Spatiotemporal CNN for Video Object Segmentation

Kai Xu[1], Longyin Wen[2], Guorong Li[1,3,*] Liefeng Bo[2], Qingming Huang[1,3,4]

[1] School of Computer Science and Technology, UCAS, Beijing, China.
[2] JD Digits, Mountain View, CA, USA.
[3] Key Laboratory of Big Data Mining and Knowledge Management, CAS, Beijing, China.
[4] Key Laboratory of Intell. Info. Process. (IIP), Inst. of Computi. Tech., CAS, China.

xukai16@mails.ucas.ac.cn, {longyin.wen,liefeng.bo}@jd.com, {qmhuang,liguorong}@ucas.ac.cn

Abstract

In this paper, we present a unified, end-to-end trainable spatiotemporal CNN model for VOS, which consists of two branches, i.e., the temporal coherence branch and the spatial segmentation branch. Specifically, the temporal coherence branch pretrained in an adversarial fashion from unlabeled video data, is designed to capture the dynamic appearance and motion cues of video sequences to guide object segmentation. The spatial segmentation branch focuses on segmenting objects accurately based on the learned appearance and motion cues. To obtain accurate segmentation results, we design a coarse-to-fine process to sequentially apply a designed attention module on multi-scale feature maps, and concatenate them to produce the final prediction. In this way, the spatial segmentation branch is enforced to gradually concentrate on object regions. These two branches are jointly fine-tuned on video segmentation sequences in an end-to-end manner. Several experiments are carried out on three challenging datasets (i.e., DAVIS-2016, DAVIS-2017 and Youtube-Object) to show that our method achieves favorable performance against the state-of-the-arts. Code is available at https://github. com/longyin880815/STCNN.

1. Introduction

Video object segmentation (VOS) becomes a hot topic in recent years, which is a crucial step for many video analysis tasks, such as video summarization, video editing, and scene understanding. It aims to extract foreground objects from video clips. Existing VOS methods can be divided into two settings based on the degrees of human involvement, namely, *unsupervised* and *semi-supervised*. The unsupervised VOS methods [49, 44, 17, 32, 29] do not require any manual annotation, while the semi-supervised

methods [47, 6, 9, 18] rely on the annotated mask for objects in the first frame. In this paper, we are interested in the semi-supervised VOS task, which can be treated as the label propagation problem through the entire video. To maintain the temporal associations of object segments, optical flow is usually used in most of previous methods [48, 46, 5, 23, 44, 2, 15] to model the pixel consistency across the time for smoothness. However, optical flow annotation requires significant human effort, and estimation is challenging and often inaccurate, and thus it is not always helpful in video segmentation. To that end, Li *et al*. [33] design an end-to-end trained deep recurrent network to segment and track objects in video simultaneously. Xu *et al*. [51] present a sequence-to-sequence network to fully exploit long-term spatial-temporal information for VOS.

In contrast to the aforementioned methods, we design a spatiotemporal convolutional neural network (CNN) algorithm (denoted as STCNN, for short) for VOS, which is a unified, end-to-end trainable CNN. STCNN is formed by two branches, *i.e*., the temporal coherence branch and the spatial segmentation branch. The features in both branches are able to obtain useful gradient information during back-propagation. Specifically, the temporal coherence branch focuses on capturing the dynamic appearance and motion cues to provide the guidance of object segmentation, which is pre-trained in an adversarial manner from unlabeled video data following [24]. The spatial segmentation branch is a fully convolutional network focusing on segmenting objects based on the learned appearance and motion cues from the temporal coherence branch. Inspired by [15], we design a coarse-to-fine process to sequentially apply a designed attention module on multi-scale feature maps, and concatenate them to produce the final accurate prediction. In this way, the spatial segmentation branch is enforced to gradually concentrate on the object regions, which benefits both training and testing. These two branches are jointly fine-tuned on the video segmentation sequences (*e.g*., the `training` set in DAVIS-2016 [39]) in an end-to-end man-

---

*Corresponding author.

ner. We conduct several experiments on three challenging datasets, *i.e.*, DAVIS-2016 [39], DAVIS-2017 [40] and Youtube-Object [41, 20], to demonstrate the effectiveness of the proposed method against the state-of-the-art methods. Specifically, our STCNN method produces 0.838 in mIOU for semi-supervised task on the DAVIS-2016 [39], and achieves the state-of-the-art results with 0.796 in mIoU on Youtube-Object [41, 20].

Contributions. (1) We present a unified, end-to-end trainable spatiotemporal CNN algorithm for VOS without relying on optical flow, which is formed by two branches, *i.e.*, spatial segmentation branch and temporal coherence branch. (2) The temporal coherence branch is designed to capture the dynamic appearance and motion cues across the time to guide object segmentation, which is pre-trained in an adversarial manner from unlabeled video data. (3) We design a coarse-to-fine process to sequentially apply a designed attention module on multi-scale features maps, and concatenate them to produce the final accurate prediction. (4) Extensive experiments are conducted on three datasets, namely, DAVIS-2016, DAVIS-2017, and Youtube-Object, to demonstrate that the proposed method achieves favorable performance compared to the state-of-the-arts.

2. Related Work

Semi-supervised video object segmentation. Semi-supervised VOS aims to segment video objects based on the preliminarily provided foreground regions, and propagates them to the remaining frames. In [1], a patch-based probabilistic graphical model is presented for semi-supervised VOS, which uses a temporal tree structure to link patches in adjacent frames to exactly infer the pixel labels in video. Jain *et al.* [20] design a higher-order supervoxel label consistency potential for foreground region propagation, which leverages bottom-up supervoxels to guide the estimation towards long-range coherent regions. Wen *et al.* [48] integrate the multi-part tracking and segmentation into a unified energy objective to handle the VOS, which is efficiently solved by a RANSAC-style approach. Tsai *et al.* [46] jointly optimize VOS and optical flow estimation in a unified framework using an iterative scheme to exploit mutually bootstrapping information between the two tasks for better performance.

Recently, the deep neural network based methods dominate the VOS task. Khoreva *et al.* [26] describe a CNN-based algorithm, which combines offline and online learning strategies, where the former produces a refined mask from the estimation of previous frame, and the latter aims to capture the appearance of the specific object instance. Cheng *et al.* [5] presents an end-to-end trainable network for simultaneously predicting pixel-wise object segmentation and optical flow in videos, which is pre-trained offline to learn a generic notion, and fine-tuned online for spe-

cific objects. Caelles *et al.* [3] design the one-shot video object segmentation (OSVOS) approach based on a fully-convolutional neural network to transfer generic semantic information to tackle the video object segmentation task. After that, Voigtlaender *et al.* [47] improve the OSVOS method by updating the network online using training examples selected based on the confidence of the network and the spatial configuration. The online updating strategy noticeably improves the accuracy but sacrifices the running efficiency. To tackle time-consuming finetuning stage in the first frame, Cheng *et al.* [6] propose a fast VOS approach, which is formed by three modules, *i.e.*, the part-based tracking, region-of-interest segmentation, and similarity-based aggregation. This method is able to immediately start to segment a specific object through the entire video fast and accurately. In [16], a recurrent neural net approach is proposed to fuse the outputs of a binary segmentation net providing a mask and a localization net providing a bounding box for each object instance in each frame, which is able to take advantage of long-term temporal structures of the video data as well as rejecting outliers. Bao *et al.* [2] propose a spatio-temporal Markov Random Field (MRF) model for VOS, which uses a CNN to encode the spatial dependencies among pixels, and optical flow to establish the temporal dependencies. An efficient CNN-embedded algorithm is presented to perform approximate inference in the MRF to complete the VOS task.

Unsupervised video segmentation. Some unsupervised video segmentation algorithms use the bottom-up strategy to group spatial-temporal coherent tubes without any prior information. Xu *et al.* [50] implement a graph-based hierarchical segmentation method within the streaming framework, which enforces a Markovian assumption on the video stream to approximate full video segmentation. Yu *et al.* [53] propose an efficient and robust video segment algorithm based on parametric graph partitioning, that identifies and removes between-cluster edges to generate node clusters to complete video segmentation.

Several other unsupervised video segmentation methods upgrade bottom-up video segmentation to object-level segments. Lee *et al.* [30] use the static and dynamic cues to identify object-like regions in any frame, and discover hypothesis object groups with persistent appearance and motion. Then, each ranked hypothesis is used to estimate a pixel-level object labeling across all frames. Li *et al.* [31] track multiple holistic figure-ground segments simultaneously to generate video object proposals, which trains an online non-local appearance models for each track using a multi-output regularized least squares formulation. Papazoglou *et al.* [36] present a fast unsupervised VOS method, which simply aggregates the pixels in video by combining two kinds of motion boundaries extracted from optical flow to generate the proposals. In [49], a series of easy-
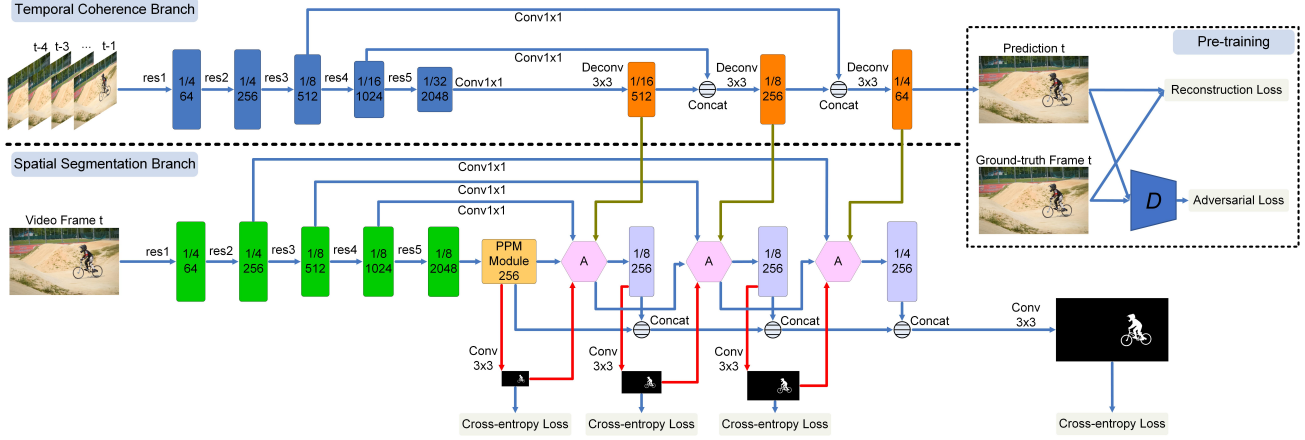
Figure 1: Overview of the network architecture of our STCNN algorithm. The part above the dashed line is the temporal coherence branch, and the part below the dashed line is the spatial segmentation branch. The red lines indicate the attention mechanism used in our model, and the hexagon indicates the attention module. Notably, each convolution layer is followed by a batch normalization layer [19] and a ReLU layer.

to-group instances of an object are discovered, and the appearance model of the instances are iteratively updated to detect harder instances in temporally-adjacent frames. Tokmakov *et al.* [44] use a fully convolutional network to learn motion patterns in videos to handle VOS, which designs an encoder-decoder style architecture to first learn a coarse representation of the optical flow field features, and then refine it iteratively to produce motion labels at high-resolution.

3. Spatiotemporal CNN for VOS

As described above, we design a spatiotemporal CNN for VOS. Specifically, given a video sequence $\mathcal{X} = \{X_1, \cdots, X_i, \cdots\}$, we aim to use our STCNN model to generate the segmentation results, *i.e.*, $\mathcal{S} = \{S_1, \cdots, S_i, \cdots\}$, where S_i is the segmentation mask corresponding to X_i. At time t, STCNN takes the previous δ frames, *i.e.*, $X_{t-\delta}, \cdots, X_{t-1}$, and the current frame X_t, to predict the segmentation results at current frame S_t[1]. As shown in Figure 1, STCNN is formed by two branches, *i.e.*, the temporal coherence branch and the spatial segmentation branch. The temporal coherence branch learns the spatiotemporal discriminative features to capture the dynamic appearance and motion cues of video sequences instead of using optical flow. Meanwhile, the spatial segmentation branch is a fully convolutional network designed to segment objects with temporal constraints from the temporal coherence branch. In the following sections, we will describe these two branches in detail.

3.1. Temporal Coherence Branch

Architecture. As shown in Figure 1, we construct the temporal coherence branch based on the backbone ResNet-101 network [14], with the input number of channels 3δ. That is, we concatenate the previous δ frames and feed them into the temporal coherence branch for prediction. After that, we use three deconvolution layers with the kernel size 3×3. To preserve spatiotemporal information in each resolution, we use three skip connections to concatenate low layer features. The convolution layer with kernel size 1×1 is used to compact features for efficiency. Notably, each convolution or deconvolution layer is followed by a batch normalization layer [19] and a ReLU layer for non-linearity.

Pretraining. Motivated by [24], we use the adversarial manner to train the temporal coherence branch by predicting future frames from unlabeled video data. Specifically, we set the temporal coherence branch as the generator $\mathcal{G}$, and construct a discriminator $\mathcal{D}$ to identify the generated video frames from $\mathcal{G}$ and the real video frames. Here, we use the Inception-v3 network [43] pretrained on the ILSVRC CLS-LOC dataset [42]. We replace the last fully connected (FC) layer by a randomly initialized 2-class FC layer as the discriminator $\mathcal{D}$.

At time t, we use the generator $\mathcal{G}$ to produce the prediction $\hat{X}_t$ of the current frame, based on previous δ frames $X_{t-\delta}, \cdots, X_{t-1}$, *i.e.*, $\hat{X}_t = \mathcal{G}(\{X_{t-i}\}_{i=1}^{\delta})$. Then, the discriminator $\mathcal{D}$ is adopted to distinguish the generated frame $\hat{X}_t$ from the real one X_t. The generator $\mathcal{G}$ and discriminator $\mathcal{D}$ are trained iteratively in an adversarial manner [11]. That is, for the fixed parameter $W_{\mathcal{G}}$ of the generator $\mathcal{G}$, we aims to optimize the discriminator $\mathcal{D}$ to minimize the probability

---

[1]For the time index $t < \delta$, we copy the first frame $\delta - t$ times to get the δ frames for segmentation.

of making mistakes, which is formulated as:

$$\min_{W_\mathcal{D}} -\log\left(1 - \mathcal{D}(\hat{X}_t)\right) - \log \mathcal{D}(X_t) \qquad (1)$$

where $\hat{X}_t = \mathcal{G}(\{X_{t-i}\}_{i=1}^{\delta})$ is the generated frame from $\mathcal{G}$ based on previous δ frames, and X_t is the real video frame. Meanwhile, for the fixed parameter $W_\mathcal{D}$ of the discriminator $\mathcal{D}$, we expect the generator $\mathcal{G}$ to generate a video frame more like a real one, *i.e.*,

$$\min_{W_\mathcal{G}} \|X_t - \hat{X}_t\|_2 - \lambda_{\text{adv}} \cdot \log \mathcal{D}(\hat{X}_t) \qquad (2)$$

where the first term is the mean square error, penalizing the differences between the fake frame $\hat{X}_t$ and the real frame X_t, the second term is the adversarial term used to maximize the probability of $\mathcal{D}$ making a mistake, and λ_{adv} is the predefined parameter used to balance these two terms. In this way, the discriminator $\mathcal{D}$ and generator $\mathcal{G}$ are optimized iteratively to make the generator $\mathcal{G}$ capturing the discriminative spatiotemporal features in the video sequences.

3.2. Spatial Segmentation Branch

The spatial segmentation branch is constructed based on the ResNet-101 network [14] by replacing the convolution layers in the last two residual blocks (*i.e.*, res4 and res5) with the dilated convolution layers [4] of stride 1, which aims to preserve the high resolution for segmentation accuracy. Then, we use the PPM module [54] to exploit the global context information by different-region-based context aggregation, followed by three designed attention modules to refine the predictions. That is, we apply the attention modules sequentially on multi-scale feature maps to help the network focus on object regions and ignore the background regions. After that, we concatenate the multi-scale feature maps, followed by a 3×3 convolution layer to produce the final prediction, see Figure 1.

Notably, we design the attention module to focus on object regions for accurate results. As shown in Figure 2, we first use the element-wise addition to exploit high-level context, and concatenate the temporal coherence features to integrate temporal constraints. After that, we use the predicted mask from the previous coarse scale feature map to guide the attention of the network, *i.e.*, use the element-wise multiplication to mask the feature map in the current stage. Let $\hat{S}_t$ to be the predicted mask at current stage. We multiply $\hat{S}_t$ on the feature map in element-wise and add it to the concatenated features for prediction. In this way, the features around the object regions are enhanced, which enforces the network gradually to concentrate on object regions for accurate results.

The pixel-wise binary cross-entropy with the softmax function $P(\cdot)$ is used in multi-scale feature maps to guide the network training, see Figure 1, which is defined as,

$$\mathcal{L}(S_t, S_t^*) \quad = -\sum\nolimits_{\ell_{i,j,t}^*=1} \log P(\ell_{i,j,t} = 1)$$
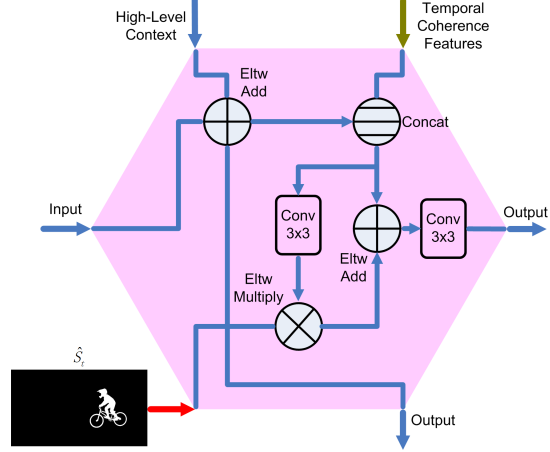$$-\sum\nolimits_{\ell_{i,j,t}^*=0} \log P(\ell_{i,j,t} = 0) \qquad (3)$$

Figure 2: The architecture of the attention module. $\hat{S}_t$ denotes the segmented mask in the current stage.

where $\ell_{i,j,t}^*$ and $\ell_{i,j,t}$ are the labels of the ground-truth mask S_t^* and the predicted mask S_t at the coordinate (i, j), $\ell_{i,j,t} = 1$ indicates that the prediction is foreground at the coordinate (i, j), and $\ell_{i,j,t} = 0$ indicates that the prediction is background at the coordinate (i, j).

3.3. Network Implementation and Training

We implement our STCNN algorithm in Pytorch [37]. All the training and testing codes and the trained models are available at https://github.com/longyin880815/STCNN. In training phase, we first pretrain the temporal coherence branch and the spatial segmentation branch individually, and iteratively update the models of both branches. After that, we finetune both models on each sequence for online processing.

Pretraining temporal coherence branch. We pretrain the temporal coherence branch in the adversarial manner on the training and validation sets of the ILSVRC 2015 VID dataset [42], which consists of $4,417$ video clips in total, *i.e.*, $3,862$ video clips in the training set and 555 video clips in the validation set. The backbone ResNet-101 network in our generator $\mathcal{G}$ is initialized by the pretrained model on the ILSVRC CLS-LOC dataset [42], and the other convolution and deconvolution layers are randomly initialized by the method [13]. While the discriminator $\mathcal{D}$ is initialized by the pretrained model on the ILSVRC CLS-LOC dataset [42], with the last 2-class FC layer initialized by the method [13]. Meanwhile, we randomly flip all frames in a video clip horizontal to augment the training data, and resize all frames to the size $(480, 854)$ for training. The batch size is set to 3, and the Adam optimization algorithm [27] is used to train the model. We set δ to 4, and use the learning rates 10^{-7} and 10^{-4} to train the generator $\mathcal{G}$ and the discriminator $\mathcal{D}$, respectively. The adversarial weight λ_{adv} is set to 0.001 in training phase.

Table 1: Performance on the validation set of DAVIS-2016. The performance of the semi-supervised VOS methods are shown in the left part, while the performance of the unsupervised VOS methods are shown in the right part. The symbol ↑ means higher scores indicate better performance, while ↓ means lower scores indicate better performance. In the last row, the numbers in parentheses are running time reported in the original papers of the corresponding methods.

Metric		Semi-supervised								Unsupervised			
	Ours	CRN[15]	OnAVOS[47]	OSVOS[3]	MSK[38]	CTN[23]	SegFlow[5]	VPN[22]	ARP[28]	LVO[45]	FSEG[21]	LMP[44]
$\mathcal{J}$ Mean (↑)	0.838	0.844	**0.861**	0.798	0.797	0.735	0.761	0.750	0.762	0.759	0.707	0.700
Recall (↑)	0.961	**0.971**	0.961	0.936	0.931	0.874	0.906	0.901	0.911	0.891	0.835	0.850
Decay (↓)	**0.049**	0.056	0.052	0.149	0.089	0.156	0.121	0.093	0.007	0.000	0.015	0.013
$\mathcal{F}$ Mean (↑)	0.838	**0.857**	0.849	0.806	0.754	0.693	0.760	0.724	0.706	0.721	0.653	0.659
Recall (↑)	0.915	**0.952**	0.897	0.926	0.871	0.796	0.855	0.842	0.835	0.834	0.738	0.792
Decay (↓)	0.064	**0.052**	0.058	0.150	0.090	0.129	0.104	0.136	0.079	0.013	0.018	0.025
$\mathcal{T}$ (↓)	0.191	-	0.190	0.376	0.189	0.198	**0.182**	0.300	0.359	0.255	0.295	0.688
Time(s/f)	3.90	(0.73)	(15.57)	(9.24)	(12.0)	(1.3)	(7.9)	-	-	-	-	-

Pretraining spatial segmentation branch. We use the MSRA10K salient object dataset [8] and the PASCAL VOC 2012 segmentation dataset [10] to pretrain the spatial segmentation branch. The MSRA10K dataset contains $10,000$ images, and the PASCAL VOC 2012 dataset contains $11,355$ images. Meanwhile, we randomly flip the images horizontally, and rotate the images to augment the data for training. Each training image is resized to $(300, 300)$. The SGD algorithm with the batch size 8 and learning rate 10^{-3} is used to optimize the model. In addition, we directly add the cross-entropy losses on multi-scale predictions (see Figure 1) to compute the overall loss for training.

Iterative offline training for VOS. After pretraining, we jointly finetune the model on the training set of DAVIS-2016 [39] for VOS, which includes 30 video clips. Specifically, we train the temporal coherence branch and the spatial segmentation branch iteratively. When optimizing the temporal coherence branch, we freeze the weights of the spatial segmentation branch, and use the learning rates 10^{-8} and 10^{-4} to train the generator $\mathcal{G}$ and the discriminator $\mathcal{D}$, respectively. The Adam algorithm is used to optimize the weights in temporal coherence branch with the batch size 1. For training the spatial segmentation branch, similarly we fix the weights in the temporal coherence branch and only update the weights in the spatial segmentation branch using the SGD algorithm with the learning rate 10^{-4}. For better training, we randomly flip horizontally, rotate and rescale to augment the training data. For this iterative learning process, each branch in the network is able to obtain useful information from another branch through backpropagation. In this way, the spatial segmentation branch can receive useful temporal information from the temporal coherence branch, while the temporal coherence branch can learn more effective spatiotemporal features for accurate segmentation.

Online training for VOS. To adapt the network to a specific object for VOS, we finetune the network on the first frame for each video clip. Since we only have the annotation mask in the first frame, only the spatial segmentation branch is optimized. Each mask in the first frame is augmented to generate multiple training samples to increase the diversity. Specifically, we use the "lucid dream" strategy [25] to generate in-domain training data based on the provided annotation in the first frame, including 5 steps, *i.e.*, illumination changing, foreground-background splitting, object motion simulating, camera view changing, and foreground-background merging. Notably, in contrast to [25], we do not generate the optical flow since our STCNN do not require the optical flow for video segmentation. The SGD algorithm with the learning rate 10^{-4} and batch size 1 is used to train the network online.

4. Experiment

We evaluate the proposed algorithm against state-of-the-art VOS methods on three challenging datasets, namely the DAVIS-2016 [39], DAVIS-2017 [40], and Youtube-Object [41, 20]. All the experiments are conducted on a workstation with a 3.6 GHz Intel i7-4790 CPU, 16GB RAM, and a NVIDIA Titan 1080ti GPU. The quantitative results are presented in Table 1 and 2. Some qualitative segmentation results are shown in Figure 3, and more video segmentation results can be found in supplementary material.

4.1. DAVIS-2016 Dataset

The DAVIS-2016 dataset [39] comprises of 50 sequences, $3,455$ annotated frames with a binary pixel-level foreground/background mask. Due to the computational complexity being a major bottleneck in video processing, the sequences in the dataset have a short temporal extent (about 2-4 seconds), but include all major challenges typically found in longer video sequences, such as background clutter, fast-motion, edge ambiguity, camera-shake, and out-of-view. We tested the proposed method on the 480p resolution set.

Table 2: The results on the Youtube-Objects dataset. The mean intersection-over-union is used to evaluate the performance of methods. The results are directly taken from the original paper. The symbol ↑ means higher scores indicate better performance. Bold font indicates the best result.

Method	BVS[34]	JFS[35]	SCF[20]	MRFCNN[2]	LT[25]	OSVOS[3]	MSK[38]	OFL[46]	CRN[15]	DRL[12]	OnAVOS[47]	Ours
aeroplane	0.868	0.890	0.863	-	-	0.868	0.845	**0.899**	-	0.852	-	0.869
bird	0.809	0.816	0.810	-	-	0.851	0.837	0.842	-	0.868	-	**0.879**
boat	0.651	0.742	0.686	-	-	0.754	0.774	0.740	-	**0.799**	-	0.786
car	0.687	0.709	0.694	-	-	0.709	0.640	0.809	-	0.672	-	**0.859**
cat	0.559	0.677	0.589	-	-	0.676	0.698	0.683	-	0.746	-	**0.772**
cow	0.699	0.791	0.686	-	-	0.762	0.767	**0.798**	-	0.746	-	0.781
dog	0.685	0.703	0.618	-	-	0.779	0.745	0.766	-	**0.827**	-	0.800
horse	0.589	0.678	0.540	-	-	0.714	0.641	0.726	-	0.736	-	**0.738**
motorbike	0.605	0.615	0.609	-	-	0.582	0.892	0.481	-	**0.737**	-	0.680
train	0.652	0.782	0.663	-	-	0.746	0.744	0.763	-	**0.830**	-	0.796
Mean (↑)	0.680	0.740	0.676	0.784	0.762	0.744	0.717	0.776	0.766	0.781	0.774	**0.796**

4.1.1 Evaluation

For comprehensive evaluation, we use three measures provided by the dataset, *i.e.*, region similarity $\mathcal{J}$, contour accuracy $\mathcal{F}$ and temporal instability $\mathcal{T}$. Specifically, region similarity $\mathcal{J}$ measures the number of mislabeled pixels, which is defined as the intersection-over-union (IoU) of the estimated segmentation and the ground-truth mask. Given a segmentation mask S and the ground-truth mask S^*, $\mathcal{J}$ is calculated as $\mathcal{J} = \frac{S \cap S^*}{S \cup S^*}$. The contour accuracy $\mathcal{F}$ computes the F-measure of the contour-based precision P_c and recall R_c between the contour points of estimated segmentation S and the ground-truth mask S^*, defined as $\mathcal{F} = \frac{2 P_c R_c}{P_c + R_c}$. In addition, the temporal instability $\mathcal{T}$ measures oscillations and inaccuracies of the contours, which is calculated by following [39].

4.1.2 Ablation Study

To comprehensively understand the proposed method, we conduct several ablation experiments. Specifically, we construct three variants and evaluate them on the `validation` set of DAVIS-2016, to validate the effectiveness of different components (*i.e.*, the "Lucid dream" augmentation, the attention module, and the temporal coherence branch) in the proposed method, shown in Table 3. Meanwhile, we also conduct experiments to analyze the importance of different training phases in Table 5. For a fair comparison, we use the same parameter settings except for the specific declaration.

Lucid dream augmentation. To demonstrate the effect of the "Lucid Dream" augmentation, we remove it from our STCNN model (see the forth column in Table 3). As shown in Table 3, we find that the region similarity $\mathcal{J}$ is reduced from 0.838 to 0.832. This decline (*i.e.*, 0.006) demonstrate that the "Lucid dream" data augmentation is useful to improve the performance.

Attention module. To validate the effectiveness of the attention module, we construct an algorithm by further removing the attention mechanism in the spatial segmentation branch. That is, we remove the red lines in Figure 1 to directly generate the output mask. In this way, the object region is not specifically concentrated by the network. The segmentation results of the model is reported in the third column in Table 3. We compare the third and forth columns in Table 3, and find that the attention module improves 0.01 region similarity $\mathcal{J}$, and 0.015 contour accuracy $\mathcal{F}$, which demonstrates that the attention module is critical to the performance. The main reason is that the attention module is gradually applied on multi-scale features maps, enforcing the network to focus on the object regions to generate more accurate results.

Temporal coherence branch. We construct a network based on the spatial segmentation branch without the attention module and report its results in the second column in Table 3. Comparing the results between the second and third columns in Table 3, we observe that the temporal coherence branch is critical to the performance of video segmentation, *i.e.*, it improves 0.01 mean region similarity $\mathcal{J}$ (0.812 *vs.* 0.822) and 0.013 mean contour accuracy $\mathcal{F}$ (0.807 *vs.* 0.820). Most importantly, the temporal coherence branch significantly reduces the temporal instability, *i.e.*, it reduces relative 13.4% temporal instability $\mathcal{T}$ (0.231 *vs.* 0.200). The results demonstrate that the temporal coherence branch is effective to capture the dynamic appearance and motion cues of video sequences to help generate accurate and consistent segmentation results.

Training analysis. As described in Section 3.3, we first iteratively update the pretrained temporal coherence branch and the spatial segmentation branch offline. After that, we finetune both branches on each sequence for online processing. We evaluate the proposed STCNN method with different training phase on the `validation` set of DAVIS-2016 to analyze their effects on performance in Table 5. As shown in Table 5, we find that without online training phase, the mean region similarity $\mathcal{J}$ of STCNN drops 0.096 (*i.e.*, 0.838 *vs.* 0.742), while without offline training phase, $\mathcal{J}$
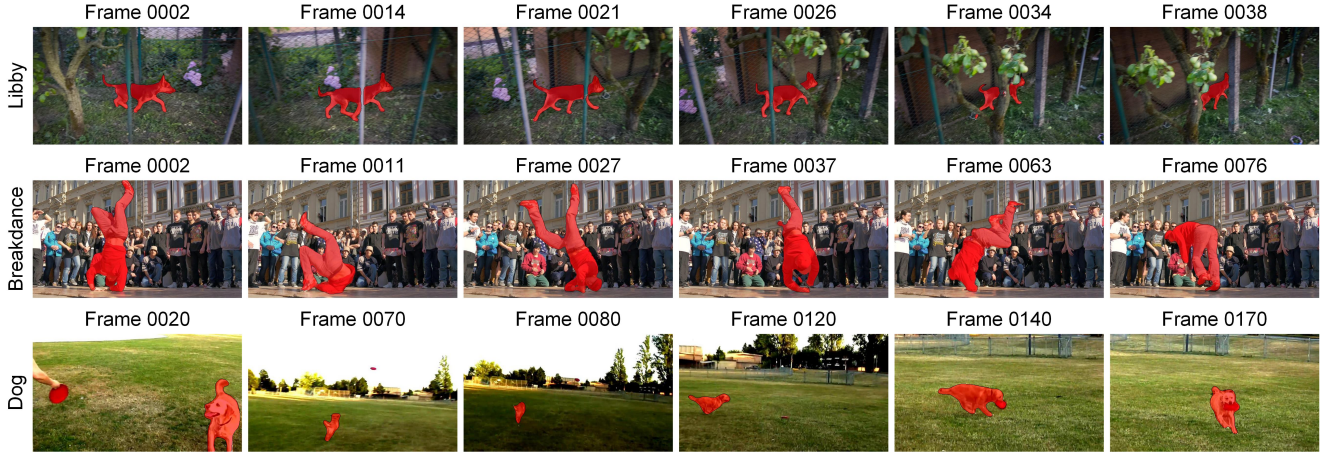
Figure 3: The qualitative segmentation results of STCNN on the DAVIS-2016 (first two rows) and Youtube-Objects (last row) datasets. The output on the pixel level are indicated by the red mask. The results show that our method is able to segment objects under several challenges, such as occlusions, deformed shapes, fast motion, and cluttered backgrounds.

Table 3: Effectiveness of various components in the proposed method. All models are evaluated on the DAVIS-2016 dataset. The symbol ↑ means high scores indicate better result, while ↓ means lower scores indicate better result.

Component	STCNN			
Temporal Coherence Branch?		✓	✓	✓
Attention Module?			✓	✓
Lucid Dream?				✓
$\mathcal{J}$ Mean (↑)	0.812	0.822	0.832	0.838
$\mathcal{F}$ Mean (↑)	0.807	0.820	0.835	0.838
$\mathcal{T}$ (↓)	0.231	0.200	0.192	0.191

of STCNN drops 0.052 (*i.e.*, 0.838 *vs.* 0.786). In summary, both training phases are extremely important to our STCNN, especially for the online training phase.

4.1.3   Comparison with State-of-the-Arts

We compare the proposed method with 7 state-of-the-art semi-supervised methods, *i.e.*, CRN [15], OnAVOS [47], OSVOS [3], MSK [38], CTN [23], SegFlow [5], and VPN [22], and 4 state-of-the-art unsupervised methods, namely ARP [28], LVO [45], FSEG [21], and LMP [44] in Table 1.

As shown in Table 1, our algorithm outperforms the existing semi-supervised algorithms (*e.g.*, OSVOS [3] and MSK [38]) and unsupervised algorithms (*e.g.*, ARP [28] and LVO [45]) with 0.838 mean region similarity $\mathcal{J}$, 0.838 mean contour accuracy $\mathcal{F}$, and 0.191 temporal stability $\mathcal{T}$, except CRN [15] and OnAVOS [47]. The OnAVOS algorithm [47] updates the network online using training examples selected based on the confidence of the network and the spatial configuration, which requires heavy consumption of time and computation resource. Our algorithm is much more efficient and do not require the optical flow in both training and testing phase. The online updating mechanism in OnAVOS [47] is complementary to our method. We believe that is can be used in our STCNN to further improve the performance. In addition, in contrast to CRN [15] relying on optical flow to render temporal coherence in both training and testing, our method uses a self-supervised strategy to implicitly exploit the temporal coherence without relying on the expensive human annotations of optical flow. The temporal coherence branch is able to capture the dynamic appearance and motion cues of video sequences, pretrained in an adversarial manner from nearly unlimited unlabeled video data.

4.1.4   Runtime Performance

We present the inference time of STCNN and the state-of-the-art methods on the `validation` set of DAVIS-2016 in the last row of Table 1. Since different algorithms are developed and evaluated on different platforms (*e.g.*, different algorithms are evaluated on different types of GPUs), it is difficult to compare the running time efficiency fairly. We report the running speed for reference. Meanwhile, we also analyze the influence of the number of iterations in the online training phase of STCNN to the segmentation accuracy and running speed in Table 4. With the number of iterations increasing, the mean region similarity $\mathcal{J}$ increases to reach a maximal value 0.838. Continue training is not able to obtain the accuracy gain, but slows down the inference speed. Thus, we set the number of iterations in online training to 400 in our experiments. Compared to the state-of-the-art methods such as OSVOS (9.24 s/f), OnAVOS (15.57 s/f), our method achieves impressive results with much faster running speed.

Table 4: Performance and running speed of the proposed STCNN with different number of iterations in online training phase.

#Iter	100	200	300	400	500	600
mIOU	0.830	0.834	0.836	0.838	0.838	0.838
time(s/f)	1.11	2.04	2.97	3.90	4.83	5.76

Table 5: Performance on DAVIS-2016 for different training phases of STCNN.

	Metric	Offline Training	Online Training	All
$\mathcal{J}$	Mean ($\uparrow$)	0.742	0.786	0.838
	Recall ($\uparrow$)	0.854	0.921	0.961
	Decay ($\downarrow$)	-0.004	0.075	0.049
$\mathcal{F}$	Mean ($\uparrow$)	0.743	0.79	0.838
	Recall ($\uparrow$)	0.806	0.871	0.915
	Decay ($\downarrow$)	0.018	0.089	0.064

Table 6: The results on the DAVIS-2017 dataset. The symbol $\uparrow$ means higher scores indicate better performance. Bold font indicates the best result.

Metric	[46]	[38]	[52]	[7]	[3]	[47]	Ours
$\mathcal{J}$ Mean ($\uparrow$)	43.2	51.2	52.5	54.6	56.6	**61.6**	58.7
$\mathcal{F}$ Mean ($\uparrow$)	-	57.3	57.1	61.8	63.9	**69.1**	64.6

4.2. DAVIS-2017 Dataset

We evaluate our STCNN on the DAVIS-2017 validation set [40], which consists of 30 video sequences with various challenging cases including multiple objects with similar appearance, heavy occlusion, large appearance variation, clutter background, etc. The mean region similarity $\mathcal{J}$ and contour accuracy $\mathcal{F}$ are used to evaluate the performance in Table 6. Our STCNN performs favorably against most of the semi-supervised methods, *e.g.*, OSVOS [3], FAVOS [7], and MSK [38], with 58.7% mean region similarity $\mathcal{J}$ and 64.6% contour accuracy $\mathcal{F}$. The results demonstrate that our STCNN is effective to segment objects in more complex scenarios with similar appearance.

4.3. Youtube-Objects Dataset

The Youtube-Objects dataset [41, 20] contains web videos from 10 object categories. 126 video sequences with more than 20, 000 frames and ground-truth masks provided by [20] are used for evaluation, where a single object or a group of objects of the same category are separated from the background. The videos in Youtube-Objects have a mix of static and moving objects, and the number of frames in each video clip ranges from 2 to 401. The mean IoU between the estimated results and the ground-truth masks in all video frames is used to evaluate the performance of the algorithms.

We compare the proposed STCNN method to 11 state-of-the-art semi-supervised algorithms, namely BVS [34],

JFS [35], SCF [20], MRFCNN [2], LT [25], OSVOS [3], MSK [38], OFL [46], CRN [15], DRL [12], and OnAVOS [47] in Table 2. As shown in Table 2, we observe that the STCNN method produces the best results with 0.796 mean IoU, which surpasses the state-of-the-art results, *i.e.*, MR-FCNN [2] (0.784 mean IoU), with 0.012 mIoU. Compared to the optical flow based methods [46, 38], our STCNN method performs well on fast moving objects, such as *car* and *cat*. The estimation of optical flow for fast moving objets is inaccurate, affecting the segmentation accuracy. Our STCNN relies on the temporal coherence branch to capture discriminative spatiotemporal features, which is effective to tackle such scenario. Meanwhile, the algorithm [20] use long-term supervoxels to capture the temporal coherence. Only the superpixels are used in segmentation, causing the inaccurate boundaries of objects. In contrast, our algorithm design a coarse-to-fine process to sequentially apply the attention module on multi-scale feature maps, enforcing the network to focus on object regions to generate accurate results, especially for the non-rigid objects, *e.g.*, *cat* and *horse*. The qualitative results are shown in the last three rows in Figure 3.

5. Conclusion

In this work, we present an end-to-end trained spatiotemporal CNN for VOS, which is formed by two branches, *i.e.*, the temporal coherence branch and the spatial segmentation branch. The temporal coherence branch is pretrained in an adversarial fashion, and used to predict the appearance and motion cues in the video sequence to guide object segmentation without using optical flow. The spatial segmentation branch is designed to segment object instance accurately based on the predicted appearance and motion cues from the temporal coherence branch. In addition, to obtain accurate segmentation results, a coarse-to-fine process is iteratively applied on multi-scale feature maps in the spatial segmentation branch to refine the predictions. These two branches are jointly trained in an end-to-end manner. Extensive experimental results on three challenging datasets, *i.e.*, DAVIS-2016, DAVIS-2017, and Youtube-Object, demonstrate that the proposed method achieves favorable performance against the state-of-the-arts.

References

[1] Vijay Badrinarayanan, Ignas Budvytis, and Roberto Cipolla. Semi-supervised video segmentation using tree structured graphical models. *TPAMI*, 35(11):2751–2764, 2013. 2

[2] Linchao Bao, Baoyuan Wu, and Wei Liu. CNN in MRF: video object segmentation via inference in A cnn-based higher-order spatio-temporal MRF. In *CVPR*, 2018. 1, 2, 6, 8

[3] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, pages 5320–5329, 2017. 2, 5, 6, 7, 8

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *CoRR*, abs/1412.7062, 2014. 4

[5] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *ICCV*, pages 686–695, 2017. 1, 2, 5, 7

[6] Jingchun Cheng, Yi-Hsuan Tsai, Wei-Chih Hung, Shengjin Wang, and Ming-Hsuan Yang. Fast and accurate online video object segmentation via tracking parts. In *CVPR*, 2018. 1, 2

[7] Jingchun Cheng, Yi-Hsuan Tsai, Wei-Chih Hung, Shengjin Wang, and Ming-Hsuan Yang. Fast and accurate online video object segmentation via tracking parts. In *CVPR*, pages 7415–7424, 2018. 8

[8] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. Global contrast based salient region detection. *TPAMI*, 37(3):569–582, 2015. 5

[9] Hai Ci, Chunyu Wang, and Yizhou Wang. Video object segmentation by learning location-sensitive embeddings. In *ECCV*, pages 524–539, 2018. 1

[10] Mark Everingham, S. M. Ali Eslami, Luc J. Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015. 5

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014. 3

[12] Junwei Han, Le Yang, Dingwen Zhang, Xiaojun Chang, and Xiaodan Liang. Reinforcement cutting-agent learning for video object segmentation. In *CVPR*, pages 9080–9089, 2018. 6, 8

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015. 4

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3, 4

[15] Ping Hu, Gang Wang, Xiangfei Kong, Jason Kuen, and Yap-Peng Tan. Motion-guided cascaded refinement network for video object segmentation. In *CVPR*, pages 1400–1409, 2018. 1, 5, 6, 7, 8

[16] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G. Schwing. Maskrnn: Instance level video object segmentation. In *NIPS*, pages 324–333, 2017. 2

[17] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G. Schwing. Unsupervised video object segmentation using motion saliency-guided spatio-temporal propagation. In *ECCV*, pages 813–830, 2018. 1

[18] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G. Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, pages 56–73, 2018. 1

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 3

[20] Suyog Dutt Jain and Kristen Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, pages 656–671, 2014. 2, 5, 6, 8

[21] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *CVPR*, pages 2117–2126, 2017. 5, 7

[22] Varun Jampani, Raghudeep Gadde, and Peter V. Gehler. Video propagation networks. In *CVPR*, pages 3154–3164, 2017. 5, 7

[23] Won-Dong Jang and Chang-Su Kim. Online video object segmentation via convolutional trident network. In *CVPR*, pages 7474–7483, 2017. 1, 5, 7

[24] Xiaojie Jin, Xin Li, Huaxin Xiao, Xiaohui Shen, Zhe Lin, Jimei Yang, Yunpeng Chen, Jian Dong, Luoqi Liu, Zequn Jie, Jiashi Feng, and Shuicheng Yan. Video scene parsing with predictive feature learning. In *ICCV*, pages 5581–5589, 2017. 1, 3

[25] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for object tracking. *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*, 2017. 5, 6, 8

[26] Anna Khoreva, Federico Perazzi, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 2

[27] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 4

[28] Yeong Jun Koh and Chang-Su Kim. Primary object segmentation in videos based on region augmentation and reduction. In *CVPR*, pages 7417–7425, 2017. 5, 7

[29] Yeong Jun Koh, Young-Yoon Lee, and Chang-Su Kim. Sequential clique optimization for video object segmentation. In *ECCV*, pages 537–556, 2018. 1

[30] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *ICCV*, pages 1995–2002, 2011. 2

[31] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, pages 2192–2199, 2013. 2

[32] Siyang Li, Bryan Seybold, Alexey Vorobyov, Xuejing Lei, and C.-C. Jay Kuo. Unsupervised video object segmentation with motion-based bilateral networks. In *ECCV*, pages 215–231, 2018. 1

[33] Xiaoxiao Li and Chen Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *ECCV*, pages 93–110, 2018. 1

[34] Nicolas Marki, Federico Perazzi, Oliver Wang, and Alexander Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, pages 743–751, 2016. 6, 8

[35] Naveen Shankar Nagaraja, Frank R. Schmidt, and Thomas Brox. Video segmentation with just a few strokes. In *ICCV*, pages 3235–3243, 2015. 6, 8

[36] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstraint video. In *ICCV*, pages 1–6, 2013. 2

[37] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS Workshop Autodiff*. 4

[38] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, pages 3491–3500, 2017. 5, 6, 7, 8

[39] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc J. Van Gool, Markus H. Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016. 1, 2, 5, 6

[40] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbelaez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *CoRR*, abs/1704.00675, 2017. 2, 5, 8

[41] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, pages 3282–3289, 2012. 2, 5, 8

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 3, 4

[43] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 3

[44] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning motion patterns in videos. In *CVPR*, pages 531–539, 2017. 1, 3, 5, 7

[45] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In *ICCV*, pages 4491–4500, 2017. 5, 7

[46] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J. Black. Video segmentation via object flow. In *CVPR*, pages 3899–3908, 2016. 1, 2, 6, 8

[47] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017. 1, 2, 5, 6, 7, 8

[48] Longyin Wen, Dawei Du, Zhen Lei, Stan Z. Li, and Ming-Hsuan Yang. JOTS: joint online tracking and segmentation. In *CVPR*, pages 2226–2234, 2015. 1, 2

[49] Fanyi Xiao and Yong Jae Lee. Track and segment: An iterative unsupervised approach for video object proposals. In *CVPR*, 2016. 1, 2

[50] Chenliang Xu, Caiming Xiong, and Jason J. Corso. Streaming hierarchical video segmentation. In *ECCV*, pages 626–639, 2012. 2

[51] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian L. Price, Scott Cohen, and Thomas S. Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, pages 603–619, 2018. 1

[52] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K. Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, pages 6499–6507, 2018. 8

[53] Chen-Ping Yu, Hieu Le, Gregory J. Zelinsky, and Dimitris Samaras. Efficient video segmentation using parametric graph partitioning. In *ICCV*, pages 3155–3163, 2015. 2

[54] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 6230–6239, 2017. 4