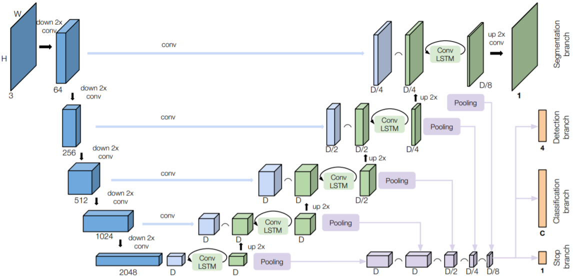


RVOS - Encoder-Decoder architecture

- **RVOS**



- **RSIS**



RVOS - Encoder

- Similar to RSIS
- ResNet-101 pretrained on ImageNet

- Input: x_t - RGB image
- Output: $f_t = f_{t,1}, f_{t,2}, ..., f_{t,k}$ - features at different resolutions

- Configurations:
  1. original RSIS architecture
  2. add the masks of the instances from the previous frame as one additional channel of the output features

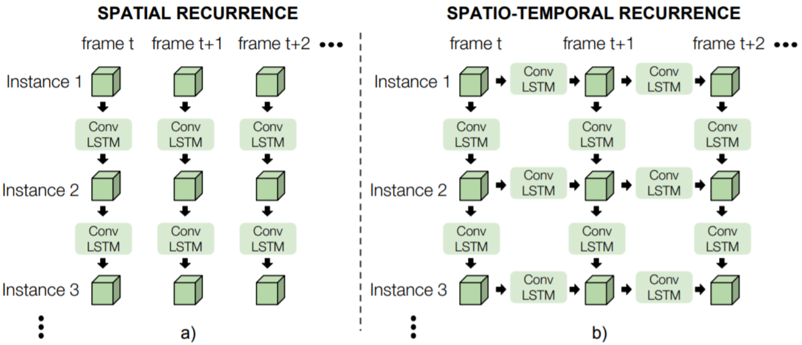
RVOS - Decoder (1)

- Hierarchical recurrent architecture of ConvLSTMs
- Output for frame t: $S_{t,1}, S_{t,2}, ..., S_{t,N}$
- Temporal recurrence - ensures an object has the same index

RVOS - Decoder (2)

- $h_{t,i,k}$ - output of k-th ConvLSTM layer for object i at frame t
- $h_{t,i,k} = \text{ConvLSTM}_k(h_{input}, h_{state})$
  - $h_{input} = [B_2(h_{t,i,k-1})|f'_{t,k}|S_{t-1,i}]$
  - $h_{state} = [h_{t,i-1,k}|h_{t-1,i,k}]$
- $h_{t-1,i,k}$ - temporal hidden state
- $h_{t,i-i,k}$ - spatial hidden state
- First ConvLSTM $\Rightarrow h_{input} = [f'_{t,0}|S_{t-1,i}]$
- First object $\Rightarrow h_{state} = [Z|h_{t-1,i,k}]$
- $S_{t-1,i}$ - used only for one-shot VOS

RVOS - Decoder (3)



RVOS - Experiments

- videos containing multiple objects
- 3-6 seconds per video

- YouTube-VOS [12]
  - training set 3471 videos
    - 65 unique object categories - seen categories
  - **validation set 474 videos**
    - 91 unique object categories - 26 unseen + 65 seen categories
- DAVIS-2017 [5]
  - training set 60 videos
  - validation set 30 videos
  - **test-dev set 30 videos**

Xu et al. [12], Pont-Tuset et al. [5]

RVOS - Runtime analysis

- RVOS
  - 44 ms/frame - GPU P100
  - 67 ms/frame - GPU K80
- OSMN - 150 ms/frame
- S2S - 160 ms/frame
- other methods using online learning are two order of magnitude slower

RVOS - Training details

- RGB images: 256 x 448
- batch: 4 clips of 5 consecutive frames
- 20 epochs using the previous ground truth mask
- 20 epochs using the previous inferred mask

RVOS - Experiments
One-shot VOS - YouTube-VOS

	YouTube-VOS one-shot			
	J_{seen}	J_{unseen}	F_{seen}	F_{unseen}
RVOS-Mask-S	54.7	37.3	57.4	42.4
RVOS-Mask-T	59.9	39.2	63.1	45.6
RVOS-Mask-ST	60.8	**44.6**	63.7	50.3
RVOS-Mask-ST+	**63.1**	44.5	**67.1**	**50.4**

Table 1. Ablation study about spatial and temporal recurrence in the decoder for one-shot VOS in YouTube-VOS dataset. Models have been trained using 80%-20% partition of the training set and evaluated on the validation set. + means that the model has been trained using the inferred masks.

RVOS - Experiments
One-shot VOS - YouTube-VOS

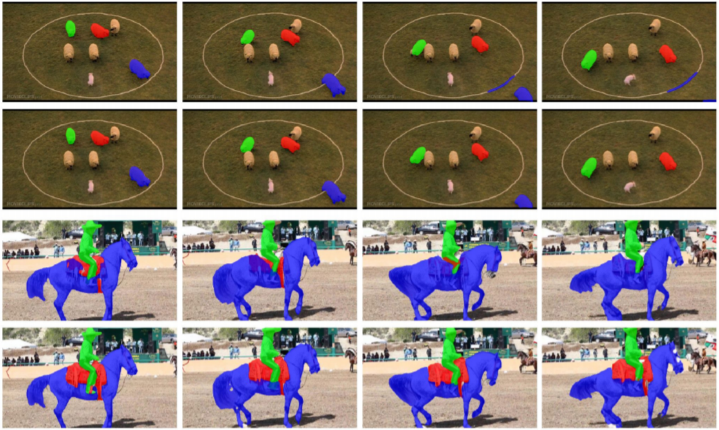


Figure 4. Qualitative results comparing spatial (rows 1,3) and spatio-temporal (rows 2,4) models.

RVOS - Experiments
One-shot VOS - YouTube-VOS

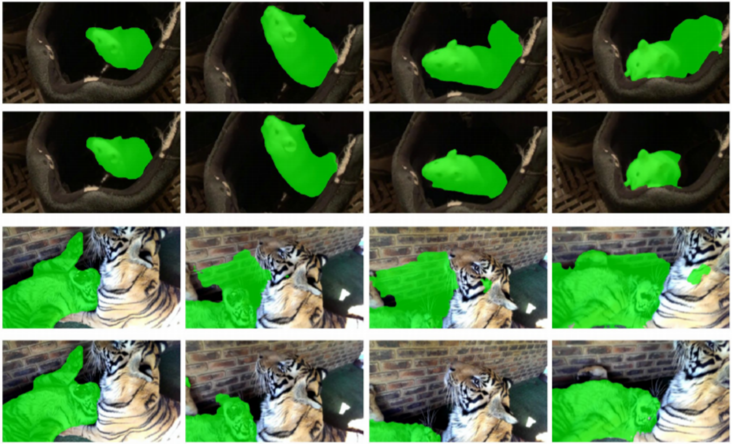


Figure 5. Qualitative results comparing training with ground truth masks (rows 1,3) and training with inferred masks (rows 2,4).

RVOS - Experiments
One-shot VOS - YouTube-VOS

		YouTube-VOS one-shot			
	OL	J_{seen}	J_{unseen}	F_{seen}	F_{unseen}
OSVOS [3]	✓	59.8	**54.2**	60.5	**60.7**
MaskTrack [20]	✓	59.9	45.0	59.5	47.9
OnAVOS [30]	✓	**60.1**	46.6	**62.7**	51.4
OSMN [34]	✗	60.0	40.6	60.1	44.0
S2S w/o OL [33]	✗	**66.7**	**48.2**	65.5	50.3
RVOS-Mask-ST+	✗	63.6	45.5	**67.2**	**51.0**

Table 2. Comparison against state of the art VOS techniques for one-shot VOS on YouTube-VOS validation set. OL refers to online learning. The table is split in two parts, depending on whether the techniques use online learning or not.

RVOS - Experiments
One-shot VOS - YouTube-VOS

	Number of instances (YouTube-VOS)				
	1	2	3	4	5
J mean	78.2	62.8	50.7	50.2	56.3
F mean	75.5	67.6	56.1	62.3	66.4

Table 3. Analysis of our proposed model RVOS-Mask-ST+ depending on the number of instances in one-shot VOS.

RVOS - Experiments
One-shot VOS - YouTube-VOS

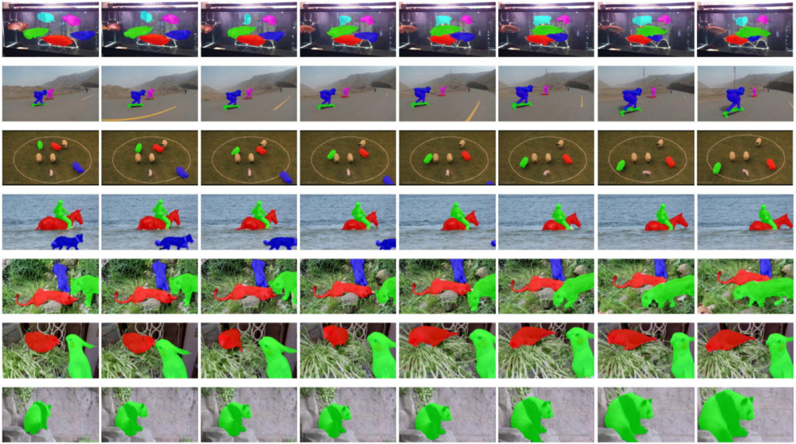


Figure 6. Qualitative results for one-shot video object segmentation on YouTube-VOS with multiple instances.

RVOS - Experiments
One-shot VOS - DAVIS-2017

	OL	DAVIS-2017 one-shot J	F
OSVOS [3]	✓	47.0	54.8
OnAVOS [30]	✓	49.9	55.7
OSVOS-S [17]	✓	52.9	62.1
CINM [2]	✓	**64.5**	**70.5**
OSMN [34]	✗	37.7	44.9
FAVOS [4]	✗	42.9	44.2
RVOS-Mask-ST+ (pre)	✗	46.4	50.6
RVOS-Mask-ST+ (ft)	✗	**48.0**	**52.6**

Table 4. Comparison against state of the art VOS techniques for one-shot VOS on DAVIS-2017 test-dev set. OL refers to online learning. The model RVOS-Mask-ST+(pre) is the one trained on Youtube-VOS, and the model RVOS-Mask-ST+ (ft) is after fine-tuning the model for DAVIS-2017. The table is split in two parts, depending on whether the techniques use online learning or not.

RVOS - Experiments
One-shot VOS - DAVIS-2017

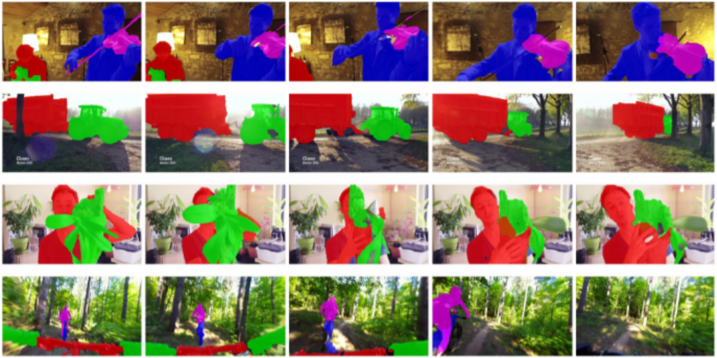


Figure 7. Qualitative results for one-shot on DAVIS-2017 test-dev.

RVOS - Experiments
Zero-shot VOS

- No dataset specially designed for this task

Figure 8. Missing object annotations may suppose a problem for zero-shot video object segmentation.

RVOS - Experiments
Zero-shot VOS

- Allow to segment up to 10 object instances, expecting that the up to 5 object instances are amond the predicted ones
- during training, each annotated object is uniquely assigned to one predicted object
- not-assigned predicted object do not contribute to loss function
- during testing, first frame annotation are used to compute correspondences between predictions and ground truth

Figure 8. Missing object annotations may suppose a problem for zero-shot video object segmentation.

RVOS - Experiments
Zero-shot VOS - YouTube-VOS

	YouTube-VOS zero-shot			
	J_{seen}	J_{unseen}	F_{seen}	F_{unseen}
RVOS-S	40.8	19.9	43.9	23.2
RVOS-T	37.1	20.2	38.7	21.6
RVOS-ST	**44.7**	**21.2**	**45.0**	**23.9**

Table 5. Ablation study about spatial and temporal recurrence in the decoder for zero-shot VOS in YouTube-VOS dataset. Our models have been trained using 80%-20% partition of the training set and evaluated on the validation set.

RVOS - Experiments
Zero-shot VOS - YouTube-VOS

Figure 9. Qualitative results for zero-shot video object segmentation on YouTube-VOS with multiple instances.

RVOS - Experiments
Zero-shot VOS - DAVIS-2017

- RVOS-Mask-ST+(pre)

  J 21.7
  F 27.3
- RVOS-Mask-ST+(ft)

  J 23.0
  F 29.9

- bad performance explainable in conjunction to bad performance for unseen objects in YouTube-VOS

RVOS - Experiments
Zero-shot VOS - DAVIS-2017

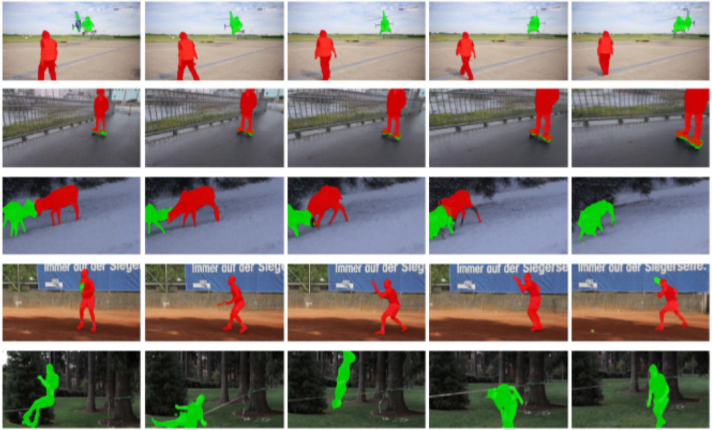


Figure 10. Qualitative results for zero-shot video object segmentation on DAVIS-2017 with multiple instances.

Thank you!

References I

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. 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, 2017.

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

References II

[4] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

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

[6] B. Romera-Paredes and P. H. S. Torr. Recurrent instance segmentation. In *European conference on computer vision*, pages 312–329. Springer, 2016.

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

References III

[8] A. Salvador, M. Bellver, V. Campos, M. Baradad, F. Marques, J. Torres, and X. Giro-i Nieto. Recurrent neural networks for semantic instance segmentation. *arXiv preprint arXiv:1712.00617*, 2017.

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

[10] C. Ventura, M. Bellver, A. Girbau, A. Salvador, F. Marques, and X. 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.

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

References IV

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