Noname manuscript No.
(will be inserted by the editor)

VOSTR: Video Object Segmentation via Transferable Representations

Yi-Wen Chen[1,2] · **Yi-Hsuan Tsai**[3] · **Yen-Yu Lin**[2,4] · **Ming-Hsuan Yang**[1,5]

Received: date / Accepted: date

Abstract In order to learn video object segmentation models, conventional methods require a large amount of pixel-wise ground truth annotations. However, collecting such supervised data is time-consuming and labor-intensive. In this paper, we exploit existing annotations in source images and transfer such visual information to segment videos with unseen object categories. Without using any annotations in the target video, we propose a method to jointly mine useful segments and learn feature representations that better adapt to the target frames. The entire process is decomposed into three tasks: 1) refining the responses with fully-connected CRFs, 2) solving a submodular function for selecting object-like segments, and 3) learning a CNN model with a transferable module for adapting seen categories in the source domain to the unseen target video. We present an iterative update scheme between three tasks to self-learn the final solution for object segmentation. Experimental results on numerous benchmark datasets demonstrate that the proposed method performs favorably against the state-of-the-art algorithms.

Yi-Wen Chen
E-mail: ychen319@ucmerced.edu

Yi-Hsuan Tsai
E-mail: ytsai@nec-labs.com

Yen-Yu Lin
E-mail: lin@cs.nctu.edu.tw

Ming-Hsuan Yang
E-mail: mhyang@ucmerced.edu

[1]University of California, Merced, CA, USA
[2]Academia Sinica, Taipei, Taiwan
[3]NEC Laboratories America, CA, USA
[4]National Chiao Tung University, Hsinchu, Taiwan
[5]Google, CA, USA

Keywords Video Object Segmentation · Transfer Learning · Weakly-supervised Learning.

1 Introduction

Nowadays, video data can be easily accessed and hence visual analytics has become an important task in computer vision. In this line of research, video object segmentation is one of the effective ways to understand visual contents and can facilitate various applications, such as video editing, content retrieval, and object identification. While conventional methods rely on the supervised learning strategy to effectively localize and segment objects in videos, collecting such ground truth annotations is expensive and cannot scale well to a large number of object categories in videos.

Recently, weakly-supervised methods for video object segmentation (Tsai et al. 2016b; Zhang et al. 2017; Saleh et al. 2017; Yan et al. 2017) have been developed to relax the need for annotations where only class-level labels are required. These approaches have significantly reduced the labor-intensive step of collecting pixel-wise annotated training data on target categories. However, these target categories are pre-defined. Thus, the trained model cannot be directly applied to videos with unseen categories, i.e., object categories that are not covered by training data. Annotating additional categories during the phase of testing would require more efforts and is less practical. In this paper, we propose an algorithm to reduce efforts in annotating both pixel-level and class-level ground truths, in order to segment objects of unseen categories in videos.

To this end, we make use of existing pixel-level annotations in images from the PASCAL VOC dataset (Everingham et al. 2010) with pre-defined categories,

and design a framework that transfers this knowledge to videos with unseen object categories. That is, the proposed method is able to learn useful representations for segmentation from the data in the image domain and adapt these representations to segment objects in videos regardless of whether the object categories are covered by the PASCAL VOC dataset. Thus, while performing video object segmentation, our algorithm does not require annotations in any forms, such as pixel-level or class-level ground truths.

We formulate the video object segmentation problem for unseen categories as a joint objective of refining and mining useful segments from videos while learning transferable knowledge from image representations. Since annotations are not provided in videos in our setting, one can rely on the response output from a convolutional neural network (CNN) to segment the object. However, these responses are often over-smoothed due to multiple max-pooling and down-sampling processes. Thus, the responses need refinement in order to recover the high-resolution details for better object localization and segmentation. To this end, we first refine the responses using fully-connected conditional random fields (CRFs) (Krähenbühl and Koltun 2011). Second, we design an energy function to discover object-like segments from the refined responses in videos based on the feature representations learned from the image data. We then utilize these discovered segments to update feature representations in the CNN model, while a transferable module is developed to learn the relationships between multiple seen categories in images and the unseen category in a video. By jointly considering both energy functions for refining and mining better segments while learning transferable representations, we develop an iterative optimization method to self-guided video object segmentation. We also note that the proposed framework is flexible as we can input either weakly-labeled or unlabeled videos.

To validate the proposed method, we conduct experiments on benchmark datasets for video object segmentation. First, we evaluate our method on the DAVIS 2016 dataset (Perazzi et al. 2016) where some object categories are not covered by the PASCAL VOC dataset. Based on this setting, we compare our method with the state-of-the-art methods for object segmentation via transfer learning, including those using the NLP-based GloVe embedding (Pennington et al. 2014) and a decoupled network (Hong et al. 2016). In addition, we demonstrate the effectiveness of the proposed iterative self-learning strategy by comparing the results with and without using this strategy. Second, we adopt the weakly-supervised setting on the YouTube-Objects dataset (Prest et al. 2012) and show that the proposed method performs favorably against the state-of-the-art algorithms in both visual quality and accuracy. Third, we further evaluate the proposed algorithm on unseen object segmentation, and apply it to the SegTrack v2 dataset (Li et al. 2013), which contains numerous object categories that do not appear in the PASCAL VOC dataset.

The contributions of this work are summarized as follows. First, we propose a framework for object segmentation in unlabeled videos through a self-guided learning method. Second, we develop a joint formulation to refine and mine useful segments while adapting the feature representations to the target videos. Third, we design a CNN module that can transfer knowledge from multiple seen categories in images to an arbitrary, i.e., either seen or unseen, object category in a video.

We note that this paper is an extension of our previous work (Chen et al. 2018b), which is referred to as VOSTR$_a$. We make additional contributions in our method for video object segmentation via transferable representations (VOSTR). First, we leverage the fully-connected CRFs to refine the responses and obtain proposals of higher quality, which help the CNN model learn better feature representations. Second, we integrate this refinement process into the original objective, in which a joint formulation is proposed and is optimized. Third, we provide comprehensive experimental results and analysis on one additional dataset, i.e., SegTrack v2, to demonstrate the effectiveness of the proposed method.

2 Related Work

Video Object Segmentation. Video object segmentation aims to separate foreground objects from the background. Conventional methods utilize object proposals (Lee et al. 2011; Perazzi et al. 2015; Koh and Kim 2017) or graphical models (Tsai et al. 2016a; Märki et al. 2016), while recent approaches focus on learning CNN models from image sequences with frame-by-frame pixel-level ground truth annotations to achieve the state-of-the-art performance (Cheng et al. 2017; Tokmakov et al. 2017b; Jain et al. 2017). For CNN-based methods, motion cues (Li et al. 2018) are usually used to effectively localize objects. Jain et al. (2017) utilize a two-stream network by jointly considering appearance and motion information. The SegFlow method (Cheng et al. 2017) further shows that jointly learning segmentation and optical flow in videos enhances both performance. Note that, these approaches usually require pre-training on videos with frame-by-frame pixel-level annotations (Cheng et al. 2017; Tokmakov et al. 2017b) or bound-

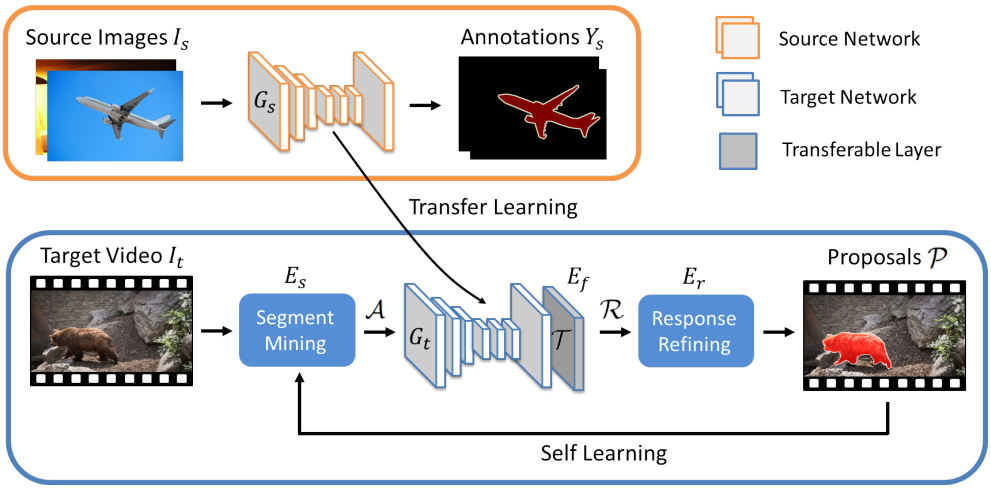

Fig. 1 Overview of the proposed framework. Given a set of source images $\mathcal{I}_s$ with semantic segmentation annotations Y_s, we first train a source CNN model G_s. To predict object segmentation on a target video $\mathcal{I}_t$ without knowing any annotations, we initialize the target network G_t from the parameters in G_s and perform adaptation via a transferable layer $\mathcal{T}$. A three-step self-learning scheme is performed. We minimize the function E_r to generate refined proposals $\mathcal{P}$ from responses $\mathcal{R}$, optimize the function E_s for selecting object-like segments $\mathcal{A}$ from proposals $\mathcal{P}$, and adapt feature representations in the CNN model via optimizing E_f. The entire self-learning process is performed via iteratively updating the three energy functions to obtain the final segmentation results.

ing box ground truths (Jain et al. 2017) to obtain better foreground segmentation.

Another line of research is to fine-tune the model based on the object mask in the first frame (Caelles et al. 2017; Khoreva et al. 2017) and significantly improves the segmentation quality. More recently, Cheng et al. (2018) adopt a part-based tracking method to deal with challenging factors such as deformation and occlusion. Oh et al. (2018) propose a siamese network to take advantage of mask propagation and object detection. Other methods such as pixel-wise metric learning (Chen et al. 2018a) or network modulation (Yang et al. 2018) are proposed to facilitate the segmentation runtime performance. In contrast to using the annotation from the first frame, the proposed algorithm uses only a smaller number of existing annotations from the image dataset and transfers the feature representations to unlabeled videos for object segmentation. In addition, our method is flexible for the weakly-supervised learning setting, which cannot be achieved by the above approaches.

Weakly-supervised Video Object Segmentation. To reduce the need of pixel-level annotations, weakly-supervised methods (Shi et al. 2017) have been developed to facilitate the segmentation process, where only class-level labels are required in videos. Numerous approaches are proposed to collect useful semantic segments by training segment-based classifiers (Tang et al. 2013) or ranking supervoxels (Zhong et al. 2016). However, these methods rely on the quality of the generated segment proposals and may produce inaccurate results when taking low-quality segments as the input. Zhang et al. (2015b) propose to utilize object detectors together with object proposals to refine segmentation results in videos. Furthermore, Tsai et al. (2016b) develop a co-segmentation framework by linking object tracklets from all the videos and improve the result. Recently, the SPFTN method (Zhang et al. 2017) utilizes a self-paced learning scheme to fine-tune segmentation results from object proposals. Different from the above algorithms that only target on a pre-defined set of categories, our approach further extends this setting to segmenting unlabeled videos where unseen object categories are present.

Domain Adaptation and Transfer Learning. Using cross-domain data for unsupervised learning has been explored in domain adaptation (Saenko et al. 2010; Gopalan et al. 2011; Patricia and Caputo 2014; Ganin and Lempitsky 2015; Luo et al. 2017; Tsai et al. 2018). While domain adaptation methods make the assumption that the same categories are shared across different domains, transfer learning approaches focus on transferring knowledge between categories. Numerous transfer learning methods have been developed for object classification (Tommasi et al. 2014) and detection (Lim et al. 2011; Hoffman et al. 2014). Similar efforts have been made for object segmentation. Hong et al. (2016) propose a weakly-supervised semantic segmentation method by exploiting pixel-level annotations from different categories. Recently, Hu et al. (2018) design a weighted transform function to transfer knowledge between the detected bounding boxes and instance segments. In this work, we share the similar motivation

with Hong et al. (2016) but remove the assumption of weak supervisions. Luo et al. (2017) tackle the problem of domain adaptation for image classification with few annotations available in the target domain. On the contrary, we tackle the video object segmentation task, where there are no available labels provided in the target video. To the best of our knowledge, except for our conference version (Chen et al. 2018b), this work is the first attempt for video object segmentation by transferring knowledge from annotated images to an unlabeled video of an unseen category.

3 Algorithmic Overview

This section describes an overview of the proposed framework and the developed objective function.

3.1 Overview of the Proposed Framework

We first describe the problem context of this work. Given a number of source images $\mathcal{I}_s = \{I_s^1, ..., I_s^N\}$ with pixel-level semantic segmentation annotations $Y_s = \{y_s^1, ..., y_s^N\}$ and the target sequence $\mathcal{I}_t = \{I_t^1, ..., I_t^M\}$ without any labels, our objective is to develop a self-guided learning algorithm that segments the object in $\mathcal{I}_t$ by transferring knowledge from $\mathcal{I}_s$ to $\mathcal{I}_t$. In this work, the object category in $\mathcal{I}_t$ is allowed to be arbitrary. It can be either covered by or different from those in $\mathcal{I}_s$.

Fig. 1 illustrates the proposed framework for segmenting the object in video $\mathcal{I}_t$. First, we train a source CNN model G_s using $\mathcal{I}_s$ and Y_s as the input and the desired output, respectively. Second, we initialize the target network G_t from the parameters in G_s. Since $\mathcal{I}_s$ and $\mathcal{I}_t$ may not share common object categories, we design a transferable layer $\mathcal{T}$ that enables cross-category knowledge transfer, and append it to the target network. The initialization of the transferable layer $\mathcal{T}$ will be discussed later. With the input video $\mathcal{I}_t$ of an unseen object category, we aim at adapting the target network G_t so that the object in $\mathcal{I}_t$ can be better segmented.

To this end, we present a self-learning procedure with three key components, namely 1) a fully-connected CRF model for refining responses, 2) a ranking module for mining segment proposals, and 3) a CNN model for learning the transferable feature representations. The three components work sequentially and iteratively to discover the object in $\mathcal{I}_t$. After the target network G_t is applied to the input video $\mathcal{I}_t$ to generate response outputs, we first use fully-connected CRFs to refine the responses $\mathcal{R}$ produced by G_t, and compile a set of segment proposals $\mathcal{P}$ on the target video $\mathcal{I}_t$. Second, to select a set of more object-like proposals $\mathcal{A}$ among $\mathcal{P}$, we develop an energy function to re-rank these proposals based on their objectness scores and mutual relationships. Third, by treating the selected proposals $\mathcal{A}$ as the pseudo ground truth, we update the transferable feature representations to better segment the object in the video. The entire process can be formulated as a joint and iterative optimization problem with the objective function described in the following.

Table 1 Notations in the proposed algorithm.

Notation	Representative
G_s	Source network
G_t	Target network
$\mathcal{T}$	Transferable layer
$\mathcal{C}$	Fully-connected CRFs
$\mathcal{I}_s$	Source images
$\mathcal{I}_t$	Target sequence
Y_s	Semantic segmentation annotations of $\mathcal{I}_s$
$\mathcal{R}$	Responses produced by G_t
$\mathcal{P}$	Proposals generated from $\mathcal{R}$
$\mathcal{A}$	Segments selected from $\mathcal{P}$
E_r	Energy for refining $\mathcal{R}$
E_s	Energy for selecting $\mathcal{A}$ from $\mathcal{P}$
E_f	Energy for optimizing G_t
θ	Parameters of G_t

3.2 Objective Function

Our goal is to find high-quality segment proposals $\mathcal{P}$ from the target video $\mathcal{I}_t$ that can guide the network to learn feature representations $\mathcal{F}$ for better segmenting the given video $\mathcal{I}_t$. We carry out this task by jointly optimizing an energy function E that accounts for segment proposals $\mathcal{P}$ and features $\mathcal{F}$:

$$\max_{\mathcal{A},\theta} E(\mathcal{I}_t, \mathcal{P}, \mathcal{F}; \mathcal{A}, \theta) = \max_{\mathcal{A},\theta} E_r(\mathcal{I}_t, \mathcal{R}; \mathcal{P}) + E_s(\mathcal{P}, \mathcal{F}; \mathcal{A}) + E_f(\mathcal{I}_t, \mathcal{A}; \theta), \quad (1)$$

where E_r is the energy for refining the responses $\mathcal{R}$ yielded by the CNN model G_t via using fully-connected CRFs, E_s is the energy for selecting a set of high-quality segments $\mathcal{A}$ from the proposals $\mathcal{P}$ based on the features $\mathcal{F}$, and θ is the parameters of the CNN model that aims to optimize E_f and learn the feature representations $\mathcal{F}$ from the selected proposals $\mathcal{A}$. After the optimization process, we obtain the final segmentation results, which is the network output $\mathcal{P}$. Note that, here we do not include the responses $\mathcal{R}$ as the input in E, since $\mathcal{R}$ is a intermediate product of the optimization process. We summarize the notations in the proposed algorithm in Table 1. Details of each energy function and the optimization process are described in the following section.

4 Transferring Knowledge for Segmentation

In this section, we describe the proposed energy functions for refining responses, mining segments, and learning the transferable feature representations, respectively. Response refining is carried out by using fully-connected CRFs, segment mining is formulated as a submodular optimization problem, and transferable feature learning is accomplished through a CNN model with a transferable module. After introducing the energy functions for the three tasks, we present an iterative optimization scheme to jointly optimize the objective in (1).

4.1 Refining Responses

Given a target video $\mathcal{I}_t$, we can perform frame-by-frame object segmentation by using the CNN model G_t with the proposed transferable layer $\mathcal{T}$. However, the deep CNN model G_t with multiple max-pooling and downsampling layers typically yields over-smoothed responses for segmentation. To refine its quality for localization and segmentation, we apply fully-connected CRFs to the responses produced by the CNN model, so that the high-resolution details for segmentation can be recovered, which can in turn help the other components.

Refinement with Fully-connected CRFs. To recover the detailed local structure, we utilize an energy function based on the fully-connected CRFs:

$$E_r(\mathcal{I}_t, \mathcal{R}; \mathcal{P}) = -\sum_i \delta_i(x_i) - \sum_{ij} \delta_{ij}(x_i, x_j), \tag{2}$$

where $\delta_i(x_i) = -\log r(x_i)$ is the unary potential, and $r(x_i)$ is the label assignment probability at pixel i, which is predicted by the CNN model, i.e., obtained from $\mathcal{R}$. $\delta_{ij}(x_i, x_j)$ is the pairwise potential for a pixel pair (i, j), which is formulated as:

$$\begin{aligned}&\delta_{ij}(x_i, x_j)\\ &= \mu(x_i, x_j)\left[w_1 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\alpha^2} - \frac{\|I_i - I_j\|^2}{2\sigma_\beta^2}\right)\right.\\ &\qquad \left. + w_2 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\gamma^2}\right)\right],\end{aligned} \tag{3}$$

where $\mu(x_i, x_j) = 1$ if $y_i \neq y_j$, and zero otherwise, which means that only pixels with distinct labels y are penalized. The remaining function contains two Gaussian kernels in different feature spaces. The first kernel forces pixels in neighboring positions (denoted as p) and with similar RGB colors (denoted as I) to have the same label, while the second kernel only considers pixel positions. The constants σ_α, σ_β, and σ_γ are hyper parameters introduced to control the scales of the Gaussian kernels. The constants w_1 and w_2 are the weights of the two Gaussian kernels.

4.2 Mining Segment Proposals

After refining the responses of the segmentation result, there is still another defect about the generated segments due to the unsupervised nature of this task. Namely, some generated segments do not well cover objects. Thus, we aim to select high-quality segments and eliminate noisy ones from the generated object segments. The major challenge of this task lies in the lack of ground truth annotations in the target video, and thus we cannot train a classifier to guide the selection process.

Inspired by the co-segmentation method (Tsai et al. 2016b), we observe that high-quality segments typically have higher mutual relationships. As a result, we gather all the predicted segments from the target video and construct a graph to link each segment. We then formulate segment mining as a submodular optimization problem, aiming to select a subset of more object-like segments that share higher similarities.

Graph Construction on Segments. We first feed the target video $\mathcal{I}_t$ into the CNN model frame-by-frame and obtain a set of segment proposals $\mathcal{P}$, where each proposal is a connected-component in the predicted segmentation of the video $\mathcal{I}_t$. Then we construct a fully-connected graph $G = (\mathcal{V}, \mathcal{E})$ on the set $\mathcal{P}$, where each vertex $v \in \mathcal{V}$ is a segment, and each edge $e \in \mathcal{E}$ models the pairwise relationship between two segments. Our goal is to find a subset $\mathcal{A}$ of $\mathcal{P}$ that contains proposals with higher object-like confidence.

Submodular Function. Since there is no ground truth available, we design a submodular function for mining the segments belonging to the object by leveraging the following three properties: 1) the selected segments should be similar to each other since they belong to the same object; 2) the selected segments have higher responses in the output of the CNN model; and 3) the selected segments usually move differently from the background area in the video.

We formulate the submodular function for selecting object-like segments by a facility location term $\mathcal{H}$ (Lazic et al. 2009) and a unary term $\mathcal{U}$. The former enhances the similarity between the selected segments, while the latter encourages the high probability of each selected segment being a true object. Both terms are defined based on the segment proposals $\mathcal{P}$ and the adopted feature representation $\mathcal{F}$.

Specifically, we define the facility location term as

$$\mathcal{H}(\mathcal{P},\mathcal{F};\mathcal{A}) = \sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{V}} W(v_i,v_j) - \sum_{i\in\mathcal{A}}\phi_i, \tag{4}$$

where W denotes the pairwise relationship between a potential facility v_i and a vertex v_j, while ϕ_i is the cost to open a facility, which is fixed to a constant α. We define W as the similarity between two segments in order to encourage the submodular function to choose a facility v_i that is similar to v_j. To estimate this similarity, we represent each segment as a feature vector and compute the inner product of the two vectors. To form the feature vector for each segment, we draw feature maps from the CNN model (**conv1** to **conv5**) and perform the global average pooling on each segment. It is the adopted feature representation $\mathcal{F}$ in this work.

In addition to the facility location term, we employ a unary term to evaluate the quality of segments

$$\mathcal{U}(\mathcal{P},\mathcal{F};\mathcal{A}) = \lambda_o \sum_{i\in\mathcal{A}} \Phi_o(i) + \lambda_m \sum_{i\in\mathcal{A}} \Phi_m(i), \tag{5}$$

where $\Phi_o(i)$ is the objectness score that measures the probability of segment i belonging to the region of the object, and $\Phi_m(i)$ is the motion score that estimates the motion difference between segment i and the background region. λ_o and λ_m are the weights for the two terms, respectively. The objectness score $\Phi_o(i)$ is calculated by averaging the probability map of the CNN output layer on all the pixels within the segment. For the motion score $\Phi_m(i)$, we first compute the optical flow (Liu et al. 2009) for two consecutive frames, and then we utilize the minimum barrier distance (Strand et al. 2013; Zhang et al. 2015a) to convert the optical flow into a saliency map, where larger distances represent larger motion difference with respect to the background region.

Formulation for Segment Mining. Our goal is to find a subset $\mathcal{A}$ of $\mathcal{P}$ containing segments that are similar to each other and have higher object-like confidence. Therefore, we combine the facility location term $\mathcal{H}$ and the unary term $\mathcal{U}$ to yield the energy E_s in (1) as:

$$E_s(\mathcal{P},\mathcal{F};\mathcal{A}) = \mathcal{H}(\mathcal{P},\mathcal{F};\mathcal{A}) + \mathcal{U}(\mathcal{P},\mathcal{F};\mathcal{A}). \tag{6}$$

We also note that the linear combination of two non-negative terms preserves the submodularity (Zhu et al. 2014).

Discussions. In this work, we are more interested in video segmentation than image segmentation as pixel-level annotations in videos are more difficult to obtain, especially when there are unseen objects in videos. To this end, we take advantages of continuous frames in the video, in which nearby frames share high similarities in appearance, whereas separate images would not have this property. This provides us with a useful cue to perform segment mining in videos via constructing the submodular objective. Although the general objective and pipeline introduced in this work are also applicable to image segmentation, it would require other ways to effectively mine useful segments, which is outside the scope of this paper.

4.3 Learning Transferable Feature Representations

Given the selected set of object-like segment proposals, the ensuing task is to learn better feature representations based on these segments. To this end, we propose to use a CNN model fine-tuned on these segments via a self-learning scheme. Since our target video may have a different set of object categories from those in the source domain, we further develop a transfer learning method where a transferable layer is augmented to the CNN model. With the proposed layer, our network is able to transfer knowledge from seen categories to the unseen category, without the need of any supervision in the target video.

Inspired by the observation that an unseen object category can be represented by a series of seen objects (Rochan and Wang 2015), we develop a transferable layer that approximates an unseen category as a linear combination of seen ones in terms of the output feature maps. In the following, we first present our CNN objective for learning the feature representations based on the selected segment proposals. Then we introduce the details of the proposed layer for transferring knowledge from the source domain to the target one.

Objective Function. Given the target video $\mathcal{I}_t$ and the selected segment proposals $\mathcal{A}$ as described in Section 4.2, we use $\mathcal{A}$ as our pseudo ground truths and optimize the target network G_t with parameters θ_g to obtain better feature representations that match the target video. Specifically, we define the energy function E_f in (1) as:

$$E_f(\mathcal{I}_t,\mathcal{A};\theta_g,\theta_{\mathcal{T}}) = -\mathcal{L}(\mathcal{T}(G_t(\mathcal{I}_t)),\mathcal{A}), \tag{7}$$

where $\theta_{\mathcal{T}}$ is the parameters of the transferable layer $\mathcal{T}$ and $\mathcal{L}$ is the cross-entropy function to measure the loss between the network prediction $\mathcal{T}(G_t(\mathcal{I}_t))$ and the pseudo ground truth $\mathcal{A}$. Note that we use the minus sign for the loss function $\mathcal{L}$ to match the maximization formulation in (1).

Learning Transferable Knowledge. Suppose there are C_s categories in the source domain, we aim to transfer a source network G_s pre-trained on the source images $\mathcal{I}_s$ to the target video. To achieve this, we first

initialize the target network G_t using the parameters in G_s. Given the target video $\mathcal{I}_t$, we can generate frame-wise feature maps $R = G_t(\mathcal{I}_t) = \{r_c\}_{c=1}^{C_s}$ through the network with C_s channels, where r_c is the output map of source category c. Since the target category is unknown, we then approximate the desired output map, r, for the unseen category as a linear combination of these seen categories through the proposed transferable layer $\mathcal{T}$:

$$r = \mathcal{T}(R) = \sum_{c=1}^{C_s} w_c \, r_c, \tag{8}$$

where w_c is the weight of the seen category c. Specifically, the proposed transferable layer $\mathcal{T}$ can be performed via a 1×1 convolutional layer with C_s channels, in which the parameter of channel c in $\theta_{\mathcal{T}}$ corresponds to w_c.

Since w_c is not supervised by any annotations from the target video, the initialization of w_c is critical for obtaining a better combination of feature maps from the seen categories. Thus, we initialize w_c by calculating the similarity between each source category c and the target video. For each image in the source and target domains, we extract its feature maps from the **fc7** layer of the network and compute a 4096-dimensional feature vector on the predicted segment via global average pooling. By representing each image as a feature vector, we measure the similarity score between source and target images by their inner product. Finally, the initialized weight w_c^{init} for the category c can be obtained by averaging largest scores on each target frame with respect to the source images:

$$w_c^{init} = \frac{1}{|\mathcal{I}_t|} \sum_{i=1}^{|\mathcal{I}_t|} \max_j \langle \mathcal{F}_t^i, \mathcal{F}_{s,c}^j \rangle, \tag{9}$$

where $|\mathcal{I}_t|$ is the number of frames in the target video, $\mathcal{F}_t^i \in \mathbb{R}^{4096}$ is the feature vector of the ith frame of $\mathcal{I}_t$, and $\mathcal{F}_{s,c}^j \in \mathbb{R}^{4096}$ is the feature vector of the jth image of source category c.

Discussions. In the proposed method, we do not make any assumption about the number of segments in each frame during learning a set of weights for linear combination in (8). Thus, our method can predict multiple "instances" (e.g., Fig. 6) of one object category, in which these segments share the same weights for linear combination and tend to be similar to each other. However, the current method may not predict multiple "objects" with diverse appearance, unless we introduce other sets of linear combinations.

4.4 Joint Formulation and Model Training

Based on the formulations for response refinement in (2), segment mining in (6), and transferable feature representation learning in (7), we jointly solve the three objectives, i.e., E_r, E_s, and E_f in (1), via

$$\begin{aligned} &\max_{\mathcal{A},\theta} E(\mathcal{I}_t, \mathcal{P}, \mathcal{F}; \mathcal{A}, \theta) \\ &= \max_{\mathcal{A},\theta} E_r(\mathcal{I}_t, \mathcal{R}; \mathcal{P}) + E_s(\mathcal{P}, \mathcal{F}; \mathcal{A}) + E_f(\mathcal{I}_t, \mathcal{A}; \theta) \\ &= \max_{\mathcal{A},\theta_g,\theta_{\mathcal{T}}} -\sum_i \delta_i(x_i) - \sum_{ij} \delta_{ij}(x_i, x_j) \\ &\quad + [\mathcal{H}(\mathcal{P}, \mathcal{F}; \mathcal{A}) + \mathcal{U}(\mathcal{P}, \mathcal{F}; \mathcal{A})] - \mathcal{L}(\mathcal{T}(G_t(\mathcal{I}_t)), \mathcal{A}). \end{aligned} \tag{10}$$

We decompose the optimization of (10) into three sub-problems: 1) utilizing fully-connected CRFs for response refinement to yield the proposal set $\mathcal{P}$, 2) solving the submodular function for segment mining to generate pseudo ground truth $\mathcal{A}$, and 3) updating the CNN model θ_g and $\theta_{\mathcal{T}}$ for transferable feature representation learning. We adopt an iterative procedure to alternately optimize the three sub-problems. The initialization strategy and the optimization of the three sub-problems are described below.

Initialization. We first pre-train the source network G_s on the PASCAL VOC training set (Everingham et al. 2010) containing 20 object categories. We then initialize the target network G_t from parameters in G_s and the transferable layer $\mathcal{T}$ as described in Section 4.3. To obtain an initial set of segment proposals, we forward the target video $\mathcal{I}_t$ to the target model G_t with $\mathcal{T}$ and generate responses $\mathcal{R}$.

Optimizing E_r by Fixing E_s and E_f. To refine the responses produced by the CNN which are over-smoothed due to the max-pooling and downsampling in the CNN model, we optimize E_r following Krähenbühl and Koltun (2011) to provide higher quality proposals, in which we denote this process as $\mathcal{C}$. Note that, here we fix the parameters of CRFs and infer refined proposals that provide the minimum of energy function $-E_r$.

Optimizing E_s by Fixing E_r, E_f. After generating the refined proposals $\mathcal{P}$, we fix the network parameters and optimize $\mathcal{A}$ via E_s in (10). We adopt a greedy algorithm similar to Tsai et al. (2016b). Starting from an empty set of $\mathcal{A}$, we add an initial element $a \in \mathcal{V} \backslash \mathcal{A}$ to $\mathcal{A}$ that gives the largest energy gain. The process is then repeated and stops when one of the following conditions is satisfied: 1) the number of selected proposals reaches a threshold, i.e., $|\mathcal{A}| > N_{\mathcal{A}}$, and 2) the ratio of the energy gain between two rounds is below a threshold, i.e., $\mathcal{D}(\mathcal{A}^i) < \beta \cdot \mathcal{D}(\mathcal{A}^{i-1})$, where $\mathcal{D}(\mathcal{A}^i)$ stands for the

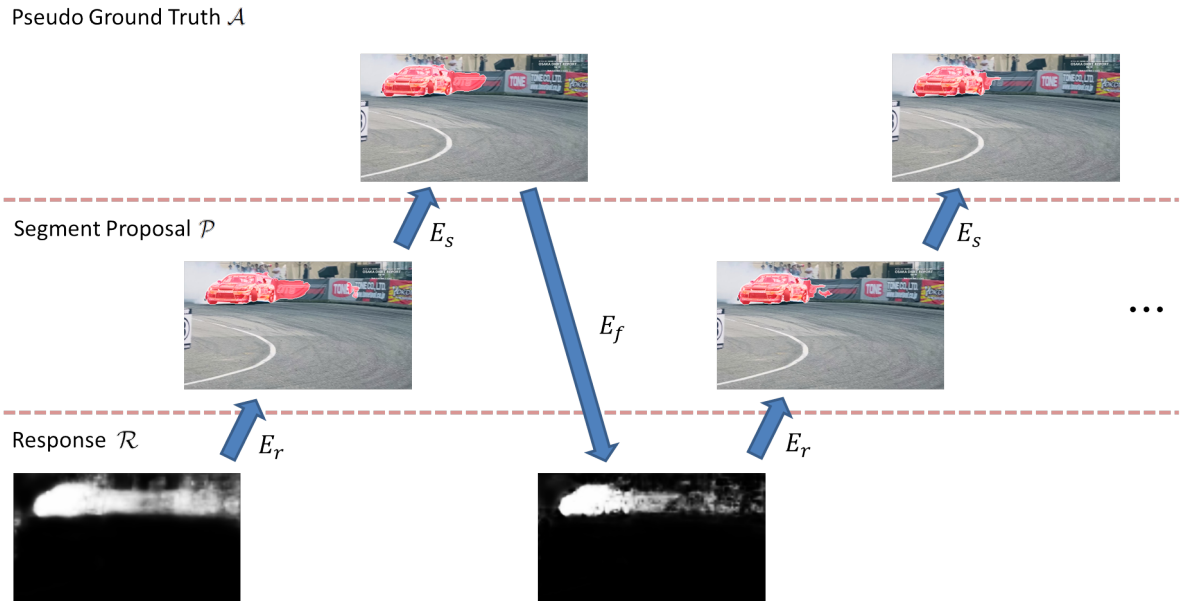

Fig. 2 Sample results of iteratively optimizing E_r, E_s, and E_f. Starting from the initial response $\mathcal{R}$, we generate the proposals $\mathcal{P}$ via E_r. Then we solve E_s to obtain object-like segments $\mathcal{A}$ as our pseudo ground truth to optimize E_f. By iteratively optimizing the three energy functions, our algorithm gradually improves the quality of $\mathcal{R}$, $\mathcal{P}$ and $\mathcal{A}$ to obtain the final segmentation results.

Algorithm 1 Unseen Object Segmentation

Source Image: $\mathcal{I}_s$, Y_s
Target Video: $\mathcal{I}_t$
Initialization: pre-trained G_s on source inputs, $G_t \leftarrow G_s$, w_c^{init} via (9)
$(\mathcal{R}, \mathcal{F}) \leftarrow \mathcal{T}(G_t(\mathcal{I}_t))$
$\mathcal{P} \leftarrow \mathcal{C}(\mathcal{I}_t, \mathcal{R})$ via (2)
while $\mathcal{P}$ not converged **do**
  $\mathcal{A}^0 \leftarrow \emptyset$, $i \leftarrow 1$
  loop
    $a^* = \arg\max_{\{\mathcal{A}^i \in \mathcal{V}\}} E_s(\mathcal{P}, \mathcal{F}; \mathcal{A}^i)$, where $\mathcal{A}^i \leftarrow \mathcal{A}^{i-1} \cup a$,
    $a \in \mathcal{V} \backslash \mathcal{A}$
    if $|\mathcal{A}| > N_{\mathcal{A}}$ or $\mathcal{D}(\mathcal{A}^i) < \beta \cdot \mathcal{D}(\mathcal{A}^{i-1})$ when $i \geq 2$ **then**
      break
    end if
    $\mathcal{A}^i \leftarrow \mathcal{A}^{i-1} \cup a^*$, $i \leftarrow i + 1$
  end loop
  $\mathcal{A} \leftarrow \mathcal{A}^i$
  Optimize E_f: $(\theta_g, \theta_{\mathcal{T}}) \leftarrow \min \mathcal{L}(\mathcal{T}(G_t(\mathcal{I}_t)), \mathcal{A})$
  $(\mathcal{R}, \mathcal{F}) \leftarrow \mathcal{T}(G_t(\mathcal{I}_t))$
  $\mathcal{P} \leftarrow \mathcal{C}(\mathcal{I}_t, \mathcal{R})$ via (2)
end while
Output: object segmentation $\mathcal{P}$ of $\mathcal{I}_t$

energy gain, i.e., difference of E_s between two rounds during the optimization process, and β is the ratio.

Optimizing E_f by Fixing E_r and E_s. Once obtaining $\mathcal{A}$ as the pseudo ground truths, we fix $\mathcal{A}$ and optimize the network with the transferable layer, i.e., θ_g and $\theta_{\mathcal{T}}$, in E_f of (10). We alter the problem to a task that minimizes the network loss $\mathcal{L}$ in an end-to-end fashion, jointly for θ_g and $\theta_{\mathcal{T}}$ using the SGD method.

Iterative Optimization. To obtain the final $\mathcal{A}$, θ_g, and $\theta_{\mathcal{T}}$, instead of directly solving (10), we solve it via an iterative updating scheme among E_r, E_s, and E_f until convergence. To determine the convergence, we set the conditions: 1) The IoU of segmentation between two iterations is larger than 90%. Namely, segmentation almost retains the same one. 2) The maximum number of iterations is 3. Empirically, we find that our method on most sequences converges in three iterations.

Our algorithm contains three components: proposal mining via submodular optimization, proposal refinement via CRFs, and pseudo ground truth training via CNNs. The first and third steps are sub-optimal, while the second step has an optimal solution. Therefore, the energy of each term could be optimized individually during the iterative optimization process. Fig. 2 shows an example of gradually updating $\mathcal{R}$, $\mathcal{P}$ and $\mathcal{A}$ via iteratively optimizing E_r, E_s, and E_f. The overall optimization process is summarized in Algorithm 1.

Discussions. Different from previous methods that use the fully-connected CRFs as post-processing to improve the final results, our method adopts the fully-connected CRFs as one component during the training stage. That is, the energy function in CRFs receives the response from proposals to generate refined ones, which in turn serve as better pseudo ground truth to help the CNN model learn better transferable feature representations. In addition, we integrate this energy function into the final objective and perform iterative updating to achieve final results.

Table 2 Training and testing time of our method on the DAVIS dataset.

Stage	Time (second)
Motion prior computing (per frame pair)	0.01
Feature extraction (per frame)	1.72
Response refining via (2) (per frame)	0.78
Segment mining via (6) (per frame)	0.01
CNN model training via (7) (per frame)	7.31
Inference (per frame)	0.01

5 Experimental Results

In this section, we first present implementation details of the proposed method, and then we show experimental results on numerous benchmark datasets. In addition, ablation studies for evaluating the effects of the proposed components in the algorithm are conducted. The source code and trained models will be made available to the public.

5.1 Implementation Details

In the submodular function for segment mining, we set $\lambda_o = 20$ and $\lambda_m = 35$ for the unary term in (5). During the submodular optimization in (6), we use $N_{\mathcal{A}} = 0.8 \cdot |\mathcal{P}|$ and $\beta = 0.8$. All the parameters are fixed in all the experiments. For training the CNN model in (7), we employ two types of fully convolutional networks (FCNs) (Long et al. 2015) including the VGG-16 (Simonyan and Zisserman 2014) and ResNet-101 (He et al. 2016) architectures for both the source and target networks using the Caffe library. The learning rate, momentum, and batch size are set as 10^{-14}, 0.99, and 1, respectively. To further refine the final segmentation results, we apply additional CRFs to the responses produced by our fully-trained CNN model.

5.2 Training Time and Runtime Analysis

The training and testing (inference) time of each component of our method is shown in Table 2. In the proposed method, we pre-train the source network on the image dataset and use its parameters to initialize the target network. For each new video, we train the target network via the proposed iterative optimization in Algorithm 1 so that the target network can be applied to segment the unseen object in the testing video. The first five rows of runtime in Table 2 are for training on the new video, while the inference time is for applying the trained model to each frame of the video.

All the timings are measured on a machine with an Intel Xeon 2.5GHz processor and an NVIDIA GTX 1080 Ti graphics card with 11GB memory. We compute the optical flow (Liu 2009) and utilize the minimum barrier distance (Zhang et al. 2015a) to generate motion prior using MATLAB. In the proposed algorithm,including feature extraction, response refining, segment mining, and CNN model training are implemented by using Python and the Caffe library on the graphics card. The CNN model is fine-tuned for $2,000$ iterations. Note that we report the runtime averaged over all the frames.

Table 3 IoU of the selected segments with different weights of the motion term on the DAVIS dataset.

λ_m	0	5	15	25	35	45
Avg. IoU	57.2	57.4	60.5	60.6	61.0	60.3

Table 4 IoU of the selected segments with and without CRFs on the DAVIS dataset.

	w/o CRFs	w/ CRFs
Avg. IoU	61.0	63.5

Table 5 IoU of the final results with different learning rates on the DAVIS dataset.

lr	10^{-15}	10^{-14}	10^{-13}
Avg. IoU	67.9	68.4	68.2

Table 6 IoU of the final results with different values of β on the DAVIS dataset.

β	0.6	0.7	0.8	0.9
Avg. IoU	68.0	68.1	68.4	68.3

5.3 DAVIS Dataset

We first conduct experiments on the DAVIS 2016 benchmark dataset (Perazzi et al. 2016). Since our goal is to transfer the knowledge from seen categories in images to unseen objects in the video, we manually select all the videos with object categories that are different from the 20 categories in the PASCAL VOC dataset. In the following, we first conduct ablation studies and experiments to validate the proposed method. Second, we show that our algorithm can be applied under various settings on the entire set of the DAVIS 2016 dataset.

Impact of the Motion Terms. One critical component of our framework is to mine useful segments for the further CNN model training step. In the submodular function of (5), we incorporate a motion term that accounts for object movements in the video. To validate its effectiveness, we fix the weight $\lambda_o = 20$ for the appearance and vary the weight λ_m for the motion term.

Table 7 Learned weights of the transferable layer on the DAVIS dataset for transferring knowledge from seen/source categories (rows) to unseen/target object categories (columns). For each unseen category, the largest weight over all seen categories is marked in bold.

Sequence	bear	bswan	camel	eleph	goat	malw	rhino
aero	0.286	0.419	0.381	0.412	0.279	0.430	0.325
bike	0.317	0.372	0.393	0.423	0.358	0.309	0.432
bird	0.624	**0.891**	0.538	0.572	0.614	**0.780**	0.595
boat	0.392	0.419	0.358	0.460	0.323	0.474	0.428
bottle	0.401	0.336	0.307	0.410	0.349	0.387	0.368
bus	0.392	0.262	0.266	0.440	0.306	0.200	0.327
car	0.488	0.317	0.469	0.559	0.379	0.292	0.508
cat	**0.756**	0.436	0.417	0.574	0.609	0.398	0.492
chair	0.507	0.314	0.406	0.528	0.466	0.362	0.450
cow	0.701	0.409	0.715	0.748	0.618	0.346	**0.846**
table	0.341	0.310	0.186	0.301	0.291	0.504	0.257
dog	0.700	0.476	0.534	0.603	**0.788**	0.417	0.576
horse	0.547	0.330	**0.898**	**0.770**	0.692	0.260	0.776
mbike	0.301	0.287	0.346	0.408	0.371	0.287	0.355
person	0.504	0.429	0.731	0.639	0.554	0.366	0.629
plant	0.463	0.418	0.364	0.437	0.428	0.451	0.474
sheep	0.721	0.525	0.491	0.662	0.616	0.348	0.605
sofa	0.366	0.309	0.366	0.447	0.404	0.291	0.412
train	0.298	0.260	0.343	0.488	0.320	0.204	0.419
tv	0.369	0.252	0.277	0.425	0.271	0.248	0.303

In Table 3, we show the IoU of the selected segment proposals via solving (6) under various values of λ_m. The results show that the IoU is gradually improved when increasing the motion weight, which indicates that the quality of selected segments becomes better, and hence we use $\lambda_m = 35$ in all the following experiments.

Impact of Response Refinement. In Table 4, we present the IoU of the selected segment proposals with and without using fully-connected CRFs. With the refinement by CRFs, the IoU of the selected segments is improved by 2.5%. Therefore, the CNN model is able to learn better feature representations.

Sensitivity to Learning Rate. We provide the final results under different learning rates on the DAVIS dataset in Table 5. We fix the initial learning rate as 10^{-14} according to the results. We use a small learning rate to account for the unnormalized loss computed across spatial dimensions in our implementation. For example, if we perform normalization on the loss, the corresponding learning rate is around 10^{-8}.

Sensitivity to β for Submodular Optimization. In Table 6, we report the average IoU of the final results using different values of β for submodular optimization in (6). It can be observed that our method is robust to the value of β. Based on the results in Table 6, we set β to 0.8.

Analysis of Transferring Visual Information. We analyze the proposed method for transferring visual information by investigating the weights of the transferable layer. Table 7 presents the learned weights of the transferable layer on the DAVIS dataset for unseen object categories. For each target video, the source categories with higher weights are similar to the target video in appearance, which gives reasonable transform of visual information.

Ablation Study. In the middle group of Table 8, we show the final segmentation results of our method using VGG-16 architecture with various baselines and settings. We first present a baseline method that uses the GloVe embeddings (Pennington et al. 2014) to initialize weights, i.e., the similarity between two categories, of the transferable layer. Since the GloVe is not learned in the image domain between categories, the initialized weights may not reflect the true relationships between the seen and unseen categories, and hence the results are worse than the proposed method for initializing the transferable layer.

Furthermore, we show results at different stages, including using the model with initialization before optimizing (10), after optimization, after response refinement during training and the final result with CRF refinement as post-processing. After the optimization, the IoU is improved in 5 out of 7 videos, which shows the effectiveness of the proposed self-learning scheme without using any annotations in the target video. In addition, compared to our conference version, VOSTR$_a$, using CRFs at training and inference stages improve

Table 8 Results on the DAVIS 2016 dataset with categories excluded from the PASCAL VOC dataset.

Methods	bear	bswan	camel	eleph	goat	malw	rhino	Avg.
CVOS (Taylor et al. 2015)	86.4	42.2	85.0	49.4	7.4	24.5	52.0	49.6
MSG (Ochs and Brox 2011)	85.1	52.6	75.6	68.9	73.5	4.5	90.2	64.3
FST (Papazoglou and Ferrari 2013)	89.8	73.2	56.2	82.4	55.4	8.7	77.6	63.3
NLC (Faktor and Irani 2014)	90.7	87.5	76.8	51.8	1.0	76.1	68.2	64.6
LMP (Tokmakov et al. 2017a)	69.8	50.9	78.3	78.9	75.1	38.5	76.8	66.9
SPFTN (Zhang et al. 2017)	74.8	87.6	76.2	75.6	72.8	65.8	55.2	72.6
TransferNet (Hong et al. 2016)	73.7	83.4	65.5	**76.1**	78.1	17.9	42.4	62.4
VOSTR$_a$ (Chen et al. 2018b) (GloVe)	82.6	67.2	68.8	61.2	70.4	64.7	32.0	63.8
VOSTR$_a$ (Chen et al. 2018b) (init)	80.3	75.6	70.9	70.4	83.1	40.9	57.7	68.4
VOSTR$_a$ (Chen et al. 2018b) (final)	88.8	80.6	68.6	71.8	82.4	43.8	67.3	71.9
VOSTR (w/ response refinement at training)	90.1	84.2	72.3	70.6	82.7	72.4	66.5	77.0
VOSTR (final)	**94.5**	**92.8**	**79.0**	75.0	**85.0**	**84.9**	**67.5**	**82.7**
ARP (Koh and Kim 2017)	92.0	88.1	**90.3**	84.2	77.6	58.3	**88.4**	82.7
FSEG (Jain et al. 2017)	91.5	89.5	76.4	86.2	84.1	83.3	77.6	84.1
VOSTR$_a$ (Chen et al. 2018b) (ResNet)	91.8	90.3	77.5	85.7	84.8	84.9	86.0	85.9
VOSTR (ResNet)	**93.3**	**92.7**	80.7	**87.7**	**85.4**	**88.2**	88.2	**88.0**

the performance by 5.1% (from 71.9% to 77.0%) and 5.7% (from 77.0% to 82.7%), respectively. The results indicate that the CRFs enhance our method when they are used for post-processing as well as when they are employed to help the CNN model learn better feature representations.

Overall Comparisons. In Table 8, we show the comparisons between our method and the state-of-the-art approaches. We first demonstrate the performance of our method using VGG-16 architecture. The work closest in the scope to the proposed framework is the TransferNet method (Hong et al. 2016) that transfers the knowledge between two image domains with mutually exclusive categories in a weakly-supervised setting. To compare with this approach, we use the authors' public implementation and train the models with the same setting as our method. We first show that VOSTR$_a$ achieves better IoUs in 5 out of 7 videos and improves the overall IoU by 9.5% on average. With the response refinement step in our final model, the performance is further improved by 20.3% in IoU. We also note that our model with initialization already performs favorably against Hong et al. (2016), which demonstrates that the proposed transferable layer is effective in learning knowledge from seen categories to unseen ones.

In addition, we present more results of video object segmentation methods in Table 8 and show that the proposed algorithm achieves better performance. Different from existing approaches that rely on long-term trajectory (Taylor et al. 2015; Ochs and Brox 2011) or motion saliency (Papazoglou and Ferrari 2013; Faktor and Irani 2014) to localize foreground objects, we use the proposed self-learning framework to segment unseen object categories via transfer learning. We note that the proposed method performs better than the CNN-based model (Tokmakov et al. 2017a) that utilizes synthetic videos with pixel-wise segmentation annotations.

We further employ the stronger ResNet-101 architecture and compare with state-of-the-art unsupervised video object segmentation methods. In the bottom group of Table 8, we show that our approach performs better than FSEG (Jain et al. 2017) using the same architecture and training data from PASCAL VOC, i.e., the setting of the appearance stream in FSEG (Jain et al. 2017). Since the motion stream in FSEG adopts additional training data form the ImageNet-Video dataset (Russakovsky et al. 2015), it is not fair to compare our method with the motion stream and the joint model in FSEG. In addition, compared to ARP (Koh and Kim 2017) that adopts a non-learning based framework via proposal post-processing and is specifically designed for video object segmentation, our algorithm performs better and is flexible under various settings such as using weakly-supervised signals. Visual comparisons are presented in Fig. 3 and Fig. 4.

Results on the Entire DAVIS 2016 Dataset. In addition to performing object segmentation on unseen object categories, our method can adapt to the weakly-supervised setting by simply initializing the weights in the transferable layer as a one-hot vector, where only the known category is set to 1 and the others are 0. We evaluate this setting on the DAVIS 2016 dataset with categories shared in the PASCAL VOC dataset. Note that, we still adopt the unsupervised setting for the unseen categories. The results of each video from the DAVIS 2016 dataset are shown in Table 9. In comparison with a recent weakly-supervised method SPFTN (Zhang et al. 2017) and the baseline FCN (Long et al.

Fig. 3 Sample results on the DAVIS dataset for unseen object categories. Our results contain less noisy segments and more details than the approaches CVOS (Taylor et al. 2015), MSG (Ochs and Brox 2011), FST (Papazoglou and Ferrari 2013), TransferNet (Hong et al. 2016) and VOSTR$_a$ (Chen et al. 2018b).

2015) (our initial result), our approach addresses the transfer learning problem and already outperforms their methods without refining responses. By integrating the fully-connected CRFs objective, we further improve the performance by 8.9% and 8.5% with respect to SPFTN and FCN, respectively.

Although the same categories are shared between the source and target domains in this setting, we can still assume that the object category is unknown in the target video. Under this fully unsupervised setting without using any pixel-wise annotations in videos during training, we show that our method improves the results of FSEG (Jain et al. 2017) and other unsupervised algorithms (Ochs and Brox 2011; Papazoglou and Ferrari 2013; Faktor and Irani 2014). Sample results are presented in Fig. 5. In addition, we provide some

Fig. 4 Sample results on the DAVIS dataset for unseen object categories. Our results contain less noisy segments and more details than the approaches CVOS (Taylor et al. 2015), MSG (Ochs and Brox 2011), FST (Papazoglou and Ferrari 2013), TransferNet (Hong et al. 2016) and VOSTR$_a$ (Chen et al. 2018b).

failure cases in Fig. 6 caused by the objective of our method, which is to segment all the objects with the same category in a video.

5.4 YouTube-Objects Dataset

We evaluate the proposed method on the YouTube-Objects dataset (Prest et al. 2012) with annotations provided by Jain and Grauman (2014) for 126 videos. Since this dataset contains 10 object categories that are shared with the PASCAL VOC dataset, we conduct experiments using the weakly-supervised setting.

Table 9 Per-video results on the DAVIS 2016 dataset.

Methods	Weak Supervision				No Supervision					
	SPFTN (Zhang et al. 2017)	FCN (Long et al. 2015)	VOSTR$_a$ (Chen et al. 2018b)	VOSTR (Ours)	MSG (Ochs and Brox 2011)	FST (Papazoglou and Ferrari 2013)	NLC (Faktor and Irani 2014)	FSEG (Jain et al. 2017)	VOSTR$_a$ (Chen et al. 2018b)	VOSTR (Ours)
bear	74.8	80.3	89.8	**94.5**	85.1	89.8	90.7	91.5	91.8	**93.3**
bswan	87.6	75.6	76.7	**92.8**	52.6	73.2	87.5	89.5	90.3	**92.7**
bumps	29.7	29.9	36.2	**43.6**	35.3	24.1	**63.5**	38.8	42.1	43.2
trees	35.0	29.2	**40.5**	24.6	18.8	18.0	21.2	34.7	38.9	**39.1**
boat	35.9	63.4	**67.0**	61.0	14.4	36.1	0.7	63.8	**63.8**	62.6
bdan	37.1	14.6	**46.0**	14.7	23.6	46.7	**67.3**	14.2	13.1	13.0
bdanF	70.0	51.4	**80.0**	49.7	15.7	61.6	**80.4**	54.9	62.7	68.0
bus	**81.5**	61.1	81.2	62.6	**88.5**	82.5	62.9	80.4	80.5	81.2
camel	76.2	70.9	72.0	**79.0**	75.6	56.2	76.8	76.4	77.5	**80.7**
carR	76.8	71.0	**88.8**	71.4	63.0	**80.8**	50.9	74.8	79.6	79.1
carS	78.1	87.1	**92.5**	91.0	88.0	69.8	64.5	88.4	93.3	**94.0**
carT	75.4	86.7	**90.4**	88.9	62.1	85.1	83.3	90.7	92.5	**93.6**
cows	77.0	85.7	88.1	**90.5**	79.9	79.1	88.3	88.0	88.3	**90.1**
jump	34.2	33.6	**63.8**	40.9	6.5	59.8	**71.8**	10.3	11.2	11.4
twirl	46.1	27.8	**65.5**	31.1	36.6	45.3	34.7	**46.2**	41.0	41.8
dog	85.6	71.2	89.1	**89.5**	33.1	70.8	80.9	90.4	91.6	**93.8**
dogA	7.1	39.3	**72.9**	54.6	11.0	28.0	65.2	**68.9**	65.1	65.9
drtC	55.9	58.9	**67.1**	48.3	**75.8**	66.7	32.4	46.1	65.1	67.6
drtS	62.3	69.9	79.4	**80.2**	57.5	**68.3**	47.3	67.2	66.4	67.2
drtT	67.8	76.4	80.6	**81.1**	63.8	53.3	15.4	85.1	89.7	**90.6**
eleph	**75.6**	70.4	73.8	75.0	68.9	82.4	51.8	86.2	85.7	**87.7**
flamg	38.1	33.5	34.5	**44.4**	79.4	**81.7**	53.9	44.5	47.8	50.9
goat	72.8	83.1	83.3	**85.0**	73.5	55.4	1.0	84.1	84.8	**85.4**
hike	89.3	84.1	79.0	**90.8**	60.3	88.9	**91.8**	82.5	83.4	90.4
hockey	60.2	72.7	73.1	**83.8**	71.3	46.7	**81.0**	66.0	70.7	75.2
hjH	35.1	77.6	67.0	**82.1**	73.4	57.8	**83.4**	71.1	72.1	74.7
hjL	41.1	**79.5**	73.6	78.1	68.2	52.6	65.1	70.2	76.5	**78.7**
ksurf	58.3	55.8	46.5	**59.4**	41.9	27.2	45.3	47.7	49.0	**50.9**
kwalk	**73.3**	52.1	48.9	58.3	59.7	64.9	**81.3**	52.7	51.3	53.0
libby	50.8	49.5	59.4	**68.8**	5.0	50.7	63.5	67.7	68.1	**72.1**
lucia	83.3	84.2	78.9	**90.7**	41.7	64.4	**87.6**	79.9	81.0	85.5
malf	70.8	47.5	45.8	**74.4**	3.3	60.1	61.7	74.6	75.2	**77.5**
malw	65.8	40.9	41.6	**84.9**	4.5	8.7	76.1	83.3	84.9	**88.2**
motob	75.0	77.7	71.6	**82.5**	46.6	61.7	61.4	83.8	85.2	**87.3**
motoj	60.8	61.5	65.5	**72.7**	61.8	60.2	25.1	**80.4**	77.2	79.2
mbike	47.6	**78.5**	58.4	30.4	**73.8**	55.9	71.4	28.7	38.6	40.3
parag	72.6	30.9	28.1	**91.3**	**93.3**	72.5	88.0	17.7	5.5	3.5
paral	**62.8**	57.0	58.1	60.5	51.2	50.6	**62.8**	58.9	59.4	61.8
park	67.7	84.0	78.2	**89.8**	29.5	45.8	**90.1**	79.4	79.5	83.3
rhino	55.2	57.7	**71.0**	67.5	**90.2**	77.6	68.2	77.6	86.0	88.2
rolb	12.5	64.2	73.2	**83.5**	80.1	31.8	**81.4**	63.3	72.7	75.7
scbla	58.8	45.0	**72.1**	27.6	**57.9**	52.2	16.2	36.1	36.4	36.9
scgra	67.0	**73.7**	72.9	70.2	34.5	32.5	58.7	73.2	75.7	**77.6**
sobox	57.8	47.5	51.9	**74.2**	**67.2**	41.0	63.4	49.7	47.4	48.4
socB	49.0	49.5	46.3	**88.8**	37.0	**84.3**	82.9	29.3	28.3	28.2
strol	65.4	58.7	58.7	**82.5**	**67.8**	58.0	84.9	63.9	62.8	64.6
surf	87.0	78.4	79.1	**92.2**	77.0	47.5	77.5	88.8	91.2	**93.0**
swing	75.5	75.5	76.4	**80.7**	62.2	43.1	**85.1**	73.8	74.0	77.6
tennis	62.5	78.2	73.0	**82.4**	59.0	38.8	**87.1**	76.9	78.4	83.5
train	73.6	46.9	**77.3**	64.1	**88.7**	83.1	72.9	42.5	51.1	50.7
Avg.	61.2	61.6	67.7	**70.1**	54.3	57.5	64.1	64.7	66.5	**68.4**

Table 10 Results on the YouTube-Objects dataset.

Methods	aero	bird	boat	car	cat	cow	dog	horse	mbike	train	Avg.
DSA (Tang et al. 2013)	17.8	19.8	22.5	38.3	23.6	26.8	23.7	14.0	12.5	40.4	23.9
FCN (Long et al. 2015)	68.3	65.7	55.7	76.6	52.3	50.4	55.6	52.6	35.7	55.9	56.9
DET (Zhang et al. 2015b)	72.4	66.6	43.0	58.9	36.4	58.2	48.7	49.6	41.4	49.3	52.4
CoSeg (Tsai et al. 2016b)	69.3	**76.1**	57.2	70.4	67.7	59.7	64.2	57.1	44.1	57.9	62.3
SPFTN (Zhang et al. 2017)	**81.1**	68.8	63.4	73.8	59.7	64.5	63.4	58.2	52.4	45.5	63.1
VOSTR$_a$ (Chen et al. 2018b) (VGG)	74.6	65.3	**66.9**	**79.5**	64.2	68.3	67.3	61.7	51.5	59.4	65.9
VOSTR (VGG)	79.5	67.6	65.7	77.9	**68.2**	**72.8**	**73.0**	**63.4**	**61.9**	**60.0**	**69.0**
DeepLab (Chen et al. 2016)	80.6	67.8	66.9	73.3	55.3	61.8	63.9	45.5	54.7	56.4	62.6
FSEG (Jain et al. 2017)	83.4	60.9	**72.6**	74.5	**68.0**	69.6	69.1	**62.8**	61.9	**62.8**	68.6
VOSTR$_a$ (Chen et al. 2018b) (ResNet)	83.5	76.4	70.0	75.3	65.9	69.7	71.6	54.7	63.8	58.7	69.0
VOSTR (ResNet)	**85.2**	**77.3**	72.5	**77.9**	67.5	**70.5**	**74.4**	56.1	**66.0**	61.2	**70.9**

Table 11 Results on the SegTrack v2 dataset.

Methods	FST (Papazoglou and Ferrari 2013)	KEY (Lee et al. 2011)	HVS (Grundmann et al. 2010)	FSEG (Jain et al. 2017)	VOSTR$_a$ (Chen et al. 2018b)	VOSTR (Ours)
bird of paradise	81.8	92.2	86.8	49.8	49.2	50.7
birdfall	17.5	49.0	57.4	6.9	8.4	7.6
bmx	67.0	63.0	35.9	59.5	61.5	62.7
cheetah	28.0	28.1	21.6	71.2	72.0	73.8
drift	60.5	46.9	41.2	82.2	88.5	91.8
frog	54.1	0.0	67.1	54.9	59.5	61.5
girl	54.9	87.7	31.9	81.1	83.8	86.6
hummingbird	52.0	60.2	19.5	61.5	65.3	67.8
monkey	65.0	79.0	61.9	86.4	86.0	89.3
monkeydog	61.7	39.6	43.6	39.1	37.5	38.4
parachute	76.3	96.3	69.1	24.9	28.1	29.1
penguin	18.3	9.3	74.5	66.2	59.4	60.6
soldier	39.8	66.6	66.5	83.7	84.6	87.6
worm	72.8	84.4	34.7	29.1	29.6	31.0
Avg.	53.6	57.3	50.8	56.9	58.1	59.9

In Table 10, we compare our method with the state-of-the-art algorithms that use the class-level weak supervision. With the VGG-16 architecture, the proposed framework performs well in 6 out of 10 categories and achieves the best IoU on average. Compared to the baseline FCN model (Long et al. 2015) used in our algorithm, there is a performance gain of 9% in our VOSTR$_a$ method. With the response refinement, i.e., VOSTR, we further improve the baseline FCN model by 12.1%. In addition, while existing methods rely on training the segment classifier (Tang et al. 2013), integrating object proposals with detectors (Zhang et al. 2015b), co-segmentation via modeling relationships between videos (Tsai et al. 2016b), or self-paced fine-tuning (Zhang et al. 2017), the proposed method utilizes a self-learning scheme to achieve better segmentation results. With the ResNet-101 architecture, we compare our method with DeepLab (Chen et al. 2016) and FSEG (Jain et al. 2017). We show that the proposed method improves the performance in 6 out of 10 categories and achieves the best averaged IoU.

5.5 SegTrack v2 Dataset

In Table 11, we provide experiments on the SegTrack v2 dataset (Li et al. 2013) that contains numerous unseen objects. We use the ResNet-101 architecture and the training data from PASCAL VOC, which is the same setting as the appearance stream in FSEG (Jain et al. 2017). We show that the proposed method performs better than FSEG (Jain et al. 2017), other unsupervised algorithms (Papazoglou and Ferrari 2013; Lee et al. 2011) and HVS (Grundmann et al. 2010) that includes human annotations in the procedure. Compared to our conference version, VOSTR$_a$, we further improve the performance by integrating the fully-connected CRFs to our network.

6 Concluding Remarks

In this paper, we propose a self-learning framework to segment objects in unlabeled videos. By utilizing exist-

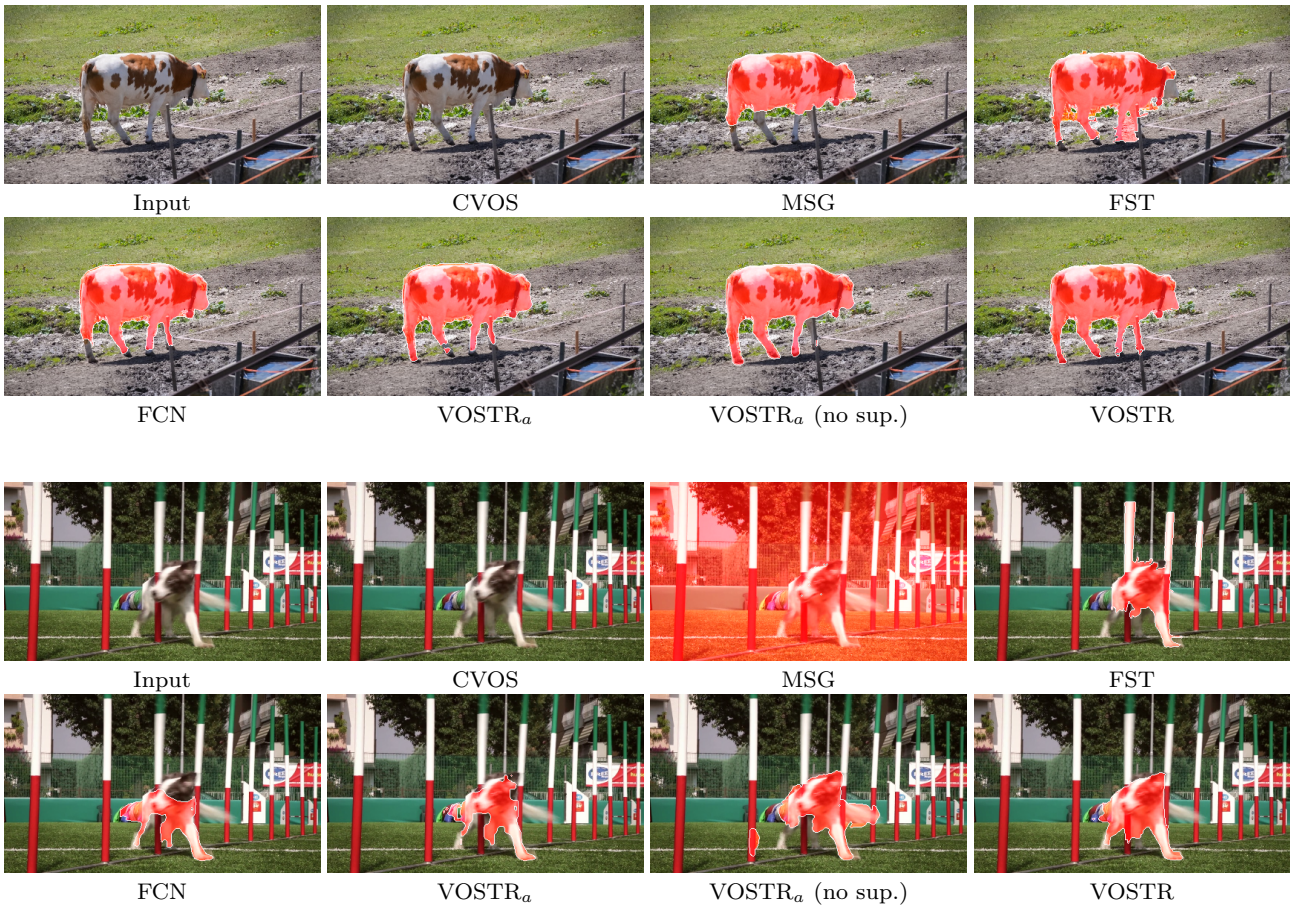

Fig. 5 Sample results on the DAVIS dataset with categories shared in the PASCAL VOC dataset. Comparing with the approaches CVOS (Taylor et al. 2015), MSG (Ochs and Brox 2011), FST (Papazoglou and Ferrari 2013), FCN (Long et al. 2015), and VOSTR$_a$ (Chen et al. 2018b), our approach VOSTR produces more complete object segments with details.

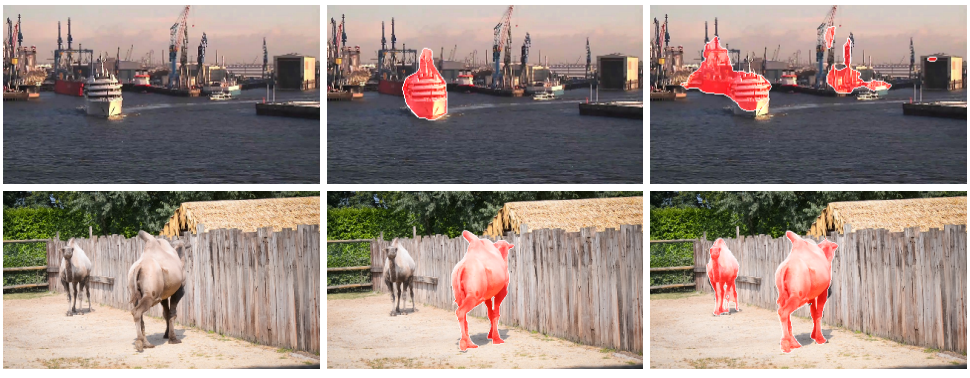

Fig. 6 Sample failure cases. Although our results differ from the ground truths, the segmented areas belong to the same semantic category.

ing annotations in images, we design a model to adapt seen object categories from source images to the target video. The entire process is decomposed into three sub-problems: 1) a fully-connected CRF model to refine responses from the CNN output, 2) a segment mining module to select object-like proposals, and 3) a CNN model with a transferable layer that adapts feature representations for target videos. To optimize the proposed formulation, we adopt an iterative scheme to obtain final solutions. Extensive experiments and ablation study show the effectiveness of our algorithm against other state-of-the-art methods on numerous datasets.

References

Caelles S, Maninis KK, Pont-Tuset J, Leal-Taixé L, Cremers D, Gool LV (2017) One-shot video object segmentation. In: CVPR 3

Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2016) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv:160600915 15

Chen Y, Pont-Tuset J, Montes A, Gool LV (2018a) Blazingly fast video object segmentation with pixel-wise metric learning. In: CVPR 3

Chen YW, Tsai YH, Yang CY, Lin YY, Yang MH (2018b) Unseen object segmentation in videos via transferable representations. In: ACCV 2, 4, 11, 12, 13, 14, 15, 16

Cheng J, Tsai YH, Wang S, Yang MH (2017) Segflow: Joint learning for video object segmentation and optical flow. In: ICCV 2

Cheng J, Tsai YH, Hung WC, Wang S, Yang MH (2018) Fast and accurate online video object segmentation via tracking parts. In: CVPR 3

Everingham M, Gool LJV, Williams CKI, Winn JM, Zisserman A (2010) The pascal visual object classes (voc) challenge. IJCV 88(2):303–338 1, 7

Faktor A, Irani M (2014) Video segmentation by non-local consensus voting. In: BMVC 11, 12, 14

Ganin Y, Lempitsky V (2015) Unsupervised domain adaptation by backpropagation. In: ICML 3

Gopalan R, Li R, Chellappa R (2011) Domain adaptation for object recognition: An unsupervised approach. In: ICCV 3

Grundmann M, Kwatra V, Han M, Essa I (2010) Efficient hierarchical graph-based video segmentation. In: CVPR 15

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: CVPR 9

Hoffman J, Guadarrama S, Tzeng ES, Hu R, Donahue J, Girshick R, Darrell T, Saenko K (2014) Lsda: Large scale detection through adaptation. In: NIPS 3

Hong S, Oh J, Lee H, Han B (2016) Learning transferrable knowledge for semantic segmentation with deep convolutional neural network. In: CVPR 2, 3, 4, 11, 12, 13

Hu R, Dollár P, He K, Darrell T, Girshick R (2018) Learning to segment every thing. CVPR 3

Jain S, Xiong B, Grauman K (2017) Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. In: CVPR 2, 3, 11, 12, 14, 15

Jain SD, Grauman K (2014) Supervoxel-consistent foreground propagation in video. In: ECCV 13

Khoreva A, Perazzi F, Benenson R, Schiele B, Sorkine-Hornung A (2017) Learning video object segmentation from static images. In: CVPR 3

Koh YJ, Kim CS (2017) Primary object segmentation in videos based on region augmentation and reduction. In: CVPR 2, 11

Krähenbühl P, Koltun V (2011) Efficient inference in fully connected crfs with gaussian edge potentials. In: NIPS 2, 7

Lazic N, Givoni I, Frey B, Aarabi P (2009) Floss: Facility location for subspace segmentation. In: ICCV 5

Lee YJ, Kim J, Grauman K (2011) Key-segments for video object segmentation. In: ICCV 2, 15

Li F, Kim T, Humayun A, Tsai D, Rehg JM (2013) Video segmentation by tracking many figure-ground segments. In: ICCV 2, 15

Li S, Seybold B, Vorobyov A, Lei X, Kuo CCJ (2018) Unsupervised video object segmentation with motion-based bilateral networks. In: ECCV 2

Lim JJ, Salakhutdinov R, Torralba A (2011) Transfer learning by borrowing examples for multiclass object detection. In: NIPS 3

Liu C (2009) Beyond pixels: Exploring new representations and applications for motion analysis. PhD thesis, MIT 6, 9

Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: CVPR 9, 12, 14, 15, 16

Luo Z, Zou Y, Hoffman J, Fei-Fei L (2017) Label efficient learning of transferable representations across domains and tasks. In: NIPS 3, 4

Märki N, Perazzi F, Wang O, Sorkine-Hornung A (2016) Bilateral space video segmentation. In: CVPR 2

Ochs P, Brox T (2011) Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In: ICCV 11, 12, 13, 14, 16

Oh SW, Lee JY, Sunkavalli K, Kim SJ (2018) Fast video object segmentation by reference-guided mask propagation. In: CVPR 3

Papazoglou A, Ferrari V (2013) Fast object segmentation in unconstrained video. In: ICCV 11, 12, 13, 14, 15, 16

Patricia N, Caputo B (2014) Learning to learn, from transfer learning to domain adaptation: A unifying perspective. In: CVPR 3

Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: EMNLP, pp 1532–1543 2, 10

Perazzi F, Wang O, Gross M, Sorkine-Hornung A (2015) Fully connected object proposals for video seg-

mentation. In: CVPR 2
Perazzi F, Pont-Tuset J, McWilliams B, Gool LV, Gross M, Sorkine-Hornung A (2016) A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR 2, 9
Prest A, Leistner C, Civera J, Schmid C, Ferrari V (2012) Learning object class detectors from weakly annotated video. In: CVPR 2, 13
Rochan M, Wang Y (2015) Weakly supervised localization of novel objects using appearance transfer. In: CVPR 6
Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) Imagenet large scale visual recognition challenge. IJCV 115(3):211–252 11
Saenko K, Kulis B, Fritz M, Darrell T (2010) Adapting visual category models to new domains. In: ECCV 3
Saleh FS, Aliakbarian MS, Salzmann M, Petersson L, Alvarez JM (2017) Bringing background into the foreground: Making all classes equal in weakly-supervised video semantic segmentation. In: ICCV 1
Shi Z, Yang Y, Hospedales TM, Xiang T (2017) Weakly-supervised image annotation and segmentation with objects and attributes. PAMI 39(12):2525–2538 3
Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556:1187–1200 9
Strand R, Ciesielski KC, Malmberg F, Saha PK (2013) The minimum barrier distance. CVIU 117(4):429–437 6
Tang K, Sukthankar R, Yagnik J, Fei-Fei L (2013) Discriminative segment annotation in weakly labeled video. In: CVPR 3, 15
Taylor B, Karasev V, Soatto S (2015) Causal video object segmentation from persistence of occlusions. In: CVPR 11, 12, 13, 16
Tokmakov P, Alahari K, Schmid C (2017a) Learning motion patterns in videos. In: CVPR 11
Tokmakov P, Alahari K, Schmid C (2017b) Learning video object segmentation with visual memory. In: ICCV 2
Tommasi T, Orabona F, Caputo B (2014) Learning categories from few examples with multi model knowledge transfer. PAMI 36:928–941 3
Tsai YH, Yang MH, Black MJ (2016a) Video segmentation via object flow. In: CVPR 2
Tsai YH, Zhong G, Yang MH (2016b) Semantic co-segmentation in videos. In: ECCV 1, 3, 5, 7, 15
Tsai YH, Hung WC, Schulter S, Sohn K, Yang MH, Chandraker M (2018) Learning to adapt structured output space for semantic segmentation. In: CVPR 3
Yan Y, Xu C, Cai D, Corso JJ (2017) Weakly supervised actor-action segmentation via robust multi-task ranking. In: CVPR 1
Yang L, Wang Y, Xiong X, Yang J, Katsaggelos AK (2018) Efficient video object segmentation via network modulation. In: CVPR 3
Zhang D, Yang L, Meng D, Xu D, Han J (2017) Spftn: A self-paced fine-tuning network for segmenting objects in weakly labelled videos. In: CVPR 1, 3, 11, 14, 15
Zhang J, Sclaroff S, Lin Z, Shen X, Price B, Mech R (2015a) Minimum barrier salient object detection at 80 fps. In: ICCV 6, 9
Zhang Y, Chen X, Li J, Wang C, Xia C (2015b) Semantic object segmentation via detection in weakly labeled video. In: CVPR 3, 15
Zhong G, Tsai YH, Yang MH (2016) Weakly-supervised video scene co-parsing. In: ACCV 3
Zhu F, Jiang Z, Shao L (2014) Submodular object recognition. In: CVPR 6