Instance Embedding Transfer to Unsupervised Video Object Segmentation

Siyang Li[1,2], Bryan Seybold[2], Alexey Vorobyov[2], Alireza Fathi[2], Qin Huang[1], and C.-C. Jay Kuo[1]
[1]University of Southern California, [2]Google AI Perception
siyangl@usc.edu, {seybold,vorobya,alirezafathi}@google.com,
qinhuang@usc.edu, cckuo@sipi.usc.edu

Abstract

We propose a method for unsupervised video object segmentation by transferring the knowledge encapsulated in image-based instance embedding networks. The instance embedding network produces an embedding vector for each pixel that enables identifying all pixels belonging to the same object. Though trained on static images, the instance embeddings are stable over consecutive video frames, which allows us to link objects together over time. Thus, we adapt the instance networks trained on static images to video object segmentation and incorporate the embeddings with objectness and optical flow features, without model retraining or online fine-tuning. The proposed method outperforms state-of-the-art unsupervised segmentation methods in the DAVIS dataset and the FBMS dataset.

1. Introduction

One important task in video understanding is object localization in time and space. Ideally, it should be able to localize familiar or novel objects consistently over time with a sharp object mask, which is known as video object segmentation (VOS). If no indication of which object to segment is given, the task is known as unsupervised video object segmentation or primary object segmentation. Once an object is segmented, visual effects and video understanding tools can leverage that information [2, 26].

Related object segmentation tasks in static images are currently dominated by methods based on the fully convolutional neural network (FCN) [5, 28]. These neural networks require large datasets of segmented object images such as PASCAL [9] and COCO [27]. Video segmentation datasets are smaller because they are more expensive to annotate [25, 32, 34]. As a result, it is more difficult to train a neural network explicitly for video segmentation. Classic work in video segmentation produced results using optical flow and shallow appearance models [12, 22, 24, 31, 33, 42] while more recent methods typically pretrain the network on image segmentation datasets and later adapt the network to the video domain, sometimes combined with optical flow [4, 6, 17, 37, 38, 41].

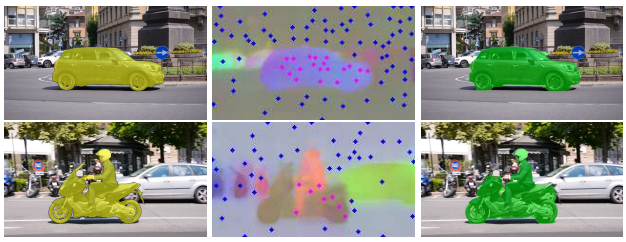

Figure 1. An example of the changing segmentation target (foreground) in videos depending on motion. A car is the foreground in the top video while a car is the background in the bottom video. To address this issue, our method obtains embeddings for object instances and identifies representative embeddings for foreground/background and then segments the frame based on the representative embeddings. **Left**: the ground truth. **Middle**: A visualization of the embeddings projected into RGB space via PCA, along with representative points for the foreground (magenta) and background (blue). **Right**: the segmentation masks produced by the proposed method. Best viewed in color.

In this paper, we propose a method to transfer the knowledge encapsulated in instance segmentation embeddings learned from static images and integrate it with objectness and optical flow to segment a moving object in video. Instead of training an FCN that directly classifies each pixel as foreground/background as in [6, 17, 37, 38], we train an FCN that jointly learns object instance embeddings and semantic categories from images [11]. The distance between the learned embeddings encodes the similarity between pixels. We argue that the instance embedding is a more useful feature to transfer from images to videos than a foreground/background prediction. As shown in Fig. 1, cars appear in both videos but belong to different categories (foreground in the first video and background in the second video). If the network is trained to directly classify cars as foreground on the first video, it tends to classify the cars as foreground in the second video as well. As a result, the network needs to be fine-tuned for each sequence [4]. In contrast, the instance embedding network can produce unique

embeddings for the car in both sequences without interfering with other predictions or requiring fine-tuning. The task then becomes selecting the correct embeddings to use as an appearance model. Relying on the embeddings to encode object instance information, we propose a method to identify the representative embeddings for the foreground (target object) and the background based on objectness scores and optical flow. Visual examples of the representative embeddings are displayed in the middle column of Fig. 1. Finally, all pixels are classified by finding the nearest neighbor in a set of representative foreground or background embeddings. This is a non-parametric process requiring no video specific supervision for training or testing.

We evaluate the proposed method on the DAVIS dataset [34] and the FBMS dataset [32]. Without fine-tuning the embedding network on the target datasets, we obtain better performance than previous state-of-the-art methods. More specifically, we achieve a mean intersection-over-union (IoU) of 78.5% and 71.9% on the DAVIS dataset [34] and the FBMS dataset [32], respectively.

To summarize, our main contributions include

- A new strategy for adapting instance segmentation models trained on static images to videos. Notably, this strategy performs well on video datasets without requiring any video object segmentation annotations.
- This strategy outperforms previously published unsupervised methods on both DAVIS benchmark and FBMS benchmark and approaches the performance of semi-supervised CNNs without requiring retraining any networks at test time.
- Proposal of novel criteria for selecting a foreground object without supervision, based on semantic score and motion features over a track.
- Insights into the stability of instance segmentation embeddings over time.

2. Related Work

Unsupervised video object segmentation. Unsupervised video object segmentation discovers the most salient, or primary, objects that move against a video's background or display different color statistics. One set of methods to solve this task builds hierarchical clusters of pixels that may delineate objects [12]. Another set of methods performs binary segmentation of foreground and background. Early foreground segmentation methods often used Gaussian Mixture Models and Graph Cut [29, 39], but more recent work uses convolutional neural networks (CNN) to identify foreground pixels based on saliency, edges, and/or motion [17, 37, 38]. For example, LMP [37] trains a network which takes optical flow as an input to separate moving and non-moving regions and then combines the results with objectness cues from SharpMask [35] to generate the moving object segmentation. LVO [38] trains a two-stream network, using RGB appearance features and optical flow motion features that feed into a ConvGRU [44] layer to generate the final prediction. FSEG [17] also proposes a two-stream network trained with mined supplemental data. SfMNet [40] uses differentiable rendering to learn object masks and motion models without mask annotations. Despite the risk of focusing on the wrong object, unsupervised methods can be deployed in more places because they do not require user interaction to specify an object to segment. Since we are interested in methods requiring no user interaction, we choose to focus on unsupervised segmentation.

Semi-supervised video object segmentation. Semi-supervised video object segmentation utilizes human annotations on the first frame (or more) of a video indicating the object to track. Importantly, the annotation provides a good appearance model initialization that unsupervised methods lack. The problem can be formulated as either a binary segmentation task conditioned on the annotated frame or a mask propagation task between frames. Non-CNN methods typically rely on Graph Cut [29, 39], but CNN-based methods offer better accuracy [4, 6, 19, 21, 41, 45]. Mask propagation CNNs take in the previous mask prediction and a new frame to segment the new frame. VPN [18] trains a bilateral network to propagate masks to new frames. MSK [21] trains a propagation network with synthetic transformations of still images and applies the same technique for online fine-tuning. SegFlow [6] finds that jointly learning moving object masks and optical flow helps to boost the segmentation performance. Binary segmentation CNNs typically utilize the first frame for fine-tuning the network to a specific sequence. The exact method for fine-tuning varies: OSVOS [4] simply fine-tunes on the first frame. OnAVOS [41] fine-tunes on the first frame and a subset of predictions from future frames. Fine-tuning can take seconds to minutes, and longer fine-tuning typically results in better segmentation. Avoiding the time cost of fine-tuning is a further inducement to focus on unsupervised methods.

Image segmentation. Many video object segmentation methods [4, 21, 41] are built upon image semantic segmentation neural networks [5, 15, 28], which predict a category label for each pixel. These fully convolutional networks allow end-to-end training on images of arbitrary sizes. Semantic segmentation networks do not distinguish different instances from the same object category, which limits their suitability to video object segmentation. Instance segmentation networks [11, 13, 30] can label each instance uniquely. Instance embedding methods [7, 11, 30] provide an embedding space where pixels belonging to the same instance have similar embeddings. Spatial variations in the embeddings indicate the edges of object masks. Relevant details are given in Sec. 3.1. It was unknown if instance embeddings are stable over time, but we hypothesized that these embeddings might be useful for video object segmentation.

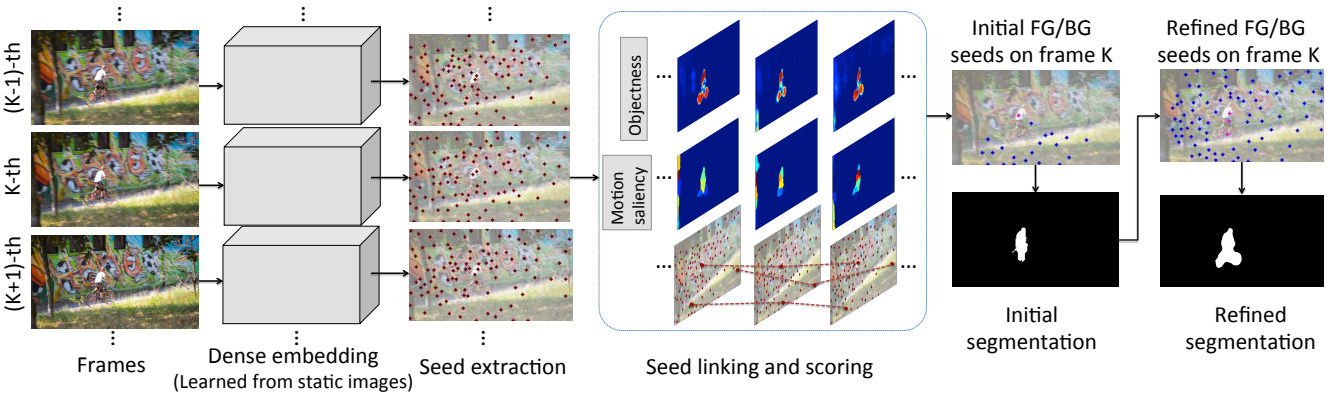

Figure 2. An overview of the proposed method. Given the video sequences, the dense embeddings are obtained by applying an instance segmentation network trained on static images. Then representative embeddings, called seeds, are obtained. Seeds are linked across the whole sequence (we show 3 consecutive frames as an illustration here). The seed with the highest score based on objectness and motion saliency is selected as the initial seed (in magenta) to grow the initial segmentation. Finally, more foreground seeds as well as background seeds are identified to refine the segmentation.

3. Proposed Method

An overview of the proposed method is depicted in Fig. 2. We first obtain instance embeddings, objectness scores, and optical flow that we will use as inputs (Sec. 3.1). Based on the instance embeddings, we identify "seed" points that mark segmentation proposals (Sec. 3.2). Consistent proposal seeds are linked to build seed tracks, and we rank the seed tracks by objectness scores and motion saliency to select a foreground proposal seed on every frame (Sec. 3.3). We further build a set of foreground/background proposal seeds to produce the final segmentation mask in each frame (Sec. 3.4).

3.1. Extracting features

Our method utilizes three inputs: instance embeddings, objectness scores, and optical flow. None of these features are fine-tuned on video object segmentation datasets or fine-tuned online to specific sequences. The features are extracted for each frame independently as follows.

Instance embedding and objectness. We train a network to output instance embeddings and semantic categories on the image instance segmentation task as in [11]. Briefly, the instance embedding network is a dense-output convolutional neural network with two output heads trained on static images from an instance segmentation dataset.

The first head outputs an embedding for each pixel, where pixels from same object instance have smaller Euclidean distances between them than pixels belonging to separate objects. Similarity R between two pixels i and j is measured as a function of the Euclidean distance in the E-dimensional embedding space, $\mathbf{f}$,

$$R(i,j) = \frac{2}{1+\exp(||\mathbf{f}(i)-\mathbf{f}(j)||_2^2)}. \quad (1)$$

This head is trained by minimizing the cross entropy between the similarity and the ground truth matching indicator $g(i,j)$. For locations i and j, the ground truth matching indicator $g(i,j)=1$ if pixels belong to the same instance and $g(i,j)=0$ otherwise. The loss is given by

$$L_s = -\frac{1}{|A|}\sum_{i,j\in A} w_{ij}[g(i,j)\log(R(i,j)) + (1-g(i,j))\log(1-R(i,j))], \quad (2)$$

where A is a set of pixel pairs, $R(i,j)$ is the similarity between pixels i and j in the embedding space and w_{ij} is inversely proportional to instance size to balance training.

The second head outputs an objectness score from semantic segmentation. We minimize a semantic segmentation log loss to train the second head to output a semantic category probability for each pixel. The objectness map is derived from the semantic prediction as

$$O(i) = 1 - P_{BG}(i), \quad (3)$$

where $P_{BG}(i)$ is the probability that pixel i belongs to the semantic category "background"[1]. We do not use the scores for any class other than the background in our work.

Embedding graph. We build a 4-neighbor graph from the dense embedding map, where each embedding vector becomes a vertex and edges exist between spatially neighboring embeddings with weights equal to the Euclidean distance between embedding vectors. This embedding graph will be used to generate image regions later. A visualized embedding graph is shown in Fig. 3.

[1]Here in semantic segmentation, "background" refers to the region that does not belong to any category of interest, as opposed to video object segmentation where the "background" is the region other than the target object. We use "background" as in video object segmentation for the rest of the paper.

Optical flow. The motion saliency cues are built upon optical flow. For fast optical flow estimation at good precision, we utilize a reimplementation of FlowNet 2.0 [16], an iterative neural network.

3.2. Generating proposal seeds

We propose a small number of representative seed points S^k in frame k for some subset of frames K (typically all) in the video. Most computations only compare against seeds within the current frame, so the superscript ^{k} is omitted for clarity unless the computation is across multiple frames. The seeds we consider as FG or BG should be diverse in embedding space because the segmentation target can be a moving object from an arbitrary category. In a set of diverse seeds, at least one seed should belong to the FG region. We also need at least one BG seed because the distances in the embedding space are relative. The relative distances in embedding space, or similarity from Eq. 1, from each point to the FG and BG seed(s) can be used to assign a labels to all pixels.

Candidate points. In addition to being diverse, the seeds should be representative of objects. The embeddings on the boundary of two objects are usually not close to the embedding of either object. Because we want embeddings representative of objects, we exclude seeds from object boundaries. To avoid object boundaries, we only select seeds from candidate points where the instance embeddings are locally consistent. (An alternative method to identify the boundaries to avoid would be to use an edge detector such as [8, 43].) We construct a map of embedding edges by mapping discontinuities in the embedding space. The embedding edge map is defined as the "inverse" similarity in the embedding space within the neighbors around each pixel,

$$c(p) = \max_{q \in N(p)} 1 - R(p, q), \tag{4}$$

where p and q are pixel locations, $N(p)$ contains the four neighbors of p, and $R(r, q)$ is the similarity measure given in Eq. 1. Then in the edge map we identify the pixels which are the minimum within a window of $n \times n$ centered at itself. These pixels from the candidate set C. Mathematically,

$$C = \{p | c(p) = \min_{q \in W(p)} c(q)\}, \tag{5}$$

where $W(p)$ denotes the local window.

Diverse seeds. These candidate points, C, are diverse, but still redundant with one another. We take a diverse subset of these candidates as seeds by adopting the sampling procedure from KMeans++ initialization [1]. We only need diverse sampling rather than cluster assignments, so we do not perform the time-consuming KMeans step afterwards. The sampling procedure begins by adding the candidate point with the largest objectness score, $O(i)$, to the seed set, S. Sampling continues by iteratively adding the candidate, s_{n+1}, with smallest maximum similarity to all previously selected seeds and stops when we reach N_S seeds,

$$s_{n+1} = \arg \min_{i \in C} \max_{j \in S} R(i, j). \tag{6}$$

We repeat this procedure to produce the seeds for each frame independently, forming a seed set S. Note that the sampling strategy differs from [11], where they consider a weighted sum of the embedding distance and semantic scores. We do not consider the semantic scores because we want to have representative embeddings for all regions of the current frame, including the background, while in [11], the background is disregarded. Fig. 3 shows one example of the visualized embedding edge map, the corresponding candidate set and the selected seeds.

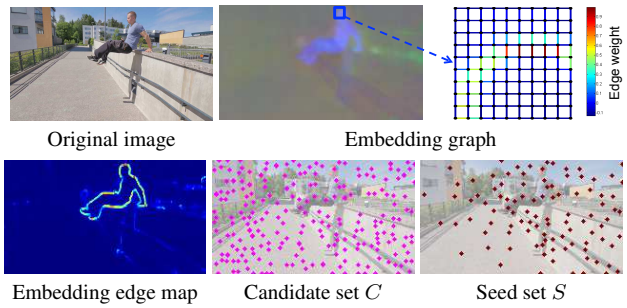

Figure 3. **Top:** An image (*left*) and the visualization of its embedding graph in the 10×10 box in blue. The edge colors on the *right* reflect distances between the embeddings at each pixel (the *center* subfigure visualizes the embeddings via PCA). High costs appear along object boundaries. **Bottom:** The embedding edge map (*left*), the candidate set C (*center*) and the seed set S (*right*). Best viewed in color.

3.3. Ranking proposed seeds

In the unsupervised video object segmentation problem, we do not have an explicitly specified target object. Therefore, we need to identify a moving object as the segmentation target (i.e., FG). We first score the seeds based on objectness and motion saliency. To find the most promising seed for FG, we then build seed tracks by group embedding-consistent seeds across frames into seed tracks and aggregate scores along tracks. The objectness score is exactly $O(s)$ in Eq. 3 for each seed. The motion saliency as well as seed track construction and ranking are explained below.

Motion saliency. Differences in optical flow can separate objects moving against a background [3]. Because optical flow estimation is still imperfect [16], we average flow within the image regions rather than using the flow from a single pixel. The region corresponding to each seed consists of the pixels in the embedding graph from Sec. 3.1 with the shortest geodesic distance to that seed. For each seed s, we use the average optical flow in the corresponding region as $\mathbf{v}_s$. An example of image regions is shown in Fig. 4.

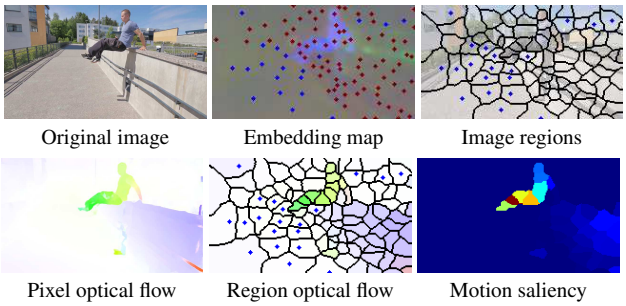

Figure 4. **Top:** *left* - an image; *center* - a projection of the embedding map into RBG space (via PCA) with the initial background seeds S_{BG} marked in blue and other seeds in red; *right* - the regions near each seed in the embedding graph. **Bottom:** *left* - the optical flow; *center* - average flow within each region; *right* - a map of motion saliency scores. Best viewed in color.

Then we construct a model of the background. First, N_{BG} seeds with the lowest objectness score, $O(s)$, are selected as the initial background seeds, denoted by S_{BG}. The set of motion vectors associated with these seeds forms our background motion model V_{BG}. The motion saliency for each seed, s, is the normalized distance to the nearest background motion vector,

$$M(s) = \frac{1}{Z} \min_{\mathbf{v}_b \in V_{BG}} ||\mathbf{v}_s - \mathbf{v}_b||_2^2, \tag{7}$$

where the normalization factor Z is given by

$$Z = \max_{s \in S} (\min_{\mathbf{v}_b \in V_{BG}} ||\mathbf{v}_s - \mathbf{v}_b||_2^2). \tag{8}$$

There are other approaches to derive motion saliency from optical flow. For example, in [22], motion edges are obtained by applying some edge detector to optical flow and then motion saliency of some region is computed as a function of motion edge intensity. The motion saliency proposed in this work is more efficient and works well in terms of the final segmentation performance.

Seed tracks. Another property of the foreground object is that it should be a salient object in multiple frames. We score this by linking similar seeds together across frames into a seed track and taking the average product of objectness and motion saliency scores over each track. The j-th seed on frame 0, s_j^0, initiates a seed track, T_j. T_j is extended frame by frame by adding the seed with the highest similarity to T_j. Specifically, supposing that we have a track T_j^m across frames 0-m, it is extended to frame $m+1$ by adding the most similar seed on frame $m+1$ to T_j^m, forming T_j^{m+1}:

$$r = \arg \max_{s \in S^{m+1}} \sum_{t \in T_j^m} R(s,t), \tag{9}$$

where $R(s,t)$ is the similarity measure given by Eq. 1, and r is the seed in frame $m+1$ with the highest similarity to T_j^m. Then we have $T_j^{m+1} = T_j^m \bigcup \{r\}$. Eventually, we have T_j starting from s_j^0 and ending at some seed on the last frame. The foreground score for T_j is

$$F(T_j) = \frac{1}{|T_j|} \sum_{s \in T_j} O(s) M(s), \tag{10}$$

where $|T_j|$ is the size of the seed track, equal to the sequence length.

3.4. Segmenting a foreground proposal

Initial foreground segmentation. The seed track with the highest foreground score is selected on each frame to provide an initial foreground seed, denoted by s_{FG}^k. We obtain an initial foreground segmentation by identifying pixels close to the foreground seed s_{FG}^k in the embedding graph explained in Sec. 3.1. Here the distance, denoted by $d(p,s)$, between any two nodes, p and s, is defined as the maximum edge weight along the shortest geodesic path connecting them. We again take the N_{BG} seeds with the lowest objectness scores as the initial background seed set, S_{BG}^k. Then the initial foreground region I_{FG} is composed of the pixels closer to the foreground seed s_{FG}^k than any background seeds,

$$I_{FG} = \{p | d(p, s_{FG}^k) < \min_{b \in S_{BG}^k} d(p,b)\}. \tag{11}$$

Adding foreground seeds. Often, selecting a single foreground seed is overly conservative and the initial segmentation fails to cover an entire object. To expand the foreground segmentation, we create a set of foreground seeds, S_{FG}^k from the combination of s_{FG}^k and seeds marking image regions mostly covered by the initial foreground segmentation. These image regions are the ones described in Sec. 3.3 and Fig. 4. If more than a proportion α of a region intersects with the initial foreground segmentation, the corresponding seed is added to the S_{FG}^k.

Adding background seeds. The background contains two types of regions: non-object regions (such as sky, road, water, etc.) that have low objectness scores, and static objects, which are hard negatives because in the embedding space, they are often closer to the foreground than the non-object background. Static objects are particularly challenging when they belong to the same semantic category as the foreground object[2]. We expand our representation of the BG regions by taking the union of seeds with object scores less than a threshold O_{BG} and seeds with motion saliency scores less than a threshold M_{BG}:

$$S_{BG}^k = \{s | O(s) \leq O_{BG}\} \bigcup \{s | M(s) \leq M_{BG}\}. \tag{12}$$

[2]E.g., the "camel" sequence in the DAVIS dataset [34].

Final segmentation. Once the foreground S^k_{FG} and background S^k_{BG} sets are established, similarity to the nearest foreground and background seeds is computed for each pixel. It is possible to use the foreground and background sets from one frame to segment another frame for FG/BG similarity computation:

$$R^l_{FG}(i^l) = \max_{s \in S^k_{FG}} R(i^l, s), \tag{13}$$

$$R^l_{BG}(i^l) = \max_{s \in S^k_{BG}} R(i^l, s), \tag{14}$$

where pixels on the target frame l are denoted by i^l. Instead of directly propagating the foreground or background label from the most similar seed to the embedding, we obtain a soft score as the confidence of the embedding i^l being foreground:

$$P_{FG}(i^l) = \frac{R^k_{FG}(i^l)}{R^k_{FG}(i^l) + R^k_{BG}(i^l)}. \tag{15}$$

Finally, the dense CRF [23] is used to refine the segmentation mask, with the unary term set to the negative log of $P_{FG}(i^l)$, as in [5].

Online adaptation. Online adaptation of our method is straightforward: we simply generate new sets of foreground and background seeds. This is much less expensive than fine-tuning an FCN for adaptation as done in [41]. Though updating the foreground and background sets could result in segmenting different objects in different frames, it improves the results in general, as discussed in Sec. 4.4.

4. Experiments

4.1. Datasets and evaluation

We evaluate the proposed method on the DAVIS dataset [34], Freiburg-Berkeley Motion Segmentation (FBMS) dataset [32], and the SegTrack-v2 dataset [25]. Note that neither the embedding network nor the optical flow network has been trained on these datasets.

DAVIS. The DAVIS 2016 dataset [34] is a recently constructed dataset, containing 50 video sequences in total, with 30 in the *train* set and 20 in the *val* set. It provides binary segmentation ground truth masks for all 3455 frames. This dataset contains challenging videos featuring object deformation, occlusion, and motion blur. The "target object" may consist of multiple objects that move together, e.g., a bike with the rider. To evaluate our method, we adopt the protocols in [34], which include region similarity and boundary accuracy, denoted by $\mathcal{J}$ and $\mathcal{F}$, respectively. $\mathcal{J}$ is computed as the intersection over union (IoU) between the segmentation results and the ground truth. $\mathcal{F}$ is the harmonic mean of boundary precision and recall.

FBMS. The FBMS dataset [32] contains 59 video sequences with 720 frames annotated. In contrast to DAVIS, multiple moving objects are annotated separately in FBMS. We convert the instance-level annotations to binary masks by merging all foreground annotations, as in [38]. The evaluation metrics include the F-score evaluation protocol proposed in [32] as well as $\mathcal{J}$ used for DAVIS.

SegTrack-v2. The SegTrack-v2 dataset [25] contains 14 videos with a total of 976 frames. Annotations of individual moving objects are provided for all frames. As with FBMS, the union of the object masks is converted to a binary mask for unsupervised video object segmentation evaluation. For this dataset, we only use $\mathcal{J}$ for evaluation to be consistent with previous work.

4.2. Implementation details

We use the image instance segmentation network trained on the PASCAL VOC 2012 dataset [11] to extract the object embedding and objectness. The instance segmentation network is based on DeepLab-v2 [5] with ResNet [14] as the backbone. We use the stabilized optical flow from a re-implementation of FlowNet2.0 [16]. The dimension for the embedding vector, E, is 64. The window size n to identify the candidate set C is set to 9 for DAVIS/FBMS, and 5 for SegTrack-v2. For frames in DAVIS dataset, the 9x9 window results in approximately 200 candidates in the embedding edge map. We select $N_S = 100$ seeds from the candidates. The initial number of background seeds N_{BG} is $N_S/5$. To add FG seeds as in Sec. 3.4, α is set to 0.5. The thresholds for BG seed selection are $O_{BG} = 0.3$ and $M_{BG} = 0.01$. The CRF parameters are identical with the ones in DeepLab [5] (used for the PASCAL dataset). We first tuned all of these parameters on the DAVIS *train* set, where we our $\mathcal{J}$ was 77.5%. We updated the window size n for SegTrack-v2 empirically, considering the video resolution.

4.3. Comparing to the state-of-the-art

DAVIS. As shown in Tab. 1, we obtain the best performance for unsupervised video object segmentation: 2.4% higher in $\mathcal{J}$ and 4.0% in $\mathcal{F}$ than the second best for each metric. Our unsupervised approach even outperforms some of the semi-supervised methods that have access to the first frame annotations, VPN [18] and SegFlow [6], by more than 2%[3]. Some qualitative segmentation results are shown in Fig. 5.

FBMS. We evaluate the proposed method on the *test* set, with 30 sequences in total. The results are shown in Tab. 2. Our method achieves an F-score of 82.8%: 5.0% higher than the second best method [38]. Our method's $\mathcal{J}$ mean is more than 10% better than ARP [22], which performs the second best on DAVIS.

SegTrack-v2. We achieve a $\mathcal{J}$ of 59.3% on this dataset, which is higher than other methods that do well on DAVIS, LVO [38] (57.3%) and FST [33] (54.3%). Due to the low

[3]Numeric results for these two methods are listed in Tab. 5.

	NLC [10]	CUT [20]	FST [33]	SFL [6]	LVO [38]	MP [37]	FSEG [17]	ARP[22]	Ours
Fine-tune on DAVIS?	No	Yes	No	Yes	Yes	No	No	No	No
$\mathcal{J}$ Mean	55.1	55.2	55.8	67.4	75.9	70.0	70.7	76.2	**78.6**
$\mathcal{F}$ Mean	52.3	55.2	51.1	66.7	72.1	65.9	65.3	70.6	**76.1**

Table 1. The results on the *val* set of DAVIS 2016 dataset [34]. Our method achieves the highest in both evaluation metrics, and outperforms the methods fine-tuned on DAVIS. Online adaptation is applied on every frame.

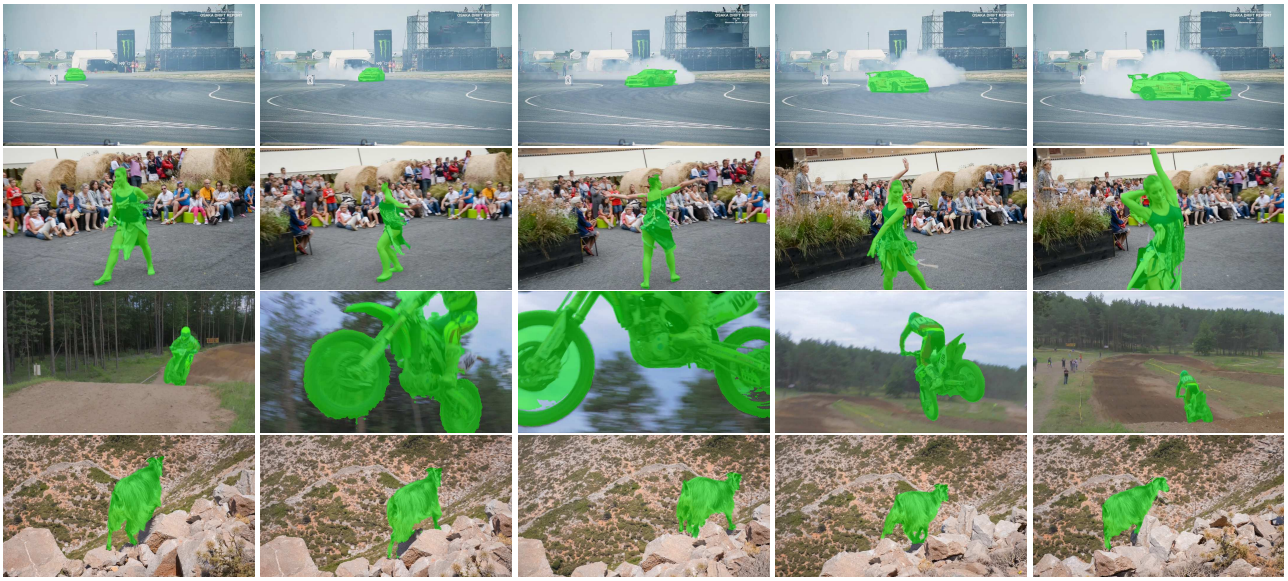

Figure 5. Example qualitative results on DAVIS dataset. Our method performs well on videos with large appearance changes (top row), confusing backgrounds (second row, with people in background), and changing viewing angle (third row). Best viewed in color.

	NLC [10]	CUT [20]	FST [33]	CVOS[36]	LVO [38]	MP [37]	ARP[22]	Ours
F-score	-	76.8	69.2	74.9	77.8	77.5	-	**82.8**
$\mathcal{J}$ Mean	44.5	-	55.5	-	-	-	59.8	**71.9**

Table 2. The results on the *test* set of FBMS dataset [32][4]. Our method achieves the highest in both evaluation metrics.

resolution of SegTrack-v2 and the fact that SegTrack-v2 videos can have multiple moving objects of the same class in the background, we are weaker than NLC [10] (67.2%) in this dataset.

4.4. Ablation studies

The effectiveness of instance embedding. To prove that instance embeddings are more effective than the embeddings from semantic segmentation networks in our method, we compare against the features from DeepLab-v2 [5]. Replacing the instance embedding in our method with DeepLab second-last-layer features achieves 65.2% in $\mathcal{J}$, more than 10% worse than the instance embedding features. This proves that the instance embeddings are much more suited to linking objects over time and space than semantic segmentation features. The explicit pixel-wise similarity loss (Eq. 2) used to train instance embeddings helps to produce more stable features than semantic segmentation.

[4]The numeric results of previous methods are taken from corresponding papers, where oftentimes only one evaluation metric is reported.

Embedding temporal consistency and online adaptation. We analyze whether embeddings for an object are consistent over time. Given the embeddings for each pixel in every frame and the ground truth foreground masks, we determine how many foreground embeddings in later frames are closer to the background embeddings than foreground embeddings from the first frame. If a foreground embedding from a later frame is closer to any background embedding in the first frame, we call it an incorrectly classified foreground pixel. We plot the proportion of foreground pixels that are incorrectly classified as a function of relative timestep in Fig. 6. As the timestep increases, more foreground embeddings are incorrectly classified. This "embedding drift" problem is probably caused by the changing appearance and location of objects in the video.

To overcome "embedding drift", we do online adaptation by updating the foreground and background seed sets. Updating the seeds is much faster than fine-tuning a neural network for adaptation, as done in OnAVOS [41]. The effects of doing online adaptation every k frames are detailed

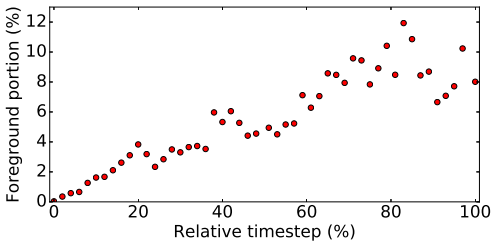

Figure 6. The proportion of incorrectly classified foreground embeddings versus relative timestep. As time progresses, more foreground embeddings are closer to first frame's background than the foreground.

in Tab. 3. More frequent online adaptation results in better performance: per-frame online adaptation boosts $\mathcal{J}$ by 7.0% over non-adaptive seed sets from the first frame.

Adapt every k frame	k=1	k=5	k=10	k=20	k=40	k=∞
$\mathcal{J}$ Mean	77.5	76.4	75.9	75.0	73.6	70.5

Table 3. The segmentation performance versus online adaptation frequency. Experiments conducted on DAVIS *train* set. Note that k=∞ means no online adaptation.

Foreground seed track ranking. In this section, we discuss some variants of foreground seed track ranking. In Eq. 10, the ranking is based on objectness as well as motion saliency. We analyze three variants: motion saliency alone, objectness alone, and objectness+motion saliency. The results are shown in Tab. 4. The experiments are conducted on the DAVIS *train* set. The initial FG seed accuracy (second row in Tab. 4) is evaluated as the proportion of the initial foreground seeds located within the ground truth foreground region. We see that combining motion saliency and objectness performs the best, outperforming "motion alone" and "objectness alone" by 4.0% and 6.4%, respectively. Final segmentation performance is consistent with the initial foreground seed accuracy, with the combined ranking outperforming the "motion alone" and "objectness alone" by 3.2% and 1.8%, respectively. The advantage of combining motion and objectness is also reported in several previous methods [6, 17, 38]. It is interesting that using objectness gives lower initial foreground seed accuracy but higher $\mathcal{J}$ mean than motion only. It is probably because of the different errors the two scores make. When errors occur in "motion only" mode, it is more likely that seeds representing "stuff" (sky, water, road, etc) are selected as the foreground, but when errors occur in "objectness only" mode, incorrect foreground seeds are usually located on static objects. In the embedding space, static objects are usually closer to the target object than "stuff", so these errors are more forgiving.

4.5. Preliminary results for Semi-supervised VOS

We further prove the effectiveness of the instance embeddings by extending them to semi-supervised video object segmentation, where the foreground/background seeds are selected based on the first frame annotation. The seeds covered by the ground truth mask are added to the foreground set S^0_{FG} and the rest are added to S^0_{BG}. Then we apply Eqs. 13-15 with $k = 0$ to all embeddings of the sequence. Results are further refined by a dense CRF. As shown in Tab. 5, we achieve 77.6% in $\mathcal{J}$, better than [6] and [18]. Note that there are more options for performance improvement such as motion/objectness analysis and online adaptation as experimented in unsupervised video object segmentation. We leave those options for future exploration.

	Motion	Obj.	Motion + Obj.
Init. FG seed acc.	90.6	88.2	94.6
$\mathcal{J}$ Mean	74.3	75.7	77.5

Table 4. The segmentation performance versus foreground ranking strategy. Experiments are conducted on DAVIS *train* set.

	DAVIS fine-tune?	Online fine-tune?	$\mathcal{J}$ Mean
OnAVOS [41]	Yes	Yes	86.1
OSVOS [4]	Yes	Yes	79.8
SFL [6]	Yes	Yes	76.1
MSK [21]	No	Yes	79.7
VPN [18]	No	No	70.2
Ours	No	No	77.6

Table 5. The results of semi-supervised video object segmentation on DAVIS *val* set by adopting the instance embeddings.

5. Conclusions

In this paper, we propose a method to transfer the instance embedding learned from static images to unsupervised object segmentation in videos. To be adaptive to the changing foreground in the video object segmentation problem, we train a network to produce embeddings encapsulating instance information rather than training a network that directly outputs a foreground/background score. In the instance embeddings, we identify representative foreground/background embeddings from objectness and motion saliency. Then, pixels are classified based on embedding similarity to the foreground/background. Unlike many previous methods that need to fine-tune on the target dataset, our method achieves the state-of-the-art performance under the unsupervised video object segmentation setting without any fine-tuning, which saves a tremendous amount of labeling effort.

Acknowledgements. This work was started when Siyang Li was an intern at Google and later continued at USC with support from Ittiam Systems. We would like to thank Susanna Ricco, Cordelia Schmid, David Ross, and Rahul Sukthankar for helpful discussions.

References

[1] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007. 4

[2] M. Billinghurst, A. Clark, G. Lee, et al. A survey of augmented reality. *Foundations and Trends® in Human–Computer Interaction*, 8(2-3):73–272, 2015. 1

[3] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. *Computer Vision–ECCV 2010*, pages 282–295, 2010. 4

[4] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *CVPR 2017*. IEEE, 2017. 1, 2, 8

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016. 1, 2, 6, 7

[6] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 2, 6, 7, 8

[7] B. De Brabandere, D. Neven, and L. Van Gool. Semantic instance segmentation with a discriminative loss function. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2017. 2

[8] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1841–1848, 2013. 4

[9] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010. 1

[10] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *BMVC*, volume 2, page 8, 2014. 7

[11] A. Fathi, Z. Wojna, V. Rathod, P. Wang, H. O. Song, S. Guadarrama, and K. P. Murphy. Semantic instance segmentation via deep metric learning. *arXiv preprint arXiv:1703.10277*, 2017. 1, 2, 3, 4, 6

[12] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph based video segmentation. *IEEE CVPR*, 2010. 1, 2

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 2

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[15] Q. Huang, C. Xia, C. Wu, S. Li, Y. Wang, Y. Song, and C.-C. J. Kuo. Semantic segmentation with reverse attention. In *British Machine Vision Conference*, 2017. 2

[16] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 4, 6

[17] S. D. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2, 7, 8

[18] V. Jampani, R. Gadde, and P. V. Gehler. Video propagation networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2017. 2, 6, 8

[19] W.-D. Jang and C.-S. Kim. Online video object segmentation via convolutional trident network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5849–5858, 2017. 2

[20] M. Keuper, B. Andres, and T. Brox. Motion trajectory segmentation via minimum cost multicuts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3271–3279, 2015. 7

[21] A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 8

[22] Y. J. Koh and C.-S. Kim. Primary object segmentation in videos based on region augmentation and reduction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 5, 6, 7

[23] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011. 6

[24] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1995–2002. IEEE, 2011. 1

[25] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2192–2199, 2013. 1, 6

[26] H.-D. Lin and D. G. Messerschmitt. Video composition methods and their semantics. In *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, pages 2833–2836. IEEE, 1991. 1

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[28] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1, 2

[29] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 743–751, 2016. 2

[30] A. Newell, Z. Huang, and J. Deng. Associative embedding: End-to-end learning for joint detection and grouping. In

Advances in Neural Information Processing Systems, pages 2274–2284, 2017. 2

[31] P. Ochs and T. Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1583–1590. IEEE, 2011. 1

[32] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1187–1200, 2014. 1, 2, 6, 7

[33] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1777–1784, 2013. 1, 6, 7

[34] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 2, 5, 6, 7

[35] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *European Conference on Computer Vision*, pages 75–91. Springer, 2016. 2

[36] B. Taylor, V. Karasev, and S. Soatto. Causal video object segmentation from persistence of occlusions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4268–4276, 2015. 7

[37] P. Tokmakov, K. Alahari, and C. Schmid. Learning motion patterns in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2, 7

[38] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 1, 2, 6, 7, 8

[39] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3899–3908, 2016. 2

[40] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017. 2

[41] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *British Machine Vision Conference*, 2017. 1, 2, 6, 7, 8

[42] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3395–3402, 2015. 1

[43] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015. 4

[44] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015. 2

[45] J. S. Yoon, F. Rameau, J. Kim, S. Lee, S. Shin, and I. S. Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2186–2195. IEEE, 2017. 2