Video Object Segmentation

Haller Emanuela
ehaller@bitdefender.com

Table of contents

- Introduction to video object segmentation
- COSNet

  "See More, Know More: Unsupervised Video Object Segmentation with Co-Attention Siamese Networks" [8]
  CVPR 2019
- RVOS

  "RVOS: End-to-End Recurrent Network for Video Object Segmentation" [15]
  CVPR 2019
- A-GAME

  "A Generative Appearance Model for End-to-end Video Object Segmentation" [6]
  CVPR 2019

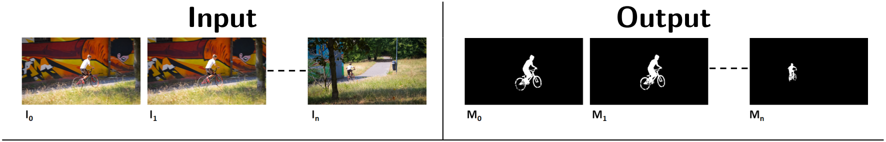
Lu et al. [8], Ventura et al. [15], Johnander et al. [6]

Video Object Segmentation - VOS

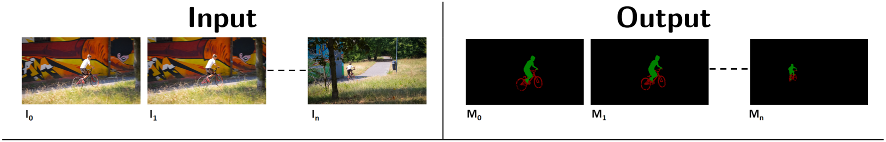
- Pixel-level binary masks for the object/objects of interest
- Level of supervision
  - Train:
    - Unsupervised VOS methods
    - Supervised VOS methods
  - Test:
    - Unsupervised VOS task
    - Semi-supervised VOS task
- Number of objects
  - Single Object VOS task
  - Multi Object VOS task

Unsupervised VOS task

- Single Object

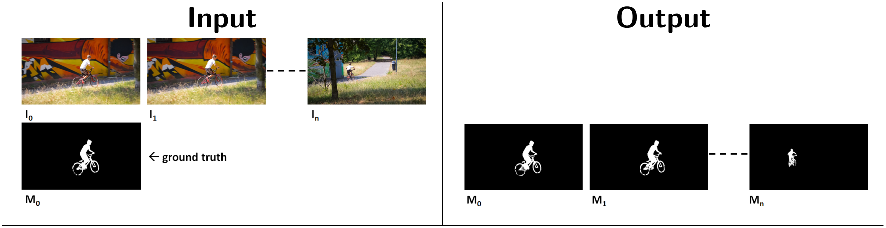


- Multi Object - Ill posed problem, with no special dataset



Semi-supervised VOS task

- Single Object



- Multi Object

VOS datasets

- DAVIS
  - 150 videos
  - Pixel level annotations
- YouTube-VOS
  - 4519 videos
  - Pixel level annotations
- SegTrack
  - 14 videos
  - Pixel level annotations
- YouTube-Objects
  - 2511 video shots, 720000 frames
  - Bounding box annotations
  - Subset of 126 videos with pixel level annotations
- FBMS
  - 59 videos
  - Pixel level annotations

See More, Know More: Unsupervised Video Object Segmentation with Co-Attention Siamese Networks

Xiankai Lu[1*], Wenguan Wang[1*], Chao Ma[2], Jianbing Shen[1†], Ling Shao[1], Fatih Porikli[3]

[1] Inception Institute of Artificial Intelligence, UAE

[2] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China

[3] Australian National University, Australia

carrierlxk@gmail.com wenguanwang.ai@gmail.com chaoma@sjtu.edu.cn

shenjianbingcg@gmail.com ling.shao@ieee.org fatih.porikli@anu.edu.au

https://github.com/carrierlxk/COSNet

COSNet

- Unsupervised VOS task
- Single Object: primary object
- Supervised method

COSNet: Intuition

- Primary objects
  - locally salient
    distinguishable in an individual frame
  - globally consistent
    frequently appearing throughout the video sequence

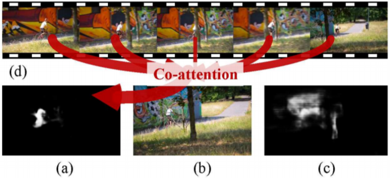


Figure 1. Illustration of our intuition. Given an input frame (b), our method leverages information from multiple reference frames (d) to better determine the foreground object (a), through a co-attention mechanism. (c) An inferior result without co-attention.

COSNet

- Segment the main object of a frame $\mathbf{F}_a$, exploiting consistencies with a set of frames $\{\mathbf{F}_{b_1}, \mathbf{F}_{b_2}, \ldots \mathbf{F}_{b_N}\}$.

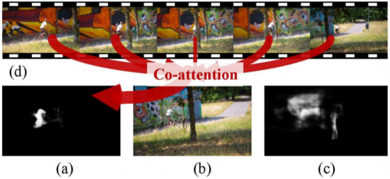
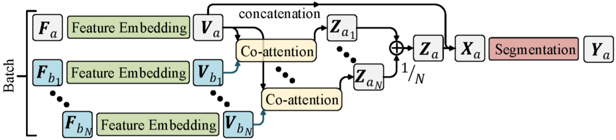


Figure 1. Illustration of our intuition. Given an input frame (b), our method leverages information from multiple reference frames (d) to better determine the foreground object (a), through a co-attention mechanism. (c) An inferior result without co-attention.

COSNet

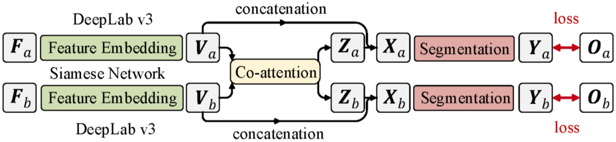
- Target frame: $\mathbf{F}_a$
- Reference frames: $\{\mathbf{F}_{b_1}, \ldots, \mathbf{F}_{b_N}\}$

- $\mathbf{X}_a$ defined by $\mathbf{F}_a, f(\mathbf{F}_{b_1}, \mathbf{F}_a), \ldots, f(\mathbf{F}_{b_N}, \mathbf{F}_a)$
  - features of frame $\mathbf{F}_a$
  - summary of $\{\mathbf{F}_{b_1}, \ldots, \mathbf{F}_{b_N}\}$ in light of $\mathbf{F}_a$



(b) Testing Phase

- Learn how to exploit consistencies, considering pairs of frames: $f(\mathbf{F}_a, \mathbf{F}_b)$



(a) Training Phase

COSNet: Architecture Training phase

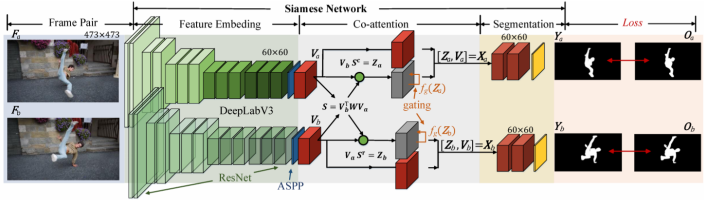


Figure 2. Overview of COSNet in the training phase. A pair of frames $\{\mathbf{F}_a, \mathbf{F}_b\}$ is fed into a feature embedding module to obtain the feature representations $\{\mathbf{V}_a, \mathbf{V}_b\}$. Then, the co-attention module computes the attention summaries that encode the correlations between $\mathbf{V}_a$ and $\mathbf{V}_b$. Finally, $\mathbf{Z}$ and $\mathbf{V}$ are concatenated and handed over to a segmentation module to produce segmentation predictions.

COSNet: Features Embedding Module

- Input: $\{\mathbf{F}_a, \mathbf{F}_b\} \in \mathbb{R}^{H' \times W' \times 3}$
- Output: $\{\mathbf{V}_a, \mathbf{V}_b\} \in \mathbb{R}^{H \times W \times C}$
- DeepLabv3 [2]

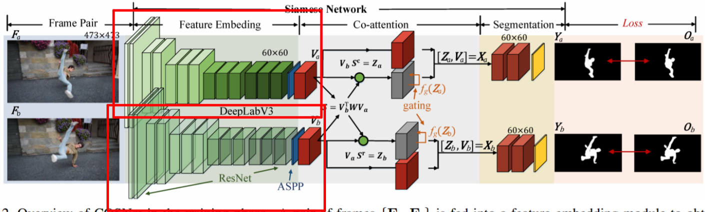


Figure 2. Overview of COSNet in the training phase. A pair of frames $\{\mathbf{F}_a, \mathbf{F}_b\}$ is fed into a feature embedding module to obtain the feature representations $\{\mathbf{V}_a, \mathbf{V}_b\}$. Then, the co-attention module computes the attention summaries that encode the correlations between $\mathbf{V}_a$ and $\mathbf{V}_b$. Finally, $\mathbf{Z}$ and $\mathbf{V}$ are concatenated and handed over to a segmentation module to produce segmentation predictions.

Chen et al. [2]

COSNet: Co-Attention Module

- Input: $\{\mathbf{V}_a, \mathbf{V}_b\} \in \mathbb{R}^{H \times W \times C}$
- Output: $\{\mathbf{X}_a, \mathbf{X}_b\} \in \mathbb{R}^{H \times W \times 2C}$
- $\mathbf{X}_a = [\mathbf{Z}_a, \mathbf{V}_a]$, $\mathbf{Z}_a$ - co-attention representation for frame $\mathbf{F}_a$

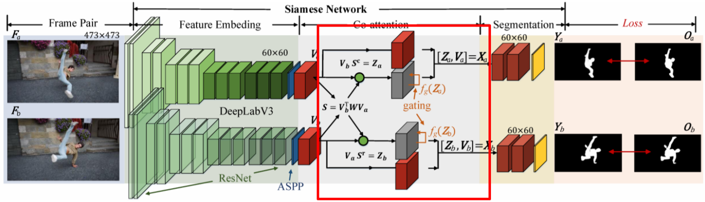


Figure 2. Overview of COSNet in the training phase. A pair of frames $\{\mathbf{F}_a, \mathbf{F}_b\}$ is fed into a feature embedding module to obtain the feature representations $\{\mathbf{V}_a, \mathbf{V}_b\}$. Then, the co-attention module computes the attention summaries that encode the correlations between $\mathbf{V}_a$ and $\mathbf{V}_b$. Finally, $\mathbf{Z}$ and $\mathbf{V}$ are concatenated and handed over to a segmentation module to produce segmentation predictions.

COSNet: Co-Attention Module

- $\mathbf{V}_a$, $\mathbf{V}_b$ - features of frames $\mathbf{F}_a$ and $\mathbf{F}_b$

  $\mathbf{V}_a, \mathbf{V}_b \in \mathbb{R}^{C \times WH}$

- $\mathbf{S} \in \mathbb{R}^{WH \times WH}$ - affinity matrix

  $\mathbf{S}_{i,j}$ - similarity between location i of $\mathbf{F}_b$ and location j in $\mathbf{F}_a$
- $\mathbf{S}^c, \mathbf{S}^r \in \mathbb{R}^{WH \times WH}$ - attention weights

  Normalize $\mathbf{S}$ row-wise and column-wise, using softmax

- $\mathbf{X}_a = [\mathbf{Z}_a, \mathbf{V}_a]$
- $\mathbf{Z}_a = f(\mathbf{F}_a, \mathbf{F}_b) \in \mathbb{R}^{C \times WH}$
  - i-th column of $\mathbf{Z}_a$ - weighted average of all columns of $\mathbf{V}_b$
  - weights defined by i-th column of $\mathbf{S}^c$

COSNet: Co-Attention Module

- Gated co-attention

  Decide how much information will be preserved

$$\mathbf{Z}_a = \mathbf{Z}_a * f_g(\mathbf{Z}_a)$$

$$f_g(\mathbf{Z}_a) = \sigma(\mathbf{w}_f \mathbf{Z}_a + b_f)) \in [0,1]^{WH}$$

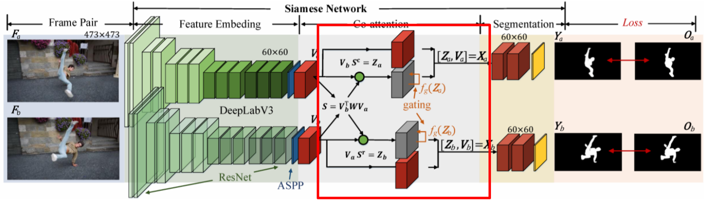


Figure 2. Overview of COSNet in the training phase. A pair of frames $\{\mathbf{F}_a, \mathbf{F}_b\}$ is fed into a feature embedding module to obtain the feature representations $\{\mathbf{V}_a, \mathbf{V}_b\}$. Then, the co-attention module computes the attention summaries that encode the correlations between $\mathbf{V}_a$ and $\mathbf{V}_b$. Finally, $\mathbf{Z}$ and $\mathbf{V}$ are concatenated and handed over to a segmentation module to produce segmentation predictions.

COSNet: Co-Attention Module
Definition of **S**

- Simple affinity matrix:
  $\mathbf{A} = \mathbf{V}_b^T \mathbf{V}_a \in \mathbb{R}^{(WH)\times(WH)}$
  $\mathbf{A}_{i,j}$ - similarity between location i of $\mathbf{F}_b$ and location j in $\mathbf{F}_a$
- Weighted affinity matrix:
  $\mathbf{S} = \mathbf{V}_b^T \mathbf{W} \mathbf{V}_a \in \mathbb{R}^{(WH)\times(WH)}$
  $\mathbf{W}^{C\times C}$ - weight matrix
  $\mathbf{S}_{i,j}$ - weighted similarity between location i of $\mathbf{F}_b$ and location j in $\mathbf{F}_a$
- Constraints on $\mathbf{W} \Rightarrow$ different co-attention mechanisms:
  - Vanilla co-attention
  - Symmetric co-attention
  - Channel-wise co-attention

COSNet: Co-Attention Module

- Vanilla co-attention - $\mathbf{W}$ diagonalizable matrix
  $$\mathbf{S} = \mathbf{V}_b^T \mathbf{W} \mathbf{V}_a = \mathbf{V}_b^T \mathbf{P}^{-1} \mathbf{D} \mathbf{P} \mathbf{V}_a$$
  Feature representation of each frame undergoes linear transformations.

- Symmetric co-attention - $\mathbf{W}$ symmetric matrix
  $$\mathbf{S} = \mathbf{V}_b^T \mathbf{W} \mathbf{V}_a = \mathbf{V}_b^T \mathbf{P}^T \mathbf{D} \mathbf{P} \mathbf{V}_a = (\mathbf{P}\mathbf{V}_\mathbf{b})^T \mathbf{D} (\mathbf{P}\mathbf{V}_a)$$
  Project $\mathbf{V}_a$ and $\mathbf{V}_b$ into an orthogonal common space - eliminate correlation between different channels.

- Channel-wise co-attention - $\mathbf{W}$ diagonal matrix
  $$\mathbf{S} = \mathbf{V}_b^T \mathbf{W} \mathbf{V}_a = \mathbf{V}_b^T \mathbf{D} \mathbf{V}_a = \mathbf{V}_b^T \mathbf{D}_a \mathbf{D}_b \mathbf{V}_a = (\mathbf{D}_a \mathbf{V}_b)^T (\mathbf{D}_b \mathbf{V}_a)$$
  Apply channel-wise weights - alleviate channel-wise redundancy.

COSNet: Co-Attention Module

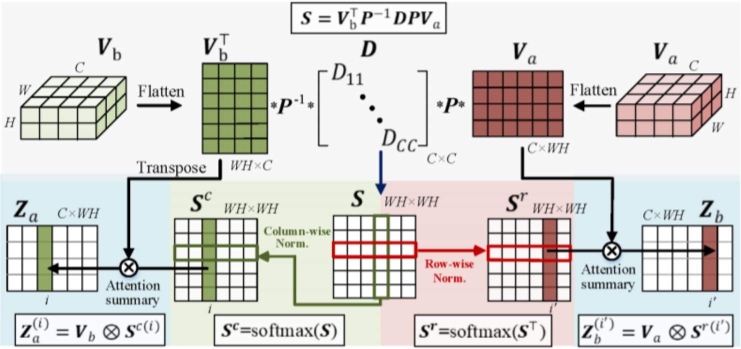


Figure 3. Illustration of our co-attention operation.

COSNet: Segmentation Module

- Input: $\{\mathbf{X}_a, \mathbf{X}_b\} \in \mathbb{R}^{H \times W \times 2C}$
- Output: $\{\mathbf{Y}_a, \mathbf{Y}_b\} \in \mathbb{R}^{H' \times W'}$
- Multiple convolutional layers

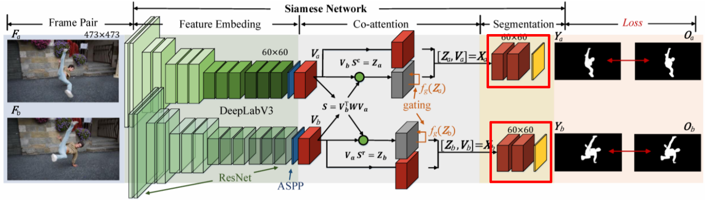


Figure 2. Overview of COSNet in the training phase. A pair of frames $\{\mathbf{F}_a, \mathbf{F}_b\}$ is fed into a feature embedding module to obtain the feature representations $\{\mathbf{V}_a, \mathbf{V}_b\}$. Then, the co-attention module computes the attention summaries that encode the correlations between $\mathbf{V}_a$ and $\mathbf{V}_b$. Finally, $\mathbf{Z}$ and $\mathbf{V}$ are concatenated and handed over to a segmentation module to produce segmentation predictions.

COSNet: Training

- Datasets:
  - Saliency datasets: MSRA10k [3] and DUT [19]
  - Video object segmentation: DAVIS2016 [9]
- Training procedure consists of two alternated steps:
  - Backbone trained for salient object segmentation
    - with an additional convolutional layer for generating segmentations
  - COSNet trained with video segmentation data: pairs of randomly selected video frames
- Weighted binary cross entropy loss

Cheng et al. [3], Yang et al. [19], Perazzi et al. [9]

COSNet: Testing

- Query frame $\mathbf{F}_a$
- Reference frame set $\{\mathbf{F}_{\mathbf{b_n}}\}_{n=1}^{N}$
- $\mathbf{Z}_a \leftarrow \frac{1}{N}\sum_{n=1}^{N} \mathbf{Z}_{a_n} * f_g(\mathbf{Z}_{a_n})$
- CRF refinement step

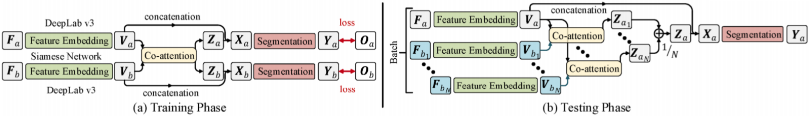


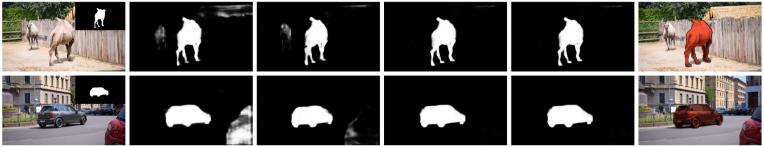
Figure 4. Schematic illustration of training pipeline (a) and testing pipeline (b) of COSNet.

COSNet: Ablation study

Network Variant	DAVIS		FBMS		Youtube-Objects	
	mean $\mathcal{J}$	$\Delta\mathcal{J}$	mean $\mathcal{J}$	$\Delta\mathcal{J}$	mean $\mathcal{J}$	$\Delta\mathcal{J}$
Co-attention Mechanism						
Vanilla co-attention (Eq. 3)	80.0	-0.5	75.2	-0.4	70.3	-0.2
Symmetric co-attention (Eq. 4)	80.5	-	75.6	-	70.5	-
Channel-wise co-attention (Eq. 5)	77.2	-3.3	72.7	-2.9	67.5	-3.0
w/o. Co-attention	71.3	-9.2	70.1	-5.5	62.9	-7.6
Fusion Strategy						
Attention summary fusion (Eq. 13)	80.5	-	75.6	-	70.5	-
Prediction segmentation fusion	79.5	-1.0	74.2	-1.4	69.9	-0.6
Frames Selection Strategy						
Global uniform sampling	80.53	-	75.61	-	70.54	-0.01
Global random sampling	80.52	-0.01	75.54	-0.02	70.55	-
Local consecutive sampling	80.26	-0.27	75.52	-0.09	70.43	-0.12

Table 1. Ablation study (§4.2) of COSNet on DAVIS16 [45], FBMS [41] and Youtube-Objects [47] datasets with different co-attention mechanisms, fusion strategies and sampling strategies.

COSNet: Ablation study



(a) Testing Frame with GT (b) $N = 0$ (c) $N = 1$ (d) $N = 2$ (e) $N = 5$ (f) Segmentation ($N = 5$)

Figure 5. Performance improvement for an increasing number of reference frames (§4.2). (a) Testing frames with ground-truths overlaid. (b)-(e) Primary object predictions with considering different number of reference frames ($N = 0, 1, 2,$ and 5). (f) Binary segments through applying CRF to (e). We can see that without co-attention, the COSNet degrades to a frame-by-frame segmentation model ((b): $N = 0$). Once co-attention is added ((c): $N = 1$), similar foreground distraction can be suppressed efficiently. Furthermore, more inference frames contribute to better segmentation performance ((c)-(e)).

Dataset	Number of reference frames (N)				
	0	1	2	5	7
DAVIS	71.3	77.6	79.7	80.5	80.5
FBMS	70.2	74.8	75.3	75.6	75.6
Youtube-Objects	62.9	67.7	70.5	70.5	70.5

Table 2. Comparisons with different numbers of reference frames during the testing stage on DAVIS16 [45], FBMS [41] and Youtube-Objects [47] datasets (§4.2). The mean $\mathcal{J}$ is adopted.

COSNet: Quantitative results DAVIS2016

	Method	TRC [17]	CVOS [51]	KEY [31]	MSG [40]	NLC [14]	CUT [9]	FST [42]	SFL [28]	LMP [52]	FSEG [24]	LVO [53]	ARP [30]	PDB [49]	**COSNet**
	Mean	47.3	48.2	49.8	53.3	55.1	55.2	55.8	67.4	70.0	70.7	75.9	76.2	77.2	**80.5**
$\mathcal{J}$	Recall	49.3	54.0	59.1	61.6	55.8	57.5	64.9	81.4	85.0	83.0	89.1	91.1	90.1	**94.0**
	Decay	8.3	10.5	14.1	2.4	12.6	2.2	**0.0**	6.2	1.3	1.5	**0.0**	7.0	0.9	**0.0**
	Mean	44.1	44.7	42.7	50.8	52.3	55.2	51.1	66.7	65.9	65.3	72.1	70.6	74.5	**79.4**
$\mathcal{F}$	Recall	43.6	52.6	37.5	60.0	61.0	51.9	51.6	77.1	79.2	73.8	83.4	83.5	84.4	**90.4**
	Decay	12.9	11.7	10.6	5.1	11.4	3.4	2.9	5.1	2.5	1.8	1.3	7.9	**-0.2**	0.0
$\mathcal{T}$	Mean	39.1	**25.0**	26.9	30.2	42.5	27.7	36.6	28.2	57.2	32.8	26.5	39.3	29.1	31.9

Table 3. Quantitative results on the test set of DAVIS16 [45][1] (see §4.3), using the region similarity $\mathcal{J}$, boundary accuracy $\mathcal{F}$ and time stability $\mathcal{T}$. We also report the recall and the decay performance over time for both $\mathcal{J}$ and $\mathcal{F}$. The best scores are marked in **bold**.

COSNet: Quantitative results FBMS

Method	NLC [14]	FST [42]	FSEG [24]	MSTP [21]	ARP [30]
Mean $\mathcal{J}$	44.5	55.5	68.4	60.8	59.8
Method	IET [32]	OBN [33]	PDB [49]	SFL [9]	**COSNet**
Mean $\mathcal{J}$	71.9	73.9	74.0	56.0	**75.6**

Table 4. Quantitative performance on the test sequences of FBMS [41] (§4.3) using region similarity (mean $\mathcal{J}$).

COSNet: Quantitative results YouTube-Objects

Method	FST [42]	COSEG [55]	ARP [30]	LVO [53]	PDB [49]	FSEG [24]	SFL [9]	COSNet
Airplane (6)	70.9	69.3	73.6	86.2	78.0	81.7	65.6	81.1
Bird (6)	70.6	76.0	56.1	81.0	80.0	63.8	65.4	75.7
Boat (15)	42.5	53.5	57.8	68.5	58.9	72.3	59.9	71.3
Car (7)	65.2	70.4	33.9	69.3	76.5	74.9	64.0	77.6
Cat (16)	52.1	66.8	30.5	58.8	63.0	68.4	58.9	66.5
Cow (20)	44.5	49.0	41.8	68.5	64.1	68.0	51.1	69.8
Dog (27)	65.3	47.5	36.8	61.7	70.1	69.4	54.1	76.8
Horse (14)	53.5	55.7	44.3	53.9	67.6	60.4	64.8	67.4
Motorbike (10)	44.2	39.5	48.9	60.8	58.3	62.7	52.6	67.7
Train (5)	29.6	53.4	39.2	66.3	35.2	62.2	34.0	46.8
Mean $\mathcal{J}$	53.8	58.1	46.2	67.5	65.4	68.4	57.0	**70.5**

Table 5. Quantitative performance of each category on Youtube-Objects [47] (§4.3) with the region similarity (mean $\mathcal{J}$). We show the average performance for each of the 10 categories from the dataset and the final row shows an average over all the videos.

COSNet: Qualitative results

Figure 6. Qualitative results on three datasets (§4.3). From top to bottom: *dance-twirl* from the DAVIS16 dataset [45], *horses05* from the FBMS dataset [41], and *bird0014* from the Youtube-Objects dataset [47].

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

Carles Ventura[1], Miriam Bellver[2], Andreu Girbau[3], Amaia Salvador[3],
Ferran Marques[3] and Xavier Giro-i-Nieto[3]

[1]Universitat Oberta de Catalunya [2]Barcelona Supercomputing Center
[3]Universitat Politècnica de Catalunya

cventuraroy@uoc.edu, miriam.bellver@bsc.es, {andreu.girbau, amaia.salvador, ferran.marques, xavier.giro}@upc.edu

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

- Unsupervised VOS task
  - Extension for semi-supervised VOS task
- Multi Object
- Supervised method

RVOS

- Recurrent model - spatial and temporal domains
- Handles multiple objects in a unified manner
- Suitable for both unsupervised and semi-supervised VOS tasks

RVOS: prior work

- RIS - "Recurrent Instance Segmentation" [12] - ECCV 2016
- RSIS - "Recurrent Neural Networks for Semantic Instance Segmentation" [14] - arXiv 2019
- RVOS - adds recurrence in the temporal domain on top of RSIS

Romera et al. [12], Salvador et al. [14]

RVOS: prior work

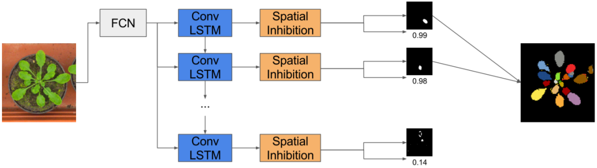
- **RIS - "Recurrent Instance Segmentation" [12] - ECCV 2016**
- RSIS - "Recurrent Neural Networks for Semantic Instance Segmentation" [14] - arXiv 2019
- RVOS - adds recurrence in the temporal domain on top of RSIS

Romera et al. [12], Salvador et al. [14]

Recurrent Instance Segmentation

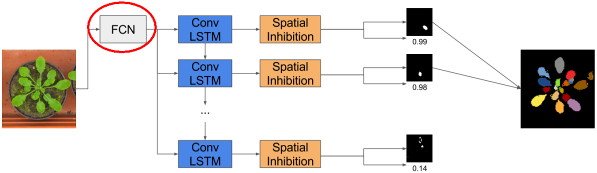
Bernardino Romera-Paredes
Philip Hilaire Sean Torr

Department of Engineering Science,
University of Oxford
bernard@robots.ox.ac.uk, philip.torr@eng.ox.ac.uk



- New instance segmentation paradigm: an end-to-end method that learns how to segment instances sequentially
- Input
  - image $\mathbf{I} \in \mathbb{R}^{h \times w \times 3}$
- 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 [7]



- $\mathbf{I} \in \mathbb{R}^{h \times w \times 3} \Rightarrow \mathbf{B} \in \mathbb{R}^{h' \times w' \times d}$

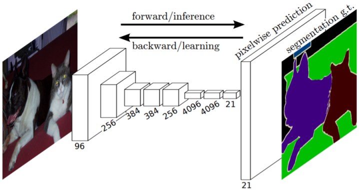
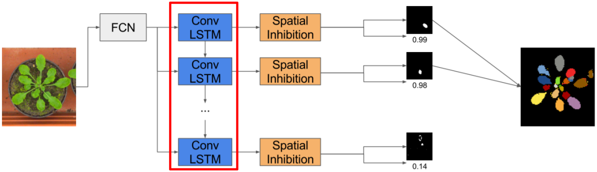
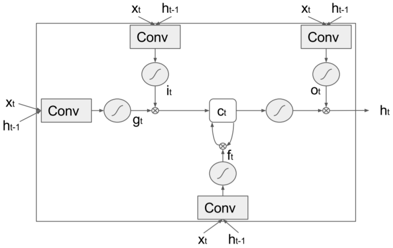


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

Long et al. [7] CVPR 2015

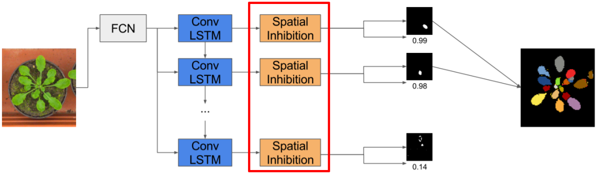
RIS: ConvLSTM [17]



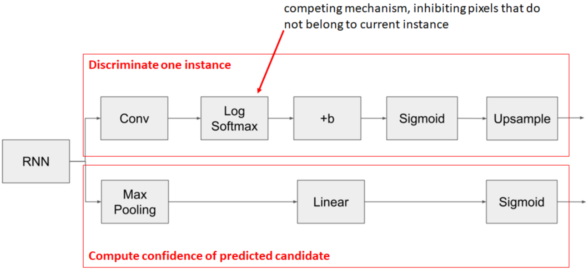


Shi et al. [17] NIPS 2015

RIS: Attention by Spatial Inhibition



- $r : \mathbb{R}^{h' \times w' \times d} \rightarrow [0,1]^{h \times w}, [0,1]$



- Train set:
  - $\mathbf{I}^{(i)} \in \mathbb{R}^{h \times w \times c}$
  - $\mathbf{Y}^{(i)} = \{\mathbf{Y}_1{}^{(i)}, \mathbf{Y}_2{}^{(i)}, ..., \mathbf{Y}_{n_i}{}^{(i)}\}$, $\mathbf{Y}_\mathbf{t}{}^{(i)} \in \{0,1\}^{h \times w}$
- 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: Training

- Match predictions to ground truth
  - $\delta \in \{0,1\}^{\widetilde{n} \times n}$
    - $\delta_{i,j}$ specifies if predicted mask i is associated to ground truth mask j
    - $\widetilde{n} = min(\hat{n}, n)$ - keep first predictions

    - Bipartite graph
    - Cost of edge between a predicted mask $\hat{\mathbf{Y}}_{\hat{t}}$ and a ground truth mask $\mathbf{Y}_t$:
      $f_{IoU}(\hat{\mathbf{Y}}_{\hat{t}}, \mathbf{Y}_t) = \frac{\langle \hat{\mathbf{Y}}_{\hat{t}}, \mathbf{Y}_t \rangle}{\left\|\hat{\mathbf{Y}}_{\hat{t}}\right\|_1 + \|\mathbf{Y}_t\|_1 + \langle \hat{\mathbf{Y}}_{\hat{t}}, \mathbf{Y}_t \rangle}$ - relaxed version of IoU
- Loss
  - High IoU according to δ
  - s_t should be 1 as long as $t \leq n$

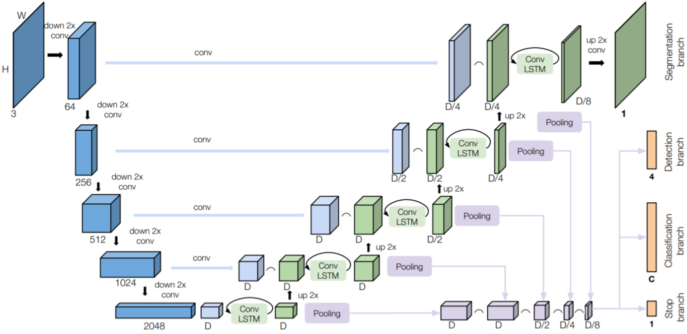
RVOS: prior work

- RIS - "Recurrent Instance Segmentation" [12] - ECCV 2016
- **RSIS - "Recurrent Neural Networks for Semantic Instance Segmentation" [14] - arXiv 2019**
- RVOS - adds recurrence in the temporal domain on top of RSIS

Romera et al. [12], Salvador et al. [14]

- Semantic instance segmentation
- Input:
  - image $x \in \mathbb{R}^{h \times w \times 3}$
- 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

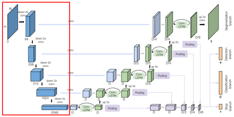
RSIS: Encoder-Decoder Architecture



RSIS: Encoder

- ResNet-101 [5], pretrained on ImageNet [13]

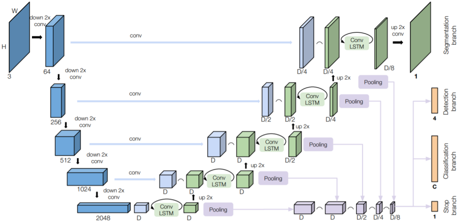
layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
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

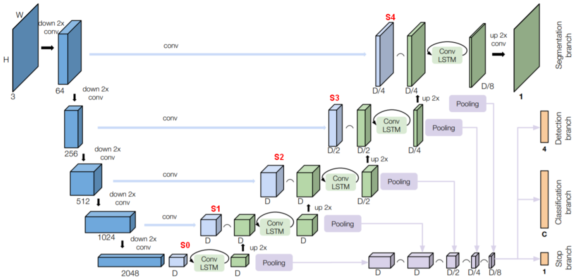


He et al. [5] - CVPR 2016, Russakovsky et al. [13] - IJCV 2015

RSIS: Decoder

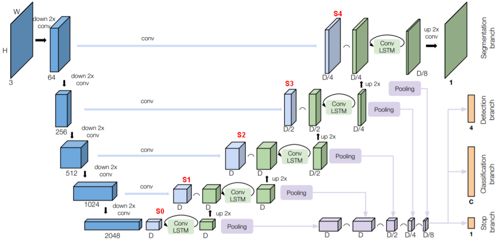
- Hierarchical recurrent architecture
- Upsampling network composed of a series of ConvLSTM layers
- Skip connections that bypass the previous recurrent layers
- Reliance on the features changes across different time steps





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

RSIS: Decoder



- 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 ($\mathbf{L}_s$)

- Considering δ

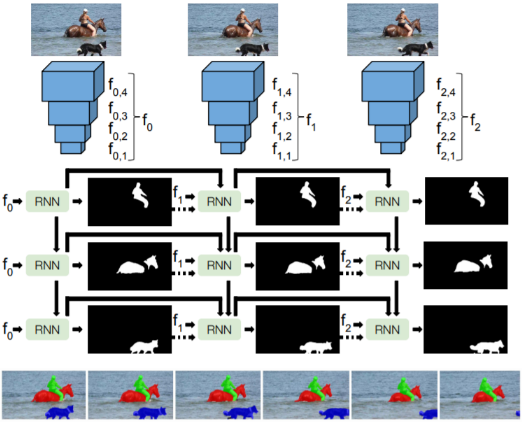
- Loss terms are subsequently added as training progresses
- For large number of objects per image - curriculum learning
  - start by learning to predict two objects and increase the number of objects once the validation loss plateaus

RVOS: related work

- RIS - "Recurrent Instance Segmentation" [12] - ECCV 2016
- RSIS - "Recurrent Neural Networks for Semantic Instance Segmentation" [14] - arXiv 2019
- **RVOS - adds recurrence in the temporal domain on top of RSIS**

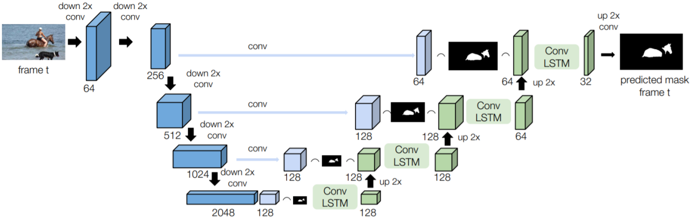
Romera et al. [12], Salvador et al. [14]

RVOS



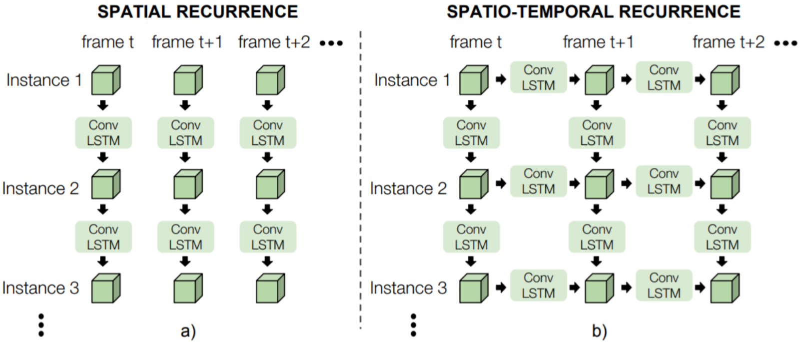
f0,4
f0,3
f0,2
f0,1
f0
f1,4
f1,3
f1,2
f1,1
f1
f2,4
f2,3
f2,2
f2,1
f2
f0
RNN
f1
RNN
f2
RNN
f0
RNN
f1
RNN
f2
RNN
f0
RNN
f1
RNN
f2
RNN

RVOS: Encoder-Decoder architecture



- Configurations:
  1. **Unsupervised VOS**
     original RSIS architecture
  2. **Semi-supervised VOS**
     add the mask of the instance from the previous frame as one additional channel of the output features

RVOS


SPATIAL RECURRENCE
frame t
frame t+1
frame t+2
Instance 1
Conv LSTM
Instance 2
Instance 3
a)
SPATIO-TEMPORAL RECURRENCE
frame t
frame t+1
frame t+2
Instance 1
Conv LSTM
Instance 2
Instance 3
b)

RVOS

- Encoder
  - ResNet-101, pretrained on ImageNet
- Decoder
  - Hierarchical recurrent architecture of ConvLSTMs
  - Temporal recurrence

RVOS: Decoder

- $h_{t,i,k}$ - 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 semi-supervised VOS

RVOS: Training

- 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
Semi-supervised 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
Semi-supervised VOS - YouTube-VOS

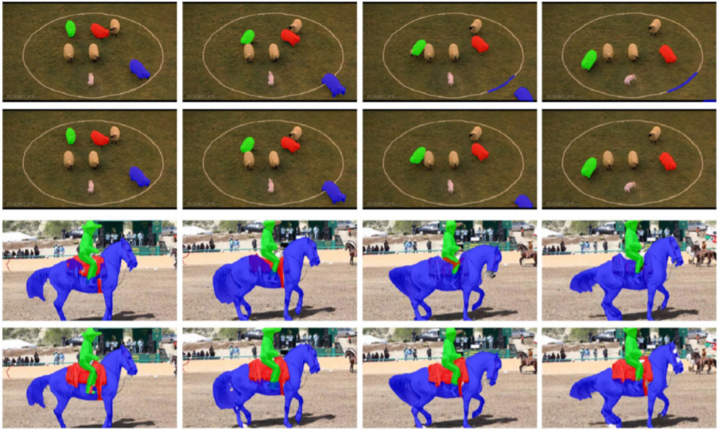


Figure 4. Qualitative results comparing spatial (rows 1,3) and spatio-temporal (rows 2,4) models.

RVOS: Experiments
Semi-supervised VOS - YouTube-VOS

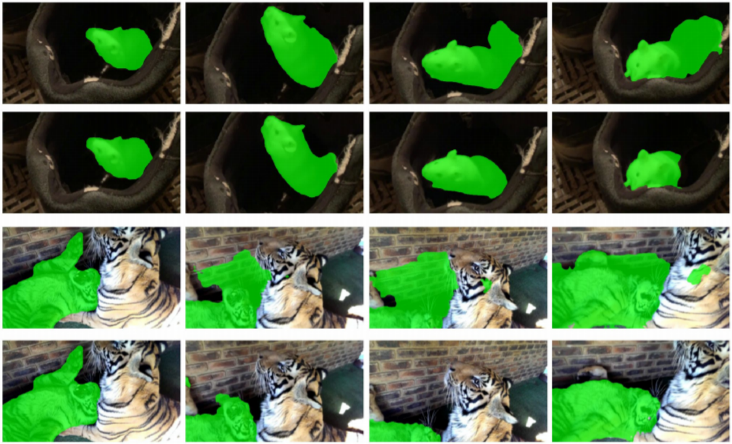


Figure 5. Qualitative results comparing training with ground truth masks (rows 1,3) and training with inferred masks (rows 2,4).

RVOS: Experiments
Semi-supervised VOS - YouTube-VOS

	OL	YouTube-VOS one-shot			
		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
Semi-supervised 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
Semi-supervised VOS - YouTube-VOS

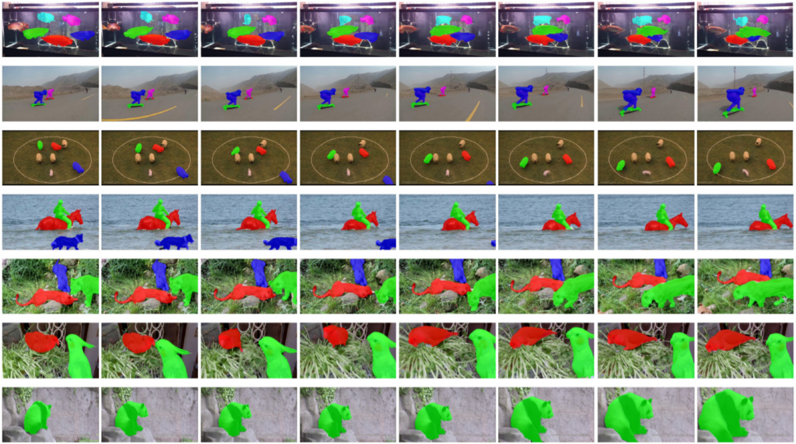


Figure 6. Qualitative results for one-shot video object segmentation on YouTube-VOS with multiple instances.

RVOS: Experiments
Semi-supervised VOS - DAVIS2017

		DAVIS-2017 one-shot	
	OL	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
Semi-supervised VOS - DAVIS-2017

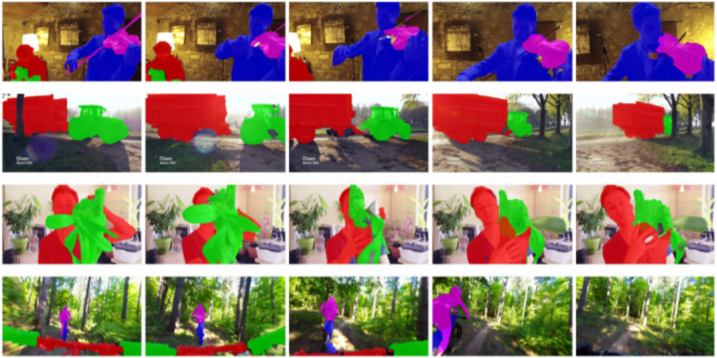


Figure 7. Qualitative results for one-shot on DAVIS-2017 test-dev.

RVOS: Experiments
Unsupervised VOS

- No dataset specially designed for this task
- Allow to segment up to 10 object instances, expecting the annotated ones to be among them
- 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
Unsupervised 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
Unsupervised VOS - YouTube-VOS

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

RVOS: Experiments
Unsupervised VOS - DAVIS2017

	J	**F**
RVOS-ST (pre)	21.7	27.3
RVOS-ST (ft)	23.0	29.9

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

RVOS: Experiments
Unsupervised VOS - DAVIS2017

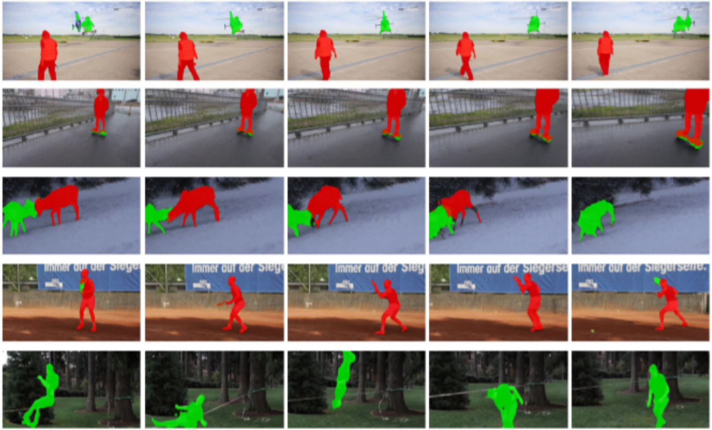


Figure 10. Qualitative results for zero-shot video object segmentation on DAVIS-2017 with multiple instances.

A Generative Appearance Model for End-to-end Video Object Segmentation

Joakim Johnander[1,3] Martin Danelljan[1,2] Emil Brissman[1,4] Fahad Shahbaz Khan[1,5]
Michael Felsberg[1]

[1] CVL, Linköping University, Sweden [2] CVL, ETH Zürich, Switzerland [3] Zenuity, Sweden [4] Saab, Sweden [5] IIAI, UAE

A-GAME

- Semi-supervised VOS task
- Single / Multi Object
- Supervised method

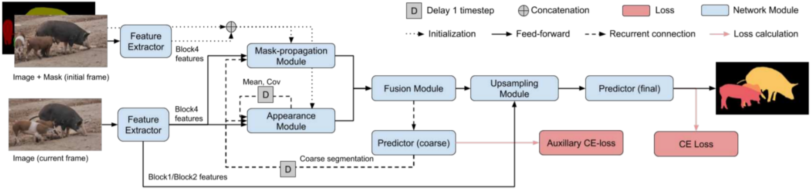
- Network learns in a one-shot manner to discriminate between target and background pixels, without invoking stochastic gradient descent.
  - Appearance model that learns a probabilistic generative model of target and background feature distributions.

A-GAME

- Semi-supervised VOS $\Rightarrow$ first frame annotations are used to compute the initial parameters.
- Parameters are updated online, based on predictions.

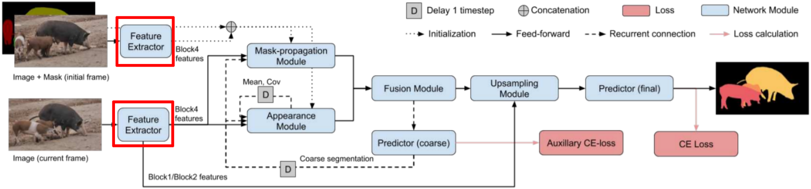
- For a given frame, the appearance model will define a coarse segmentation mask based on previous parameters.
- Further, the coarse mask is used to update model parameters.

A-GAME: Architecture



A-GAME: Backbone

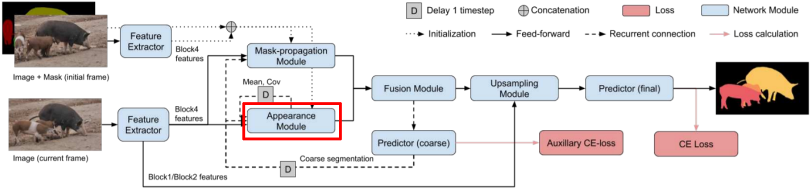
- ResNet101 [5],[1], pretrained on ImageNet [13]
- All network, except last block, is frozen
- Input: $\mathbf{I}^t \in \mathbb{R}^{h \times w \times 3}$ - frame t
- Output: $\{\mathbf{x}_1^t, \mathbf{x}_2^t, \ldots, \mathbf{x}_m^t\}$, $m = hw$ - nr pixels in image, $\mathbf{x}_i^t \in \mathbb{R}^{Dx1}$



He et al. [5], Chen et al. [1], Russakovsky et al. [13]

A-GAME: Appearance Module

- Input:
  - $\{\mathbf{x}_1^t, \mathbf{x}_2^t, \ldots, \mathbf{x}_m^t\}$
  - θ^{t-1} - previous frame parameters of the appearance model -
  - $\tilde{y}_p^t$ - coarse segmentation
- Output:
  - $s_{p,k}^t$ - score for component k at location p, in frame t -
  - θ^t



A-GAME: Appearance Module

- K components
- Each such component exclusively models the feature vectors of either foreground or background
- 4 Gaussians:
  $k \in \{0,2\}$ - background
  $k \in \{1,3\}$ - foreground
  - 0 & 1 - base components
  - 2 & 3 - distractors

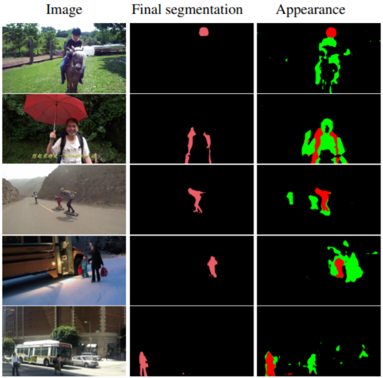


Figure 3. Visualization of the appearance module on five videos from YouTube-VOS. The final segmentation of our approach is shown (middle) together with output of the appearance module (right). The appearance module accurately locates the target (red) with the foreground representation while accentuating potential distractors (green) with the secondary mixture component.

A-GAME: Appearance Module

- Model output:

$$p(z_p^t = k|\mathbf{x}_p^t, \theta^{t-1}) = \frac{p(z_p^t = k)p(\mathbf{x}_p^t|z_p^t = k)}{\sum_i p(z_p^t = i)p(\mathbf{x}_p^t|z_p^t = i)}$$

  - In practice, log-probabilities are fed to the fusion module
  - $s_{p,k}^t \approx log(p(z_p^t = k)p(x_p^t|z_p^t = k))$
- z_p discrete random variable assigning observation $\mathbf{x}_p$ to a specific component
- Uniform prior: $p(z_p = k) = \frac{1}{K}$
- $p(\mathbf{x}_p) = \sum_{k=1}^{K} p(z_p = k)p(\mathbf{x}_p|z_p = k)$
- $p(\mathbf{x}_p|z_p = k) = \mathcal{N}(\mathbf{x}_p|\mu_k, \mathbf{\Sigma}_k)$

A-GAME: Appearance Module

- First frame:
  - Initial parameters are inferred from the extracted features and initial target mask
- Subsequent frames:
  - Update the model using soft component assignment variables $\alpha^t_{p,k} \in [0,1]$ ($\alpha^0_{p,k} \in \{0,1\}$)

- Model parameters updates

$$\widetilde{\mu}_k^t = \frac{\sum_p \alpha_{p,k}^t \mathbf{x}_p^t}{\sum_p \alpha_{p,k}^t}$$

$$\widetilde{\boldsymbol{\Sigma}}_k^t = \frac{\sum_p \alpha_{p,k}^t \mathrm{diag}\{(\mathbf{x}_p^t - \widetilde{\mu}_k^t)^2 + \mathbf{r}_k\}}{\sum_p \alpha_{p,k}^t}$$

$\mathbf{r}_k$ - trainable

- Model update

$$\mu_k^0 = \widetilde{\mu}_k^0$$

$$\boldsymbol{\Sigma}_k^0 = \widetilde{\boldsymbol{\Sigma}}_k^0$$

-

$$\mu_k^t = (1-\lambda)\mu_k^{t-1} + \lambda \widetilde{\mu}_k^t$$

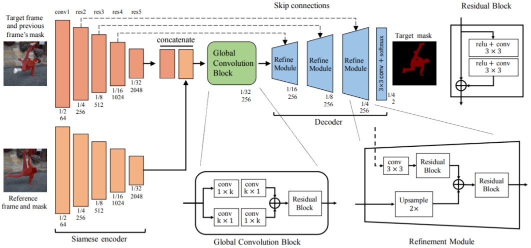
$$\boldsymbol{\Sigma}_k^t = (1-\lambda)\boldsymbol{\Sigma}_k^{t-1} + \lambda \widetilde{\boldsymbol{\Sigma}}_k^t$$

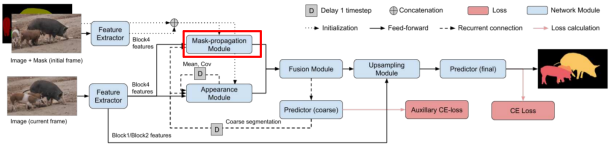
A-GAME: Appearance Module

- Base components:
  - First frame ($y_p \in \{0,1\}$):
    $$\alpha^0_{p,0} = 1 - y_p$$
    $$\alpha^0_{p,1} = y_p$$
  - Subsequent frames:
    $$\alpha^t_{p,0} = 1 - \widetilde{y}_p(\mathbf{I}^t, \theta^{t-1}, \Phi)$$
    $$\alpha^t_{p,1} = \widetilde{y}_p(\mathbf{I}^t, \theta^{t-1}, \Phi)$$
    Φ - network parameters
- Additional components
  $$\alpha^t_{p,2} = max(0, \alpha^t_{p,0} - p(z^t_p = 0 | \mathbf{x}^t_p, \mu^t_0, \mathbf{\Sigma}^t_0))$$
  $$\alpha^t_{p,3} = max(0, \alpha^t_{p,1} - p(z^t_p = 1 | \mathbf{x}^t_p, \mu^t_1, \mathbf{\Sigma}^t_1))$$

  Posteriors evaluated using only the base components

A-GAME: Mask-Propagation Module [16]



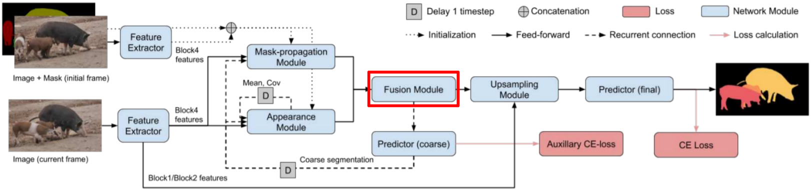


- Three convolutional layers

Wug et al. [16]

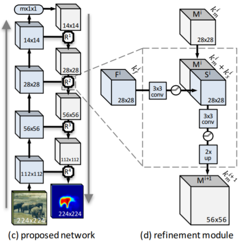
A-GAME: Fusion Module

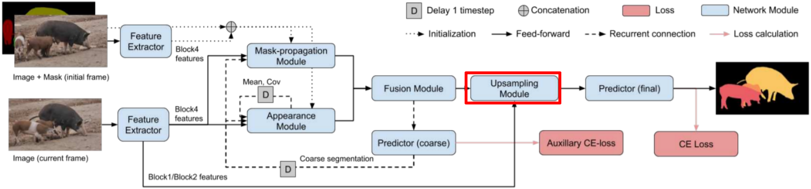
- Concatenate results of Appearance and Mask-Propagation Modules
- 2 convolutional layers



A-GAME: Upsampling Module

- Predicts a soft-segmentation mask $\hat{y}_p$
- Coarse representation is successively combined with successively shallower features [10]

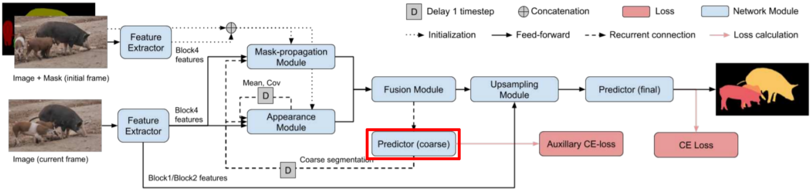




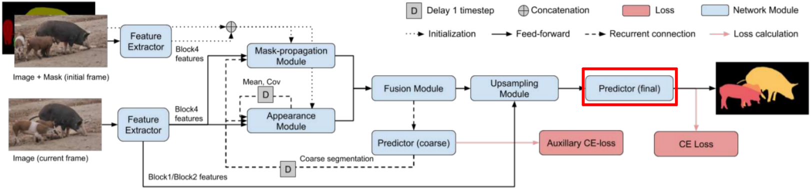
Pinheiro et al. [10]

A-GAME: Predictor Coarse

- Generates a coarse soft-segmentation mask $\widetilde{y}_p$
- Will be used by the Appearance and Mask-Propagation Modules



A-GAME: Predictor Final



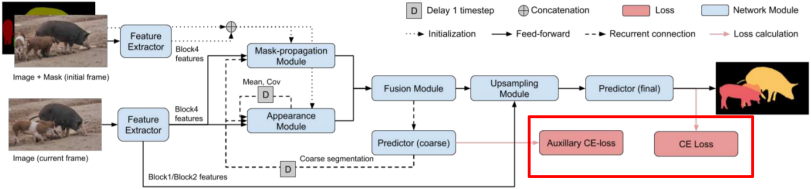
A-GAME: Multi-Object

- Run the model once per object
- Combine resulting soft-segmentations with softmax-aggregation [16]
- Aggregated soft-segmentations will replace $\widetilde{y}_p$ in the recurrent connection

Wug et al. [16]

A-GAME: Training

- Training sample: one video, with n frames, along with the annotation for the first frame
- Cross-entropy loss on the final mask
- Auxiliary loss for coarse segmentation $\widetilde{y}_p$



A-GAME: Training

- Datasets:
  - DAVIS2017 [11]
  - YouTube-VOS [18]
  - SynthVOS
    - Add 1-5 objects from MSRA10k [3] (salient objects) into images from VOC2012 [4]
    - Move objects across the image $\Rightarrow$ synthetic video

Pont-Tuset et al. [11], Xu et al. [18], Cheng et al. [3], Everingham et al. [4]

A-GAME: Training

- Initial training
  - 80 epochs
  - All 3 datasets
  - Half resolution images
  - Batch: 4 sequences of 8 frames
- Finetuning
  - 100 epochs
  - DAVIS2017 & YouTube-VOS
  - Full resolution images
  - Batch: 2 sequences of 14 frames

A-GAME: Ablation study

Version	$\mathcal{G}$	$\mathcal{J}$ seen (%)	$\mathcal{J}$ unseen (%)
A-GAME	66.0	66.9	61.2
No appearance module	50.0	57.8	40.6
No mask-prop module	64.0	65.5	59.5
Unimodal appearance	64.4	65.8	58.8
No update	64.9	66.0	59.8
Appearance SoftMax	55.8	59.3	50.7
No end-to-end	58.8	62.5	53.1

Table 1. Ablation study on YouTube-VOS. We report the overall performance $\mathcal{G}$ along with segmentation accuracy $\mathcal{J}$ on classes seen and unseen during training. See text for further details.

A-GAME: Quantitative results YouTube-VOS

Method	O-Ft	$\mathcal{G}$ overall (%)	$\mathcal{J}$ seen (%)	$\mathcal{J}$ unseen (%)
S2S [33]	✓	64.4	71.0	55.5
OSVOS [2]	✓	58.8	59.8	54.2
OnAVOS [30]	✓	55.2	60.1	46.6
MSK [23]	✓	53.1	59.9	45.0
OSMN [34]	×	51.2	60.0	40.6
S2S [33]	×	57.6	66.7	48.2
RGMP [31]	×	53.8	59.5	45.2
RGMP† [31]	×	50.5	54.1	41.7
A-GAME	×	66.0	66.9	61.2
A-GAME†	×	66.1	67.8	60.8

Table 2. State-of-the-art comparison on the YouTubeVOS benchmark. Our approach obtains the best overall performance ($\mathcal{G}$) despite not performing any online fine-tuning (O-Ft). Further, our approach provides a large gain in performance for categories unseen during training ($\mathcal{J}$ unseen), compared to existing methods. Entries marked by † were trained with only YouTube-VOS data.

A-GAME: Quantitative results DAVIS2017

Method	O-Ft	Causal	$\mathcal{J}\&\mathcal{F}$ mean (%)	$\mathcal{F}$ (%)	$\mathcal{J}$ (%)
CINM [1]	✓	✓	70.6	74.0	67.2
OSVOS-S [21]	✓	✓	68.0	71.3	64.7
OnAVOS [30]	✓	✓	65.4	69.1	61.6
OSVOS [2]	✓	✓	60.3	63.9	56.6
DyeNet [18]	×	×	69.1	71.0	67.3
RGMP [31]	×	✓	66.7	68.6	64.8
VM [13]	×	✓	-	-	56.5
FAVOS [5]	×	✓	58.2	61.8	54.6
OSMN [34]	×	✓	54.8	57.1	52.5
A-GAME	×	✓	70.0	72.7	67.2

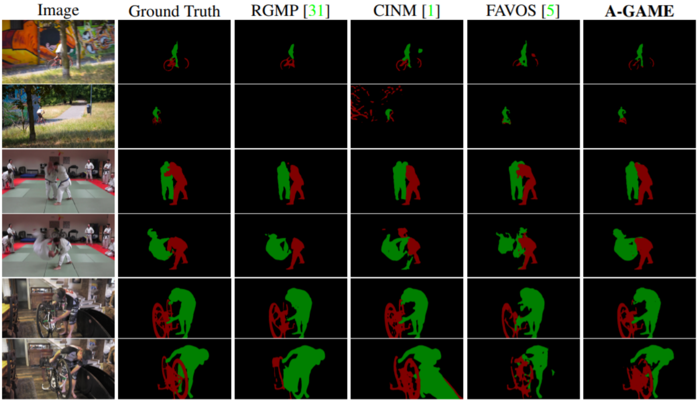
Table 3. State-of-the-art comparison on the DAVIS2017 validation set. For each method we report whether it employs online fine-tuning (O-Ft), is causal, and the final performance $\mathcal{J}$ (%). Our approach obtains superior results compared to state-of-the-art methods without online fine-tuning. Further, our approach closes the performance gap to existing methods employing online fine-tuning.

A-GAME: Quantitative results DAVIS2016

Method	O-Ft	Causal	Speed	$\mathcal{J}\&\mathcal{F}$ mean (%)	$\mathcal{F}$ (%)	$\mathcal{J}$ (%)
OnAVOS [30]	✓	✓	13s	85.5	84.9	86.1
OSVOS-S [21]	✓	✓	4.5s	86.6	87.5	85.6
MGCRN [12]	✓	✓	0.73s	85.1	85.7	84.4
CINM [1]	✓	✓	>30s	84.2	85.0	83.4
LSE [8]	✓	✓		81.5	80.1	82.9
OSVOS [2]	✓	✓	9s	80.2	80.6	79.8
MSK [23]	✓	✓	12s	77.6	75.4	79.7
SFL [6]	✓	✓	7.9s	75.4	76.0	74.8
DyeNet [18]	×	×	0.42s	-	-	84.7
FAVOS [5]	×	✓	1.80s	81.0	79.5	82.4
RGMP [31]	×	✓	0.13s	81.8	82.0	81.5
VM [13]	×	✓	0.32s	-	-	81.0
MGCRN [12]	×	✓	0.36s	76.5	76.6	76.4
PML [4]	×	✓	0.28s	81.2	79.3	75.5
OSMN [34]	×	✓	0.14s	73.5	72.9	74.0
CTN [15]	×	✓	1.30s	71.4	69.3	73.5
VPN [14]	×	✓	0.63s	67.9	65.5	70.2
MSK [23]	×	✓	0.15s	-	-	69.9
A-GAME	×	✓	0.07s	82.1	82.2	82.0

Table 4. State-of-the-art comparison on DAVIS2016 validation set, which is a subset of DAVIS2017. For each method we report whether it employs online fine-tuning (O-Ft), is causal, the computation time (if available), and the final performance $\mathcal{J}$ (%). Our approach obtains competitive results compared to causal methods without online fine-tuning.

A-GAME: Qualitative results



Thank you!

References I

[1] 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.

[2] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[3] M. Cheng. Msra10k database, 2015.

[4] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

References II

[5] 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.

[6] J. Johnander, M. Danelljan, E. Brissman, F. S. Khan, and M. Felsberg. A generative appearance model for end-to-end video object segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[7] 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.

[8] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3623–3632, 2019.

[9] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016.

[10] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *European Conference on Computer Vision*, pages 75–91. Springer, 2016.

References IV

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] S. Wug Oh, J.-Y. Lee, K. Sunkavalli, and S. 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.

[17] 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.

[18] 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.

[19] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3166–3173, 2013.