SegFlow: Joint Learning for Video Object Segmentation and Optical Flow

Jingchun Cheng[1,2] Yi-Hsuan Tsai[2,4] Shengjin Wang[1*] Ming-Hsuan Yang[2,3]
[1]Tsinghua University [2]University of California, Merced
[3]NVIDIA Research [4]NEC Laboratories America
[1]chengjingchun@gmail.com, wgsgj@tsinghua.edu.cn [2]{ytsai2, mhyang}@ucmerced.edu

Abstract

This paper proposes an end-to-end trainable network, SegFlow, for simultaneously predicting pixel-wise object segmentation and optical flow in videos. The proposed SegFlow has two branches where useful information of object segmentation and optical flow is propagated bi-directionally in a unified framework. The segmentation branch is based on a fully convolutional network, which has been proved effective in image segmentation task, and the optical flow branch takes advantage of the FlowNet model. The unified framework is trained iteratively offline to learn a generic notion, and fine-tuned online for specific objects. Extensive experiments on both the video object segmentation and optical flow datasets demonstrate that introducing optical flow improves the performance of segmentation and vice versa, against the state-of-the-art algorithms.

1. Introduction

Video analysis has attracted much attention in recent years due to the numerous vision applications, such as autonomous driving [15, 9, 33], video surveillance [40, 10, 20] and virtual reality [1]. To understand the video contents for vision tasks, it is essential to know the object status (e.g., location and segmentation) and motion information (e.g., optical flow). In this paper, we address these problems simultaneously, i.e., video object segmentation and optical flow estimation, in that these two tasks have been known to be closely related to each other [41, 35]. Figure 1 illustrates the main idea of this paper.

For video object segmentation [25], it assumes that the object mask is known in the first frame, and the goal is to assign pixel-wise foreground/background labels through the entire video. To maintain temporally connected object segmentation, optical flow is typically used to improve the smoothness across the time [28]. However, flow estimation itself is a challenging problem and is often inaccurate,

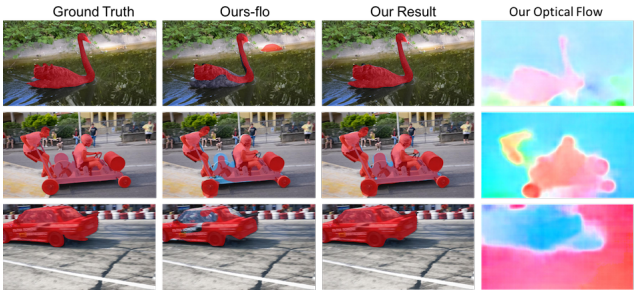

Figure 1. An illustration of the main idea in the proposed *SegFlow* model. Our model produces better segmentation results than the one without using the optical flow (Ours-flo), where the flow within the object is smooth and complete, providing a guidance to improve segmentation outputs.

and thus the provided information does not always help segmentation. For instance, when an object moves fast, the optical flow methods [2, 5, 37] are not effective in capturing the movement and hence generate incomplete flow within the object (see Figure 4 for an example). To overcome this problem, bringing the objectness information (i.e., segmentation) can guide the algorithm to determine where the flow should be smooth (within the object). A few algorithms have been developed to leverage both information from the objectness and motion discussed above. In [41], a method is proposed to simultaneously perform object segmentation and flow estimation, and then updates both results iteratively. However, the entire process is optimized online and is time-consuming, which limits the applicability to other tasks.

Based on the above observations, we propose a learning-based approach to jointly predict object segmentation and optical flow in videos, which allows efficient inference during testing. We design a unified, end-to-end trainable convolutional neural network (CNN), which we refer to as the *SegFlow*, that contains one branch for object segmentation and another one for optical flow. For each branch, we learn the feature representations for each task, where the segmentation branch focuses on the objectness and the optical flow one exploits the motion information. To bridge

*Corresponding Author

two branches to help each other, we propagate the learned feature representations bi-directionally. As such, these features from one branch can facilitate the other branch while obtaining useful gradient information during back-propagation.

One contribution of the proposed network is the bi-directional architecture that enables the communication between two branches, whenever the two objectives of the branches are closely related and can be jointly optimized. To train this joint network, a large dataset with both ground truths of two tasks (i.e., foreground segmentation and optical flow in this paper) is required. However, such dataset may not exist or is difficult to construct. To relax such constrains, we develop an iterative training strategy that only requires one of the ground truths at a time, so that the target function can still be optimized and converge to a solution where both tasks achieve reasonable results.

To evaluate our proposed network, we carry out extensive experiments on both the video object segmentation and optical flow datasets. We compare results on the DAVIS segmentation benchmark [29] with or without providing motion information, and evaluate the optical flow performance on the Sintel [6], Flying Chairs [12] and Scene Flow [27] datasets. In addition, analysis on the network convergence is presented to demonstrate our training strategy. We show that the bi-directional network through feature propagation performs favorably against state-of-the-art algorithms on both video object segmentation and optical flow tasks in terms of visual quality and accuracy.

The contributions of this work are summarized below:

- We propose an end-to-end trainable framework for simultaneously predicting pixel-wise foreground object segmentation and optical flow in videos.
- We demonstrate that optical flow and video object segmentation tasks are complementary, and can help each other through feature propagation in a bi-directional framework.
- We develop a method to train the proposed joint model without the need of a dataset that contains both segmentation and optical flow ground truths.

2. Related Work

Unsupervised Video Object Segmentation. Unsupervised methods aim to segment foreground objects without any knowledge of the object (e.g., an initial object mask). Several methods have been proposed to generate object segmentation via saliency [31, 11, 42], optical flow [4, 28] or superpixel [17, 46, 13]. To incorporate higher level information such as objectness, object proposals are used to track object segments and generate consistent regions through the video [22, 23]. However, these methods usually require heavy computational loads to generate region proposals and associate thousands of segments, making such methods only applicable to offline applications.

Semi-supervised Video Object Segmentation. Semi-supervised methods [14, 47, 26] assume an object mask in the first frame is known, and track this object mask through the video. To achieve this, existing approaches focus on propagating superpixels [19], constructing graphical models [25, 41] or utilizing object proposals [30]. Recently, CNN based methods [21, 7] are developed by combining offline and online training processes on static images. Although outstanding performance has been achieved, the segmentation results are not guaranteed to be smooth in the temporal domain. In this paper, we use CNNs to jointly estimate optical flow and provide the learned motion representations to generate consistent segmentations across time.

Optical Flow. It is common to apply optical flow to video object segmentation to maintain motion consistency. One category of the approaches is to solve a variational energy minimization problem [2, 5, 37] in a coarse-to-fine scheme. To better determine the correspondences between images, matching based optimization algorithms [43, 32] are developed, in which these methods usually require longer processing time. On the other hand, learning based methods are more efficient, which can be achieved via Gaussian mixture models [34], principle components [44] or convolutional networks [12, 27]. Considering the efficiency and accuracy, we apply the FlowNet [12] as our baseline in this work, while we propose to improve optical flow results by feeding the information from the segmentation network as guidance, which is not studied by the above approaches.

Fusion Methods. The joint problem of video segmentation and flow estimation has been studied by layered models [8, 38]. Nevertheless, such methods rely on complicated optimization during inference, thereby limiting their applications. Recently, significant efforts have been made along the direction of video object segmentation while considering optical flow. In [21], a network that uses pre-computed optical flow as an additional input to improve segmentation results is developed. Different from this work, our model only requires images as the input, and we aim to jointly learn useful motion representations to help segmentation.

Closest in scope to our work is the ObjectFlow algorithm (OFL) [41] that formulates an objective function to iteratively optimize segmentation and optical flow energy functions. However, this method is optimized online and is thus computationally expensive. In addition, it requires the segmentation results before updating the estimation for optical flow. In contrast, we propose an end-to-end trainable framework for simultaneously predicting pixel-wise foreground object segmentation and optical flow.

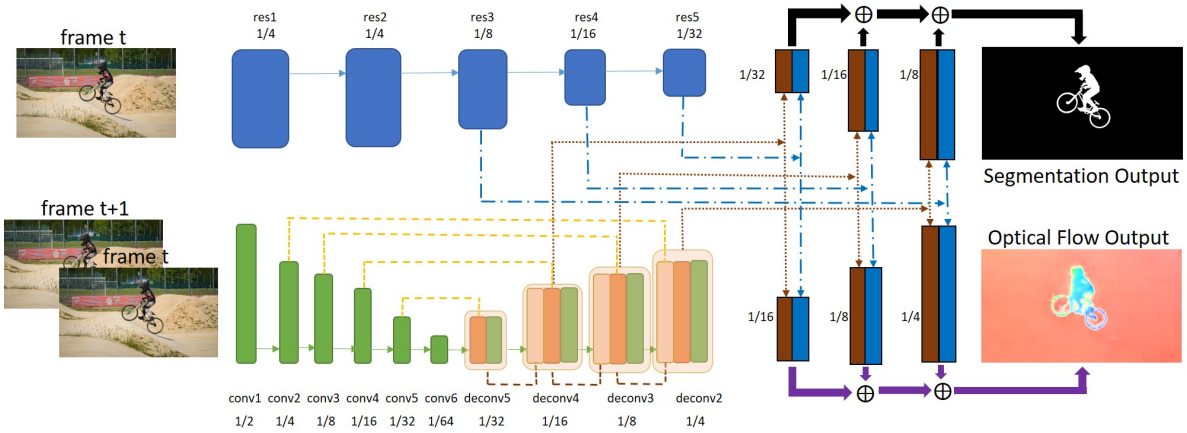

Figure 2. The proposed *SegFlow* architecture. Our model consists of two branches, the segmentation network based on a fully-convolutional ResNet-101 and the flow branch using the FlowNetS [12] structure. In order to construct communications between two branches, we design an architecture that bridges two networks during the up-sampling stage. Specifically, feature maps are propagated bi-directionally through concatenations at different scales with proper operations (i.e., up-sampling or down-sampling) to match the size of different features. Then an iterative training scheme is adopted to jointly optimize the loss functions for both segmentation and optical flow tasks.

3. SegFlow

Our goal is to segment objects in videos, as well as estimate the optical flow between frames. Towards this end, we construct a unified model with two branches, a segmentation branch based on fully-convolutional network, and an optical flow branch based on the FlowNetS [12].

Due to the lack of datasets with both segmentation and optical flow annotations, we initialize the weights of two branches from legacy models trained on different datasets, and optimize the *SegFlow* on segmentation and optical flow datasets via iterative offline training and online finetuning. In the following, we first introduce the baseline model of the segmentation and optical flow branch, and then explain how we construct the joint model using the proposed bi-directional architecture. The overall architecture of our proposed joint model is illustrated in Figure 2.

3.1. Segmentation Branch

Inspired by the effectiveness of fully-convolutional networks in image segmentation [24] and the deep structure in image classification [18, 36], we construct our segmentation branch based on the ResNet-101 architecture [18], but modified for binary (foreground and background) segmentation predictions as follows: 1) the fully-connected layer for classification is removed, and 2) features of convolution modules in different levels are fused together for obtaining more details during up-sampling.

The ResNet-101 has five convolution modules, and each consists of several convolutional layers, Relu, skip links and pooling operations after the module. Specifically, we draw feature maps from the 3-th to 5-th convolution modules after pooling operations, where score maps are with sizes of 1/8, 1/16, 1/32 of the input image size, respectively. Then these score maps are up-sampled and summed together for predicting the final output (upper branch in Figure 2).

A pixel-wise cross-entropy loss with the softmax function $\mathbb{E}$ is used during optimization. To overcome imbalanced pixel numbers between foreground and background regions, we use the weighted version as adopted in [45], and the loss function is defined as:

$$\mathcal{L}_s(X_t) = -(1-w)\sum_{i,j\in fg}\log\mathbb{E}(y_{ij}=1;\theta) \\ -w\sum_{i,j\in bg}\log\mathbb{E}(y_{ij}=0;\theta), \quad (1)$$

where i, j denotes the pixel location of foreground fg and background bg, y_{ij} denotes the binary prediction of each pixel of the input image X at frame t, and w is computed as the foreground-background pixel-number ratio.

3.2. Optical Flow Branch

Considering the efficiency and accuracy, we choose the FlowNetS [12] as our baseline for flow estimation. The optical flow branch uses an encoder-decoder architecture with additional skip links for feature fusions (feature concatenations between the encoder and decoder). In addition, a down-scaling operation is used at each step of the encoder, where each step of the decoder up-samples back the output (see the lower branch in Figure 2). Based on such structure, we find that it shares similar properties with the segmentation branch and their feature representations are in similar scales, which enables plausible connections to the segmentation model, and vice versa, where we will introduce in the next section.

To optimize the network, the optical flow branch uses an endpoint error (EPE) loss as adopted in [12], which is defined as the following:

$$\mathcal{L}_f(X_t, X_{t+1}) = \sum_{i,j}((u_{ij} - \delta_{u_{ij}})^2 + (v_{ij} - \delta_{v_{ij}})^2), \quad (2)$$

where u_{ij}, v_{ij} denotes the motion at pixel (i, j) of input images from X_t to X_{t+1}, and $\delta_{u_{ij}}$ and $\delta_{v_{ij}}$ are network predictions. We use the images at frame t and $t + 1$ as the computed optical flow should align with the segmentation output (e.g., object boundaries) at frame t, so that their information can be combined later naturally.

3.3. Bi-directional Model

In order to make communications between two branches as mentioned above, we propose a unified structure, *SegFlow*, to jointly predict segmentation and optical flow outputs. Therefore, the new optimization goal becomes to solve the following loss function that combines (1) and (2): $\mathcal{L}(X) = \mathcal{L}_s(X) + \lambda\mathcal{L}_f(X)$. As shown in Figure 2, our architecture propagates feature maps between two branches bi-directionally at different scales for the final prediction. For instance, features from each convolution module in the segmentation branch are first up-scaled (to match the size of optical flow features), and then concatenated to the optical flow branch. Similar operations are adopted when propagating features from segmentation to flow. Note that, a convolutional layer is also utilized (with the channel number equal to the output channel number) after fused features for network predictions, further regularizing the information from both the segmentation and optical flow branches.

Different from directly using final outputs to help both tasks [41], we here utilize information in the feature space. One reason is that our network is able to learn useful feature representations (e.g., objectness and motion) at different scales. In addition, with the increased model capacity but without adding too much burden for training the network, the joint model learns better representations than the single branch. For instance, the single flow network does not have the ability to learn representations similar to the segmentation branch, while our model provides the chance for two tasks sharing their representations. Note that, our bi-directional model is not limited to the current architecture or tasks, while it should be a generalized framework that can be applied to co-related tasks.

4. Network Implementation and Training

In this section, we present more details regarding how we train the proposed network. To successfully train the joint model, a large-scale dataset with both the segmentation and optical flow ground truths is required. However, it is not feasible to construct such a dataset. Instead, we develop a training procedure that only needs one of the ground truths at a time by iteratively updating both branches and gradually optimizing the target function. In addition, a data augmentation strategy is described for both tasks to enhance the diversity of data distribution and match the need of the proposed model.

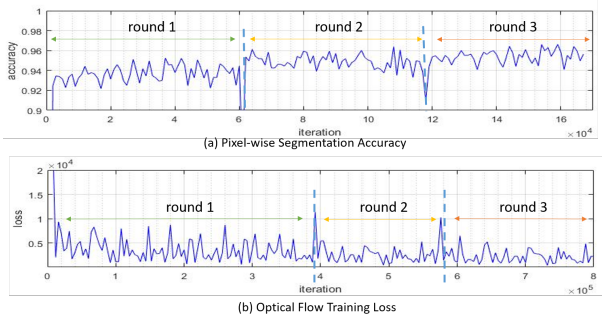

Figure 3. During offline training, (a) shows the training accuracy for object segmentation, while (b) presents the loss for optical flow, with respect to the number of training iterations (both results are obtained on a training subset). After three rounds, convergences can be observed for both segmentation and optical flow.

4.1. Network Optimization

First, we learn a generic model by iteratively updating both branches, where the goal of the segmentation network at this stage is to segment moving objects. To focus on a certain object (using the mask in the first frame), we then finetune the model for the segmentation branch on each sequence of the DAVIS dataset for online processing.

Iterative Offline Training. To start training the joint model, we initialize two branches using the weights from ResNet-101 [18] and FlowNetS [12], respectively. When optimizing the segmentation branch, we freeze the weights of the optical flow branch, and train the network on the DAVIS training set. We use SGD optimizer with batch size 1 for training, starting from learning rate 1e-8 and decreasing it by half for every 10000 iterations.

For training the optical flow branch, similarly we fix the segmentation branch and only update the weights in the flow network using the target optical flow dataset (described in Section 5.1). To balance the weights between two different losses, we use a smaller learning rate 1e-9 for the EPE loss in (2), addressing the λ in the combined loss in Section 3.3. Note that, to decide when to switch the training process to another branch, we randomly split a validation set and stop training the current branch when the error on the validation set reaches a convergence. In addition, this validation set is also used to select the best model with respect to the iteration number [12].

For this iterative learning process, each time the network focuses on one task in a branch, while obtaining useful representations from another branch through feature propagation. Then after switching to train another branch, better

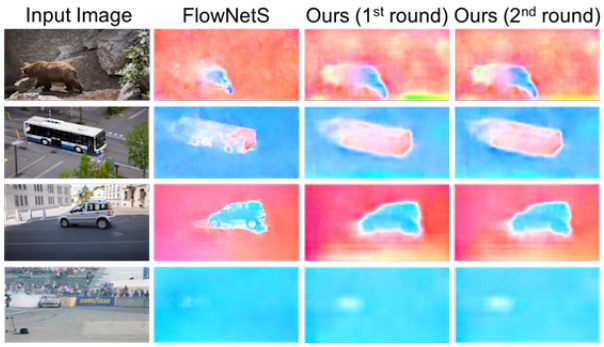

Figure 4. Iteratively improving optical flow results on DAVIS. Given an input image, we show the flow estimation from the initial model, FlowNetS [12], and our results during optimizing the *SegFlow* in the first and the second round. The results are gradually improved during optimization.

features learned from the previous stage are used in the branch currently optimized. We show one example of how the network gradually move toward a convergence by iteratively training both branches in Figure 3 (with three rounds). In addition, Figure 4 shows visual improvements during iteratively updating the flow estimation.

Online Training for Segmentation. The model trained offline is able to separate moving object from the video. To adapt the model on a specific object for online processing, we finetune the segmentation network using the object mask in the first frame on each individual sequence. Here, we call the process online in the semi-supervised setting, as the model is needed to update with the guidance of mask in the first frame before testing on the sequence.

Each mask is then augmented to multiple training samples for both branches to increase the data diversity (described in Section 4.2). After data augmentation, we use the same training strategy introduced in the offline stage with a fixed learning rate of 1e-10. At this stage, we note that the flow branch still provides motion representations to segmentation, but does not update the parameters.

4.2. Data Augmentation

Segmentation We use the pre-defined training set of the DAVIS benchmark [29] to train the segmentation branch. Since this training set is relatively small, we adopt affine transformations (i.e., shifting, rotation, flip) to generate one thousands samples for each frame. Since the flow branch requires two adjacent frames as the input, each affine transformation is carried out through the entire sequence to maintain the inter-frame (temporal) consistency during training (see Figure 5 for an example).

Optical Flow. The flow data during offline training step is generated as the approach described for segmentation. However, when training the online model using the first

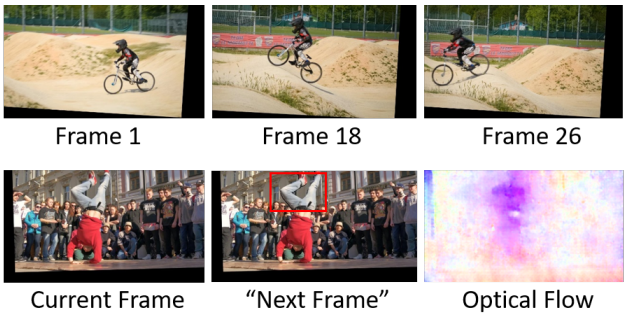

Figure 5. Examples for data augmentation. The first row shows the data augmentation for segmentation with the same transform through the video for maintaining the temporal consistency. The second row presents one example of the augmented flow, where the transform is applied on the object mask to simulate the slight movement in the "next frame" (highlighted within the red rectangle), where the optical flow shows the corresponding transform.

frame of a test set video, we have no access to its next frame. To solve this problem, we present an optical flow data augmentation strategy. First, we augment the first frame with the transformed method used in segmentation. Then, based on each image and its object mask, we simulate an object movement by slightly deforming the foreground object region to generate a synthesized "next frame". Since we only focus on the specific object at this online stage, the missing area caused by the object movement can be treated as occlusions and is left as empty (black) area. We find this synthesized strategy is effective for training without harming the network property (see Figure 5 for an example).

5. Experimental Results

We present the main experimental results with comparisons to the state-of-the-art video object segmentation and optical flow methods. More results and videos can be found in the supplementary material. The code and model are available at https://github.com/JingchunCheng/SegFlow.

5.1. Dataset and Evaluation Metrics

The DAVIS benchmark [29] is a recently-released high-quality video object segmentation dataset that consists of 50 sequences and 3455 annotated frames of real-world moving objects. The videos in DAVIS are also categorized according to various attributes, such as background clutter (BC), deformation (DEF), motion blur (MB), fast motion (FM), low resolution (LR), occlusion (OCC), out-of-view (OV), scale-variation (SV), appearance change (AC), edge ambiguity (EA), camera shake (CS), heterogeneous objects (HO), interesting objects (IO), dynamic background (DB), shape complexity (SC), as shown in Figure 6. We use the pre-defined training set to optimize our framework and its validation set to evaluate the segmentation quality.

For optical flow, we first use the MPI Sintel dataset [6] that contains 1041 pairs of images in synthesized scenes, with a $Clean$ version containing images without motion blur and atmospheric effects, and a $Final$ version of images with complicated environment variables. Second, we use the KITTI dataset [16], which has 389 pairs of flow images for real-world driving scenes. Finally, we use the Scene Flow dataset [27], which is a large-scale synthesized dataset recently established for flow estimation. Considering the realism, we use two subsets, Monkaa and Driving, where Monkaa has a collection of 24 video sequences with more than 34000 annotations for optical flow, and Driving has 8 videos with around 17000 annotations. Similar to Sintel, Driving and Monkaa both provide two versions: $Clean$ with clear images and $Final$ with more realistic ones.

Since the exact training and test sets are not specified in the Scene Flow dataset, we split our own sets for comparisons (training and validation sets do not intersect). For Monkaa, we use three videos ($eating\_ \times 2$, $flower\_storm\_ \times 2$, $lonetree\_ \times 2$) as the validation set, and use the rest of 21 sequences for training. For Driving, 7 videos are selected for training, and use the one with the attribute of $15mm\_focallength$, $scene\_forwards$ and $fast$ for testing. Note that, every video in both Monkaa and Driving has two views of $left$ and $right$, which results in 63400 training and 5720 validation pairs on Monkaa, and 32744 training and 2392 validation pairs on Driving.

To evaluate the segmentation quality, we use three measures (evaluation code from DAVIS website [29]): region similarity J, contour accuracy F and temporal stability T. For optical flow, we compute the average endpoint error from every pixel for evaluation.

5.2. Ablation Study on Segmentation

To analyze the necessity and importance of each step in the proposed framework, we carry out extensive ablation studies on DAVIS, and summarize the results in Table 1. We validate our method by comparing the proposed *SegFlow* to the ones without online training (**-ol**), iterative training (**-it**), offline training (**-of**) and flow branch (**-flo**). The detailed settings are as follows:

-ol: only uses the offline training without the supervised information in the first frame, which is categorized as unsupervised video object segmentation.

-it: only trains the model once for each of the segmentation and optical flow branches.

-of: trains the model directly on the testing video with the object mask in the first frame and its augmentations.

-flo: only uses the segmentation branch without the feature propagation from the flow network.

Table 1 shows that the offline training plays an important role in generating better results, improving the $Jmean$ by 21%. It demonstrates that the network needs a generic model to discover moving objects before online finetuning. The combined online and iterative strategy also improve the overall $Jmean$ by 7.9%. Compared to the model without using the flow branch, our joint model not only improves the $Jmean$ but also produces smooth results temporally, resulting in a significant improvement in $Tmean$ by 5.6%.

Table 1. Ablation study on the DAVIS validation set. We show comparisons of the proposed *SegFlow* model with different components removed, i.e., online-training (ol), offline-training (of), iterative learning (it), flow data augmentation (fda), optical flow branch (flo) and segmentation data augmentation (sda).

Method	Ours	-ol	-of	-ol -it	-fda	-flo	-flo -ol	-flo -ol -sda
J Mean ↑	**0.748**	0.674	0.538	0.669	0.739	0.724	0.654	0.606
J Recall ↑	**0.900**	0.814	0.575	0.803	0.891	0.882	0.787	0.677
J Decay ↓	0.137	0.062	0.227	0.005	0.124	0.119	0.021	**0.006**
F Mean ↑	**0.745**	0.667	0.515	0.658	0.741	0.735	0.640	0.604
F Recall ↑	**0.853**	0.771	0.540	0.765	0.839	0.841	0.750	0.717
F Decay ↓	0.136	0.051	0.251	0.043	0.122	0.132	0.017	**0.001**
T Mean ↓	**0.194**	0.276	0.302	0.279	0.225	0.250	0.354	0.335

We evaluate the effectiveness of our data augmentation steps in Table 1. Without the data augmentation for segmentation (**-sda**) and augmented flow data (**-fda**), the performance both degrades in terms of $Jmean$. In addition, the $Tmean$ is worse without augmenting flow data (**-fda**), which shows the importance of the synthesized data described in Section 4.2.

5.3. Segmentation Results

Table 2 shows segmentation results on the DAVIS validation set. We improve the performance by considering the prediction of the image and its flipping one, and averaging both outputs to obtain the final result, where we refer to as Ours[2]. Without adding much computational cost, we further boost the $Jmean$ with 1.3%. We compare the proposed *SegFlow* model with state-of-the-art approaches, including unsupervised algorithms (FST [28], CVOS [39], KEY [22], NLC [11]), and semi-supervised methods (OVOS [7], MSK [21], OFL [41], BVS [25]).

Among unsupervised algorithms, our *SegFlow* model with or without the flow branch both performs favorably against other methods with a significant improvement (more than 10% in $Jmean$). For semi-supervised methods, our model performs competitively against OSVOS [7] and MSK [21], where their methods require additional inputs (i.e., superpixels in OSVOS and optical flow in MSK[1]

[1]With image only as the input, the $Jmean$ of MSK [21] on the DAVIS validation set is 69.8, which is much lower than ours as 74.8.

Table 2. Overall results on the DAVIS validation set with the comparisons to unsupervised and semi-supervised methods.

		Semi-Supervised							Unsupervised						
	Measure	Ours[2]	Ours	Ours-flo	OSVOS	MSK	OFL	BVS	Ours-ol	Ours-flo-ol	OSVOS	FST	CVOS	KEY	NLC
J	Mean ↑	0.761	0.748	0.724	**0.798**	0.797	0.680	0.600	**0.674**	0.654	0.525	0.558	0.482	0.498	0.551
	Recall ↑	0.906	0.900	0.882	**0.936**	0.931	0.756	0.669	**0.814**	0.787	0.577	0.649	0.540	0.591	0.558
	Decay ↓	0.121	0.137	0.119	0.149	**0.089**	0.264	0.289	0.062	0.021	**-0.019**	0.000	0.105	0.141	0.126
F	Mean ↑	0.760	0.745	0.735	**0.806**	0.754	0.634	0.588	**0.667**	0.640	0.477	0.511	0.447	0.427	0.523
	Recall ↑	0.855	0.853	0.842	**0.926**	0.871	0.704	0.679	**0.771**	0.750	0.479	0.516	0.526	0.375	0.519
	Decay ↓	0.104	0.136	0.132	0.150	**0.090**	0.272	0.213	0.051	0.017	**0.006**	0.029	0.117	0.106	0.114
T	Mean ↓	**0.182**	0.194	0.250	0.376	0.211	0.217	0.345	0.276	0.354	0.538	0.343	**0.244**	0.252	0.414

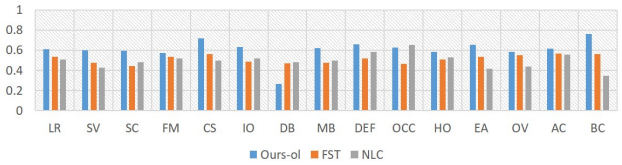

Figure 6. Attribute based evaluation on the DAVIS validation set using $Jmean$ compared with unsupervised methods.

with CRF refinement) to achieve higher performance, while our method only needs images as inputs. Furthermore, we show consistent improvements over the model without the flow branch, especially in the temporal accuracy ($Tmean$), which demonstrates that feature representations learned from the flow network help the segmentation.

Figure 6 shows the attributes-based performance ($Jmean$) for different methods. Our unsupervised method (offline training) performs well on all the attributes except for Dynamic Background (DB). One possible reason is that motion representations generated from the flow branch may not be accurate due to the complexity in the background. Figure 7 presents some example results for segmentation. With the flow branch jointly trained with segmentation, the model is able to recover the missing area of the object that is clearly a complete region from the flow estimation. A full comparison per sequence and more results are provided in the supplementary material.

5.4. Optical Flow Results

Table 3 and Table 4 show the average endpoint error of the proposed *SegFlow* model and the comparisons to other state-of-the-art methods, including our baseline model (FlowNetS) used in the flow branch. To validate the effectiveness of our joint training scheme, we use the pre-trained FlowNetS on the Flying Chair dataset [12] as the baseline, and finetune on the target dataset using the FlowNetS and our *SegFlow* model for comparisons.

We note that the data layer used in [12] is specifically designed for FlowNetS, and thus we cannot directly apply it to our model. Hence we report performance using various training data, where FlowNetS+ft denotes the results reported in [12] and FlowNetS+ft* denotes the model fine-tuned with the same training data as used in *SegFlow*. As a result, we show that our *SegFlow* model consistently improves endpoint errors against the results of FlowNetS+ft*, which validates the benefit of incorporating the information from the segmentation branch. On KITTI, SegFlow without any data augmentation even outperforms FlowNetS+ft that uses extensive data augmentation. However, we observe that our model slightly overfits to the data on Sintel, due to the need of data augmentation on a much smaller dataset than the others.

Table 3. Average endpoint errors for optical flow. FlowNetS+ft denotes the results presented in [12]. FlowNetS+ft* denotes FlowNetS trained with the same data as SegFlow+ft.

Method	Sintel *Clean*		Sintel *Final*		Chairs	KITTI	
	train	test	train	test	test	train	test
EpicFlow [32]	2.40	4.12	3.70	6.29	2.94	3.47	3.8
DeepFlow [43]	3.31	5.38	4.56	7.21	3.53	4.58	5.8
EPPM [3]	-	6.49	-	8.38	-	-	9.2
LDOF [5]	4.29	7.56	6.42	9.12	3.47	13.73	12.4
FlowNetS [12]	4.50	7.42	5.45	8.43	2.71	8.26	-
FlowNetS+ft	2.97	6.16	4.07	7.22	3.03	6.07	7.6
FlowNetS+ft*	3.31	7.89	4.26	8.50	-	7.37	8.7
SegFlow+ft	2.50	7.45	2.61	7.87	2.83	4.40	7.1

In Table 4, we also compare the results with SceneFlowNet [27] on the training and validation sets of Monkaa and Driving, and show that our method performs favorably against it. Figure 8 shows some visual comparisons of optical flow. Intuitively, the segmentation provides the information to guide the flow network to estimate the output that aligns with the segmentation output (e.g., the flow within the segmentation is smooth and complete). More results and analysis are provided in the supplementary material.

5.5. Runtime Analysis

For the model trained offline, the proposed *SegFlow* predicts two outputs (segmentation and optical flow) simulta-

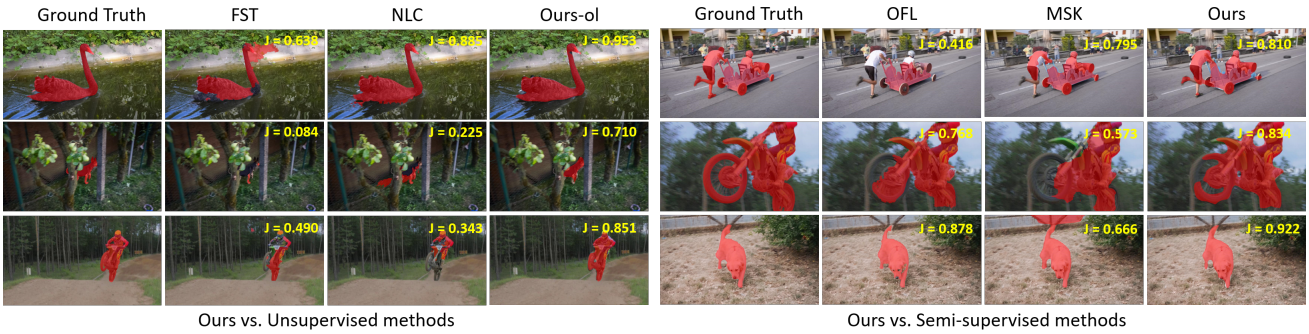

Figure 7. Qualitative results on the DAVIS validation set with comparisons to unsupervised and semi-supervised algorithms.

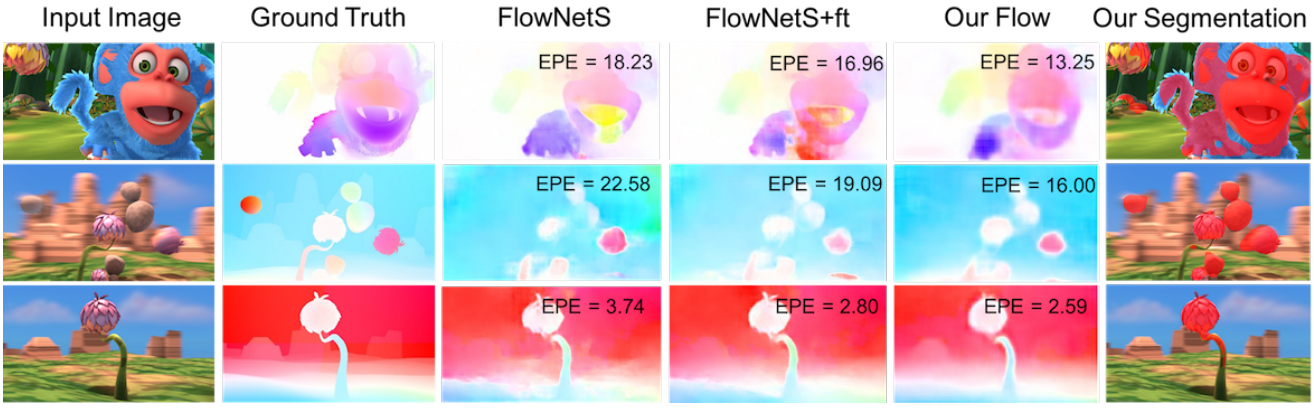

Figure 8. For each input image, we show the optical flow results of the baseline FlowNetS [12], fine-tuned FlowNetS and our *SegFlow* model on the Scene Flow dataset. Our method produces outputs with lower endpoint error, especially with the visual improvement within the object, in which the flow is smoother than the other methods due to the guidance from the segmentation network.

Table 4. Average endpoint errors on the Scene Flow dataset. The evaluations for train and val on the Monkaa and Driving datasets use both $forward$ and $backward$ samples, while evaluations on train+val use $forward$ ones with the comparison as reported in [27].

Method	Monkaa $Clean$			Monkaa $Final$		Driving $Clean$			Driving $Final$	
	train	val	train+val	train	val	train	val	train+val	train	val
SceneFlowNet [27]	-	-	6.54	-	-	-	-	23.53	-	-
FlowNetS [12]	5.60	10.51	6.15	5.48	10.47	13.29	66.93	23.90	13.14	67.15
FlowNetS+ft	4.93	8.40	5.01	4.37	8.44	10.31	52.67	18.22	10.38	52.20
SegFlow+ft	**4.06**	**7.94**	**4.49**	**3.78**	**7.90**	**9.17**	**37.91**	**14.35**	**9.41**	**37.93**

neously in 0.3 seconds per frame on a Titan X GPU with 12 GB memory. When taking the online training step into account, our system runs at 7.9 seconds per frame averaged over the DAVIS validation set. Compared to other methods such as OFL (30 seconds per frame for optical flow generation and 90 seconds per frame for optimization), MSK (12 seconds per frame) and OSVOS (more than 10 seconds per frame at its best performance), our method is faster and can output an additional result of optical flow.

6. Concluding Remarks

This paper proposes an end-to-end trainable network *SegFlow* for joint optimization of video object segmentation and optical flow estimation. We demonstrate that with this joint structure, both segmentation and optical flow can be improved via bi-directional feature propagations. To train the joint model, we relax the constraint of a large dataset that requires both foreground segmentation and optical flow ground truths by developing an iterative training strategy. We validate the effectiveness of our joint training scheme through extensive ablation studies and show that our method performs favorably on both the video object segmentation and optical flow tasks. The proposed model can be easily adapted to other architectures and can be used for joint training other co-related tasks.

Acknowledgments. This work is supported in part by the NSF CAREER Grant #1149783 and NSF IIS Grant #1152576.

References

[1] R. Anderson, D. Gallup, J. T. Barron, J. Kontkanen, N. Snavely, C. Hernández, S. Agarwal, and S. M. Seitz. Jump: virtual reality video. *ACM Transactions on Graphics (TOG)*, 35(6), 2016. 1

[2] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *ICCV*, 2007. 1, 2

[3] L. Bao, Q. Yang, and H. Jin. Fast edge-preserving patchmatch for large displacement optical flow. In *CVPR*, 2014. 7

[4] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010. 2

[5] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *PAMI*, 33(3):500–13, 2011. 1, 2, 7

[6] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 2, 6

[7] 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. 2, 6

[8] J. Chang and J. W. Fisher. Topology-constrained layered tracking with latent flow. In *ICCV*, 2013. 2

[9] C. Chen, A. Seff, A. Kornhauser, and J. Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *ICCV*, 2015. 1

[10] I. Cohen and G. Medioni. Detecting and tracking moving objects for video surveillance. In *CVPR*, 1999. 1

[11] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014. 2, 6

[12] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 2, 3, 4, 5, 7, 8

[13] F. Galasso, R. Cipolla, and B. Schiele. Video segmentation with superpixels. In *ACCV*, 2012. 2

[14] F. Galasso, N. Nagaraja, T. Cardenas, T. Brox, and B. Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *ICCV*, 2013. 2

[15] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 1

[16] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 6

[17] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010. 2

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 4

[19] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, 2014. 2

[20] I. N. Junejo, O. Javed, and M. Shah. Multi feature path modeling for video surveillance. In *ICPR*, 2004. 1

[21] A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 2, 6

[22] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, 2011. 2, 6

[23] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013. 2

[24] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 3

[25] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, 2016. 1, 2, 6

[26] N. S. Nagaraja, F. Schmidt, and T. Brox. Video segmentation with just a few strokes. In *ICCV*, 2015. 2

[27] N.Mayer, E.Ilg, P.Häusser, P.Fischer, D.Cremers, A.Dosovitskiy, and T.Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 2, 6, 7, 8

[28] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013. 1, 2, 6

[29] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 2, 5, 6

[30] F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung. Fully connected object proposals for video segmentation. In *CVPR*, 2015. 2

[31] E. Rahtu, J. Kannala, M. Salo, and J. Heikkilä. Segmenting salient objects from images and videos. In *ECCV*, 2010. 2

[32] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015. 2, 7

[33] G. Ros, S. Ramos, M. Granados, A. Bakhtiary, D. Vazquez, and A. M. Lopez. Vision-based offline-online perception paradigm for autonomous driving. In *WACV*, 2015. 1

[34] D. Rosenbaum, D. Zoran, and Y. Weiss. Learning the local statistics of optical flow. In *NIPS*, 2013. 2

[35] L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black. Optical flow with semantic segmentation and localized layers. In *CVPR*, 2016. 1

[36] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. In *ICML*, 2015. 3

[37] D. Sun, S. Roth, and M. J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *IJCV*, 106(2):115–137, 2014. 1, 2

[38] D. Sun, J. Wulff, E. B. Sudderth, H. Pfister, and M. J. Black. A fully-connected layered model of foreground and background flow. In *CVPR*, 2013. 2

[39] B. Taylor, V. Karasev, and S. Soatto. Causal video object segmentation from persistence of occlusions. In *CVPR*, 2015. 6

[40] Y.-L. Tian, M. Lu, and A. Hampapur. Robust and efficient foreground analysis for real-time video surveillance. In *CVPR*, 2005. 1

[41] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *CVPR*, 2016. 1, 2, 4, 6

[42] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, 2015. 2

[43] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *ICCV*, 2013. 2, 7

[44] J. Wulff and M. J. Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *CVPR*, 2015. 2

[45] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015. 3

[46] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *ECCV*, 2012. 2

[47] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, 2013. 2