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

Haller Emanuela
ehaller@bitdefender.com

1 October 2019

Table of contents

- "A Generative Appearance Model for End-to-end Video Object Segmentation" [6] - CVPR 2019
- "See More, Know More: Unsupervised Video Object Segmentation with Co-Attention Siamese Networks" [7] - CVPR 2019

Johnander et al. [6], Lu et al. [7]

A-GAME

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]

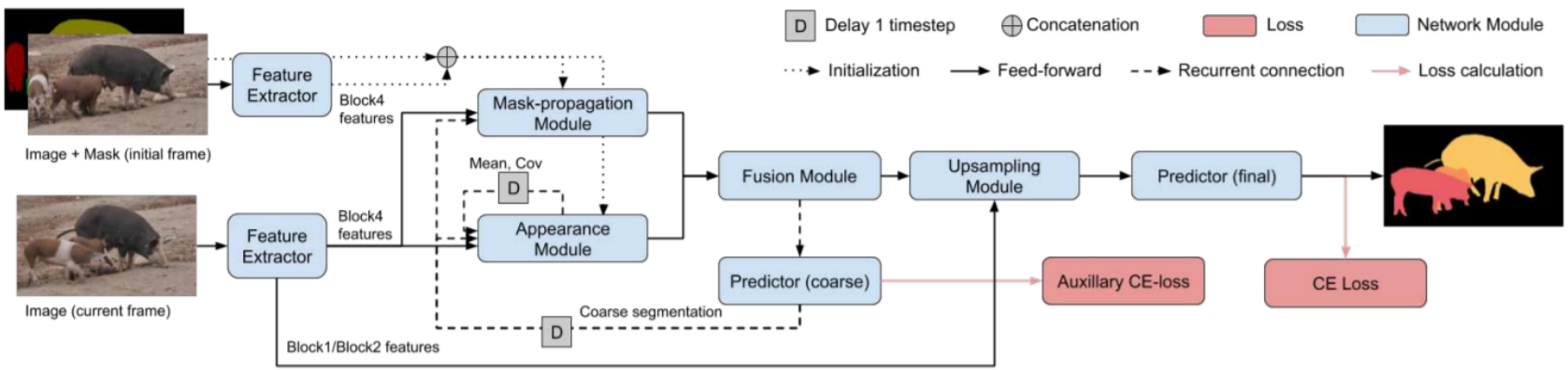
[1] CVL, Linköping University, Sweden [2] CVL, ETH Zürich, Switzerland [3] Zenuity, Sweden [4] Saab, Sweden [5] IIAI, UAE

A-GAME

- Supervised method
- Semi-Supervised video object segmentation task
- Single/Multi object

- Network learns in a one-shot manner to discriminate between target and background pixels, without invoking stochastic gradient descent

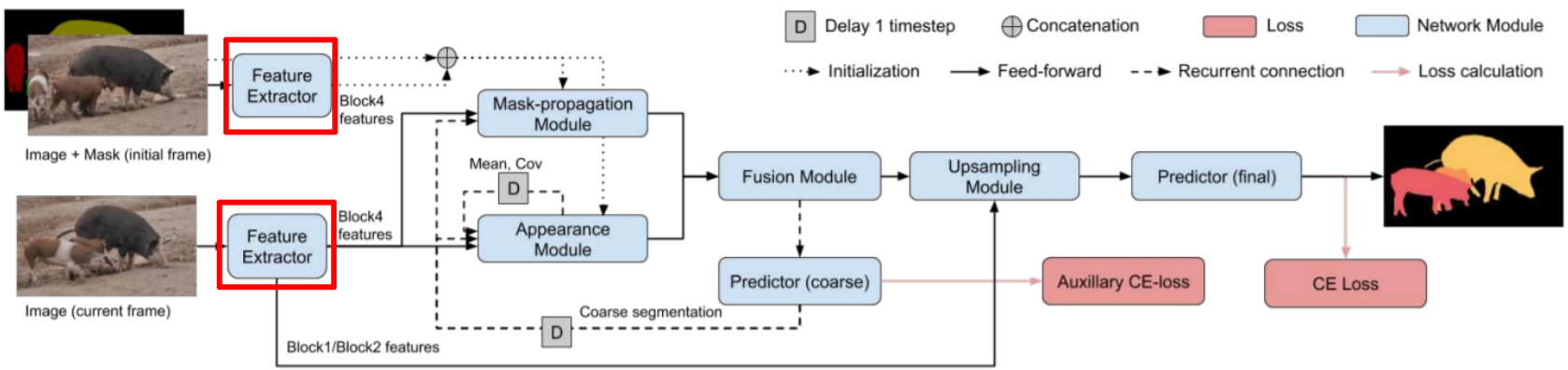
A-GAME: Architecture



A-GAME: Backbone

- ResNet101 [5] with dilated convolutions [1]
- Pretrained on ImageNet [11]
- All network, except last block, is frozen

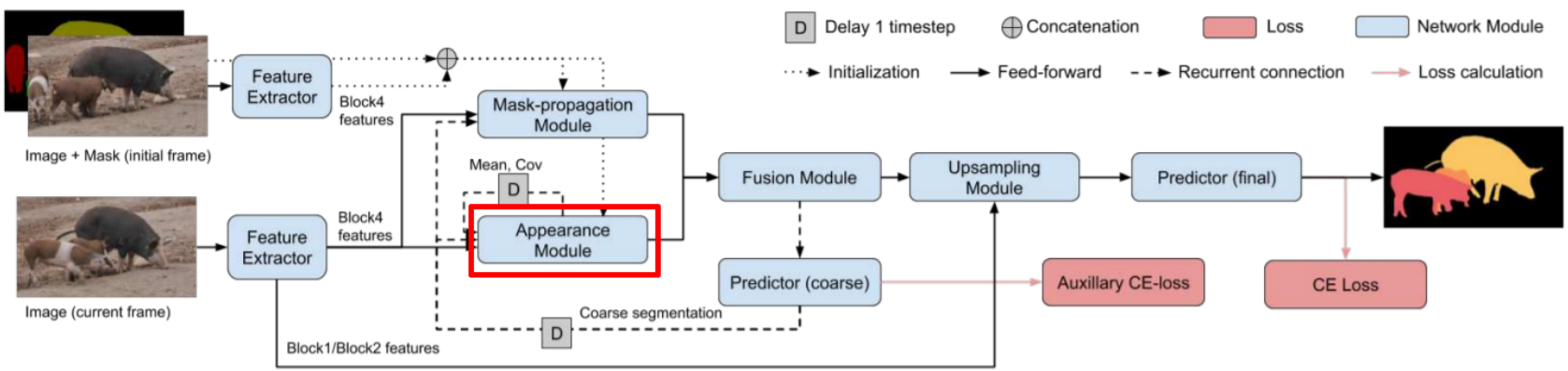
- Input: $\mathbf{I}^t \in \mathbb{R}^{h \times w \times 3}$ - frame i
- Output: $\{\mathbf{x}_1^t, \mathbf{x}_2^t, ..., \mathbf{x}_m^t\}$, $m = hw$ - nr pixels in image, $\mathbf{x}_i^t \in \mathbb{R}^{D \times 1}$



He et al. [5], Chen et al. [1], Russakovsky et al. [11]

A-GAME: Appearance Module

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



A-GAME: Appearance Module
Model learning

- Assume we have 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

- 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

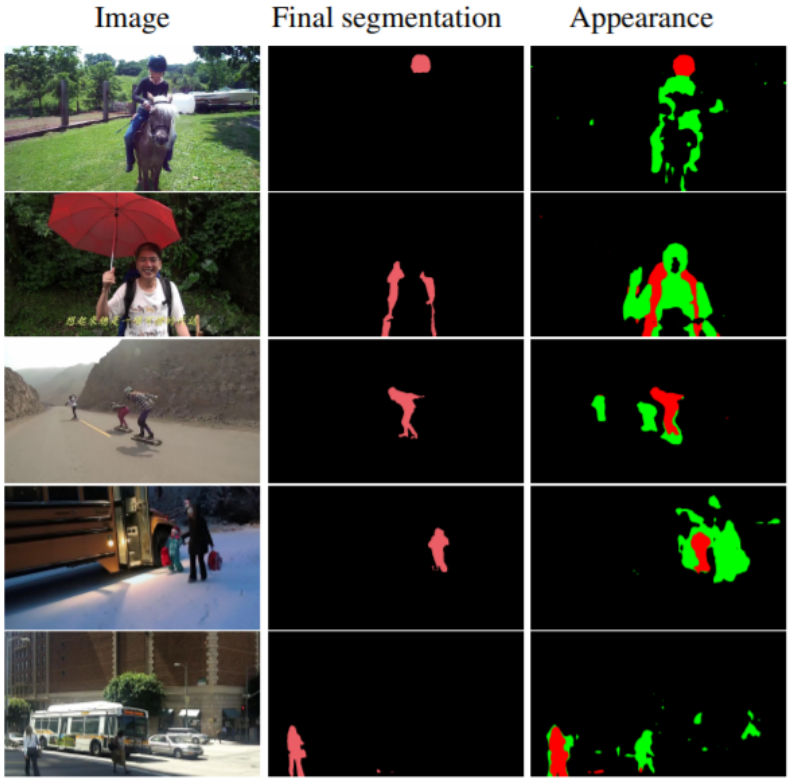


Figure 3. Visualization of the appearance module on five videos from YouTube-VOS. The final segmentation of our approach is shown (middle) together with output of the appearance module (right). The appearance module accurately locates the target (red) with the foreground representation while accentuating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

A-GAME: Appearance Module

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

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
  - $s_{p,k}^t \approx log(p(z_p^t = k)p(x_p^t|z_p^t = k))$
  - $s_{p,k}^t = -\frac{ln|\Sigma_k^{t-1}|+(\mathbf{x}_p^t-\mu_k^{t-1})^T(\Sigma_k^{t-1})^{-1}(\mathbf{x}_p^t-\mu_k^{t-1})}{2}$

A-GAME: Appearance Module

- Model parameters updates

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

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

- Model update

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

-

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

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: Appearance Module

Algorithm 1: The appearance module inference and update. **Inference:** Based on the appearance model parameters, $\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i$, and the input feature map $\mathbf{x}_p^i$, a soft segmentation is constructed for the background, foreground, and the two residual components. **Update:** The appearance model parameters are updated based on the coarse segmentation $\tilde{y}_p^i$.

`Inference` $(\mathbf{x}_p^i, \boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i)$ **:**

  for $k = 0, 1, 2, 3$: compute s_{pk}^i from (8)

  return s_{pk}^i

`Update` $(\mathbf{x}_p^i, \tilde{y}_p^i, \boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i)$ **:**

  for $k = 0, 1$: compute $\alpha_{p,k}^i$ from (5)

  for $k = 0, 1$: compute $\tilde{\boldsymbol{\mu}}_k^i, \tilde{\boldsymbol{\Sigma}}_k^i$ based on (3)

  for $k = 0, 1$: compute s_{pk}^i based on (8)

  for $k = 0, 1$: compute
  $p(z_p^i = k | \mathbf{x}_p^i, \boldsymbol{\mu}_0^i, \boldsymbol{\Sigma}_0^i) = \mathrm{Softmax}(s_{p0}^i, s_{p1}^i)$

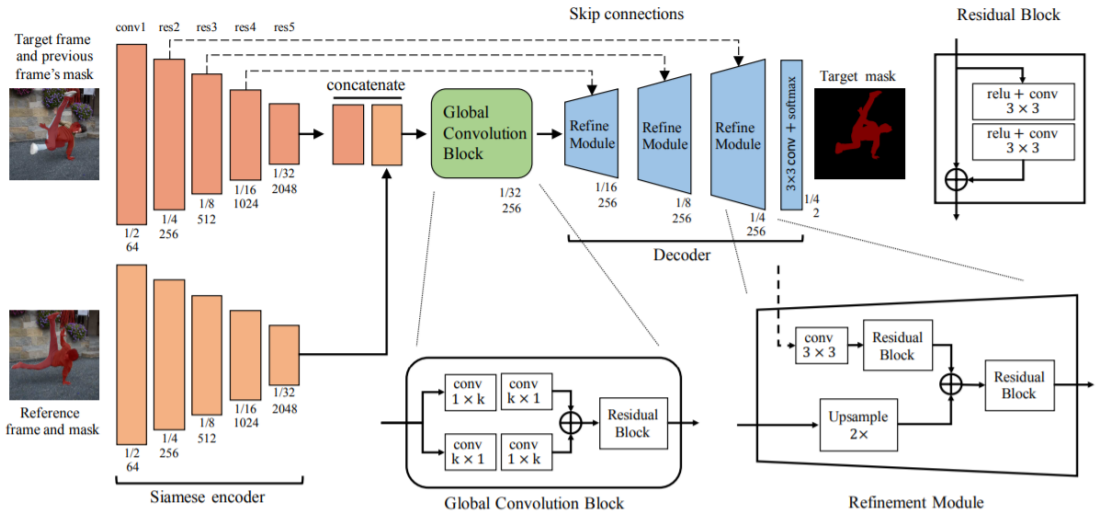
  for $k = 2, 3$: compute $\alpha_{p,k}^i$ from (6)

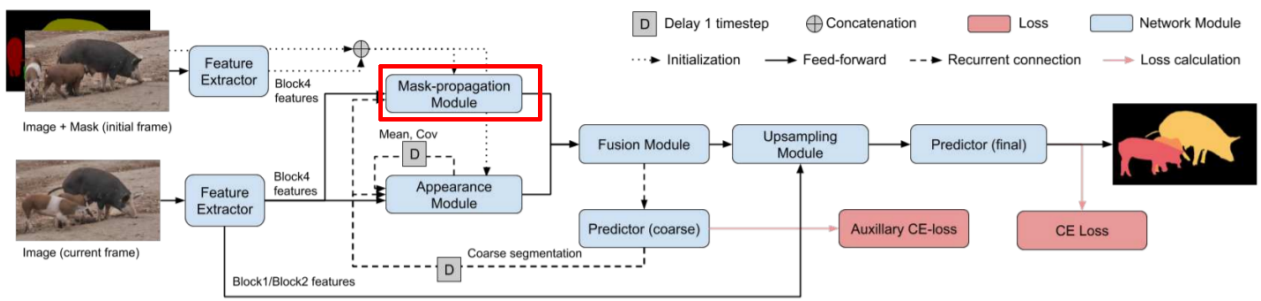
  for $k = 2, 3$: compute $\tilde{\boldsymbol{\mu}}_k^i, \tilde{\boldsymbol{\Sigma}}_k^i$ based on (3)

  for $k = 0, 1, 2, 3$: update $\boldsymbol{\mu}_k^i$ and $\boldsymbol{\Sigma}_k^i$ from (4)

  return $\boldsymbol{\mu}_k^i$ and $\boldsymbol{\Sigma}_k^i$

A-GAME: Mask-Propagation Module [12]



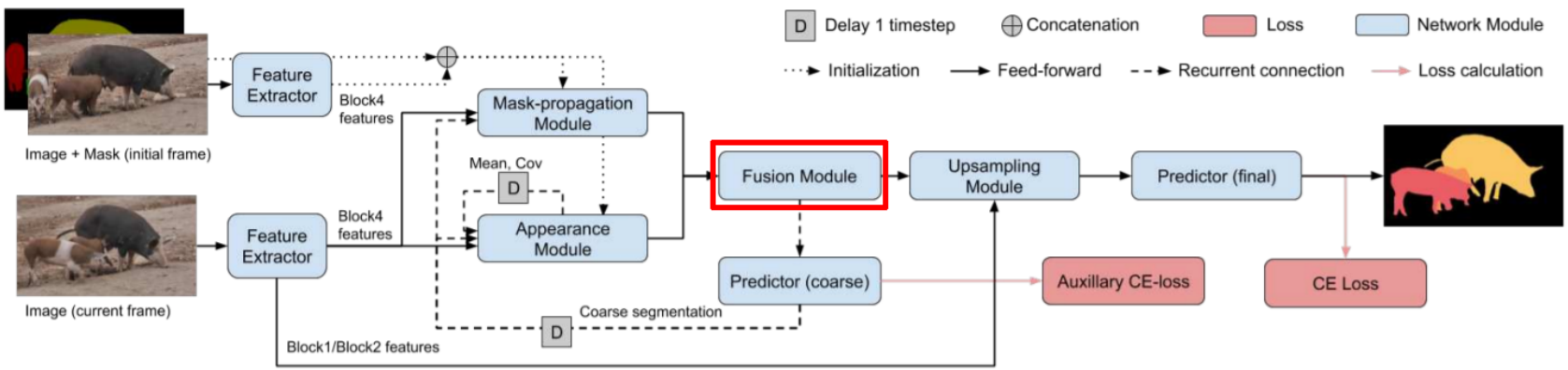


- Three convolutional layers

Wug et al. [12]

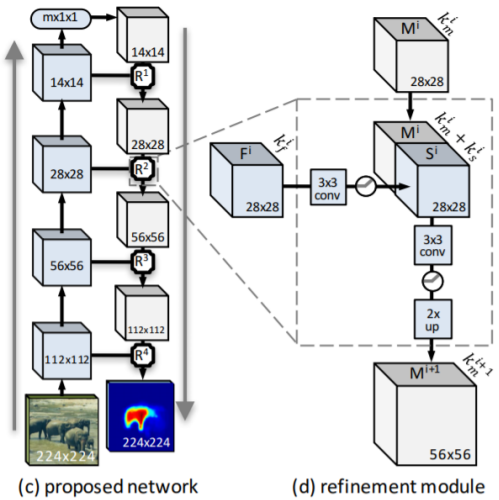
A-GAME: Fusion Module

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

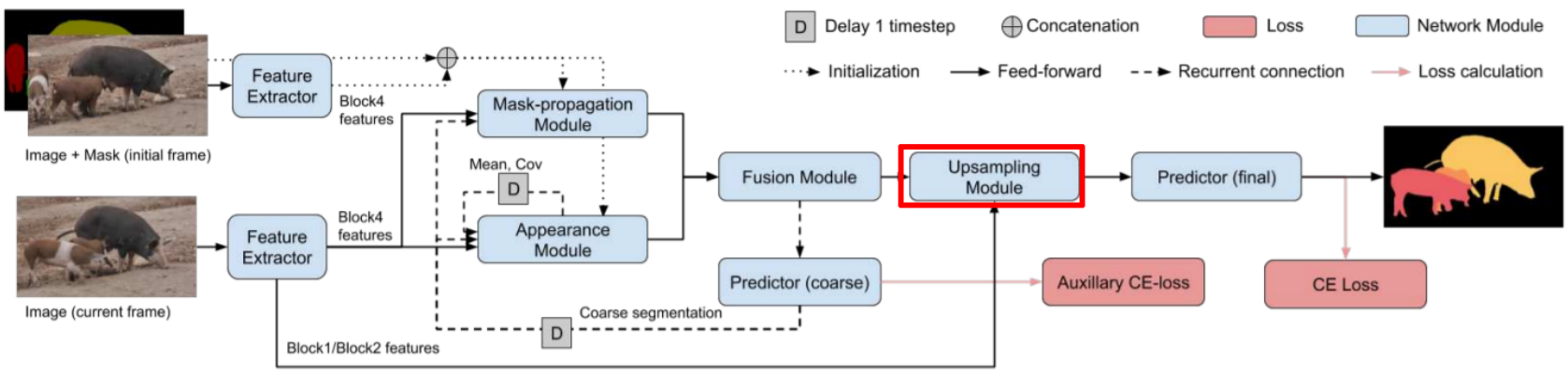


A-GAME: Upsampling Module

- Predicts a soft-segmentation mask $\hat{y}_p$
- Coarse representation is successively combined with successively shallower features [9]



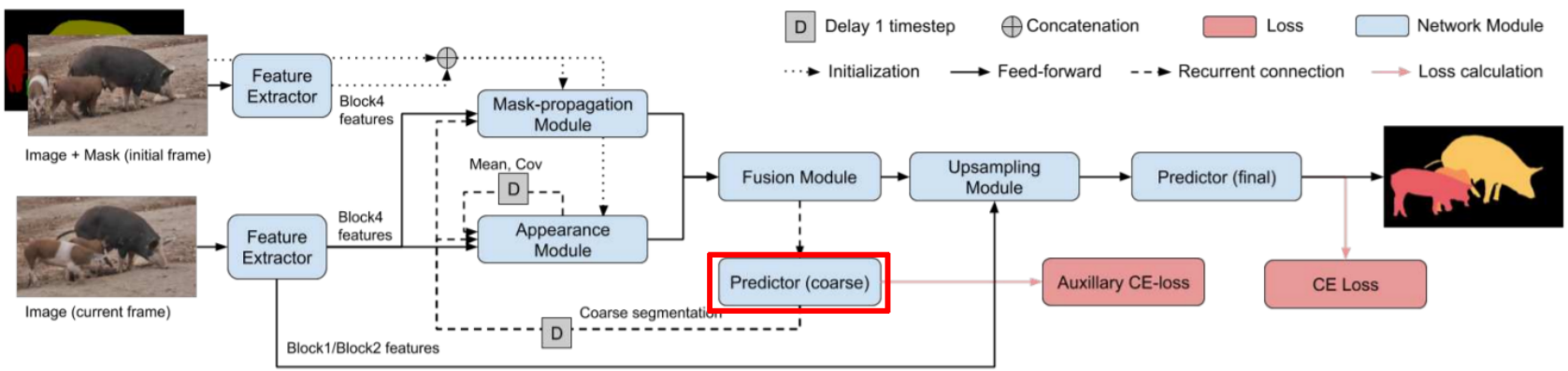
(c) proposed network (d) refinement module



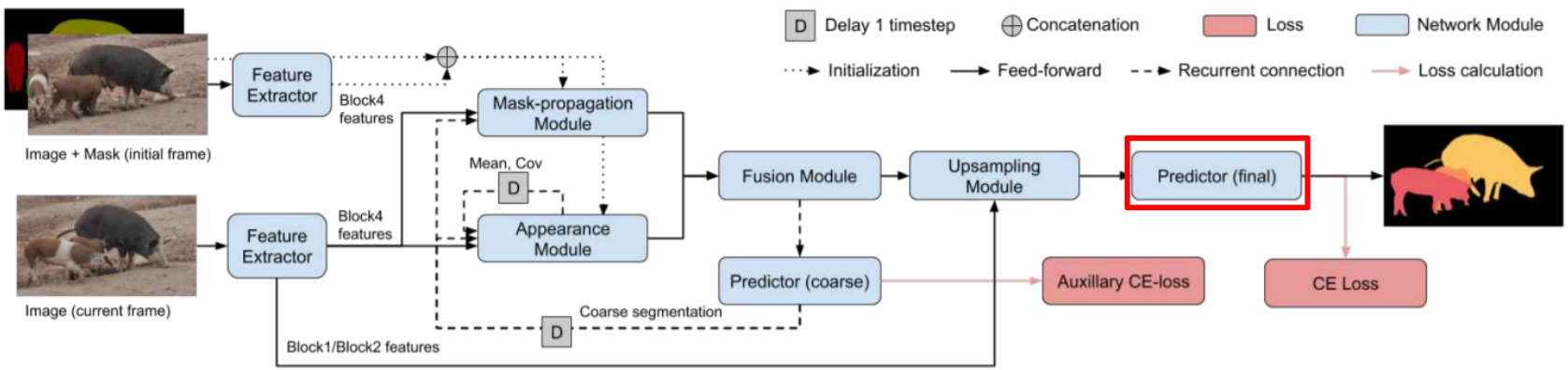
Pinheiro et al. [9]

A-GAME: Predictor Coarse

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



A-GAME: Predictor Final



A-GAME: Multi-Object

- Run the model once per object
- Combine resulting soft-segmentations with softmax-aggregation [12]
- Aggregated soft-segmentations will replace $\widetilde{y}_p$ in the recurrent connection

- Softmax-aggregation

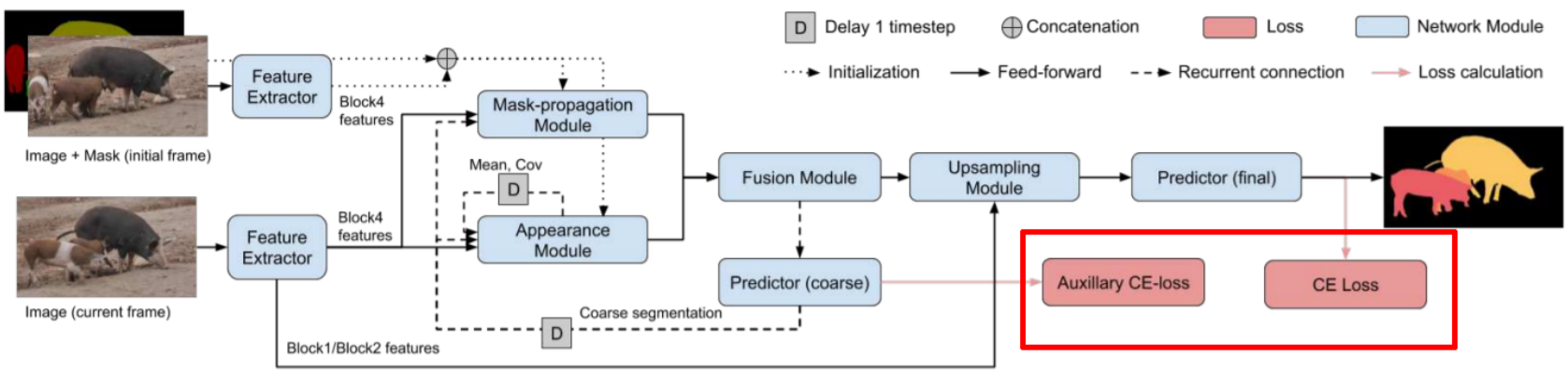
  $y_{p,o}$ - probability of object o at location p

$$y_{p,o} = \sigma(\text{logit}(y_{p,o})) = \frac{\frac{y_{p,o}}{1-y_{p,o}}}{\sum_{i=1}^{O} \frac{y_{p,o}}{1-y_{p,o}}}$$

Wug et al. [12]

A-GAME: Training Loss

- Training sample: video of n frames and 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

- Datasets:
  - DAVIS2017 [10]
  - YouTube-VOS [13]
  - 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. [10], Xu et al. [13], Cheng et al. [3], Everingham et al. [4]

A-GAME: Training Steps

- 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

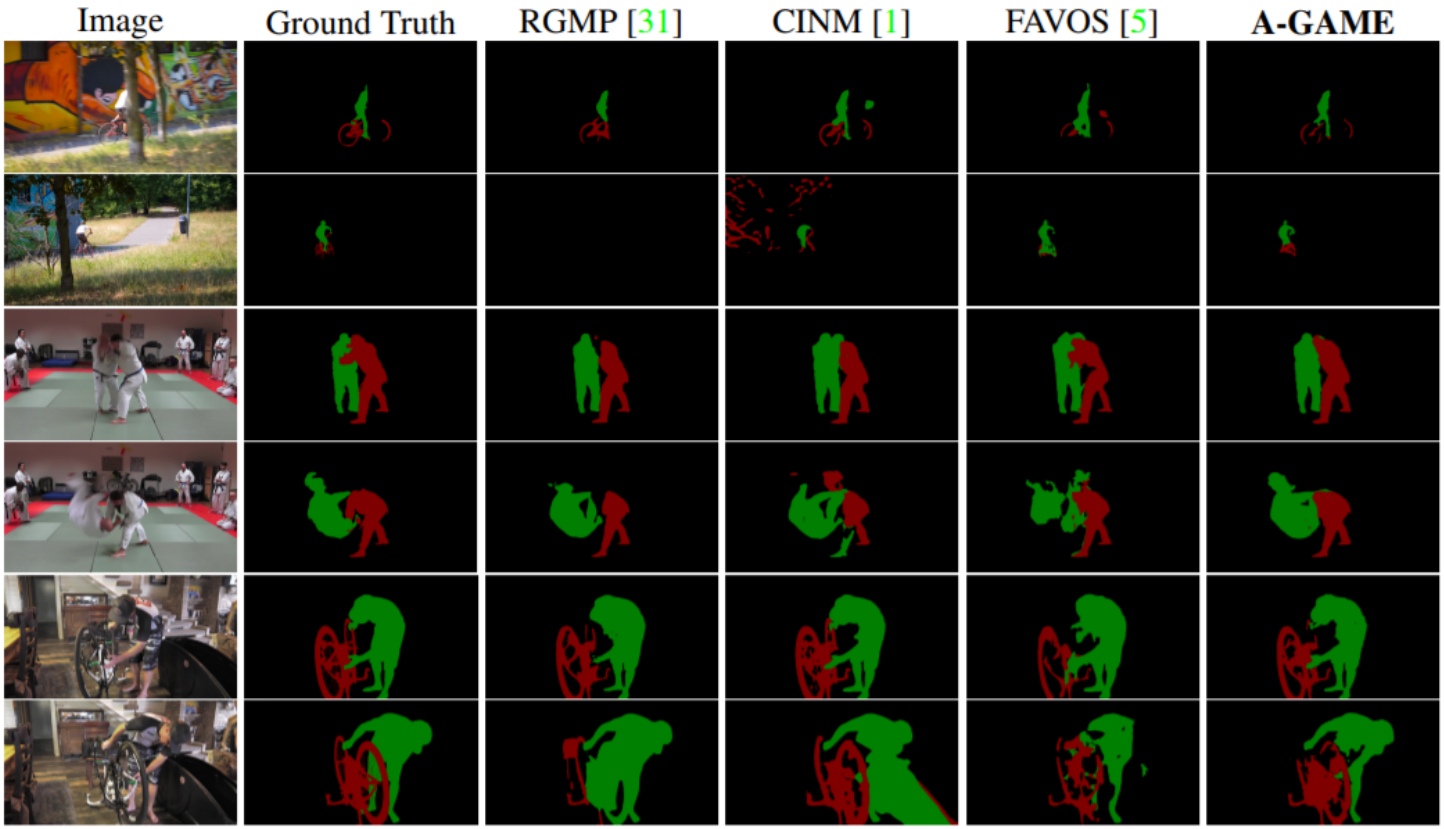
Table 3. State-of-the-art comparison on the DAVIS2017 validation set. For each method we report whether it employs online fine-tuning (O-Ft), is causal, and the final performance $\mathcal{J}$ (%). Our approach obtains superior results compared to state-of-the-art methods without online fine-tuning.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



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

- Supervised method
- Unsupervised video object segmentation task
- Single object: primary object

COSNet: Intuition

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

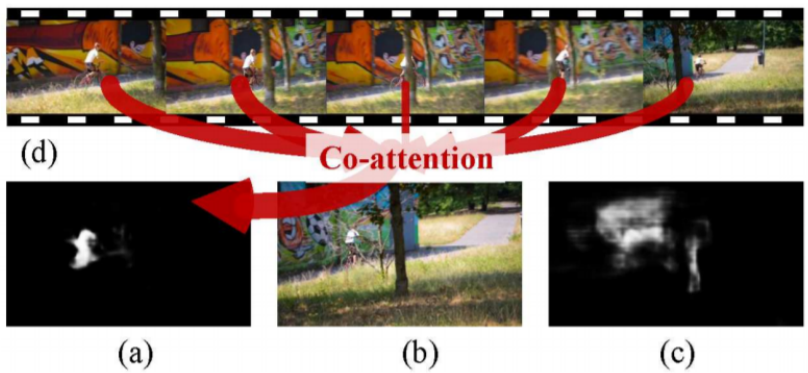


Figure 1. Illustration of our intuition. Given an input frame (b), our method leverages information from multiple reference frames (d) to better determine the foreground object (a), through a co-attention mechanism. (c) An inferior result without co-attention.

COSNet: Training & Testing

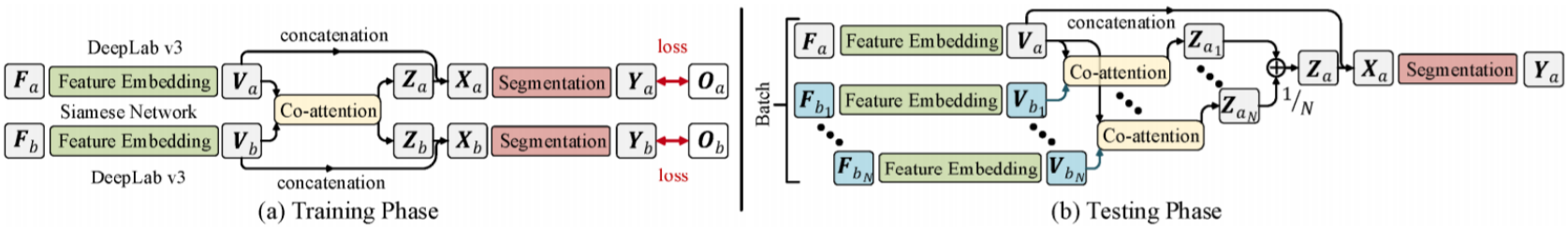


Figure 4. Schematic illustration of training pipeline (a) and testing pipeline (b) of COSNet.

COSNet: Architecture

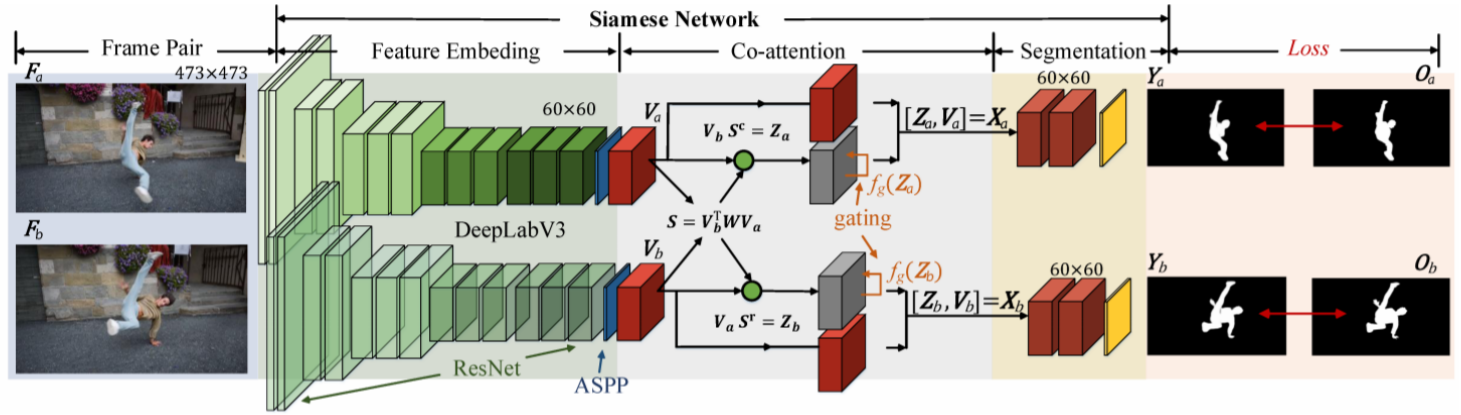


Figure 2. Overview of COSNet in the training phase. A pair of frames $\{\mathbf{F}_a, \mathbf{F}_b\}$ is fed into a feature embedding module to obtain the feature representations $\{\mathbf{V}_a, \mathbf{V}_b\}$. Then, the co-attention module computes the attention summaries that encode the correlations 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]

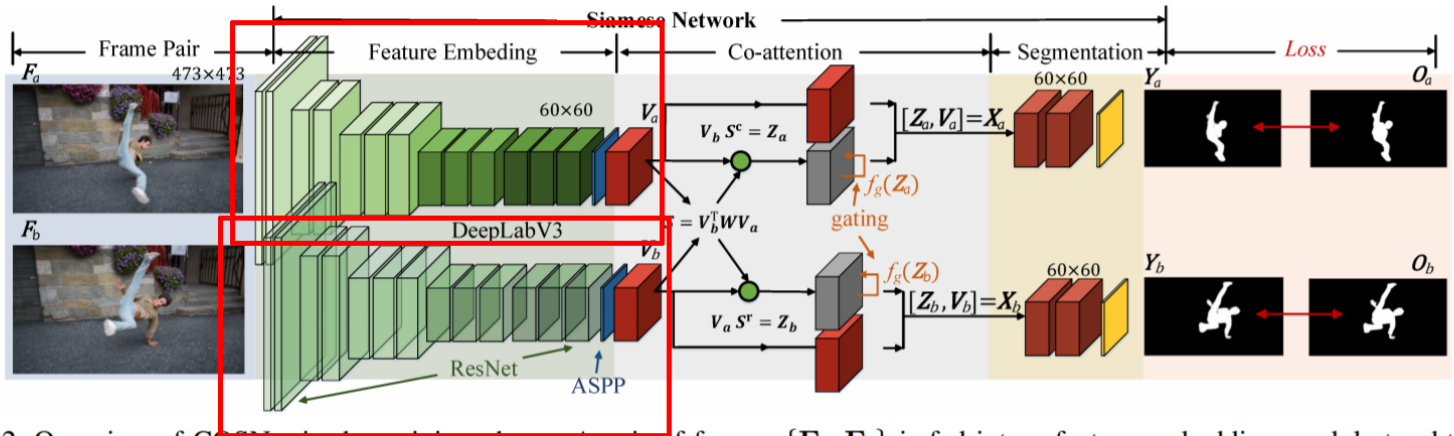


Figure 2. Overview of COSNet in the training phase. A pair of frames $\{\mathbf{F}_a, \mathbf{F}_b\}$ is fed into a feature embedding module to obtain the feature representations $\{\mathbf{V}_a, \mathbf{V}_b\}$. Then, the co-attention module computes the attention summaries that encode the correlations 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 a

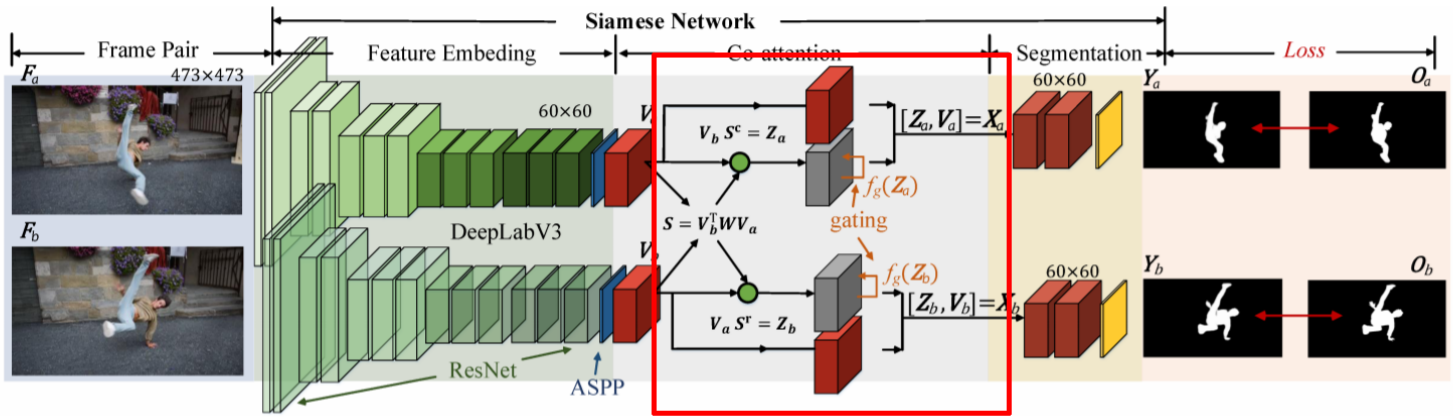


Figure 2. Overview of COSNet in the training phase. A pair of frames $\{\mathbf{F}_a, \mathbf{F}_b\}$ is fed into a feature embedding module to obtain the feature representations $\{\mathbf{V}_a, \mathbf{V}_b\}$. Then, the co-attention module computes the attention summaries that encode the correlations 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

- Co-Attention mechanisms:
  - Vanilla co-attention
  - Symmetric co-attention
  - Channel-wise co-attention

- Gated co-attention

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

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

COSNet: Co-Attention Module
Vanilla co-attention

- 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

  $\mathbf{S}_{i,j}$ - similarity between location i in image b and location j in image a

  $\mathbf{W} = \mathbf{P}^{-1}\mathbf{D}\mathbf{P}$, $\mathbf{D}$ - diagonal matrix

  $\Rightarrow \mathbf{S} = \mathbf{V}_b^T \mathbf{P}^{-1}\mathbf{D}\mathbf{P}\mathbf{V}_a$

- Each location first undergoes linear transformation and then we compute distances

COSNet: Co-Attention Module
Symmetric co-attention

- Constraint $\mathbf{W}$ to be symmetric

  $\Rightarrow \mathbf{P}^T\mathbf{P} = \mathbf{I}$

  $\Rightarrow \mathbf{S} = (\mathbf{PV_b})^T\mathbf{D}(\mathbf{PV}_a)$

- Project $\mathbf{V}_a$ and $\mathbf{V}_b$ into an othogonal common space
- Eliminate correlations between different channels

COSNet: Co-Attention Module
Channel-wise co-attention

- Replace $\mathbf{P}$ by $\mathbf{I}$

  $\Rightarrow \mathbf{W}$ diagonal matrix $\Rightarrow \mathbf{W} = \mathbf{D}_a\mathbf{D}_b$, where $\mathbf{D}_a$ and $\mathbf{D}_b$ are diagonal matrices

  $\Rightarrow \mathbf{S} = (\mathbf{D}_a\mathbf{V}_b)^T(\mathbf{D}_b\mathbf{V}_a)$

- Apply channel-wise weight
- Alleviate channel-wise redundancy

COSNet: Co-Attention

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

$$\mathbf{S}^c = softmax(\mathbf{S})$$

$$\mathbf{S}^r = softmax(\mathbf{S}^T)$$

$$\mathbf{Z}_a = \mathbf{V}_b \mathbf{S}^c = [\mathbf{Z}_a^{(1)}, \mathbf{Z}_a^{(2)}, ..., \mathbf{Z}_a^{(WH)}] \in \mathbb{R}^{C \times (WH)}$$

$\mathbf{Z}_a^{(i)}$ i-th column of $\mathbf{Z}_a$

$$\mathbf{Z}_a^{(i)} = \mathbf{V}_b \otimes \mathbf{S}^{c(i)} = \sum_{j=1}^{WH} \mathbf{V}_b^{(j)} \mathbf{s}_{ij}^c$$

COSNet: Co-Attention Module

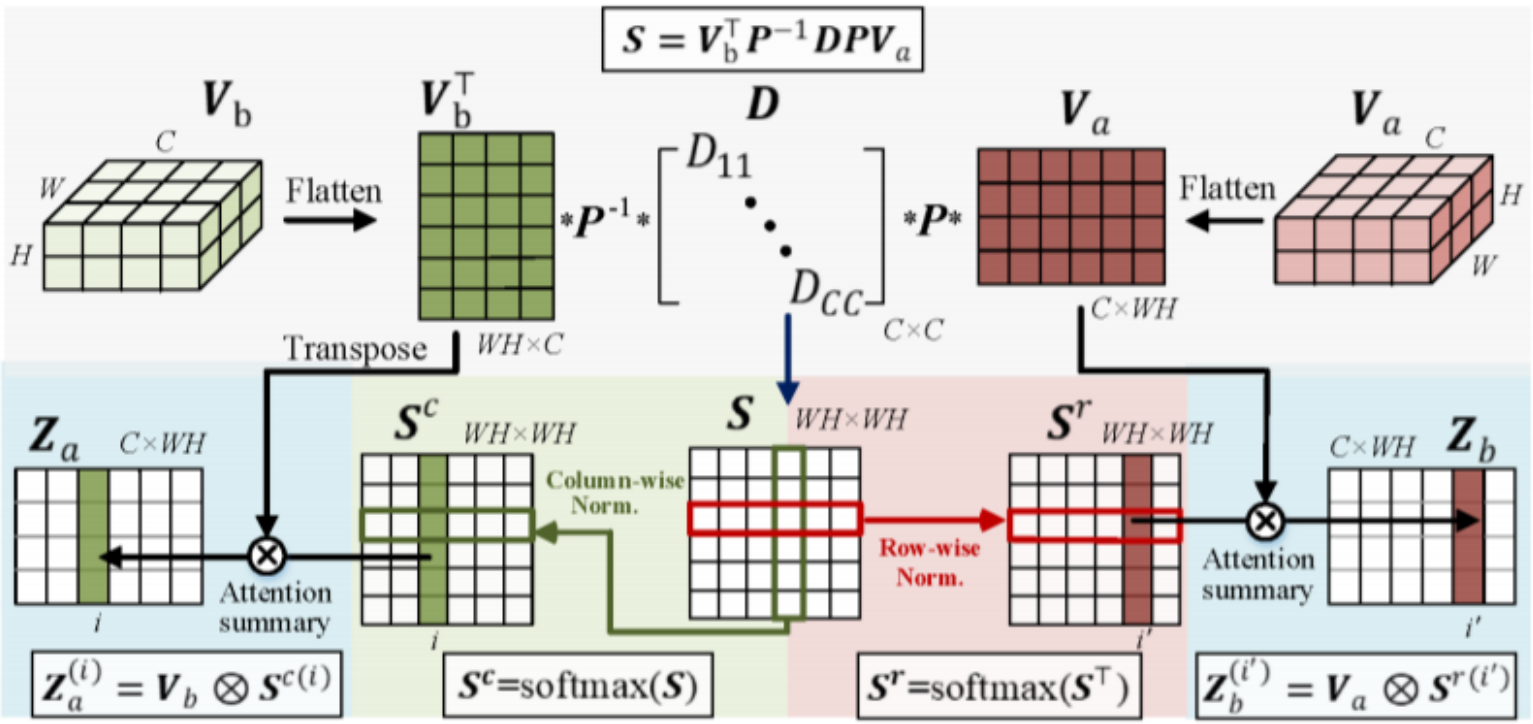


Figure 3. Illustration of our co-attention operation.

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

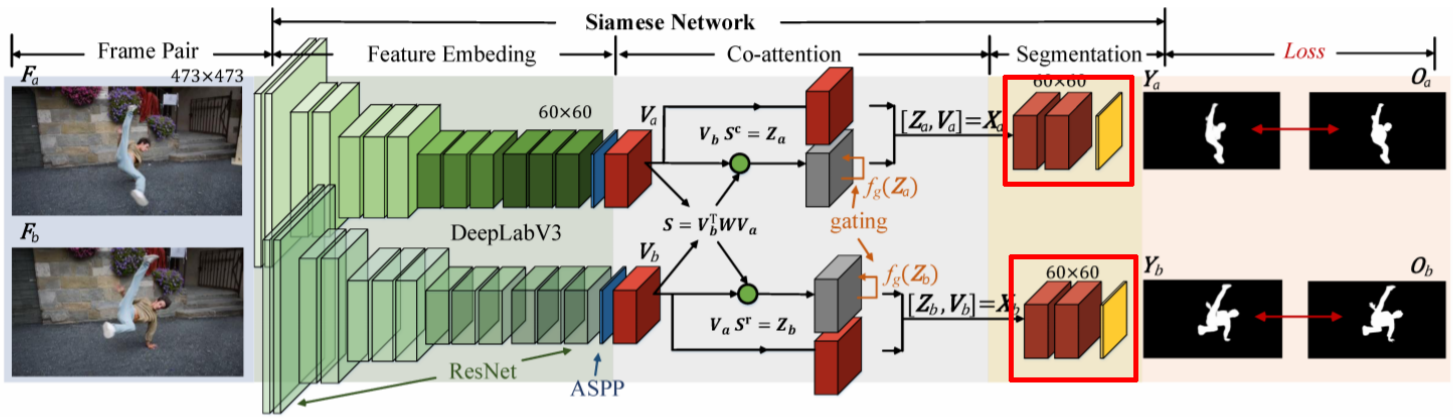


Figure 2. Overview of COSNet in the training phase. A pair of frames $\{\mathbf{F}_a, \mathbf{F}_b\}$ is fed into a feature embedding module to obtain the feature representations $\{\mathbf{V}_a, \mathbf{V}_b\}$. Then, the co-attention module computes the attention summaries that encode the correlations 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

- 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

- All in an iterative process

- $\mathbf{L}_C(\mathbf{Y}, \mathbf{O}) = -\sum_x (1-\eta) o_x log(y_x) + \eta(1-o_x) log(1-y_x)$
- $\mathbf{O}$ - ground truth
- $\mathbf{Y}$ - prediction
- η - foreground-background pixel ratio
- $\mathbf{L} = \mathbf{L}_C + \lambda |\mathbf{W}\mathbf{W}^T - \mathbf{I}|$ - to keep $\mathbf{W}$ symmetric

COSNet: Training Datasets

- Saliency datasets: MSRA10k [3] and DUT [14]
- Video object segmentation: DAVIS2016 [8]

Cheng et al. [3], Yang et al. [14], Perazzi et al. [8]

COSNet: Testing

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

- CRF refinement step

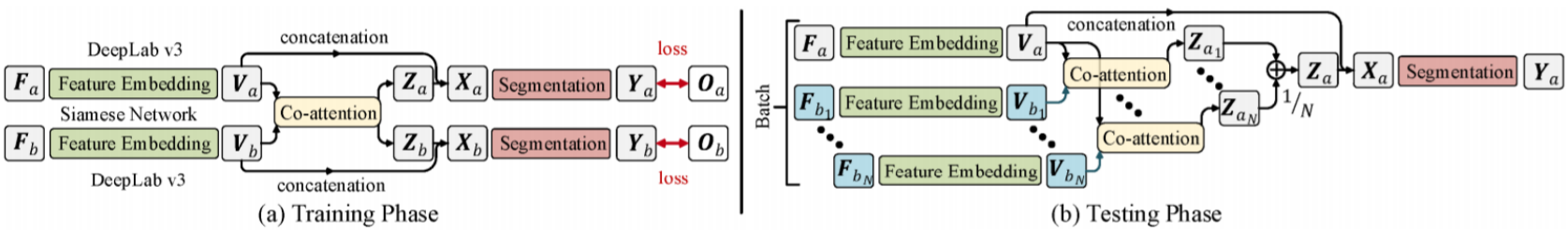


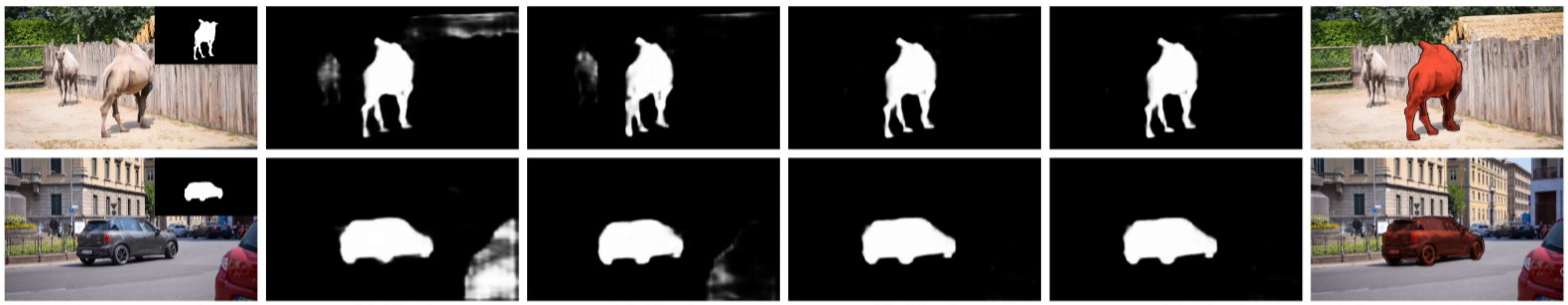
Figure 4. Schematic illustration of training pipeline (a) and testing pipeline (b) of COSNet.

COSNet: Ablation study

Network Variant	DAVIS mean $\mathcal{J}$	DAVIS $\Delta\mathcal{J}$	FBMS mean $\mathcal{J}$	FBMS $\Delta\mathcal{J}$	Youtube-Objects mean $\mathcal{J}$	Youtube-Objects $\Delta\mathcal{J}$
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].

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] 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.

References III

[8] 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.

[9] 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.

[10] 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.

[11] 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 IV

[12] 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.

[13] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 585–601, 2018.

[14] 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.