Low-Latency Video Semantic Segmentation

Hongzhi Liu

Jun 3, 2018

Abstract

As is known to all, autonomous driving has become a hot topic for people in recent years and there is still a long way for the smart car to really go on the road. Fortunately, we have made some breakthroughs in some areas. Today, I read a thesis written by Yule Li, who is from Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences. His team develop a framework for video semantic segmentation to tackle the combined challenge that applying segmentation techniques to video-based applications. The proposed framework obtained competitive performance compared to the state of the art.

1. Overview of Video Segmentation Framework

Semantic segmentation is a task to divide observed scenes into semantic regions, having been an active research topic in computer vision. The advances in deep learning [4] and in particular the development of Fully Convolutional Network (FCN) [6] have brought the performance of this task to a new level. The challenges of video-based semantic segmentation consist in two aspects. On one hand, videos usually involve significantly larger volume of data compared to images. And on the other hand, many real-world systems that need video segmentation. Previous efforts on video semantic segmentation mainly fall into two classes, namely *high-level modeling* and *feature-level propagation*. The former [2] integrates frame-wise analysis via a sequential model but are unable to reduce the computing cost. The latter attempts to reuse the features in preceding frames to accelerate computation. However, it lacks the flexibility of handling complex variations.

In the paper, Yule Li and his team present a framework that achieves low-latency video segmentation [5] in which they introduce two new components that are a network using spatially variant convolution to propagate features adaptively and an adaptive scheduler to reduce the overall computing cost and ensure low latency. This way not only leads to more efficient use of computational resources but also reduce the maximum latency.

2. Video Segmentation Framework

Previous works usually select key frames based on fixed intervals [9] or simple heuristics [8], and propagate features based on optical flows that are costly to compute or CNNs with fixed kernels. Such methods often lack the capability of handling complex variations in videos. Hence, Li and his team try to select key frames and propagate features more effectively by exploiting the information contained in the low-level features, while maintaining a relatively low computing cost.

2.1. Overall Pipeline of the Framework

The team use a deep convolutional network (ResNet101 [3] in their implementation) to extract visual features from frames. They divide the network into two parts, the lower part S_l and the higher-part S_h. The low-level features derived from S_l will be used for selecting key frames and controlling how high-level features are propagated. Fig. 1 shows the overall pipeline of our framework.

When at runtime, the framework will feed the first frame I^0 through the entire CNN and obtain both low-level and high-level features to initialize the entire procedure. At a later time step t, it performs the computation adaptively. In particular, it first feeds the corresponding frame I^t to S_l, and computes the low-level features F_l^t. Based on F_l^t, it decides whether to treat I^t as a new key frame, depending on how much it deviates from the previous one. If the decision is yes, it will continue to feed F_l^t to S_h to compute the high-level features F_h^t , and then the segmentation map. Otherwise, it will feed F_l^t to a *kernel predictor*, obtain a set of convolution kernels therefrom, and use them to propagate the high-level features from the previous key frame via spatially variant convolution.

2.2. Adaptive Selection of Key Frames

An important step in Li's pipeline which is mentioned above is to decide which frames are the key frames. A good strategy is to select key frames more frequently when the video is experiencing rapid changes, while reducing the computation when the observed scene is stable. According to the rationale above, a natural criterion for judging whether

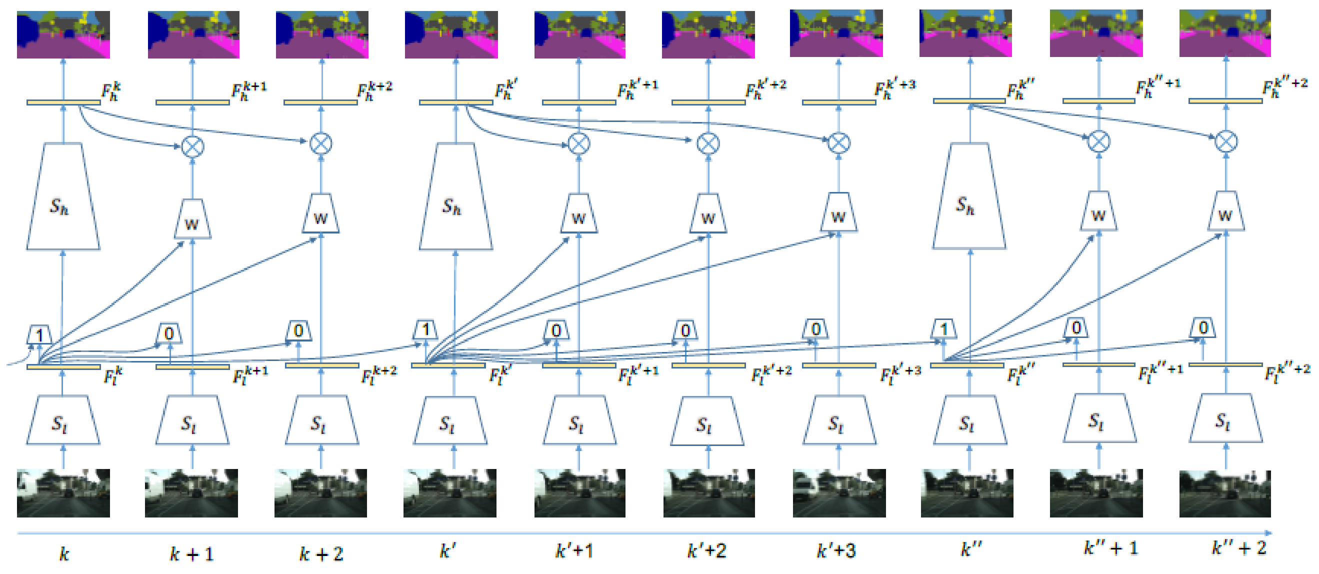

Figure 1. The overall pipeline. At each time step t, the lower-part of the CNN S_l first computes the low-level features F_l^t. Based on both F_l^k (the low-level features of the previous key frame) and F_l^t, the framework will decide whether to set I^t as a new key frame. If yes, the high-level features F_h^t will be computed based on the expensive higher-part S_h; otherwise, they will be derived by propagating from F_h^k using spatially variant convolution. The high-level features, obtained in either way, will be used in predicting semantic labels.

a frame should be chosen as a new key frame is the deviation of its segmentation map from that of the previous key frame.

Motivated by the observation that strong correlation exists between features and the values, the team devise a small neural network to make the prediction. Let k and t be the indexes of two frames, this network takes the differences between their low-level features. Specifically, their design of this prediction network comprises two convolutional kernels with 256 channels, a global pooling and a fully-connected layer that follows. In runtime, at time step t, Li uses this network to predict the deviation from the previous key frame, after the low-level features are extracted. As shown in Fig. 2, they observed that the predicted deviation would generally increases over time. If the predicted deviation goes beyond a pre-defined threshold, the team set the current frame as a key frame, and computes its highlevel features with S_h, the higher part of the CNN.

2.3. Adaptive Feature Propagation

The team propose to propagate the features by spatially variant convolution, that is, using convolution to express linear combinations of neighbors, with the kernels varying across sites. Let the size of the kernels be $H_K \times H_K$, then the propagation from the high-level features of the previous key frame (F_h^k) to that of the current frame (F_h^t) can be ex-

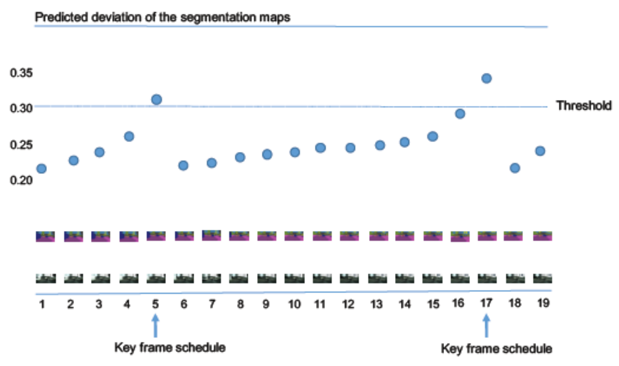

Figure 2. Adaptive key frame selection. As we proceed further away from the key frame, the predicted deviation of the segmentation map, depicted with blue dots, gradually increases. In Li's scheme, when the deviation goes beyond a pre-defined threshold, the current frame will be selected as a new key frame.

pressed as Equation 1:

$$F_h^t(l,i,j) = \sum_{u=-\Delta}^{\Delta} \sum_{v=-\Delta}^{\Delta} W_{ij}^{(k,t)}(u,v) \cdot F_h^k(l, i-u, j-v). \tag{1}$$

Here, $\Delta = [H_K/2]$, $F_h^t(l,i,j)$ is the feature value at $(i;j)$ of the l-th channel in F_h^t, $W_{ij}^{(k,t)}$ is an $H \times H$ kernel used to compute the feature at $(i;j)$ when propagating from F_h^k to F_h^t. Note that the kernel values are to assign weights to different neighbors, which are dependent on the feature lo-

Table 1. Comparison with state-of-the-art on Cityscapes dataset.

Method	mIOU	Avg RT	Latency
Clockwork Net [8]	67.7%	141ms	360ms
Deep Fea. Flow [9]	70.1%	273ms	654ms
GRFP(5) [7]	69.4%	470ms	470ms
baseline	80.2%	360ms	360ms
AFP + fix schedule	75.26%	151ms	360ms
AFP + AKS	76.84%	171ms	380ms
AFP + AKS + LLS	75.89%	119ms	119ms

cation $(i; j)$ but shared across all channels.

To increase the robustness against scene changes, we fuse the low-level features F_l^t with the propagated high level feature F_h^t for predicting the labels. The AdaptNet is jointly learned with the weight predictor in model training. In this way, the framework can learn to exploit the complementary natures of both components more effectively.

2.4. Low-Latency Scheduling

Based on the framework presented above, the team devise a new scheduling scheme that can substantially reduce the maximum latency. The key of our approach is to introduce a fast track at key frames. Specifically, when a frame It is decided to be a key frame, this scheme will compute the segmentation of this frame through the fast track, *i.e.* via feature propagation. The high-level features resulted from the fast track are temporarily treated as the new key frame feature and placed in the cache. In the mean time, a background process is launched to compute the more accurate version of F_h^t via the slow track S_h, without blocking the main procedure. When the computation is done, this version will replace the cached features.

3. Experimental Results of Framework

Li evaluates their framework on Cityscapes [1] and the method outperforms previous methods significantly with lowest latency. The team compared their low-latency video semantic segmentation framework with recent state-of-the-art methods, following their evaluation protocol. Table 1 shows the quantitative comparison.

From the results, we can see that other methods proposed recently fall short in certain aspects. In particular, Clockwork Net [8] has the same latency, but at the cost of significantly dropped performance. DFF [9] maintains a better performance, but at the expense of considerably increased computing cost and dramatically larger latency. Their final result, with low latency schedule for video segmentation, outperforms previous methods by a large margin. This validates the effectiveness of our entire design.

References

[1] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 3

[2] M. Fayyaz, M. H. Saffar, M. Sabokrou, M. Fathy, R. Klette, and F. Huang. STFCN: Spatio-Temporal FCN for semantic video segmentation. *arXiv preprint arXiv:1608.05971*, 2016. 1

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[5] Y. Li, J. Shi, and D. Lin. Low-latency video semantic segmentation. In *CVPR*, 2018. 1

[6] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[7] D. Nilsson and C. Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. *arXiv preprint arXiv:1612.08871*, 2016. 3

[8] E. Shelhamer, K. Rakelly, J. Hoffman, and T. Darrell. Clockwork convnets for video semantic segmentation. In *ECCV*, 2016. 1, 3

[9] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei. Deep feature flow for video recognition. In *CVPR*, 2017. 1, 3