Fast Semantic Segmentation on Video Using Block Motion-Based Feature Interpolation

Samvit Jain and Joseph E. Gonzalez

University of California, Berkeley CA 94704, USA
{samvit,jegonzal}@eecs.berkeley.edu

Abstract. Convolutional networks optimized for accuracy on challenging, dense prediction tasks are often prohibitively slow to run on each frame in a video. The spatial similarity of nearby video frames, however, suggests opportunity to reuse computation. Existing work has explored basic feature reuse and feature warping based on optical flow, but has encountered limits to the speedup attainable with these techniques. In this paper, we present a new, two part approach to accelerating inference on video. First, we propose a fast feature propagation technique that utilizes the block motion vectors present in compressed video (e.g. H.264 codecs) to cheaply propagate features from frame to frame. Second, we develop a novel feature estimation scheme, termed feature interpolation, that fuses features propagated from enclosing keyframes to render accurate feature estimates, even at sparse keyframe frequencies. We evaluate our system on the Cityscapes and CamVid datasets, comparing to both a frame-by-frame baseline and related work. We find that we are able to substantially accelerate semantic segmentation on video, achieving twice the average inference speed as prior work at any target accuracy level.

Keywords: semantic segmentation · efficient inference · video segmentation · video compression · H.264 video

1 Introduction

Semantic segmentation, the task of assigning each pixel in an image to a semantic object class, is a problem of long-standing interest in computer vision. Since the first paper to suggest the use of fully convolutional networks to segment images [7], increasingly sophisticated architectures have been proposed, with the goal of segmenting more complex images, from larger, more realistic datasets, at higher accuracy [1–3, 6, 9, 10]. The result has been a ballooning in both model size and inference times, as the core feature networks, borrowed from image classification models, have grown in layer depth and parameter count, and as the cost of a forward pass through the widest convolutional layers, a function of the size and detail of the input images, has risen in step. As a result, state-of-the-art networks today require between 0.5 to 3.0 seconds to segment a *single*, high-resolution image (e.g. 2048×1024 pixels) at competitive accuracy [5, 11].

At the same time, a new target data format for semantic segmentation has emerged: video. The motivating use cases include both batch settings, where

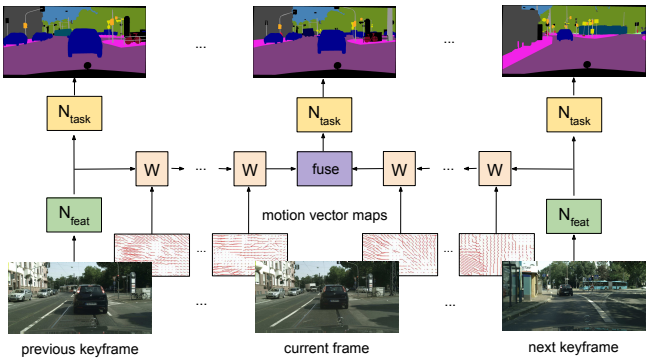


Fig. 1. Feature interpolation warps (W) and fuses the features of enclosing keyframes to generate accurate feature estimates for intermediate frames, using the block motion vectors in compressed (e.g. H.264) video.

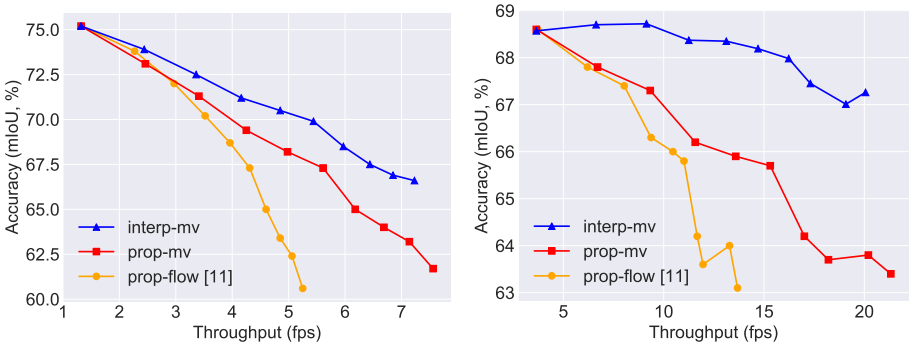
video is segmented in bulk to generate training data for other models (e.g. autonomous control systems), and streaming settings, where high-throughput video segmentation enables interactive analysis of live footage (e.g. at surveillance sites). Video here consists of long sequences of images, shot at high frame rates (e.g. 30 frames per second) in complex environments (e.g. urban cityscapes) on modern, high-definition cameras (i.e. multi-megapixel). Segmenting individual frames at high accuracy still calls for the use of competitive image segmentation models, but the inference cost of these networks precludes their naïve deployment on every frame in a multi-hour raw video stream.

A defining characteristic of realistic video is its high level of temporal continuity. Consecutive frames demonstrate significant spatial similarity, which suggests the potential to reuse computation across frames. Building on prior work, we exploit two observations: 1) higher-level features evolve more slowly than raw pixel content in video, and 2) feature computation tends to be much more expensive than task computation across a range of vision tasks (e.g. object detection, semantic segmentation) [8, 11]. Accordingly, we divide our semantic segmentation model into a deep feature network and a cheap, shallow task network [11]. We compute features only on designated keyframes, and propagate them to intermediate frames, by warping the feature maps with a frame-to-frame motion estimate. The task network is executed on all frames. Given that feature warping and task computation is much cheaper than feature extraction, a key parameter we aim to optimize is the interval between designated keyframes.

Here we make two key contributions to the effort to accelerate semantic segmentation on video. First, noting the high level of data redundancy in video, we successfully utilize an artifact of compressed video, block motion vectors, to cheaply propagate features from frame to frame. Unlike other motion estimation techniques, which introduce extra computation on intermediate frames, block motion vectors are freely available in modern video formats, making for a simple, fast design. Second, we propose a novel feature estimation scheme that enables

the features for a large fraction of the frames in a video to be inferred accurately and efficiently (see Fig. 1). The approach works as follows: when computing the segmentation for a keyframe, we also precompute the features for the *next* designated keyframe. Features for all subsequent intermediate frames are then computed as a *fusion* of features warped forward from the last visited keyframe, and features warped backward from the incoming keyframe. This procedure thus implements an *interpolation* of the features of the two closest keyframes.

We evaluate our framework on the Cityscapes and CamVid datasets. Our baseline consists of running a state-of-the-art segmentation network, DeepLab [3], on every frame, a setup that achieves published accuracy [4], and a throughput of 1.3 frames per second (fps) on Cityscapes and 3.6 fps on CamVid. Our improvements come in two phases. Firstly, our use of block motion vectors for feature propagation allow us to cut inference time on intermediate frames by 53%, compared to approaches based on optical-flow, such as [11]. Second, our bi-directional feature warping and fusion scheme enables substantial accuracy improvements, especially at high keyframe intervals. Together, the two techniques allow us to operate at *twice the average inference speed* as the fastest prior work, at any target level of accuracy (see Figure 2). For example, if we are willing to tolerate *no worse* than 65 mIoU on our CamVid video stream, we are able to operate at a throughput of 20.1 fps, compared to the 8.0 fps achieved by the forward flow-based propagation from [11] (see Table 1). Overall, even when operating in high accuracy regimes (e.g. within 3% mIoU of the baseline), we are able to accelerate segmentation on video by a factor of 2-6×.



(a) **Cityscapes**. (b) **CamVid**. Data from Table 1.

Fig. 2. Accuracy (**avg.**) vs. throughput on Cityscapes and CamVid for three schemes: (1) optical-flow based feature propagation [11] (**prop-flow**), (2) motion vector-based feature propagation (**prop-mv**), and (3) motion vector-based feature interpolation (**interp-mv**).

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. In: PAMI (2017)

Table 1. Accuracy and throughput on **CamVid** for three schemes: (1) optical-flow based feature propagation [11] (**prop-flow**), (2) motion vector-based feature propagation (**prop-mv**), and (3) motion vector-based feature interpolation (**interp-mv**). Average accuracy refers to mean accuracy across all labeled frames in the test set. Minimum accuracy refers to accuracy on frames farthest away from keyframes.

		keyframe interval									
Metric	Scheme	1	2	3	4	5	6	7	8	9	10
mIoU (**avg.**)	prop-flow	68.6	67.8	67.4	66.3	66.0	65.8	64.2	63.6	64.0	63.1
(%)	prop-mv	68.6	67.8	67.3	66.2	65.9	65.7	64.2	63.7	63.8	63.4
	interp-mv	**68.6**	**68.7**	**68.7**	**68.4**	**68.4**	**68.2**	**68.0**	**67.5**	**67.0**	**67.3**
mIoU (**min.**)	prop-flow	68.5	67.0	66.2	64.9	63.6	62.7	61.3	60.5	59.7	58.7
(%)	prop-mv	68.5	67.0	65.9	64.7	63.4	62.7	61.4	60.8	60.0	59.3
	interp-mv	**68.5**	**68.6**	**68.4**	**68.2**	**67.9**	**67.4**	**67.0**	**66.4**	**66.1**	**65.7**
throughput	prop-flow	3.6	6.2	8.0	9.4	10.5	11.0	11.7	12.0	13.3	13.7
(fps)	prop-mv	3.6	6.7	9.3	11.6	13.6	15.3	17.0	18.2	20.2	21.3
	interp-mv	3.6	6.6	9.1	11.3	13.1	14.7	16.2	17.3	19.1	20.1

2. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: ICLR (2016)
3. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. In: PAMI (2017)
4. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: ICCV (2017)
5. Gadde, R., Jampani, V., Gehler, P.V.: Semantic video cnns through representation warping. In: ICCV (2017)
6. Lin, G., Milan, A., Shen, C., Reid, I.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: CVPR (2017)
7. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
8. Shelhamer, E., Rakelly, K., Hoffman, J., Darrell, T.: Clockwork convnets for video semantic segmentation. In: Video Semantic Segmentation Workshop at ECCV (2016)
9. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: ICLR (2016)
10. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017)
11. Zhu, X., Xiong, Y., Dai, J., Yuan, L., Wei, Y.: Deep feature flow for video recognition. In: CVPR (2017)